Measurement: Sensors 24 (2022) 100497

Contents lists available at ScienceDirect

Measurement: Sensors

journal homepage: www.sciencedirect.com/journal/measurement-sensors

Current issues and emerging techniques for VLSI testing - A review☆

Garima Thakur, Shruti Jain *, Harsh Sohal

Department of ECE, Jaypee University of Information Technology, Waknaghat, India

ARTICLE INFO

Keywords:
Digital circuit
Testing
Machine learning
Diagnosis

ABSTRACT

Circuit designers are always faced with new obstacles as a result of the persistent trend in today's nanoscale technology to follow Moore's law. The complexities inherent in the production process have increased dramatically due to the rapid downscaling of integration. Parallel to this, the complexity and unpredictability of silicon chip flaws have increased, making circuit testing and diagnosis more challenging. The amount of test data has multiplied, and the criteria governing integrated circuit testing have grown both in size and in the complexity of correlation. The modern situation provides a useful framework for investigating novel machine learning-based test solutions. In this paper, the authors examine different recent developments in this developing field in the context of digital logic testing and diagnosis.

1. Introduction

Machine learning (ML), which is armed with the strength of contemporary cloud computing and the accessibility of enormous amounts of data and storage, is at the forefront of current technological development. Since ML-based techniques can effectively solve complex problems that were thought to be intractable a decade ago, they are now widely used in modern industry. Their applications' range almost completely encompasses all fields, and they hold possibilities only constrained by human creativity. Through accurate illness diagnosis and a grasp of biological systems like gene control, which are incredibly useful in the drug discovery process, it has revolutionized the medical and pharmaceutical industries. Its redefining impact on communication, production, manufacturing, and the entertainment sector is comparable. Because many of these applications require evaluating vast amounts of historical data, machine learning (ML) is pervasive in the technological world. By utilizing relevant information, ML offers a method to turn this data into knowledge that is then used for future analyses and prediction solutions. The essential prerequisite for developing a ML engine is a sufficient volume of trustworthy data. Since the decision-making process for predicting the parameters is reduced to a simple function evaluation once the model has been trained from the data, the success of ML in all these fields can be credited to the speedy solutions it offers.

The availability of high-speed hardware and graphics processing units, which speed up the necessary computation, is a crucial factor behind the upswing in ML, especially in deep learning. In this study, the authors concentrate on the difficulties of testing digital hardware logic and talk about how ML might help with these issues. Although ML in this field is still in its infancy, there are plenty of opportunities to investigate applications of these methods and to create original solutions. Technology for digital electronics has been around for almost 50 years. Transistors were created in 1954, which opened the door to creating smaller devices and ultimately the creation of integrated circuits (ICs) in 1962. The development of complementary metal-oxide-semiconductor (CMOS) technology brought about a new paradigm for low-power circuit design. For the implementation of digital circuits with very large-scale integration, CMOS design styles are frequently employed in VLSI. There are billions of transistors on a single die in today's IC devices. In the production cycle of digital IC chips, testing for manufacturing flaws is just as important as designing because it influences dependability, price, and time-to-delivery. Effective testing is also required to calculate chip yield and reveal process variances. Over the past three decades, numerous aspects of fault modeling, detection, and diagnosis, as well as fault simulation, built-in self-test, and Design-for-Testability (DfT), have been thoroughly studied, resulting in effective test generation and fault-diagnosis algorithms, as well as testable designs [1–3]. For evaluating digital logic, several industrial tools have been created over the years. However, as IC chips have become more complicated, testing and, more specifically, diagnosis, have become increasingly difficult.

☆ Given her role as a guest editor Shruti Jain had no involvement in the peer-review of this article and has no access to information regarding its peer-review. Full responsibility for the editorial process for this article was delegated to Sudip Paul.

* Corresponding author.
E-mail address: jain.shruti15@gmail.com (S. Jain).

https://doi.org/10.1016/j.measen.2022.100497
Received 15 September 2022; Received in revised form 26 September 2022; Accepted 27 September 2022
Available online 7 October 2022
2665-9174/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

According to Moore's law, the number of transistors on a single IC chip has reached billions with today's deep-submicron technology, with the newest technology being less than 7 nm. Low technology nodes boost the yield of high-speed electronic products and decrease silicon area, but they necessitate complex production procedures. Intricate manufacturing procedures and high levels of integration have increased ICs' susceptibility to failure and introduced additional faults with more complicated behaviors. The advantage of miniaturization is thus accompanied by more recent testing difficulties. Defect localization must be carried out to identify defective chips and increase the yield of IC chips so that the process can be enhanced in subsequent manufacturing cycles. Defect/fault detection, which entails finding systematic faults before their location [4], has thus become a crucial stage in IC design and manufacture. The majority of machine learning (ML) based digital test applications are in the diagnosis field. There have been various attempts to apply ML to circuit testing in the industrial setting [5]. The lack of meaningful data is a significant impediment in this sector. Additionally, ML modeling in the subject of digital testing is still relatively new, with only a small number of research having been described in the literature thus far, in contrast to other sectors where ML techniques have been intensively investigated and successfully utilized. This survey focuses on the numerous features and modeling approaches that have been employed while walking through the various problem scenarios in the context of digital logic testing where ML techniques have been applied. The majority of the issues in this field are brand-new, and there are many opportunities to increase the efficiency of current solutions. It's crucial to remember that ML offers automated tools to solve many challenging test problems that would have been difficult to solve otherwise.

Follow of paper: Section 2 consists of the machine learning concept and its use in testing. Yield learning and diagnosis are shown in Section 3 and Section 4 consisting of ATPG, test cost, and testability issues. In Section 5, a summary, challenges and future directions is shown which is followed by conclusion in Section 6.

2. Machine learning

As the name implies, ML is a field that tries to "learn" knowledge and information from data using the computational capacity of the modern "machine," primarily computers. While its overall goal—concluding the data—is similar to that of statistical learning, it is more open to incorporating different approaches and the types of data it can be used beyond statistical learning. Its methodology also applies to computer science, and geometry, and simply mimics the biological processes of brain networks, which are still poorly understood. The types of data include unstructured data like themes and graphs, as well as simple tabular, structured data and more complicated structured data like photos and videos. Two fundamental methods of ML are used, depending on the type of data. The first strategy involves clustering, which is a type of unsupervised learning and involves examining the data to see if any patterns exist. In the second method, each data point has labels that can be used to approximate some function or model that the labels are thought to represent, allowing any ensuing unlabeled data to be given an appropriate label. Under supervised learning, Bayesian inference, k-means clustering, and spectral clustering are the frequently used methods. Decision trees, support vector machines (SVMs), artificial neural networks (ANNs), Bayesian networks, and random forests (RFs) are prominent methods for supervised learning [6]. The prevalence of supervised learning is increased by the accessibility of common tools, particularly SVMs and ANNs.

