



**HAL**  
open science

# Machine Learning Support for Diagnosis of Analog Circuits

Haralampos-G. Stratigopoulos

► **To cite this version:**

Haralampos-G. Stratigopoulos. Machine Learning Support for Diagnosis of Analog Circuits. Machine Learning Support for Fault Diagnosis of System-on-Chip, Springer Nature, 2023. hal-03872972

**HAL Id: hal-03872972**

**<https://hal.science/hal-03872972v1>**

Submitted on 26 Nov 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Machine Learning Support for Diagnosis of Analog Circuits

Haralampos-G. Stratigopoulos

**Abstract** We discuss the state-of-the-art on fault diagnosis for analog circuits with a focus on techniques that leverage machine learning. For a chip that has failed either in post-manufacturing testing or in the field of operation, fault diagnosis is launched to identify the root-cause of failure at sub-block level and transistor-level. In this context, machine learning can be used to build a smart system that predicts the fault that has occurred from diagnostic measurements extracted on the chip. We discuss the different elements of a diagnosis flow for analog circuits, including fault modeling, fault simulation, diagnostic measurement extraction and selection, and the machine learning algorithms that compose the prediction system. We also demonstrate a machine learning-based diagnosis flow on an industrial case study.

## 1 Introduction

The fabrication process of Integrated Circuits (ICs) is imperfect and is likely that a fabricated IC will not meet the intent specifications. Main sources of failure include silicon defects and variations in process parameters. For this reason, before deployment in an application all fabricated ICs need to go through post-manufacturing testing which aims at screening out non-functional instances. Post-manufacturing testing procedures include wafer-level testing aiming at identifying gross instabilities in the fabrication process, final testing of packaged dies aiming at verifying that the actual design specifications of the IC are met, and burn-in testing, where the chip is exercised sufficiently long in stress conditions, in order to avoid early in-use system failures. The demanded outgoing quality level is defined by the criticality of the application. Safety-critical and mission-critical applications, i.e., automotive, biomedical, defense, aerospace, etc., require zero defective parts per million (DPPM),

---

Haralampos-G. Stratigopoulos  
Sorbonne Université, CNRS, LIP6, 4 Place Jussieu, 75252 Paris, France, e-mail: haralampos.stratigopoulos@lip6.fr

which poses a great challenge. Even more, an IC that passes post-manufacturing testing may fail during mission-mode due to latent defects, i.e., defects that manifest themselves later during the application, silicon aging, or environmental factors. For critical applications it is often demanded that the IC is self-tested so as to detect reliability hazards and take preventive actions, i.e., halt the application before a fatal consequence occurs. Self-testing requires on-chip resources and is performed on-line in idle times or concurrently with the operation.

Diagnosis of ICs refers to the analysis performed to identify the root cause of a hardware-level failure that has occurred. Diagnosis can be launched on an IC that fails post-manufacturing testing before deployment in an application or on an IC that fails in the field of application and is returned by the customer. In this latter scenario, failures are not easy to reproduce in the laboratory as the real mission conditions and executed workload are unknown and cannot be exhaustively modeled. The diagnosis output is the isolation of the fault responsible for the failure, its localization at transistor-level, as well as its quantification, e.g., the deviation of a circuit parameter from its nominal expected value.

Diagnosis is a crucial step in a product life-cycle. In the first phases of production, it can reveal important statistics, such as defect distribution and yield detractors. It can assist the designers in gathering valuable information regarding the underlying failure mechanisms. The objective here is to make use of the diagnosis results to enhance yield for future products through improvement of the fabrication environment and development of design techniques that minimize the failure rate, and also to evaluate and improve the quality of post-manufacturing tests. In the case of failures in the field of application, diagnosis of customer returns is important to identify the root-cause of failure so as to repair the system if possible, gain insight about environmental conditions that can jeopardize the system's health, and apply corrective actions that will prevent failure re-occurrence and, thereby, expand the safety features.

Diagnosis is a multi-step procedure. In a first step, diagnosis generates a set of candidate hardware-level faults (or defects) based on diagnostic measurements or observed symptoms. In a second step, the IC is physically examined, for example using a thermal camera or laser probing, to highlight anomalies in the operation and narrow down further the set of candidate defects. In the third and last step, the IC is submitted to Physical Failure Analysis (PFA) where de-layering and cross-sectioning of the die is performed to confirm the defect using imaging. Since PFA is destructive and irreversible, ideally the first step should pinpoint the actual defect. However, very often the first step results in ambiguity groups of candidate defects. According to industrial experience, the size of ambiguity groups should be less than 5-10 to increase the PFA success rate. Last but not least, an additional constraint is that the diagnosis should be completed in reasonable time, i.e., typically maximum in a few days, given that the number of diagnoses performed per week per design can be in the order of thousands. Achieving a short diagnosis cycle is challenging considering the status of diagnosis tools that are available today.

In summary, diagnosis metrics include: (a) resolution, i.e., the size of an ambiguity group; (b) accuracy, i.e., whether the reported diagnosed defect corresponds to the actual defect that has occurred; and (c) diagnosis cycle time.

An IC comprises numerous interconnected blocks that are heterogeneous, i.e., processor, memory, analog, mixed-signal, RF, etc. Diagnosis starts by first identifying the source of the failure at the system-level, that is, the failed sub-block. This can be achieved by post-manufacturing testing using the test infrastructure already present on-chip to support post-manufacturing testing, i.e., on-chip test instruments, test access and control mechanisms, etc. [112, 86]. The rest of the diagnosis procedure aiming at identify the root cause of observed failure at transistor-level is specific to the failed block type.

While for digital blocks there exists several in-house frameworks and commercial Electronic Design Automation (EDA) tools for diagnosis [62, 57, 17, 115, 78, 80], for analog blocks there exists neither a commercial tool nor a standardized diagnosis approach. Analog diagnosis is still an ad-hoc, manual, tedious, and time-consuming process very often resulting in large ambiguity groups and no actionable diagnosis information. In fact, there is a vast literature on analog diagnosis [12, 45, 19], yet none of the proposed solutions has matured enough to meet industry standards.

In this Chapter, we discuss diagnosis for analog circuits, where the term “analog” is used in a broad sense and includes also mixed-signal and RF circuits. The focus will be on the use of machine learning support for diagnosis. Machine learning finds numerous applications in several other test-related tasks [102], i.e., test cost reduction, yield learning, adaptive testing, post-manufacturing circuit tuning, outlier detection, test metrics estimation, etc. Essentially, diagnosis can be viewed as a pattern recognition task. Each candidate fault has its own pattern, where in the diagnosis context the pattern is composed of the values of a set of diagnostic measurements, and the goal is to distinguish the patterns of any two different faults.

We will start by discussing fault modeling and fault simulation for analog circuits in Sections 2 and 3, respectively. Fault modeling results in the fault hypotheses, i.e., the list of potential faults that may be responsible for a circuit failure. Fault simulation is used extensively in a diagnosis flow either in a pre-diagnosis phase to build the diagnosis tools or iteratively during the actual diagnosis procedure. We will provide a concise overview of the aprior art on analog diagnosis approaches in Section 4, and in Section 5 we will present a diagnosis flow based on an ensemble of different machine learning models. In Section 6, we will demonstrate a machine learning-based diagnosis flow on an industrial case study. Section 7 concludes the Chapter and provides perspectives.

## 2 Fault modeling

A prerequisite for launching diagnosis is the *a priori* generation of fault hypotheses. Fault hypotheses compose a fault model, i.e., a set of hardware-level faults (or defects) that may be responsible for the failure, hoping that the true occurring fault is part of this set. For analog ICs there is no widely accepted fault model, unlike digital ICs for which there are well established fault models, i.e., stuck-at, bridging, and delay faults, that have driven for years the development of Automatic Test Pattern

Generation (ATPG) algorithms and diagnosis procedures. The reason is that for analog ICs the fault universe is immeasurable. In general, faults are categorized into parametric (or soft) faults and catastrophic (or hard) faults. A parametric fault refers to the deviation of a component value, i.e., transistor geometry, resistance value, etc. A catastrophic fault refers to a structural change in the circuit topology and is typically modelled as an open-circuit or short-circuit. Component deviation can be in any range and incomplete open-circuits and short-circuits can take any resistance value, thus the fault model size easily explodes.

Inevitably a reduced-size fault model needs to be considered, which may be sufficient for assessing the quality of a test, i.e., based on the resultant fault coverage, but it is problematic for diagnosis since the actual fault occurring may have not been included in the fault model. At best, the diagnosis procedure should respond that the fault is not found. However, due to incomplete fault modeling the diagnosis result may be misleading erroneously predicting another fault in the set.

To this end, a probable list of catastrophic faults can be defined based on historical silicon defective data and Inductive Fault Analysis (IFA) which is used to predict faults that are likely to occur [46]. To reduce the catastrophic fault size, often ideal short-circuits with  $10\Omega$  resistance and ideal open-circuits with  $1M\Omega$  resistance are assumed. To account for the fact that an ideal open-circuit does not exist and cannot be handled by a SPICE simulator, a weak pull-up or pull-down is assigned to each open-circuit [43]. The modeling approach relies on controlling the gate-to-source voltage by the drain-to-source voltage multiplied with a coefficient that depends on transistor parameters. Another simplification is to reduce the number of faults in the transistor terminals. In particular, for MOS transistors traditionally six faults are considered, i.e., shorts across gate-to-source, gate-to-drain, and drain-to-source, and opens in each terminal. However, all shorts have a similar effect on the transistor being stuck-on and all opens have a similar effect on the transistor being stuck-off. Thus, for MOS transistors one can use only gate open and drain-to-source short faults [113].

Parametric faults are more difficult to model. Often fixed percentage variations on component nominal values are considered, i.e., transistor length, transistor threshold voltage, resistance value, etc. A parametric fault can be due to variations in fabrication process-related parameters, i.e., doping concentration variations, but the origin may be as well a silicon defect. In the first case, all process parameters vary simultaneously following a distribution, thus the failure may be due to a combination of component variations. In this case, a parametric fault model can be defined based on Monte Carlo simulations using the process parameter distributions defined in the Process Design Kit (PDK) of the technology [107]. More specifically, successive Monte Carlo simulations are performed to approximate the boundary in the process parameter space that encloses all combinations of process parameters that result in circuit performances compliant to the specifications. The fault model can include circuit instances with process parameter samples distributed around the outer side of the boundary. However, this parametric fault model is not useful for diagnosis since it implies distributed faults, i.e., there is no specific fault location.

So far, we discussed faults occurring during manufacturing. In addition, faults can be induced in the field due to ageing phenomena, including Negative Bias Temperature Instability (NTBI) [93], Hot Carrier Injection (HCI) [116], Time-Dependent Dielectric Breakdown (TDDB) [116], and electromigration [117]. To simplify the analysis, often ageing effects are abstracted and modelled with parametric and catastrophic faults.

Another type of failure mechanism are latent defects that are created during manufacturing but are manifested after some period of operation. The most common latent defect is the rupture of the gate oxide of MOS transistors known as pinhole which accelerates the TDDB [51]. It has been recently shown that a pinhole can be modeled as a decrease in the effective value of the oxide thickness [51].

### 3 Fault simulation

An indispensable step in diagnosis procedures is fault simulation. A main challenge here is that analog simulation can be very long, thus simulating a large number of faults entails a computational burden. For many types of analog ICs, i.e., data converters and Phase-Locked Loops (PLLs), simulation time may be in the order of days. Performing fault simulations at behavioral-level [16] may not provide the necessary resolution to carry out the diagnosis task. Besides, a behavioral-level model is not detailed enough to perform fault injection at this level. A solution to this problem is to consider a divide-and-conquer approach where the circuit is described at transistor-level only in the vicinity of the fault location. More specifically, an analog IC can be decomposed into sub-blocks. Only the sub-block wherein the fault is located is described at transistor-level, while the rest of the blocks are described at behavioral-level, i.e. in a hardware description language (HDL) such as Verilog. A mixed-mode simulation will complete faster compared to simulating the full transistor-level netlist.

Another requirement for efficient diagnosis procedures is the automation of fault simulation. Performing fault simulations manually by injecting one fault at a time is very tedious given the large fault model size. To this end, the recently proposed analog defect simulators [113, 131, 48] can help automating, speeding up, and tackling the complexity of fault simulation and will prove to be very useful for assisting diagnosis.

The main input to a defect simulator tool includes the circuit netlist and sub-block behavioral-level descriptions, a fault model, and test benches. In its basic operation, the tool outputs the list of all possible faults based on the fault model, it injects one fault into the netlist at a time, and simulates each faulty netlist to compute the different test values, i.e., diagnostic measurements.

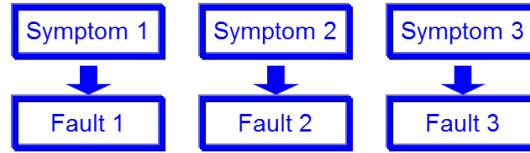


Fig. 1 Rule-based diagnosis.

## 4 Overview of diagnosis approaches for analog circuits

### 4.1 Rule-based diagnosis

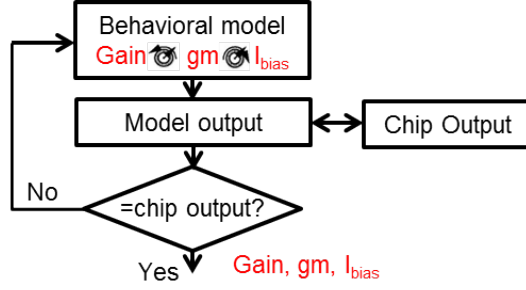
Rule-based diagnosis approaches represent the experience of skilled diagnosticians in the form of rules which generally take the form “IF symptom(s) THEN fault(s)” [45, 42], as illustrated in Fig. 1. This approach can only locate the faulty block in a larger system or an assembly fault, i.e., broken interconnect, but it cannot diagnose faulty components down to the transistor-level. The main challenge with this approach is to acquire the knowledge to build the rules. The problem arises from the fact that different faults often have the same influence on the IC behavior and, thereby, result in the same symptoms. Moreover, typically limited diagnostic information is available. Only few faulty IC samples are available to build the rules, thus not permitting case-based reasoning.

