Received: 19 May 2019 | Revised: 27 October 2019 | Accepted: 19 November 2019

DOI: 10.1002/widm.1360

ADVANCED REVIEW

WIREs DATA MINING AND KNOWLEDGE DISCOVERY

WILEY

A survey of digital circuit testing in the light of machine learning

Manjari Pradhan[1] | Bhargab B. Bhattacharya[2]

[1]Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India

[2]Department of Computer Science & Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India

Correspondence
Manjari Pradhan, Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India.
Email: manjari23march@gmail.com

Abstract

The insistent trend in today's nanoscale technology, to keep abreast of the Moore's law, has been continually opening up newer challenges to circuit designers. With rapid downscaling of integration, the intricacies involved in the manufacturing process have escalated significantly. Concomitantly, the nature of defects in silicon chips has become more complex and unpredictable, adding further difficulty in circuit testing and diagnosis. The volume of test data has surged and the parameters that govern testing of integrated circuits have increased not only in dimension but also in the complexity of their correlation. Evidently, the current scenario serves as a pertinent platform to explore new test solutions based on machine learning. In this survey, we look at various recent advances in this evolving domain in the context of digital logic testing and diagnosis.

This article is categorized under:
Algorithmic Development > Structure Discovery
Technologies > Machine Learning
Technologies > Prediction

KEYWORDS

diagnosis, digital circuit, machine learning, testing

1 | INTRODUCTION

Fortified with the power of modern cloud computing and the availability of massive storage and data, machine learning (ML) stands at the crest of recent technological evolution. ML-based techniques are now being widely used in modern industry as they are capable of providing efficient solutions to complicated problems that were deemed to be unsolvable a decade ago. The scope of their applications covers almost every domain and holds promises that are limited only by human imagination. It has revolutionized the medical and drug industry through efficient diagnosis of diseases and understanding biological mechanisms such as gene regulation, which are immensely helpful in drug discovery. Similar is its redefining influence in information technology, communication, production, manufacturing, and the entertainment industry. The ubiquitous presence of ML in the technological landscape is being felt because of the need for analyzing large amount of historical data that are available for all such applications. ML provides a mechanism to transform this data into knowledge by harnessing meaningful information, which is used for future analytics and predictive solutions. The only requirement for building an ML engine is a good amount of reliable data. The success of ML in all these fields can be attributed to the quick solutions it provides because once the model has been trained from the data, the decision-making process for predicting the parameters reduces to a mere function evaluation.

https://doi.org/10.1002/widm.1360

Another significant reason behind the upthrust in ML, especially deep learning, is due to the availability of high-speed hardware and graphic processing units which expedite the required computation. In this survey, we focus on the challenges of testing digital hardware logic and discuss the role of ML in solving such problems. Although ML is nascent in this area, there are enough opportunities to explore applications of these techniques and to develop new solutions.

Digital electronics technology is around five decades old. The invention of transistors in 1954 paved the way to building miniaturized devices, which ultimately led to the design of integrated circuits (ICs) in 1962. Next, with the advent of complementary metal-oxide-semiconductor (CMOS) technology, a new circuit design paradigm had evolved with low power consumption. CMOS design styles are commonly used for implementing digital circuits with very large-scale integration (VLSI). Today's IC-chips comprise billions of transistors on a single die. Apart from designing, testing for manufacturing defects is a crucial step in the production cycle of digital IC-chips since it affects the reliability, cost, and time-to-delivery. Effective testing is also necessary to estimate the yield of the chip and to throw light on process variations. Various facets of fault modeling, detection, and diagnosis, fault simulation, built-in self-test, and design-for-testability (DfT) have been elaborately studied in the past three decades leading to efficient test generation and fault-diagnosis algorithms and testable designs (Abramovici, Breuer, & Friedman, 2002; Bushnell & Agrawal, 2005; Wang, Wu, & Wen, 2006). Over the years, a number of industrial tools have been developed for digital logic testing. However, the challenges in testing, and more so in diagnosis, have been ever increasing with the complexities of IC-chips.

As predicted by Moore's law, in today's deep-submicron technology, the number of transistors in a single IC-chip has reached the range of billions with the current technology approaching less than 7 nm. While low technology nodes reduce silicon area and increase the yield of high-speed electronic products, they demand elaborate manufacturing processes. Both the factors of intricate manufacturing processes and the magnitude of integration not only have made ICs more susceptible to malfunctioning but also introduced newer defects with more complex behavior. Thus, the benefit of miniaturization is coupled with newer challenges in testing. In order to improve the yield of IC chips, defective chips need to be identified and defect localization is to be performed so that the process can be improved in the subsequent manufacturing cycles. Thus, defect/fault diagnosis has become an integral step in IC design and fabrication, which involves identifying systematic defects (Mutschler, 2014) prior to their location. Majority of ML based digital test applications belong to the area of diagnosis. In the industrial domain, there have been some efforts to apply ML to circuit testing (Lapedus, 2019). A major bottleneck in this area is the unavailability of useful data. Additionally, unlike other fields where ML tools have been thoroughly investigated and successfully applied, their modeling in the area of digital testing is quite new and only a few studies have been reported so far in the literature. The aim of this survey is to walk through the different problem settings in the context of digital logic testing where ML-techniques have been applied and to focus on the various features and modeling approaches that have been used. Most of the problems in this area are novel and there are ample scopes for improving the effectiveness of available solutions. What is important to note is that ML provides automated tools to handle many hard test problems, which otherwise would not have been easy to tackle.

An overview of digital logic testing and ML-techniques is presented in the following two subsections. Some techniques for analog circuit testing based on ML are also briefly discussed in the background since ML was mostly used earlier for analog testing prior to its recent applications to digital test.

The rest of the paper is organized as follows. We discuss applications of ML to diagnosis in Section 2. In Section 3, implementations of ML-tools on various DfT and test applications, such as test compression, circuit testability, and timing analysis, have been reported. Finally, possible future directions and challenges of circuit testing in the perspective of ML are highlighted in Section 4.

1.1 | Overview of digital circuit testing

From a theoretical point of view, a digital circuit comprises combinational components consisting of interconnected logic gates that allow signals to flow in forward direction, sequential components that consist of memory elements called flip-flops (F/Fs), which may feed the signal both in forward and reverse directions, and input/output ports. Unlike in an analog circuit, signals in a digital circuit are discrete in nature and assume only two states denoted as “0” and “1”. A typical representation of a digital circuit is shown in Figure 1a (Abramovici et al., 2002). The F/Fs may not be directly observable or controllable, and hence they add more difficulty in testing the circuit-under-test (CUT). A

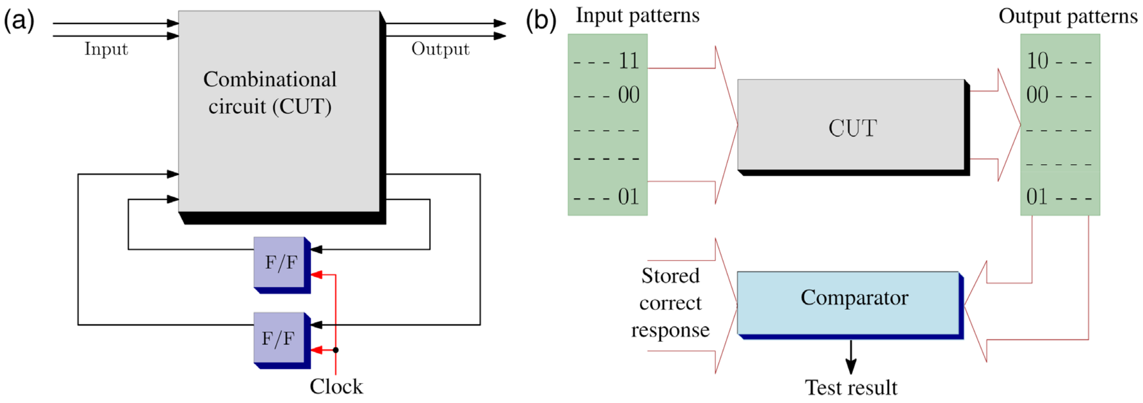

FIGURE 1 Digital circuit testing. (a) Canonical structure of a digital circuit (Abramovici et al., 2002). (b) Test application of input patterns generated by ATPG, to a chip in the ATE (Bushnell & Agrawal, 2005)

popular DfT approach called *scan-chain* is adopted to alleviate this situation, which allows a mechanism where the F/Fs are made directly controllable (observable) via a scan input (output). The F/F-outputs (inputs) are called pseudo-inputs (pseudo-outputs) of the circuit.

There are two aspects in testing: *test generation* and *test application* (Bushnell & Agrawal, 2005). For a circuit with n inputs, since the input vector space is explosive (2^n), test generation, commonly called automatic test patterns generation (ATPG), involves devising algorithms to select a set of input vectors, called a *test set*, from the input vector space so that maximum possible defects can be detected at observable outputs, on application of the test set to the inputs. For this purpose, the defects are modeled as logical *faults*. Two common fault models are widely used: stuck-at faults and transition faults. Once we have the test set, during test application, the chip is tested using ATE ("Automatic Test Equipment (ATE) and Production Test", 2006), where input patterns are applied and the output patterns are collected. They are then compared to the expected (correct) circuit outputs to decide if they pass or fail the test. If a test fails, the output response is further analyzed for diagnosis. A simple flow of test application is shown in Figure 1b.