Although supervised learning is frequently favored over unsupervised methods (labels are frequently absent or challenging to collect). As a result, the approach must be based on the type of data that is provided. There are numerous prospects for supervised learning applications in the field of digital logic testing, different data sources that have been or could be perhaps applied to machine learning applications in the realm of digital electronic testing.

- Manufacturing test response
- Simulation
- Historical data on diagnosis
- Circuit Parameters
- Circuit Structure
- Physical Layout

To create a useful model or inference, we frequently don't interact with the data directly because of the following problems: (a) the data's dimensionality may be too large; (b) the data may be biased, and (c) the data may not be representative (d) They may be chaotic and boisterous. Thus, before we can make them useable, substantial preprocessing may be needed for computations based on ML. Feature engineering, or the creation of features, is one of the key steps in this path from the information at hand. Feature selection is also necessary for some circumstances. When it comes to data like graphs, a new technique called representation learning is gaining popularity [7].

2.1. ML in analog circuit testing

Analog and digital components both make up an electrical chip. The two, however, have quite different working theories, levels of complexity, and testing requirements. Compared to digital operations, analog operations are significantly more complex, and testing analog circuitry is significantly more difficult [8,9]. The signals in analog circuits, in contrast to those in digital circuits, are not discrete, making it difficult to think of an appropriate fault model that captures all mistake patterns when testing them. Additionally, they behave in a nonlinear manner, and the characteristics of the environment and the circuit greatly affect how they behave and how they get the outcome. As a result, the majority of analog circuit testing methods are parameter-based, making it challenging to develop deterministic testing procedures. In this field, a variety of statistical and ML approaches have been investigated [10–12]. The good news is that analog circuits are compact and only makeup about 10% of the device [13].

The testing of digital circuits, however, has been the subject of intensive research over many years and is today a subject that is well understood. An analog circuit uses significantly more complicated components, while a digital circuit uses much more clearly defined failure models. Additional possibilities for creating effective test strategies are provided by the automated tools for test generation, fault simulation, and DfT insertion that are readily available. But further difficulties have emerged for digital logic testing as a result of numerous technical developments [14]. In this developing field, several methods based on machine learning have recently been created.

3. Yield learning and diagnosis

Technology that scales quickly, necessitates a complex and accurate production process. Since manufacturing methods only get better with time, the initial yield (percentage of excellent chips) is poor. Thus, the yield ramp-up during volume production has become dependent on yield learning [15]. Yield learning entails comprehension, of the failures, identifying the flaws, and then putting the necessary corrective procedures in place to enhance the manufacturing process. Fig. 1 depicts the several phases of yield learning [16].

3.1. Wafer-level diagnosis

It is difficult to maintain good yield when fabricating IC chips because of the complexity of today's production process. In addition, silicon wafer manufacture has a long time cycle [12]. As a result, it's critical to spot wafer flaws as soon as possible to optimize the manufacturing process and cut down on time and yield loss. It has been

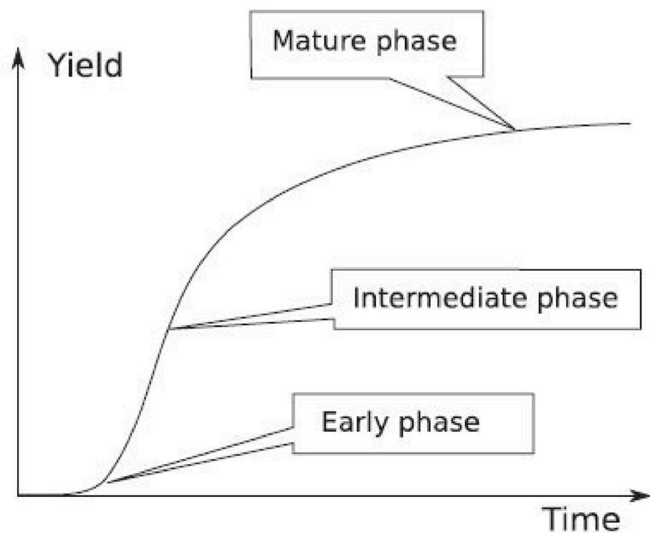

Fig. 1. Yield learning phases [16].

noted that the faults typically appear in groups in specific places on a wafer [17]. [18] proposes a kernel-based approach to identify such clusters. Die inking is a procedure used to identify dies with latent flaws. Additionally, burn-in tests are frequently used for identifying these latent flaws. However, the expense and complexity of burn-in testing make them unaffordable [19]. In [19], a way to automate this process using ML was described. During training, faulty dies that are adjacent to defective clusters in a wafer are manually inked. For example, a binary classifier is utilized during the testing phase to determine whether or not a die is flawed. One of the most popular kernels used in the literature that employs the Gaussian function for distance computation is an SVM with a radial basis function kernel. To eliminate noise, morphological processes like erosion and dilation are applied. During classification, a feature vector depending on the distance of the die from faulty clusters is utilized. Fig. 2 depicts the related flow diagram.

3.2. Fault diagnosis: preprocessing

The fault diagnostic process may be aided by the information found in the failure record of damaged chips. Because data collecting is costly and time-consuming, the whole failure data may not be available for diagnosis [20]. Poor diagnosis frequently results because just a tiny portion of the entire response data is provided. For relief to solve this issue [20], propose a method to establish the bare minimum of test results necessary for accurate diagnosis. The technique uses a binary classifier to determine when to halt the response collection operation. The classifier determines whether to go on to the following test response after collecting each test response. The features are determined by the chip's output response upon application of the final test pattern. Results for different classifiers, including KNN, SVM, and decision trees, have been presented. The research by [21] introduces a classifier to forecast the failure log's diagnostic value, the location of faults (scan-chain or functional logic), and the time of failure essential to diagnosis. To create the classifier, they applied RF and supplied a collection of features based on the failure log as illustrated in Fig. 3.