### 4.2 Model-based diagnosis

Model-based diagnosis approaches target parametric faults. The idea is to first build a model linking diagnostic measurements to circuit parameters. Then, given the diagnostic measurements from the real failing device, the model is used to identify the faulty circuit parameter, as well as its deviation from the nominal value. The model can be constructed using nonlinear circuit equations, sensitivity analysis, and behavioral modeling.

#### 4.2.1 Explicit nonlinear diagnosis equations

Explicit nonlinear diagnosis equations take the form  $F(\mathbf{p}, \mathbf{m}) = 0$ , where  $F$  is a matrix with elements that are nonlinear functions of the circuit parameters  $\mathbf{p}$  and diagnostic measurements  $\mathbf{m}$ . Diagnosis equations can be derived analytically using a combination of component connection models, component transfer functions, and composite circuit transfer functions [94, 90]. Given the diagnostic measurement pattern of the failed chip, denoted by  $\mathbf{m}_{chip}$ , the solution  $\mathbf{p}^*$  to these equations can be reached using a Newton-Raphson iteration scheme, namely,



**Fig. 2** Model-based diagnosis based on behavioral modeling.

$J_F(\mathbf{p}^k)(\mathbf{p}^{k+1} - \mathbf{p}^k) = -F(\mathbf{p}^k, \mathbf{m}_{chip})$ , where  $\mathbf{p}^k$  is the  $k$ -th estimate of the solution and  $J_F$  is the Jacobian. This formulation goes along with diagnosability tests, i.e., ambiguity tests, to examine whether  $\mathbf{p}$  can be uniquely determined given  $\mathbf{m}_{chip}$  [47]; however, no automated method exists to select diagnostic measurements that satisfy the diagnosability criterion. Moreover, it is not always guaranteed that the Newton-Raphson scheme will converge to a solution.

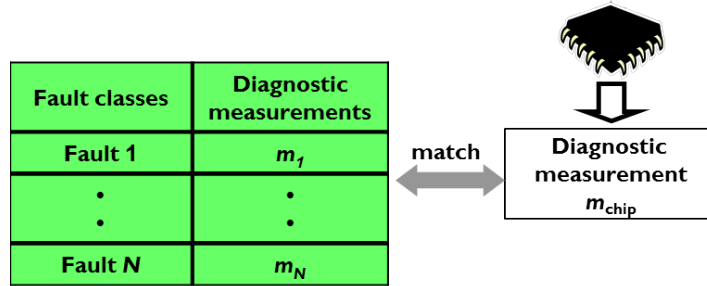
#### 4.2.2 Sensitivity analysis

This approach employs linear error models [36, 97] of the form  $\Delta \mathbf{m} = S(\Delta \mathbf{p}/\mathbf{p})$ , where  $\Delta \mathbf{p}/\mathbf{p}$  is the normalized vector of parameter deviations,  $\Delta \mathbf{m}$  is the vector of diagnostic measurement deviations from the nominal expected values obtained on the fault-free device, and  $S$  is a sensitivity matrix evaluated at the nominal  $\mathbf{p}$ . Thus, we can write  $\Delta \mathbf{p}/\mathbf{p} = (S^T S)^{-1} S^T \Delta \mathbf{m}_{chip}$ , provided that  $(S^T S)^{-1}$  exists. However, in the presence of fault ambiguity,  $S^T S$  is not full rank. Secondly, even with numerically full rank,  $S^T S$  may still be nearly singular, in which case the solution will be unstable. Several algorithms have been proposed to determine fault ambiguity in this formulation resulting in a column-reduced sensitivity matrix with full rank [36, 56]. Clearly, linear error models are inadequate for substantial deviations of  $\mathbf{p}$ . To address this issue, an iterative procedure is implemented that requires to update the sensitivity matrix at each step [36, 97]; however, there is no formal proof that guarantees convergence.

#### 4.2.3 Behavioral modeling

Behavioral model-based techniques rely on generating an approximate behavioral model of the circuit [31, 75, 68]. During fault diagnosis, this reference behavioral model is perturbed by varying behavioral-level parameters until its response matches the response of the faulty chip, as illustrated in Fig. 2. A match is found if  $|\mathbf{m}_{chip} - \mathbf{m}_{model}| < \epsilon$ , where  $\mathbf{m}_{model}$  is the diagnostic measurements provided





**Fig. 3** Fault dictionary-based diagnosis.

by the behavioral model and  $\epsilon$  is a small error quantity. When a match is found, the behavioral parameters that have deviated from their nominal values are retained. Given the mapping between circuit components and behavioral parameters, a candidate set of components which may have caused the failure is retained. The main disadvantage of this approach is the limited diagnosis resolution. It generates the faulty behavioral parameters; however, a behavioral parameter can depend on several components and each component can be assigned several different faults, thus the number of fault candidates can be prohibitively high. Another challenge is to build a behavioral model that faithfully reproduces the analog circuit behavior.

### 4.3 Fault dictionary-based diagnosis

Perhaps the most common diagnosis approach is based on the use of a fault dictionary [98, 76, 99, 23, 9, 61], as illustrated in Fig. 3. A fault dictionary contains fault hypothesis/diagnostic measurement pattern pairs. More specifically, given a list of  $N$  faults  $F_i, i = 1, \dots, N$ , one fault is injected at a time in the netlist, and the circuit is simulated to obtain the diagnostic measurement pattern  $\mathbf{m}(F_i), i = 1, \dots, N$ , where  $\mathbf{m}$  is composed of  $d$  test values. During diagnosis, the same diagnostic measurement pattern is obtained on the failed chip and is compared to the logged diagnostic measurement patterns in the faulty dictionary using a similarity measure  $d(\mathbf{m}(F_i), \mathbf{m}_{chip})$ . The diagnosed fault is the one that presents the most similar diagnostic measurement pattern, i.e.,  $\min_i d(\mathbf{m}(F_i), \mathbf{m}_{chip})$ . This is in essence a pattern recognition, e.g., classification, approach. As such, it is mostly suitable for catastrophic faults whose diagnostic measurement patterns are more separable. Diagnostic measurement patterns of parametric faults tend to overlap and, thereby, are less distinguishable.

#### 4.4 DfT-assisted diagnosis

Design-for-test (DfT) consists in embedding test structures on-chip with the aim to facilitate testing, i.e., improve fault coverage and reduce costs. The cost is dictated by the test application time, as well as the Automatic Test Equipment (ATE) requirements. Built-in Self-test (BIST) is a special form of DfT where the test procedure takes place entirely on-chip without needing to interface the chip to external ATE. DfT and BIST are traditionally used for post-manufacturing test, while BIST can be reused in the field of operation to perform on-line test in idle times or concurrent error detection.

In general, the DfT circuitry can comprise one or more of the following test structures: test access points, digitally-controlled re-configuration schemes, and test instruments, i.e., test stimulus generators, actuators, sensors, checkers, and test response analyzers.

For analog circuits, DfT can be functional, i.e., targeting measuring the performances promised in the datasheet, or fault-oriented, i.e., targeting detecting faults. DfT can be generic and applicable to many circuit classes [64, 32, 85] or can be specific to the circuit class, i.e., linear time-invariant circuits [24, 65, 82, 111], PLLs [122, 114, 73, 66], data converters [39, 11, 88, 49, 71, 13, 26, 77, 91] and RF transceivers [118, 29, 63, 41, 2, 1]. Circuit class specific DfT approaches can also be specific to the architecture within a circuit class. For example, for the class of Analog-to-Digital Converters (ADCs), different DfT approaches exist for Successive Approximation Register (SAR) ADCs, pipeline ADCs,  $\Sigma\Delta$  ADCs, etc.

DfT can assist the diagnosis task in multiple ways. First, for a large circuit, observing primary outputs can result in large ambiguity groups and limited diagnosis accuracy. DfT can help accessing internal sub-blocks and obtaining information-rich diagnostic measurements at sub-block level so as to break the ambiguity groups and improve the overall diagnosis accuracy [126]. For example, on-chip sensors, i.e., current sensors [127, 29], amplitude detectors [63], and temperature sensors [1], can be used in this context. Second, DfT can assist in forcing specially crafted test signals at internal nodes to more effectively sensitize faults and generate distinct diagnostic measurement patterns per fault. Third, DfT can help reconfiguring the circuit to better expose the faults and create distinct diagnostic measurements. An example here is topology modification by using 1-bit controlled Pull-Down (PD) and Pull-Up (PU) transistors to tie a node to power supply or ground, respectively [32]. Using  $K$  PD/PU transistors, we can reconfigure the circuit into  $2^K$  different topologies.

As an example, let us consider in more detail the BIST-assisted diagnosis methodology recently proposed in [84]. The underlying BIST technique is Symmetry-based BIST (SymBIST), which is a generic fault-oriented BIST approach. SymBIST is based on constructing and monitoring invariant properties into the design, as illustrated in Fig. 4. Invariant properties refer to invariant signals, i.e., signals that by construction are confined within a tolerance window for any circuit input. Invariant signals can be built by exploiting symmetries into the design. For example, for a differential sub-block, the sum of the voltages at two symmetrical nodes of the

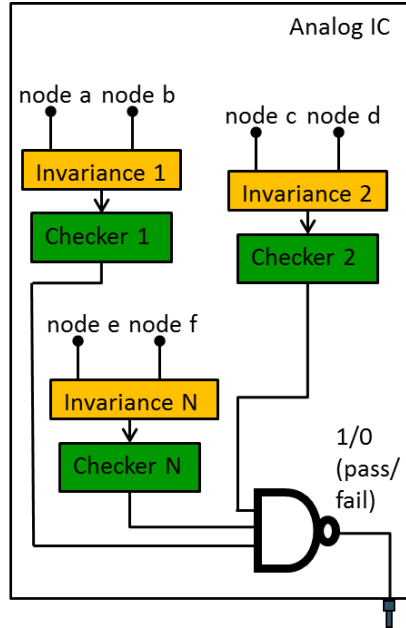


Fig. 4 SymbiST principle of operation.

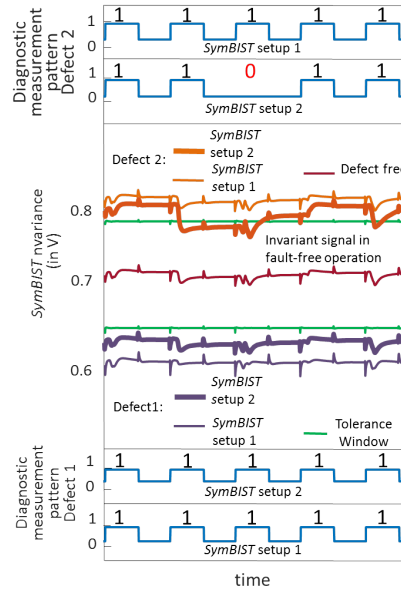


Fig. 5 SymbiST example for diagnosis.

two differential paths is constant [101, 82]. Other invariant signals can be built by summing up complementary signals or subtracting identical signals from replicated sub-blocks. Each invariant signal is then monitored by a checker which flags an error whenever the invariant property is violated, i.e., when the invariant signal slides outside the tolerance window [67, 110]. The checker outputs a 1-bit decision, where by convention 0 refers to invariant property violation. The checker is clocked such that its output observed during a time window corresponds to a bit-string of size  $\ell$ . Invariant properties are identified and built to cover the entire design. The premise is that if a fault occurs, then one or more invariant properties will be violated. For  $K$  checkers, we have a diagnostic measurement pattern  $\mathbf{m}$  of size  $K \times \ell$  that is the concatenation of all checkers' output bit-strings. In fault-free operation,  $\mathbf{m}$  is a vector of ones, while in the presence of a fault some bits will be flipped to zero. The problem then is to vary the test setup, i.e., test stimulus, internal re-configuration, nominal width of tolerance window, etc., so as to obtain a different diagnostic measurement pattern for any two different faults  $F_i$  and  $F_j$ , i.e.,  $\mathbf{m}(F_i) \neq \mathbf{m}(F_j)$ . A fault dictionary-based diagnosis approach is used to identify the fault that has occurred given the diagnostic measurement pattern  $\mathbf{m}_{chip}$ , i.e., checkers' response, of the failed chip.

An example is shown in Fig. 5 considering 1 checker with a 5-bit output, 2 defects, and 2 test setups. The invariant signal in the fault-free scenario is depicted in the middle and is shown to be confined within the tolerance window. For setup 1, the invariant signal violates the lower limit of the tolerance window for defect

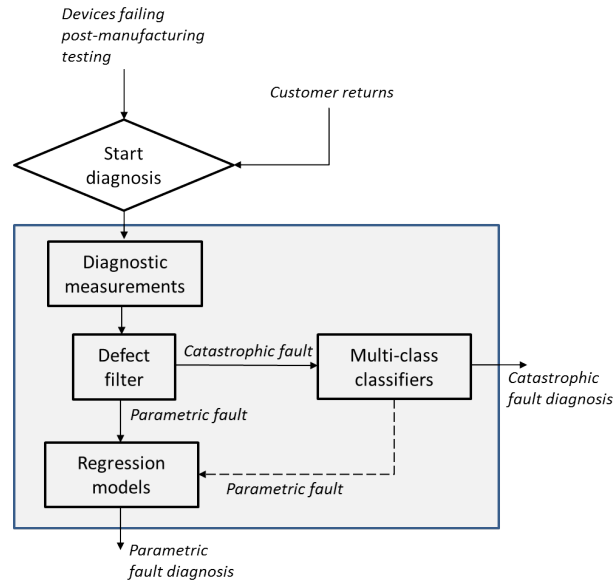
1 and the upper limit of the tolerance window for defect 2. Thus, the diagnostic measurement pattern has all ones for both defects and the two defects cannot be distinguished using test setup 1. For test setup 2, we observe that the invariant signal in the case of defect 2 slides within the tolerance window for some time. As a result, the diagnostic measurement becomes 11011 and is now distinguishable from the diagnostic measurement of defect 1 which remains 11111. In this way, using test setup 2 we can diagnose which of the two defects has occurred.