While an ATPG targets to find tests for individual faults, diagnostic tests aim to discriminate a pair of faults f_1 and f_2. Consider a combinational circuit which produces an output function $F_i(t)$ at its ith output, for a test pattern t. Let us assume that in the presence of f_1 (f_2), the output function becomes $F_i(t)_{f_1}$ ($F_i(t)_{f_2}$). The test vector t would distinguish the fault pair, if the following Boolean equation is satisfied (Gruning, Mahlstedt, & Koopmeiners, 1991).

$$F_i(t)_{f_1} \oplus F_i(t)_{f_2} = 1. \tag{1}$$

Equation (1) implies that t should produce different output values in the presence of the two faults. This implies that only one of the two faults should be detected by t. A test pattern which can detect both the fault can distinguish the fault pair only if they are detected at two different outputs. Since a circuit may have several output ports, there are many ways the fault pair can be distinguished.

In Figure 2, if the fault f_1 (f_2) is propagated to any one output in O_1 (O_2), by t, they are distinguishable by it. However, if both the faults are propagated to O_{12}, then they should be propagated to different outputs for that to be distinguished by t. If no input vector distinguishes a fault pair then these two faults must be functionally equivalent. If the diagnostic/detection test vectors do not distinguish a fault pair, then it is called a test-equivalent fault pair. Thus, there are two aspects of diagnostic test generation, identifying distinguishable fault-pairs along with the distinguishing vector, and identifying the functionally equivalent fault pairs. Both help in improving the diagnostic test generation process.

Another important aspect of test application is test compression that is employed to reduce test cost. One of the features of test compaction environment is the response compactor (Figure 3). As shown in Figure 3, in each scan-cycle, the data from n scan chains is compacted into m bits where m is much less than n. The compactor can either be a spatial or temporal compactor.

In the case of a spatial compactor, the data per cycle is compressed. However, in a temporal compactor, which uses multiple input signature registers (MISR), response data observed over multiple scan cycles is considered for

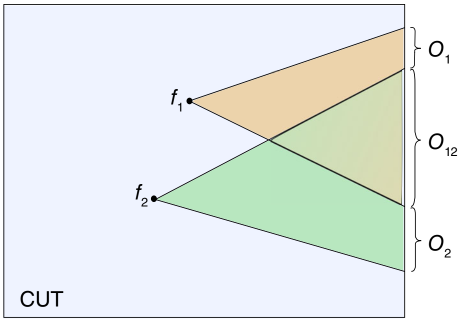

FIGURE 2 Intersecting output cones of two faults. O_1, O_{12}, and O_2 are the set of output ports reachable from only f_1, both f_1 and f_2 and, only f_2, respectively

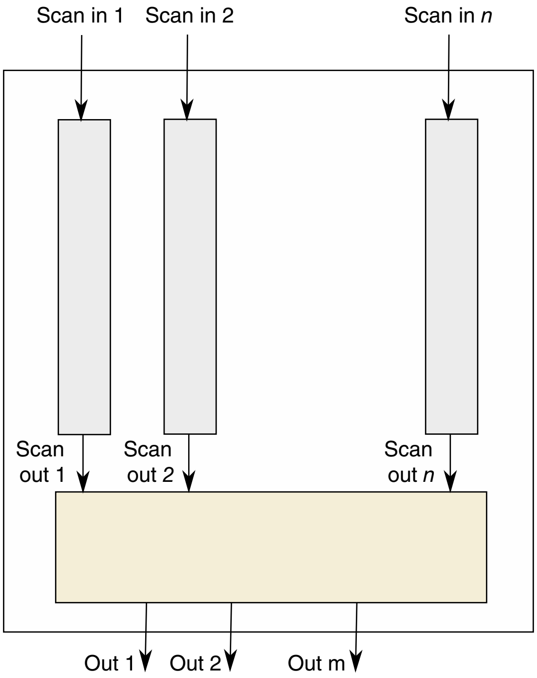

FIGURE 3 Multiple scan with response compactor (Mitra & Kim, 2002)

compression. The problems of signature aliasing, and the effects of unknown (X) on error bits have been studied by many researchers (Lee, Lien, & Hsieh, 2011; Mitra & Kim, 2002; Pomeranz, Kundu, & Reddy, 2002; Rajski, Tyszer, Wang, & Reddy, 2003).

1.2 | Overview of machine learning

ML, as the name suggests, is a field that aims at "learning" information and knowledge from data utilizing the computing power of the modern day "machine," namely computers. While its general aim, that is, making inference from the data, is similar to that of statistical learning, it is more receptive in integrating various methodologies and the kind of data for which it is applicable. Its methodology extends beyond statistical learning, from applying geometry and computer science to just mimicking biological process of neural networks, which are yet to be fully understood. The range of data may vary from simple tabular, structured data to more complex structured data like images and videos, to unstructured data such as motifs and graphs. Depending on the nature of data, two basic approaches in ML are deployed. The first approach is to analyze the data for the presence of any pattern, which is called clustering, and falls under unsupervised learning. In the second approach, each data point has labels attached to it and the data can be used to approximate some function/model that the labels are assumed to represent so that any future unlabeled data can be given a suitable label. This comes under supervised learning. The common techniques used for unsupervised ML are Bayesian inference, k-means clustering, and spectral clustering. The popular techniques that are used for supervised learning are decision trees, support vector machines (SVMs), artificial neural networks (ANNs), Bayesian networks and random forests (RFs) (Hastie, Tibshirani, & Friedman, 2009). Supervised learning is more popular because of the availability of standard tools, especially SVMs and ANNs.

While supervised learning is often preferred over unsupervised approaches, many a time, labels are not present or difficult to obtain. Hence, the method has to depend on the kind of the data available. In the area of digital logic testing, there are many opportunities for applying supervised learning. Various data sources that have been used or can be potentially used for ML applications in the field of digital electronic testing are listed below:

1 Manufacturing test response: The failure response patterns of a volume of defective chips obtained during production testing can be used for diagnosis (Huisman, Kassab, & Pastel, 2004; Xue, Poku, Li, & Blanton, 2013).
2 Simulation: A variety of labeled test data can be generated using simulation with CAD tools for testing. They can be collected for different fault models or defect types (Chern et al., 2019; L. R. Gómez & Wunderlich, 2016).
3 Historical data on diagnosis: These are labeled data, which consist of the faulty components responsible (root cause) which were diagnosed during earlier test cycles and their test failure response/syndromes. These techniques are applied especially to board-level diagnosis (Sun et al., 2013; Ye, Chakrabarty, Zhang, & Gu, 2015; Ye, Zhang, Chakrabarty, & Gu, 2013).
4 Circuit parameters: Number of input/output port, details of scan-chain (Li, Colburn, Pagalone, Narayanun, & Chakrabarty, 2017).
5 Circuit structure: The logical network structure, represented as a directed graph, provides a rich source data (Ma et al., 2019; Pradhan, Bhattacharya, Chakrabarty, & Bhattacharya, 2019). The gate-level description of a netlist also provides functional and state information of the logic circuit.
6 Physical layout (L. R. Gómez & Wunderlich, 2016; Nelson, Tam, & Blanton, 2010).

Oftentimes, we do not work with the data directly in order to obtain a meaningful model/inference because of the following issues: (a) the dimensionality of the data may be too high, (b) the data may be biased, (c) the data may be noisy, and (d) they may be unstructured. Thus, extensive preprocessing may be required before we make them suitable for ML-based computation. One of the major steps in this direction is feature engineering, that is, formulation of features from the available data. In some cases, feature selection is also needed. In the case of data such as graphs, a new method called representation learning (Bengio, Courville, & Vincent, 2013) is becoming popular.

Once we obtain a set of good data for learning (also called training data), we need to choose which learning technique to apply. Next, for supervised learning, in order to obtain a model which has good generalization capability such that it gives good results for yet unseen data (test data), the various hyper parameters should be carefully selected. Since a detailed overview of ML tools and techniques is beyond the scope of this paper, we do not discuss them here, and the reader may refer to Hastie et al. (2009). A nice introduction to ML is presented in L'Heureux, Grolinger, Elyamany, and Capretz (2017), while more insights into trends can be found in Jordan and Mitchell (2015). A number of guidelines on how to obtain a good model appear in Domingos (2012) and electronic design automation and test specific discussions in L. Wang (2017). A discussion on the application of ML on emerging problems of test can be found in the tutorial material delivered by Huang and Veda (2018).

1.3 | ML in analog circuit testing

An electronic chip consists of both analog and digital components. However, the two vary widely in their working principle and complexity and hence their testing. Analog operations are much more complex compared to digital operations and testing of analog circuits is much more challenging (Hatzopoulos, 2017; Milor, 1998). Unlike digital circuits, the signals in analog circuits are not discrete in nature, and thus, it is not easy to consider a suitable fault model that captures all error patterns while testing them. Furthermore, they exhibit nonlinear behavior and their outputs are very sensitive to various circuit and environment parameters. Therefore, most of the test techniques used for analog circuits are parameter-based, and it is difficult to design a deterministic test method. Various statistical and ML approaches have been explored in this area (Butler, Nahar, & Daasch, 2016; Stratigopoulos, 2018; L. Wang, 2017). The good side is that analog circuits are small in size and they constitute only around 10% of the chip (Karmani, Khedhiri, & Hamdi, 2011).