3.3. Fault diagnosis: post-processing

Despite playing a significant role in directing the PFA process, fault detection is done on an abstract level. Additionally, the reported number of candidate problems (diagnostic resolution) is typically high. The findings of fault diagnosis using ML approaches can be adjusted using a variety of strategies. They typically have two goals in mind: (a) defect identification, which entails linking the identified fault to a flaw. This is difficult, particularly if it simply considers the circuit's failure reaction [22–24] (b) enhancing diagnostic resolution, in which the possible defects are examined to further trim the set to enhance the diagnostic

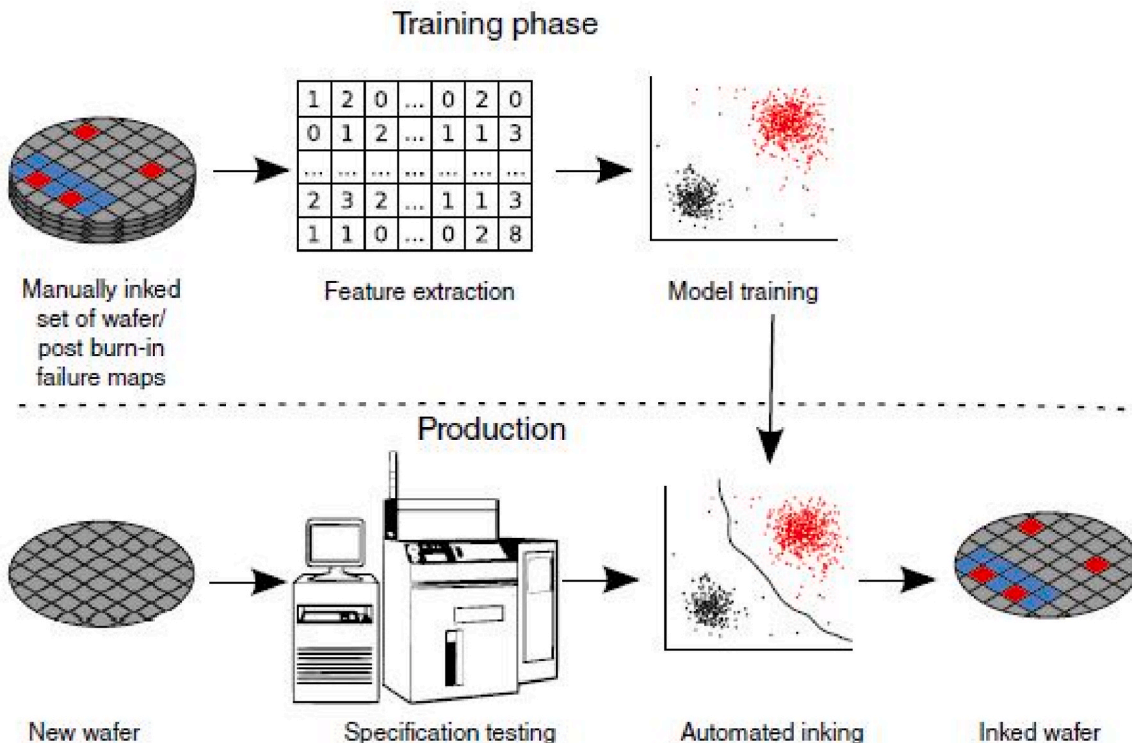

Fig. 2. Automated die-inking [19].

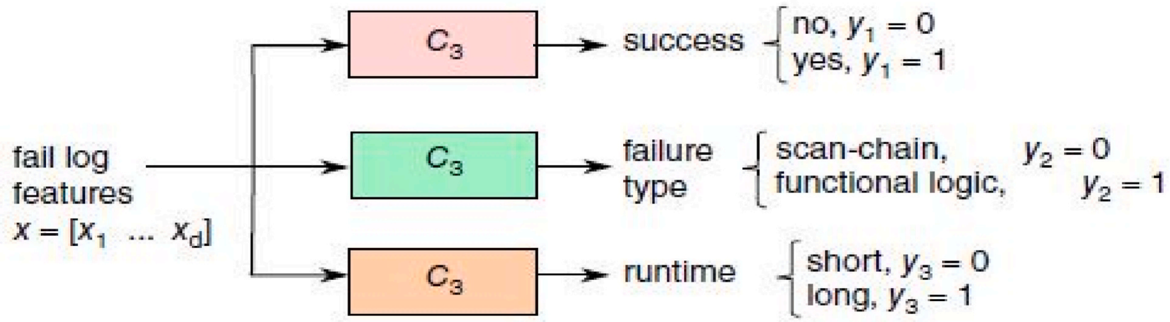

Fig. 3. Three-output classifiers where X is a feature vector with d elements, and y_1, y_2, and y_3 are discrete variables denoting the classes [21].

resolution [25]. The circuit's structure, logical information, and output response are where both techniques get the properties they employ faulty chips.

3.4. Defect identification

The issue of locating bridging flaws in a failed circuit is tackled in [23]. According to [26], this data is useful for calculating defect density and size distribution, which are necessary for yield learning. Since a bridging defect denotes a short between two signal lines, a set of bridging faults involving line A's nearby lines (B, X, and, Y) that are " (A, B), " (A, X), " (A, Y)" are taken into consideration for each candidate fault involving line A. The circuit's logical data is expressed as Boolean characteristics. For instance, the "feedback" feature examines if a structural path exists between a bridging fault's two lines. Such a channel under the fault may cause a latch or provoke an oscillating behavior that affects the test result. As a result, these locations could not be considered potential candidates for bridging faults. Other Boolean properties are applied similarly to determine if the lines drive parity gates, identical gates, or have a logical correlation. By examining the relationship between the tester output and the simulated response of the circuit in the presence of the candidate defect under various bridging fault models, test-dependent features are created [3]. A rule-based classifier and decision-tree-based classification are used to process these errors. Bridge faults are those that receive high marks for both logical and test-based fault characteristics. A decision tree is used to categorize the remaining faults after rule-based categorization to identify the non-bridge problems [27]. Fig. 4 depicts the categorization scheme's flow [23]. The training set is developed utilizing SPICE simulation and the findings of PFA's defect diagnosis.

