Efficient Low Cost Alternative Testing of Analog Crossbar Arrays for Deep Neural Networks

Kwondo Ma, Anurup Saha, Chandramouli Amarnath and Abhijit Chatterjee
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA 30332
Email: {kma64, asaha74, chandamarnath}@gatech.edu, abhijit.chatterjee@ece.gatech.edu

***Abstract*—Analog crossbar arrays have recently attracted significant attention due to their usefulness for deep neural net (DNN) computations with ultra-low power consumption. However, recent studies have shown that DNNs implemented with such crossbar arrays suffer from as high as 30% degradation in performance due to the effects of manufacturing process variability effects resulting in degradation of their functional safety. One way to test these DNNs is to apply an exhaustive set of test images to each device to ascertain its performance. This is expensive and time-consuming. We propose an alternative test scheme in which a small subset of test images is applied to each DNN and the classification accuracy of the DNN is predicted directly from observation of the final layer outputs of the network. This saves test cost while allowing binning of DNNs for performance. Experimental results for a variety of test cases are presented and show test efficiency improvements of 10.3X over testing with the exhaustive test image set.**

I. Introduction

Analog deep neural networks (DNNs) offer orders of magnitude lower power consumption than corresponding digital architectures and are attractive for battery-powered applications. However, RRAM crossbar-based neural networks [1] suffer from parametric manufacturing process variability effects resulting in as much as 30% degradation in performance (e.g., classification accuracy) [2], [3]. Traditionally, determining the classification accuracy of DNNs requires the (expensive) application of an exhaustive set of test images to each device for pass-fail classification. Consequently, a post-manufacture test methodology is needed that can efficiently test DNN devices in the presence of manufacturing process variability effects, with *minimum test effort* and *high test coverage*. In this context, we propose a machine-learning assisted *alternative test strategy* for analog crossbar based DNNs in which *a small subset of test images is applied to the DNN-under-test and its classification accuracy is predicted from its response to the applied test images using a trained regressor*.

Prior Work: Of key relevance to the work reported in this research is prior work on alternative testing of analog mixed-signal (AMS) circuits and systems [4], [5]. In this approach, transient test stimulus is optimized in such a way that the response of the device-under-test (DUT) to the applied stimulus bears strong correlation to its performance specifications under expected process variability statistics. Consequently, the DUT specifications can be predicted directly from the observed DUT test response using trained regression models. While we adopt a test strategy for DNNs inspired by alternative test of AMS circuits, there are key differences. Crossbar arrays contain large numbers of analog devices as opposed to the comparatively fewer number of transistors in AMS circuits resulting in relatively *very large problem dimensionality*. Second, DNN operation is *highly non-linear*. Small changes in system parameters or inputs can cause a DNN to misclassify images. Use of subsets of test images to test deep learning hardware accelerators [6] is studied for hard faults in systolic array datapaths and control logic. It is seen that DNN accuracy can drop to 8% from 98% in the presence of such faults and that 93% fault coverage can be obtained with just 0.1% of the test dataset. However, this work is not easily scalable to analog crossbar arrays under ”continuous” multi-parameter variations across large numbers of memristor devices. In [7], a misclassification driven training algorithm is used to efficiently identify functionally critical faults in memristive crossbar arrays. The work addresses stuck-on and stuck-off failures in RRAM devices and does not investigate parametric variability effects. Of interest also, is the work of [8] where stochastic noise is added to DNN training parameters to enhance the robustness of the network to crossbar device parameter variations. The work reported in this research is orthogonal to and can be used over and above such training techniques to reduce analog DNN test complexity under manufacturing variability effects.

Key contributions of this research: A novel methodology for alternative testing of RRAM analog crossbar arrays using a compact set of test images for DNNs is developed. The *accuracy of the DNN is predicted directly from the ensemble of DNN test responses to the set of test images*. Depending on variability statistics and the test acceptance threshold, 3X - 10X speedup in test effort is achieved as compared to testing each device with an exhaustive set of test images. The method allows *performance binning* of devices with little extra test effort. Further, the proposed test methodology is *adaptive and automatically recalibrates itself to respond to manufacturing process statistics, process shifts and hard defects*. Hard defects are treated as extreme cases of parametric deviations.