On the other hand, testing of digital circuits has been extensively studied over several decades and is now well understood. The components used in a digital circuit are much simpler and fault models therein are well defined.

Automated tools for test generation, fault simulation, and DfT insertion are available, which provide further options for designing efficient test strategies. Nonetheless, various technological advancements have added newer challenges to digital logic testing (Kundu, Mak, & Galivanche, 2004). A number of ML based approaches have recently been developed in this evolving area. We will review some of them and discuss future challenges in this direction.

2 | YIELD LEARNING AND DIAGNOSIS

Rapidly scaling technology demands the manufacturing process to be intricate and precise. Since manufacturing processes only improve over time, there is a low yield (fraction of good chips) in the initial phase. Yield learning (Aitken, 2012) has thus become a crucial step for the yield ramp up during volume production. Yield learning involves understanding the failures, locating the defects, and thereafter applying corrective measures to improve the manufacturing process. Figure 4 shows the various stages in yield learning (Hora et al., 2002).

The method of defect location utilizes a feedback mechanism called diagnosis. As the IC technology scales down and the level of integration scales up, the number and variety of defects inevitably escalates. Understanding these defects is an integral step in improving the manufacturing process. Traditionally, defects are located using a physical-level process called physical failure analysis (PFA) (S.-Y. Liu, Hou, Chang, & Lin, 2013; Zhao et al., 2018). However, the intricacy and multitude of defects have not only made this process challenging but also very time consuming and costly. In order to guide PFA, a common technique is to deploy a logic-level process called *fault diagnosis*. Given the failure response and the circuit netlist, fault-diagnosis produces a set of candidate faults (root cause) responsible for the failed response. Fault diagnosis follows either a cause–effect (dictionary based) or an effect–cause (inject and evaluate based) approach. There are also techniques to analyze the response for useful inference before fault diagnosis (preprocessing) and after fault diagnosis to improve PFA (postprocessing). One drawback of these methods is that they are inefficient for diagnosis of a variety of unmodeled defects, which are common in the submicron technology. Recently, a new method called volume diagnosis has been studied and also integrated in the diagnosis process, to guide PFA. The failure response of a volume of defective chips is analyzed to understand the defects and their root cause (Tang, Manish, Rajski, Keim, & Benware, 2007). Hence, volume diagnosis turns out to be a suitable arena for exploring ML applications.

Note that both combinational and memory components may be affected by defects. Thus, while conducting manufacturing test, the flip-flops are tested first and diagnosed for defects, if any, through a process called scan-chain diagnosis (Huang, Guo, Cheng, & Li, 2008). Thereafter, the faults in the rest of the circuit are diagnosed. Diagnosis is carried out hierarchically. In this section, we look at various ML-approaches that have been used for diagnosis at different levels of circuit hierarchy: wafer-level, scan-chain level, chip-level (fault diagnosis, prediagnosis and postdiagnosis, and volume diagnosis), and board-level.

2.1 | Wafer-level diagnosis

As mentioned earlier, the intricacies involved in the modern manufacturing process have made it challenging to maintain good yield during the fabrication of IC-chips. Also, the production of silicon wafers requires a long time-cycle

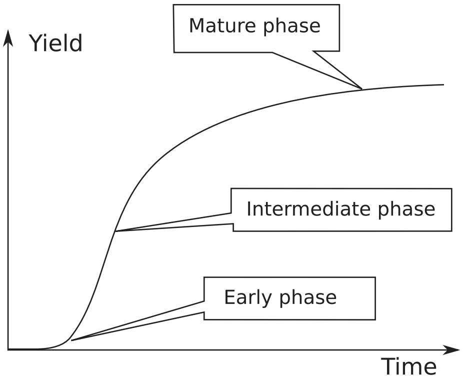

FIGURE 4 Yield learning phases (Hora, Segers, Eichenberger, & Lousberg, 2002)

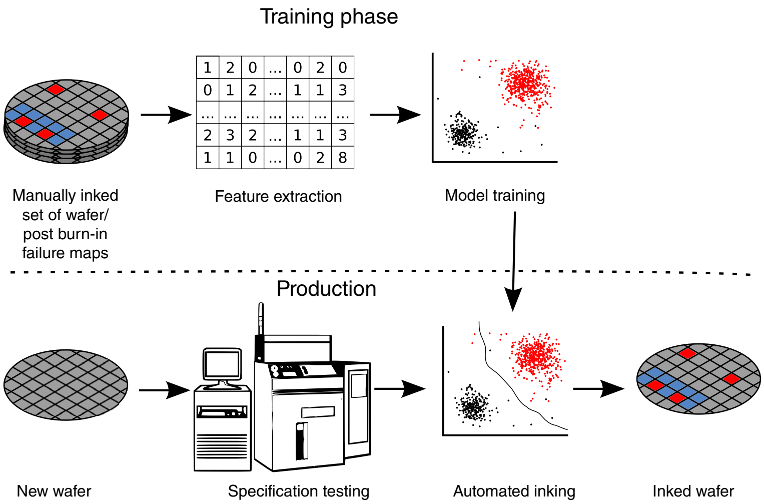

FIGURE 5 Automated die-inking (Xanthopoulos et al., 2017)

(Wang, 2017). It is therefore important to identify the defects in the wafer early in the manufacturing process so that the process can be improved to reduce time and yield loss. It has been observed that the defects generally occur in clusters in a wafer in certain locations (Ooi et al., 2010). A kernel-based method to detect such clusters is proposed in Sumikawa, Nero, and Wang (2017).

Die inking is a process of marking those dies which have latent defects. Burn-in tests are also generally applied for detecting such latent defects. However, burn-in tests are prohibitive because of the cost and the complexity involved (Xanthopoulos, Sarson, Reiter, & Makris, 2017). A method to automate this process using ML has been reported in Xanthopoulos et al. (2017). Faulty dies, which are close to defective clusters in a wafer, are inked manually while training the model. During the test phase, a binary classifier is used to decide whether or not a die is defective. An SVM with a radial basis function kernel is used, which is one of the most common kernels used in the literature that uses Gaussian function for distance computation. Morphological operations consisting of erosion/dilation are used to remove noise. A feature vector based on the distance of the die from defective clusters is used during classification. The corresponding flow diagram is shown in Figure 5.

One of the main reasons behind chip failure is process variation. There are numerous process parameters which contribute to such defect distributions. The values of these parameters may vary from wafer to wafer, and also within the same wafer. A major objective in testing is to select appropriate parameters that take care of most of the failures. In Tikkanen, Siatkowski, Sumikawa, Wang, and Abadir (2014), a canonical correlation analysis is used to determine the correlation by noting measurements over several samples. It also discusses the correlation of parameters corresponding to two different locations (inner and outer) of a wafer.

2.2 | Scan-chain diagnosis

Scan-chains, inserted in a sequential circuit as a DfT mechanism, allow accessibility to flip-flops such that they can be externally loaded with the required value corresponding to test-bits (to improve controllability), and the response to the test input can be read (to improve observability) during test application. The scan-chain is implemented by adding a multiplexer to the input of each flip-flop (also called scan cells) such that during test-mode, all flip-flops can be configured serially as a chain with the input of the first flip-flop acting as a scan-in pin, and the output of the last flip-flop acting as a scan-out pin. When the number of flip-flops becomes large, they are configured as multiple scan-chains to reduce test time. In test-mode, a test vector is loaded into the chain and the CUT is switched to the functional mode using multiplexers. Then in the next clock cycle, the test vector is applied and the responses are captured in the flip-flops. The CUT is switched back to the test mode, and as the test response is scanned out through the scan-out pin, the next test vector is loaded in the chain. Scan-chain diagnosis aims at locating the faulty scan cells. The presence of a defective scan cell might affect a large number of response bits. Some defects, called permanent faults, are known to be easy-to-model since they always produce a fixed failure response pattern for a given test vector. These faults can be diagnosed by heuristic based techniques. The remaining scan-chain defects may be due to intermittent or hard-to-diagnose faults since they do not produce a fixed failure response pattern during the production test. A method based on Bayesian learning (Tipping, 2004) is proposed in Huang et al. (2017) to identify the faulty scan cells in the presence of such defects. Given a test set and the failure log for a faulty scan-chain, for each scan cell i, two kinds of bit-count are considered: $sbit_i$ is the count of patterns for which the cell i is supposed to capture a failure bit (sensitive bit) differing from the fault-free response bit; $fbits_i$ is the count of bits out of $sbit_i$ which actually fail in the production test. For a given faulty chain, the probability that a scan cell i captures $fbits_i$ faulty bits is computed by assuming a binomial distribution. Diagnosis is then facilitated using the Bayes formula.