3.5. Board-level diagnosis

Printed circuit board technology has made it possible to incorporate a variety of components, including memory, I/O, ASIC, and memory, on a single board. Therefore, diagnosis is also required at the board level. Although the individual components may pass the manufacturing test in the ATE, it has been observed that they fail the board-level functional test. The components are marked as "no issue discovered," mostly because the real testing environment differs from the ATE environment. To maintain the dependability of digital systems and for routine maintenance, this challenge in the industry needs to be carefully managed. The diagnosis of functional faults at the board level is conducted using logic. When predicting problematic components for fresh boards, the knowledge of the primary causes of failure syndromes for an initial group of boards that might be repaired is employed as training data. The failure data of the components in a test set is used to identify the symptoms. The underlying root-cause cases that are diagnosed for these disorders produce a set of features, and they act as labels in the training set. In this area, several methods based on different ML techniques have been presented, including decision trees, ANN, SVM, and vector machines [28–31]. The inputs are supplied with various syndromes in the ANN-based technique [32], and the outputs indicate the components [32]. used a collection of two-layer, single-output ANNs (Fig. 5) to address large-size board-level diagnosis issues. The output node represents a component and categorizes it as the main cause of a failure or not. The dependence of training sets on historical data, which is frequently constrained, is the main issue with the majority of ML applications in this field. In addition to having restricted access to historical data, the size of the feature vector is typically enormous due to the size of the test set, which causes overfitting during training. This was remedied by utilizing a technique known as syndrome merging to condense the feature vector [33]. Keep in mind that some syndromes

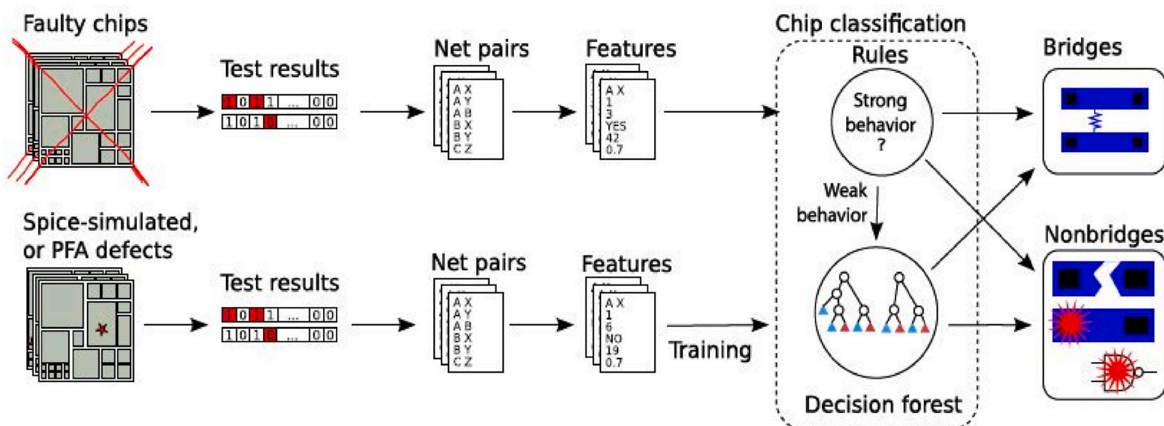

Fig. 4. Flow for the classification of bridging defects [23].

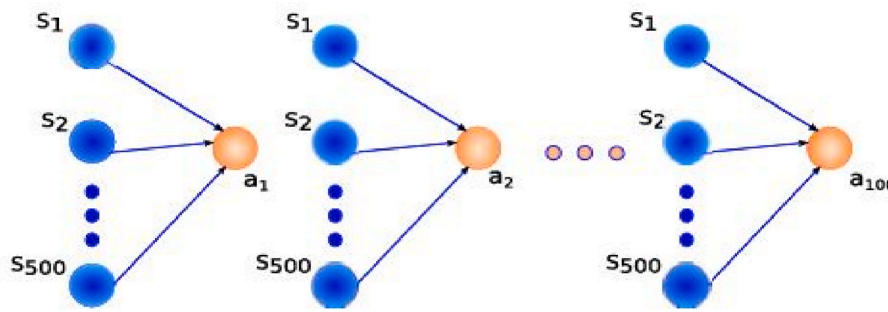

Fig. 5. An illustration of the ANN architecture used by [32].

maybe not be observable or calculable [34]. suggested another method to process the training set, which included models using naive Bayes classifiers.

4. ATPG, test cost, and testability issues

A few recent developments include ATPG and testability analysis, where ML techniques have been shown to offer unique answers. The majority of ML-based test methodologies established thus far focus on fault diagnosis in digital circuits. Applications include test compression, estimating fault coverage under unknowable (*X*) inputs, and circuit testability, in particular analysis of the time and improvement.

4.1. Test compression

The test time and test data volume are used to calculate the test cost. The use of the compressor/decompressor architecture in scan-based test environments is one technique to lower test costs. The scan chains are loaded using a pseudo-random pattern generator (PRPG), occasionally in conjunction with a decompressor. The test-response data is similarly compressed using a MISR. It has been demonstrated that, in addition to different circuit parameters, the PRPG's length significantly influences the test cost [35]. Running ATPG can exhaustively answer the pattern generator design problem, however, due to the time required, it can become impractical [35]. suggest using a predictor based on a Support Vector Regression (SVR) to address this issue. From the ATPG log file, several features are extracted, and the most appropriate features are chosen. Fig. 6 provides an example for deciding on PRPG length. two distinct predictors, one for the test's duration and the other for the volume of test data, are taught. Test costs are anticipated for each option for PRPG length, and the length with the lowest cost is chosen [35,36].

4.2. Circuit testability

Two types of ML-based research have been done in the field of circuit testability. The first seeks to determine the amount of fault coverage lost as a result of an *X*-signal (an unknown logic value) at a CUT's input. The second deals with the issue of adding test points to increase testability.

