15th IMEKO TC10 Workshop on Technical Diagnostics
Technical Diagnostics in Cyber-Physical Era
Budapest, Hungary, June 6-7, 2017

Unsupervised learning-based hierarchical diagnostics of analog circuits

Piotr Bilski[1]

[1] *Institute of Radioelectronics and Multimedia Technology, Warsaw University of Technology, ul. Nowowiejska 15/19 00-665 Warsaw, pbilski@ire.pw.edu.pl*

***Abstract* – The paper presents the automated diagnostic method of analog circuits combining supervised and unsupervised learning. The purpose of this sophisticated approach is to effectively distinguish easily identified states of the analyzed system from the more difficult ones. The latter are often related with existence of ambiguity groups, which create problems for the distinction of specific states. The proposed approach uses subsequently two algorithms. The first one extracts and learns to classify such "simple" training examples. The second one aims at the classification of more difficult ones. For both stages, Self-Organizing Maps and Random Forest were used, respectively. The scheme was tested on the model of the 3rd order Bessel highpass filter, confirming effectiveness of the approach.**

I. INTRODUCTION

The modern diagnostics of analog system strongly relies on sophisticated data processing methods. The Simulation Before Test (SBT) approach is typically used to analyze analog circuits [1], which, although not the most popular, are still in use. Their applications include, among others, RF or audio technologies [2]. Therefore it is important to obtain knowledge about influence of the parameters of the System Under Test (SUT) on its behavior. Determining testability of the circuit allows for identifying the simplest faults and the most difficult to detect and locate. The ability to do so depends on the analyzed domains [3], the set of available nodes, and features extracted from measured signals. The significant problem here is the existence of ambiguity groups (AG) [4], which make the flawless separation of particular faults impossible. To ensure the high diagnostic accuracy, the first step should be detection of AG, followed by the use of identification methods adjusted to separate the most difficult cases.

Methods to monitor the SUT with the highest accuracy (evaluated as the number of errors made by the fault classifier operating on the testing data set) often belong to the Artificial Intelligence (AI) domain, which covers algorithms from the wide range of available approaches. The key issue is extraction of knowledge from available data. The most popular are supervised learning algorithms, such as Multi-Layered Perceptrons (MLP) [5], Support Vector Machines (SVM) [6] or Random Forest (RF) [7]. However, the scheme where the fault categories are assigned by the designer to the vector of the measured symptoms (further called example), is susceptible to the AG, where similar examples belong to different SUT states and are difficult to distinguish. This suggests application of the unsupervised learning first to separate AG from other states.

The paper presents the novel diagnostic scheme using both unsupervised and supervised learning methods. Benefits of the proposed methodology are twofold. First, it is possible to increase the diagnostic accuracy by focusing on the difficult examples. Second, the combined approach simplifies the form of knowledge stored by both classifiers. The diagnostic module is used to identify and locate parametric faults in the analog circuit (i.e. its topology remains intact).

The paper structure is as follows. In Section II the diagnostic architecture is presented. Section III describes the data set used for training and testing the scheme. In Section IV unsupervised and supervised learning classifiers implemented in the research are presented. In Section V the analyzed SUT, i.e. the analog highpass filter is introduced. Section VI contains experimental results, comparing the effectiveness of various classifier configurations. Section VII concludes the paper with the summary about the effectiveness of the approach.

II. HIERARCHICAL DIAGNOSTICS SCHEME

The proposed automated diagnostic scheme is presented in Fig. 1. This is the data-driven approach, which requires data sets for both training and testing the classifiers, indicated as L and T, respectively. The set L is used to train both classifiers. The processing of "real-life" examples, represented by the set T is compliant with the following procedure. In the first step the classifier, aimed at detecting easily distinguishable states (collected in the T_1 subset) is used ("introductory classification" step). Its task is to detect one of "simpler" categories or invoke the more sophisticated classifier for the fault identification. The second module attempts to determine the SUT state (collected in examples belonging to the T_2 subset) using different form of knowledge than the introductory classifier. The latter is simple and fast, its knowledge relatively compact and memory efficient. The example of such a method is the Decision Tree (DT). The detailed classifier should consider more difficult fault states, therefore the sophisticated algorithm has to be used. Methods considering measurements in uncertainty conditions include SVM, RF or Fuzzy Logic (FL). They should be used as the auxiliary classifier ("detailed classification" step) to separate the hardly distinguishable SUT states.