#### **4.5 Diagnosis with defect simulator in-the-loop**

An iterative diagnosis flow invoking a defect simulator, such as the one in [113], was recently demonstrated in [79]. First the failed sub-block of the chip is identified. Let us assume that provisions have been made to facilitate diagnosis. For example, some test structures have been added on-chip to extract diagnostic measurements at the sub-block level of the chip. Additional diagnostic measurements can be obtained with physical examination of the failed chip, i.e., using imaging or probing, and can be reproduced at simulation level. The next step is to perform fault injection into the netlist of the sub-block using a defect simulator. The complete fault model is employed and all the diagnostic measurements are simulated for each fault. Then, comparing the simulated and measured diagnostic measurements, we can narrow down the list of candidate faults, that is, we can exclude faults for which there is a mismatch between simulated and measured diagnostic measurements. If only one candidate is found, then we can proceed to the next step which is PFA. Otherwise, if there are ambiguity groups, then additional measurements are defined, for example by adding probes into the chip, and the defect simulator is invoked again but this time starting with the reduce fault model containing only the candidate faults found in the first iteration. The procedure is repeated, where in each iteration the ambiguity groups are progressively resolved and a progressively smaller fault model is used, until only one fault candidate is left.

#### **4.6 Machine learning-based diagnosis**

The machine learning concept is well-suited for performing diagnosis. Applications of machine learning in the context of diagnosis include: (a) defect filter development for the automatic identification of the fault type that has occurred, i.e., parametric or catastrophic, so as to launch the appropriate fault diagnosis procedure; (b) parametric fault diagnosis based on regression modeling; (c) using classifiers to establish the mapping between diagnostic measurement patterns and fault classes in fault dictionary-based diagnosis. These applications will be discussed in detail in Section 5. We will show that a complete diagnosis flow can be designed based on an ensemble of different machine learning models.



**Fig. 6** Diagnosis flow based on machine learning.

## 5 Machine learning support

### 5.1 A unified diagnosis flow based on machine learning

A unified diagnosis flow based on machine learning is illustrated in Fig. 6 [60]. Devices subjected to diagnosis can be devices failing during post-manufacturing testing and customer returns, i.e., devices that fail in the field of application.

The diagnosis starts by obtaining the diagnostic measurements on the failed device. The diagnostic measurements are typically pre-selected and have been also simulated on the circuit netlist beforehand under different conditions, i.e., process variations and fault injection. Examples of diagnostic measurements include standard specification-based tests used in post-manufacturing testing, circuit responses to specially crafted test stimuli, DfT-assisted measurements at the sub-block level, and measurement of voltages or currents on internal nodes, which are performed on the actual device using probing. The available set of diagnostic measurements may show some redundancy, i.e., the same diagnostic resolution and accuracy can be achieved with a shorter diagnosis cycle time using a compacted set of diagnostic measurements. Diagnostic measurement extraction and selection will be discussed in more detail in Section 5.2.

The diagnostic measurement pattern is first processed through a defect filter whose function is to automatically identify the fault type, i.e., parametric or catastrophic.

The design of a defect filter using non-parametric kernel density estimation (KDE) will be discussed in detail in Section 5.3.

Thereafter, the appropriate diagnosis procedure is applied. For parametric faults, the flow uses regression functions that map the diagnostic measurements to low-level process parameter values, circuit macro-model parameter values, or circuit component values. Based on the predicted values, the parametric fault is quantified and can be localized or attributed to process variations. Parametric fault diagnosis based on regression modeling will be discussed in more detail in Section 5.4. For catastrophic faults, the flow uses the fault dictionary approach where the mapping between diagnostic measurements and fault classes is established based on multi-class classifiers. This approach will be discussed in more detail in Section 5.5.

The central components of the diagnosis flow, i.e. defect filter, regression models, and multi-class classifiers, are learning machines which are tuned in a pre-diagnosis learning phase. Diagnostic measurement extraction and selection can also make use of machine learning algorithms.

## 5.2 Diagnostic measurement extraction and selection

The diagnosis success indicators, i.e., resolution, accuracy, and diagnosis cycle time, are largely dependent on the selection of diagnostic measurements. This makes the selection of diagnostic measurements a critical step. The primary goal is to identify diagnostic measurements that make the effect of the different faults separable so as to reduce the ambiguity groups. The standard specification-based tests typically cannot provide the required resolution. Therefore, it is needed to identify additional and alternative diagnostic measurements obtained at the circuit outputs or at sub-block level by monitoring internal nodes.

In machine learning terminology, diagnostic measurements serve as features or inputs to the machine learning model, and diagnostic measurement identification is called feature extraction. The extraction of appropriate features is key for machine learning model construction as it determines the learning capacity and the lower bound of the generalization error. In regression modelling used for parametric fault diagnosis, the goal is to select features that correlate well with the parameters that are being predicted. In classification used for catastrophic fault diagnosis, the goal is to select features such that devices with different catastrophic faults are separable in the feature space, i.e., we can allocate hyper-dimensional boundaries in the feature space that clearly separate the clusters of different catastrophic faults such that they do not overlap.

### 5.2.1 Extraction

Feature extraction involves test stimuli generation, defining test responses on the circuit output or internal nodes, and then extracting features on the measured test

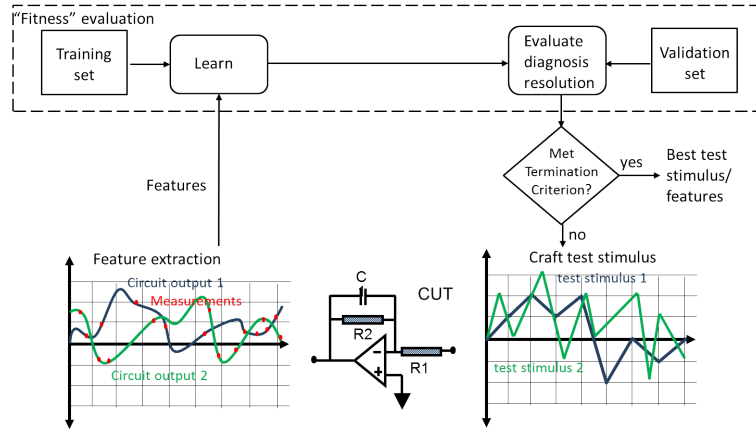


Fig. 7 Feature optimization by involving the machine learning model.

responses. For analog circuits feature extraction is a circuit-specific problem and, although there exists generic feature extraction approaches applicable to many circuit classes, better diagnosis results can be achieved if we incorporate *a priori* domain knowledge, i.e., features tailored to the particularities of the circuit under diagnosis.

For analog circuits, standard test stimuli include sinusoidals, multi-tone sinusoidals, ramp signals, etc., and standard feature extraction involves performing DC probing, output sampling, and post-processing using Fast Fourier Transform (FFT).

The key is that the diagnosis practitioner thinks out of the box as non-conventional features can prove to be very relevant. In this regard analog circuits offer more possibilities compared to their digital counterparts. On the test stimulus side, non-conventional signal waveforms, such as white noise [99], chirp signals [92], piecewise linear signals [120], applied also on nodes other than the circuit's inputs, i.e., bias nodes, power supply node, etc., identification curves [35], and power supply ramping [98] have shown to be very efficient as they are rich in frequencies and can excite the circuit across its bandwidth. On the test response side, we can consider various post-processing algorithms, i.e., Principal Component analysis (PCA) [21, 128], Wavelet transform [18, 9], etc., statistical features [129, 121], i.e. variance, entropy, kurtosis, etc., or simple feature construction by associating tests [44, 115], i.e. subtracting tests. One can also rely on dedicated on-chip test structures for extracting useful features. For example, one can rely on the SymBIST principle [84], Process Control Monitors (PCMs) at die-level [132, 2, 3, 28, 133], amplitude detectors [119], current sensors [29], digital signatures of analog waveforms [14], etc. Another approach is to optimize the features towards a low generalization error for the machine learning models [120, 6]. For example, we can consider a parameterizable input test stimulus and craft it using a gradient descent approach where evaluating the fitness in each step involves learning and evaluation of the diagnosis resolution, as shown in Fig. 7.

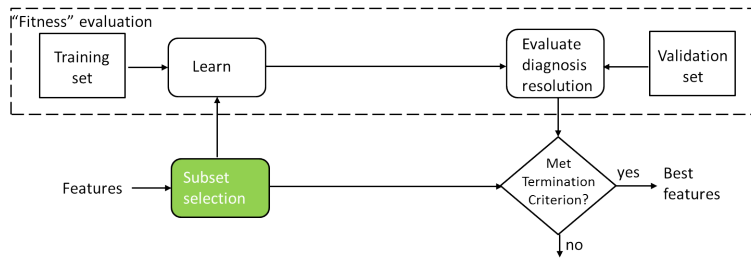


Fig. 8 Wrapper method for feature selection.

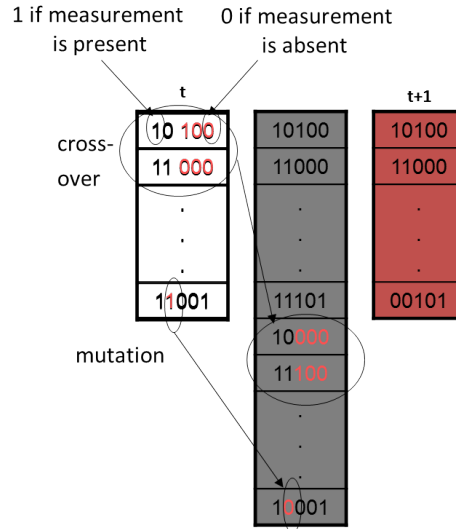
### 5.2.2 Selection

Very often there is no clear rationale on how to choose features. The reason is the large number of process parameters and their intricate interactions which makes impossible to foresee the correlation between features on one hand and circuit parameters or fault classes on the other hand. The brute-force approach is to extract as many features as possible in an *ad hoc* fashion and then, in a subsequent step, select the most relevant and eliminate redundancy. In machine learning terminology, this procedure is called feature selection.

Reducing the set of features is desired for two reasons. First, a smaller set of features means smaller on-chip resources and faster diagnosis cycle time. Second, the volume of the training data required to train the defect filter, regression models, and classifiers, increases with the dimensionality of the feature space. The higher the dimensionality of the feature space is, the more training data is needed to span the feature space and improve learning. By using many features we may run into the problem of the curse of dimensionality. In particular, for a given training set size, by increasing the dimensionality of the feature space the error on the training set decreases. The generalization error initially decreases too but after some dimensionality it starts increasing. The underlying reason is that the training data tend to have a larger amount of surface area when projected in higher dimensional spaces, meaning they become more sparse. Since the density of training samples in a given neighborhood decreases, it implies that learning in this neighborhood becomes less accurate or even random. Typically, the required training set size increases exponentially with the feature dimensionality. For example, consider a two-class classification problem. By increasing the feature dimensionality, the training data become sparse leaving large empty spaces between them. In this case, many classification boundaries can be allocated separating perfectly the two classes, but any new sample lying in the space empty of training data will be somewhat randomly classified.

Feature selection algorithms search in the power-set of features and select best subsets. If there are  $d$  features available, then there are  $2^d$  possible subsets and searching for the best subset is an NP-complete problem. There exist several heuristic feature selection algorithms that a diagnosis practitioner can choose from [87, 53, 7,





**Fig. 9** Feature selection using genetic algorithms.

81, 100, 108, 52, 15, 72]. They can be categorized into filter and wrapper methods. Filter methods select features regardless of the machine learning model, i.e., based on feature correlation [83]. In wrapper methods the fitness of each feature subset is evaluated using the machine learning model guiding the search, as shown in Fig. 8. The fitness is defined based on the diagnosis accuracy predicted by the machine learning model, and can also incorporate the cost for obtaining the features, i.e., on-chip resources needed, test time, etc. Popular wrapper feature selection algorithms include Genetic Algorithms (GAs) [95, 50, 37] and floating search algorithms [89].

GAs maintain a population of chromosomes of fixed size. In our case, chromosomes are bit-strings of length equal to the number of diagnostic measurements, where the  $j$ -th bit is set to 1 if the  $j$ -th diagnostic measurement is present in the subset and 0 otherwise. As shown in Fig. 9, starting with a base population, new chromosomes are generated using the mutation and crossover operators. In crossover, parts of two different parent chromosomes are mixed to produce an offspring. In mutation, bits of a single parent chromosome are randomly perturbed to create a child. At the end of each generation, each chromosome is evaluated to determine its fitness criteria. Only the fittest chromosomes are likely to survive and breed into the next generation. GAs evolve with the juxtaposition of bit templates, quickly optimizing the target fitness criteria.

### 5.3 Defect filter

The role of the defect filter is to identify the fault type, i.e., parametric or catastrophic, based on the diagnostic measurement pattern and, thereafter, forward the device to the appropriate diagnosis tier. The defect filter is required since different diagnoses flows are used for parametric and catastrophic faults and applying the wrong flow may inadvertently lead to poor diagnosis. Thus, the defect filter enables a unified catastrophic/parametric fault diagnosis approach without needing to specify in advance the fault type.

The defect filter can be built based on an one-class classifier in the space of diagnostic measurements. The classifier allocates a boundary enclosing data from devices with process variations such that a device with a catastrophic fault has a footprint outside the boundary.

