A Novel Iron Loss Reduction Technique for Distribution Transformers Based on a Combined Genetic Algorithm—Neural Network Approach

Pavlos S. Georgilakis, *Member, IEEE*, Nikolaos D. Doulamis, *Student Member, IEEE*, Anastasios D. Doulamis, *Student Member, IEEE*, Nikos D. Hatziargyriou, *Senior Member, IEEE*, and Stefanos D. Kollias, *Member, IEEE*

***Abstract*—This paper presents an effective method to reduce the iron losses of wound core distribution transformers based on a combined neural network- genetic algorithm approach. The originality of the work presented in this paper is that it tackles the iron loss reduction problem during the transformer production phase, while previous works were concentrated on the design phase. More specifically, neural networks effectively use measurements taken at the first stages of core construction in order to predict the iron losses of the assembled transformers, while genetic algorithms are used to improve the grouping process of the individual cores by reducing iron losses of assembled transformers. The proposed method has been tested on a transformer manufacturing industry. The results demonstrate the feasibility and practicality of this approach. Significant reduction of transformer iron losses is observed in comparison to the current practice leading to important economic savings for the transformer manufacturer.**

***Index Terms*—Core grouping process, decision trees, genetic algorithms, intelligent core loss modeling, iron loss reduction, neural networks.**

I. Introduction

In today's competitive market environment, there is an urgent need for a transformer manufacturing industry to improve transformer efficiency and to reduce cost, since high quality, low cost products have become the key to survival. Transformer efficiency is improved by reducing *load* and *iron losses*. To reduce load losses, the designer can do one or more of the following: use lower loss conductor materials or decrease the current path length or the current density. On the other hand, the designer can reduce iron losses by using lower loss core materials or reducing core flux density or flux path length [1]. In general, attempts to reduce load losses cause increase of iron losses and vice versa [1]. As a result, before deciding the optimal design method, it is necessary to determine which of the two losses should be minimized. Usually, the transformer users (e.g., electric utilities) specify a desired level of iron losses (guaranteed losses) to determine the transformer quality. This is due to the fact that the accumulated iron losses in a distribution network are high since a large amount of distribution transformers is involved. In addition, iron losses appear 24 hours per day, every day, for a continuously energized transformer. Thus, it is in general preferable to design a transformer at minimum iron losses [2] and this is addressed in this paper.

Initially, transformers are designed so that their iron losses are equal (with perhaps a safety margin) to the guaranteed ones. In practice, however, transformer actual iron losses deviate from the designed (theoretical) ones due to constructional defects, which appear during the production phase. Reduction of transformer actual losses, by minimizing the effect of constructional defects, is a very important task for a manufacturing industry. In particular,

1) it increases the reliability of the manufacturer;
2) it reduces the material cost, since smaller safety margin is used during the transformer design phase;
3) it helps the manufacturer not to pay loss penalties.

The latter occurs in case the actual transformer losses are greater (usually 15%) than the guaranteed ones. In general, it is clear that manufacturers, who are able to offer transformers of better quality (lower losses) at the same price, will increase their market share.

Several works have been proposed in the literature for the estimation of transformer iron losses during the design phase. These approaches can be grouped into two main categories. The first group is based on the arithmetic analysis of the electromagnetic field of the transformer cores, while the second group uses iron loss models based on experimental observations. In the former approach, finite elements and finite difference methods are mainly used. The potentials of the electromagnetic fields are calculated, by creating mesh models of the transformer geometry, and using several field parameters, such as the magnetic flux distribution. This analysis is very important during the transformer design phase, when the manufacturer needs to check the correctness of the transformer drawings. Key works adopting this approach are provided next. In [3], the three-dimensional (3-D) leakage fields are estimated and in [4] the spatial loss distribution is investigated using a generic

Manuscript received December 9, 2000; revised December 20, 2000. This work was supported by the General Secretariat of Research and Technology of Greece within the YPER '94 Research Programme. This paper was recommended by Associate Editor J. Lee.

P. S. Georgilakis is with Schneider Electric AE, Elvim Plant, Inofyta Viotia, Greece (e-mail: pavlos_georgilakis@mail.schneider.fr).

N. D. Doulamis, A. D. Doulamis, and S. D. Kollias are with the Digital Signal Processing Laboratory, Department of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece (e-mail: ndoulam@image.ece.ntua.gr).

N. D. Hatziargyriou is with the Electric Energy Systems Laboratory, Department of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece (e-mail: nh@power.ece.ntua.gr).

Publisher Item Identifier S 1094-6977(01)03527-1.

two-dimensional (2-D) finite difference method. Three-dimensional magnetic-field calculations are performed in [5] to evaluate several transformer parameters, while in [6] the effects of a number of core production attributes on core loss performance have been examined. Other works, in this category, model three-phase transformers based on the equivalent magnetic circuit of their cores [7], [8].

In the second approach, experimental curves are usually extracted using a large number of measurements to investigate the effect of several transformer parameters on iron losses [2]. However, due to the continuous evolution both of technical characteristics of the magnetic materials and the design of cores, the experimental curves should systematically be reconstructed when data change. Alternatively, linear or simple nonlinear models are used in order to relate transformer iron losses to the magnetic induction and geometrical properties of the magnetic core [9]–[11]. The parameters of these models are estimated based on experimental observations. However, these methods provide satisfactory results only for data (transformers) or conditions on which they have been estimated. Their performance deteriorates severely in case of new samples, which are not included in the "training" set.

Although, all the aforementioned approaches (theoretical or experimental) provide a sufficient framework for the calculation of transformer iron losses during the design phase, they do not take into account the effect of constructional defects, which cause the deviation of the actual losses from the theoretical ones. More specifically, it has been found that the maximum divergence between the theoretical and actual iron losses of a specific production batch could as high as ±10%. These deviations are to a great extent attributed to the deviations of the actual core characteristics from the designed ones. For example, the maximum deviation of the iron losses of the individual cores can reach up to ±15%, while the maximum deviation of the core weights up to ±1.5% [12].

In this paper, reduction of transformer iron losses is achieved during the transformer production phase. In particular, an optimal method is presented to estimate the most appropriate arrangement of individual cores, which yields transformers of minimum actual iron losses. This is achieved by compensating the constructional defects, which appear in the production phase. The method is relied on a combined neural network-genetic algorithm (GA) scheme. The goal of the neural network architecture is to predict transformer actual losses prior to their assembly. For this reason, several measurements (attributes) are obtained during the transformer production phase. A decision tree methodology is adopted next to select the most significant attributes, which are fed as inputs to the neural network. A genetic algorithm is finally applied to estimate the optimal arrangement of individual cores that assemble a transformer. In our case, optimality means that the iron losses of all constructed transformers in a production batch should be as minimal as possible. The genetic algorithm exploits information provided by the neural network architecture to perform the minimization task. In particular, the network predicts the transformer quality (iron losses) of a given core arrangement. The proposed scheme has been applied in a transformer manufacturing industry and the results reveal a significant economic benefit.

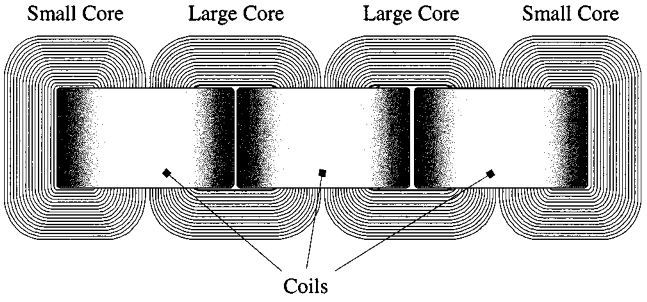


Fig. 1. Assembled active part of a wound core transformer.

This paper is organized as follows. Section II describes the current practice for estimating iron losses and for grouping the individual cores. Section III presents a general overview of the proposed method. Section IV presents the prediction of iron losses using neural networks. In particular, in this section we describe the constructive algorithm used to train the network, the method applied for attribute selection and the weight adaptation algorithm used for improving the network performance. Section V presents the reduction of iron losses using a GA. In this section, we also discuss issues related to the GA convergence. Finally, Section VI shows the results and economic benefits obtained from the application of the proposed techniques in a transformer industry. Section VII concludes the paper.

II. Current Practice for Predicting Iron Losses and Grouping Individual Cores

A three-phase wound core distribution transformer is constructed by assembling two small and two large individual cores, according to the arrangement described in Fig. 1. In particular, the four cores are placed as follows: a small core, followed by two large cores, followed by another small core (from left to right). The window width of large cores is twice of the width of small cores. Based on the previous arrangement, N three-phase transformers are constructed from $2*N$ small and $2*N$ large individual cores. Let us denote as V_s (V_l) the set of all $2*N$ small (large) cores. A transformer is represented by a vector $\mathbf{t}_i$, the elements of which corresponds to the four individual cores that assemble the transformer

$$\mathbf{t}_i = [s_i^l \, l_i^l \, l_i^r s_i^r]^T. \tag{1}$$

Variables $s_i^l, s_i^r \in V_s$ represent the left and right small core of transformer $\mathbf{t}_i$, while $l_i^l, l_i^r \in V_l$ the left and right large core, respectively. Since only one core (small or large) can be assigned to one transformer and one position (left or right), the following restrictions are held:

$$s_i^l \neq s_i^r, \quad l_i^l \neq l_i^r \tag{2a}$$

$$s_k^{\{l,r\}} \neq s_i^{\{l,r\}}, \quad l_k^{\{l,r\}} \neq l_i^{\{l,r\}}, \quad \text{with} \quad k \neq i \tag{2b}$$

where $s_i^{\{l,r\}}$ $(l_i^{\{l,r\}})$ indicates the small (large) core in the left or right position for the transformer $\mathbf{t}_i$. In the following subsection, we analyze how the iron losses are estimated in current practice, while Section II-B presents the current core grouping process, used to assemble a transformer.

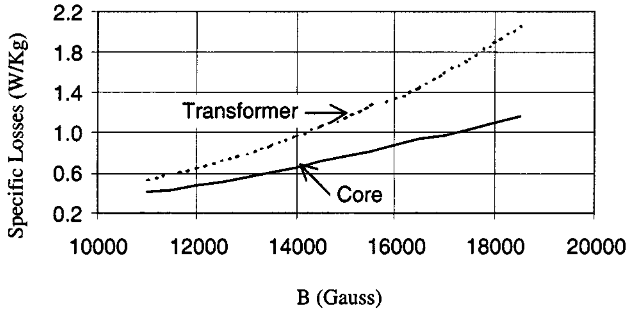


Fig. 2. Typical loss curve, which is used in the examined industrial environment.

A. Core Loss Estimation

Iron losses constitute one of the main parameters for determining the transformer quality. Usually, customers' specifications define an upper limit, say P_0 concerning transformer iron losses. For this reason, the transformer is designed [13] so that its theoretical (design) iron losses $P^d_{\mathbf{t}_i}$ are less or equal to the specified loss limit P_0:

$$P^d_{\mathbf{t}_i} \leq (1-m)P_0 \tag{3}$$

where m corresponds to the safety margin used during the transformer design.

In current practice, the typical loss curve is used to estimate the theoretical iron losses $P^d_{\mathbf{t}_i}$ of the transformer $\mathbf{t}_i$. The loss curve expresses the relationship between specific iron losses $S^d_{\mathbf{t}_i}$, i.e., losses normalized per weight unit (in W/Kg) versus magnetic induction B (in Gauss). A typical loss curve used in the considered industrial environment is depicted in Fig. 2 as the dotted line. The design iron losses $P^d_{\mathbf{t}_i}$ of the transformer $\mathbf{t}_i$ are estimated by multiplying specific iron losses $S^d_{\mathbf{t}_i}$, calculated from Fig. 2 at a given rated magnetic induction, by the theoretical (design) total core weight, $K^d_{\mathbf{t}_i}$, of transformer $\mathbf{t}_i$:

$$P^d_{\mathbf{t}_i} = S^d_{\mathbf{t}_i} * K^d_{\mathbf{t}_i}. \tag{4}$$

The theoretical core weight of transformer $\mathbf{t}_i$, i.e., $K^d_{\mathbf{t}_i}$, is calculated from the theoretical weights of its four individual cores. That is

$$K^d_{\mathbf{t}_i} = 2 * (K^d_s + K^d_l) \tag{5}$$

where K^d_s and K^d_l are the theoretical weights of small and large cores.

The theoretical weights of individual cores depend on their geometrical dimensions (i.e., width and height of core window, thickness and width of core leg), the core space factor and the rated magnetic induction, as described in [14]. The magnetic induction is the same as the one used for the three-phase transformer to estimate the specific losses $S^d_{\mathbf{t}_i}$ based on the curve of Fig. 2.