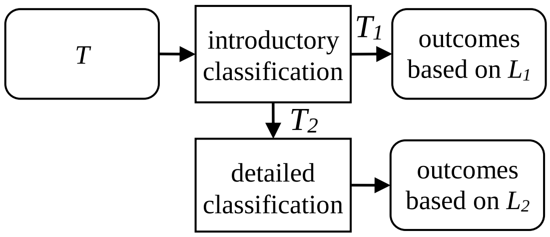

Fig. 1. Architecture of the hierarchical AI-based diagnostic scheme.

The proposed architecture assumes that machine learning is used to train classifiers. The need to divide examples into two subsets originates from the concept of the AG [8], identified using the comparison between features in the vector examples. Therefore the key operation during the design of the scheme from Fig. 1 is the AG detection (see Fig. 2), followed by the separation of the original training set L into two subsets: the one containing examples being part of the AG (L_2), and the second containing examples, which should be easy to identify (L_1). The first set is used to train the unsupervised learning classifier (as there are no problems of incorrectly identifying subsequent faults here). The second set is used to train the supervised learning classifier.

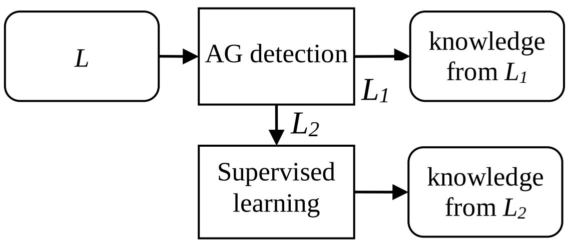

Fig. 2. Knowledge extration for the hierarchical diagnostic scheme.

The key operation in the scheme from Fig. 2 is the AG detection. Details of this operation were presented in [8]. Here the method can be extended to not only detect AG, but also extract knowledge required to identify examples not belonging to such sets. The AG is defined as the set of examples belonging to different categories, but being similar based on the analyzed features. Therefore the key operation is to define the similarity measure between the examples. It is the most important parameter of the implemented unsupervised learning (clustering) approach. The second issue is the selection of the threshold, below which the similarity indicates that different examples are difficult to distinguish.

III. DATA SET DESCRIPTION

Data-driven approaches extract knowledge from data sets having the structure of the table with n examples $\boldsymbol{e}$, each represented by m features $\{f_1, \dots, f_m\}$. The additional element is the fault category $c(\boldsymbol{e})$ assigned by the designer. As before [7], it is the code determining the identifier of the parameter responsible for the changes in the SUT's behavior, and the deviation of its actual value from the nominal one. The exception is the value "*0*", indicating the fault-free system. Otherwise the code presents the intensity of the parameter change. It has the form like *"21"*, which means that the second parameter has the value greater than the nominal, while *"-32"* means the third parameter has the value much smaller than the nominal. Such a definition of the fault code not only allows for the intuitive description of the actual SUT states, but also enables the additional analysis of faults varying only in the intensity, such as *"12"* and "*11*". They describe the same parameter with identical direction of changes, but with different deviations from the nominal value. Failure in distinguishing between these two categories is not as problematic as the incorrect identification of the parameter. It may indicate low sensitivity of the SUT on the changes in this parameter. The codes are assigned according to the actual value of the parameter provided that observable changes in the SUT's behavior are greater than 15% of the ideal response. Otherwise, all changes are treated as the behavior within the tolerance margin (and are labeled with the *"0"* code). The difference between similar states like "greater than" or "much greater than" depends on the designer's choice. In the presented research the values above 50% of the nominal ones are treated as "much greater" or "much smaller" (depending on the direction of changes). The number of the codes for each parameter depends on the desired resolution of the diagnostic system (here there are four fault categories for each parameter, two in the positive and two in the negative direction). Both sets L and T have identical form. Their cardinality may be the same (as is in the presented project); alternatively the cross-validation may be used to divide the data into L and T.