In its simplest form, a defect filter can be a hyper-rectangle [4]. However, a hyper-rectangle is a crude approximation of the area surrounding the distribution of devices with process variations. Another approach is using nonlinear guard-bands in the space of diagnostic measurements [70]. However, the correct positioning of guard-bands requires detailed information on defective devices which is not readily available. In addition, the positioning will be to some degree random due to the non-statistical nature of defective devices.

Herein, we describe in more detail a defect filter based on the joint probability density function of the diagnostic measurements [105], denoted by  $f(\mathbf{m})$ , where  $\mathbf{m} = [m_1, \dots, m_d]$  is the  $d$ -dimensional diagnostic measurement pattern.

The form of  $f(\mathbf{m})$  is unknown, thus we will estimate it using a training set, in particular using the  $d$ -dimensional diagnostic measurement patterns of  $n$  circuit instances, denoted by  $\mathbf{M}^1, \dots, \mathbf{M}^n$ . The estimate, denoted by  $\tilde{f}(\mathbf{m})$ , is derived using a non-parametric kernel density estimator defined by [96]

$$\tilde{f}(\mathbf{m}, \alpha) = \frac{1}{n} \sum_{j=1}^n \frac{1}{(\lambda_j(\alpha) \cdot h)^d} K_e \left( \frac{\mathbf{m} - \mathbf{M}^j}{\lambda_j(\alpha) \cdot h} \right), \quad (1)$$

where

$$h = \{8c_d^{-1}(d+4)(2\sqrt{\pi})^d\}^{1/(d+4)} n^{-1/(d+4)} \quad (2)$$

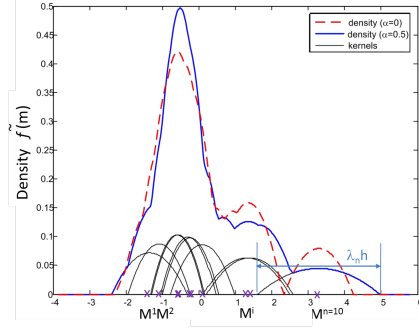
is a smoothing parameter called bandwidth,

$$c_d = 2\pi^{d/2}/(d \cdot \Gamma(d/2)) \quad (3)$$

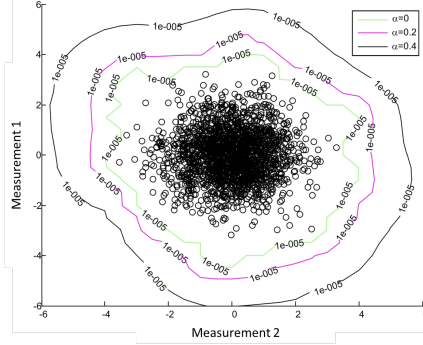
is the volume of the unit  $d$ -dimensional sphere,

$$K_e(\mathbf{t}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1 - \mathbf{t}^T \mathbf{t}) & \text{if } \mathbf{t}^T \mathbf{t} < 1 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

is the Epanechnikov kernel,  $\lambda_j$  are local bandwidth factors defined by



**Fig. 10** Non-parametric kernel density estimation.



**Fig. 11** Defect filters in a 2-dimensional diagnostic measurement space. The black circles correspond to devices with process variations which are used to construct the defect filters.

$$\lambda_j(\alpha) = \{\tilde{f}(\mathbf{M}^j, 0)/g\}^{-\alpha}, \quad (5)$$

and  $g$  is the geometric mean

$$\log g = n^{-1} \sum_{j=1}^n \log \tilde{f}(\mathbf{M}^j, 0). \quad (6)$$

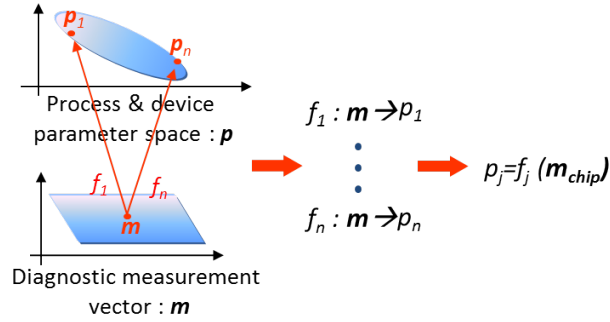
The density estimate in (1) is a weighted sum of kernels centered on the  $n$  observations, as shown with the one-dimensional example of Fig. 10. The bandwidth  $h$  defines the half-width of the kernels. The parameter  $\lambda_j(\alpha)$  multiplies the bandwidth of the kernel of the  $j$ -th observation. The default value of  $\alpha$  is  $\alpha = 0$ , resulting in  $\lambda_j(0) = 1$  for all  $n$  observations. By increasing  $\alpha$ , the tails of the density estimate become smoother and longer, but less heavier [96].

Non-parametric means that no assumption is made regarding the parametric form of  $f(\mathbf{m})$ , thus any diagnostic measurement pattern can be handled. The density function is estimated using only devices with process variations. Such a training set can be generated by a Monte Carlo simulation. The fact that no devices with catastrophic faults are required to estimate the density function makes the defect filter independent of fault dictionaries, which is an appealing property.

Noticing that the density estimate vanishes at some point, we can choose naturally to filter a device as catastrophic if its measurement pattern  $\mathbf{m}_{chip}$  lies in an area that has zero probability density, i.e., it satisfies

$$\tilde{f}(\mathbf{m}_{chip}, \alpha) = 0, \quad (7)$$

since in this case it is inconsistent with the statistical nature of the bulk of the data from devices with process variations that was used to estimate the density. The solutions to the above equation compose the frontier of the defect filter. The parameter  $\alpha$  controls the extent of the filtering. The larger  $\alpha$  is, the more “lenient” the



**Fig. 12** Parametric fault diagnosis using regression functions.

defect filter is. This is depicted in two dimensions in Fig. 11, where the frontiers of three progressively “lenient” defect filters are displayed with contours of low density.

The devices with catastrophic faults that are filtered out are forwarded to a multi-class classifier that is trained in the pre-diagnosis phase to map the diagnostic measurement pattern to the underlying catastrophic fault, as it will be described in more detail in Section 5.5. On the other hand, if  $\tilde{f}(\mathbf{m}_{chip}, \alpha) > 0$ , the device is considered to contain process variations, that is, a parametric fault has occurred. For parametric fault diagnosis, we use nonlinear inverse regression functions that are trained in the pre-diagnosis phase to map the diagnostic measurement pattern to the values of circuit parameters of interest, as it will be described in more detail in Section 5.4. The defect filter is always tuned to filter out devices with catastrophic faults. However, this could inadvertently result in some devices with parametric faults being also screened out and forwarded to the classifier. To correct this leakage, the multi-class classifier is trained during the pre-diagnosis phase to include detection of devices with process variations as well, i.e., the class of process variations is added. Thus, in the unlikely case where a device with a parametric fault is presented to a classifier, the classifier kicks it back to the regression tier.

#### 5.4 Parametric fault diagnosis based on regression modeling

Regression can be used to develop an alternative model-based diagnosis approach for parametric faults. In particular, regression can be used to approximate the functions relating the diagnostic measurement pattern to the values of circuit parameters of interest. These functions are intricate and explicit formulations using circuit equations or linear error models are difficult to derive and are valid for small parameter deviations. Regression can be used to learn these functions for the entire circuit parameter ranges using training data. Formally, for each circuit parameter  $\{p_j\}_{j=1, \dots, n}$ , we train a regression function  $f_j : \mathbf{m} \mapsto p_j, j = 1, \dots, n$ , as shown in Fig. 12 [23, 25, 60, 38]. On the failed device, we obtain the diagnostic measurement pattern  $\mathbf{m}_{chip}$  and we

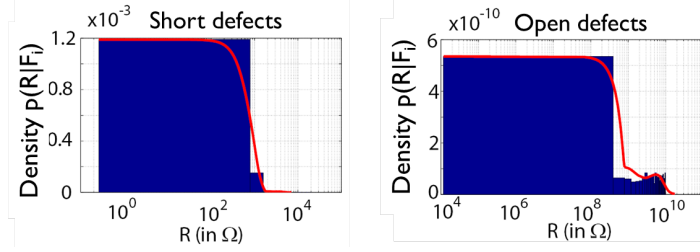
use the regression functions to predict the values of circuit parameters. This way we can diagnose off-target circuit parameter values.

This approach is similar to the alternate test principle where regression modeling is used to learn the mapping between low-cost measurements and the performances promised in the datasheet [120, 124, 5, 103, 14, 104, 10]. The goal here is to circumvent the explicit measurement of performances which is very costly as it entails long test times and sophisticated ATE. By employing a measurement pattern obtained on a single test configuration using low-cost ATE we can achieve significant test cost reduction.

Regression-based parametric fault diagnosis gives flexibility as to what circuit parameters are diagnosed. Circuit parameters can vary from low-level process parameters, i.e., threshold voltage, oxide thickness, and junction capacitances, to component values, i.e., transistor geometries, resistor values, and capacitor values, to design parameters, i.e., transistor transconductance values, gains of stages, and biasing values.

The learning phase employs a training set of devices with typical and extreme process variations on which both the circuit parameters and the diagnostic measurements are obtained. The device instances can be generated by Monte Carlo simulation and corner analysis using the statistical PDK of the technology. Real training data can also be generated on fabricated chips from various lots, wafers, and wafer sites as long as the circuit parameters can be measured. This typically requires some on-chip test infrastructure. For example, low-level process parameters can be measured using PCMs that are typically placed in the scribe lines of the wafer.

The diagnosis practitioner can use one of the several available regression tools, i.e., polynomial regression, neural networks, Multivariate Adaptive Regression Splines (MARS), Support Vector Machine (SVM) regression, etc. [21, 27, 54]. There are many regression modeling approaches to choose from and each comes with a suite of training algorithms. The main issue is choosing the tool with the right complexity. By using a simplistic regression model we may under-fit the training data, resulting also in high generalization error, while by using a more sophisticated regression modeling approach that can draw highly nonlinear approximations we may over-fit the training data, resulting inadvertently in poor generalization. To give a simple example, let us consider a set of training observations  $(x, y)$  generated according to the function  $y = ax$ . If the observations are noisy, then using a high-order polynomial model to derive an exact fit through all observations will result in zero error on the training observations, but this will result in a poor generalization on new observations, which is the main issue at stake in machine learning. In this case, a first-order polynomial would be the best choice. Most advance regression tools offer some inherent criterion and form of regularization to control the complexity of the learned model.

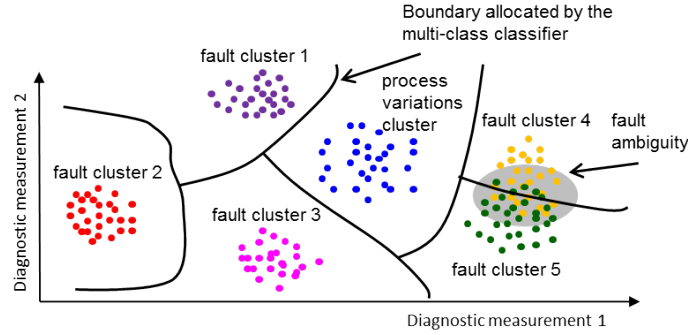


**Fig. 13** Estimated probability density function of resistance (in  $\Omega$ ) for (a) open defects and (b) short defects, plotted in logarithmic scale.

### 5.5 Fault dictionary-based diagnosis using classifiers

Classifiers offer an elegant solution for fault dictionary-based diagnosis. The first step in fault dictionary-based diagnosis is to create a list of the  $N$  most probable catastrophic short- or open-circuit fault locations, denoted by  $\{F_i\}, i = 1, \dots, N$ , for example based on IFA and historical defect data. The catastrophic faults are injected sequentially in the netlist of the device and we perform Monte Carlo simulation, where in each pass a different short or open resistance  $R$  is used. These values are sampled from the resistance distributions for short- and open-circuits, denoted by  $p(R|F_i)$ , as shown in Fig. 13 [59]. In this way, a set of  $n$  samples is created for each catastrophic fault  $i$ , denoted by  $\{F_i^j\}, j = 1, \dots, n$ . It is also possible to enhance each set with more points that represent process spread. This is recommended if we can afford the extra simulation effort. For example for each fault value  $F_i^j$  we can run a Monte Carlo simulation of the circuit using the PDK of the technology. For each value  $\{F_i^j\}$  we obtain the  $d$ -dimensional diagnostic measurement pattern denoted by  $\mathbf{m}_i^j = [m_{i,1}^j, m_{i,2}^j, \dots, m_{i,d}^j]$ . The fault cluster for fault  $F_i, i = 1, \dots, N$  is  $FC_i = \{\mathbf{m}_i^j\}, j = 1, \dots, n$ . It can be further enhanced by more samples by estimating the density function of the samples  $\{\mathbf{m}_i^j\}, j = 1, \dots, n$ , for example using non-parametric KDE [96], and, thereafter, sample it to generate an arbitrarily large number of synthetic samples [106]. In summary, the  $i$ -th fault cluster  $FC_i$  of the  $i$ -th fault,  $i = 1, \dots, N$ , consists of  $n$  samples  $\{\mathbf{m}_i^j\}, j = 1, \dots, n$  allocated in the space of diagnostic measurements. Additionally, we can define the process variation cluster  $FC_{PV}$  corresponding to circuit instances with process variations, which can be generated with Monte Carlo simulation and corner analysis where in this case there is no fault injected. The set comprising the fault clusters  $\{FC_i\}, i = 1, \dots, N$  and the process variation cluster  $FC_{PV}$  compose the fault dictionary.

Thereafter, the fault dictionary is used in the pre-diagnosis phase to train a multi-class classifier that allocates a boundary in the space of diagnostic measurements to separate one fault cluster from another. The boundary divides the diagnostic measurement space into subspaces each corresponding to a single fault, while there is also one subspace corresponding to process variations. On the failed device, we obtain the  $d$ -dimensional diagnostic measurement pattern  $\mathbf{m}_{chip} = [m_{chip}^1, m_{chip}^2, \dots, m_{chip}^d]$



**Fig. 14** Fault dictionary-based diagnosis using a multi-class classifier.

and present it to the classifier. The classifier examines where the footprint of  $\mathbf{m}_{chip}$  lies with the respect to the boundary, i.e., in which subspace it lies, and the fault corresponding to the subspace is diagnosed. If  $\mathbf{m}_{chip}$  falls into the process variations cluster, then parametric fault diagnosis is launched.

