# A Classification Approach for an Accurate Analog/RF BIST Evaluation Based on the Process Parameters

Ahcène Bounceur, Samia Djemai, Belkacem Brahmi, Mohand Ouamer Bibi, Reinhardt Euler

**Noname manuscript No.**
(will be inserted by the editor)

# A Classification Approach for an Accurate Analog/RF BIST Evaluation Based on the Process Parameters

**Ahcène Bounceur · Samia Djemai · Belkacem Brahmi · Mohand Ouamer Bibi · Reinhardt Euler**

Received : date / Accepted : date

**Résumé** Specifications of Radio Frequency (RF) analog integrated circuits have increased strictly as their applications tend to be more complicated and high test cost demanding. This makes them very expensive due to an increased test time and to the use of sophisticated test equipment. Alternative test measures, extracted by means of Built-In Self Test (BIST) techniques, are useful approaches to replace standard specification-based tests. One way to evaluate the efficiency of the CUT measures at the design stage is by estimating the Test Escapes ($T_E$) and the Yield Loss ($Y_L$) at ppm level. Unfortunately, an important number of Monte Carlo simulations must be run in order to guarantee their accuracy. For certain types of circuits, this requires many months or even years to generate millions of circuits. To overcome this limitation, we present in this paper a new technique where a small number of simulations is sufficient to reach an important precision. This method is based on a classification using machine learning methods, such as SVM and Neural Networks based classifiers to determine pass/fail regions. The proposed approach requires a few number of simulations only to determine the region separating the process parameters generating good and faulty, or pass and fail circuits. Then only this region is needed to estimate the test metrics without running any additional simulation. The proposed methodology is illustrated for the evaluation of a filter BIST technique.

**Keywords** Analog/RF test · SVM classification · Neural Network classification · BIST evaluation

## 1 Introduction

The test is a very important notion in the field of the validation step of analog circuit design. It is the way to confirm whether a circuit under test (CUT) satisfies all functional specifications that represent its performance parameters. Since there are no manufactured circuits at the design stage, test errors can only be

A. Bounceur
Lab-STICC Laboratory, University of Brest, France
E-mail: Ahcene.Bounceur@univ-brest.fr

S. Djemai
LaMOS Research Unit, Department of Mathematics, University of Jijel, Algeria
E-mail: samia_djemai@hotmail.fr

B. Brahmi
LaMOS Research Unit, Department of Operational Research, University of Bejaia, Algeria
E-mail: bra_belka@yahoo.fr

M.O. Bibi
LaMOS Research Unit, Department of Operational Research, University of Bejaia, Algeria
E-mail: mobibi.dz@gmail.com

R. Euler
Lab-STICC Laboratory, University of Brest, France
E-mail: reinhardt.euler@orange.fr

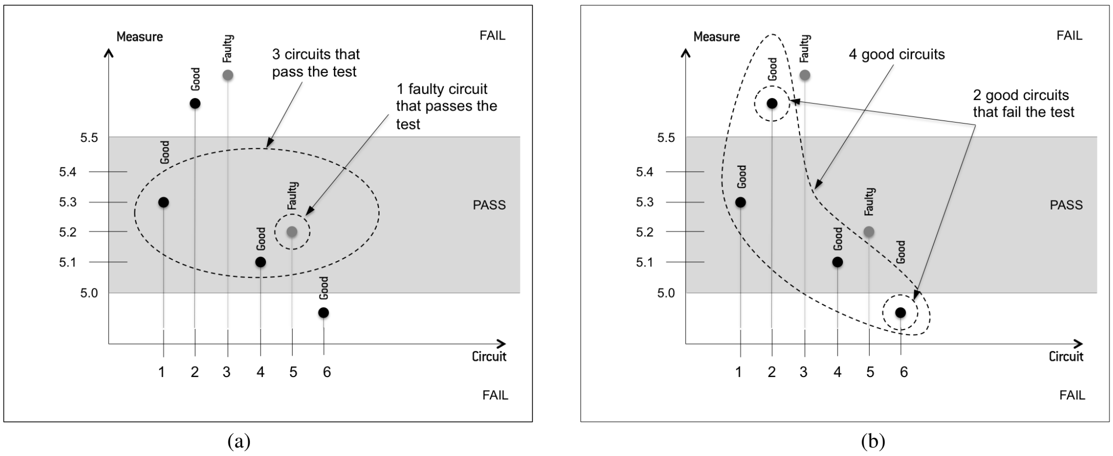
validated by simulation. Monte Carlo circuit simulation allows the generation of a relatively small sample of circuit instances under process variations. Each circuit instance is represented by a vector of its output parameters. A functional circuit is the one for which all performances meet the specifications. A circuit is faulty if at least one performance does not meet its specifications. Similarly, a circuit passes the test if all its test measures are within their pre-defined limits. Otherwise, the circuit fails the test. A test error occurs when either a faulty circuit escapes the test or a functional circuit fails the test. These two cases lead to Test Escapes ($T_E$) representing the probability of a faulty circuit to pass the test, and to Yield Loss ($Y_L$) representing the probability of a functional circuit to fail the test, respectively. The authors of [1] have presented brief details on the definition and the computation of these test metrics.

Given a circuit with $n$ performances $P = (P_1, P_2, ..., P_n)$ and $n$ specifications $s = (s_1, s_2, ..., s_n)$, this circuit has also a set of $m$ test measures $T = (T_1, T_2, ..., T_m)$ and $m$ test limits $l = (l_1, l_2, ..., l_m)$. Generally, we assume that a functional (or good) circuit satisfies the specification $P \leq s$ and a circuit passes the test if $T \leq l$. Formally,

$$T_E = Pr(P > s | T \leq l). \tag{1}$$

$$Y_L = Pr(T > l | P \leq s). \tag{2}$$

As an example, let us consider the case of 6 circuits having one performance and one test measure as shown by Figure 1. The black points show the good circuits that satisfy their specification. The gray ones represent the faulty circuits that do not satisfy their specifications. The circuits that are inside the gray zone are those that pass the test, and those that are outside are those that fail the test. The estimation of the Test Escape $\hat{T}_E$ is given by the proportion of the faulty circuits that pass the test (the gray points that are in the gray zone) with respect to the total number of circuits that pass the test (the points that are in the gray zone). Thus, $\hat{T}_E = \frac{1}{3}$. The estimation of the Yield Loss $\hat{Y}_L$ is given by the proportion of the good circuits that fail the test (the black points that are in the white zone) with respect to the total number of the good circuits (the black points). Thus, $\hat{Y}_L = \frac{2}{4}$.


Measure
FAIL
Faulty
Good
3 circuits that pass the test
1 faulty circuit that passes the test
5.5
5.4
5.3
5.2
5.1
5.0
PASS
Circuit
1 2 3 4 5 6
4 good circuits
2 good circuits that fail the test

(a) (b)

**FIGURE 1** Example of calculating test metrics.

The evaluation of these test metrics with high precision, at ppm (parts per million) level, can be done just with a very large sample of circuits. Unfortunately, Monte Carlo circuit simulation is time-consuming and can only generate a relatively small sample in reasonable time. Indeed, the number of performance parameters of a circuit to be verified has increased too much over the past few decades. Therefore, the verification of all specifications has become a costly operation, in particular, due to lengthy test times and expensive test equipment. So there is a requirement to develop alternative test approaches for classifying the circuits into pass and fail categories without performing their intended specifications.

Some creative binary classification techniques, such as Support Vector Machines (SVM) and Artificial Neural Networks (ANN) have been successfully applied to fault diagnosis for analog circuits [2].

SVM-based binary classifiers are powerful techniques of statistical learning [3]. They can be applied effectively to learning problems with small sample size, high dimension, and non-linearity. Especially in a high dimensional data space, the support vector machine has a good generalization ability. It can successfully solve problems of overfitting, local optimality, and low-convergence rate existing in ANNs. SVM has been used for the statistical test. However, it is challenging to select appropriate SVM parameters. Indeed, the selection of SVM parameters has a significant influence on the classification accuracy of SVM.

Artificial Neural Networks [4] are used in all kinds of applications thanks to their benefits such as large scale parallel processing, distributed storage and nonlinear mapping. Recently, ANNs have been used to construct a diagnostic model for large-scale analog circuit test. This classification technique organizes simple functions into a network of nodes, where each node is called a perceptron. A network of perceptrons results from a complex nonlinear classification function. This method, however, suffers from a large amount of required training and a consequent increase in test application time, so the scheme is prohibitive for even medium sized analog circuits at production.