The main problem in the correct fault identification is that different changes in SUT parameters manifest similarly in the responses, making such exampled hardly distinguishable. Although there are multiple methods of eliminating AG (such as increasing the number of accessible nodes or features) [3], they are rarely eliminated completely, requiring time-consuming analysis of the SUT behavior.

The general requirement to determine the AG is to autonomously find similar (i.e. with distance d below the threshold θ) examples from L belonging to different fault categories:

$$\left|d(\boldsymbol{e}_i) - d(\boldsymbol{e}_j)\right| \leq \theta : c(\boldsymbol{e}_i) \neq c(\boldsymbol{e}_j) \tag{1}$$

In [8] the Self-Organizing Map was used to for this purpose. Here its application is extended to the role of the introductory classifier.

Accuracy of the hierarchical classification module trained on the set L and tested on T is measured as:

$$\begin{aligned} acc &= \alpha \cdot acc_{\text{int}} + \beta \cdot acc_{\text{det}} = \\ &= \frac{\left|\boldsymbol{e} \in T : c(\boldsymbol{e}) = h_{\text{int}}(\boldsymbol{e})\right| + \left|\boldsymbol{e} \in T : c(\boldsymbol{e}) = h_{\text{det}}(\boldsymbol{e})\right|}{|T|} \end{aligned} \tag{2}$$

where $|T|$ is the cardinality of the testing set, acc_{int} and acc_{det} stand for introductory and detailed classification accuracies (weighted by α and β depending on the number of examples from T classified by both algorithms), respectively. The actual category of the example $c(\boldsymbol{e})$ is compared to the decision (hypothesis) of the particular classifier about this example, $h_{int}(\boldsymbol{e})$ or $h_{det}(\boldsymbol{e})$, respectively.

IV. DESCRIPTION OF THE IMPLEMENTED CLASSIFIERS

This section introduces two classification methods (SOM and RF) used in the hierarchical diagnostic system. Other approaches may be used for this purpose. Both algorithms were implemented in Matlab environment.

A. Self-Organizing Map

This is the one-layered unsupervised learning neural network (Fig. 3) with wide range of applications [9,10]. Computational units (neurons) are located in the plane – each potentially representing the separate category. The data set L (but without the category labels) is provided to the input of the network, therefore its number of inputs is equal to the number of features m in the example. Sequentially, neurons are taught to react stronger to the specific examples, specializing in detecting them. The result of training is the set of adjusted units, each representing the newly created category.

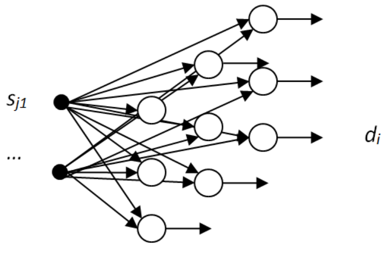

Fig. 3. SOM architecture.

The AG detection consisted in comparing fault codes of all examples detected by the same neuron. If different codes are identified, such examples will be difficult to distinguish from each other. This leads to separating computational units into three categories:

- N_1, detecting examples belonging to only one category, therefore not being part of the AG. Such neurons are labeled by the proper fault codes. All examples detected by them are added to the set L_1.
- N_2, detecting examples from L belonging to multiple categories and therefore being the part of the AG. This neuron is labeled as *"-1"*, indicating that the auxiliary classifier will have to be run to make the identification. All examples detected by it are added to the set L_2.
- N_3 (dead ones), not reacting to any examples from the set L. Such a neuron is labeled as "-Inf", which means it should never be active during the identification process.