Based on the above, various transformer parameters, which affect the theoretical transformer weight and its specific iron losses, are examined and the design which satisfies the customers' requirements (3) at a minimum cost, is selected as the most appropriate.

The total theoretical (design) losses of the four individual cores assembled to construct the transformer $\mathbf{t}_i$ are given by

$$F^d_{\mathbf{t}_i} = 2 * (P^d_s + P^d_l) \tag{6}$$

where P^d_s, P^d_l are the theoretical (design) iron losses of small and large individual cores, while $F^d_{\mathbf{t}_i}$ represents the theoretical (design) total iron losses of the four individual cores of $\mathbf{t}_i$. The theoretical (design) iron losses of the four individual cores can be computed based on their loss curve (solid line of Fig. 2) at the rated magnetic induction used for the three-phase transformer. It should be mentioned that the total iron losses $F^d_{\mathbf{t}_i}$ are not equal to the transformer iron losses $P^d_{\mathbf{t}_i}$ since additional losses in general appear during the assembly of the four individual cores, i.e., $F^d_{\mathbf{t}_i} < P^d_{\mathbf{t}_i}$.

B. Core Grouping Process

Although all transformers constructed under the same design should present the same iron losses $P^d_{\mathbf{t}_i}$, their actual losses, say $P^a_{\mathbf{t}_i}$ usually diverge from the designed ones. This is due to the fact that several parameters, involved in the construction process, such as the formation of individual cores, the conditions of transformer production, and the quality of magnetic material, affect the final transformer quality. Thus, it is possible for the actual iron losses of a transformer to exceed the upper loss limit P_0. The same happens with the actual losses of individual cores, which in general differ from the designed ones. In the following, we denote as P^a_l (P^a_s) the actual iron losses of a large (small) individual core from all $2 * N$ available large (small) cores. Therefore, random assembly of two small and two large cores to form a three-phase transformer may result in transformers of significant deviation from their designed quality. In particular, grouping together only cores of low quality constructs transformers of unacceptable quality. For this reason, a grouping process of individual cores is performed by assembling cores of high and low quality together. In this way, cores of low quality are compensated with cores of high quality to reduce the deviation of transformer actual losses from the designed ones. In current practice, the following grouping method is used.

Initially, individual cores (small or large) are classified into "quality classes" according to the deviation of their actual losses from the designed ones. In particular, the quality classes for small/large cores are defined as follows:

$$\begin{aligned} C^s_k = &\{s \in V_s : (1+(2k-1)\delta)P^d_s < P^a_s \\ &\leq (1+(2k+1)\delta)P^d_s\}, \qquad k=-3\cdots 3 \end{aligned} \tag{7a}$$

$$\begin{aligned} C^l_k = &\{l \in V_l : (1+(2k-1)\delta)P^d_l < P^a_l \\ &\leq (1+(2k+1)\delta)P^d_l\}, \qquad k=-3\cdots 3 \end{aligned} \tag{7b}$$

where seven "quality classes" are assumed. The 2δ corresponds to the class width and P^d_s, P^d_l are the theoretical iron losses of a small/large core as it has been defined in the previous subsection. Positive values of index k correspond to cores with actual iron losses greater than the designed ones. On the contrary, negative values indicate actual losses smaller than the designed ones. Consequently, as the index k increases, the core quality decreases and vice versa. Cores belonging to the class of zero index, i.e., C_0^s or C_0^l, present actual iron losses close to the theoretical ones within a deviation of $\pm\delta$.

A grade is assigned to each class indicating its quality, so that all cores of a class are characterized by the same quality grade. Since the class index k is inversely proportional to the core quality, the negative index of the respective class is defined as its grade

$$g(s) = -k, \quad \text{if } P_s^a \in C_k^s \tag{8a}$$

$$g(l) = -k, \quad \text{if } P_l^a \in C_k^l \tag{8b}$$

where we recall that $s \in V_s$, and $l \in V_l$ is a small/large core from all $2 * N$ small and $2 * N$ large available.

Based on the quality grade of each individual core, a grouping process is applied to reduce the deviation of the actual iron losses of the constructed transformers. In particular, cores of high and low quality grades are assembled together to prevent production of transformers with very low or too high quality. This is accomplished by selecting the four individual cores, $s_i^r, s_i^l, l_i^r, l_i^l$, comprising the transformer $\mathbf{t}_i$, so that the sums of the quality grades of the two small and two large cores are close to zero, that is

$$\begin{gathered} g(s_i^r) + g(s_i^l) = 0, \quad g(l_i^r) + g(l_i^l) = 0 \\ \forall\, s_i^r, s_i^l \in V_s \quad \text{and} \quad \forall\, l_i^r, l_i^l \in V_l \end{gathered} \tag{9}$$

or equivalently is held that

$$(1-\delta)F_{\mathbf{t}_i}^d \; < \; F_{\mathbf{t}_i}^a \le (1+\delta)F_{\mathbf{t}_i}^d \tag{10}$$

where $F_{\mathbf{t}_i}^a$ represents the total actual losses of the four individual cores assembled to construct the transformer $\mathbf{t}_i$. Equation (10) indicates that the average actual iron losses of the two small and two large individual cores for all transformers $\mathbf{t}_i$ are close to the theoretical ones with an uncertainty interval of $\pm\delta$, i.e., the class width.

In the above method, the quality of individual cores is used to indicate the quality of three-phase transformers. However, the actual losses of a transformer are not equal to the losses of its individual cores. This is due to the fact that additional parameters appear during the transformer construction, like the exact arrangement of the four individual cores, which are not considered by the above-mentioned technique. For example, reordering the two small or the two large cores of a transformer, results in different actual iron losses though the average losses of the four cores remain the same. Another drawback of the current grouping process is that it does not provide the optimal arrangement of the $2 * N$ small and $2 * N$ large cores so that the iron losses of the N constructed transformers are as minimal as possible.

III. Proposed Method

In this paper, a novel technique is proposed so that the $2 * N$ small and $2 * N$ large cores are appropriately arranged to construct transformers of optimal quality. Fig. 3 presents a block diagram of the proposed scheme. First, the transformer design is accomplished based on customers' specifications and several techno-economical criteria as described in Section II-A. In this phase, several constructional parameters of the transformer are specified, such as the geometric characteristics of individual cores, the thickness, grade and supplier of magnetic material, and the rated magnetic induction. Then, the individual cores are constructed and several measurements are taken for each core to determine the core performance. Next, a combined neural network and genetic algorithm approach is used to estimate the optimal core arrangement which results in three-phase transformers of minimum iron losses. More specifically, the measurements taken from the core construction phase, as well as additional parameters, affecting the transformer quality, are used to predict the actual iron losses of the transformer. The prediction is accomplished through a neural network that relates all the parameters, called attributes, with the actual transformer losses. A new grouping process is then applied to minimize the iron losses of all constructed transformers by the available small and large cores. In general, the number of core combinations is extremely large for a typical number of small/large cores. For that reason a genetic algorithm has been adopted to estimate within a few iterations the optimal arrangement of the four individual cores so that transformers of the best quality are constructed. In particular, at each step, a population of new core arrangements is generated and prediction of the actual iron losses of the respective transformers is accomplished by the neural network model until minimal losses are provided for one specific (optimum) arrangement.

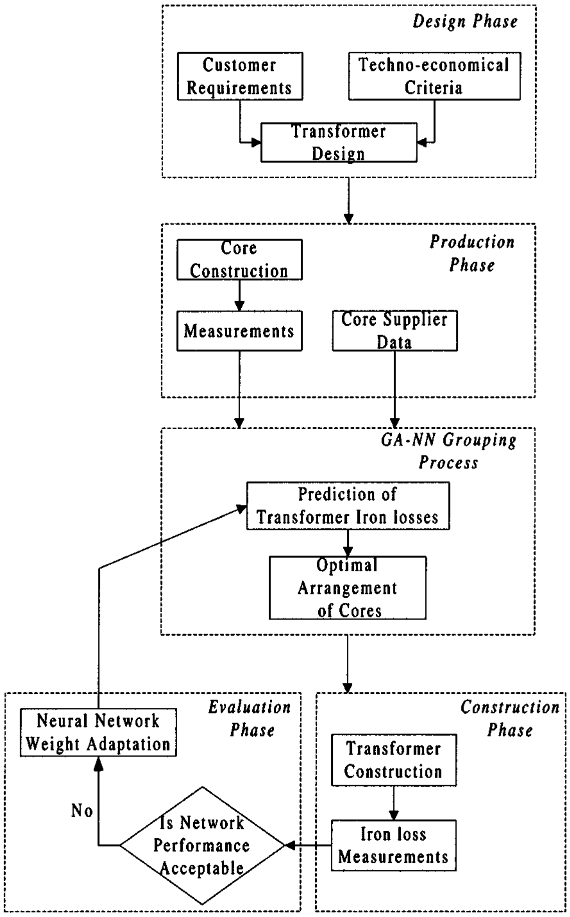


Fig. 3. Proposed combined neural network-genetic algorithm method applied for iron loss reduction.

TABLE I
THREE ENVIRONMENTS CONSIDERED IN THE EXAMINED MANUFACTURING INDUSTRY

	Environment		
Characteristic	#1	#2	#3
Supplier	SUP_A	SUP_B	SUP_A
Steel grade	M3	M4	Hi-B
Thickness (mm)	0.23	0.27	0.23

IV. NEURAL NETWORKS FOR PREDICTING IRON LOSSES

The neural network architecture used for predicting the actual iron losses of a three-phase transformer is analyzed in this section. For each transformer $\mathbf{t}_i$, several attributes are extracted and gathered in a vector, say $\mathbf{a}(\mathbf{t}_i)$. This vector is fed as input to the neural network. However, for different types of magnetic material and supplier, different relations between the extracted attributes and transformer actual losses are expected. This is due to the fact that each supplier follows a specific technology of magnetic material production, while the grade and thickness present their own characteristics. In the following, the term environment is used to indicate a given supplier, thickness and grade of magnetic material. Table I presents the three different environments used in the considered industry.

Let us assume in the following that M environments are available, denoted as Π_i, $i = 1, 2, \cdots, M$. In this case, M nonlinear functions, say $h_c(\cdot)$ with $c \in \{\Pi_1, \cdots, \Pi_M\}$ are defined which relate the attributes $\mathbf{a}(\mathbf{t}_i)$ of $\mathbf{t}_i$ with the respective actual specific iron losses $S^a_{\mathbf{t}_i}$. That is

$$S^a_{\mathbf{t}_i} = h_c(\mathbf{a}(\mathbf{t}_i)). \tag{11}$$

Since functions $h_c(\cdot)$ are actually unknown, feedforward neural networks are used to estimate them. The use of feedforward networks is due to the fact that they can approximate any nonlinear function within any degree of accuracy [15, pp. 208–213, 249]. In our case, M feedforward neural networks are implemented, each of which corresponds to a specific environment. A single neural network can be also applied but using the environment type as additional network input. However, such an approach provides greater generalization error than using M independent networks as is shown in the section of the experimental results.

Let us denote as $\hat{h}_c(\cdot)$ an approximate of function $h_c(\cdot)$ as is provided by the network. Then the estimate of specific iron losses, say $\hat{S}^a_{\mathbf{t}_i}$, of a transformer $\mathbf{t}_i$ with attributes $\mathbf{a}(\mathbf{t}_i)$ is given as

$$\hat{S}^a_{\mathbf{t}_i} = \hat{h}_c(\mathbf{a}(\mathbf{t}_i)). \tag{12}$$

As can be seen, in (11) and (12), the actual specific iron losses (in watts per kilogram) have been used as output of the neural network model, instead of the actual iron losses (in watts). This selection improves the network performance (generalization) since normalization of the network output is performed per weight unit. Furthermore, neural network training is made more efficient by using such a normalization scheme. Then, the actual transformer iron losses are calculated by multiplying $\hat{S}^a_{\mathbf{t}_i}$ by the sum of the actual weights of the four individual cores that assemble the transformer $\mathbf{t}_i$.

Selection of the most appropriate environment is performed during the design phase, where the type of the magnetic material and the respective supplier are determined. Consequently, the environment type is known before the transformer construction.

The neural network structure used to approximate $h_c(\cdot)$ is depicted in Fig. 4. As is observed, the network consists of a hidden layer of n neurons, J input elements and one output neuron. In our case, a linear output unit is used, since the network approximates a continuous valued signal, i.e., the specific iron losses of a transformer. The number of hidden neurons n, as well as the network weights are appropriately estimated based on a constructive training algorithm, which is described in the following subsection. Furthermore, a decision tree (DT) methodology is adopted to select the most appropriate attributes used as inputs to the network among a large number of candidates ones (see Section IV-B.) Finally, Section IV-C presents a weight adaptation algorithm used to adapt the network weights in case that a slight modification of the environment conditions is encountered.