This approach is illustrated in Fig. 14 for a 2-dimensional diagnostic measurement pattern and 5 faults. As it can be seen, while faults 1-3 are clearly separable, faults 4-5 overlap in this diagnostic measurement space resulting in fault ambiguity around the boundary. The goal is that the diagnostic measurement pattern is selected so as to result in clear separation of the fault clusters. The accuracy of the fault classification defines the diagnosis resolution and accuracy.

Note that if we define a single fault cluster with circuit instances that violate one or more performance specifications, either because of catastrophic faults or process variations, and we use an additional nominal cluster with circuit instances that satisfy all performance specifications, then a fault detection classification-based system can be developed [109].

Furthermore, as suggested by practitioners in the field of pattern recognition [123, 69], the overall classification accuracy can be improved by combining the response of different classifiers [61]. Thus, instead of using a single classifier one can use an ensemble of  $c$  classifiers  $\{C_j\}$ ,  $j = 1, \dots, c$ , each trained separately, also possibly in a different diagnostic measurement space. Each classifier  $j$  assigns a score to each fault  $i$ , denoted by  $s_j(F_i)$ , instead of just making a deterministic judgment about which fault is present in the faulty device. Thereafter, the individual scores of the classifiers are combined to assign a single score to each fault  $i$ , denoted by  $s(F_i)$ . Various combination methods have been proposed in the literature, including averaging, weighted averaging, majority vote, etc. [123, 69]. The simplest is the averaging method which consists of computing the average value of scores obtained by different classifiers, i.e.,  $s(F_i) = \frac{1}{c} \sum_{j=1}^c s_j(F_i)$ . The weighted averaging method requires a validation set to assign weights to the classifiers, yet this validation set is typically not available at the time when the diagnosis tools are built. The majority vote method renders a deterministic diagnosis rather than a ranking of catastrophic

faults that are likely to have occurred. Using an ensemble of classifiers, the output of the catastrophic diagnosis flow is the ranking of the catastrophic faults according to their likelihood of occurrence in the faulty device.

Similar to regression tools for parametric fault diagnosis, there is a multitude of classifier tools one can use, i.e.,  $k$ -nearest neighbors, discrimination analysis, neural networks, SVMs, decision trees, etc. [21, 27, 54]. The selection criteria can be based on the dataset size, feature dimensionality, presence of outliers, training runtime, etc. Most importantly, the practitioner needs to understand how the classifier works and tune its user-defined parameters which may require trial and error. A good practice is to put to the test different classifiers, including advance ones that have internal regularization, and then choose the one(s) that offer the highest classification and decision confidence.

The discussed catastrophic fault diagnosis flow using classification has been explored extensively in the literature where different combinations of diagnostic measurements (including test stimuli and test response post-processing) and classifiers have been studied on different case studies [40, 30, 99, 76, 125, 23, 22, 9, 8, 98, 34, 128, 61, 130, 20, 55].

## 6 Industrial case study

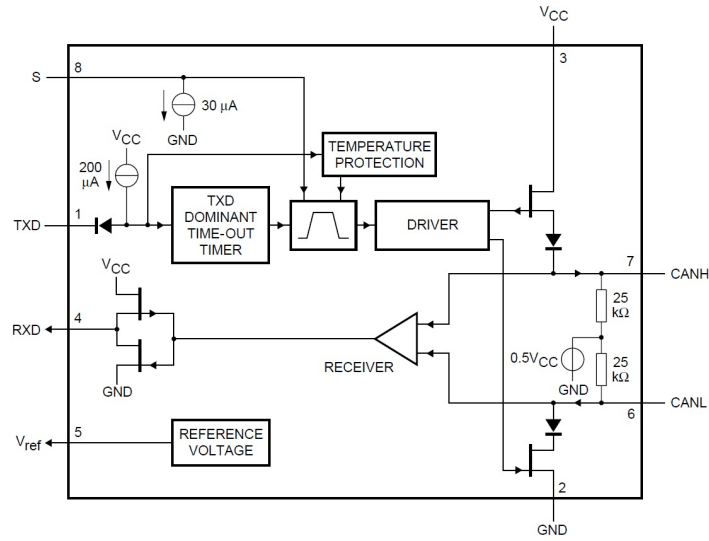
### 6.1 Device under test

We will show the application of the diagnosis flow presented in Section 5 to an industrial case study [61]. The Device Under Test (DUT) is a Controller Area Network (CAN) transceiver designed by NXP Semiconductors in a BiCMOS-DMOS process. The circuit netlist has 1032 elements of which 613 are transistors. A high-level block diagram of the circuit is shown in Fig. 15. This device is produced in high-volume and constitutes an essential part in the electronic system of automobiles. It is deployed in a safety-critical application, thus it has to meet stringent specifications and demands practically zero test escapes. Therefore, it is of vital importance to diagnose the sources of failure, in order to achieve better quality control and, when possible, improve the design such that similar failures do not emerge in the field during the lifetime of the operation.

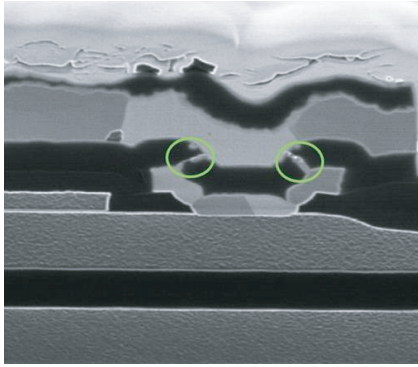
### 6.2 Real dataset

We have at hand a set of 29 devices from different lots that failed at least one of the specifications during production test. Time-consuming failure analysis was carried out for all these devices and it was observed in all cases that the cause of failure is a short-circuit defect. For example, Fig. 16 shows a Focused Ion Beam (FIB) image of the short-circuit defect observed in DUT 18 and Fig. 17 shows a Scanning Electron

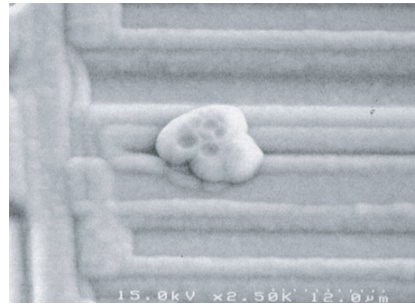




**Fig. 15** High-level block diagram of the CAN transceiver.



**Fig. 16** FIB image of the short-circuit defect diagnosed in DUT 18.



**Fig. 17** SEM image of the short-circuit defect diagnosed in DUT 26.

Microscope (SEM) image of the short-circuit defect observed in DUT 26. For the purpose of the experiment, we assume that the actual defects that have occurred in each of these devices are unknown and we set out to diagnose them by applying the diagnosis flow presented in Section 5.

### 6.3 Fault modeling

For this particular device produced in high volume under a mature technology where process variation is well understood and controlled, device failures due to parametric deviation of process and device parameters are very unlikely to occur. Furthermore, for this particular technology, open-circuit defects are less likely to occur than short-circuit defects. In fact, in analog designs, typically one has space to do via doubling, which makes open-circuit defects even less likely. As a result, more than 90% of the observed defects in production are short-circuits. Thus, only catastrophic short-circuit defects are considered for fault modeling.

### 6.4 Fault dictionary

An IFA was performed which resulted in a list of  $N=923$  probable short-circuit defects. Each short-circuit is modeled with  $n=3$  different bridge resistance values, i.e.,  $5\ \Omega$ ,  $50\ \Omega$ ,  $200\ \Omega$ . These values are chosen according to defect data characterization analysis for this particular technology. Subsequently, a total of  $3 \times 923 = 2769$  fault simulations were carried out to generate the 923 fault clusters. In this large-scale industrial case study, we cannot afford extra simulation effort to consider process variation in fault simulation. Thus, each simulation consists of inserting a short-circuit defect in the netlist with a specific bridge resistance value, while the circuit parameters are fixed at their nominal design values. In each fault simulation we collect the same diagnostic measurements.

The standard production tests for this device include digital, analog, and IDDQ tests. We consider as diagnostic measurements  $d = 97$  non-digital tests (i.e., voltage, current, timing and hysteresis measurements) which dominate the test time. Each measurement is scaled in the range  $[-1, 1]$ .

Fault simulation took approximately 12 hours. Notice that fault simulation is a one time effort. Building the diagnosis tools and performing the diagnosis of a faulty device takes only a few minutes.

### 6.5 Missing values problem

In this real-world case study, the injection of a defect in the device netlist might render the system of equations during circuit simulation unsolvable. Therefore, it is highly likely that there exist diagnostic measurements that are unattainable for specific defects and specific resistance values. The problem of missing values also concerns the real diagnostic measurement pattern. Indeed, a diagnostic measurement might hit the instrument limit, in which case its value is artificially “forced” to equal the instrument limit. In this case, we can only use the pass/fail information provided by the diagnostic measurement and we should consider the absolute value as missing.

To account for missing values, the recommendations in [74] are followed. In short, missing values force us to exclude either diagnostic measurements or defects from the analysis. In the former case, we remove information that may be useful for performing diagnosis. In the latter case, we are bound to obtain misleading diagnosis results if the defect that is present in the faulty device has been inadvertently excluded from the analysis.

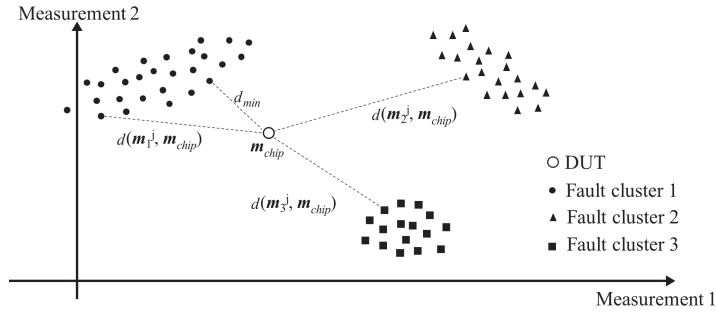
## 6.6 Classifiers

As mentioned in Section 5.5, numerous classifiers, ranging from simple to more elaborate ones, can be employed to diagnose catastrophic faults. In general, the efficiency of a classifier depends on the distribution of fault clusters and the extent to which they overlap. In the context of diagnosis, there is no solution for choosing the optimal classifier among an array of different classifiers. The reason is that classifiers can be compared only on the basis of a real validation set, but such a set is not available at the time we build the diagnosis tools. In this section, we describe in detail five different classifiers that were considered for this experiment, based on pass/fail verification, Euclidean distance, Mahalanobis distance, non-parametric KDE, and SVM. Our intention is to list a number of popular classifiers. In theory, any other classifier can be used in the same context. In addition, we show how these classifiers assign to each defect a normalized score between  $[0, 1]$ , where the highest score is given to the most probable defect. The normalized scores are combined to consider the ensemble of classifiers as opposed to using a single classifier, so as to improve the diagnosis accuracy.

### 6.6.1 Pass/fail verification

This classifier simply examines the similarity of the patterns  $\mathbf{m}_{chip}$  and  $\mathbf{m}_i^j$  by verifying the pass/fail information for each diagnostic measurement. Formally, we consider the specification indicator  $I_{j,k}^i$ , such that (a)  $I_{j,k}^i = 1$  if both  $\mathbf{m}_{chip}$  and  $\mathbf{m}_i^j$  comply with the specification of the  $k$ -th diagnostic measurement or if both  $\mathbf{m}_{chip}$  and  $\mathbf{m}_i^j$  fail the specification of the  $k$ -th diagnostic measurement and (b)  $I_{j,k}^i = 0$  if only one of  $\mathbf{m}_{chip}$  and  $\mathbf{m}_i^j$  complies with the specification of the  $k$ -th diagnostic measurement. The normalized score between  $[0,1]$  for defect  $F_i$  is defined as

$$s_1(F_i) = \frac{1}{n} \sum_{j=1}^n \frac{1}{d} \sum_{k=1}^d I_{j,k}^i. \quad (8)$$



**Fig. 18** Classifier based on Euclidean distance.

### 6.6.2 Euclidean distance

This classifier relies on the distances between the patterns  $\mathbf{m}_{chip}$  and  $\mathbf{m}_i^j$ ,  $i = 1, \dots, N$ ,  $j = 1, \dots, n$ , as illustrated in Fig. 18. The Euclidean distance is used to determine pattern proximity

$$d(\mathbf{m}_i^j, \mathbf{m}_{chip}) = \sqrt{(m_{i,1}^j - m_{chip}^1)^2 + \dots + (m_{i,d}^j - m_{chip}^d)^2}. \quad (9)$$

We define the minimum distance as

$$d_{min} = \min_{i,j} d(\mathbf{m}_i^j, \mathbf{m}_{chip}) \quad (10)$$

which allows us to scale the distances between [0,1]

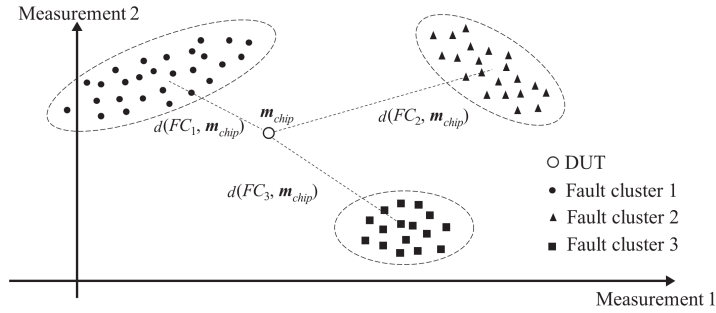
$$d'(\mathbf{m}_i^j, \mathbf{m}_{chip}) = d_{min} / d(\mathbf{m}_i^j, \mathbf{m}_{chip}). \quad (11)$$