The trained SOM can then be used as the introductory classifier (see Fig. 1) by presenting neurons reacting the strongest to the examples from the set T. It is done in the following way:

1. if the processed example from the set T is identified by the neuron pointing unanimously at the fault code, the latter is returned as the diagnostic outcome.
2. If the example is identified by the neuron labeled as the AG, the auxiliary classifier is executed.
3. The undesired outcome during the testing phase is activation of the dead neuron, not representing any category. In such a case, the example is also processed by the detailed classifier. Experiments show that the accuracy for such examples is low, therefore the number of such situations should be minimized.

Parameters of the SOM, which have to be checked as they influence the extracted knowledge, include the number of neurons in the network (related to the percentage of dead units) and the learning strategy (Winner Takes Most against the Winner Takes All). Too small number of neurons leads to the large AG, while too large causes the excessive percentage of dead neurons. To minimize the latter, the WTM strategy was used, allowing for training the larger number of units. The minimal number of neurons is equal to the number of categories in L.

B. Random Forest

This is one of the supervised learning classifiers working well in the uncertainty conditions [11]. This makes it a good choice for identifying hardly distinguishable faults. The advantages over the SVM include legible knowledge and simpler configuration. On the other hand, FL, having comparable characteristics, does not have the machine learning module implemented.

The RF classifier is trained on the set L_2, separated from the original set L by the SOM. The structure of the forest covers the set of decision trees (Fig. 4), which are all generated on the same set L_2. The structure of every tree is similar, as the result of the tree induction algorithm.

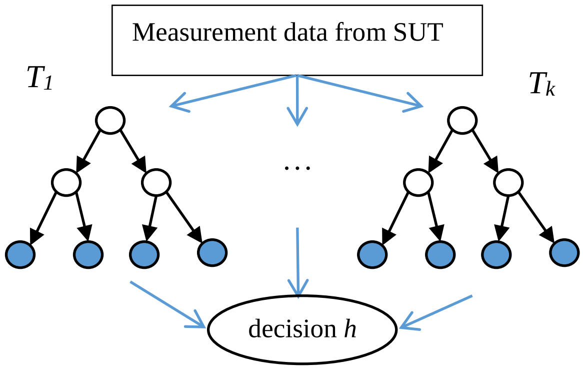

Fig. 4. Random Forest structure.

Differences between trees are in the content of subsequent nodes. Induction consists in generating each node by adding the test to it (the threshold value of the selected feature, which allows for dividing processed examples into two subsets, according to the comparison with the threshold). The entropy criterion is used to select the best test from available candidates. In the RF generation, the test in the node is randomly selected from the group of the candidates, which are sorted according to the entropy criterion. This way there is the chance of avoiding overlearning during the tree construction.

Application of the RF to the diagnostics consists in checking the examples by all trees. Their separate outcomes are combined into the single result h by the voting mechanism. Here the majority voting is used. Tested parameters of the RF are the number of trees and the size of the candidates' list.

V. SUT DECRIPTION

The Bessel highpass filter (Fig. 5) of the third order was used as the testing example. It is the relatively simple circuit, still present in the specific applications (such as audio processing or telecommunications).

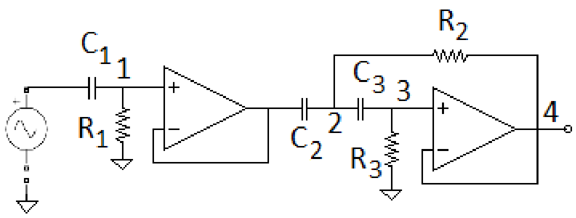

Fig. 5. Scheme of the 3rd order Bessel filter.

