XI IMEKO World Congress “Measurement in Research and Industry”

August 30 – September 4, 2015, Prague, Czech Republic

APPLICATION OF RANDOM FOREST TO THE FAULT DETECTION IN ANALOG CIRCUITS

Piotr Bilski [1,2]

[1] Institute of Radioelectronics, Warsaw University of Technology, Warsaw, Poland, pbilski@ire.pw.edu.pl
[2] Department of Applied Informatics, Warsaw University of Life Sciences, Warsaw, Poland, piotr_bilski@sggw.pl

Abstract – The paper presents the application of the Random Forest (RF) to the diagnostics of analog circuits. The modified structure of the forest proposed for the task and its generation algorithm is presented. The approach is used to detect parametric faults in the fifth order lowpass filter. Multiple versions of the forest are tested. The proposed method is also confronted against the decision tree, being superior over the traditional classifier. The future prospects of the approach are considered.

Keywords: decision tree, random forest, fault detection, analog circuits

1. INTRODUCTION

The modern diagnostics of analog systems is the task of rising importance, as the number of systems requiring monitoring increases. Analog circuits play an important role in the military, radiocommunications or audio applications. Detecting faults in their structure allows for modifying the design pattern to avoid similar failures in the future. Also, the accurate fault identification and location enables exchanging the faulty element with its nominal counterpart. This is the case in expensive systems, where repairing the circuit is more appropriate than acquiring the new one. The example of such a system is the audio tube-based amplifier.

Apart from the go-no go factory tests of newly manufactured items, diagnostics during their exploitation is equally important. Therefore the proper tests should be prepared for each system separately. Besides multiple numerical approaches, such as the sensitivity analysis [1] or dictionary method [2], analog circuits are often processed using the Artificial Intelligence (AI) approaches. The most popular are Artificial Neural Networks (ANN) [3] and Support Vector Machines (SVM) [4]. As new heuristic methods are constantly developed, their implementation for diagnostic purposes is justified.

The paper presents the application of the novel classifier for the diagnostics of analog circuit. The random forest and its modifications are presented and adjusted to the requirements of the fault detection procedure. Section 2 presents assumptions used during the diagnostic procedure implementation. The details of the classification method, the learning algorithm and proposed variants are in Section 3. The tested circuit is described in Section 4. The experimental results are presented and commented in Section 5, while Section 6 contains conclusions and future prospects for this classifier as the diagnostic tool.

2. DIAGNOSTIC PROCEDURE

The applications of AI methods to detect and identify faults are currently well established. In most cases, data-driven approaches are used, where the source of knowledge required for the classification or regression tasks is the training data set. It is constructed after extensive simulations of the Circuit Under Test (CUT) model. After inserting the fault to the system's structure, it is excited with the selected signal at the accessible (input) node. Responses are then recorded at selected partially accessible nodes. The information that allows for identifying the particular fault is extracted from them. This way the data set T_r (1) for the classification task (which is the topic of this paper) contains n examples $\boldsymbol{e_{li}}$, each being the vector of m symptoms (attributes) s_{ij} used during the fault analysis process to make decision about the CUT state. This information is supplemented with the discrete category of the fault $\boldsymbol{c_i}$, pointing at the faulty element (fault location) and (if necessary) its deviation from the nominal value (fault identification). Element tolerances must be considered as well, influencing the thresholds, beyond which the CUT is considered faulty. In the presented research the fault codes proposed in [5] were used. They indicate both the number of the faulty CUT parameter and deviation of its value from the nominal one. For instance, “11” means that the first parameter has value “large”, while “-31” is for the “small” value of the third parameter. The nominal states (here indicated by the category “0”) are also present in the data set. The assignment of the particular category to the example depends on the changes it causes in the recorded response compared to the nominal state. Some CUT elements may have the low sensitivity, so change of their value may be hardly visible on the output. In such a case, the system is considered nominal, as it behaves accordingly to the design.

$$T_r=\begin{bmatrix}\boldsymbol{e_{l1}}\\ \vdots\\ \boldsymbol{e_{ln}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{s_1} & \cdots & \boldsymbol{s_m} & C\end{bmatrix}=\begin{bmatrix}s_{11} & \cdots & s_{1m}\boldsymbol{c_1}\\ \vdots & \ddots & \vdots\ \vdots\\ s_{n1} & \cdots & s_{nm}\boldsymbol{c_n}\end{bmatrix} \quad (1)$$

During the data set preparation the selected examples should cover as wide range of the CUT states as possible. The number of examples is to be minimal, but containing all required information about the most probable faults Also, the selection of the excitation and the analysis domain (time, frequency, mixed, etc.) is important, because it affects the set of symptoms used for the analysis.