In the above-mentioned technique, Bayesian learning is used for clustering which is an unsupervised learning technique. It relies on unknown priors which may not be always readily available. The problem of scan-chain diagnosis for intermittent faults is modeled following a supervised learning method (Chern et al., 2019) using another ML-tool based on ANN. An ANN mimics the working principles of a biological neural network. Its architecture consists of several layers of nodes called perceptrons, where each node computes a linear combination of the inputs feeding it from the preceding layer. A nonlinear component is added by incorporating a nonlinear activation function at each node. The weights attached to the edges incident on each node are updated and learned during the training phase by a method called back-propagation. Chern et al. (2019) employed a multistage ANN for diagnosis of faults following a coarse-to-fine approach. Each data point, called the modeled faults, represents a fault type, the faulty cell and fault intermittency (probability that a fault is activated by the test patterns). In the first stage, called coarse-global neural network (CGNN), the binary response vectors for the test set are reduced to a single vector called integer failure vector (IFV). It is computed by performing bitwise addition of the binary response vectors. For an illustration, the figure from Chern et al. (2019) is redrawn in Figure 6. The input to the ANN is an IFV whose length is determined by the number of scan-cells, and each node in the output layer represents a scan-cell of a particular scan-chain for which the ANN is being trained. Such an ANN gives a candidate faulty scan-cell as output called center cell. An example of CGNN training vector is given in Figure 7. In the subsequent stages, called refined local neural networks, an affine group is computed over those scan-cells whose IFV is close to that of the center cell in terms of Euclidean distance. The ANN is built for each scan-cell based on its affine group. In these stages, instead of compressing the binary response vectors to a single IFV, they are concatenated sequentially to form a single vector. Its length can be reduced by removing the bits at certain positions based on the affine group and the new vector is called “reduced cascaded vector (RCV)”. Thus, the ANN for these stages

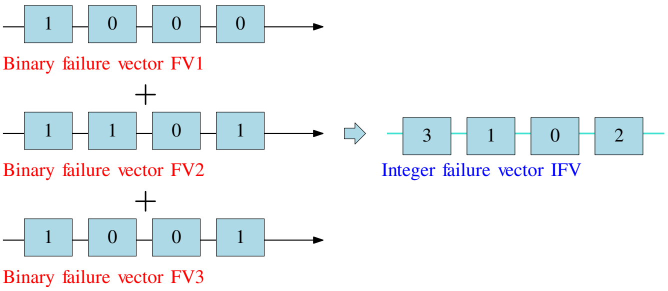
1 0 0 0
Binary failure vector FV1
+
1 1 0 1
Binary failure vector FV2
+
1 0 0 1
Binary failure vector FV3
3 1 0 2
Integer failure vector IFV

FIGURE 6 Compressing binary failure vectors into an “integer failure vector” (Chern et al., 2019)

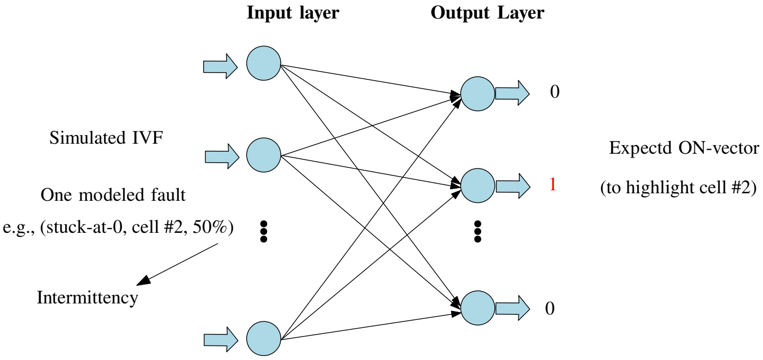

FIGURE 7 An example of CGNN training-vector (Chern et al., 2019)

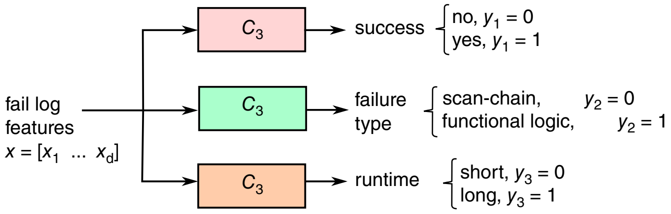

FIGURE 8 Three-output classifiers where X is a feature vector with d elements, and y_1, y_2, and y_3 are discrete variables denoting the classes (Huang et al., 2018)

turns out to be a two-layer network where the number of nodes of the first layer is equal to the length of RCV, and the number of nodes in the output layer is equal to the number of cells in the affine group. It is shown that with this supervised technique, the diagnostic accuracy could be increased by 20%.

2.3 | Fault diagnosis: preprocessing

The failure log of defective chips can provide useful information that could guide the fault diagnosis process. The entire failure data may not be available for diagnosis because data collection is expensive and time consuming (Wang et al., 2012). Since only a small fraction of the total response data is available, it usually leads to poor diagnosis. To alleviate this problem, a method is proposed in Wang et al. (2012) to determine a minimal number of test responses that are required for proper diagnosis. The method utilizes a binary classifier that decides when the response collection process should be stopped. When a test response is collected, the classifier decides whether or not to continue to the next response. The features are based on the output response of the chip up to the application of the last test pattern. They have reported results for various classifiers such as kNN, SVM, and decision tree.

The work in Huang, Fang, Mittal, and Blanton (2018) introduces a classifier to predict the following: (a) whether the failure log is at all useful for diagnosis, (b) the location of defects: scan-chain or functional logic, and (c) the time needed for diagnosis. They have presented a set of features based on the failure log and used RF to design the classifier. This is illustrated in Figure 8.

2.4 | Fault diagnosis: postprocessing

Although fault diagnosis plays a major role in guiding the process of PFA, it is conducted at the abstract level. Moreover, the number of candidate faults reported (diagnostic resolution) is generally large. Many methods have been proposed to fine-tune the results of fault diagnosis based on ML techniques. They are usually concerned with two objectives: (a) defect identification, that is, mapping the diagnosed fault to a defect. This is challenging especially when it is based only on the failure response of the circuit (Gomez, Cook, Indlekofer, Hellebrand, & Wunderlich, 2017; L. R. Gómez & Wunderlich, 2016; Nelson et al., 2010); (b) improving diagnostic resolution, where the candidate faults are analyzed so as to further prune the set in order to improve the diagnostic resolution (Xue et al., 2013). The features used in both approaches are derived from the layout and logical information of the circuit and the output response of the failing chip.

2.4.1 | Defect identification

The problem of identifying bridging defects in a failing circuit has been addressed in Nelson et al. (2010). Such information is helpful for estimating defect density and size distribution, which is required in yield learning (Nelson et al., 2006). Since a bridging defect represents a short between two signal lines, for each candidate fault involving line A, a set of bridging faults is considered involving its neighboring lines (B, X, Y) which are {(A, B), (A, X), (A, Y)}. The logical information of the circuit is expressed as Boolean features. For example, the feature called "feedback" checks whether there is a structural path between the pair of lines of a bridging fault. Under the fault, such a path might create a latch or induce an oscillating behavior affecting the test result. Hence, these sites could be disregarded as a possible candidate

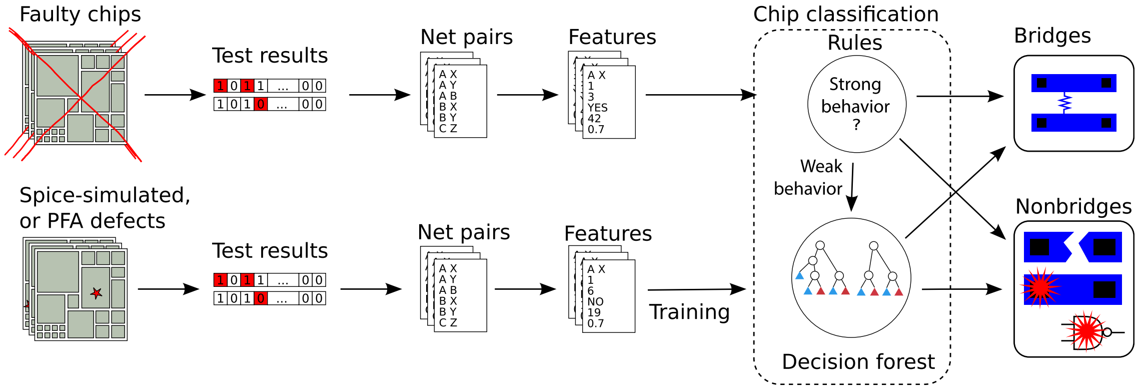

FIGURE 9 Flow for the classification of bridging defects (Nelson et al., 2010)

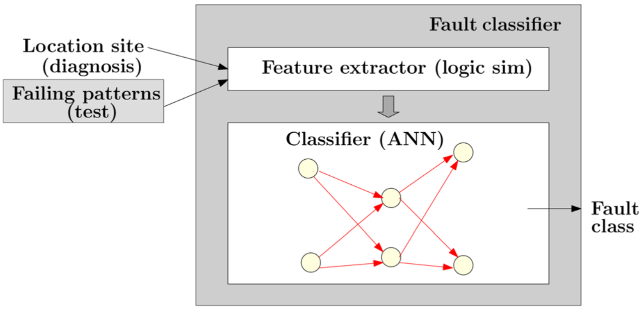

FIGURE 10 Defect classifier (L. R. Gómez & Wunderlich, 2016)

bridging fault. Similarly, other Boolean features are used for checking whether the lines drive a parity gate or same gates or have logical correlation. The test-dependent features are formulated by analyzing the correlation between the tester output and the simulated response of the circuit in the presence of the candidate defect under various bridging fault models (L.-T. Wang et al., 2006). These faults are processed by a rule-based classifier followed by decision-tree based classification. The faults that possess high scores of both logical and test-based features are classified as bridge faults. Similarly, non-bridge faults are identified by rule-based classification and the remaining faults are then classified by a decision tree (Rokach & Maimon, 2014). The flow of the classification scheme (Nelson et al., 2010) is shown in Figure 9. The training set is created by using SPICE simulation as well as from the results of defect diagnosis obtained by PFA.