The diagnosed parameters are resistances and capacitances, with the nominal values: R_1=2.1kΩ, R_2=1.65kΩ, R_3=3.16kΩ, C_1=C_2=C_3=100nF. These parameters were indexed sequentially, i.e. resistances were labeled as 1, 2 and 3, while capacitances – as 4,5 and 6. The circuit was simulated in the Simulink environment. Tolerances of elements were introduced to make the case more realistic. Their three levels were considered, leading to models of "cheap", "moderate" and "expensive" circuits (further referenced to as SUT_1, SUT_2 and SUT_3, respectively). Random values (with uniform distribution) affecting the SUT parameters were, 5, 10 and 20 percent, respectively, which are actual tolerances of resistances and capacitors. It was expected that the more expensive SUTs are easier to diagnose (because of the smaller spread of parameter values). The frequency response of the filter is in Fig. 6, with the magnitude being U_{out}/U_{in}. The system was analyzed in the time domain (which is faster to use than the frequency domain). It was excited with the sinusoidal signal of the frequency *f=1kHz*, which facilitates detecting any changes in the SUT behavior after modifying values of subsequent parameters.

Each example consisted of 36 symptoms, being the first three maximum and minimum values of the sinusoidal response measured at nodes 1,2,3 and 4 (Fig. 5), as well as the first three time instances of zero crossings. This results in 9 symptoms collected at each node.

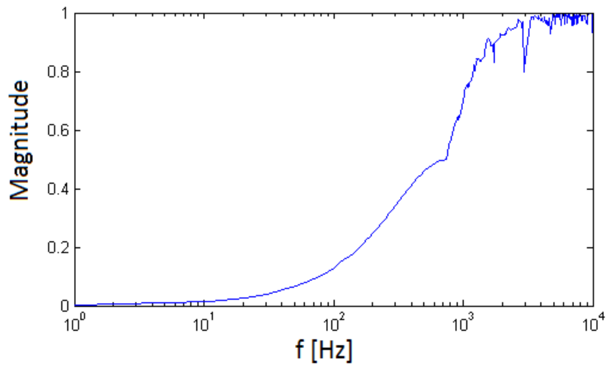

Fig. 6. Filter's frequency response.

VI. EXPERIMENTAL RESULTS

Experiments were divided into three stages. The checked parameters of the architecture included the SOM size and the number of trees in the RF, which influence the structure of sets L_1 and L_2 and the number of AG. This section explains subsequent tests. Evaluated measures include the training t_{tr} and testing t_{te} times of classifiers (i.e. processing the training and testing sets), and their accuracies (2).

A. Dependence between the SOM size and the AG structure

To simplify experiments, the square shape of the network was selected. This means that the number of neurons is equal to k^2, where k is the number of neurons in the row. During the preliminary analysis it turned out that the rectangular shape does not influence the diagnostic accuracy. Results of the SOM training for different network sizes k in SUT_1 are in Table 1. Because knowledge stored by SOM is obtained stochastically, presented values were averaged after ten trials of training and testing the network. Symbols used in the table represent the size of the sets for both classifiers $|L_1|$ and $|L_2|$, the number of ambiguity groups |AG|, the training time and the number of dead neurons taking part in the fault identification n_{dn} (i.e. having the strongest response to any of examples from the set T despite the fact they did not specialize in any category during the training).

Table 1. Results of the SOM generation.

k	$\|L_1\|$	$\|L_2\|$	$\|AG\|$	acc_{int}	t_{tr} [ms]	$\|n_{dn}\|$
6	24	78	22	0.85	0.19	4
7	36	66	20	0.85	0.29	9
8	56	46	17	0.81	0.37	10
9	58	44	16	0.785	0.49	16

Results for circuits SUT_2 and SUT_3 are comparable, showing that with increasing the number of neurons, the size of the set L_1 (used to train SOM) also increases, leading to the more significant introductory classifier and decreasing importance of the detailed classifier. The larger number of neurons decreases the number of ambiguity groups. This is because there are more units specializing in the single example. The accuracy of the trained network (evaluated on the set T) is in all cases around 80 percent. The training time of the SOM (much below 1ms) is negligible compared to the detailed classifier.