A. Network Training and Generalization Issues

The neural network size affects the prediction accuracy. Particularly, a small network is not able to approximate complicated nonlinear functions, since few neurons are not sufficient to implement all possible input–output (I/O) relations. On the other hand, recent studies on network learning versus generalization, such as the VC dimension [16], [17] indicate that an unnecessarily large network heavily deteriorates network performance. In this paper, the constructive algorithm, presented in [18], has been adopted to simultaneously estimate the network size and the respective network weights. Usually, constructive approaches present a number of advantages over other methods used for network size selection. More specifically, in a constructive scheme, it is straightforward to estimate an initial size for the network. Furthermore, in case that many networks of different sizes provide acceptable solutions, the constructive approach yields the smallest possible size [18].

Let us denote as $\hat{h}_{c,n}(\cdot)$ the function, which implements the neural network of Fig. 4, in case that n hidden neurons are used. The subscript c is omitted in the following analysis since we refer to a specific environment. If we denote as $r_j(\cdot)$, $j = 1, 2, \cdots, n$ the function that the jth hidden neuron implements, then the network output is given as

$$\hat{S}^a_{\mathbf{t}_i,n} \equiv \hat{h}_n(\mathbf{a}(\mathbf{t}_i)) = \sum_{j=1}^{n} u_j r_j(\mathbf{a}(\mathbf{t}_i)) \tag{13}$$

where u_j is the weight, which connects the jth hidden neuron to the output neuron (see Fig. 4) and $\hat{S}^a_{\mathbf{t}_i,n}$ the estimate of the actual specific iron losses provided by a network of n hidden neurons.

Based on the neural network structure of Fig. 4, function $r_j(\cdot)$ is written as

$$r_j(\mathbf{a}(\mathbf{t}_i)) = \sum_{k=1}^{J} f(\mathbf{w}_j^T \cdot \mathbf{a}(\mathbf{t}_i) + \vartheta_j) \tag{14}$$

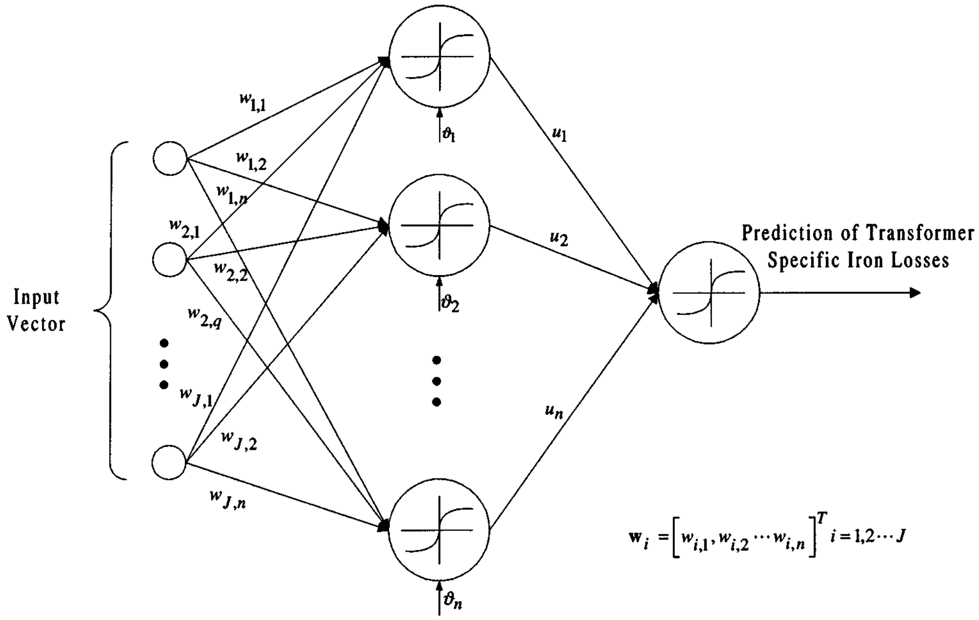


Fig. 4. Proposed feedforward neural network architecture used for iron loss prediction.

where

$f(\cdot)$ activation functions of hidden neurons (the sigmoid in our case);

$\mathbf{w}_j$ weight vector, which connects the jth hidden neuron with the input layer;

ϑ_j bias of the jth hidden neuron.

Let us now assume that a new unit (neuron) is added to the hidden layer of the network. Let us also denote as $\hat{S}^a_{\mathbf{t}_i,n+1}$ the estimate of specific actual losses provided by a network of $n+1$ hidden units. Then, based on (13), the following relationship is satisfied:

$$\begin{aligned}\hat{S}^a_{\mathbf{t}_i,n+1} \equiv & \hat{h}_{n+1}(\mathbf{a}(\mathbf{t}_i)) = \hat{S}^a_{\mathbf{t}_i,n+1} + u_{n+1}r_{n+1}(\mathbf{a}(\mathbf{t}_i)) \\ = & \hat{h}_n(\mathbf{a}(\mathbf{t}_i)) + u_{n+1}r_{n+1}(\mathbf{a}(\mathbf{t}_i)). \qquad (15)\end{aligned}$$

In the previous equation, $r_{n+1}(\cdot)$ refers to the function that implements the new added hidden neuron. As results from equation (14), function $r_{n+1}(\cdot)$ is defined by the weight vector $\mathbf{w}_{n+1}$ and the respective bias ϑ_{n+1}. In the adopted constructive method, only the parameters associated to the new hidden unit are permitted to change, i.e., the weights $\mathbf{w}_{n+1}$, the bias ϑ_{n+1} and the weight output u_{n+1}. All the other network weights are considered fixed.

In particular, the new network weights are estimated so that the error between the actual specific iron losses and the ones estimated by the network decreases as a new hidden neuron is added. To estimate the new network weights, we initially define the following quantity:

$$\Gamma = \frac{\langle e_n, r_{n+1}\rangle^2}{\|r_{n+1}\|^2} \qquad (16)$$

where

$$e_n = \|S^a_{\mathbf{t}_i} - \hat{S}^a_{\mathbf{t}_i,n}\| = \|h - \hat{h}_n\| \qquad (17)$$

represents the residual error of the target nonlinear function (actual specific iron losses) and the one implemented by a neural network of *n* hidden neurons. In (16), the $\langle\cdot\rangle$ corresponds to the inner product, while $\|\cdot\|$ to the norm.

Based on functional analysis, it has been proven in [18] that the error e_n tends to zero as the number of n increases, i.e., $\lim_{n\to\infty} e_n = 0$, if the weights associated to the new hidden added neuron are estimated by

$$\{\mathbf{w}_{n+1}, \vartheta_{n+1}\} = \arg\max \Gamma \qquad (18a)$$

$$\text{and} \qquad u_{n+1} = \frac{\langle e_n, r_{n+1}\rangle}{\|r_{n+1}\|^2.} \qquad (18b)$$

Consequently, if a neural network is constructed incrementally, with weights that satisfy (18) then strongly convergence to the target function is accomplished. Maximization of (18) is performed using the algorithm of [19].

However, in practice, the exact form of target function *h* actually is unknown, and thus the error e_n cannot be directly calculated. For this reason, a training set is used, consisting of *L* transformers, all belonging to the same environment, to provide a consistent estimate of e_n.

In particular, let us denote as S_{tr} this training set. Then, an estimate of quantity Γ is given by

$$\hat{\Gamma} = \frac{\sum\limits_{\mathbf{t}_i\in S_{\text{tr}}} (E_n(\mathbf{t}_i) - \overline{E}_n) - (r_{n+1}(\mathbf{a}(\mathbf{t}_i)) - \overline{r}_{n+1})^2}{\sum\limits_{\mathbf{t}_i\in S_{\text{tr}}} (r_{n+1}(\mathbf{a}(\mathbf{t}_i)) - \overline{r}_{n+1})^2} \qquad (19)$$

where

$$E_n(\mathbf{t}_i) = |S^a_{\mathbf{t}_i} - \hat{S}^a_{\mathbf{t}_i,n}| = |S^a_{\mathbf{t}_i} - \hat{h}_n(\mathbf{a}(\mathbf{t}_i))| \qquad (20)$$

is the absolute difference between the actual specific iron losses and the predicted ones for a network of *n* hidden neurons in case of a transformer $\mathbf{t}_i \in S_{\text{tr}}$. In (19), $\overline{E}_n$ and $\overline{r}_{n+1}$ are the mean values of functions $E_n(\mathbf{t}_i)$ and $r_n(\cdot)$ over all samples of set S_{tr}.

Equation (19) expresses the correlation between the function implemented by the new added hidden neuron and the previous residual error (before the new neuron is added) over all samples (transformers) of training set $S_{\rm tr}$. This means that the new neuron compensates the residual error as much as possible and therefore the error over data of the training set decreases as the number of hidden neurons increases.

The generalization performance, however, of the neural network, i.e., the error over data outside the training set, does not keep on improving as more hidden units are added. This is due to the fact that a large number of hidden units makes the network sensitive to the data of $S_{\rm tr}$. Particularly, what a network is learning beyond a number of hidden neurons is actually noise of data of the training set. As a result, the generalization performance starts to decrease and the incremental construction of the network is terminated. In our case, this is accomplished by applying the cross validation method. According to this method, the available data are divided into two subsets; the first subset (training set) is responsible for estimating the network parameters, while the second subset (validation set) evaluates the network performance. The error on the validation set will normally decrease during the initial phase of training, as does the error on the training set. However, when the network begins to overfit the data, the error on the validation set will typically begin to rise and the constructive training algorithm is terminated (early stopping).

B. Attribute Selection

Another factor, which affects the network performance, is the type of attributes used as network input. For attribute selection, initially, a large set of candidates is formed based on extensive research and transformer designers' experience. Particularly, in our case, 19 candidate attributes are examined, which are denoted as $I_i, i = 1, 2, \cdots 19$ and presented in Table II.

TABLE II
LIST OF THE CANDIDATE ATTRIBUTES CONSIDERED AS POSSIBLE INPUTS OF THE NETWORK

Symbol	Expression
I_1	B_0
I_2	$\left(U^x_{s^l_i} + U^x_{l^l_i} + U^x_{l^r_i} + U^x_{s^r_i}\right)/4$
I_3	$\left(U^y_{s^l_i} + U^y_{l^l_i} + U^y_{l^r_i} + U^y_{s^r_i}\right)/4$
I_4	$K^a_{t_i} / K^d_{t_i}$
I_5	$F^a_{t_i} / F^d_{t_i}$
I_6	$S^a_{s^l_i} / S^d_{s^l_i}$
I_7	$S^a_{l^l_i} / S^d_{l^l_i}$
I_8	$S^a_{l^r_i} / S^d_{l^r_i}$
I_9	$S^a_{s^r_i} / S^d_{s^r_i}$
I_{10}	$K^a_{s^l_i} / K^d_{s^l_i}$
I_{11}	$K^a_{l^l_i} / K^d_{l^l_i}$
I_{12}	$K^a_{l^r_i} / K^d_{l^r_i}$
I_{13}	$K^a_{s^r_i} / K^d_{s^r_i}$
I_{14}	$\left(S^a_{s^l_i} + S^a_{l^l_i}\right) / \left(S^d_{s^l_i} + S^d_{l^l_i}\right)$
I_{15}	$\left(S^a_{l^l_i} + S^a_{l^r_i}\right) / \left(S^d_{l^l_i} + S^d_{l^r_i}\right)$
I_{16}	$\left(S^a_{l^r_i} + S^a_{s^r_i}\right) / \left(S^d_{l^r_i} + S^d_{s^r_i}\right)$
I_{17}	$\left(S^a_{s^l_i} + S^a_{l^l_i}\right) / \left(S^a_{l^l_i} + S^a_{l^r_i}\right)$
I_{18}	$\left(S^a_{l^l_i} + S^a_{l^r_i}\right) / \left(S^a_{l^r_i} + S^a_{s^r_i}\right)$
I_{19}	$\left(S^a_{l^l_i} + S^a_{l^r_i}\right) / \left(S^a_{s^l_i} + S^a_{s^r_i}\right)$

In this table, $U^x_{s^l_i}$ ($U^y_{s^l_i}$) denotes the specific iron losses of magnetic material at 15 000 Gauss (17 000 Gauss) of the left small core s^l_i. The specific iron losses for the other three cores are denoted accordingly. $F^a_{\mathbf{t}_i}$ denotes the sum of the actual iron losses of the four individual cores that assemble the transformer $\mathbf{t}_i$ and is defined similarly to (6) as

$$F^a_{\mathbf{t}_i} = F^a_{s^l_i} + F^a_{l^l_i} + F^a_{l^r_i} + F^a_{s^r_i} \tag{21}$$

where $F^a_{s^l_i}$ and $F^a_{s^r_i}$ are the actual (measured) iron losses of the left and right small individual core of $\mathbf{t}_i$. Similarly, $F^a_{l^l_i}$ and $F^a_{l^r_i}$ correspond to the actual iron losses of the left and right large individual core. The physical meaning of the other variables of Table II are explained in Section II.