A neural-network based defect classifier for various faults is proposed in L. R. Gómez and Wunderlich (2016). This can be used to classify defects at the early stages of volume diagnosis without using any special diagnostic test patterns. The network provides a warning signal as soon as the frequency of certain defects crosses a threshold. A candidate fault may be classified as various defect types, for example, crosstalk induced delay, dominant-and(-or) bridge, Byzantine bridge, slow-to-rise or slow-to-fall bridge. Thirteen different features are computed based on the simulation of failing patterns in the presence of the faulty candidate. The features are expressed as the proportion of the number of failing patterns exhibiting certain properties. For example, one feature is defined as the proportion of failing patterns which set logic value 0 at the victim line. This is used for identifying dominant-OR-bridging. Furthermore, layout information is used to find the neighborhood geometry, which helps to define a feature that characterizes cross-talk induced faults. The scheme for this ANN-based classifier is shown in Figure 10. Two sets of data are used in the experiments. First, the simulated data for a circuit is used to classify the faults in the same circuit. Second, the simulated data for a group of circuits is used to classify faults in a new circuit. The latter experimental setup seems to be of more practical value. This is because, generating the training data by simulation for every new circuit is time consuming and may not be feasible. However, this would require rich training datasets, which are still a challenge to collect.

Another set of faults which are targeted for classification consists of transient and intermittent faults (Gomez et al., 2017). Both types produce similar test results and it is difficult to distinguish them. While intermittent faults lead to the degradation of the chip, transient faults contribute to unnecessary yield loss (Gomez et al., 2017). A tool based on Bayesian classifier is proposed in Gomez et al. (2017) to distinguish these two types of faults.

2.4.2 | Improving diagnostic resolution

As discussed in Section 2.4, the second approach towards fault diagnosis aims at processing the candidate faults to improve the diagnostic resolution. Most of the diagnostic tools produce a larger number of candidate faults compared to the actual number of faults (Xue et al., 2013). The efficiency of the ensuing steps of defect identification and PFA is thus impeded. In order to improve the diagnostic resolution, we need to pare down the set of candidate faults. A classification-based method similar to that in Nelson et al. (2010) is proposed by Xue et al. (2013) to label each candidate fault as either good or bad. A set of features is identified and thereafter the classification is performed in two steps: (a) the first one is rule-based, which identifies some bad candidates; (b) the rest of the faults are analyzed by an SVM-based classifier. For the purpose of training, the labeled data available from PFA was not found to be so useful because of the fact that they are sparse and furthermore, being some kind of "old data," they may introduce error when applied to new circuits. In order to alleviate these problems, Xue et al. (2013) proposed a method to generate labeled data for each circuit: the defective chips, for which fault diagnosis yields a single candidate fault, are used to mark "good" candidate class. The chips for which the fault diagnosis provides a large number of candidate faults (greater than a threshold value) are used to mark "bad" candidate class. It is, however, observed that the number of faults in the "bad" class is generally much larger than those in the "good class." As a result, the problem of managing unbalanced classes arises, which is handled by oversampling the faults from the "good" class.

2.5 | Volume diagnosis

Although today's diagnostic tools for handling faults in a digital circuit can achieve high accuracy, they still suffer from several drawbacks. To mention a few, they are often unable to distinguish functionally equivalent faults; they do not take into account the entire layout information on which the likelihood of defects is highly dependent. Feature-based systematic defects now impact deep submicron technology significantly. Available tools are incapable of diagnosing them. Even the yield learning methods such as PFA may not be suitable for handling them. In order to detect such defects, drawing inference from fail-logs of a large number of chips, has become imperative. This process is called volume diagnosis. Since it involves an analysis of huge amount of data, the method needs to be time-efficient.

One of the crucial needs of the yield-learning process is the ability to identify systematic defects in the chips and to distinguish them from random defects (Huisman et al., 2004; Mutschler, 2014). A signature for each defective chip is created based on its failure response. Based on these signatures, the chips are clustered using the furthest-neighbor method (Dillon & Goldstein, 1984). Such clustering would help to analyze whether the chips in a cluster are failing due to a similar defect. Thus, it can be used to determine whether or not the defect is systematic. Another classification-based method for volume diagnosis is proposed in S. Wang and Wei (2009). This classifier goes further and detects the location of the defect in terms of fan-out free regions. This is based on the observation that faults in a fan-out free region affect the same set of outputs. The CUT is decomposed into fan-out free regions, and each region is considered as a defect class. Defect classification is performed based on the failure outputs using an SVM. When a large number of chip failures occur due to a particular class, the presence of a system defect is inferred.

Volume diagnosis generally reports multiple failure features for each chip. A statistical-learning approach is presented (Tang et al., 2007) to estimate failure feature probabilities. Another method based on Bayesian network is described in Cheng, Tian, and Reddy (2017).

A method to assist PFA by narrowing down the possible set of defects is discussed in Shan, Babighian, Pan, Carulli, and Wang (2017). A defect can have various signatures called "defective modes." During volume diagnosis, χ^2 independence test is applied to check whether the defects and the "defective modes" are related. Using the data obtained from layout-aware scan diagnosis, and test results, the values of χ^2 test are found. The *p*-values for the "defective modes" are used to rank them. *p*-Value is a statistical measure used in hypothesis testing to determine the statistical significance of a result.

2.6 | Board-level diagnosis

The technology of printed circuits boards has enabled integration of diverse components such as application-specific integrated circuits, memory, and I/O on a single board. Consequently, diagnosis is needed at the board level as well. It has been experienced that though the individual components may pass the manufacturing test in the ATE, they fail the board-level functional test. This is primarily due to the difference in the real testing environment from that of ATE, and the components are marked as "no trouble found". This is a dreaded problem in industry which needs careful management to ensure reliability of digital systems and for their regular maintenance. Board-level functional fault diagnosis follows a reasoning based approach. The knowledge regarding the root cause of failure-syndromes for an initial set of boards which, could be repaired, is used as training data to predict defective components for new boards. The syndromes are gathered from the failure information of the components under a test set. These syndromes lead to a set of features and the underlying root-cause instances that are diagnosed serve as labels in the training set. A number of approaches based on various ML techniques such as ANN (O'Farrill, Moakil-Chbany, & Eklow, 2005; Zhang, Chakrabarty, Wang, Wang, & Gu, 2011), SVM (Ye, Zhang, Chakrabarty, & Gu, 2014; Zhang et al., 2012), and decision trees have been proposed in this direction (Sun et al., 2013; Ye et al., 2015, 2013). In the ANN based approach (Zhang et al., 2011), the inputs are fed with different syndromes and the outputs denote the components. In order to handle large-size board-level diagnosis problems, Zhang et al. (2011) used a group of two-layer, single-output ANNs (Figure 11), where the output node represents a component and classifies it as being the root cause of a failure or otherwise.

The major concern of most of the ML applications in this area is the dependence of training sets on historical data, which are often limited. Apart from having limited access to past data, the size of the feature vector is usually large as the test-set size is large, leading to overfitting at the time of training. In order to overcome this, a scheme called syndrome merging was used to reduce the size of the feature vector (Sun et al., 2015). Note that some type of syndromes may not be observable or computable. For such cases, another technique was proposed by Jin, Ye, Zhang, Chakrabarty, and Gu (2016) to process the training set including those based on naïve Bayes classifiers.

3 | ATPG, TEST COST, AND TESTABILITY ISSUES

While most of the ML-based test approaches developed so far focus on fault diagnosis in digital circuits, some recent advances include ATPG and testability analysis, where ML-techniques have been shown to provide novel solutions. In

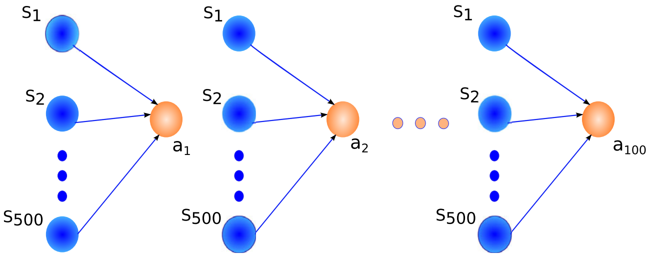

FIGURE 11 An illustration of the ANN architecture used by Zhang et al. (2011)