II. Preliminaries

The proposed alternative test approach is illustrated in Fig. 1a. In this paper, we use the terms *RRAM-based DNNs* and *DNN Under Test* (short for “device”), interchangeably. This test set is ideally selected in such a way that the *statistical correlation* between the ensemble of DUT responses to the applied test image subset and the classification accuracy of

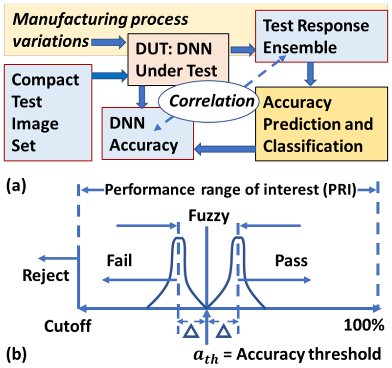
Manufacturing process variations
DUT: DNN Under Test
Test Response Ensemble
Compact Test Image Set
Correlation
Accuracy Prediction and Classification
DNN Accuracy
(a)
Performance range of interest (PRI)
Fuzzy
Fail
Pass
Reject
Cutoff
100%
Δ
Δ
(b)
a_{th} = Accuracy threshold

Fig. 1: Alternative testing of RRAM-based DNNs.

the DUT is *maximized*. In the presence of such correlation in Fig. 1a, it is possible to *directly predict DNN classification accuracy using a trained regressor* that maps the observed ensemble of test responses to the DNN accuracy.

In general, the accuracy prediction has error range of ±Δ as shown in Fig. 1b with error statistics modeled by a Gaussian with estimated mean and variance σ, $\Delta = k\sigma$, for appropriate value of k. The *accuracy threshold* a_{th} of Fig. 1b is the minimum value of DNN accuracy acceptable for a "good" device. However, for binning purposes, we desire to predict DNN accuracy upto a *cutoff accuracy* below the accuracy threshold. DNN devices with accuracy below the cutoff accuracy are rejected outright. The range of DNN accuracy from cutoff to 100% is called the *performance range of interest* (PRI). Based on this and on the predicted accuracy of the DNN as well as knowledge of Δ, any device with predicted accuracy less than $a_{th} - \Delta$ is classified as *fail* and any device with predicted accuracy greater than $a_{th} + \Delta$ is classified as *pass*. Devices with predicted accuracy in-between the pass and fail categories above are classified as *fuzzy*.

III. Overview

The proposed alternative testing methodology and tools for analog RRAM crossbar arrays for DNNs is shown in Fig. 2a. We first select a subset of images from the entire test dataset with the maximum *diversity of responses*, implicitly increases the correlation of Fig. 1, to a set of DNN devices sampled from the space of device manufacturing process variations. The ensemble of responses R of the crossbar DNN to this subset of images is passed to an outlier detector (block 3 of Fig. 2) that is used to determine if the ensemble statistics of the DUT response resembles the statistics of its own training set of DNN devices. Initially, the outlier detector is trained to recognize the ensemble of output response statistics of a set of "training devices" to the applied image set. Any device with ensemble response not conforming to such statistics is classified as an "outlier" and subject to standard testing procedures (block 5 of Fig. 2). All other devices are passed to a performance classifier (block 4 of Fig. 2). This uses a trained regressor to predict the performance of the DNN from R. The training set of devices for the regressor is always selected to be identical to the training set of devices for the outlier detector. In this way, we ensure that the performance of any DNN device that is not an outlier can be accurately predicted by the regressor. The output of the regressor module consists of the mean and variance of the predicted performance of the DUT and is used to modulate Δ in Fig. 1b. This is used to classify each device as *pass, fail or fuzzy* as described earlier.