The knowledge extraction and its exploitation for the fault detection presented in Fig. 1 assumes that only one CUT element is beyond the tolerance margins at the particular time. This is the most common situation, facilitating the machine learning process. Although multiple faults are also possible, their analysis is difficult and usually requires checking various combinations of faults occurring at the same time. Therefore their analysis should be considered in the future.

Knowledge gathered during the training is next used to make decisions about the state of the actual CUT or its model. The important aspect of the intelligent method is generalization, i.e. the ability to correctly classify examples not present during the training process. This is verified using the testing data set T_t of the same form as the T_r, but containing different examples. This allows for checking the generalization ability of the proposed classifier, which is trained and tested on different data.

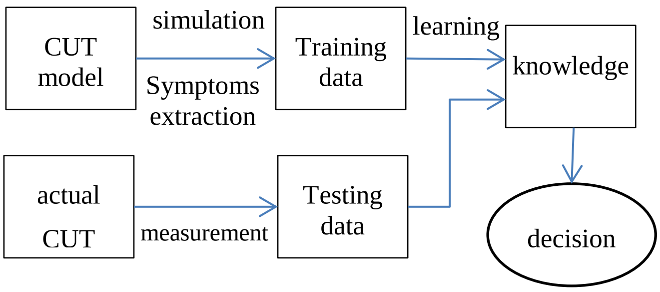

Fig. 1. Scheme of the AI-based training and testing procedure for the analog circuit diagnostics.

3. RANDOM FOREST

This is the novel classification method used with success in the analysis of large data sets in bioinformatics and environmental applications [6]. It was also used to detect faults in the aircraft system [7], but its applications in the diagnostics are seldom so far. The efficiency analysis of this approach shows its advantages over other methods, such as ANN or Naïve Bayes Classifier. On the other hand, it bears resemblance to the AdaBoost algorithm, which also was not yet exploited in the diagnostic domain. Therefore the performance of RF should be tested on as many objects as possible, considering typical challenges, such as ambiguity groups or varying sensitivity of CUT elements.

The RF [8] is the extension of the well-known Decision Tree (DT) approach, which has been extensively used for over twenty years [9]. Its advantages include the legible form of the stored knowledge (in the form of rules) and low memory requirements. The disadvantage is the difficulty in processing data affected by the measurement uncertainty and noise. Also, there is the threat of overfitting the DT to the training data, making it inefficient in the processing of testing examples.

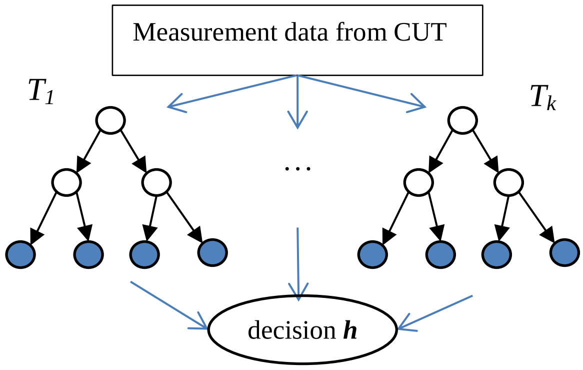

Fig. 2. Fault detection and identification using the random forest.

The random forest F (Fig. 2) overcomes these limitations, as it consists of multiple decision trees T_i (each potentially representing various knowledge extracted from the same training data set). They are obtained during the repeated decision tree induction algorithm, where for each created node in the tree T_i, the test is randomly selected from the predefined set of the best candidates. This way the result of each iteration is a little different tree constructed on the training data from CUT. During the testing phase, all trees are applied simultaneously for the example $\boldsymbol{e}_i$, making their own decisions (hypotheses) $h_k(\boldsymbol{e}_i)$. The overall diagnostic result $\boldsymbol{h}(\boldsymbol{e}_i)$ is obtained during the voting procedure. Although there are multiple voting strategies, during the simultaneous classifier implementation [10], in the presented research only the majority voting was applied, with each tree weighted equally. This way the forest becomes the ensemble of classifiers, similar in the operation regime to the fusion of SVM modules [11] or various statistical approaches (such as hidden Markov Models) [12].