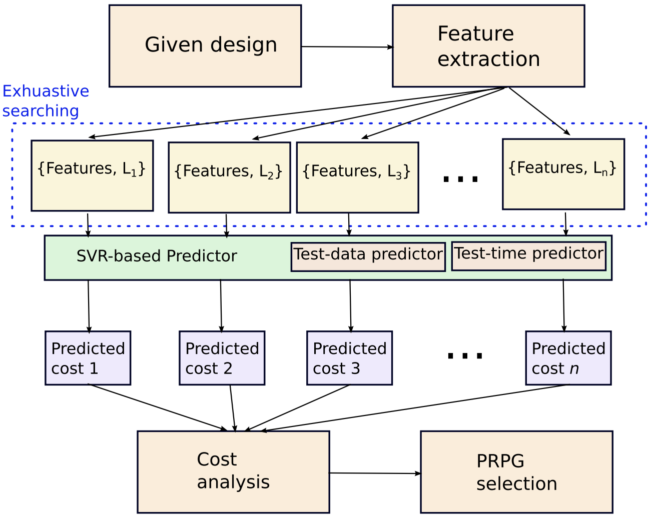

FIGURE 12 Illustration of the PRPG-selection method (Li et al., 2017)

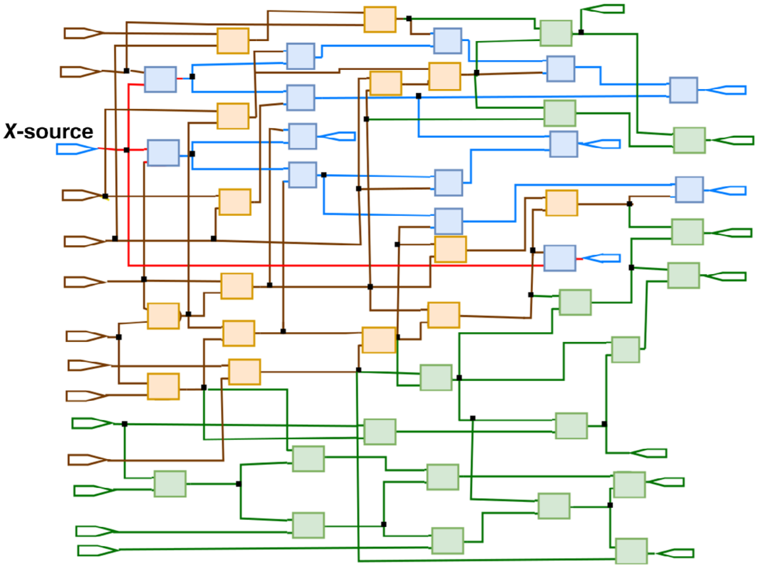

FIGURE 13 A combinational circuit showing an *X*-source (blue), the three partitions $\mathcal{P}_1$ (blue), $\mathcal{P}_2$ (brown), $\mathcal{P}_3$ (green), and the gates directly fed by the *X*-source (red) (Pradhan et al., 2019)

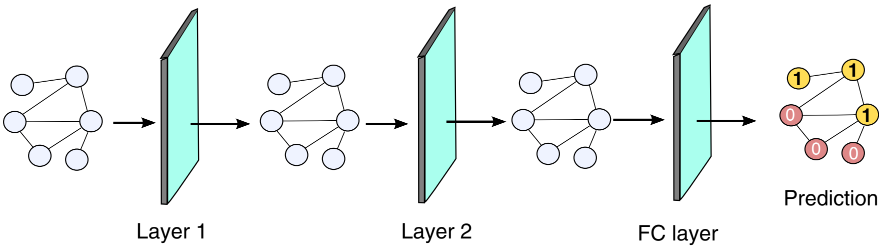

FIGURE 14 Network architecture of GCN. Node embeddings are generated in Layer 1 and Layer 2. The fully connected (FC) third layer execute nodes classification (Ma et al., 2019)

particular, the applications involve test compression, estimating fault-coverage under unknown (*X*) inputs, circuit-testability improvement, and timing analysis.

3.1 | Test compression

The test-cost is measured in terms of the test data volume, and test time. In scan-based test environment, one way to reduce test-cost is the adoption of the compressor/decompressor architecture. A pseudo-random pattern generator (PRPG), sometime along with a decompressor, is used to load the scan-chains. Similarly, the test-response data is compressed by using a MISR. It has been shown that besides various circuit parameters, the length of the PRPG greatly affects the test-cost (Li et al., 2017). While the problem of designing the pattern generator can be resolved exhaustively by running ATPG, it may become infeasible because of the time needed. A predictor-based on support vector regressor (SVR) is proposed to tackle this problem Li et al. (2017). A number of features are extracted from the ATPG log-file from which suitable features are selected.

An illustration for selecting the length of PRPG is given in Figure 12. Two separate predictors, one for test time and the other for test-data volume, are trained. For each choice of the length of PRPG, test-cost is predicted, and the length corresponding to the minimum cost is selected. (Li et al., 2017; Pradhan et al., 2019).

3.2 | Circuit testability

In the area of circuit testability, ML-based research has been conducted in two directions. The first aims to estimate the loss of fault-coverage due to an *X*-signal (unknown logic value) at the input of a CUT. The other addresses the problem of test-point insertion for improving testability.

TABLE 1 Summary

Work	Problem	Data	Method
	Wafer level diagnosis		
Sumikawa et al. (2017)	Identifying defect clusters	FD	Clustering
Xanthopoulos et al. (2017)	Automated die inking	Historical data	SVM
Tikkanen et al. (2014)	Correction of failure and parameters	FD	Statistical correlation
	Scan-chain diagnosis		
Huang et al. (2017)	Targeting hard-to-model faults	FD	Bayesian method
Chern et al. (2019)	—	SD	Multi-stage ANN
	Fault diagnosis pre-processing		
Wang et al. (2012)	Regulation of test data volume	FD	Classification
Huang et al. (2018)	Inferring diagnostic efficiency	FD	Random forest
	Fault diagnosis postprocessing		
L. R. Gómez and Wunderlich (2016)	FI: defect classification	SD	ANN
Nelson et al. (2010)	FI: identifying bridging defects	SD and FD	Decision tree
Gomez et al. (2017)	FI: transient and intermittent faults	SD	Bayesian network
Xue et al. (2013)	Improving diagnostic resolution	SD	SVM
	Volume diagnosis		
S. Wang and Wei (2009)	VD of unmodeled faults	SD	SVM
Cheng et al. (2017)	VD for root cause identification	Volume FD	Bayesian network, MLE
Huisman et al. (2004)	Identification of systematic defects	FD	Clustering (furthest neighbor)
	Board level diagnosis		
Sun et al. (2013), Ye et al. (2015, 2013)	Fault isolation	Historical data	SVM/ANN/decision tree
Sun et al. (2015)	Syndrome merging	—	—
Jin et al. (2016)	Missing syndrome computation	—	Naive Bayes
	Test compression		
Li et al. (2017)	Test cost optimization	SD	SVR
	Circuit testability		
Pradhan et al. (2019)	Prediction of *X*-sensitivity	SF and SD	SVR
Ma et al. (2019)	Test point insertion	SFs, SCOAP, SD	GCN
	Timing analysis		
Y. Liu, Han, Lin, and Li (2017)	Based on PSN	SD	Multiple tools

Abbreviations: FD, failure data; FI, fault identification; SD, simulated data; SF, structural feature; VD, volume diagnosis.

3.2.1 | Analysis of *X*-sensitivity

The presence of *X*-sources degrades the detectability of a circuit, and consequently, causes a loss in fault-coverage for a given test set because these values cannot be computed during ATPG. *X*-sources include uninitialized memory cells, bus contention, erroneous behavior of analog-to-digital converters, to name a few. Moreover, in the post-silicon validation phase, lot of design bugs that are identified exhibit themselves as *X*-values. *X*-sensitivity, that is, the effect of *X* on the loss of fault-coverage in digital circuits has been studied in Pradhan et al. (2019). A fast and effective method for predicting *X*-sensitivity of inputs in a digital circuit has been proposed. This prediction is useful for ranking the *X*-sources. The inputs that have high *X*-sensitivity can be considered for *X*-masking/elimination from the viewpoint of improving testability and also for rectifying those design bugs which induce *X*-values, by rewiring of certain components during post-silicon validation. Likewise, in the case of memory cells capturing *X*-values, those among them which have negligible impact on detectability can be left uninitialized, thus saving test-time. One way of measuring the detectability-loss would be to exhaustively run ATPG. However, it is impractical due to huge computation time involved. An SVR based *X*-sensitivity predictor has been developed by Pradhan et al. (2019).

A number of features, which are exclusively based on the structural characteristics of the circuit, have been identified. The structural influence is studied based on three partitions of the circuit as shown in Figure 13: $\mathcal{P}_1$, the output cone of the *X*-source, $\mathcal{P}_2$, the subcircuit that influences the propagation of *X*, and $\mathcal{P}_3$, the rest of the circuit. It is easy to observe that only the gates in $\mathcal{P}_1$ and $\mathcal{P}_2$ are affected by the *X*-source, and $\mathcal{P}_1$ is affected more than $\mathcal{P}_2$. Taking these facts into consideration, some novel features are proposed based on the two partitions. Some features are determined by the fraction of sensitive gates in $\mathcal{P}_1$, fraction of output ports in $\mathcal{P}_1$, the number of gates directly reachable from the *X*-source.