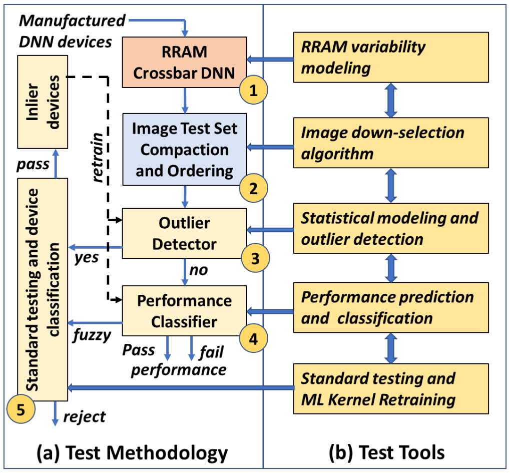
Manufactured DNN devices
RRAM Crossbar DNN
1
Inlier devices
retrain
Image Test Set Compaction and Ordering
2
pass
Outlier Detector
yes
3
no
Standard testing and device classification
Performance Classifier
fuzzy
4
Pass
fail
performance
5
reject
RRAM variability modeling
Image down-selection algorithm
Statistical modeling and outlier detection
Performance prediction and classification
Standard testing and ML Kernel Retraining
(a) Test Methodology
(b) Test Tools

Fig. 2: Alternative testing methodology and tools.

All *fuzzy* devices are kicked back to standard testing procedures to reduce the uncertainty of performance prediction close to the performance acceptance threshold for further testing. Finally, the standard testing and device classification module of Fig. 2 uses the entire image test dataset for direct determination of DNN device performance. Devices outside the PRI are rejected. The remaining devices within the PRI, called *inlier devices*, are added to the current training set of the outlier detector and performance predictor to retrain both with the expanded set of training devices in batches of 100-250 devices for improving corresponding test effort over time. The proposed DNN alternative test approach is designed to *automatically detect, learn and compensate for manufacturing process shifts* as and when they occur over time. Fig. 2b shows test tools that have been developed to enable the proposed alternative test methodology.

IV. Proposed Alternative Testing

A. RRAM variability modeling

Vector Matrix Multiplications (VMM) in DNNs can be mapped to RRAM crossbars, where the crossbar receives inputs from Digital to Analog Converters (DACs) and the outputs of the crossbar are converted to the digital domain using Analog to Digital Converters (ADCs). Despite their energy efficiency, RRAM crossbars suffer from a range of nonidealities which degrade inference accuracy of RRAM-based DNN accelerators [2], [3]. The goal of the variability modeling framework is to *quantify the impact of nonidealities in DACs, process variations in a crossbar and operating temperature* by transforming an ideal weight matrix W_i into a nonideal matrix W_{ni}. To design the crossbar, we use HfO$_x$ based RRAM devices [9], which have a Low Resistance State ($R_{LRS} = 50k\Omega$) and a High Resistance State

($R_{HRS} = 1M\Omega$). The effect of process variations on the conductance of RRAM devices can be modeled by Gaussian conductance distributions. Nonidealities of DACs are modeled by perturbing the output voltages from their ideal values. The range of operating temperature is between 273K to 373K. The impact of derived nonideal voltages and process/temperature variations on the dot product computation of the crossbar is evaluated simultaneously. Finally, nonlinear quantization effects are considered in ADC models.

Now, variability modeling is performed in three main steps as follows to obtain the process-perturbed nonideal weights.

Step 1: We introduce *systematic weight perturbation coefficient* p_{ij}^{sys} and *random weight perturbation coefficient* p_{ij}^{rand} to model the effects of systematic and random RRAM conductance variations on the ideal weight values used in vector matrix multiplications. A distribution of such coefficient values is generated from Monte-Carlo SPICE simulation of an RRAM cell under assumed process statistics.

Step 2: We derive *weight perturbation coefficient* p_{ij} as a weighted sum of the systematic and random weight perturbation coefficients as:

$$p_{ij} = \alpha p_{ij}^{sys} + (1 - \alpha) p_{ij}^{rand} \tag{1}$$

where α is a process-calibrated parameter, representing the percentage contribution of systematic variability to overall variability effects.