4.2.1. Analysis of X-sensitivity

Because these values cannot be calculated during ATP, the existence of sources reduces the detectability of a circuit, which results in a reduction in fault-coverage for a specific test set. Uninitialized memory cells, bus congestion, and improper operation of analog-to-digital

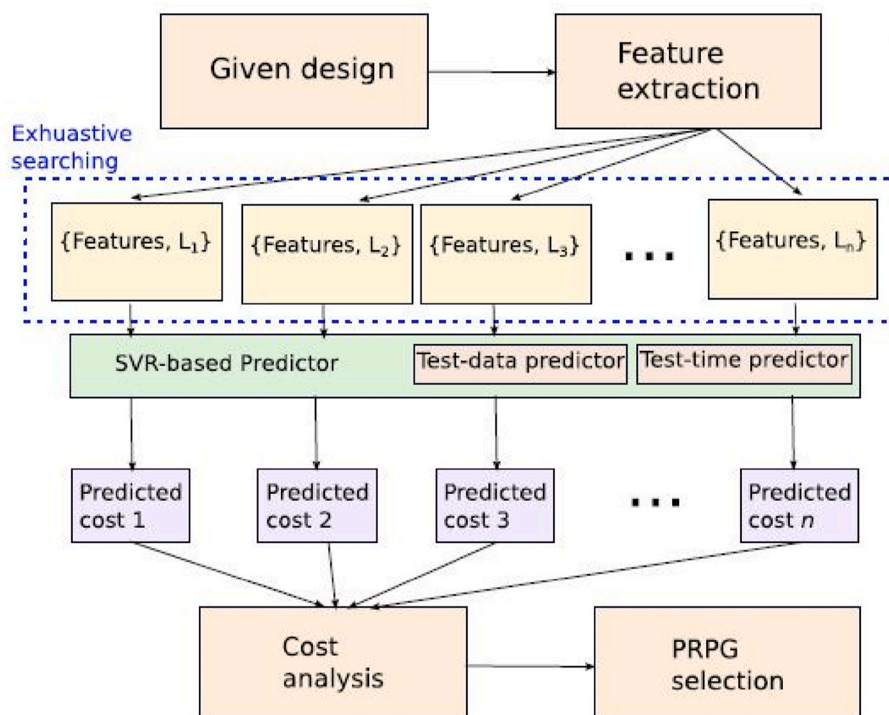

Fig. 6. Illustration of the PRPG-selection method [35].

converters are only a few examples of *X*-sources. Additionally, many design flaws that are discovered during the post-silicon validation stage manifest themselves as *X*-values. In [36], *X*-sensitivity, or the impact of *X* on the reduction in fault coverage in digital circuits, has been explored. The *X*-sensitivity of inputs in a digital circuit can be predicted quickly and accurately. The *X*-sources can be ranked using this forecast. By rewiring specific components during post-silicon validation, the inputs with high *X*-sensitivity can be taken into account for *X*-masking or deletion from the perspective of enhancing testability and also for fixing those design defects that cause *X*-values. Similar to memory cells, those that store *X*-values can be left without initialization if their impact on detection ability is minimal, saving test time. Running ATPG repeatedly is one method of calculating the detect ability loss. However, due to the lengthy computation required, it is not practicable. By [36], an SVR-based *X*-sensitivity predictor has been created. There are a few traits that have been identified that are solely dependent on the circuit's structural characteristics. Fig. 7 illustrates the three circuit partitions used to study the structural influence: *P*1, the output cone of the *P*1 is the *X*-source, *P*2 is the circuit subcircuit that affects *X* propagation, and *P*3 is the remainder of the circuit. It is simple to see that the X-source primarily affects the gates in *P*1 and *P*2, with *P*1 being more influenced than *P*2. Taking these facts into account, some innovative features based on the two partitions are offered. The percentage of sensitive gates in *P*1, the percentage of output ports in *P*1, and the number of gates that are immediately reachable from the *X*-source all affect certain aspects.

4.2.2. Test-point insertion

In [37], the issue of test-point insertion in a logic circuit was investigated from an ML perspective, and a classifier was created. This is the first instance of a test problem being handled by a deep learning-based technique. Additionally, efforts have been made to learn from circuit graphs, but this has proven difficult because ML tools whereas a graph mostly contain unstructured data are more suited for structural/vector data. To analyze graphical data [37], suggested using a neural network dubbed a Graph Convolution Network (GCN). The circuit netlist's nodes are categorized as either easy-to-observe points or difficult-to-observe points in the graphs that depict them. A node embedding method is used to express graphic aspects. Each node has several testability-related attributes that were acquired using the SCOAP tool [38]. The GCN creates an embedding for each node based on these properties and its local neighborhood data. Fig. 8 depicts the classifier's overall flow.

4.3. Timing analysis

The clock frequency of a circuit must be determined by timing analysis. Numerous static and dynamic (input pattern dependent) variables affect a circuit's timing. The input voltage that reaches the gates and, as a result, the propagation delay are both impacted by Power Supply Noise (PSN). It is one of the elements that affect the circuit's Dynamic Timing Analysis (DTA). Y created an ML method to estimate the circuit timing while accounting for the PSN effect to accelerate DTA [39]. [40] proposed utilizing SVM to estimate circuit delay caused by voltage droop.

5. Summary, challenges and future directions

In Table 1, a synopsis of the pertinent literature is provided. We'll then examine the numerous difficulties and directions for the future. Although there are numerous instances of digital logic testing where machine learning has been or might be used, these instances nevertheless seem disjointed and disorganized. The availability of sufficient data that is high quality and volumetrically sufficient is essential for the success of ML-based approaches. While some of the prospective data sources indicated, standard ML databases for IC testing have not yet been created, and as a result, their absence represents a significant barrier to the adoption of ML tools. Below are a few of the causes of this bottleneck:

a. lack of industrial time-series test data [47]: The vast majority of databases containing the failure logs gathered during the production testing of integrated circuits (ICs) as well as the associated diagnostic data are not accessible to the general public. Such information would be a valuable resource for upgrading training models and assisting with future IC-chip diagnosis procedures.