In this work, we propose a new method based on the classification of the process parameters of analog/RF circuits in order to evaluate their test technique. This method uses machine learning approaches to minimize the test cost while maintaining product quality and limiting yield loss. We use binary classifier based specifications and test limits to determine a good/faulty and a pass/fail circuit model. This latter allows more precision in the detection of the CUT state and then a high test accuracy. The main issue that we have to deal with is that a binary classification method works only when there are two families of circuits. However, in our case, a very small number of Monte Carlo simulations can be run and only one family of circuits is generated. To address this issue, we propose to first generate circuits using Monte Carlo circuit simulation by forcing the generation of two families of circuits. This can be done by modifying the distributions of the initial process parameters. Note, that the modification of the distribution will be used only to determine the classifier and not to estimate the test metrics. Once the classifier is determined, we propose to generate only the process parameters without running any simulation. Based on the found classifier, it is possible to predict the nature of the circuits, and then to estimate the test metrics. Since there is no simulation to run, it is possible to generate a huge number of circuits in a few seconds, which allows to reach a huge accuracy in calculating the test metrics. In this paper, we have used three classification methods that are : two SVM methods, where the first is the classical one implemented in the R software, and the second is an improvement of the first that we have presented in [41]. The third classification method is an Artificial Neural Network (ANN) method. These methods are suitable to solve nonlinear classification problems and they have been compared to the existing Statistical Blockade method presented in [31].

The rest of the paper is organized as follows : Section 2 presents an overview of the related literature. Algorithms for classification are described in Section 3, while Section 4 presents the principles of the proposed methodology. Simulation results are given in Section 5 and we finally conclude in Section 6.

## 2 Prior Work

Many approaches which optimize the cost of analog/RF test have been published in the literature. These methods differ in the technique used to represent the pass/fail regions with a low-cost RF test.

Different approaches have been proposed to test parametric faults in analog circuits. They include classification methods that distinguish between functional and faulty circuits by building a discriminating border in the space of measurements. A prediction model based on a binary classifier using SVM is used in [5], [6], [7], [8] and in [9] to eliminate costly RF tests. Artificial Neural Networks are handled in [10], [11] and [12] and have been successfully applied to fault diagnosis for an analog circuit. In [13], binary decision trees are used to eliminate redundant tests using binary classifiers in order to compact the complete test set. The method presented in [14] consists of a test strategy based on validating CUT specifications and using octrees.

Fault dictionary methods are proposed in [15], [16], [17] and [18], which model analog components and obtain the corresponding fault responses.

Along the same lines, some methods are based on performance ordering [19] and start with the test of performances having a high probability to detect faulty circuits. These methods adopt die-level statistical models to approximate the original test set or to predict pass/fail labels from a reduced or alternate low-cost set of measurements. Authors of [20]and [21] have introduced an estimation of test yield and the optimization of fault coverage, to reduce the number of tests. In [22], a similar approach including the cost of test has been proposed.

In the same purpose, some approaches have introduced the specification test compaction as described in [23] and [24] that leverages the correlation among specification tests in order to perform only a few of these tests during production and predict the values of the omitted ones to reduce test cost as well as the use of BIST techniques [25]. On the other hand, several methods based on the evaluation of test metrics are proposed in [26] and [27]. Most of them are based on the estimation of the distribution of the output parameters using a small sample of circuits generated via Monte Carlo simulation. Next, by sampling this statistical model, it is possible to numerically generate an arbitrarily large sample of circuits that follow the same distribution. This large sample will contain a representative set of defective parametric devices, and test metrics can be calculated using relative frequencies. In [28], the statistical model is assumed to be a multivariate Gaussian and in [29], a general parametric method is proposed using the theory of Copulas. A copula is a multivariate distribution that models the dependence structure between the output parameters, independently of the marginal laws of the parameters. However, a copula is not easy to identify. When a copula is not known, [30] proposes the use of non-parametric density estimations. The major problem with these techniques is that the estimated PDF of the output parameters is less accurate at the tails of the distribution, where the defective circuits, necessary for estimating test errors, are found. This is because in the initial sample of Monte Carlo circuits used for PDF estimation, there are in general no circuits with output parameters at the tails of the distribution. Thus, the accuracy at the ppm level of the test errors may be questionable.

Recently, the paper [31] has presented a technique that uses statistical learning to accelerate Monte Carlo simulation. First, a multidimensional classifier is used to learn a boundary of the input parameter space of the circuit under test (CUT) for which the output parameters are in the bulk of the distribution. Next, during Monte Carlo simulation, the simulation of an instance is blocked if the classifier predicts that the output parameters will fall in the bulk of the distribution. On the other hand, those circuits for which the classifier predicts that the output parameters will fall beyond the boundary, are simulated. The work [31] has considered extreme circuits generated this way and Univariate Extreme Value Theory for the estimation of memory yield. Later, a similar technique has been considered in [32] to estimate test metrics in a univariate case (a single output performance). The work [33] shows how extreme circuits can be used for an accurate computation of test metrics within a multivariate case study, assuming that the statistical learning technique proposed in [31] can be applied. However, the theory used is very complicated and not easy to apply, especially for circuits having an important number of performances and test measures.

In [2], a method is presented that improves the results obtained by the Statistical Blockade proposed in [31]. In this paper, we propose to use three machine learning methods such as SVM and ANN classifiers based on the specifications and the test limits instead of the extreme thresholds to determine pass/fail regions.

## 3 Classification Algorithms used in this study

### 3.1 SVM classification

Support Vector Machines (SVM) [3] represent one of the best binary classifiers with a large variety of applications. They are based on the principle of separation between the instances of two classes, positive and negative. Existing methods mainly try to optimize two phases : a training phase and a classification phase.

The first phase constructs the hyperplane, and the second uses it. The evaluation of the methods is based on the evaluation of the performances of the two phases. In addition, SVM is an interesting model whose geometrical interpretation is clear and easy to understand. It is relatively insensitive to the size of the dataset, and the classification complexity does not depend on the dimension of the feature space.

Training an SVM classification reduces to a convex quadratic minimization problem with one linear equality constraint and a number of box constraints.
The idea of SVM is based on the notions of margin maximization and kernel functions to find a separating hyperplane. The binary SVM solves the problem of separation into two classes, represented by a set of training examples $(x_i, y_i)$, $i \in I = \{1, \cdots, n\}$, where $y_i \in \{-1, +1\}$ is the class of the example $x_i \in \mathbb{R}^p$, $n$ is the size of the sample and $p$ is the number of features of each data point. It produces a hyperplane that splits the input sample into two categories.
Formally, for a binary classification problem, we consider the following optimization problem :

$$\min_{\alpha} L(\alpha) = \frac{1}{2}\alpha' D\alpha - e'\alpha, \tag{3a}$$

$$y'\alpha = 0, \tag{3b}$$

$$0 \leq \alpha \leq Ce, \tag{3c}$$

where $\alpha \in \mathbb{R}^n$, $e$ is an $n$-vector of ones and $y$ is an $n$-vector with $y = y(I) = (y_i, i \in I)$. The matrix $D$, defined by its elements $d_{ij} = y_i y_j k(x_i, x_j)$ is square of order $n$, symmetric and positive semidefinite, with $k$ being the kernel function. We deduce that the coefficients $\alpha_i$ associated with the examples $x_i$ of the training set are only the nonzero ones which are on the margin ($\alpha_i = C$) or inside the space defined by the margin ($0 < \alpha_i < C$). These examples are called *support vectors* and their corresponding index set is

$$I_{sv} = \{i \in I : \ \ \alpha_i > 0\}.$$

Once the dual optimal solution $\hat{\alpha}$ obtained, the rule for classifying a new observation $x$, based on a maximum margin hyperplane, is given by :

$$h(x) = sign\left(\sum_{i \in I_{sv}} \hat{\alpha}_i y_i k(x, x_i) + b\right). \tag{4}$$

The values of $C$ and kernel parameters significantly affect the classification accuracy of SVM. At present, the most common optimization technique is Cross Validation.

### 3.2 Neural Network classification

An Artificial Neural Network (ANN) [4] is a computational model based on the structure and the functions of biological neural networks. It consists of three interconnected layers : the input layer composed of the inputs $(x_i)$, the output layer of the output units $(y_j)$ and the hidden layer that interconnects units with the input and output layers. Each connection between units is correlated to a weight $(w_{ji})$, and to an associated additional weight $b_j$ which is called the bias. After the ANN has been trained with a sample of data, it can predict the classification for new data in the output layer. At present, neural networks have been extensively applied to data classification [35]. The general structure of an artificial neuron is shown in Figure 2.