The pattern  $\mathbf{m}_i^j$  with the shortest distance from the pattern  $\mathbf{m}_{chip}$  is mapped to 1. We assign a score to each defect  $F_i$  by computing the average normalized distance over all resistance values  $j = 1, \dots, n$

$$s_2(F_i) = \frac{1}{n} \sum_{j=1}^n d'(\mathbf{m}_i^j, \mathbf{m}_{chip}). \quad (12)$$

### 6.6.3 Mahalanobis distance

This classifier considers the Mahalanobis distance between the pattern  $\mathbf{m}_{chip}$  and each fault cluster  $FC_i$ ,  $i = 1, \dots, N$ . As shown in Fig. 19, this form of distance represents the difference between the pattern  $\mathbf{m}_{chip}$  and the mean of the fault cluster  $FC_i$ , normalized by the within-cluster covariance which is a measure of the spread of the cluster around the center of its mass



**Fig. 19** Classifier based on Mahalanobis distance.

$$d(FC_i, \mathbf{m}_{chip}) = \sqrt{(\mathbf{m}_{chip} - \mathbf{u}_i)^T \times S_i^{-1} \times (\mathbf{m}_{chip} - \mathbf{u}_i)}, \quad (13)$$

where

$$\mathbf{u}_i = [u_{i,1}, \dots, u_{i,d}] \quad (14)$$

is the mean vector with

$$u_{i,k} = \frac{1}{n} \sum_{j=1}^n m_{i,k}^j, \quad (15)$$

$S_i$  is the covariance matrix

$$S_i = \begin{bmatrix} s_{11}^i & \dots & s_{1d}^i \\ \dots & \ddots & \dots \\ s_{d1}^i & \dots & s_{dd}^i \end{bmatrix} \quad (16)$$

with

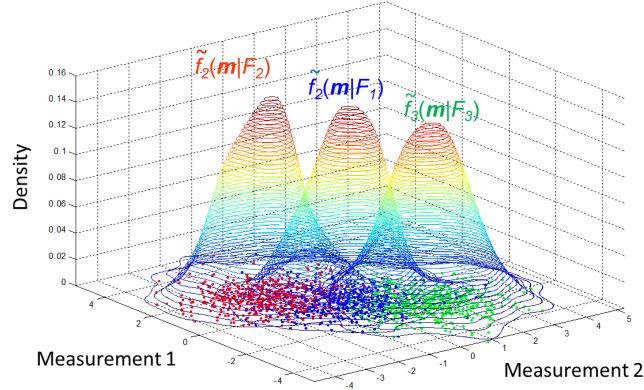
$$s_{k,\ell}^i = \frac{1}{n} \sum_{j=1}^n (m_{i,k}^j - u_{j,k})(m_{i,\ell}^j - u_{j,\ell}). \quad (17)$$

This method favors fault clusters for which the distance between their center of mass and the pattern  $\mathbf{m}_{chip}$  is small and penalizes fault clusters for which this distance is large compared to their spread. By defining the minimum distance as

$$d_{min} = \min_i d(FC_i, \mathbf{m}_{chip}), \quad (18)$$

we assign a score to each defect  $F_i$  between  $[0,1]$

$$s_3(F_i) = d_{min}/d(FC_i, \mathbf{m}_{chip}). \quad (19)$$



**Fig. 20** Classifier based on non-parametric kernel density estimation.

#### 6.6.4 Non-parametric kernel density estimation

According to the Bayes' theorem, the *posterior* probability that a faulty DUT with pattern  $\mathbf{m}_{chip}$  contains fault  $F_i$  is expressed as

$$P(F_i|\mathbf{m}_{chip}) = \frac{f_i(\mathbf{m}_{chip}|F_i)P(F_i)}{p(\mathbf{m}_{chip})}, \quad (20)$$

where  $P(F_i)$  is the *prior* probability of defect  $F_i$ ,  $f_i(\mathbf{m}_{chip}|F_i)$  is the conditional joint probability density function of  $\mathbf{m}_{chip}$  given the presence of defect  $F_i$  (also called the likelihood), and  $p(\mathbf{m}_{chip})$  is the probability density function of  $\mathbf{m}_{chip}$ . A faulty DUT will most likely contain defect  $F_m$  if

$$P(F_m|\mathbf{m}_{chip}) > P(F_i|\mathbf{m}_{chip}), \quad \forall i \neq m. \quad (21)$$

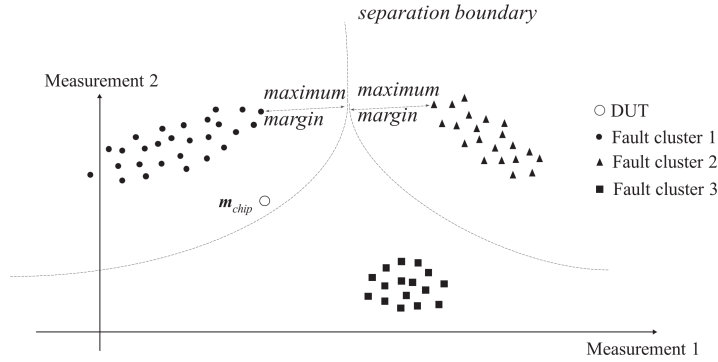
Combining (20) and (21), the above inequality becomes

$$f_m(\mathbf{m}_{chip}|F_m)P(F_m) > f_i(\mathbf{m}_{chip}|F_i)P(F_i), \quad \forall i \neq m. \quad (22)$$

The *prior* probabilities of faults can be extracted from IFA. Here, for the purpose of simplicity and without loss of generality, we assume that they are equal. Under this scenario, a faulty DUT will most likely contain defect  $F_m$  if

$$f_m(\mathbf{m}_{chip}|F_m) > f_i(\mathbf{m}_{chip}|F_i), \quad \forall i \neq m. \quad (23)$$

This method relies on the estimation of the densities  $f_i(\mathbf{m}|F_i)$ ,  $i = 1, \dots, N$  using the available observations  $\mathbf{m}_i^j$ ,  $j = 1, \dots, n$ , contained in the  $i$ -th fault cluster  $FC_i$ . To estimate  $f_i(\mathbf{m}|F_i)$  we use non-parametric KDE, as explained in Section 5.3, so as not to make any assumption regarding its parametric form, e.g., Gaussian, and allow



**Fig. 21** Classifier based on SVM.

the observations to speak for themselves [96]. Fig. 20 shows the estimated densities  $\tilde{f}_i(\mathbf{m}|F_i)$  for three defects in a 2-dimensional diagnostic measurement space.

Given a DUT with pattern  $\mathbf{m}_{chip}$ , we assign a normalized score between  $[0,1]$  to each defect

$$s_4(F_i) = \frac{\tilde{f}_i(\mathbf{m}_{chip}|F_i) - \tilde{f}_{min}}{\tilde{f}_{max} - \tilde{f}_{min}}, \quad (24)$$

where

$$\tilde{f}_{min} = \min_i \tilde{f}_i(\mathbf{m}_{chip}|F_i) \quad (25)$$

$$\tilde{f}_{max} = \max_i \tilde{f}_i(\mathbf{m}_{chip}|F_i). \quad (26)$$

As before, the defect that achieves the highest density  $\tilde{f}_i(\mathbf{m}_{chip}|F_i)$  is mapped to 1. Furthermore, if  $s_4(F_i)$  is zero for every defect  $i$ , then the pattern  $\mathbf{m}_{chip}$  is considered to be “foreign” to all fault clusters. In this case, we can conclude that the fault that has occurred had not been modeled in the fault dictionary. Thus, unlike the other methods that always assign a score to each fault, the non-parametric KDE method is the only one that in theory can identify an “unexpected” fault. This is a very appealing attribute of this classifier.

### 6.6.5 Support vector machine

This classifier aims to allocate nonlinear boundaries in the space of diagnostic measurements to separate the  $N$  fault clusters. In particular, we use SVMs [33] to learn the boundaries that traverse the middle of the Euclidean distance between the  $N$  fault clusters. This is shown in Fig. 21 for a 2-dimensional diagnostic measurement space.

The SVM classifier was originally used to solve binary classification problems. For multi-class classification with  $N$  fault clusters ( $N > 2$ ), we can reduce the problem into either  $\binom{N}{2}$  or  $N$  distinct binary classification problems and apply either the “one-against-one” or the “one-against-all” strategies. Experiments on large problems show that the “one-against-one” strategy is more suitable for practical use [58]. In this approach, the classification is carried out by a max-wins voting strategy, where each binary classifier assigns the DUT to one of two fault clusters, then the vote for the assigned fault cluster is increased by one vote, and finally the fault cluster with the largest number of votes determines the fault cluster to which the DUT belongs to.

This method assigns normalized scores between  $[0,1]$  to each defect according to

$$s_5(F_i) = N_i / N_{max}, \quad (27)$$

where  $N_i$  denotes the number of classifiers that assign the pattern  $\mathbf{m}_{chip}$  to defect  $F_i$  and

$$N_{max} = \max_i N_i. \quad (28)$$

## 6.7 Diagnosis results

Table 1 shows the 5 most highly ranked defects according to their scores for each of the 29 failed devices. The first column shows the DUT number, the second column shows the actual defect that is present, the third column shows the ranking of defects, and the fourth column shows the corresponding (rounded) final scores. As it can be observed from Table 1, the proposed method diagnoses correctly 17 out of the 29 failed devices with the true defect matching with the first choice and for 4 failed devices the true defect appears in the first three choices. In some cases the ranking indicates with high confidence the location of the defect. For example, for DUT 2, the five defects that come first in the ranking (e.g. 320, 341, 126, 374, 111) are short-circuits across nodes of a transistor pair. The ranking of these defects can be subsequently used to speed up a classical FA method by placing the emphasis on the locations of the chip where the defect has probably occurred.

By comparing the diagnosis predictions to the true defect existing in each DUT, we identify the defects that we are unable to diagnose. We were unable to diagnose correctly defects 21, 28, 156, 300, 376, 380, and in one case defect 101. Furthermore, in some cases the true defects are not ranked as the first priority, such as the cases of DUT 3, 9, 14, and 18. The reason for the above fault ambiguities is that there are different defects whose patterns tend to overlap in the diagnostic measurement space. In other words, the impact of these defects on the diagnostic measurements is very similar. Fault ambiguity can be observed as early as in the fault simulation phase. To resolve fault ambiguity additional diagnostic measurements need to be considered.



**Table 1** Diagnosis results.

DUT	True defect	Defect ranking	Normalized scores
1	107	<b>107</b> 90 920 114 347	0.924 0.923 0.923 0.923 0.923
2	320	<b>320</b> 341 126 374 111	0.948 0.867 0.833 0.827 0.822
3	125	47 616 <b>125</b> 681 360	0.914 0.839 0.838 0.837 0.837
4	101	<b>101</b> 117 459 50 388	0.831 0.829 0.826 0.817 0.817
5	216	<b>216</b> 666 192 516 120	0.831 0.795 0.792 0.788 0.785
6	300	524 608 744 294 789	0.900 0.890 0.862 0.855 0.850
7	20	<b>20</b> 126 24 27 111	0.889 0.866 0.862 0.850 0.849
8	27	<b>27</b> 111 126 446 341	0.891 0.856 0.837 0.834 0.834
9	104	111 <b>104</b> 465 721 126	0.848 0.844 0.839 0.823 0.822
10	21	310 682 524 789 608	0.867 0.858 0.855 0.855 0.851
11	101	<b>101</b> 117 459 50 388	0.831 0.829 0.826 0.818 0.817
12	19	<b>19</b> 541 106 562 595	0.810 0.794 0.780 0.780 0.780
13	19	<b>19</b> 541 562 595 106	0.799 0.791 0.788 0.771 0.771
14	140	401 <b>140</b> 457 40 919	0.936 0.912 0.911 0.910 0.910
15	20	<b>20</b> 24 126 27 111	0.887 0.865 0.862 0.853 0.849
16	101	<b>101</b> 117 459 50 388	0.831 0.829 0.826 0.817 0.817
17	107	<b>107</b> 90 920 114 347	0.924 0.923 0.923 0.923 0.923
18	31	117 <b>31</b> 50 388 622	0.901 0.888 0.882 0.881 0.880
19	101	252 305 366 363 31	0.883 0.857 0.846 0.844 0.843
20	19	<b>19</b> 541 106 562 595	0.821 0.794 0.793 0.780 0.780
21	156	524 608 744 789 682	0.903 0.893 0.872 0.872 0.866
22	20	<b>20</b> 126 24 27 111	0.882 0.870 0.867 0.864 0.853
23	107	<b>107</b> 90 920 114 347	0.924 0.923 0.923 0.923 0.923
24	22	<b>22</b> 19 541 338 106	0.826 0.808 0.808 0.795 0.795
25	107	<b>107</b> 90 920 114 347	0.924 0.923 0.923 0.923 0.923
26	380	666 192 516 676 457	0.910 0.906 0.905 0.904 0.903
27	376	383 456 112 34 196	0.924 0.920 0.830 0.826 0.824
28	28	666 192 516 355 676	0.910 0.907 0.898 0.896 0.896
29	300	524 608 744 475 215	0.896 0.896 0.866 0.864 0.862

## 7 Conclusion and discussion

Diagnosis for analog circuits is still a manual, tedious, and time-consuming procedure in industry, unlike digital circuits for which a large degree of automation has been achieved and there are available EDA tools. Apart from modern analog defect simulators, which can be used to speed up and automate iterative fault simulations during the diagnosis procedure, all other tasks in a diagnosis flow remain manual. Identification of diagnostic measurements for analog circuits, including test stimulus generation and feature extraction on test responses, is a circuit-specific problem and there is no standardized approach. However, the practitioner can borrow ideas from the literature for literally every analog circuit class. To improve diagnosis resolution, diagnostic measurements need also to be derived at sub-block level requiring DfT support, yet analog DfT infrastructures are still *ad-hoc*. Another challenge is the definition of the analog fault model. The analog fault universe is infinite increasing prohibitively fault simulation time and entailing a risk that the actual fault that

has occurred is unmodeled. On the algorithmic side, machine learning can provide support to develop all the required diagnosis tools, i.e., for fault type identification and for fault localization and quantification. There is a vast literature on diagnosis algorithms using different machine learning models. Machine learning has shown to be successful in a number of tasks and diagnosis is not an exemption. Machine learning can be used to train a system that predicts the occurring fault from a set of features, i.e., diagnostic measurements, on the faulty device. Thus, the problem is two-fold: identify effective diagnostic measurements in terms of fault prediction capability and cost, i.e., area overhead of required on-chip test resources and test time, and train the system, which comprises density estimation, regression, and classification, all standard learning problems for which numerous tools are readily available. Machine learning algorithms can help circumvent to a large extent the manual diagnosis procedures and have shown to be effective in mining information into the fault dictionaries to resolve ambiguity groups and improve the success rate of the subsequent PFA. It is up to the EDA industry now to adopt these algorithms, automate their execution, and showcase them on real case studies.