3.2.2 | Test-point insertion

The problem of test-point insertion in a logic circuit has been studied in Ma et al. (2019) from an ML perspective and a classifier has been built. This is the first time where a deep-learning based technique has been deployed for handling a test problem. Moreover, attempts have been made to learn from circuit-graphs; this was a challenge because ML-tools are more suited for structural/vector data, whereas a graph mostly comprises unstructured information. Ma et al. (2019) proposed a neural network called graph convolutional network (GCN) to analyze graphical data. The nodes of the graphs representing the circuit netlist are classified as either easy-to-observe or difficult-to-observe points. Graphical features are represented using a node embedding method. A number of attributes related to testability obtained using the tool SCOAP (Goldstein & Thigpen, 1980), are attached to each node. Based on these attributes and the local neighborhood information of a node, an embedding of each node is produced by the GCN. The overall flow of the classifier is shown in Figure 14.

3.3 | Timing analysis

Timing analysis of a circuit is required to determine the clock frequency of the circuit. The timing of a circuit depends on many static as well as dynamic (input pattern dependent) characteristics.

Power supply noise (PSN) affects the input voltage reaching the gates and hence the propagation delay. It is one of the factors that influence dynamic timing analysis (DTA) of the circuit. In order to speed up DTA, a ML approach to predict the circuit timing, taking into account the PSN-effect was developed by Y. Liu, Han, Lin, and Li et al. (2017). The prediction of circuit delay due to voltage droop using SVM was proposed by Ye, Firouzi, Yang, Chakrabarty, and Tahoori (2016).

4 | SUMMARY, CHALLENGES AND FUTURE DIRECTIONS

A summary of the relevant literature is given in Table 1. Next, we will look at the various challenges and future directions. Although there are many scenarios concerning digital logic testing where ML has been or could be applied, they still appear to be fragmented and unorganized. The success of ML-based techniques strongly relies on the availability of sufficient data with good quality and volume. While some potential data sources can be accessed as enlisted in Section 1.2, standard ML-databases in regard to IC testing are yet to be prepared, and thus, their unavailability stands as major impediment towards the adoption of ML-tools. Some of the reasons behind this bottleneck are listed below:

1 Absence of industrial time-series test data (Rehani, Abercrombie, Madge, Teisher, & Saw, 2004): Most of the databases on the failure log collected during production testing of ICs along with the corresponding diagnostic information are not available in the public domain. Such data would serve as a rich source for updating training models and help guide future diagnosis processes for IC-chips.

2 Complexity in simulation: The generation of simulated data is a very time consuming process (Li et al., 2017; Pradhan et al., 2019). A general repository of simulated data would serve as a good data source and may strengthen ML-based tools.

3 Absence of baseline: There are no benchmark circuits for evaluating ML-approaches to test problems (L. I. R. Gómez, 2017). The circuit structure can, however, be extracted from the netlist of the benchmark circuits. They can serve as a potential source of data from which structural as well as functional features can be derived. The benchmark suites

currently available (Albrecht, 2005; Brglez, Bryan, & Kozminski, 1989; Brglez & Fujiwara, 1985; Corno, Reorda, & Squillero, 2000) are not meant for ML studies. Among them, there is a lack of diversity and volume. In order to create such data, a number of large benchmark circuits having variety in their interconnection structure and functionality are to be built. Such a repository may be built by collecting industrial circuits or by unbiased synthesis.

4 Feature extraction from circuits: It is evident that feature engineering is a crucial step in most of the settings. In the context of digital logic testing, there is an urgent need for automatic feature extraction from circuit netlists. It is also required to avoid over-fitting and to filter out noisy data (in the case of output response data from a CUT). Note that circuit-data are not always in learnable or structured format (e.g., logical interconnection or physical layout data). Hence, a significant amount of time and effort is lost in this process. Also the features are rarely reusable, because most of the time, a new feature set is required for every different test problem. As manual feature engineering is a cumbersome process, automated feature extraction from circuits is highly needed. This is still an open problem in the area of logic testing albeit there have been many such endeavors in other fields such as pattern recognition and image analysis. Deep learning has been applied successfully on image and video data where the underlying features, instead of being separately extracted/selected, are implicitly utilized during learning and testing, based on convolutional techniques. Also, new methods such as representation learning have been developed to handle unstructured data, which can make the data ready for direct application of ML-tools. The work proposed by Ma et al. (2019) has pioneered efforts in this direction, with the introduction of GCN for node embedding of a graph, representing the netlist of a circuit.

5 | CONCLUSION

In this survey, we have looked at various problems that arise in VLSI circuit testing and diagnosis where ML has been applied successfully. They have outperformed traditional heuristic-based approaches in handling the complexity of the problem and provided realistic solutions much faster.

Solutions to the challenges of data generation and automated feature engineering will kick off further adoption of ML-approaches to other problems of chip testing in the future. Needless to say, there remains enough scope for data generation and development of representation techniques for digital circuits that will enrich industrial as well as academic research in the area of ML-guided test.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

AUTHOR CONTRIBUTIONS

Manjari Pradhan: Conceptualization; formal analysis; investigation; methodology; resources; software; validation; visualization; writing-original draft; writing-review and editing. **Bhargab B. Bhattacharya:** Conceptualization; formal analysis; investigation; methodology; resources; software; supervision; validation; visualization; writing-original draft; writing-review and editing-equal.

ORCID

Manjari Pradhan https://orcid.org/0000-0002-6453-4971
Bhargab B. Bhattacharya https://orcid.org/0000-0002-5890-2483

RELATED WIRES ARTICLES

Survey of neural network-based models for short-term traffic state prediction

REFERENCES

Abramovici, M., Breuer, M. A., & Friedman, A. D. (2002). *Digital systems testing and testable design*. NJ: IEEE Press.

Aitken, R. (2012, February). Yield learning perspectives. *IEEE Design and Test of Computers*, *29*(1), 59–62.

Albrecht, C. (2005). *IWLS 2005 Benchmarks* (Tech. Rep.).

Automatic Test Equipment (ATE) and Production Test. (2006). *Integrated circuit test engineering: Modern techniques* (pp. 257–266). London: Springer.

Bengio, Y., Courville, A., & Vincent, P. (2013, August). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Brglez, F., Bryan, D., & Kozminski, K. (1989). Combinational profiles of sequential benchmark circuits. In *Proc. ISCAS* (pp. 1929–1934).

Brglez, F., & Fujiwara, H. (1985). A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran. In *Proc. ISCAS* (pp. 695–698).

Bushnell, M., & Agrawal, V. (2005). *Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits*. New York: Springer Publishing Company.

Butler, K. M., Nahar, A., & Daasch, W. R. (2016, November). What we know after twelve years developing and deploying test data analytics solutions. In *Proc. ITC* (pp. 1–8).

Cheng, W., Tian, Y., & Reddy, S. M. (2017, May). Volume diagnosis data mining. In *Proc. ETS* (pp. 1–10).

Chern, M., Lee, S.-W., Huang, S.-Y., Huang, Y., Veda, G., Tsai, K.-H. H., & Cheng, W.-T. (2019). Improving scan chain diagnostic accuracy using multi-stage artificial neural networks. In *Proc. ASPDAC* (pp. 341–346). New York, NY: ACM.

Corno, F., Reorda, M., & Squillero, G. (2000, July). RT-level ITC'99 benchmarks and first ATPG results. *IEEE Design and Test of Computers*, *17*(3), 44–53.

Dillon, W. R., & Goldstein, M. (1984). *Multivariate analysis methods and applications*. New York, NY: John Wiley & Sons.

Domingos, P. (2012, October). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87.

Goldstein, L. H., & Thigpen, E. L. (1980, June). SCOAP: Sandia controllability/observability analysis program. In *Proc. DAC* (pp. 190–196).

Gómez, L. I. R. (2017). *Machine learning support for logic diagnosis*. (Unpublished doctoral dissertation). University of Stuttgart.

Gomez, L. R., Cook, A., Indlekofer, T., Hellebrand, S., & Wunderlich, H. (2017). Adaptive Bayesian diagnosis of intermittent faults. *Journal of Electronic Testing: Theory and Applications*, *30*(5), 527–540.

Gómez, L. R., & Wunderlich, H. (2016, November). A neural-network-based fault classifier. In *Proc. ATS* (pp. 144–149).

Gruning, T., Mahlstedt, U., & Koopmeiners, H. (1991, November). DIATEST: A fast diagnostic test pattern generator for combinational circuits. In *Proc. ICCAD* (pp. 194–197).

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd ed.). New York, NY: Springer New York.

Hatzopoulos, A. A. (2017, July). Analog circuit testing. In *Proc. IMSTW* (pp. 1–6).

Hora, C., Segers, R., Eichenberger, S., & Lousberg, M. (2002, October). An effective diagnosis method to support yield improvement. In *Proc. ITC* (pp. 260–269).

Huang, Q., Fang, C., Mittal, S., & Blanton, R. D. S. (2018, October). Improving diagnosis efficiency via machine learning. In *Proc. ITC* (pp. 1–10).

Huang, Y., Benware, B., Klingenberg, R., Tang, H., Dsouza, J., & Cheng, W. (2017, November). Scan chain diagnosis based on unsupervised machine learning. In *Proc. ATS* (pp. 225–230).

Huang, Y., Guo, R., Cheng, W., & Li, J. C. (2008, May). Survey of scan chain diagnosis. *IEEE Design and Test of Computers*, *25*(3), 240–248.

Huang, Y., & Veda, G. (2018, July). Machine learning and its applications in test. In *Tutorial*. India: ITC India.