The DT implemented for the forest F (Fig. 3) is the directed acyclic graph with the hierarchical structure of nodes (starting from the initial root at level 1) and ending with leaves, which contain the particular category. To make the simplest structure, each parent node is connected to two child nodes or leaves from the lower level. The edges connecting nodes are related to the result of the test stored in the node. It is the threshold value θ of the selected symptom s_i. The sequence of tests enables the tree to make the decision about the fault category. The example $\boldsymbol{e}_i$ (containing the symptoms extracted from the measured responses of the CUT) "travels" from the root to one of the leaves. In each visited node, the test is executed on $\boldsymbol{e}_i$. The comparison between the threshold θ and the value of the corresponding attribute s_j in the example causes its relocation to one of two lower nodes. The possible results are "$s_j \geq \theta$" or "$s_j < \theta$". This process is repeated until the example reaches the leaf, which points at the particular category.

During the single tree construction it is possible to obtain multiple candidates for the test with the minimum entropy. Although in the RF this is not a problem, as the threshold is selected randomly, in the single DT (used in Section 5 as the reference method) the method of selecting the test from equally good may influence the accuracy of the structure. The following options were considered during the experiments:

a) selection as the threshold the attribute with the largest distance from the neighbouring values

b) selection as the threshold the attribute with the smallest distance from the neighbouring values
c) selection as the threshold the attribute occurring in the tree nodes the greatest number of times
d) selection as the threshold the features occurring in the tree nodes the smallest number of times
e) random selection of the feature from the subset of the ones with the smallest entropy.

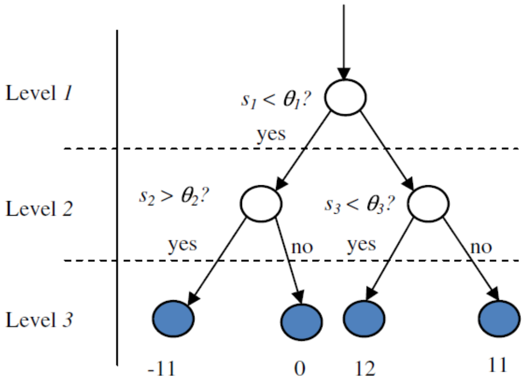

Fig. 3. Structure of the single decision tree.

Unlike distance-based methods (such as k Nearest Neighbours), the decision tree is resilient to the number of symptoms in the data set. Only the attributes required to distinguish between different categories are considered, which significantly limits the amount of processed data during the classification (although each tree may exploit the different set of symptoms from the original set).

The RF accuracy is measured as the sample error, i.e. the relative number of incorrectly classified examples from the testing set T_t. The evaluation is based on the comparison between the produced hypothesis $\boldsymbol{h(e_i)}$ by the RF and the actual category c_i of this example.

$$E_c = \frac{|e_i : h(e_i) \neq c_i|}{|e_i|} \cdot 100\% \qquad (2)$$

In the presented work the number of examples in T_r and T_e is equal. Because the amount of simulations executed to create the sets was relatively small (see Section 4), the cross-validation was not necessary. The experiments for both sets were selected to represent the most significant and often occurring faults in the system. Both the DT induction and the RF were implemented in the Matlab environment.

3.1. Forest induction procedure

The generation of the trees to the RF is the iterative procedure, during which a single tree is constructed at a time. The recurrent induction algorithm is used for this purpose [11]. In each step the node with the threshold is created. It allows for division of the currently processed set into two subsets of the similar cardinality, so that each category should be present only in one subset. This way to completely separate examples based on their categories, the minimum set of nodes is required, leading to the simplest tree possible. If the subset contains only examples belonging to one category, further division is not required. Instead, the leaf is created, representing this category. It is possible that examples belonging to different fault categories have similar (but not identical) values for all symptoms. The DT will be able to distinguish them during the training, but this may lead to the overlearning and the poor generalization abilities. All examples from the set T_r will be separated correctly, but the performance (2) on the set T_t will be low. This disadvantage should be minimized in the forest, as multiple trees process the same example, focusing on different symptoms and their values.

The entropy is used to select the best candidates for the threshold. This is the measure of disorder in data, with high value for the threshold θ creating two subsets, which contain the same categories. The low entropy value is for the thresholds dividing examples into two subsets with mutually exclusive categories. This outcome is preferred, as it quickly partitions data into separate fault identifiers. Candidates for thresholds are calculated as the middle points between neighbouring values of sorted symptom values. Next, one of them belonging to the group of candidates with the smallest entropy is randomly selected.