The number of dead neurons taking part in the fault identification process also increases for larger networks. This is because the higher number of neurons is trained to respond to the specific examples. Therefore the chance that such a neuron will have the strongest response to the example from T is high.

Contents of the ambiguity groups in most cases are as follows: *{12, 42}*, *{11, 41}*, *{21, 51}*, *{22, -51}*, *{-31, -52}*, *{-11, -41}*, *{-12, -42}*. Additionally, there are AG related with the minor mistakes by the classifier: *{-21, -22}* and *{61, 62}*. Problems are expected during distinguishing between the resistor R_1 and capacitor C_1, and resistor R_2 and capacitor C_2. Analysis of the detailed classifier shows indeed these are the biggest problems for the RF.

B. Influence of the RF configuration one the diagnostic quality

The RF was used as the detailed classifier to identify faults represented by examples initially detected by neurons belonging to categories N_2 or N_3. Each time RF was trained on the set L_2, which content changes, depending on the SOM training. The main parameter here is the number of trees, influencing the diagnostic outcomes, especially when the voting is not unanimous. Example of the RF performance for the circuit SUT_2 is presented in Fig. 7. Here the acc (2) and acc_{det} measures are shown for the constant value of acc_{int}. Diagnostic quality of the RF is lower than of the SOM (see Table 1), because the set of examples to process is more difficult. In most cases the optimal number of trees (ensuring the highest accuracy with their smallest number) is between 11 and 13. The higher number gives no improvement, but increases t_{tr} and t_{te}. Considering minor mistakes of the diagnostic module (i.e. the proper location of the fault, but its incorrect identification) by giving 0.5 of the point for such a situation, leads to increasing the accuracy at about 1 percent for each classifier.

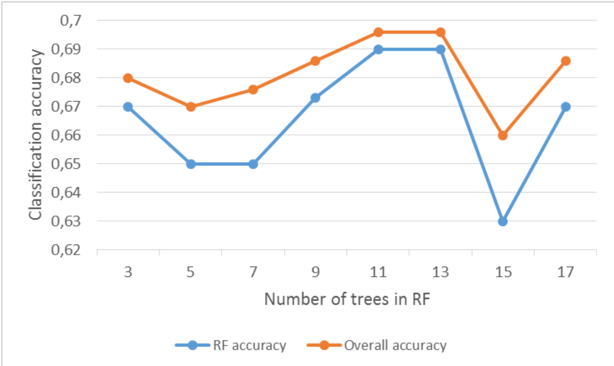

Fig. 7. RF performance for SUT_1 depending on the number of trees.

The time of training and testing the RF classifier depends on the number of trees and the size of the set L_2, which contains various number of examples and fault codes. In all cases the dependence between the number of trees and t_{tr} or t_{te} is linear. The overall operation time for the whole hierarchical module is mostly influenced by the RF, as the duration of SOM is 10 to 100 times shorter.

Fig. 8. Time efficiency of the RF for $|L_2|$=46.

C. Influence of SUT tolerances on the diagnostic accuracy

Comparative analysis for all three SUT models was conducted. It was determined that although there is the slight decrease in accuracy with increasing tolerances, it has the minor influence on the overall performance. Results are presented in Fig 9.