1) Decision Tree (DT) Methodology: A decision tree (DT) methodology [20], [21] has been adopted in this paper for attribute selection. Initially, an acceptability criterion is defined. Let us denote as C_a the class, which contains all acceptable transformers and as C_u the class, which contains all unacceptable transformers. In our case, classes C_a and C_u are defined as follows:

$$C_a = \{\mathbf{t}_i : P^a_{\mathbf{t}_i} < (1+\xi)P_0) \tag{22a}$$

$$C_u = \{\mathbf{t}_i : P^a_{\mathbf{t}_i} \geq (1+\xi)P_0) \tag{22b}$$

where ξ is a constant indicating the unacceptability threshold.

In order to describe the structure of a DT, we initially present an example in Fig. 5 created from a set of 1680 transformers of the first environment. As observed, the tree consists of two different types of nodes; the terminal nodes and the nonterminal nodes. A node is said to be terminal if it has no children. On the contrary, each nonterminal node has two children and is characterized by an appropriate test (condition) of the following form:

$$T : I_i \leq \gamma \tag{23}$$

where γ is a threshold value of attribute I_i, optimally estimated during the DT construction. This test dichotomizes the nonterminal node in the sense that the left child contains all transformer (samples), which satisfy the test of parent node, while the right child contains the remaining transformers. For each node, the number of transformers (samples) that this node contains and the respective acceptability ratio is also presented.

Based on the acceptability ratio, a terminal node is classified to one of the two available classes. In particular, in case that the acceptability ratio is greater than 50%, the terminal node is assigned to class C_a. Otherwise, it is assigned to class C_u. The exact notation used for each DT node of Fig. 5 is explained in Fig. 6.

A DT is created by applying two main operators; the splitting operator and the stopping operator. The first estimates the most appropriate test that should be applied to a nonterminal node, while the second determines whether a node is terminal or not.

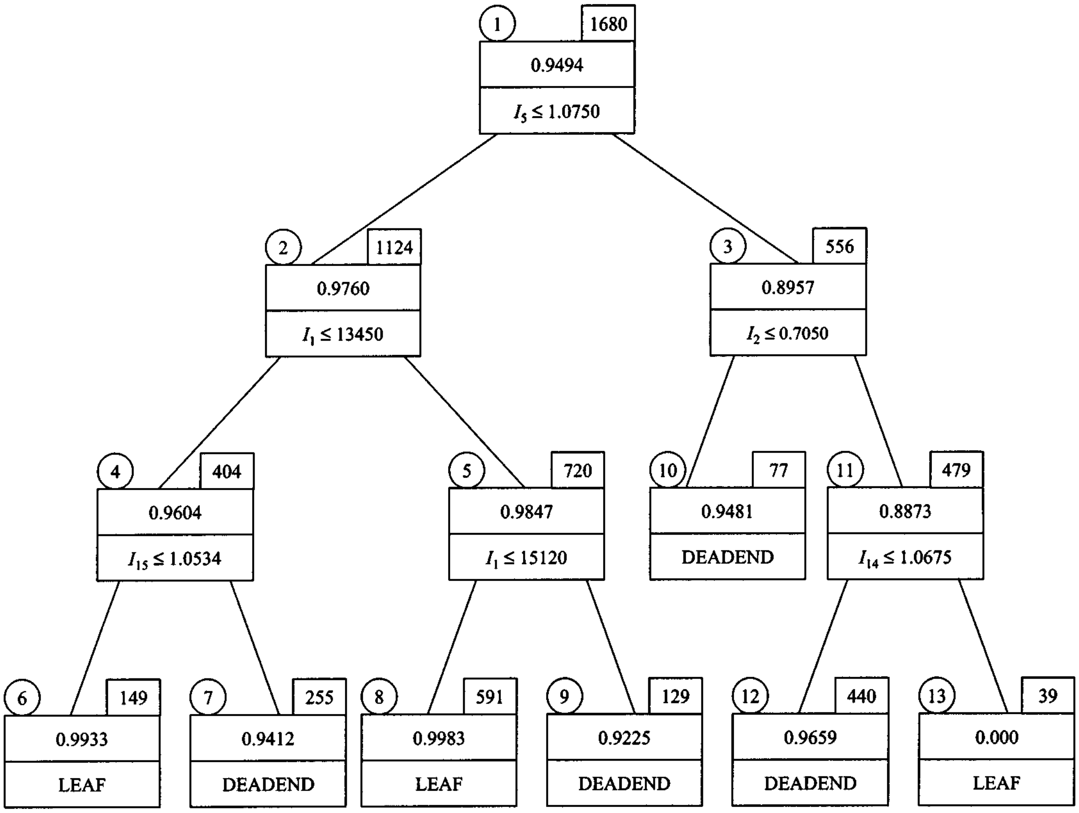


Fig. 5. Decision tree created from a set of 1680 transformers of the first environment.

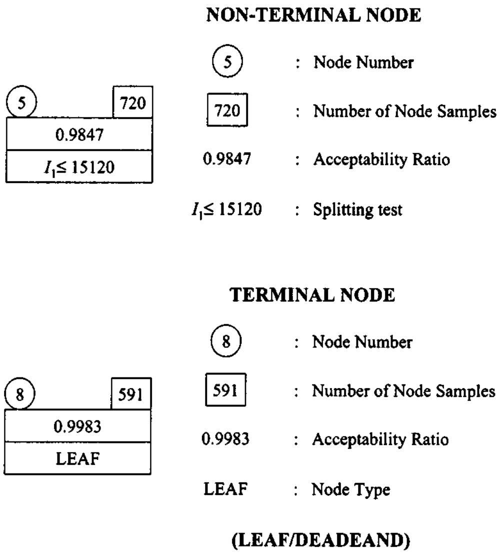

NON-TERMINAL NODE
5
720
0.9847
$I_1 \leq 15120$
5 : Node Number
720 : Number of Node Samples
0.9847 : Acceptability Ratio
$I_1 \leq 15120$: Splitting test
TERMINAL NODE
8
591
0.9983
LEAF
8 : Node Number
591 : Number of Node Samples
0.9983 : Acceptability Ratio
LEAF : Node Type
(LEAF/DEADEAND)

Fig. 6. Explanation of the notation of the decision tree, which is used in Fig. 5.

For the spitting operator, the optimal splitting rule described in [20] is used in our case. More specifically, the algorithm estimates the test that provides the best separation of all transformers of the examined node into acceptable and unacceptable samples. The optimal slitting rule is repeated for each node of the tree, until a node is labeled as terminal according to the stopping criterion. Two different types of terminal nodes are distinguished; the "LEAF" and "DEADEND" nodes. A node is said to be "LEAF" if it contains transformers, which completely belong (or in practice almost completely) in one of the two classes. On the other hand, a node is denoted as "DEADEND" if the gain by splitting this node provides no significant statistical information. This gain is determined by the risk level a of the DT [20]–[22].

2) Implementation Issues: The risk level a affects the structure of a DT. In particular, in case a small value of risk level is used, the tree is grown with a small number of nodes and vice versa. However, the classification performance of a DT does not keep on improving as its size increases. For this reason, the optimal value of risk level a is the one that provides the maximum classification accuracy with the minimum possible DT complexity (minimum number of tree nodes). In order to estimate the classification performance of a DT, we use a different *evaluation set*. For each sample (transformer) of this set, the *tests* (conditions) of the nonterminal nodes are evaluated until a terminal node is reached. Then, the classification accuracy is computed by comparing the actual class that this sample belongs to, with the class of terminal node, which this sample is assigned to.

Fig. 7 illustrates the classification accuracy of the DT of Fig. 5 using a set of 560 transformers (samples) of the first environment for risk levels in the range of 0.001% to 10%. As it can be seen, the classification accuracy increases until a risk level smaller than 0.75%. Then, it starts to decrease. Furthermore, the maximum accuracy (i.e., 95.5%) is reached for risk levels in the interval 0.20%–0.75%. Fig. 8 illustrates the DT complexity (number of nodes) versus the risk level. As observed, the DT complexity increases with respect to the risk level. By com-

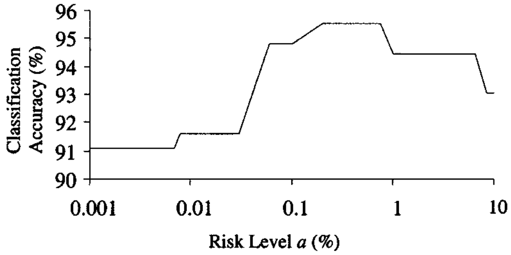


Fig. 7. Effect of risk level on the classification accuracy.

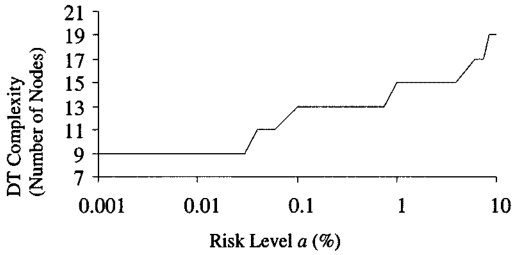


Fig. 8. Effect of risk level on the decision tree complexity (number of tree nodes).

bining Figs. 7 and 8, we can estimate the risk level value that provides the maximum accuracy at the minimum possible DT complexity. This is achieved using 13 DT nodes as illustrated in Fig. 5.

The DT of Fig. 5 has been created by applying the aforementioned splitting and stopping operators with a risk level equal to 0.25%. As observed, only five attributes among the 19 candidate ones are extracted in this case as the most appropriate, the I_1, I_2, I_5, I_{14}, and I_{15}.

It has been observed that the classification accuracy of the DT deteriorates in case it is constructed by transformers belonging to all environments [12]. For this reason, three different measurement sets, each of them corresponding to a specific environment, are used to construct the DT (2240, 2350, and 1980 samples respectively). In order to extract the most significant attributes, which are used as inputs to the neural network, we built several DTs by 1) randomly selecting different transformer samples of each measurement set to build the tree and 2) by using different values of constant ξ. In our case, 30 randomly selected sets have created for each measurement set $(3 \times 30 = 90$ sets for all environments), and five different values of ξ uniformly distributed in the interval 7%–15%. Then, for each case, the optimal risk level is estimated. This is performed by examining 20 different risk levels in the interval 0.001%–10% and the one which maximizes the DT classification accuracy at the minimum DT complexity is selected as the optimal one, as described above. Consequently 9000 DTs are examined $(90 \times 20 \times 5 = 9000)$, 450 of which correspond to the optimal risk level value. The latter (i.e., the 450) are used for the attribute selection. As we have observed, in most cases, the same attributes are extracted, whereas some of them are not selected at all. Furthermore, the same attributes are extracted, even for transformers belonging to different environments. This is due to fact that the environment type determines the influence of an attribute value on transformer iron losses but not the type of attributes.

TABLE III
SELECTED ATTRIBUTES BY THE DECISION TREE METHODOLOGY

Symbol	Expression
I_1	B_0
I_2	$\left(U^x_{s_i^l} + U^x_{l_i^l} + U^x_{l_i^r} + U^x_{s_i^r}\right)/4$
I_3	$\left(U^y_{s_i^l} + U^y_{l_i^l} + U^y_{l_i^r} + U^y_{s_i^r}\right)/4$
I_4	$K^a_{t_i} / K^d_{t_i}$
I_5	$F^a_{t_i} / F^d_{t_i}$
I_6	$\left(S^a_{s_i^l} + S^a_{l_i^l}\right)/\left(S^d_{s_i^l} + S^d_{l_i^l}\right)$
I_7	$\left(S^a_{l_i^l} + S^a_{l_i^r}\right)/\left(S^d_{l_i^l} + S^d_{l_i^r}\right)$
I_8	$\left(S^a_{l_i^r} + S^a_{s_i^r}\right)/\left(S^d_{l_i^r} + S^d_{s_i^r}\right)$

Taking into account all DTs, the attributes with a probability of appearance greater than 3% are selected as network inputs. These attributes are presented in Table III. It should be mentioned that in this case we renumbered the selected attribute indices of Table II as they are presented in consecutive order in Table III. A small value of probability has been chosen since it is more preferable to use more attributes as inputs to the network architecture than discard some (maybe significant for some situations) of them.