ANN classifiers can be defined by :

— Input vector :

$$X = (x_i \in \mathbb{R}^p, i = 1, 2, ..., n) \in R^{p*n}, \tag{5}$$

— Connection weights :

$$W = (w_{ji}, j = 1, 2, ..., m,\ i = 1, 2, ..., n) \in R^{m*n}, \tag{6}$$

— Output vectors :

$$Y = (y_1, y_2, ..., y_m) \in R^m, \tag{7}$$

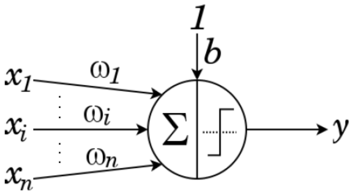


**FIGURE 2** Structure of an artificial neuron.

— The classification :

$$y_j = f(w_{ji}x_i + b_j), \tag{8}$$

where $w_{ji}$ is the weight from the input $x_i$ to the output $y_j$, and $f$ represents the activation function which determines how a neuron will scale its response to incoming signals and produce an activation. The activation functions are generally threshold-logic, hard-limit, continuous functions (sigmoidal) nodes, softmax, and radial basis functions. In our case, we used softmax functions which are defined by the following formulas :

$$y_k = \frac{e^{a_k}}{\sum_{l=1}^{k} e^{a_l}}, a_k = \sum_{i=0}^{n} w_{ki}x_i, \tag{9}$$

where $a_k$ is called the activation value of the output $k$.

To train an ANN classifier, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network must estimate the error derivative of the weights $E(W)$. In other words, it must describe how the error changes as soon as a weight is increased or decreased slightly. The back-propagation algorithm is the most widely used method for determining $E(W)$ given by

$$E(W) = \sum_{j=1}^{m}(y_j - t_j)^2, \tag{10}$$

where $t_j$ denotes the $j^{th}$ desired response of the $j^{th}$ output neuron, and $y_j$ represents the actual output of the $j^{th}$ output unit.

It is possible to use the gradient descent of $E(W)$ to update the weights at an iteration :

$$W^{T+1} = W^T - \eta\frac{\partial E(W^T)}{\partial W}, \tag{11}$$

where $W$ represents the matrix of neural weights, $T = 1, 2, ...$, denotes the iteration index during the training procedure, and $\eta$ denotes the learning rate specifying how much the parameters should be adjusted.

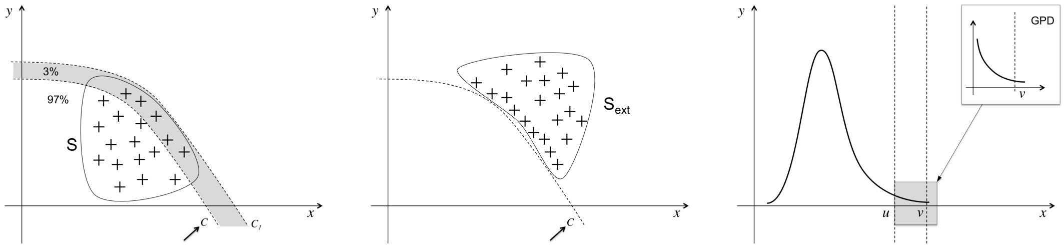
### 3.3 Statistical Blockade

The Statistical Blockade method uses the Extreme Value Theory (EVT) to estimate the test metrics. The EVT allows to estimate the probability of rare events by generating samples only in the part where the rare events are located. It is then possible to estimate any probability as if extreme values were generated. This is possible because the distribution of the extreme values can be estimated using an analytical model.

To explain the concept of an EVT, let us consider the distribution of Figure 3 (c) where we want to estimate the probability $P(x > v)$. Let us assume that this distribution is estimated from a set of samples generated after transforming (or simulating) another set of samples as shown by Figure 3 (a). We call this set the parameter set. The classifier $C_2$ in this figure corresponds to the limit $v$ of Figure 3 (c). As we can see, there is no generated samples having an $x > v$. Thus, an accurate calculation of the probability $P(x > v)$ is not guaranteed. To overcome this issue, another limit $u$ is fixed so that $3\%$ of the samples will have an $x > u$. The classifier corresponding to this limit in the parameter set is $C$. Then only samples that are in the right side of the classifier $C$ will be generated, as shown by Figure 3 (b). The transformation or the simulation of these samples will allow to generate samples that have an $x$ greater than $u$. According to the EVT, these samples will follow a Generalized Pareto distribution (GPD). It is then possible to estimate the probability $P(x > v|x > u)$, which is the probability that $x$ is greater than $v$ in the space of extreme values. The calculation of this probability will allow to calculate accurately the probability $P(x > v)$ as follows :

$$\begin{aligned} P(x > v) &= P(x > v \wedge x > u) \\ &= P(x > v|x > u) \times P(x > u) \end{aligned}$$

where, $P(x > v|x > u)$ follows the GPD distribution and $P(x > u)$ is already fixed to $3\%$.



(a) (b) (c)

**FIGURE 3** Statistical Blockade concept.

## 4 Proposed Methodology

### 4.1 The problematic and the proposed solution

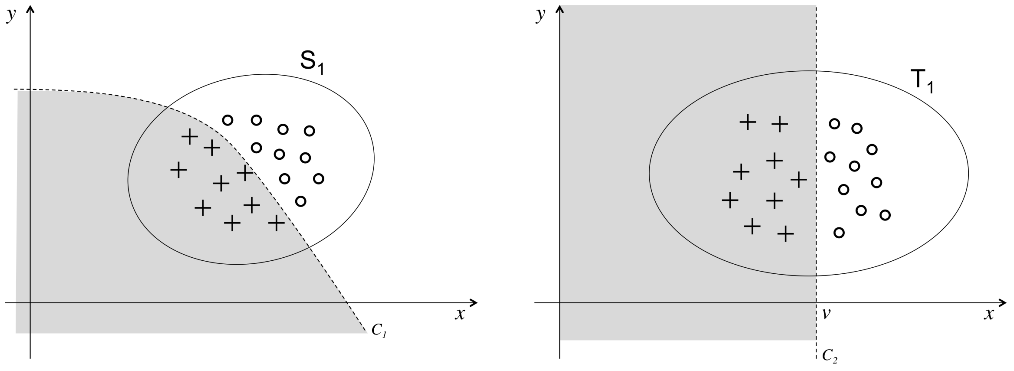
Before presenting the concept used in this paper, let us first present the problematic in a theoretical way. For simplicity, we will consider a two-dimensional space.

First, we start from a given set of points $S_1$ having a certain distribution, as presented by Figure 4 (a). Assume that these points have been transformed by means of a given function or by simulation. Figure 4 (b) shows an example of the obtained transformation. Now, assume that we want to classify these points into two families. The family of points having an $x < v$ (cross points in the gray part of Figure 4 (b)) and the family of points having an $x > v$ (circular points in the white part of Figure 4 (b)). These two families correspond to two families in the set of the non-transformed set of Figure 4 (a) (gray and white part). In this figure, the classifier $C_1$ separating the two families of points corresponds to the line $C_2$ separating the two families in Figure 4 (b).

In case of this paper, we don't know the analytical form of the transformation because we are using simulation. Thus, the analytical equation of the classifier $C_1$ cannot be found directly from the classifier or the separator $C_1$. A way to find its analytical form is to use classification methods. It consists in finding the curve that separates the cross points from the circular ones. However, the main problematic in the real case is that the initial set of points does not contain the circular points, as shown by Figure 5 (a). Thus, it is not possible to use classification methods.

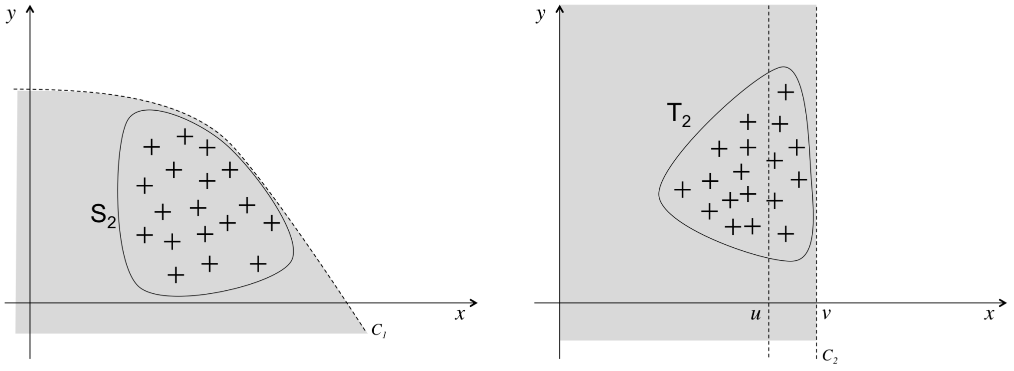
It is however possible to use the Statistical Blockade concept [31], which tries to estimate the distribution of the extreme points that have an $x > u$, where $u < v$. The value of $u$ is chosen so that the points follow the Extreme Value distribution. However, the main drawback of this method is double. The first one is that it can be used in a simple way only in the case of a two-dimensional space and its generalization to the case of an n-dimensional space is very complex. The second one is that a good estimation requires an important number of points. In this work, we try to estimate the test metrics with high accuracy. This condition will be very difficult to reach especially in case of a high dimension.