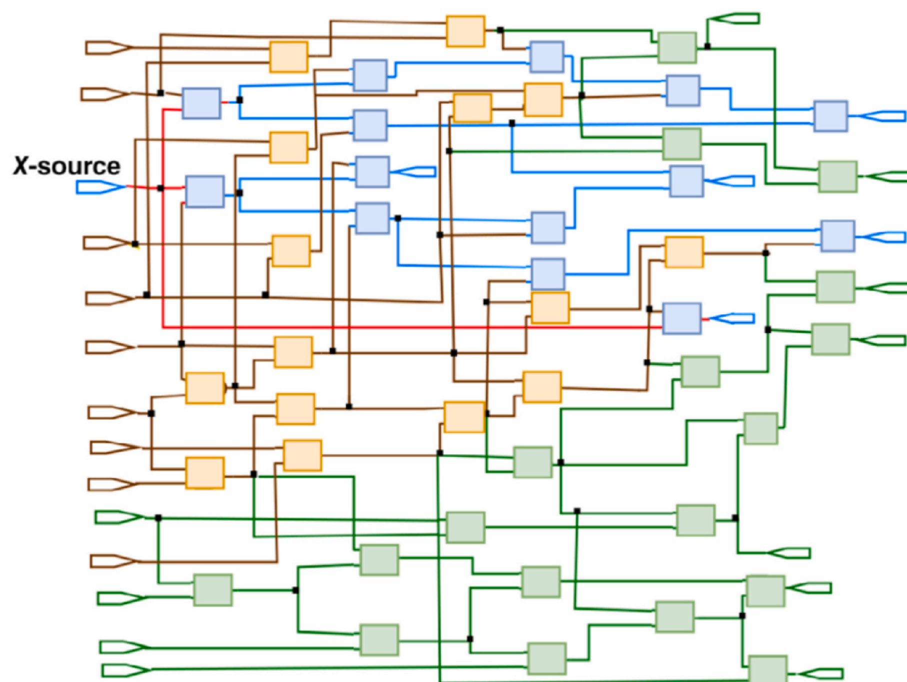

Fig. 7. A combinational circuit showing an *X*-source (blue), the three partitions *P*1 (blue), *P*2 (brown), *P*3 (green), and the gates directly fed by the *X*-source (red) [36]. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

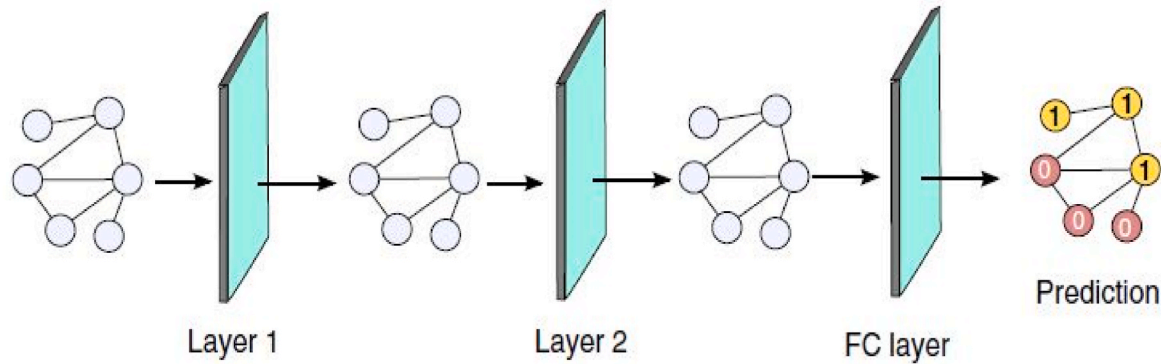

Fig. 8. Network architecture of GCN. Node embeddings are generated in Layer 1 and Layer 2. The Fully Connected (FC) third layer execute nodes classification [37].

Table 1
Comparison with state-of-art work.

Work	Problem	Data	Method
Wafer Level Diagnosis			
[18]	Identifying-Defect-Clusters	FD	Clustering
[19]	Automated-Die-Inking	Historical Data	SVM
[41]	Correction-of-Failure-and-parameters	FD	Statistical Correlation
[42]	Targeting-Hard-to-Model-Faults	FD	Bayesian Method
[43]	–	SD	Multi-Stage ANN
Fault-Diagnosis Pre-Processing			
[20]	Regulation-of test-data-volume	FD	Classification
[21]	Inferring-Diagnostic-Efficiency	FD	Random Forest
Fault-Diagnosis Post-Processing			
[24]	FI: Defect-Classification	SD	ANN
[23]	FI: Identifying-bridging-defects	SD and FD	Decision Tree
[22]	FI: Transient-and-intermittent-faults	SD	Bayesian Network
[25]	Improving-Diagnostic-Resolution	SD	SVM
Volume Diagnosis			
[44]	VD-of-unmodeled-faults	SD	SVM
[45]	VD-for-root-cause-identification	Volume FD	Bayesian network, MLE
[46]	Identification-of-systematic-defects	FD	Clustering (furthest neighbor)
Board Level Diagnosis			
[29–31]	Fault-isolation	Historical Data	SVM/ANN/Decision tree
[33]	Syndrome-merging	–	–
[34]	Missing-syndrome-computation	–	Naïve Bayes
Test Compression			
[35]	Test-cost-optimization	SD	SVR
Circuit Testability			
[36]	Prediction-of-X-sensitivity	SF and SD	SVR
[37]	Test-point-insertion	SFs, SCOAP, SD	GCN
Timing analysis			
[39]	Based-on-PSN	SD	Multiple tools

Note: FD-Failure Data, FI- Fault Identification, SD-Simulated Data, SF-Structural Feature, VD-Volume Diagnosis.

b. Simulated data production is a laborious procedure that takes a lot of time [35,36]. A broad collection of simulated data would be a useful data source and could benefit ML-based solutions.
c. Lack of baseline: According to [48], there are no benchmark circuits for analyzing ML approaches to test problems. However, it is possible to extract the circuit structure from the netlist of the benchmark circuits. They may provide information that can be used to derive both structural and functional aspects. ML studies are not intended to use the current benchmark suites [49–56]. There is a dearth of variety and quantity among them. Such data must be produced using a variety of big benchmark circuits with different connecting structures and functionalities. A repository of this kind could be created by unbiased synthesis or by gathering commercial circuits.
d. Extraction of features from circuits: It is clear that feature engineering is an important step in the majority of settings. Automatic feature extraction from circuit netlists is urgently required for digital logic testing. Additionally, it is necessary to prevent over-fitting and remove noisy data (in the case of output response data from a CUT). Be aware that circuit data are not usually presented in a comprehensible or orderly manner (e.g., logical interconnection or physical layout data). Consequently, a lot of time and effort is wasted in this process. Additionally, the features are rarely reusable since a new feature set is typically needed for each distinct test scenario. Automated feature extraction from circuits is desperately needed because human feature engineering is a laborious procedure. Even though there have been numerous similar projects in other domains like pattern recognition and picture analysis, this is still an open subject in the area of logic testing. Convolution techniques-based deep learning has been effectively used on image and video data where the underlying features are implicitly used during learning and testing rather than being explicitly retrieved or selected. Additionally, new approaches to managing unstructured data, such as representation learning, have been created. These approaches can prepare the data for the direct use of ML tools. The work put out by [37], which introduced GCN for node embedding of a graph, representing the netlist of a circuit, has pioneered efforts in this regard.