3.2. Forest parameters

The efficiency of the random forest depends on the following parameters: the number of trees k working in parallel and the number of candidates t for the threshold during the node selection in the tree (from which the test is selected). Adding more trees to the ensemble increases the classification accuracy to some extent, but also decreases the speed of training and decision making. The minimal value k of trees should be determined for the forest F ensuring its maximum efficiency in the shortest duration (see Section 5). Because during the generation of each node multiple candidates with the minimum entropy value are present, the number of considered candidates influences the variability between the trees, as they should be as diverse as possible. The original induction algorithm assumes the fixed number of candidates for each node during the forest generation. The modification proposed here (and discussed in the full paper) focuses on all candidates with the minimum entropy, as this number changes for every node. Therefore t should be interpreted here as the maximal number of the considered test candidates.

4. CIRCUIT UNDER TEST

The analog lowpass 5th order Chebyshev filter (Fig. 4) was used as the test circuit, with the frequency characteristics as in Fig. 5. The analysed elements were values of the capacitors and resistors. Their nominal values were as follows: $R_1 = R_2 = R_3 = R_4 = R_5 = 1k\Omega$, $C_1=16nF$, $C_2=19nF$, $C_3=13nF$, $C_4=51nF$, $C_5=49nF$. The circuit model was implemented in the Simulink environment.

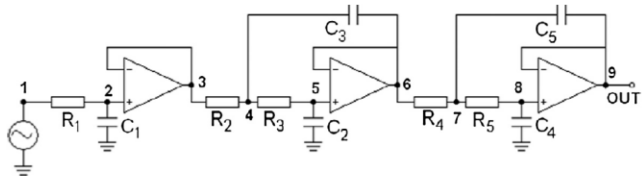

Fig. 4. Structure of the analyzed 5th order lowpass filter.

The simulation of the filter was performed in the time domain with the first maximum and minimum values, and their time stamps with the time stamp of the zero-crossing as the symptoms, recorded from the signals at the nodes 3, 5, 6, 8 and 9. The CUT was excited with the sinusoid of the frequency $f=8kHz$, which should allow for detecting all changes in the system if the drifting values of elements influence the frequency characteristics. The parametric faults were considered during the tests. The topology of the CUT remained intact, but the value of each element was changed in the range of $\pm 0.1 \cdot p_n$, where p_n is its nominal value. The random forest was trained and tested on the sets containing 70 examples, each with 54 symptoms.

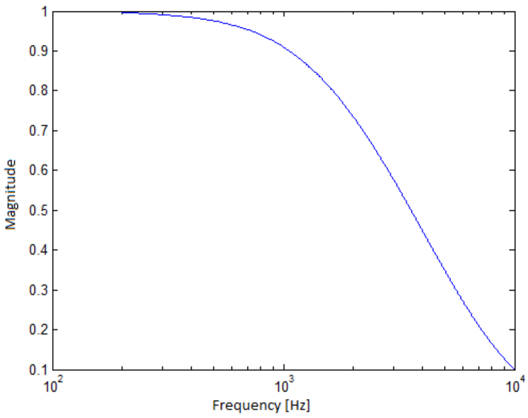

Fig. 5. Frequency passband of the analysed filter.

The conducted experiments included verification of the following features influence on the diagnostic efficiency:

- the random forest parameters (Section 3.2)
- the tolerances of CUT elements' values, determining the ability to distinguish between the nominal and faulty states. In this research the tolerances were defined as values in the range of $\pm 0.9 \cdot p_n$
- the amount of training data, represented by both the set of symptoms (Section 3.2) and the number of examples in the set.

5. EXPERIMENTAL RESULTS

The experiments were conducted according to the assumptions presented in Section 4. First, the parameters of the RF were tested. Fig. 6 presents the mean values of accuracies for the forests consisting of various number of threes (each curve is for the different value of *t*). To avoid the situation when two equally numerous groups of trees support different categories, the number of trees *k* in the forest was always determined as odd. Because the every tree in the RF is different, repeating the generation procedure from Section 3.1 twice will result in two different forests. Therefore to determine the general performance of the particular RF configuration, each creation procedure was repeated twenty times. The standard deviation of the process never exceeded 3.5 percent for each case. The presented accuracy is measured as $1\text{-}E_c$ - see (2).