Step 3: An ensemble of weight matrices corresponding to process-perturbed RRAM devices is generated by transforming the ideal weight matrix W_i to W_{ni}. Each element of the ideal weight matrix W_i is scaled by the corresponding weight perturbation coefficient p_{ij} as,

$$W_{ni} = \begin{bmatrix} w_{11} \cdot p_{11} & \dots & w_{1N} \cdot p_{1N} \\ \vdots & \ddots & \vdots \\ w_{N1} \cdot p_{N1} & \dots & w_{NN} \cdot p_{NN} \end{bmatrix} = W_i \circ P$$

where $\circ$ denotes the element-wise multiplication and P represents the matrix of weight perturbation coefficients. Finally, process-perturbed RRAM-based DNN is generated by incorporating the nonideal weights back to the model.

B. Image Down-selection

The objective of image down-selection is to identify a *compact subset of test images* of reduced size as opposed to the entire test dataset that can be used to most efficiently predict the classification accuracy of a process-perturbed DNN. A *classification matrix*, which represents the distribution of classification outcomes for DNNs under the RRAM variability modeling, is used for image down-selection. To obtain the classification matrix, a benchmarking subset of RRAM-based models is exhaustively tested across the entire test dataset. For each model, the results of image classification across the test dataset are constructed as a row vector in binary, with zero when an applied image is misclassified and one for correctly classified. For M images in the test dataset and N benchmarking devices, an $N \times M$ dimensional classification matrix is obtained. The column vector of the matrix indicates a sequence of classification outcomes across every device for a single image. This is defined as an *image vector*. Selection of the compact test subset is done using *agglomerative hierarchical clustering* [10] of these image vectors. For a selected number of clusters, an image vector per cluster is selected to be in the compact image test subset.

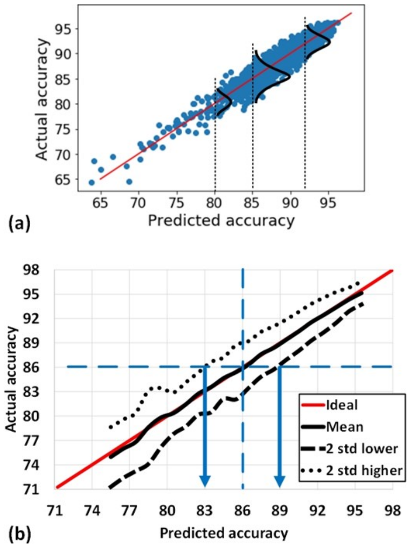

Fig. 3: Example of performance prediction and classification.

C. Statistical modeling and Outlier Detection

A vector of extracted features from the final layer of the DNN under application of the compact test image subset is defined as the *signature* vector of a DUT. The goal of our study is to *predict DNN accuracy from its signature vector*. To do so, we analyze the statistical distribution of high-dimensional signature vectors in advance. Statistical modeling for outlier detection is performed to analyze the signature vector space. An Elliptic Envelope (EE) outlier detector [11] is fitted to the high dimensional signature vectors in order to detect anomalous DUTs. The Elliptic Envelope fits an ellipse around the data that contains the majority of given signatures. Signature vectors outside the ellipse are considered *outliers* and corresponding DUTs are subjected to a standard testing.

D. Performance prediction and classification

A classification module to distinguish good/bad RRAM-based DNNs utilizes the prediction of DNN classification accuracy from the signature vector using a trained regressor.

Performance Prediction: A multivariate regression spline based regressor (MARS) [12], is used to learn the relationship between the DNN signature vector and DNN classification accuracy. Fig. 3 shows a performance prediction scatterplot, predicted values $\widehat{a}$ against actual accuracy a with correlation coefficient = 0.95, using RRAM-based CNNs on the MNIST dataset with random process variations. Of the 1000 benchmarking subset of process-perturbed ConvNet models, 80% of devices are used for regressor training and 20% of devices are used for regressor validation. The size of compact test image subset is 300 out of 10K test images.

Performance Classification: The mean and variance of the regressor prediction error across the range of predicted accuracy values, are computed as,

$$E_j = \{e | e = a_i - \widehat{a_i},\ \widehat{a_i} \in [l_j, h_j]\} \sim \mathcal{N}(\mu_j, \sigma_j)$$

where E_j is j-th set of Gaussian prediction error and $[l_j, h_j]$ represents the range of the j-th sub-interval over which the error statistics is computed across the PRI.