The selection of these attributes is reasonable and expected. More specifically, attribute I_1 is the rated magnetic induction, which is also used in order to calculate iron losses at the design phase by using the loss curve. Attributes I_2 and I_3 express the average specific losses (W/Kg at 15 000 Gauss and 17 000 Gauss, respectively) of magnetic material of the four individual cores used for transformer construction. Attribute I_4 is the ratio of actual over theoretical weight of the four individual cores. Attribute I_5 is equal to the ratio of actual over theoretical iron losses of the four individual cores. The significance of the attribute I_5 is that the iron losses of the three-phase transformer depend on the iron losses of its individual cores. In the industrial environment considered, it is observed that the arrangement of cores influences the assembled transformer core losses. This is reflected through the selection of attributes I_6, I_7, and I_8 by the DT methodology (see Table III).

C. Weight Adaptation

In some cases, the conditions under which the respective neural network has been trained may slightly change over time. For example, it is possible that different batches of magnetic material, belonging to the same environment, present small variations in their technical characteristics. In such cases, the network performance is improved by introducing a weight adaptation mechanism, which slightly modifies the network weights to the new conditions.

The weight adaptation mechanism is activated when the network performance deteriorates. This is accomplished during the *evaluation phase* (see Section III and Fig. 3), in which the predicted iron losses are compared with the actual ones. In case that the average prediction error is greater than a pre-determined threshold, the weight adaptation mechanism is activated and new network weights are estimated. The threshold considered is slightly greater than the average error over all data of a test set, which expresses the generalization performance of the network.

The adaptive training algorithm modifies the weights so that the network appropriately responds to new data, and also provides a minimal degradation of the old information [23], [24]. Training the network, without using the old information, but only the new data, would result in a catastrophic forgetting of the previous knowledge [25]. In our case, the algorithm proposed in [24] has been adopted to perform the weight adaptation.

V. Genetic Algorithms for Reducing Iron Losses

In this section, we describe the algorithm used for optimal arrangement of the individual cores so that the iron losses of all constructed transformers are as minimal as possible. In particular, in the following subsection the problem formulation is presented while Section V-B describes the genetic algorithm, which is applied for the optimization. Finally, Section V-C discusses issues related to the genetic algorithm convergence.

A. Optimization of Core Grouping Process

Let us denote as $\mathbf{c}$ a vector containing one possible combination of the N three-phase transformers $\mathbf{t}_i$, $i = 1, \cdots, N$, that can be constructed by the $2 * N$ available small/large cores

$$\mathbf{c} = [\mathbf{t}_1^T \quad \mathbf{t}_2^T \quad \cdots \quad \mathbf{t}_N^T]^T \tag{24}$$

where T indicates the transpose of a vector.

Vector $\mathbf{c}$ is of $4N \times 1$ dimensions since each transformer $\mathbf{t}_i$ is represented by a 4 × 1 vector as (1) indicates. A specific arrangement (combination) of all small and large cores, for constructing the N three-phase transformers, corresponds to a given value of vector $\mathbf{c}$. Therefore, any reordering of the elements of vector $\mathbf{c}$ results in different arrangement of individual cores, i.e., different three-phase transformers. Fig. 9 presents an example of vector $\mathbf{c}$ in case that six small and six large cores are available. In particular, the serial numbers from 1 to 6 correspond to small cores, while the numbers from 7 to 12 to large cores. A randomly selected arrangement of these cores is also presented in Fig. 9 for constructing three different transformers. For example, the first transformer consists of the small cores with labels 5 and 1 and of the large cores with labels 10 and 12. This is represented by the vector [5 10 12 1]^T in accordance with (1). Then, vector $\mathbf{c}$ is constructed by concatenating the vectors of the three transformers. The core arrangement for the other two transformers is generated accordingly and depicted in Fig. 9.

It is clear that the estimation of N transformers with optimal quality (minimum iron losses) is equivalent to the estimation of vector $\mathbf{c}$, which minimizes the following:

$$\mathbf{c}_{opt} = \arg\,\min D = \arg\,\min_{\mathbf{c}} \left\{ \sum_{i=1}^{N} P_{\mathbf{t}_i}^{a} \right\} \tag{25}$$

where $\mathbf{c}_{opt}$ is a vector which contains the optimal arrangement of all available small/large cores so that the actual losses over all N transformers are minimized.

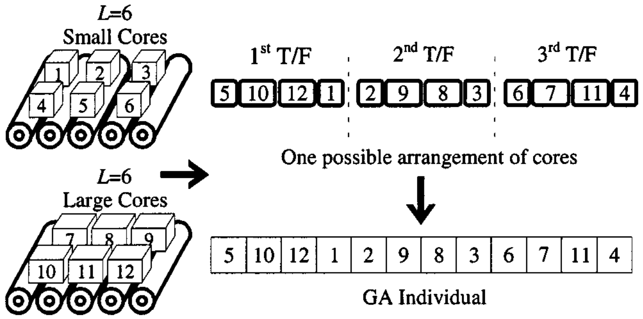


Fig. 9. Example of the adopted encoding scheme in case of six large and six small cores.

The transformer actual losses involved in (25) are estimated by the neural network architecture as has been described in the previous section. However, although the previous equation provides transformers of optimal quality, there is no guarantee that all the generated transformers belong to the acceptable class [see (22a)]. For this reason, a very large value is assigned to a transformer whose predicted iron losses satisfy (22b) (or are slightly smaller to compensate prediction errors). Thus, any unacceptable core arrangement is rejected.

As observed from (25), estimation of the optimal core arrangement results in a global combinatorial optimization problem. Consequently, the previously used feedforward neural network cannot be directly applied for minimizing (25). This is due to the fact that a feedforward neural network is usually suitable for function approximation or classification but not for function minimization. However, other neural networks models, like Hopfield networks or Boltzmann machines, trained based on the simulated annealing algorithm, can be used to find the optimal value $\mathbf{c}_{opt}$ [26, pp. 125–131, 279–301].

GAs can be also applied [27], [28]. The main advantage of these GA schemes is that they simultaneously proceed multiple stochastic solution trajectories and thus allow various interactions among different solutions toward one or more search spaces [29, p. 102], [27]. On the contrary, the neural network approaches normally follow one trajectory (deterministic or stochastic) which is repeated many times until a satisfactory solution is reached. Furthermore, neural networks can be applied more appropriately for functions whose the variables are in product form, which is not held in our case. In the following, a genetic algorithm is proposed to perform the aforementioned minimization. Table IV summarizes the main steps of the proposed method for reducing the transformer iron losses.

B. Genetic Approach

In the genetic approach, possible solutions of the optimization problem are represented by chromosomes whose "genetic material" corresponds to a specific arrangement of individual cores. This means that vector $\mathbf{c}$ of (24) is represented by a chromosome, while the serial numbers of individual cores are considered as the genetic material of the chromosome. An integer number scheme is adopted for encoding the chromosome elements (genes) as is illustrated in Fig. 9.

TABLE IV
SUMMARY OF THE MAIN STEPS OF THE PROPOSED COMBINED NEURAL NETWORK–GENETIC ALGORITHM SCHEME FOR IRON LOSS REDUCTION

Step 1: Based on customer's requirements and several techno-economical criteria, design the transformers of a specific production batch. From the transformer design, the environment type (i.e., supplier, thickness and grade of magnetic material) is defined.

Step 2: Based on the transformer design, construct the individual small and large cores and measure all necessary parameters (i.e., the actual core losses and weight) so that the eight attributes of Table 3 for a specific core arrangement can be calculated.

Step 3: Use the genetic algorithm to minimize the total specific iron losses for all transformers of the production batch [equation (25)]. At each GA cycle, the neural network architecture is used for estimating transformer losses. As input to the neural network, the attributes presented in Table III are used.

Step 4: Assemble the transformers using the results of the combined neural network-genetic algorithm scheme.

Step 5: Measure the actual iron losses for all constructed transformers of the production batch. Then, compare them with the predicted ones, which are provided by the neural network structure.

Step 6: In case of large deviation, adapt the network weights using the algorithm presented in subsection IV-C. Then, store the new estimated weights in the network database to be used for the following production batches. Otherwise, retain the same network weights.

Initially, M different chromosomes, say $\mathbf{c}_1, \cdots, \mathbf{c}_M$ are created to form a population. In our case, M possible solutions of the grouping method used in the current practice are selected for the initial population. This is performed so that the genetic material of the initial chromosomes is of somehow good quality and thus fast convergence of the GA is achieved. The performance of each chromosome, representing a particular core arrangement, is evaluated by the sum of the predicted actual iron losses of all transformers corresponding to this chromosome. The neural network model is used as iron loss predictor. For each chromosome, a *fitness function* is used to map its performance to a fitness value, following a *rank-based normalization scheme*. In particular, all chromosomes $\mathbf{c}_i$ $i = 1, 2, \cdots, M$ are ranked in ascending order according to their performance, i.e., the sum of the predicted transformer losses. Let $rank(\mathbf{c}_i) \in \{1, \cdots, M\}$ be the rank of chromosome $\mathbf{c}_i$ (*rank* = 1 corresponds to the best chromosome and *rank* = M to the worst). Defining an arbitrary fitness value F_0 for the best chromosome, the fitness $F(\mathbf{c}_i)$ of the ith chromosome is given by the linear function

$$F(\mathbf{c}_i) = F_0 - [rank(\mathbf{c}_i) - 1]\mu, \quad i = 1, \cdots, N \tag{26}$$

where μ is a decrement rate and is computed in such a way that the fitness function $F(c_i)$ takes always positive values, that is $\mu < F_0/(M-1)$. The major advantage of the rank-based normalization is that it prevents the generation of *super chromosomes*, avoiding premature convergence to local minima, since fitness values are uniformly distributed [27], [30].

The parent selection mechanism then begins by selecting appropriate chromosomes (parents) from the current population. The *roulette wheel* [28] is used as the parent selection procedure. This is accomplished by assigning to each chromosome a selection probability equal to the ratio of the fitness value of the respective chromosome over the sum of fitness values of all chromosomes, i.e.,

$$p_p(\mathbf{c}_i) = F(\mathbf{c}_i) \bigg/ \sum_{i=1}^{M} F(\mathbf{c}_i) \tag{27}$$

where $p_p(\mathbf{c}_i)$ is the probability of the chromosome $\mathbf{c}_i$ to be selected as parent. Equation (27) means that chromosomes of high quality present higher chance of survival in the next generation. Using this scheme, *M* chromosomes are selected as candidate parents for generating the next population. Obviously, some chromosomes would be selected more than once which is in accordance with the Schema Theorem [28]; the best chromosomes get more copies, the "average" stay even, while the worst die off. Consequently, each chromosome has a growth rate proportional to its fitness value.

In the following step of the algorithm, couples of chromosomes (two parents) are randomly selected from the set of candidate ones, obtained from the parent selection mechanism. Then, their genetic material is mated to generate new chromosomes (offspring). The number of couples selected depends on a crossover rate. A crossover mechanism is also used to define how the genes should be exchanged to produce the next generation. Several crossover mechanisms have been reported in the literature. In our approach, a modification of the *uniform crossover operator* [27], [28] has been adopted. As is explained in the following section, this modification does not

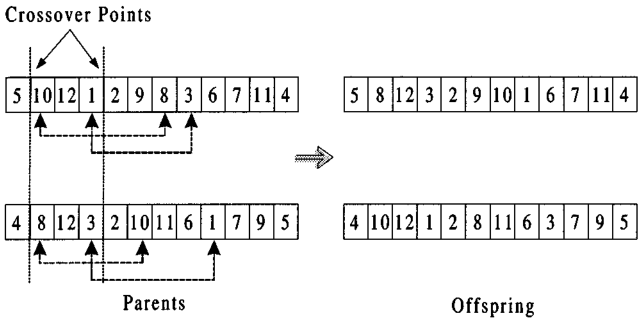


Fig. 10. Example of the proposed modification of the crossover operator.

spoil the GA convergence. In this case, each parent gene, i.e., an individual core, is considered as a potential crossover point. In particular, a gene is exchanged (undergone crossover), if a random variable, uniformly distributed in the interval [0 1], is smaller than a predetermined threshold. Otherwise, the gene remains unchanged. It is possible however for an individual core to appear more than once in the genetic material of the generated chromosome. This means that one individual core is placed to more than one transformer or to more than one position of the same transformer, which corresponds to an unacceptable core arrangement in (2). For this reason, the following modification of the uniform crossover operator is adopted. After the exchange of one gene between the two parents, it is highly possible that the gene appears twice in the chromosome. In this case the gene coinciding with the new gene is replaced with the gene before the exchange. Fig. 10 illustrates an example of the proposed crossover mechanism in case that six small and six large cores are assembled to generate three transformers. In this example, the two parents exchange their genes only between the crossover points 2, 3, and 4 for simplicity. As observed, the genes $\{10, 12, 1\}$ of the first parent are exchanged with the genes $\{8, 12, 3\}$ of the second parent. By applying this exchange of genes, in the first chromosome the genes 8 and 3 appear twice, while genes 10 and 1 disappear. An equivalent problem occurs in the second chromosome. For this reason, in the first chromosome the genes $\{10, 1\}$ are one-by-one exchanged with genes $\{8, 3\}$ as Fig. 10 depicts. The same happens for the second chromosome.