In this paper, we propose an new method to deal with these limitations. The idea is based on modifying the initial distribution of the points such that to generate points on both sides of the separating classifier $C_1$ without modifying the analytical formula of this classifier. It will serve to determine the different families in



(a) (b)

**FIGURE 4** A set of points (a) initial distribution, (b) transformed distribution - situation 1.



(a) (b)

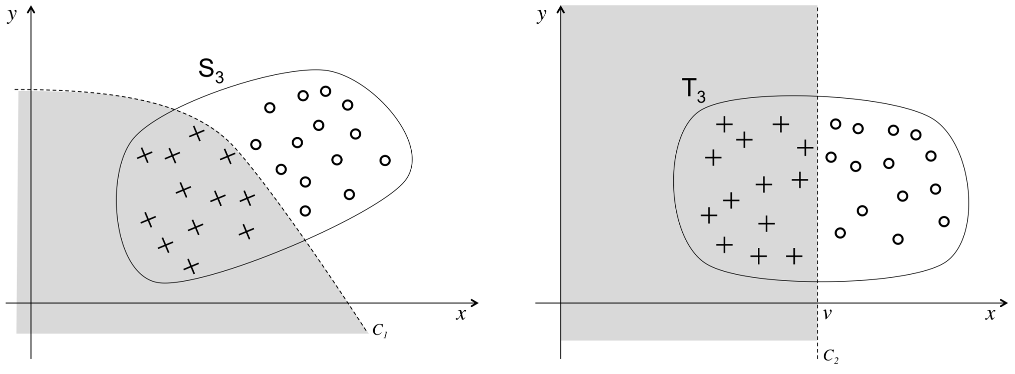
**FIGURE 5** A set of points (a) initial distribution, (b) transformed distribution - situation 2.

the first space in order to find the classifier $C_2$. This is possible because changing the distribution of the points will not lead to another classifier, since $C_2$ depends only on the transformation of $C_1$ and not on the distribution of the points. In other terms, the two families of points are determined with respect to the classifier which is fixed first and not vice versa. Based on this principle, we can then choose a distribution such that for the same number of points we can generate points on both sides of $C_2$, as shown by Figure 6 (b). Figure 6 (a) shows how the classifier $C_1$ can be found from the new distribution of the points.

Also observe that if we can generate many points then the larger the number of generated points the more accurate will be the classifier. However, in case where the number of points is limited, which is the case in our work, for which the simulations can take many months to generate only thousands of points (or circuits), it is possible to improve the classifier model by generating the points around the classifier $C_1$ as shown by Figure 7.

### 4.2 Methodology

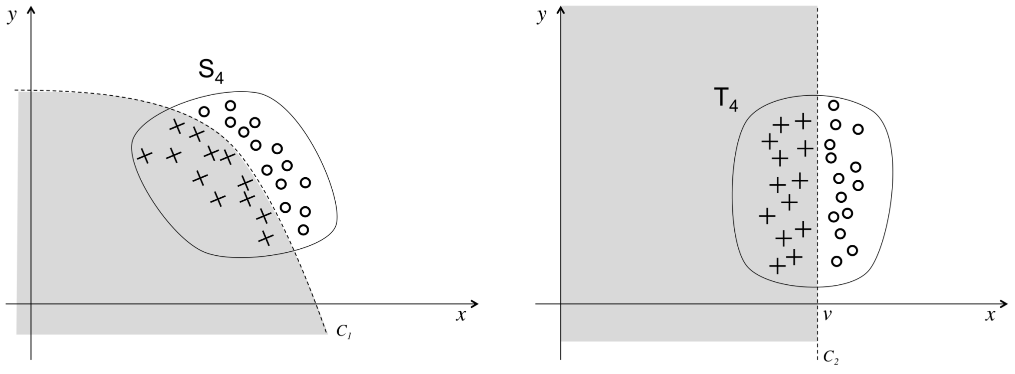
The proposed methodology is based on the previously presented solution. It is composed of two main steps. In the first step, called *the training step*, a classification model is defined and used to predict the state of the circuits : good or faulty. This model consists of three classifiers : the first one classifies the circuits according to their specification, the second one according to the first test limit and the third one according to the second test limit. In the second step, called *the prediction step*, test metrics can be directly estimated using this classification model. As mentioned above, during simulation only a few number of circuits is generated, and there is no generated faulty circuits, and thus, it is not possible to determine the classifier.



(a)

(b)

**FIGURE 6** A set of points (a) initial distribution, (b) transformed distribution - situation 3.



(a)

(b)

**FIGURE 7** A set of points (a) initial distribution, (b) transformed distribution - situation 4.

To overcome this limitation, we will modify the initial distributions of the process parameters in order to force the generation of bad circuits. Changing this distribution will not change the formula of the classifier, as explained above. That is why this step is used only to find the classifier and, indeed, it will not be used to estimate the test metrics. Once the classifier determined, we will, in the second step, just generate the process parameters without running any simulation and without modifying their distributions. In this case, it is possible to generate a huge number of circuits and to reach a very high accuracy in the estimation of the test metrics in a very reduced time (only a few seconds).

Note, that this method represents an improvement over the Statistical Blockade method presented in [31]. Instead of choosing extreme thresholds to generate extreme circuits, we choose as thresholds the specifications and the test limits. The way to generate faulty circuits is based on changing the standard deviation of each process parameter. The test metrics are then estimated directly with less theoretical effort in comparison to the Statistical Blockade method for which extreme value theory (EVT) is required to estimate the test metrics. This theory is very complex and very difficult to apply, especially for test metric estimation in a multivariate space with an important number of performances and test measures [36].

### 4.3 The algorithmic version of the proposed method

Based on this principle, we describe the algorithm of the proposed methodology in order to estimate the Test Escapes $\hat{T}_E$ and the Yield Loss $\hat{Y}_L$, that are defined as follows :

$$\hat{T}_E = \frac{1}{n}\sum_{i=1}^{n} \frac{v_{fp_i}}{v_{p_i}}, \tag{12}$$

$$\hat{Y}_L = \frac{1}{n}\sum_{i=1}^{n} \frac{v_{gf_i}}{v_{g_i}}, \tag{13}$$

where

- $v_{fp_i} (i = 1, ..., n)$ represents the number of circuits that are faulty and pass the test in the $i^{th}$ set,
- $v_{p_i} (i = 1, ..., n)$ represents the number of circuits that pass the test in the $i^{th}$ set,
- $v_{gf_i} (i = 1, ..., n)$ represents the number of good circuits that fail the test in the $i^{th}$ set,
- $v_{g_i}$ represents the number of good circuits in the $i^{th}$ set.