6. Conclusion

In this paper, authors have examined several issues that come up during the testing and diagnosis of VLSI circuits and where ML has been successfully used. In managing the intricacy of the issue, they have surpassed conventional heuristic-based approaches and delivered believable solutions far faster. Future acceptance of ML approaches to other chip testing issues will be sparked by solutions to the difficulties of data generation and automated feature engineering. There is still room to generate data and develop representation approaches for digital circuits, which will advance both academic and commercial research on ML-guided tests. ML-based testing requires either a repair or reconfiguration mechanism, which may provide redundancy in terms of area overhead, but may provide a fast prediction of anomalies in a system with robustness and resiliency.

Funding information

The authors declare no funding for this research.

Availability of data and material

Not applicable.

Code availability

Not applicable.

CRediT authorship contribution statement

Garima Thakur: Idea, Conceptualization, and Implementation. **Shruti Jain:** Writing – review & editing. **Harsh Sohal:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

Not applicable.

References

[1] M. Abramovici, M.A. Breuer, A.D. Friedman, Digital Systems Testing and Testable Design, IEEE Press, NJ, 2002.
[2] M. Bushnell, V. Agrawal, Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits, Springer Publishing Company, New York, 2005.
[3] L.-T. Wang, C.-W. Wu, X. Wen, VLSI Test Principles and Architectures: Design for Testability, Morgan Kaufmann, San Francisco, 2006.
[4] A.S. Mutschler, Yield Ramp Challenges Increase, 2014, December. https://semiengineering.com/yield-ramp-challenges-increase/.
[5] M. Lapedus, Finding Defects in Chips with Machine Learning, 2019, March. https://semiengineering.com/finding-defects-with-machinelearning/.
[6] T. Hastie, R. Tibshirani, J. Friedman, in: The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2nd ed., Springer, New York, NY, 2009. New York.
[7] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013, August) 1798–1828.
[8] A.A. Hatzopoulos, Analog circuit testing, in: Proc. IMSTW, 2017, July, pp. 1–6.
[9] L.S. Milor, A tutorial introduction to research on analog and mixed-signal circuit testing, IEEE Transact. Comp. Aided-Des. 45 (10) (1998, October) 1389–1407.
[10] K.M. Butler, A. Nahar, W.R. Daasch, What we know after twelve years developing and deploying test data analytics solutions, in: Proc. ITC, 2016, November, pp. 1–8.
[11] H. Stratigopoulos, Machine learning applications in IC testing, in: Proc. ETS, 2018, May, pp. 1–10.
[12] L. Wang, Experience of data analytics in EDA and test – principles, promises, and challenges, IEEE Transact. Comp. Aided Des. 36 (6) (2017, June) 885–898.
[13] M. Karmani, C. Khedhiri, B. Hamdi, Design and test challenges in nano-scale analog and mixed CMOS technology, in: VLSICS, 2, 2011.
[14] S. Kundu, T.M. Mak, R. Galivanche, Trends in manufacturing test methods and their implications, in: Proc. ITC, 2004, October, pp. 679–687.
[15] R. Aitken, Yield learning perspectives, IEEE Design. Test Comp. 29 (1) (2012, February) 59–62.
[16] C. Hora, R. Segers, S. Eichenberger, M. Lousberg, An effective diagnosis method to support yield improvement, in: Proc. ITC, 2002, October, pp. 260–269.
[17] M.P. Ooi, E.K.J. Sim, Y.C. Kuang, L. Kleeman, C. Chan, S. Demidenko, Automatic defect cluster extraction for semiconductor wafers, Proc. IMTCP (2010, May) 1024–1029.
[18] N. Sumikawa, M. Nero, L. Wang, Kernel based clustering for quality improvement and excursion detection, in: Proc. ITC, 2017, October, pp. 1–10.
[19] C. Xanthopoulos, P. Sarson, H. Reiter, Y. Makris, Automated die inking: a pattern recognition-based approach, in: Proc. ITC, 2017, October, pp. 1–6.
[20] H. Wang, O. Poku, X. Yu, S. Liu, I. Komara, R.D. Blanton, Test-data volume optimization for diagnosis, Proc. DAC (2012, June) 567–572.
[21] Q. Huang, C. Fang, S. Mittal, R.D.S. Blanton, Improving diagnosis efficiency via machine learning, in: Proc. ITC, 2018, October, pp. 1–10.
[22] L.R. Gomez, A. Cook, T. Indlekofer, S. Hellebrand, H. Wunderlich, Adaptive Bayesian diagnosis of intermittent faults, J. Electron. Test.: Theory and Applications 30 (5) (2017) 527–540.
[23] J.E. Nelson, W.C. Tam, R.D. Blanton, Automatic classification of bridge defects, in: Proc. ITC, 2010, November, pp. 1–10.
[24] L.R. Gómez, H. Wunderlich, A neural-network-based fault classifier, in: Proc. ATS, 2016, November, pp. 144–149.
[25] Y. Xue, O. Poku, X. Li, R.D. Blanton, Padre: physically-aware diagnostic resolution enhancement, in: Proc. ITC, 2013, September, pp. 1–10.
[26] J.E. Nelson, T. Zanon, R. Desineni, J.G. Brown, N. Patil, W. Maly, R.D. Blanton, Extraction of defect density and size distributions from wafer sort test results, Proc. DATE 1 (2006, March) 1–6.
[27] L. Rokach, O. Maimon, Data Mining with Decision Trees: Theory and Applications, 2nd ed., World Scientific, River Edge, NJ, 2014.