The next step is to apply *mutation* to the newly created population, introducing random gene variations that are useful for restoring lost genetic material, or for producing new material that corresponds to new search areas [28]. *Uniform mutation* is the most common mutation operators and is selected for our optimization problem. In particular, for each gene a uniform number is generated into the interval [0 1] and if this number is smaller than the mutation rate the respective gene is swapped for other randomly selected gene of the same category, i.e., small or large core. Otherwise, the gene remains unchanged. In our experiment, the mutation rate is selected to be 5%. Swapping genes of the same category is necessary for creating valid core arrangement.

At each iteration, a new population is created by inserting the new chromosomes, generated by the crossover mechanism, and deleting their respective parents, so that each population always consists of M chromosomes. Several GA cycles including fitness evaluation, parent selection, crossover, and mutation are repeated, until the population converges to an optimal solution. The GA terminates when the best chromosome fitness remains constant for a large number of generations, indicating that further optimization is unlikely.

C. GA Convergence

The aforementioned modifications of the crossover and mutation operators do not effect the convergence property of the GA. To show this, an analysis is presented in the following, by modeling the GA as a Markov chain. In particular, each state of the Markov state corresponds to a possible solution of the GA, i.e., a specific vector c. Let us denote as D a set, which contains all possible Markov states. Then, for two arbitrary states, say, $i, j \in D$, we denote as p_{ij} the transition probability from state i to state j. Gathering transition probabilities for all states in D, the transition matrix of the chain is formed as $\mathbf{P} = (p_{ij})$. Since in the GA, transition from one state to another is obtained by applying the crossover and mutation operator, matrix $\mathbf{P}$ can be decomposed as follows [31]:

$$\mathbf{P} = \mathbf{C} \cdot \mathbf{M} \tag{28}$$

where matrix $\mathbf{C}$ indicates the effect of crossover operator and matrix $\mathbf{M}$ the effect of mutation operator.

Let us denote as c_{ij} the elements of matrix $\mathbf{C} = (c_{ij})$. The c_{ij} express the transition probability from state $i \in D$ to the state $j \in D$, if only the effect of the crossover operator is taken into consideration. Since the crossover operator probabilistically maps any state of D to any other state of D, matrix $\mathbf{C}$ is a stochastic matrix. More specifically, a matrix is said to be stochastic if its elements c_{ij} satisfy the following property:

$$\sum_j c_{ij} = 1 \Rightarrow \text{Matrix } \mathbf{C} = (c_{ij}) \text{ is stochastic.} \tag{29}$$

The previous equation means that from a valid solution (i.e., a state of D), the crossover operator produces another valid solution (i.e., another state of D). This is exactly happened with the proposed modification of the crossover operator, since only valid solutions are permitted.

On the other hand, matrix $\mathbf{M}$ is positive. This is due to the fact that the mutation operator is applied independently to each gene of a chromosome. Furthermore, each gene can potentially undergo mutation. Consequently, the elements m_{ij} of matrix $\mathbf{M}$, which express the transition probabilities from state $i \in D$ to state $j \in D$ taking into account only the effect of the mutation operator, are strictly positive:

$$m_{ij} > 0 \Rightarrow \text{Matrix } \mathbf{M} = (m_{ij}) \text{ is positive.} \tag{30}$$

It has been proven in [31] that if matrix $\mathbf{C}$ is stochastic and matrix $\mathbf{M}$ is positive, the transition matrix $\mathbf{P} = \mathbf{C} \cdot \mathbf{M}$ [see (28)] of the Markov chain is primitive (i.e., there exists $k > 0$: $\mathbf{P}^k$ is positive). In this case, it has been shown in [31] that the GA

TABLE V
NUMBER OF DATA INCLUDED IN THE TRAINING, VALIDATION, AND TEST SET FOR ALL THREE ENVIRONMENTS

	Measurement Set	Training Set	Validation Set	Test Set
1st Environment	2240	1110	480	650
2nd Environment	2350	1150	495	705
3rd Environment	1980	960	420	600

converges to the optimum solution if the best solution is maintained over time. This means that, starting from any arbitrary state (valid solution), the algorithm visits any other state (valid solution) within a finite number of transitions.

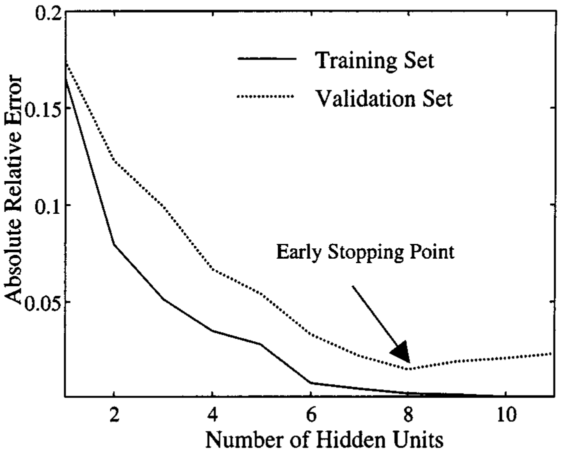


Fig. 11. Network performance, expressed in absolute relative error, versus the number of hidden neurons over data of training and validation set for the first environment.

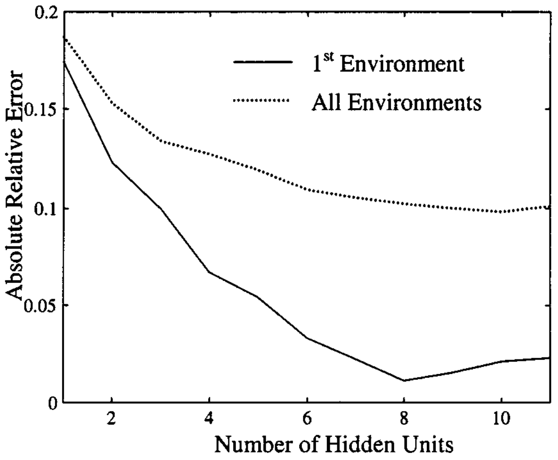


Fig. 12. Comparison of the network performance, expressed in absolute relative error, versus the number of hidden neurons over data of the validation set in case that one and all environments are used.

VI. RESULTS

In this section, we analyze the results obtained by applying the proposed neural network-genetic algorithm scheme to a manufacturing industry following the wound core technology. In particular, Section VI-A presents the performance of the neural network architecture as an accurate predictor of transformer iron losses, while Section VI-B indicates the iron loss reduction which is achieved using the combined neural-genetic scheme. Finally, Section VI-C discusses the economic advantages that arise by the use of the proposed scheme to the examined manufacturing industry.

A. Iron Loss Prediction

To predict transformer iron losses, we initially construct three industrial measurement sets (MSs), each of which corresponds to a specific environment. In particular, the measurement set of the first environment comprises 2240 actual industrial samples (transformers), while the set of the second and third environment 2350 and 1980 samples (transformers), respectively. Each sample is a pair of the eight attributes selected by the DT, (see Table III) and the associated actual specific iron losses of the transformer. The measurement set of each environment is randomly partitioned into three disjoint sets:

1) the training set;
2) the validation set;
3) the test set.

The training set is used to estimate the network parameters (i.e., weights), the validation set to terminate network training (see Section IV-A), while the test set to evaluate the network accuracy. Table V presents the number of data included in the training, validation and test set for the three examined environments.

Fig. 11 illustrates the network performance versus the number of hidden neurons over all data in the training and validation set for the first environment. In our case, the network performance is evaluated by the average absolute relative prediction error R_n, which is defined as follows:

$$R_n = \frac{1}{L} \sum_{\mathbf{t}_i \in S_{\text{tr}}} \frac{|S^a_{\mathbf{t}_i} - \hat{S}^a_{\mathbf{t}_i,n}|}{S^a_{\mathbf{t}_i}} * 100\% \tag{31}$$

where we recall that $S^a_{\mathbf{t}_i}$ are the actual (measured) specific iron losses of transformer $\mathbf{t}_i$ and $\hat{S}^a_{\mathbf{t}_i,n}$ the predicted ones in case that the number of hidden neurons of the network is equal n.

As is observed, the error on training set decreases monotonically for an increasing number of hidden neurons. Instead, the error on validation set decreases until eight hidden neurons are added and then it starts to increase. This is called early stopping point (eight hidden neurons) and is depicted in Fig. 11. When we look at the "training" curve (solid line), it appears that we could improve the network performance by using more than eight hidden neurons. This is due to the fact that the proposed constructive algorithm estimates the new network weights so that the output of the new added neuron compensates the current residual error [see (18)– (20)]. As a result, the error over data of the training set is driven to fall. In reality, however, what the network is learning beyond the early stopping point is essentially noise contained in the training data. For this reason, the "validation" curve (dotted line) increases beyond this point, indicating that the generalization performance of a network with more than eight hidden neurons begins to deteriorate, since overfitting of

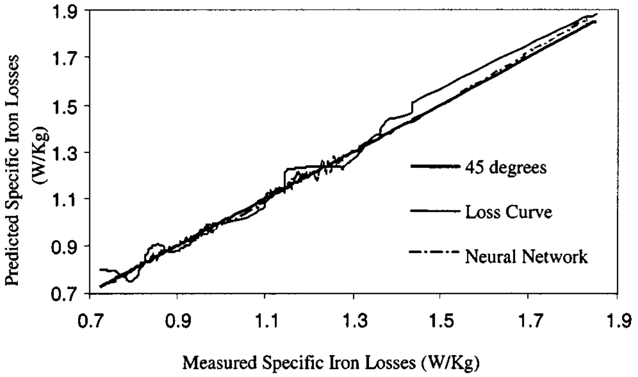


Fig. 13. Fractile diagram of transformer specific iron losses for the first environment.

TABLE VI
TRANSFORMER IRON LOSS PREDICTION USING THE LOSS CURVE AND THE NEURAL NETWORK METHOD

Average Relative Prediction Error

	Environment		
Method	#1	#2	#3
Loss curve	2.93	3.18	3.31
MLP	0.95	1.06	1.13

the training data is accomplished. As a result, eight hidden neurons are selected for the neural network associated to the first environment. Similar prediction accuracy is observed for the networks corresponding to the other two environments, where the most appropriate number of hidden neurons is estimated to be eight and nine, respectively.

The prediction accuracy of the neural network versus the number of hidden neurons when we mix data of all examined environments in the validation set, is depicted in Fig. 12 (dotted line). In this case, the environment type is fed as additional input (attribute) to the neural network. The absolute relative error obtained using data only of the first environment is also plotted in this figure for comparison purposes. As is observed, smaller prediction error is achieved if the network has trained using data of the same environment.

Fig. 13 presents the fractile diagram or the Quantile–Quantile plot [32] of the specific iron losses for the first environment. In this figure, the real (measured) specific iron losses are plotted versus the predicted (by the loss curve and the neural network) specific iron losses. Perfect prediction lies on a line of 45° slope. It is observed that the prediction provided by the neural network (dotted line in Fig. 13) is closer to the optimal line of 45° than the loss curve prediction (solid line in Fig. 13).

Table VI presents the average absolute relative error on test set, for the three environments considered. In all cases, the neural network improves the prediction accuracy by more than 65%.