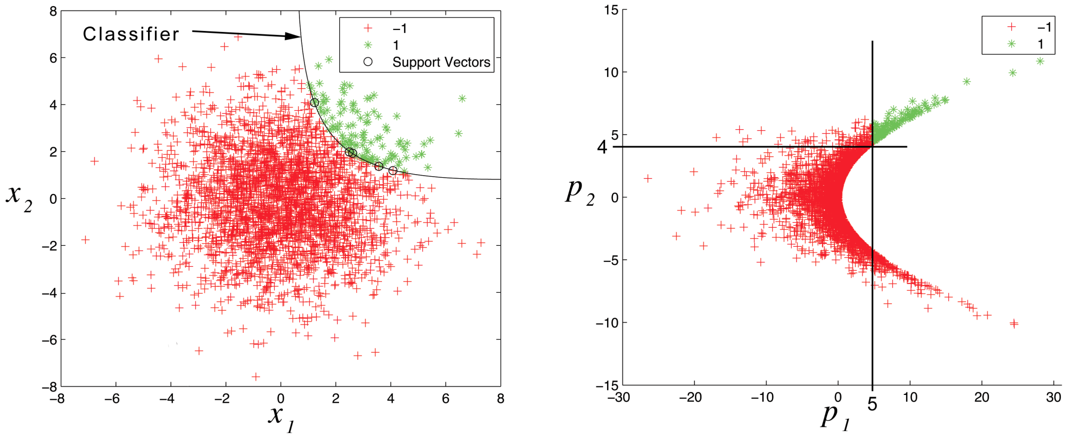
**Algorithm 1** Algorithm of the proposed method

```
Require: The performances $P$, the specifications $s$, the test measures $T$, and the test limits $l$ of circuits.
Ensure: The Test Escapes $\hat{T}_E$ and the Yield Loss $\hat{Y}_L$.
STEP 1 : Determine the classification model
Modify the distributions of the process parameters. We suggest to multiply the standard deviation of each process parameter by a factor $k > 1$ in order to force the generation of circuits of interest for a small number of Monte Carlo circuit simulations,
Run a huge number of Monte Carlo circuit simulations,
Construct the classifiers
    — Determine the classes of good and faulty circuits in the space of the performances based on the specifications,
    — Determine the classes of pass and fail circuits in the space of the performances based on the first test limit,
    — Determine the classes of pass and fail circuits in the space of the performances based on the second test limit,
Determine the corresponding circuits in the space of the process parameters for each case,
Determine the tree classifiers that separate the faulty from the good circuits and those that pass from those that fail in the space of the process parameters,
Assign the initial distribution to each process parameter.
STEP 2 : Estimate the Test metrics
Generate $n$ sets of a very important number (millions) of the process parameters without running the circuit simulation,
Based on the classifiers, calculate the values of $v_{fp_i}$, $v_{p_i}$, $v_{gf_i}$, and $v_{g_i}$, $i = 1, ..., n$.
Calculate the estimated test metrics $\hat{T}_E$ and $\hat{Y}_L$ with the formulas (12) and (13) respectively.
```

### 4.4 A numerical example

To illustrate the proposed method for each approach, let us consider the example of virtual circuits with two process parameters $x_1$ and $x_2$ that follow a Gaussian distribution and two performances $p_1 = x_1 \times x_2$ and $p_2 = x_1 + x_2$. Figure 8 (a) shows 5000 samples of the couple $(x_1, x_2)$ and Figure 8 (b) shows 5000 samples of the couple $(p_1, p_2)$. The green points represent circuits that are faulty and the red ones represent circuits that are good. The green points of Figure 8 (a) in the process parameters space correspond to the green points of Figure 8 (b) where the performance $p_1$ is greater than $5$ and the performance $p_2$ is greater than $4$. Note that these green points are not generated for $5000$ circuits with standard $(\sigma = 1)$ Gaussian distribution of the process parameters. To overcome this limitation, we propose to change the standard deviation of these parameters to $\sigma = 2$. Indeed, this operation changes the original distribution of the process parameters, but we do this only to determine the classifier which is represented by the black curve of Figure 8 (a). We can do this because the classifiers remain the same for any distribution of the process parameters.

Once the classifier is determined using each technique, it is possible to estimate the test metrics by considering the standard deviations of the process parameters. Figure 9 shows $10^6$ samples of circuits that can be classified using the classifier determined in the previous step. The main advantage of this method is that these generated process parameters can be directly applied to estimate the test metrics without running



(a) (b)

**FIGURE 8** An example for process parameter classification (a) and performance classification (b).

any additional simulation since the classifier separates the faulty from the good circuits instead of extreme from non-extreme circuits as presented in [36] and [31].

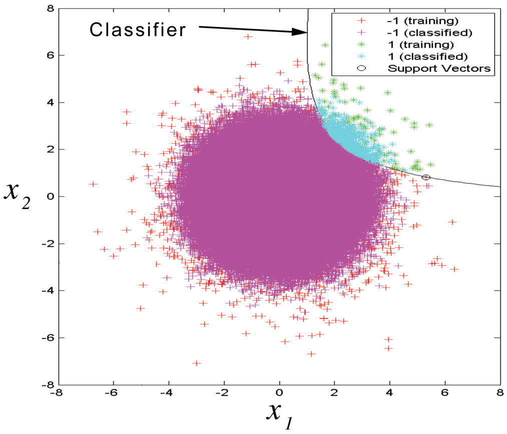


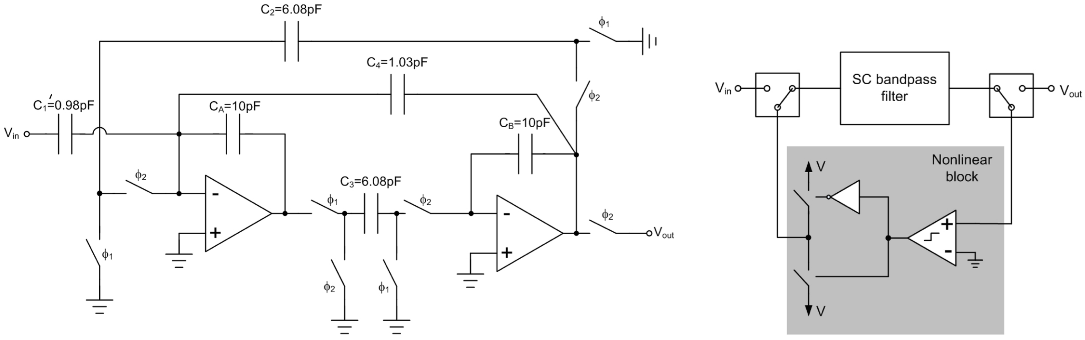
**FIGURE 9** A process parameter classification example.

## 5 Case study : an Oscillation Test

In this section, we present a case study on the application of our methodology to estimate some performances by SVM and Neural Network classification, which is the same as the one presented in [37]. In their article, the authors sought a practical example in which a performance $P$ and a candidate replacement test $T$ can both be expressed explicitly in terms of the circuit parameter vector $p$. In this scenario, the mathematical model can effectively replace the transistor level simulations, which take a long time, allowing us to calculate very quickly the test metrics with exact ppm precision.

As soon as the test measurements are precisely known, we can evaluate the quality of the estimates obtained by the proposed method. Note that, in general, it is impossible to draw such an explicit mathematical model, indicating the urgent need to develop estimation methods like the one proposed here.

To do that, we choose an oscillation test example that has a proper mathematical modeling. An oscillation test consists of reconfiguring the circuit under test (CUT) into a new circuit which oscillates [38], [39]. The



(a) (b)

**FIGURE 10** (a) High-Q SC bandpass filter ; (b) Oscillation test strategy.

test is done by measuring the oscillation frequency and/or the oscillation amplitude. An oscillation test is a low-cost approach : it requires a negligible DfT overhead, and the test signature can be exactly analyzed applying pure digital circuitry. So, we use the proposed method to evaluate an oscillation test strategy for a high-Q SC bandpass filter, see Figure 10 (a). During the oscillation test mode, the CUT is combined with a feedback loop that contains a nonlinear block [38], as presented in Figure 10 (b). Therefore, the circuit is forced to oscillate with a frequency $f_{osc}$. We stress here to answer the following questions. Is it possible to replace the test required to measure the maximum attenuation $P = \alpha_{max}$ in the pass band of the CUT with the oscillation test $T = f_{osc}$ ? What will be the effective parametric test escape and yield loss due to this replacement ?

The CUT is designed to have central frequency $f_0 = 10kHz$, bandwidth $600Hz$, and gain $0dB$ at $f = f_0$. The transfer function in the $z$ domain is

$$H(z) = \frac{\frac{C_1' C_3}{C_A C_B - C_3 C_4}\left(z^{-2} - z^{-1}\right)}{z^{-2} + \frac{C_2 C_3 + C_3 C_4 - 2C_A C_B}{C_A C_B - C_3 C_4} z^{-1} + \frac{C_A C_B}{C_A C_B - C_3 C_4}}.$$

The CUT is most sensitive to variations in the capacitors. Therefore, in general, we suppose that only the capacitors vary in a Monte Carlo circuit simulation, i.e.,

$$p = [C_1', C_2, C_3, C_4, C_A, C_B] .$$

Each instance $p$ defines a circuit instance with process variations, resulting from a specific $\alpha_{max}(p)$ and $f_{osc}(p)$ [38]

$$\alpha_{max}(p) = \max\left\{H\left(e^{jw_{3dB,1}/f_s}\right), H\left(e^{jw_{3dB,2}/f_s}\right)\right\} \tag{14}$$

$$f_{osc}(p) = \frac{f_s}{2\pi}\arccos\left[\frac{1}{2}\frac{C_2 C_3}{C_A C_B} - 1\right], \tag{15}$$

where $f_s$ is the clock frequency, and $w_{3dB,1}$, $w_{3dB,2}$ represent the $3dB$ frequencies at the edges of the pass band.