The optimal number of trees in RF ensuring the maximal accuracy is between 21 and 31, depending on *t*. It was determined depending on the number of available symptoms (here 54). For instance, "1/6" means the number of the best candidates to consider is one sixth of *m*, i.e. *54/6=9*.

For all values of *t* the tendency is identical: the accuracy quickly increases with the adding more trees up to 11 or 13. Above this threshold, the increase may be observed, but is less significant. As the RF generation is performed offline, the process duration is less significant and more trees may be created. The improvement, however, will be low. For instance, for *t=1/3* (*18* thresholds candidates considered) the acceptable mean accuracy (above *78* %) is obtained for *k=9* trees. Expanding the forest to 31 tress leads to the *80%* accuracy. Decision about the number of trees to generate is made regarding the available time and desired RF quality.

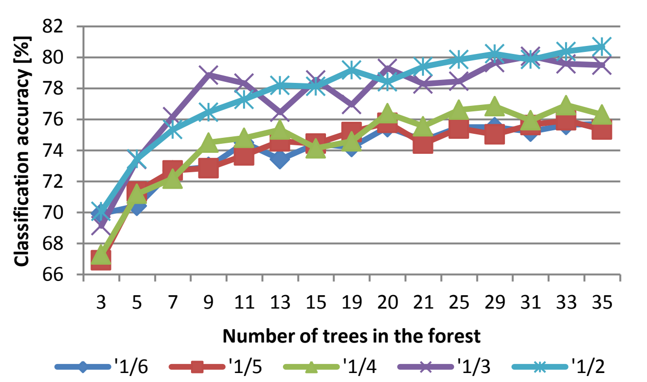

Fig. 6. Accuracy of the RF with different number of trees (for varying *t*).

The influence of the *t* parameter on the accuracy of the RF shows that increasing the number of threshold candidates leads to the accuracy maximization. The best performance was obtained for *t=1/2*, i.e. *26* values considered each time the node is created. Smaller values, such as *1/6*, *.1/5* and *1/4* give similar results for all sizes of the RF. This is because the higher number of symptoms to choose from allows for the greater variability between the trees.

Table 1. Examples of the diagnostic decisions by the RF for selected examples (*k=11* trees in the RF).

Actual category	Voting distribution	Diagnostic outcome
-12	-62 (4), -12 (7)	-12
11	0 (1), 11 (3), 21 (2), 51 (1), 61 (4)	**61**
-21	-82 (1), 81 (1),-72 (1), -22 (2), -21 (6)	-21
0	0 (11)	0
-32	-32 (8), -22 (3)	-32
41	41 (11)	41
-51	-91 (2), -52 (4), -51 (5)	-51
-72	-92 (1), -72 (2), -52 (2), -42 (2), -41 (2), -32 (1), -31 (1)	-72

The detailed analysis of the decision making process is illustrated in Table 1 (for the RF consisting of 11 trees). In the "Voting distribution" column the list of considered categories with the number of supporting trees in brackets is presented. Four results are possible here:

- unanimous voting for the proper fault identifier by all trees. This is the case for the category "0" and "41", leading to the correct diagnostic decision.

- voting split between two or three categories, with one being supported by significantly greater number of trees, as in the identifier "-12" or "-32", also leading to the correct decision.
- multiple categories considered with small number of trees supporting each fault code and one dominant. Such is the case of the identifier "-21" or "11". The first example is classified correctly, while the second gives the incorrect answer (the bold font).
- The same situation as above, but without the dominant category, which is the case for the example labelled as "-72". Despite the odd number of trees, the greatest number of votes (i.e. 2) is obtained for four fault codes, among which one is selected using any method (such as the random selection).

The time required to train and use the RF for the classification is presented in Fig. 7. The dependency between the duration of training and testing is linear. The off-line operation (RF construction) is significantly longer, but can be implemented in the general purpose computer system. The on-line diagnostics (RF decision making) is relatively short, therefore it may be considered to be implemented in the embedded system. The presented results were obtained for the sequential generation and classification of each tree. It is possible to accelerate the operation using the multi-core systems.