Huisman, L. M., Kassab, M., & Pastel, L. (2004, October). Data mining integrated circuit fails with fail commonalities. In *Proc. ITC*.

Jin, S., Ye, F., Zhang, Z., Chakrabarty, K., & Gu, X. (2016, June). Efficient board-level functional fault diagnosis with missing syndromes. *IEEE Transactions on Computer-Aided Design*, *35*(6), 985–998.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260.

Karmani, M., Khedhiri, C., & Hamdi, B. (2011). Design and test challenges in nano-scale analog and mixed CMOS technology. In *VLSICS* (Vol. 2).

Kundu, S., Mak, T. M., & Galivanche, R. (2004, October). Trends in manufacturing test methods and their implications. In *Proc. ITC* (pp. 679–687).

Lapedus, M. (2019, March). *Finding defects in chips with machine learning*. https://semiengineering.com/finding-defects-with-machine-learning/

Lee, K.-J., Lien, W.-C., & Hsieh, T.-Y. (2011, October). Test response compaction via output bit selection. *IEEE Transactions on Computer-Aided Design*, *30*(10), 1534–1544.

L'Heureux, A., Grolinger, K., Elyamany, H. F., & Capretz, M. A. M. (2017). Machine learning with big data: Challenges and approaches. *IEEE Access*, *5*, 7776–7797.

Li, Z., Colburn, J. E., Pagalone, V., Narayanun, K., & Chakrabarty, K. (2017, April). Test-cost optimization in a scan-compression architecture using support-vector regression. In *Proc. VTS* (pp. 1–6).

Liu, S.-Y., Hou, Y.-C., Chang, C.-C., & Lin, J.-C. (2013, July). Sige profile inspection by using dual beam fib system in physical failure analysis. In *Proc. IPFA* (pp. 490–492).

Liu, Y., Han, C., Lin, S., & Li, J. C. (2017). PSN-aware circuit test timing prediction using machine learning. *IET Computers Digital Techniques*, *11*(2), 60–67.

Ma, Y., Ren, H., Khailany, B., Sikka, H., Luo, L., Natarajan, K., & Yu, B. (2019, June). High performance graph convolutional networks with applications in testability analysis. In *Proc. DAC* (pp. 1–6).

Milor, L. S. (1998, October). A tutorial introduction to research on analog and mixed-signal circuit testing. *IEEE Transactions on Computer Aided-Design*, *45*(10), 1389–1407.

Mitra, S., & Kim, K. S. (2002). *X*-compact: An efficient response compaction technique for test cost reduction. In *Proc. ITC* (pp. 311–320).

Mutschler, A. S. (2014, December). *Yield ramp challenges increase*. https://semiengineering.com/yield-ramp-challenges-increase/

Nelson, J. E., Tam, W. C., & Blanton, R. D. (2010, November). Automatic classification of bridge defects. In *Proc. ITC* (pp. 1–10).

Nelson, J. E., Zanon, T., Desineni, R., Brown, J. G., Patil, N., Maly, W., & Blanton, R. D. (2006, March). Extraction of defect density and size distributions from wafer sort test results. In *Proc. DATE* (Vol. 1, pp. 1–6).

O'Farrill, C., Moakil-Chbany, M., & Eklow, B. (2005, November). Optimized reasoning-based diagnosis for non-random, board-level, production defects. In *Proc. ITC* (pp. 7–179).

Ooi, M. P., Sim, E. K. J., Kuang, Y. C., Kleeman, L., Chan, C., & Demidenko, S. (2010, May). Automatic defect cluster extraction for semiconductor wafers. In *Proc. IMTCP* (pp. 1024–1029).

Pomeranz, I., Kundu, S., & Reddy, S. M. (2002). On output response compression in the presence of unknown output values. In *Proc. DAC* (pp. 255–258).

Pradhan, M., Bhattacharya, B. B., Chakrabarty, K., & Bhattacharya, B. B. (2019, December). Predicting X-sensitivity of circuit-inputs on test-coverage: A machine-learning approach. *IEEE Transactions on Computer-Aided Design*, *38*(12), 2343–2356.

Rajski, J., Tyszer, J., Wang, C., & Reddy, S. M. (2003, September). Convolutional compaction of test responses. In *Proc. ITC* (Vol. 1, pp. 745–754).

Rehani, M., Abercrombie, D., Madge, R., Teisher, J., & Saw, J. (2004, Oct). ATE data collection – A comprehensive requirements proposal to maximize ROI of test. In *Proc. ITC* (pp. 181–189).

Rokach, L., & Maimon, O. (2014). *Data mining with decision trees: Theory and applications* (2nd ed.). River Edge, NJ: World Scientific.

Shan, C., Babighian, P., Pan, Y., Carulli, J., & Wang, L. (2017, Oct). Systematic defect detection methodology for volume diagnosis: A data mining perspective. In *Proc. ITC* (pp. 1–10).

Stratigopoulos, H. (2018, May). Machine learning applications in IC testing. In *Proc. ETS* (pp. 1–10).

Sumikawa, N., Nero, M., & Wang, L. (2017, October). Kernel based clustering for quality improvement and excursion detection. In *Proc. ITC* (pp. 1–10).

Sun, Z., Jiang, L., Xu, Q., Zhang, Z., Wang, Z., & Gu, X. (2013, September). Agentdiag: An agent-assisted diagnostic framework for board-level functional failures. In *Proc. ITC* (pp. 1–8).

Sun, Z., Jiang, L., Xu, Q., Zhang, Z., Wang, Z., & Gu, X. (2015, January). On test syndrome merging for reasoning-based board-level functional fault diagnosis. In *Proc. ASP-DAC* (pp. 737–742).

Tang, H., Manish, S., Rajski, J., Keim, M., & Benware, B. (2007, May). Analyzing volume diagnosis results with statistical learning for yield improvement. In *Proc. ETS* (pp. 145–150).

Tikkanen, J., Siatkowski, S., Sumikawa, N., Wang, L., & Abadir, M. S. (2014, October). Yield optimization using advanced statistical correlation methods. In *Proc. ITC* (pp. 1–10).

Tipping, M. E. (2004). Bayesian inference: An introduction to principles and practice in machine learning. In *Advanced lectures on machine learning: Ml summer schools* (pp. 41–62). Berlin, Heidelberg: Springer.

Wang, H., Poku, O., Yu, X., Liu, S., Komara, I., & Blanton, R. D. (2012, June). Test-data volume optimization for diagnosis. In *Proc. DAC* (pp. 567–572).

Wang, L. (2017, June). Experience of data analytics in EDA and test – Principles, promises, and challenges. *IEEE Transactions on Computer-Aided Design*, *36*(6), 885–898.

Wang, L.-T., Wu, C.-W., & Wen, X. (2006). *VLSI test principles and architectures: Design for testability*. San Francisco: Morgan Kaufmann.

Wang, S., & Wei, W. (2009, April). Machine learning-based volume diagnosis. In *Proc. DATE* (pp. 902–905).

Xanthopoulos, C., Sarson, P., Reiter, H., & Makris, Y. (2017, October). Automated die inking: A pattern recognition-based approach. In *Proc. ITC* (pp. 1–6).

Xue, Y., Poku, O., Li, X., & Blanton, R. D. (2013, September). Padre: Physically-aware diagnostic resolution enhancement. In *Proc. ITC* (pp. 1–10).

Ye, F., Chakrabarty, K., Zhang, Z., & Gu, X. (2015, January). Self-learning and adaptive board-level functional fault diagnosis. In *Proc. ASP-DAC* (pp. 294–301).

Ye, F., Firouzi, F., Yang, Y., Chakrabarty, K., & Tahoori, M. B. (2016, April). On-chip droop-induced circuit delay prediction based on support-vector machines. *IEEE Transactions on Computer-Aided Design*, *35*(4), 665–678.

Ye, F., Zhang, Z., Chakrabarty, K., & Gu, X. (2013, May). Board-level functional fault diagnosis using artificial neural networks, support-vector machines, and weighted-majority voting. *IEEE Transactions on Computer-Aided Design*, *32*(5), 723–736.

Ye, F., Zhang, Z., Chakrabarty, K., & Gu, X. (2014, February). Board-level functional fault diagnosis using multikernel support vector machines and incremental learning. *IEEE Transactions on Computer-Aided Design*, *33*(2), 279–290.

Zhang, Z., Chakrabarty, K., Wang, Z., Wang, Z., & Gu, X. (2011, September). Smart diagnosis: Efficient board-level diagnosis and repair using artificial neural networks. In *Proc. ITC* (pp. 1–9).

Zhang, Z., Gu, X., Xie, Y., Wang, Z., Wang, Z., & Chakrabarty, K. (2012, May). Diagnostic system based on support-vector machines for board-level functional diagnosis. In *Proc. ETS* (pp. 1–6).

Zhao, L., Goh, S., Chan, Y., Yeoh, B., Hu, H., Thor, M., … Lam, J. (2018, July). Prediction of electrical and physical failure analysis success using artificial neural networks. In *Proc. IPFA* (pp. 1–5).

How to cite this article: Pradhan M, Bhattacharya BB. A survey of digital circuit testing in the light of machine learning. *WIREs Data Mining Knowl Discov*. 2021;11:e1360. https://doi.org/10.1002/widm.1360