## 5.1 Direct estimation

To estimate the exact test metrics in (1) and (2), it suffices to calculate $\alpha_{max}(p)$ and $f_{osc}(p)$ for a large number (e.g. $\gg 1$ million) of instance $p$. The specification of $\alpha_{max}$ is set at $s = 10.6609dB$, corresponding to $\mu_{\alpha_{max}} + 5 \times \sigma_{\alpha_{max}}$, with $\mu_{\alpha_{max}} = 4.57dB$ and $\sigma_{\alpha_{max}} = 1.24dB$ representing, respectively, the mean value and the standard deviation of $\alpha_{max}$. The test limits of $f_{osc}$ are set visually using the scatter plot of Figure 11, that shows the projection of $10^4$ random circuits into the $(\alpha_{max} - f_{osc})$ space. The two-sided scatter plot is due to the max operation in (14). The horizontal lines represent the lower $t_l = 3.98 \times 10^4$ and

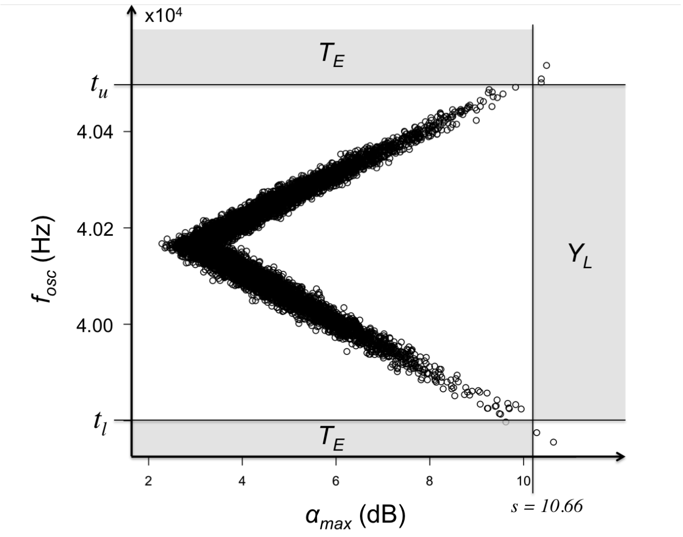


**FIGURE 11** The projection of $10^4$ devices into the $(\alpha_{max} - f_{osc})$ space.

upper $t_u = 4.05 \times 10^4$ limits of $f_{osc}$, whereas the vertical line represents the specification $s$ of $\alpha_{max}$. The regions that correspond to $T_E$ and $Y_L$ are separated by the lines. We can see that $f_{osc}$ correlates well with $\alpha_{max}$. The exact test metrics are computed based on a random population of 100 million circuit samples. It is found that $\hat{T}_{E_1} = 72ppm$ and $\hat{Y}_{L_1} = 59ppm$.

### 5.2 Estimation with the proposed method

We will apply the proposed methodology described in Section 4 to estimate the test metrics of circuits using SVM and Artificial Neural Network methods and compare the test metrics for each algorithm with the real values.

For the SVM, we use the support vector machines algorithm in the R package for kernel learning, and SVM-A proposed in [40,41] and based on previous work [42,43,44]. In our study, the upper bound $C$ is confined to 10, because with this parameter, the SVM classifier can achieve good classification performance. We mainly investigate the performance of the RBF kernel function. The values of the kernel parameter $\gamma$ vary across $\{0.01, 0.1, 1, 2, 4, 8, 16, 32, 128, 256\}$ and we select an optimal $\gamma$ depending on the classification performance of the classifier.

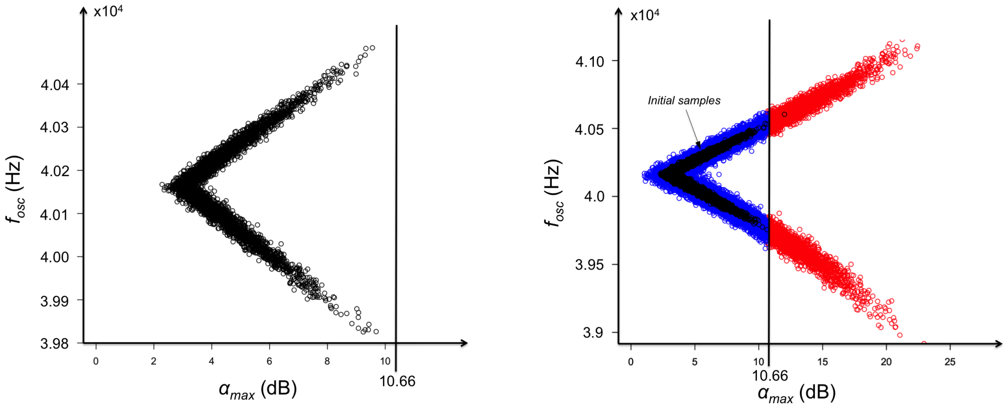
For ANN, we use the back-propagation neural network package integrated into the R environment. The package allows flexible settings through custom-choice of error and activation function. In our study, we choose the use of *softmax* as activation function, the number of units in the hidden layer is token 100 and the maximum number of iterations is confined to 200.

We consider that the CUT has 6 process parameters. The distribution of each of them is Gaussian. The parameters of each distribution are given in Table 1.

In the following we will use SVM-R to designate the classical SVM method implemented in the R software and SVM-A to designate the SVM method that we have proposed in [41]. To run the simulations, we have generated 20000 circuits to do the classification for the SVM-R and the ANN methods and only 5000 circuits for the SVM-A method. Figure 12 (a) shows the 20000 generated circuits. As we can see, there is no faulty circuits ($\alpha_{max} > 10.66$) that are generated. 12 (b) shows 20000 generated circuits where the standard deviation of $\alpha_{max}$ and of $f_{osc}$ is multiplied by 3. As we can see, faulty circuits are generated (in red).

Figure 13 shows 3D scatter plots of the distribution of some process parameters. The black points corresponds to the values that generates faulty circuits (the red points of Figure 12 (a)).

```
STEP 1 : Determine the classification model
We multiply by $2$ the standard deviation $\sigma_i$ $(i = 1, ..., 6)$ of the Gaussian of each process parameter $p_i$ in order to force the generation of circuits of interest,
We run $20k$ Monte Carlo circuit simulations in order to generate $20k$ process parameters $p^j$, performances $P^j$ and test measures $T^j$, where $j = 1, ..., 20k$,
We have determined the classes of good circuits $G1 = \{j : P^j \leq s, j = 1, ..., 20k\}$ and that of faulty circuits $B1 = \{j : P^j > s, j = 1, ..., 20k\}$,
We have determined the classes of passed circuits $G2 = \{j : T^j \geq t_l, j = 1, ..., 20k\}$ and that of failed circuits $B2 = \{j : T^j < t_l, j = 1, ..., 20k\}$,
We have determined the classes of passed circuits $G3 = \{j : T^j \leq t_u, j = 1, ..., 20k\}$ and that of faulty circuits $B3 = \{j : T^j > t_u, j = 1, ..., 20k\}$,
We have determined the corresponding circuits in the space of the process parameters for each case (i.e., $p^{G1} = \{p^j, j \in G1\}, p^{G2} = \{p^j, j \in G2\}, p^{G3} = \{p^j, j \in G3\}$ and $p^{B1} = \{p^j, j \in B1\}, p^{B2} = \{p^j, j \in B2\}, p^{B3} = \{p^j, j \in B3\}$),
We have determined the classifiers that separate the good from the faulty circuits and passed from failed in the space of the process parameters for each case (i.e., that separates $p^{G1}$ from $p^{B1}$ and that separates $p^{G2}$ from $p^{B2}$),
We have determined the corresponding circuits in the space of the process parameters, and there are four cases :
Case 1 : If $(P^j \in G1) and ((T^j \in B2) or (T^j \in B3))$, the corresponding circuit is good and fails the test,
Case 2 :If $(P^j \in B1) and ((T^j \in B2) or (T^j \in B3))$, the corresponding circuit is faulty and fails the test,
 Case 3 : If $(P^j \in G1) and ((T^j \in G2) and (T^j \in G3))$, the corresponding circuit is good and passes the test,
Case 4 : If $(P^j \in B1) and ((T^j \in G2) and (T^j \in G3))$, the corresponding circuit is faulty and passes the test,
We have then assigned the initial distribution to each process parameter (i.e., we multiply by $1$ the standard deviation $\sigma_{i,i=1,...,6}$ of each Gaussian of each process parameter $p_i$ for $i = 1, ..., 6$).
STEP 2 : Estimate the test metrics
We have generated $1000$ sets of $1$ million process parameters without running the circuit simulation,
Based on the classifier constructed in the Step 1 :
We have calculated the value $v_{fp_i}$ that represents the number of circuits that are faulty and pass the test in the $i^{th}$ set (i.e., the $4^{th}$ case),
We have calculated the value $v_{p_i}$ that represents the number of circuits that pass the test in the $i^{th}$ set (i.e., the $3^{th}$ and $4^{th}$ cases),
The test metric $\hat{T}_E$ can then be estimated by :
    $$\hat{T}_E = \frac{1}{1000} \sum_{i=1}^{n} \frac{v_{fp_i}}{v_{p_i}},$$
We have calculated the value $v_{gf_i}$ $(i = 1, ..., n)$ which represents the number of circuits that are good and fail the test in the $i^{th}$ set (i.e., $1^{st}$ case),
We have calculated the value $v_{g_i}$ that represents the number of good circuits in the $i^{th}$ set (i.e., the $1^{st}$ and $3^{rd}$ cases),
The $\hat{Y}_L$ can then be estimated by :
    $$\hat{Y}_L = \frac{1}{1000} \sum_{i=1}^{n} \frac{v_{gf_i}}{v_{g_i}}.$$
```