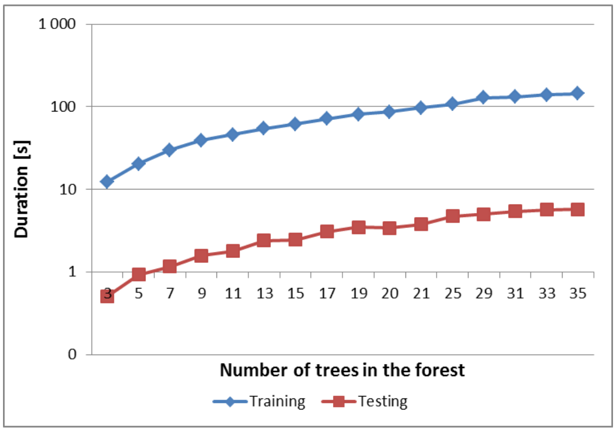

Fig. 7. Duration of training and testing the RF depending on the number of trees.

The comparison between the RF and various DT versions is in Table 2. All tree configurations described in Section 3 were implemented to classify the data from CUT. They were confronted against the optimal configuration of RF. Results prove the usefulness of the latter, as it is significantly better even than the best DT configuration.

Table 2. Comparison between the efficiency [%] of various DT configurations and RF.

DT (a)	DT (b)	DT (c)	DT (d)	DT (e)	RF
58,57	58,57	67,14	60	66,5	80,67

6. CONCLUSIONS

The RF is the promising classification method with high efficiency even on the relatively small data sets. It is resilient to the measurement uncertainty and requires small computational power. However, to verify its practical superiority over other approaches, the appropriate comparison should be made, applying ANN, SVM or other methods to the same CUT. Some challenge may be the selection of forest parameters ensuring the highest diagnostic accuracy. The conducted experiments show that it is relatively easy to obtain the acceptable configuration.

The forest is also effective compared to other ensembles of classifiers. Even the simplest voting mechanism is enough to obtain the good classification outcome, as the greatest influence on the obtained results lies in the number of trees.

Although the analysed system was the electronic circuit, the proposed scheme is generic enough to make it applicable for other objects, such as electrical machines, wind turbines, vehicular engines, etc. Introduction of other systems would help to provide the general suggestions about selecting the optimal parameters of the forest.

REFERENCES

[1] N. Ben Hamida and B. Kaminska, "Multiple fault analog circuit testing by sensitivity analysis," *Journal of Electronic Testing*, vol. 4, nº. 4, pp. 331-343, 1993.

[2] J. Starzyk, D. Liu, Z. Liu, D. Nelson and J. Rutkowski, "Entropy-based optimum test points selection for analog fault dictionary techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 53 nº. 3, pp. 754-761, 2004.

[3] S. G. Mosin, "A Neural Network Approach to Functional Test of Analogue Circuits," *MIXDES 2005 Conference*, pp. 391-396, Kraków, Poland, June 22-25, 2005.

[4] R. Sałat and S. Osowski, "Support Vector Machine for soft fault location in electrical circuits," *Journal of Intelligent and Fuzzy Systems*, vol. 22, nº. 1 pp. 21-31, 2011.

[5] P. Bilski and J. Wojciechowski, "Automated Diagnostics of Analog Systems Using Fuzzy Logic Approach", *IEEE Transactions on Instrumentation and Measurement*, vol. 56, nº. 6, pp. 2175-2185, December 2007.

[6] A.-L. Boulesteix, S. Janitza, J. Kruppa and I.R. König, "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 2, nº. 6, pp. 493–507, November/December 2012.

[7] S. Lee, W. Park and S. Jung, "Fault Detection of Aircraft System with Random Forest Algorithm and Similarity Measure," *The Scientific World Journal*, vol. 2014, http://dx.doi.org/10.1155/2014/727359.

[8] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[9] A. Jamehbozorg and S.M. Shahrtash, "A Decision-Tree-Based Method for Fault Classification in Single-Circuit Transmission Lines," *IEEE Transactions on Power Delivery*, vol. 25, nº. 4, pp. 2190-2196, October 2010.

[10] L. Kuncheva, "Combining Pattern Classifiers. Methods and Algorithms," Wiley and Sons, New Jersey, 2004.

[11] L. Batista, B. Badri, R. Sabourin, M. Thomas, "A classifier fusion system for bearing fault diagnosis," *Expert Systems with Applications*, vol. 40, pp. 6788–6797, 2013.

[12] K. Choi, S. Singh, A. Kodali, K.R. Pattipati, J.W. Sheppard, S. Madhavi Namburu, S. Chigusa, D.V. Prokhorov, and L. Qiao "Novel Classifier Fusion Approaches for Fault Diagnosis in Automotive Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, nº. 3, pp. 602-611, March 2009.