[28] F. Ye, Z. Zhang, K. Chakrabarty, X. Gu, Board-level functional fault diagnosis using multikernel support vector machines and incremental learning, IEEE Transact. Comp. Aided Des. 33 (2) (2014, February) 279–290.
[29] Z. Sun, L. Jiang, Q. Xu, Z. Zhang, Z. Wang, X. Gu, Agentdiag: an agent-assisted diagnostic framework for boardlevel functional failures, in: Proc. ITC, 2013, September, pp. 1–8.
[30] F. Ye, K. Chakrabarty, Z. Zhang, X. Gu, Self-learning and adaptive board-level functional fault diagnosis, Proc. ASPDAC (2015, January) 294–301.
[31] F. Ye, Z. Zhang, K. Chakrabarty, X. Gu, Board-level functional fault diagnosis using artificial neural networks, supportvector machines, and weighted-majority voting, IEEE Transact. Comp. Aided Des. 32 (5) (2013, May) 723–736.
[32] Z. Zhang, K. Chakrabarty, Z. Wang, Z. Wang, X. Gu, Smart diagnosis: efficient board-level diagnosis and repair using artificial neural networks, in: Proc. ITC, 2011, September, pp. 1–9.
[33] Z. Sun, L. Jiang, Q. Xu, Z. Zhang, Z. Wang, X. Gu, On test syndrome merging for reasoning-based board-level functional fault diagnosis, Proc. ASP-DAC (2015, January) 737–742.
[34] S. Jin, F. Ye, Z. Zhang, K. Chakrabarty, X. Gu, Efficient board-level functional fault diagnosis with missing syndromes, IEEE Transact. Comp. Aided Des. 35 (6) (2016, June) 985–998.
[35] Z. Li, J.E. Colburn, V. Pagalone, K. Narayanun, K. Chakrabarty, Test-cost optimization in a scan-compression architecture using support-vector regression, in: Proc. VTS, 2017, April, pp. 1–6.
[36] M. Pradhan, B.B. Bhattacharya, K. Chakrabarty, B.B. Bhattacharya, Predicting X-sensitivity of circuit-inputs on testcoverage: a machine-learning approach, IEEE Transact. Comp. Aided Des. 38 (12) (2019, December) 2343–2356.
[37] Y. Ma, H. Ren, B. Khailany, H. Sikka, L. Luo, K. Natarajan, B. Yu, High performance graph convolutional networks with applications in testability analysis, Proc. DAC (2019, June) 1–6.
[38] L.H. Goldstein, E.L. Thigpen, SCOAP: sandia controllability/observability analysis program, Proc. DAC (1980, June) 190–196.
[39] Y. Liu, C. Han, S. Lin, J.C. Li, PSN-aware circuit test timing prediction using machine learning, IET Comput. Digital Tech. 11 (2) (2017) 60–67.
[40] F. Ye, F. Firouzi, Y. Yang, K. Chakrabarty, M.B. Tahoori, On-chip droop-induced circuit delay prediction based on support-vector machines, IEEE Transact. Comp. Aided Des. 35 (4) (2016, April) 665–678.
[41] J. Tikkanen, S. Siatkowski, N. Sumikawa, L. Wang, M.S. Abadir, Yield optimization using advanced statistical correlation methods, in: Proc. ITC, 2014, October, pp. 1–10.
[42] Y. Huang, B. Benware, R. Klingenberg, H. Tang, J. Dsouza, W. Cheng, Scan chain diagnosis based on unsupervised machine learning, Proc. ATS (2017, November) 225–230.
[43] M. Chern, S.-W. Lee, S.-Y. Huang, Y. Huang, G. Veda, K.-H.H. Tsai, W.-T. Cheng, Improving scan chain diagnostic accuracy using multi-stage artificial neural networks, in: Proc. ASPDAC, ACM, New York, NY, 2019, pp. 341–346.
[44] S. Wang, W. Wei, Machine learning-based volume diagnosis, in: Proc. DATE, 2009, April, pp. 902–905.
[45] W. Cheng, Y. Tian, S.M. Reddy, Volume diagnosis data mining, in: Proc. ETS, 2017, May, pp. 1–10.
[46] L.M. Huisman, M. Kassab, L. Pastel, Data mining integrated circuit fails with fail commonalities, in: Proc. ITC., 2004, October.
[47] M. Rehani, D. Abercrombie, R. Madge, J. Teisher, J. Saw, ATE data collection – a comprehensive requirements proposal to maximize ROI of test, in: Proc. ITC, 2004, Oct, pp. 181–189.
[48] L.I.R. Gómez, Machine Learning Support for Logic Diagnosis, University of Stuttgart, 2017 (Unpublished doctoral dissertation).
[49] C. Albrecht, IWLS 2005 benchmarks (tech. Rep.). Automatic test equipment (ATE) and production test, in: Integrated Circuit Test Engineering: Modern Techniques, 2006, Springer, London, 2005, pp. 257–266.
[50] F. Brglez, D. Bryan, K. Kozminski, Combinational profiles of sequential benchmark circuits, in: Proc. ISCAS, 1989, pp. 1929–1934.
[51] F. Brglez, H. Fujiwara, A neutral netlist of 10 combinational benchmark circuits and a target translator in Fortran, in: Proc. ISCAS, 1985, pp. 695–698.
[52] F. Corno, M. Reorda, G. Squillero, RT-level ITC'99 benchmarks and first ATPG results, IEEE Design. Test Comp. 17 (3) (2000, July) 44–53.
[53] G. Thakur, H. Sohal, S. Jain, A novel parallel prefix adder for optimized Radix-2 FFT processor, Multidimens. Syst. Signal Process. 32 (2021) 1041–1063.

[54] G. Thakur, H. Sohal, S. Jain, A novel ASIC-based variable latency speculative parallel prefix adder for image processing application, Circ. Syst. Signal Process. 40 (2021) 5682–5704.

[55] S. Roy, S.K. Millican, V.D. Agrawal, Special session – machine learning in test: a survey of analog, digital, memory, and RF integrated circuits, in: 2021 IEEE 39th VLSI Test Symposium (VTS), 2021, pp. 1–14.

[56] A. Calhoun, E. Ortega, F. Yaman, A. Dubey, A. Aysu, Hands-on teaching of hardware security for machine learning, in: Proceedings of the Great Lakes Symposium on VLSI 2022, 2022, pp. 455–461.