Despite the very good performance of the neural network in predicting iron losses, its application during transformer construction must be monitored. The reason is that during the manufacturing, the conditions to which the neural network has been trained may change. The way of monitoring the performance of the neural network is to define an upper limit for the average absolute relative error. If the average error of a production batch is above this limit then the weight adaptation mechanism is activated and the neural network is retrained using the algorithm described in Section IV-C. For example, Fig. 14 shows the average absolute relative error for various production batches of the first environment, the average error on the test set and the upper limit of the average error. More specifically the average error on the test set is equal to 0.95% as given in Table VI and the upper limit is set 10% above the average error of the test set, i.e., the upper limit is 1.045%. It is observed that at the 19th production batch the average absolute relative error exceeds the defined upper limit of 1.045%. Consequently, the weight adaptation mechanism is activated. After adaptation, the average absolute relative error on the test set is 0.94% and the new upper limit is set to 1.034%, i.e., 10% above the average error of the test set. Fig. 15 presents the average absolute relative error for various production batches after the adaptation of the neural network weights. It is observed that for the following 11 production batches the average absolute relative error is within the tolerated interval.

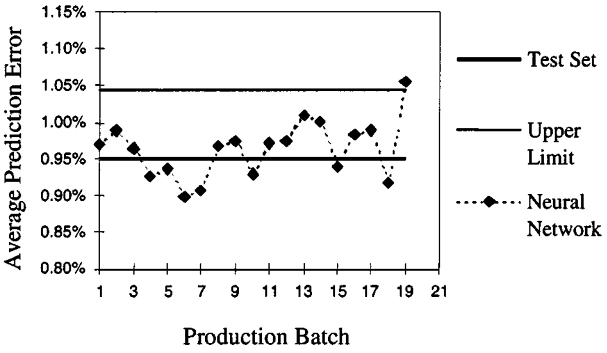


Fig. 14. Evolution of average absolute relative error through various production batches before the adaptation of the neural network weights.

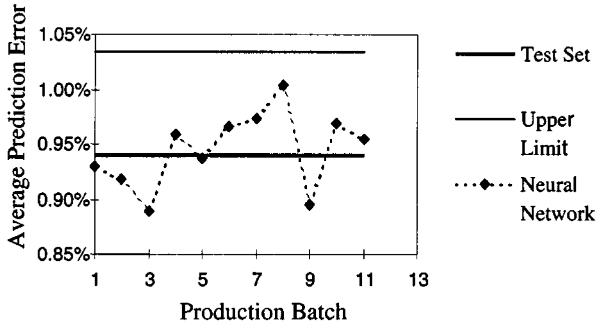


Fig. 15. Evolution of average absolute relative error after the adaptation of the neural network weights.

B. Iron Loss Reduction

The proposed GA-based grouping process was used in order to group 100 small and 100 large cores of the same production batch of 50 transformers, 100 kVA and 50 Hz of first environment.

Fig. 16 shows the minimum value, over the whole population, of the total predicted iron losses of the best chromosome versus the cycle (or generation) of the GA. The total predicted

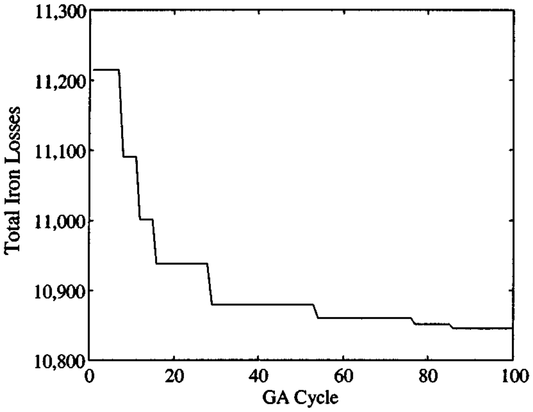


Fig. 16. GA convergence: total iron losses versus the GA cycle.

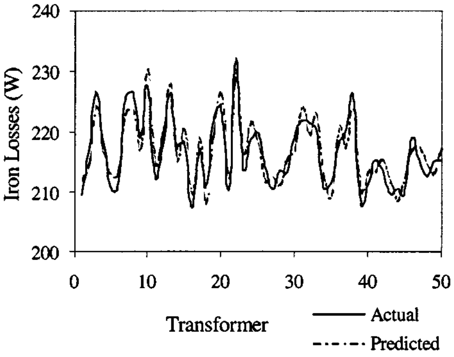


Fig. 17. Evaluating the performance of the GA for a production batch of the first environment.

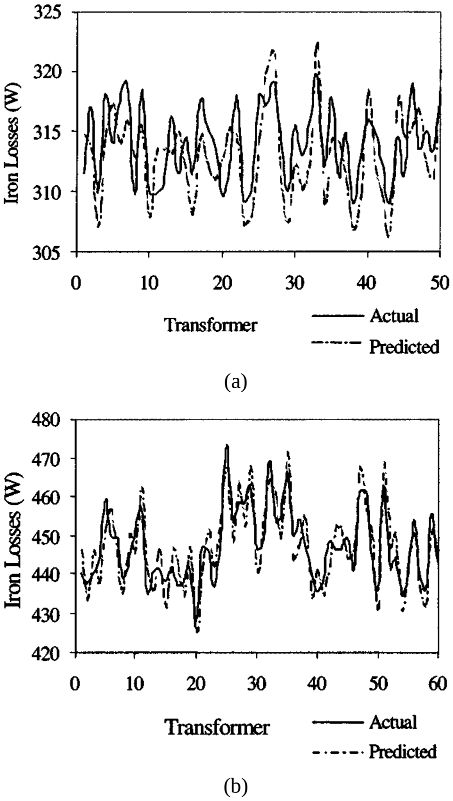


Fig. 18. Evaluation of the GA in the other two considered environments: (a) production batch of 50 transformers, 160 kVA, 50 Hz of the second environment and (b) production batch of 60 transformers, 250 kVA, 50 Hz, of the third environment.

iron losses decrease as the GA cycle increases, until it reaches a minimum value of 10 846 W at 86th generation.

The output of the genetic algorithm grouping process is not only the minimum value of the total predicted iron losses of the best chromosome (i.e., 10 846 W for the example of Fig. 16) of the 50 transformers. It also provides for each one of the 50 transformers the optimal core arrangement and the associated predicted iron losses.

Fig. 17 evaluates the performance of the GA comparing the predicted with the actual iron losses (measured after the transformer construction) for each one of the 50 transformers, for the example of Fig. 16. In this case, the *average absolute relative error* is equal to 1.03%. Fig. 18 confirms the very good performance of the GA in the other two environments considered. In this case, the average absolute relative error is 0.95% and 1.17%, respectively.

C. Exploitation of the Results

The proposed combined neural network-genetic algorithm approach has been coded in a genetic algorithm neural network (GANN) toolbox and is currently used in the considered industrial environment. Using appropriate data acquisition systems, measurements are collected and fed to the GANN toolbox as well as to a statistical processing and graphical visualization toolbox.

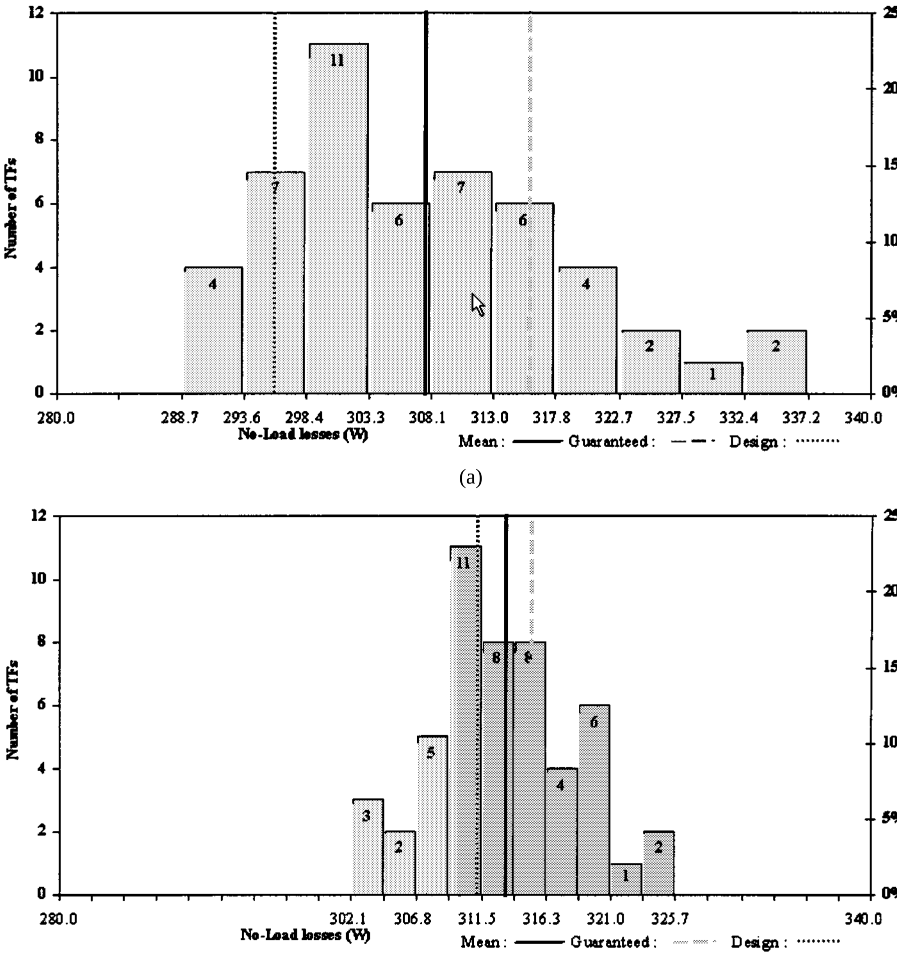
The application of the combined neural network-genetic algorithm provides significant economic advantages for the transformer manufacturer. More specifically, it helps a) to reduce transformer iron losses, b) to reduce the cost of materials, and c) to avoid paying loss penalties.

Fig. 19 shows the iron loss distribution of a production batch of 50 transformers, 160 kVA, into two different periods. In period 1, the method of grading into quality classes is used as grouping process, while in period 2, the proposed neural network-genetic algorithm is applied. In both periods, the transformer specification is the same and the desired (guaranteed) no-load losses are 315 W.

Since large deviations in iron losses were observed in period 1, the designed iron losses were 296 W, namely 6% lower than the desired iron losses. On the other hand, in period 2 the iron loss deviations were much lower and therefore the designed iron losses could be raised to 311 W, that is, 1.3% lower than the desired iron losses.

From Fig. 19 it can be concluded that the iron loss results of period 2 are by far better than the results of period 1. More specifically, the variation of iron losses is smaller (23.6 W for period 2, instead of 48.5 W for period 1) and the mean value of iron losses (313.15 W, instead of 307.31 W) is closer to the desired iron losses. The results of Fig. 19 are summarized in Table VII.

The results presented in Fig. 19 and Table VII have been confirmed for a large number of transformer constructions. Table VIII presents the evolution of the average absolute relative error of iron losses for the two different periods. In this case, the average absolute relative error is defined by the difference of the designed iron losses from the desired



(a)

(b)

Fig. 19. Iron loss distribution of 50 transformers, 160 kVA. (a) Grading into quality classes is used as grouping process (Period 1) and (b) the proposed GA is used as grouping process (Period 2).

TABLE VII
Iron Loss Results of 50 Transformers into Two Different Periods

				Actual iron losses (W)			
Period	Desired losses (W)	Designed losses (W)	Safety margin (%)	Min	Max	Mean	Standard deviation
1	315.00	296.00	6.0	288.70	337.20	307.31	11.18
2	315.00	311.00	1.3	302.10	325.70	313.15	5.37

TABLE VIII
Evolution of Average Absolute Relative Error into Two Different Periods

		Average absolute relative error (%)	
Period	Grouping process	Mean	Standard deviation
1	Quality classes	5.4	0.9
2	Genetic algorithm - neural network	1.1	0.4

losses. Each of the periods corresponds to a different grouping process of cores (period 1 refers to the "quality class" grouping method, while period 2 to the proposed neural network-genetic algorithm).

The proposed scheme leads to reduction of the production cost. Let us assume that it is required to construct 50 kVA transformers with 131 W guaranteed losses. In period 1, a 6.0% safety margin was satisfactory, and the transformer was calculated to have 123 W designed losses. In period 2 the transformer is evaluated to have 129 W designed losses (safety margin 1.5%). The ability for the reduction of the safety margin between the designed and the desired iron losses offers significant savings of magnetic material. Moreover, the reduction of the weight of the magnetic material leads to the construction of transformers of smaller dimensions. The latter results in the reduction of the weight of the material of the windings (copper), insulating materials and transformer oil.