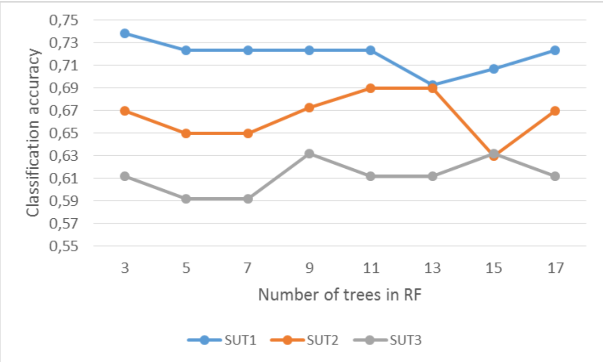

Fig. 9. Diagnostic accuracy for circuit models with increasing tolerances of parameters.

VII. CONCLUSIONS

The paper presented the novel hierarchical diagnostic module, tested against the electronic circuit with varying tolerances of parameters. Conducted experiments show effectiveness of the approach, combining the short operation time with high accuracy. Influence of parameter tolerances on the diagnostic process is limited, but to confirm conclusions generally, the scheme should be tried on other SUTs.

Future investigations would cover introducing other algorithms for both introductory and detailed classification stages and comparing their effectiveness. The additional problem is elimination of AG by introducing frequency analysis or increasing the number of symptoms, based on which the decision is made.

REFERENCES

[1] P. Jantos, D. Grzechca, T. Golonek, and J. Rutkowski, "The Influence of Global Parametric Faults on Analogue Electronic Circuits Time Domain Response Features," 11th IEEE Workshop on Design and Diagnostics of Electronic Circuits and Systems (DDECS), 16-18 April 2008, DOI: 10.1109/DDECS.2008.4538806.

[2] J. Verbrugghe, M. De Bock, and P. Rombouts, "Design of an integrated analog controller for a class-D audio amplifier," 16th IEEE International Conference on Electronics, Circuits, and Systems, ICECS 2009, 13-16 Dec. 2009, DOI: 10.1109/ICECS.2009.5410790.

[3] G.N. Stenbakken, T.M. Souders, and G.W. Stewart, 1989, "Ambiguity groups and testability," IEEE Trans. Instr. and Meas., Vol. 38, Issue 5, 1989, pp. 941-947.

[4] J.A. Starzyk, J. Pang, S. Manetti, M.C. Piccirilli, G. Fedi, "Finding ambiguity groups in low testability analog circuits," IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications, Vol. 47, Issue 8, 2000, pp. 1125-1137.

[5] R.G. Vieira, C.M.S. Medeiros, and E.T. Silva "Classification and sensitivity analysis to detect fault in induction motors using an MLP network," International Joint Conference on Neural Networks (IJCNN), 24-29 July 2016, DOI: 10.1109/IJCNN.2016.7727281.

[6] F. Ye, Z. Zhang, K. Chakrabarty, and X. Gu, "Board-Level Functional Fault Diagnosis Using Artificial Neural Networks, Support-Vector Machines, and Weighted-Majority Voting," IEEE Trans. on Computer-Aided Design of Int. Circ. and Sys., Vol.32, No.5, May 2013, pp. 723-736.

[7] P. Bilski, “Application of random forest to the fault detection in analog circuits,” Proc. XXI IMEKO World Congress, 30 Aug.-4 Sept. 2015, Prague, Czech Republic.

[8] P. Bilski, “Ambiguity groups detection in analog systems diagnostics using Self-Organizing Maps,” Proc. IMEKO TC10 Workshop, June 27-28 2016, Milan, Italy, pp. 294-299.

[9] N. Liu, J. Wang and Y. Gong, "Deep Self-Organizing Map for visual classification," International Joint Conference on Neural Networks (IJCNN), 12-17 July 2015, Killarney.

[10] J. M. Bossio, C. H. De Angelo, and G. R. Bossio, "Self-organizing map approach for classification of mechanical and rotor faults on induction motors," Neural Comput & Applic, No. 23, 2013, pp. 41–51.

[11] S. Gupta, D. Singh, K P Singh, and S. Kumar, "An efficient use of random forest technique for SAR data classification," IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 26-31 July 2015, DOI: 10.1109/IGARSS.2015.7326520.