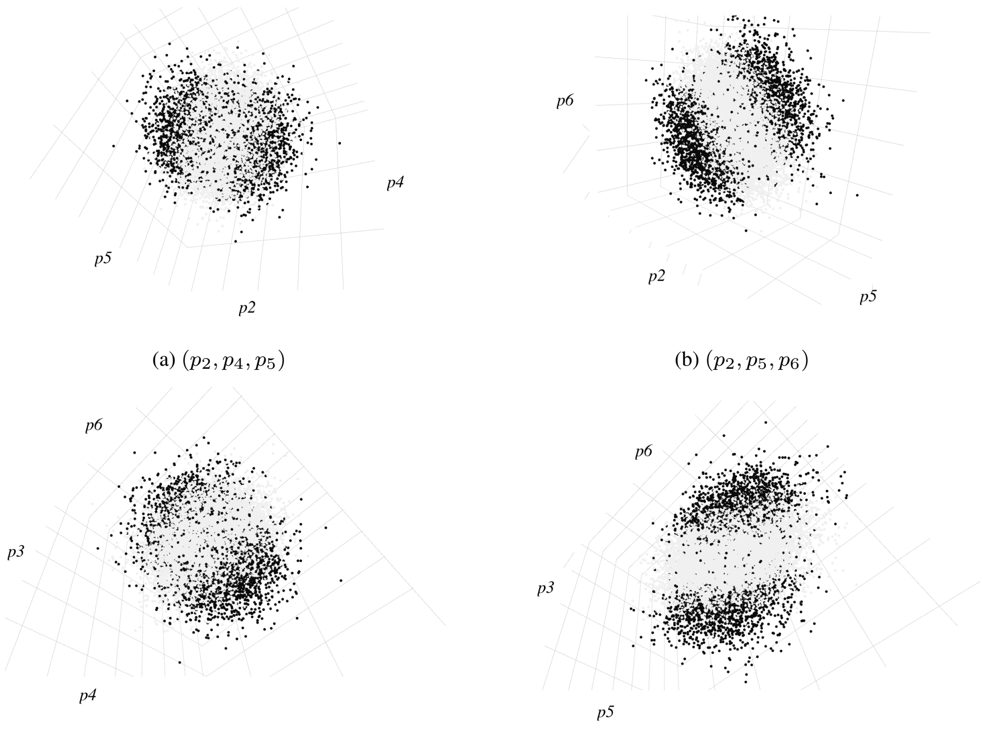


(a) (b)

**FIGURE 12** The initial (a) and the modified (b) joint distribution of the performance $\alpha_{max}$ and the test measure $f_{osc}$.

**TABLE 1** The Gaussian parameters of the process parameters

| Process parameter | $\mu$ $(\times 10^{-11})$ | $\sigma(\times 10^{-13})$ |
|---|---|---|
| $p_1$ | 0.0981 | 0.0981 |
| $p_2$ | 0.6082 | 0.6082 |
| $p_3$ | 0.6082 | 0.6082 |
| $p_4$ | 0.1033 | 0.1033 |
| $p_5$ | 1.0 | 1.0 |
| $p_6$ | 1.0 | 1.0 |



(a) $(p_2, p_4, p_5)$

(b) $(p_2, p_5, p_6)$

(c) $(p_3, p_4, p_6)$

(d) $(p_3, p_5, p_6)$

**FIGURE 13** A 3D scatter plot of some process parameters.

Then we have generated $10^6$ samples for each method but without running any simulation. We have just classified the circuits and estimated the test metrics based on the type of circuits predicted by the classifiers. The obtained results of the proposed methodology using SVM and ANN are listed in Table 2. We have compared these methods regarding the test metric values $T_E$ and $Y_L$ at ppm level with the Statistical Blockade method as presented in [32], and with the real (reference) values.

**TABLE 2** The Obtained Test Metrics

| Method | Test Escapes $(T_E)$ [95% confidence interval] | Yield Loss $(Y_L)$ [95% confidence interval] |
|---|---|---|
| SVM-R (classic) | 73 $[55, 91]$ | 55 $[41, 69]$ |
| SVM-A [41] | 75 $[63, 87]$ | 57 $[49, 65]$ |
| Neural Network (ANN) | 78 $[50, 106]$ | 55 $[43, 67]$ |
| Real Values | 72 | 59 |

Table 2 shows that the obtained values of $T_E$ and $Y_L$ of our method are equal to the values estimated directly on a reference set of circuits of size of one million. These results confirm the effectiveness of our proposition in terms of test accuracy and test cost. Figure 14 shows the obtained results in form of $95\%$ confidence intervals. The gray line shows the real values. We have not included the results of the Statistical Blockade in the table because we took directly those presented in [32] ($T_E = 35ppm$ with a $95\%$ confidence interval $[0, 88]$ and $Y_L = 142$ with a $95\%$ confidence interval $[4, 291]$). These results show that they are very far from the real result of the same paper with differences varying from 30 to 40 ppm while our results are far from the real results with a difference varying from 1 to 6 ppm.

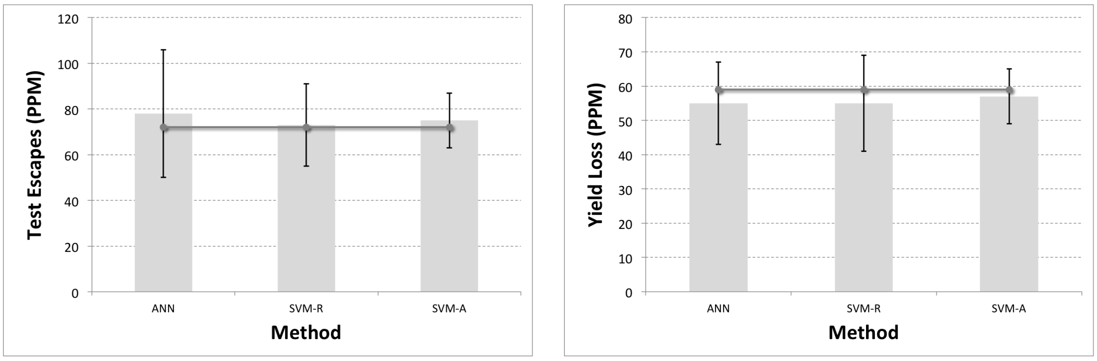


**FIGURE 14** Test metrics estimation with confidence intervals.

## 6 Conclusion

In this paper, we have introduced a new methodology to evaluate test metrics. It is based on basic feature classification using machine learning techniques SVM that we have proposed in [41], ANN, and the Statistical Blockade technique. The classification of the circuits is based on the specifications and the test limits instead of the extreme thresholds. The proposed model is defined and used to predict the state of circuits, good/faulty and pass/fail. The main principle of the proposed method is based on modifying the initial distribution of the process parameters to generate many families of circuits in order to determine the classifiers by running a small number of simulations. Once the classifiers are determined, the initial distribution is used to estimate the test metrics only by generating process parameters without running any simulation. This allows to generate a very large number of circuits in a few seconds. The illustration of the proposed method is presented for the evaluation of a filter BIST technique. The main advantage in using this filter is that its simulation is very quick and we were able to simulate millions of circuits. This set of circuits is useful for a comparison of the proposed methods with the existing ones, since we have the reference values of the test metrics estimated on that set. Experimental results obtained by use of the proposed classical classification methods have been compared and shown that the proposed method gives results that are very close to the real ones and more accurate than those obtained with the Statistical Blockade method.