**Acknowledgements** This work was supported by the ANR EDITSoc project under Grant ANR-17-CE24-0014-02.

## References

1. Abdallah, L., Stratigopoulos, H.G., Mir, S., Altet, J.: Defect-oriented non-intrusive RF test using on-chip temperature sensors. In: Proc. IEEE VLSI Test Symposium (2013)
2. Abdallah, L., Stratigopoulos, H.G., Mir, S., Kelma, C.: Experiences with non-intrusive sensors for RF built-in test. In: Proc. IEEE International Test Conference (2012). Paper 17.1
3. Ahmadi, A., Stratigopoulos, H.G., Huang, K., Nahar, A., Orr, B., Pas, M., Carulli, J.M., Makris, Y.: Yield forecasting across semiconductor fabrication plants and design generations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **36**(12), 2120–2133 (2017)
4. Akbay, S.S., Chatterjee, A.: Fault-based alternate test of RF components. In: IEEE International Conference on Computer Design, pp. 518–525 (2007)
5. Akbay, S.S., Halder, A., Chatterjee, A., Keezer, D.: Low-cost test of embedded RF/Analog/Mixed-signal circuits in SOPs. *IEEE Transactions on Advanced Packaging* **27**(2), 352–363 (2004)
6. Akbay, S.S., Torres, J.L., Rumer, J.M., Chatterjee, A., Amtsfield, J.: Alternate test of RF front ends with IP constraints: Frequency domain test generation and validation. In: Proc. IEEE International Test Conference, pp. 4.4.1–4.4.10 (2006)
7. Alippi, C., Catelani, M., Fort, A., Mugnaini, M.: Automated selection of test frequencies for fault diagnosis in analog electronic circuits. *IEEE Transactions on Instrumentation and Measurement* **54**(3), 1033–1044 (2005)
8. Aminian, F., Aminian, M., H. W. Collins, J.: Analog fault diagnosis of actual circuits using neural networks. *IEEE Transactions on Instrumentation and Measurement* **51**(3), 544–550 (2002)
9. Aminian, M., Aminian, F.: A modular fault-diagnosis system for analog electronic circuits using neural networks with wavelet transform as a preprocessor. *IEEE Transactions on Instrumentation and Measurement* **56**(5), 1546–1554 (2007)

10. Ayari, H., Azais, F., Bernard, S., Compte, M., Renovell, M., Kerzerho, V., Potin, O., Kelma, C.: Smart selection of indirect parameters for dc-based alternate RF IC testing. In: Proc. IEEE VLSI Test Symposium, pp. 19–24 (2012)
11. Azais, F., Bernard, S., Bertrand, Y., Renovell, M.: Optimizing sinusoidal histogram test for low cost ADC BIST. *Journal of Electronic Testing: Theory and Applications* **17**(3-4), 255–266 (2001)
12. Bandler, J.W., Salama, A.E.: Fault diagnosis of analog circuits. *Proceedings of the IEEE* **73**(8), 1279–1325 (1985)
13. Barragan, M., Alhakim, R., Stratigopoulos, H.G., Dubois, M., Mir, S., Gall, H.L., Bhargava, N., Bal, A.: A fully-digital BIST wrapper based on ternary test stimuli for the dynamic test of a 40nm CMOS 18-bit stereo audio  $\Sigma\Delta$  ADC. *IEEE Transactions on Circuits and Systems I: Regular Papers* **63**(11), 1876–1888 (2016)
14. Barragán, M.J., Fiorelli, R., Léger, G., Rueda, A., Huertas, J.L.: Alternate test of LNAs through ensemble learning of on-chip digital envelope signals. *Journal of Electronic Testing: Theory and Applications* **27**(3), 277–288 (2011)
15. Barragan, M.J., Léger, G.: A procedure for alternate test feature design and selection. *IEEE Design & Test* **32**(1), 18–25 (2015)
16. Barragan, M.J., Stratigopoulos, H.G., Mir, S., Gall, H.L., Bhargava, N., Bal, A.: Practical simulation flow for evaluating analog/mixed-signal test techniques. *IEEE Design & Test* **33**(6), 46–54 (2016)
17. Benware, B., Schuermyer, C., Sharma, M., Herrmann, T.: Determining a failure root cause distribution from a population of layout-aware scan diagnosis results. *IEEE Design & Test of Computers* **29**(1), 8 – 18 (2012)
18. Bhunia, S., Roy, K.: A novel wavelet transform-based transient current analysis for fault detection and localization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **13**(4), 503–507 (2005)
19. Binu, D., Kariyappa, B.: A survey on fault diagnosis of analog circuits: Taxonomy and state of the art. *AEU - International Journal of Electronics and Communications* **73**, 68–83 (2017)
20. Binu, D., Kariyappa, B.S.: RideNN: A new rider optimization algorithm-based neural network for fault diagnosis in analog circuits. *IEEE Transactions on Instrumentation and Measurement* **68**(1), 2–26 (2019)
21. Bishop, C.M.: *Neural Networks for Pattern Recognition*. Oxford University Press (1995)
22. Catelani, M., Fort, A.: Fault diagnosis of electronic analog circuits using a radial basis function network classifier. *Measurement* **28**(3), 147–158 (2000)
23. Chakrabarti, S., Cherubal, S., Chatterjee, A.: Fault diagnosis for mixed-signal electronic systems. In: Proc. IEEE Aerospace Conference, pp. 169–179 (1999)
24. Chatterjee, A.: Concurrent error detection and fault-tolerance in linear analog circuits using continuous checksums. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **1**(2), 138–150 (1993)
25. Chatterjee, S., Chatterjee, A.: Test generation based diagnosis of device parameters for analog circuits. In: Design, Automation & Test in Europe Conference and Exhibition, pp. 596–602 (2001)
26. Chen, T., Jin, X., Geiger, R.L., Chen, D.: USER-SMILE: Ultrafast stimulus error removal and segmented model identification of linearity errors for ADC built-in self-test. *IEEE Transactions on Circuits and Systems I: Regular Papers* **65**(7), 2059–2069 (2018)
27. Cherkassky, V., Mulier, F.: *Learning from Data*. John Wiley & Sons (1998)
28. Cilici, F., Barragan, M.J., Mir, S., Lauga-Larroze, E., Bourdel, S.: Assisted test design for non-intrusive machine learning indirect test of millimeter-wave circuits. In: IEEE European Test Symposium (2018)
29. Cimino, M., Lapuyade, H., Deval, Y., Taris, T., Bégueret, J.B.: Design of a 0.9V 2.45 GHz self-testable and reliability-enhanced CMOS LNA. *IEEE Journal of Solid-State Circuits* **43**(5), 1187–1194 (2008)
30. Collins, P., Yu, S., Eckersall, K.R., Jervis, B.W., Bell, I.M., Taylor, G.E.: Application of Kohonen and supervised forced organization maps to fault diagnosis in CMOS opamps. *Electronic Letters* **30**(22), 1846–1847 (1994)

31. Cota, E.F., Negreiros, M., Carro, L., Lubaszewski, M.: A new adaptive analog test and diagnosis system. *IEEE Transactions on Instrumentation and Measurement* **49**(2), 223–227 (2000)
32. Coyette, A., Esen, B., Dobbelaere, W., Vanhooren, R., Gielen, G.: Automatic generation of test infrastructures for analog integrated circuits by controllability and observability co-optimization. *Integration, the VLSI Journal* **55**, 393–400 (2016)
33. Cristianini, N., Shawe-Taylor, J.: *Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge (2000)
34. Cui, J., Wang, Y.: A novel approach of analog circuit fault diagnosis using support vector machines classifier. *Measurement* **44**(1), 281–289 (2011)
35. Czaja, Z.: Using a square-wave signal for fault diagnosis of analog parts of mixed-signal electronic embedded systems. *IEEE Transactions on Instrumentation and Measurement* **57**(8), 1589–1595 (2008)
36. Dai, H., Souders, M.: Time-domain testing strategies and fault diagnosis for analog systems. *IEEE Transactions on Instrumentation and Measurement* **39**(1), 157–162 (1990)
37. Deb, K., Pratap, A., Agarwal, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002)
38. Devarakond, S.K., Sen, S., Bhattacharya, S., Chatterjee, A.: Concurrent device/specification cause–effect monitoring for yield diagnosis using alternate diagnostic signatures. *IEEE Design & Test of Computers* **29**(1), 48–58 (2012)
39. Dufort, B., Roberts, G.W.: On-chip analog signal generation for mixed-signal built-in self-test. *IEEE Journal of Solid-State Circuits* **34**(3), 318–30 (1999)
40. Epstein, B.R., Czigler, M., Miller, S.R.: Fault detection and classification in linear integrated circuits: An application of discrimination analysis and hypothesis testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **12**(1), 102–113 (1993)
41. Erdogan, E.S., Ozev, S.: Detailed characterization of transceiver parameters through loop-back-based BiST. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **18**(6), 901–911 (2010)
42. Erdogan, E.S., Ozev, S., Cauvet, P.: Diagnosis of assembly failures for system-in-package RF tuners. In: *IEEE International Symposium on Circuits and Systems*, pp. 2286–2289 (2008)
43. Esen, B., Coyette, A., Gielen, G., Dobbelaere, W., Vanhooren, R.: Effective DC fault models and testing approach for open defects in analog circuits. In: *Proc. IEEE International Test Conference* (2016). Paper 3.2
44. Fang, L., Lemnawar, M., Xing, Y.: Cost effective outliers screening with moving limits and correlation testing for analogue ICs. In: *Proc. IEEE International Test Conference* (2006). Paper 31.2
45. Fenton, W., McGinnity, T.M., Maguire, L.P.: Fault diagnosis of electronic systems using intelligent techniques: A review. *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews* **31**(3), 269–281 (2001)
46. Ferguson, F.J., Shen, J.P.: A CMOS fault extractor for inductive fault analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **7**(11), 1181–1194 (1988)
47. Fontana, G., Luchetta, A., Manetti, S., Piccirilli, M.C.: A fast algorithm for testability analysis of large linear time-invariant networks. *IEEE Transactions on Circuits and Systems I: Regular Papers* **64**(6), 1564–1575 (2017)
48. Gil, V.G., Arteaga, A.J.G., Léger, G.: Assessing AMS-RF test quality by defect simulation. *IEEE Transactions on Device and Materials Reliability* **19**(1), 55–63 (2019)
49. Gines, A., Léger, G.: Sigma-delta testability for pipeline A/D converters. In: *Proc. Design, Automation and Test in Europe Conference* (2014)
50. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley (1989)
51. Gomez, J., Kama, N., Coyette, A., Vanhooren, R., Dobbelaere, W., Gielen, G.: Pinhole latent defect modeling and simulation for defect-oriented analog/mixed-signal testing. In: *Proc. IEEE VLSI Test Symposium* (2020)