TABLE IX
COMPARISON OF THE COST OF MATERIALS FOR THE SAME GUARANTEED LOSSES

		Magnetic material	Copper	Insulating materials	Oil	Total materials
Period 1	Weight (Kg)	152.08	58.13	12.19	101.24	-
	Cost ($)	265.54	184.54	38.70	57.85	546.63
	Cost (%)	100.00	100.00	100.00	100.00	100.00
Period 2	Weight (Kg)	146.00	57.20	12.00	98.00	-
	Cost ($)	254.92	181.59	38.10	56.00	530.60
	Cost (%)	96.00	98.40	98.44	96.80	97.07

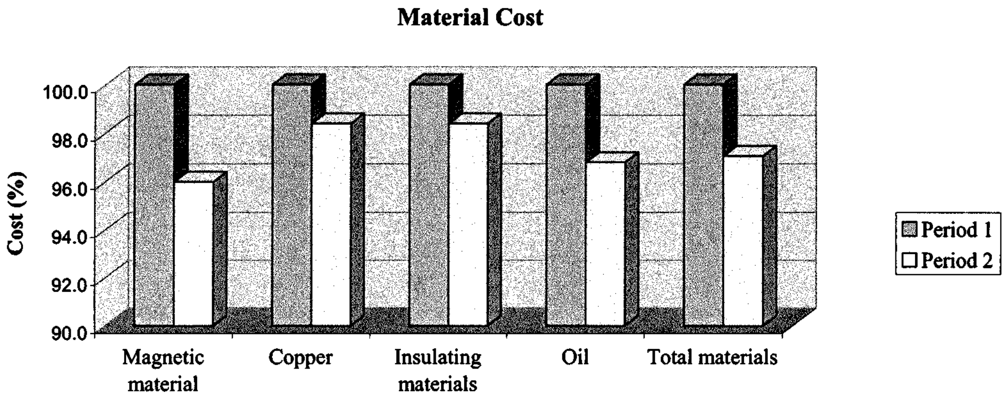


Fig. 20. Reduction of the cost of materials of two different 50 kVA transformer designs.

TABLE X
COMPARISON OF THE COST OF MATERIALS INTO TWO DIFFERENT PERIODS

				Cost of materials (% of Period 1)				
kVA	Period	Desired losses (W)	Designed losses (*W*)	Magnetic material	Copper	Insulating materials	Oil	Total materials
100	1	220	208	100.00	100.00	100.00	100.00	100.00
100	2	220	219	95.80	98.38	98.38	96.77	97.06
160	1	315	296	100.00	100.00	100.00	100.00	100.00
160	2	315	311	96.10	98.44	98.44	96.88	97.23
250	1	446	425	100.00	100.00	100.00	100.00	100.00
250	2	446	448	96.10	98.38	98.38	96.75	97.14

Table IX shows the reduction of transformer cost achieved in period 2 in relation to the period 1 for the 50 kVA transformer design. For both periods the cost of materials is considered to be $1 746/Kg for the magnetic material, $3 175/Kg for the copper and the insulating materials, and $571/Kg for the transformer oil. The cost reduction results are presented in Fig. 20.

Table X presents the reduction of the cost of materials achieved for transformers of 100, 160, and 250 kVA for period 2 in relation to period 1. In this table, the cost of each one of the materials is expressed as a percentage of the respective cost of period 1.

From Table X, it can be seen that for the three different types (kVA) of transformer designs an approximately 3.0% reduction of the cost of the four main materials of transformer is achieved. This reduction is significant since the four above-mentioned materials represent about the 75% of the total cost of transformer materials.

VII. CONCLUSION

In this paper, neural networks are combined with GAs in order to reduce transformer iron losses. More specifically, neural networks are used to predict iron losses of the wound core type transformers prior to their assembly. Each of the neural networks is suited to a different environment, i.e., to a certain supplier, grade, and thickness of magnetic material. The prediction is based on measurements on the individual cores taken at the early stages of transformer construction. Furthermore, the GAs are used in combination with neural networks in order to reduce the transformer iron losses. Application of the proposed artificial intelligence framework to a transformer manufacturing industry has verified the accuracy of the prediction in all the examined environments and reduction of the transformer losses has been achieved. These results provide significant economic gains to the transformer manufacturer.

References

[1] B. W. McConnell, "Increasing distribution transformer efficiency: Potential for energy savings," *IEEE Power Eng. Rev.*, vol. 18, pp. 8–10, July 1998.

[2] S. A. Stigant and A. C. Franklin, *The J&P Transformer Book*. London, U.K.: Butterworth, 1973.

[3] G. Lian, Y. Ruoping, and C. Pizhang, "An equivalent magnetization surface current approach of calculation 3D-leakage fields of a transformer," *IEEE Trans. Power Delivery*, vol. PWRD-2, pp. 817–822, July 1987.

[4] G. F. Mechler and R. S. Girgis, "Calculation of spatial loss distribution in stacked power and distribution transformer cores," *IEEE Trans. Power Delivery*, vol. 13, pp. 532–537, Apr. 1998.

[5] R. S. Girgis, D. A. Yannucci, and J. B. Templeton, "Performance parameters on power transformers using 3D magnetic field calculations," *IEEE Trans. Power Apparat. Syst.*, vol. PAS-103, pp. 2708–2713, Sept. 1984.

[6] R. S. Girgis, E. G. teNijenhuis, K. Gramm, and J. E. Wrethag, "Experimental investigation on effect of core production attributes on transformer core loss performance," *IEEE Trans. Power Delivery*, vol. 13, pp. 526–531, Apr. 1998.

[7] N. D. Hatziargyriou, J. M. Prousalidis, and B. C. Papadias, "A generalized transformer model based on the analysis of its magnetic circuit," *Proc. Inst. Elect. Eng.*, vol. 140, no. 4, pp. 269–278, July 1993.

[8] B. C. Papadias, N. D. Hatziargyriou, J. A. Bakopoulos, and J. M. Prousalidis, "Three phase transformer modeling for fast electromagnetic transients," *IEEE Trans. Power Delivery*, vol. 9, pp. 1151–1159, Apr. 1994.

[9] D. Paparigas, D. Spiliopoulos, S. Elefsiniotis, and J. Bakopoulos, "Linear statistical model for the prediction of transformer iron losses," in *Proc. THERMIE Euro. Symp. Technol. Econom. Advantages Use Low Loss Distrib. Transformers*, Athens, Greece, May 1994.

[10] K. Milolidakis, "Statistical quality control of the transformer core production process," Univ. Crete, Chania, Greece, Tech. Rep., Sept. 1993.

[11] D. Gounidis, "Iron loss prediction of individual cores and transformers using the linear statistical model," B.Sc. thesis, Univ. Crete, Chania, Greece, July 1995.

[12] P. S. Georgilakis, "Contribution of artificial intelligence techniques in the reduction of distribution transformer iron losses," Ph.D. dissertation, Nat. Tech. Univ. Athens, Athens, Greece, Mar. 2000.

[13] P. S. Georgilakis, N. D. Hatziargyriou, S. S. Elefsiniotis, D. G. Paparigas, and J. A. Bakopoulos, "Creation of an efficient computer-based environment for the reduction of transformer industrial cycle," in *Proc. IEE Med. Power Conf.*, Nicosia, Cyprus, Nov. 1998.

[14] N. D. Hatziargyriou, P. S. Georgilakis, D. S. Spiliopoulos, and J. A. Bakopoulos, "Quality improvement of individual cores of distribution transformers using decision trees," *Int. J. Eng. Intell. Syst.*, vol. 6, no. 3, pp. 141–146, Sept. 1998.

[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[16] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[17] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, "Learnability and the Vapnik-Chervornenkis dimension," *J. ACM*, vol. 36, pp. 929–965, 1989.

[18] T.-Y. Kwok and D.-Y. Yeung, "Objective functions for training new hidden units in constructive neural networks," *IEEE Trans. Neural Networks*, vol. 8, Sept. 1997.

[19] S. Kollias and D. Anastassiou, "An adaptive least squares algorithm for the efficient training of artificial neural networks," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1092–1101, Aug. 1989.

[20] L. Wehenkel, *Automatic Learning Techniques in Power Systems*. Norwell, MA: Kluwer, 1998.

[21] L. Wehenkel and Y. Jacquemart, "Automatic learning methods to dynamic security assessment," in *IEEE Int. Conf. Power Ind. Comput. Applicat.*,, Tutorial course notes.

[22] T. O. Kvålseth, "Entropy and correlation: Some comments," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, pp. 517–519, May/June 1987.

[23] C. Park, M. A. EL-Sharkawi, and R. J. Marks II, "An adaptively trained neural network," *IEEE Trans. Neural Networks*, vol. 2, pp. 334–345, May 1991.

[24] A. Doulamis, N. Doulamis, and S. Kollias, "On line retrainable neural networks: Improving the performance of neural networks in image analysis problems," *IEEE Trans. Neural Networks*, vol. 11, pp. 137–155, Jan. 2000.

[25] N. Doulamis, A. Doulamis, and S. Kollias, "A dynamically trained neural network for machine vision applications," *J. Mach. Graph. Vis.*, vol. 7, no. 1/2, pp. 113–124.

[26] D. Patterson, *Artificial Neural Networks: Theory and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[27] E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[28] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. New York: Springer-Verlag, 1994.

[29] X. Qi and F. Palmieri, "Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space, Part I: Basic properties of selection and mutation," *IEEE Trans. Neural Networks*, vol. 5, pp. 102–119, Jan. 1994.

[30] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. Fifth Int. Conf. Genetic Algorithms*, S. Forrest, Ed., San Mateo, CA, 1993, pp. 416–423.

[31] G. Rudolph, "Convergence analysis of canonical genetic algorithms," *IEEE Trans. Neural Networks*, vol. 5, pp. 96–101, Jan. 1994.

[32] H. Kobayashi, *Modeling and Analysis*. Reading, MA: Addison-Wesley, 1981.

Pavlos S. Georgilakis (S'98–M'01) was born in Chania, Greece, in 1967. He received the Diploma degree in electrical and computer engineering and the Ph.D. degree, both from the National Technical University of Athens (NTUA), Athens, Greece, in 1990 and 2000, respectively.

In 1994, he joined Schneider Electric AE, Inofyta Viotia, Greece, and worked in the Development and Quality Control Departments of its Industrial Division. He is currently R&D Manager of Schneider Electric AE. His research deals with application of artificial intelligence techniques to transformer design.

Dr. Georgilakis is a member of the Technical Chamber of Greece.

Nikolaos D. Doulamis (S'96) was born in Athens, Greece, in 1972. He received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA) in 1995 with the highest honor. He is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department of the NTUA supported by the Bodosakis Foundation Scholarship.

His research interests include image/video analysis and processing and multimedia systems.

Mr. Doulamis has received several awards and prizes from the Technical Chamber of Greece, the NTUA, and the National Scholarship Foundation, including the Best New Engineer Award at National Level, the Best Diploma Thesis Awards and the NTUA Medal.

Anastasios D. Doulamis (S'96) was born in Athens, Greece, in 1972. He received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA) in 1995 with the highest distinction. He is currently pursuing the Ph.D. degree at NTUA.

He joined the Image, Video, and Multimedia Lab of NTUA in 1996. His research interests include neural networks to image/signal processing, multimedia systems, and video coding based on neural networks systems.

Mr. Doulamis was awarded the Best New Engineer at the national level and received the Best Diploma Thesis Award from the Technical Chamber of Greece. He was also given the NTUA Medal, and has received several awards and prizes from the National Scholarship Foundation.

Nikos D. Hatziargyriou (S'80–M'82–SM'90) was born in Athens, Greece, in 1954. He received the Diploma degree in electrical and mechanical engineering from the National Technical University of Athens (NTUA) in 1976. He received the M.Sc. and Ph.D. degrees from The University of Manchester Institute of Science and Technology, Manchester, U.K., in 1979 and 1982, respectively.

He is currently Professor at the Power Division of the Electrical and Computer Engineering Department at NTUA. His research interests include dynamic security, artificial intelligence techniques, dispersed generation, and renewable energy sources.

Dr. Hatziargyriou is Chairman of the IEEE Greek Power Chapter, a member of CIGRE SC38, and a member of the Technical Chamber of Greece.

Stefanos D. Kollias (S'84–M'85) was born in Athens, Greece, in 1956. He received the Diploma degree in electrical engineering from the National Technical University of Athens (NTUA) in 1979, the M.Sc degree in communication engineering from the University of Manchester Institute of Science and Technology, Manchester, U.K., in 1980, and the Ph.D. degree in signal processing from the Computer Science Division of NTUA in 1984.

Since 1986, he has been with the NTUA, where he is currently a Professor. From 1987 to 1988, he was a Visiting Research Scientist in the Department of Electrical Engineering and the Center for Telecommunications Research, Columbia University, New York, NY. His current research interests include image processing and analysis, neural networks, image and video coding, and multimedia systems. He is the author of more than 140 articles in the aforementioned areas.