## Références

1. K. Beznia and A. Bounceur and R. Euler and S. Mir, A Tool for Analog/RF BIST Evaluation Using Statistical Models of Circuit Parameters, ACM Trans. Des. Autom. Electron. Syst., 20(2), 31 :1-31 :22 (2015)
2. A. Bounceur and B. Brahmi and K. Beznia and R. Euler, Accurate analog/RF BIST evaluation based on SVM classification of the process parameters, The 9th IEEE International Design and Test Symposium (IDT), 55-60 (2014)
3. V. Vapnik, The nature of statistical learning theory. Springer-Verlag, New York, USA (1995)
4. M. T. Hagan and H. B. Demuth and M. H. Beale and others, Neural network design. Pws Pub. Boston (1996)
5. S. Biswas and P. Li and R. D. Blanton and L. T. Pileggi, Specification test compaction for analog circuits and MEMS, The conference on Design, Automation and Test in Europe-Volume 1, 164-169 (2005)
6. J. Huang and X. Hu and F. Yang, Support vector machine with genetic algorithm for machinery fault diagnosis of high voltage circuit breaker, Measurement, 44(6), 1018-1027 (2011)

7. B. Long and S. Tian and H. Wang, Diagnostics of filtered analog circuits with tolerance based on LS-SVM using frequency features, Journal of Electronic Testing, 28(3), 291-300 (2012)

8. S. Cai and H. Yuan and J. Lv and Y. Cui, Application of IWO-SVM approach in fault diagnosis of analog circuits, 25th Chinese Control and Decision Conference (CCDC), 4786-4791 (2013)

9. A. S. S. Vasan and B. Long and M. Pecht, Experimental validation of LS-SVM based fault identification in analog circuits using frequency features, Engineering Asset Management, Springer, 629-641, (2014)

10. F. Aminian and M. Aminian and H. W. Collins Jr, Analog fault diagnosis of actual circuits using neural networks, IEEE Transactions on Instrumentation and Measurement, 51(3), 544-550 (2002)

11. H-G. Stratigopoulos and P. Drineas and M. Slamani and Y. Makris, RF specification test compaction using learning machines, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 18(6), 998-1002 (2010)

12. L. Yuan and Y. He and J. Huang and Y. Sun, A new neural-network-based fault diagnosis approach for analog circuits by using kurtosis and entropy as a preprocessor, IEEE Transactions on Instrumentation and Measurement, 59(3), 586-595 (2010)

13. S. Biswas and R. D. Blanton, Statistical test compaction using binary decision trees, IEEE Design & Test of Computers 23(6), 452-462 (2006)

14. A. Gómez-Pau and L. Balado and J. Figueras, MS test based on specification validation using octrees in the measure space, 18th IEEE European Test Symposium (ETS), 1-6 (2013)

15. P. M. Lin and Y. S. Elcherif, Analog circuits fault dictionary : New approaches and implementation, International Journal of Circuit Theory and Applications, 13(2), 149-172 (1985)

16. J. Starzyk and D. Liu and Z-H. Liu and D. E. Nelson and J. O. Rutkowski and others,Entropy-based optimum test points selection for analog fault dictionary techniques, IEEE Transactions on Instrumentation and Measurement, 53(3), 754-761 (2004)

17. D. Grzechca and T. Golonek and J. Rutkowski, Analog fault AC dictionary creation-the fuzzy set approach, IEEE International Symposium on Circuits and Systems (ISCAS), (2006)

18. A. Pulka, A Heuristic Fault Dictionary Reduction Methodology , 14th IEEE International Conference on Electronics, Circuits and Systems, (2007)

19. N. Akkouche and S. Mir and E. Simeu and M. Slamani Analog/RF test ordering in the early stages of production testing, IEEE 30th VLSI Test Symposium (VTS), 25-30 (2012)

20. C-Y. Chao and H-J. Lin and L. Miler, Optimal testing of VLSI analog circuits, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 16(1), 58-77 (1997)

21. L. Milor and A. L. Sangiovanni-Vincentelli, Minimizing production test time to detect faults in analog circuits, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 13(6), 796-813 (1994)

22. S. S. Akbay and J. L. Torres and J. M. Rumer and A. Chatterjee and J. Amtsfield, Alternate test of RF front ends with IP constraints : Frequency domain test generation and validation, IEEE International Test Conference (ITC'06), 1-10 (2006)

23. J. B. Brockman and S. W. Director, Predictive subset testing : Optimizing IC parametric performance testing for quality, cost, and yield, IEEE Transactions on Semiconductor Manufacturing, 2(3), 104-113 (1989)

24. R. Voorakaranam and S. S. Akbay and S. Bhattacharya and S. Cherubal and A. Chatterjee, Signature testing of analog and RF circuits : Algorithms and methodology, IEEE Transactions on Circuits and Systems I : Regular Papers, 54(5), 1018-1031 (2007)

25. S. Sunter, Mixed-signal testing and DfT, Advances in Electronic Testing, Springer, Gizopoulos Ed., 301-336 (2006)

26. M. Dubois and H-G. Stratigopoulos and S. Mir, Hierarchical parametric test metrics estimation : A $\Sigma\Delta$ converter BIST case study, IEEE International Conference on Computer Design (ICCD), 78-83 (2009)

27. K. Beznia and A. Bounceur and S. Mir and R. Euler, Statistical modelling of analog circuits for test metrics computation, 8th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS), 25-29 (2013)

28. A. Bounceur and S. Mir and E. Simeu and L. Rolíndez, Estimation of test metrics for the optimization of analog circuit testing, Journal of Electronic Testing : Theory and Applications (JETTA), 23(6), 471-484 (2007)

29. A. Bounceur and S. Mir and H-G. Stratigopoulos, Estimation of analog parametric test metrics using copulas, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 30(09), 1400-1410 (2011)

30. H-G. Stratigopoulos and S. Mir and A. Bounceur, Evaluation of analog/RF test measurements at the design stage, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,28(4), 582-590 (2009)

31. A. Singhee and R. A. Rutenbar, Statistical Blockade : Very fast statistical simulation and modeling of rare circuit events and its application to memory design, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28(8), 1176-1189 (2009)

32. H. Stratigopoulos, Test Metrics Model for Analog Test Development, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 31(07), 1116-1128 (2012)

33. K. Beznia and A. Bounceur and S. Mir and R. Euler, Accurate estimation of analog test metrics with extreme circuits, IEEE International Conference on Electronics, Circuits, and Systems (ICECS'12), (2012)

34. C-W. Hsu and C-C. Chang and C-J. Lin and others, A practical guide to support vector classification. (2003)

35. X-F. Gu and L. Liu and J-P. Li and Y-Y. Huang and J. Lin, Data Classification based on Artificial Neural Networks, International Conference on Apperceiving Computing and Intelligence Analysis, 223-226 (2008)

36. K. Beznia, Méthodes statistiques pour l'évaluation des techniques de test de circuits analogiques sous variations paramétriques multiples, Thesis report (2013)

37. H. Stratigopoulos and S. Mir, Analog test metrics estimates with PPM accuracy, IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 241-247 (2010)

38. G. Huertas and D. Vázquez and E. J. Peralías and A. Rueda and J. L. Huertas, Practical oscillation-based test of integrated filters, IEEE Design & Test of Computers, 19(6), 64-72 (2002)

39. K. Arabi and B. Kaminska, Oscillation-test methodology for low-cost testing of active analog filters, IEEE Transactions on Instrumentation and Measurement, 48(4), 798-806 (1999)

40. S. Djemai and B. Brahmi and M. O. Bibi, Méthode primale-duale pour l'apprentissage des SVM, COSI'2014, 189-197 (2014)

41. S. Djemai and B. Brahmi and M. O. Bibi, A primal-dual method for training SVM, 211, 34-40 (2016)

42. R. Gabasov and F.M. Kirillova and V.M. Raketsky and O.I. Kostyukova, Constructive methods of optimization. Volume 4 : Convex Problems, Minsk University Press, (1987)

43. B. Brahmi and M. O. Bibi, Dual support method for solving convex quadratic programs, Optimization, 59, 851-872 (2010)

44. S. Radjef and M. O. Bibi, An effective generalization of the direct support method in quadratic convex programming, Applied Mathematics Sciences, 6(31), 1525-1540 (2012)