52. Grasso, F., Luchetta, A., Manetti, S., Piccirilli, M.C.: A method for the automatic selection of test frequencies in analog fault diagnosis. *IEEE Transactions on Instrumentation and Measurement* **56**(6), 2322–2329 (2007)
53. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* **3** (2003)
54. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer (2001)
55. He, W., He, Y., Li, B.: Generative adversarial networks with comprehensive wavelet feature for fault diagnosis of analog circuits. *IEEE Transactions on Instrumentation and Measurement* **69**(9), 6640–6650 (2020)
56. Hemink, G.J., Meijer, B.W., Kerkhoff, H.G.: Testability analysis of analog systems. *IEEE Transactions on Computer-Aided Design* **9**(6), 573–583 (1990)
57. Holst, S., Wunderlich, H.: Adaptive debug and diagnosis without fault dictionaries. In: *IEEE European Test Symposium*, pp. 7–12 (2007)
58. Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks* **13**(2), 415–425 (2002)
59. Huang, K., Stratigopoulos, H.G., Mir, S.: Bayesian fault diagnosis of RF circuits using nonparametric density estimation. In: *Proc. IEEE Asian Test Symposium*, pp. 295–298 (2010)
60. Huang, K., Stratigopoulos, H.G., Mir, S.: Fault diagnosis of analog circuits based on machine learning. In: *Proc. Design, Automation & Test in Europe Conference*, pp. 1761–1766 (2010)
61. Huang, K., Stratigopoulos, H.G., Mir, S., Hora, C., Xing, Y., Kruseman, B.: Diagnosis of local spot defects in analog circuits. *IEEE Transactions on Instrumentation and Measurement* **61**(10), 2701–2712 (2012)
62. Huang, Q., Fang, C., Mittal, S., Blanton, R.D.: Towards smarter diagnosis: A learning-based diagnostic outcome previewer. *ACM Transactions on Design Automation of Electronic Systems*, **25**(5) (2020)
63. Huang, Y.C., Hsieh, H.H., Lu, L.H.: A built-in self-test technique for RF low-noise amplifiers. *IEEE Transactions on Microwave Theory and Techniques* **56**(2), 1035–1042 (2008)
64. Huertas, G., Vázquez, D., Peralfás, E.J., Rueda, A., Huertas, J.L.: Testing mixed-signal cores: A practical oscillation-based test in an analog macrocell. *IEEE Design & Test of Computers* **19**(6), 73–82 (2002)
65. Huertas, J.L., Rueda, A., Vasquez, D.: Testable switched-capacitor filters. *IEEE Journal of Solid-State Circuits* **28**(7), 719–724 (1993)
66. Ince, M., Yilmaz, E., Fu, W., Park, J., Nagaraj, K., Winemberg, L., Ozev, S.: Digital built-in self-test for phased locked loops to enable fault detection. In: *IEEE European Test Symposium* (2019)
67. Kolarik, V., Mir, S., Lubaszewski, M., Courtois, B.: Analog checkers with absolute and relative tolerances. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **14**(5), 607–612 (1995)
68. Kook, S., Banerjee, A., Chatterjee, A.: Dynamic specification testing and diagnosis of high-precision sigma-delta ADCs. *IEEE Design & Test of Computers* **30**(4), 36–48 (2013)
69. Kuncheva, L.: “Fuzzy” versus “nonfuzzy” in combining classifiers designed by Boosting. *IEEE Transactions on Fuzzy Systems* **11**(6), 729–741 (2003)
70. Kupp, N., Drineas, P., Slamani, M., Makris, Y.: On boosting the accuracy of non-RF to RF correlation-based specification test compaction. *Journal of Electronic Testing: Theory and Applications* **25**(6), 309–321 (2009)
71. Laraba, A., Stratigopoulos, H.G., Mir, S., Naudet, H.: Exploiting pipeline ADC properties for a reduced-code linearity test technique. *IEEE Transactions on Circuits and Systems I: Regular Papers* **62**(10), 2391–2400 (2015)
72. Larguech, S., Azais, F., Bernard, S., Comte, M., Kerzérho, V., Renovell, M.: Efficiency evaluation of analog/RF alternate test: Comparative study of indirect measurement selection strategies. *Microelectronics Journal* **46**(11), 1091–1102 (2015)
73. Le-Gall, H., Alhakim, R., Valka, M., Mir, S., Stratigopoulos, H., Simeu, E.: High frequency jitter estimator for SoCs. In: *IEEE European Test Symposium* (2015)

74. Little, R., Rubin, D.: Statistical Analysis with Missing data. 2nd Edition, John Wiley & Sons, Inc (2002)
75. Liu, F., Ozev, S., Brooke, M.: Identifying the source of BW failures in high-frequency linear analog circuits based on S-parameters measurements. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **25**(11), 2594–2605 (2006)
76. Maidon, Y., Jervis, B.W., Dutton, N., Lesage, S.: Diagnosis of multifaults in analogue circuits using multilayer perceptrons. *IEE Proceedings - Circuits, Devices & Systems* **144**(3), 149–154 (1997)
77. Malloug, H., Barragan, M.J., Mir, S.: Practical harmonic cancellation techniques for the on-chip implementation of sinusoidal signal generators for mixed-signal BIST applications. *Journal of Electronic Testing: Theory and Applications* **34**(3), 263–279 (2018)
78. Maxwell, P., Hapke, F., Tang, H.: Cell-aware diagnosis: Defective inmates exposed in their cells. In: *IEEE European Test Symposium* (2016)
79. Melis, T., Simeu, E., Auvray, E.: Automatic fault simulators for diagnosis of analog systems. In: *International Symposium on On-Line Testing and Robust System Design* (2020)
80. Mhamdi, S., Girard, P., Virazel, A., Bosio, A., Faehn, E., Ladhar, A.: Cell-aware defect diagnosis of customer returns based on supervised learning. *IEEE Transactions on Device and Materials Reliability* **20**(2), 329–340 (2020)
81. Mir, S., Lubaszewski, M., Courtois, B.: Fault-based ATPG for linear analog circuits with minimal size multifrequency test sets. *Journal of Electronic Testing: Theory and Applications* **9**(1-2), 43–57 (1996)
82. Mir, S., Lubaszewski, M., Kolarik, V., Courtois, B.: Fault-based testing and diagnosis of balanced filters. *Analog Integrated Circuits and Signal Processing* **11**(1), 5–19 (1996)
83. Papakostas, D.K., Hatzopoulos, A.A.: Correlation-based comparison of analog signatures for identification and fault diagnosis. *IEEE Transactions on Instrumentation and Measurement* **42**(4), 860–3 (1993)
84. Pavlidis, A., Faehn, E., Louërat, M.M., Stratigopoulos, H.G.: BIST-assisted analog fault diagnosis. In: *IEEE European Test Symposium* (2021)
85. Pavlidis, A., Louërat, M.M., Faehn, E., Kumar, A., Stratigopoulos, H.G.: *SymBIST*: Symmetry-based analog and mixed-signal built-in self-test for functional safety. *IEEE Transactions on Circuits and Systems I: Regular Papers* **68**(6), 2580–2593 (2021)
86. Portolan, M.: Automated testing flow: The present and the future. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **39**(10), 2952–2963 (2020)
87. Prasad, V.C., Babu, N.S.C.: Selection of test nodes for analog fault diagnosis in dictionary approach. *IEEE Transactions on Instrumentation and Measurement* **49**(6), 1289–1297 (2000)
88. Provost, B., Sánchez-Sinencio, E.: On-chip ramp generators for mixed-signal BIST and ADC self-test. *IEEE Journal of Solid-State Circuits* **38**(2), 263–273 (2003)
89. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection. *Pattern Recognition Letters* **15**, 1119–1125 (1994)
90. Rapisarda, L., Decarlo, R.A.: Analog multifrequency fault diagnosis. *IEEE Transactions on Circuits and Systems CAS-30*(4), 223–234 (1983)
91. Renaud, G., Diallo, M., Barragan, M.J., Mir, S.: Fully differential 4-V output range 14.5-ENOB stepwise ramp stimulus generator for on-chip static linearity test of ADCs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **27**(2), 281–293 (2019)
92. Sarson, P.: Test time efficient group delay filter characterization technique using a discrete chirped excitation signal. In: *Proc. IEEE International Test Conference* (2016)
93. Schroder, D., Babcock, J.: Negative bias temperature instability: Road to cross in deep sub-micron silicon semiconductor manufacturing. *Journal of applied Physics* **94**(1), 1–18 (2003)
94. Sen, N., Saeks, R.: Fault diagnosis for linear systems via multifrequency measurements. *IEEE Transactions on Circuits and Systems* **26**(7), 457–465 (1979)
95. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters* **10**, 335–347 (1989)
96. Silverman, B.W.: *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC (1986)

97. Slamani, M., Kaminska, B.: Analog circuit fault diagnosis based on sensitivity computation and functional testing. *IEEE Design & Test of Computers* **9**(1), 30–39 (1992)
98. Somayajula, S.S., Sanchez-Sinencio, E., de Gyvez, J.P.: Analog fault diagnosis based on ramping power supply current signature clusters. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing* **43**(10), 703–712 (1996)
99. Spina, R., Upadhyaya, S.: Linear circuit fault diagnosis using neuromorphic analyzers. *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing* **44**(3), 188–196 (1997)
100. Starzyk, J.A., Liu, D., Liu, Z.H., Nelson, D.E., Rutkowski, J.O.: Entropy-based optimum test points selection for analog fault dictionary techniques. *IEEE Transactions on Instrumentation and Measurement* **53**(3), 754–761 (2004)
101. Stessman, N.J., Vinnakota, B., Harjani, R.: System-level design for test of fully differential analog circuits. *IEEE Journal of Solid-State Circuits* **31**(10), 1526–1534 (1996)
102. Stratigopoulos, H.G.: Machine learning applications in IC testing. In: *Proc. IEEE European Test Symposium* (2018)
103. Stratigopoulos, H.G., Makris, Y.: Error moderation in low-cost machine-learning-based Analog/RF testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **27**(2), 339–351 (2008)
104. Stratigopoulos, H.G., Mir, S.: Adaptive alternate analog test. *IEEE Design & Test of Computers* **29**(4), 71–79 (2012)
105. Stratigopoulos, H.G., Mir, S., Acar, E., Ozev, S.: Defect filter for alternate RF test. In: *Proc. IEEE European Test Symposium*, pp. 101–106 (2009)
106. Stratigopoulos, H.G., Mir, S., Makris, Y.: Enrichment of limited training sets in machine-learning-based analog/RF test. In: *Proc. Design, Automation & Test in Europe Conference*, pp. 1668–1673 (2009)
107. Stratigopoulos, H.G., Sunter, S.: Fast Monte Carlo-based estimation of analog parametric test metrics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **33**(12), 1977–1990 (2014)
108. Stratigopoulos, H.G.D., Drineas, P., Slamani, M., Makris, Y.: Non-RF to RF test correlation using learning machines: A case study. In: *IEEE VLSI Test Symposium*, pp. 9–14 (2007)
109. Stratigopoulos, H.G.D., Makris, Y.: Non-linear decision boundaries for testing analog circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **24**(11), 1760–1773 (2005)
110. Stratigopoulos, H.G.D., Makris, Y.: An adaptive checker for the fully differential analog code. *IEEE Journal of Solid-State Circuits* **41**(6), 1421–1429 (2006)
111. Stratigopoulos, H.G.D., Makris, Y.: Concurrent detection of erroneous responses in linear analog circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **25**(5), 878–891 (2006)
112. Sunter, S., Côté, J.F., Rearick, J.: Streaming access to ADCs and DACs for mixed-signal ATPG. *IEEE Design & Test* **33**(6), 38–45 (2016)
113. Sunter, S., Jurga, K., Laidler, A.: Using mixed-signal defect simulation to close the loop between design and test. *IEEE Transactions on Circuits and Systems I: Regular Papers* **63**(12), 2313–2322 (2016)
114. Sunter, S., Roy, A.: On-chip digital jitter measurement, from megahertz to gigahertz. *IEEE Design & Test of Computers* **21**(4), 314–321 (2004)
115. Tikkanen, J., Siatkowski, S., Sumikawa, N., Wang, L.C., Abadir, M.S.: Yield optimization using advanced statistical correlation methods. In: *Proc. IEEE International Test Conference* (2014)
116. Tsividis, Y.: *Operational and Modeling of the MOS Transistor*, 2nd edn. McGraw-Hill (1999)
117. Tu, K.: Recent advances on electromigration in very-large-scale-integration of interconnects. *Journal of Applied Physics* **94**(9), 5451–5473 (2003)
118. Valdes-Garcia, A., Silva-Martinez, J., Sanchez-Sinencio, E.: On-chip testing techniques for RF wireless transceivers. *IEEE Design & Test of Computers* **23**(4), 268–277 (2006)

119. Valdes-Garcia, A., Venkatasubramanian, R., Silva-Martinez, J., Sanchez-Sinencio, E.: A broadband CMOS amplitude detector for on-chip RF measurements. *IEEE Transactions on Instrumentation and Measurement* **57**(7), 1470–1477 (2008)
120. Variyam, P.N., Cherubal, S., Chatterjee, A.: Prediction of analog performance parameters using fast transient testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **21**(3), 349–361 (2002)
121. Vasan, A.S.S., Long, B., Pecht, M.: Diagnostics and prognostics method for analog electronic circuits. *IEEE Transactions on Industrial Electronics* **60**(11), 5277–5291 (2013)
122. Veillette, B.R., Roberts, G.: On-chip measurement of the jitter transfer function of charge-pump phase-locked loops. *IEEE Journal of Solid-State Circuits* **33**(3), 483–491 (1998)
123. Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., Gelzinis, A.: Soft combination of neural classifiers: A comparative study. *Pattern Recognition Letters* **20**(4), 429–444 (1999)
124. Voorakaranam, R., Akbay, S.S., Bhattacharya, S., Cherubal, S., Chatterjee, A.: Signature testing of analog and RF circuits: Algorithms and methodology. *IEEE Transactions on Circuits and Systems - I* **54**(5), 1018–1031 (2007)
125. Wang, Z., Gielen, G., Sansen, W.: Probabilistic fault detection and the selection of measurements for analog integrated circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **17**(9), 862–872 (1998)
126. Wey, C.L.: Built-in self-test (BIST) structure for analog circuit fault diagnosis. *IEEE Transactions on Instrumentation and Measurement* **39**(3), 517–21 (1990)
127. Wey, C.L., Krishnan, S.: Built-in self-test (BIST) structures for analog circuit fault diagnosis with current test data. *IEEE Transactions on Instrumentation and Measurement* **41**(4), 535–9 (1992)
128. Xiao, Y., He, Y.: A novel approach for analog fault diagnosis based on neural networks and improved kernel PCA. *Neurocomputing* **74**(7), 1102–1115 (2011)
129. Yuan, L., He, Y., Huang, J., Sun, Y.: 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)
130. Zhang, C., He, Y., Yuan, L., Xiang, S.: Analog circuit incipient fault diagnosis method using DBN based features extraction. *IEEE Access* **6**, 23053–23064 (2018)
131. Zivkovic, V., Schaldenbrand, A.: Requirements for industrial analog fault-simulator. In: *International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, pp. 61–64 (2019)
132. Zjajo, A., Barragan, M.J., de Gyvez, J.P.: BIST method for die-level process parameter variation monitoring in analog/mixed-signal integrated circuits. In: *Proc. Design, Automation & Test in Europe Conference*, pp. 1301–1306 (2007)
133. Şandru, E.D., David, E., Kovacs, I., Buzo, A., Burileanu, C., Pelz, G.: Modeling the dependency of analog circuit performance parameters on manufacturing process variations with applications in sensitivity analysis and yield prediction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2021)