From the prediction result shown in Fig. 3a, a prediction error model shown in Fig. 3b is obtained, where the mean of the predicted accuracy (black solid) follows the ideal regression (red). The two dotted lines represent the 2σ decision boundaries for DNN accuracy prediction. To classify a device as "good", its predicted value, taking into account the prediction uncertainty, must lie above the required accuracy threshold a_{th} by more than a specific margin. Gaussian confidence intervals drawn from Student's t test [13] are used to make this decision based on the mean μ_x and standard deviation σ_x of the distribution around the predicted accuracy. If this distribution indicates that model accuracy is greater than a_{th} by an acceptable margin, i.e. $a_{th} \leq \mu_x - k\sigma_x$ where k is a confidence interval parameter of classification, the model is classified as "good". Similarly if the accuracy is less than a_{th} by an acceptable confidence margin, i.e. $a_{th} \geq \mu_x + k\sigma_x$, the device is classified as "bad". If there is insufficient confidence, i.e. $\mu_x - k\sigma_x < a_{th} < \mu_x + k\sigma_x$, the model requires further testing. For instance in Fig. 3b, the accuracy threshold of testing is 86%. A device is *passed* if its accuracy is above the threshold and *failed* below it. Based on the derived statistical model for predicted accuracy, any DUT with a prediction below 83% and above 89% can be confidently classified as bad and good devices, respectively. The range (83%, 89%) is defined as the *uncertain range*. Devices within this range (i.e., *fuzzy* devices) must be subjected to standard testing.

E. Standard testing and Machine Learning kernel retraining

In Block 5 of Fig. 2, *outlier* and *fuzzy* devices are tested using standard testing procedures to get the precise accuracy. The results of standard testing of remaining outlier and fuzzy devices combined with prior training data are used to retrain the ML kernels, outlier detector and regressor.

V. Experimental methodology

In this section, we describe the experimental methodology and setup used to evaluate the proposed alternative testing. Two metrics are used for evaluation: (1) *test speedup* and (2) *test quality*. Test speedup is defined as a ratio of N_{ST}/N_{AT} where N_{ST} and N_{AT} represent the total number of applied images to test a device under *Standard Testing* and *Alternate Testing*, respectively. Therefore, the test speedup demonstrates a rate of test efficiency of proposed testing over standard testing in the aspect of a computational cost. Test quality is defined as the percentage of false positives (bad devices that passed the test) among all classified devices.

The proposed testing framework is evaluated on benchmark DNN applications with a variety of datasets. Table I provides details of the models, dataset, and corresponding variability modeling. The term *systematic* in Table I represents the percentage contribution of systematic variabilty within the total variability effects and is given by the value of α in Equation 1 (see Section IV-A). The DNN models used are written in PyTorch and trained assuming ideal weights. Variability modeling framework is performed in HSPICE using PTM model [18] at 65 nm CMOS technology transforming the ideal weights of each DNN layer to a corresponding nonideal weight matrix. We generate equivalent process-perturbed DNN by incorporating these nonideal weights back to the model to evaluate our testing scheme. For all benchmark DNN applications, 16 bit precision is used for computations.

TABLE I: Benchmark DNN applications

Dataset	Network	#Conv	#FC	Systematic
CIFAR-10 [14]	MobileNet [15]	27	1	0 %
	VGG16 [16]	13	1	50 %
CIFAR-100 [14]	ResNet18 [17]	17	1	50 %
	VGG16 [16]	13	3	50 %

VI. Results

Among four benchmark DNN applications, we present the test result of VGG16 on CIFAR10 for brievty in this paper.

A. Compact Test Image Subset Analysis

Fig. 4 shows the standard deviation (SD) of prediction error and test speedup when varying the size of compact image subset. A performance of regressor is evaluated by the error SD and an overall test efficiency of proposed scheme is measured by the test speedup. The size of the compact image subset is increased from 10 to 1000 under the proposed testing using VGG16 on CIFAR10, which is 0.1% to 10% of entire test dataset. We represent the size of applied compact image subset as *a normalized percentage over the size of entire test dataset.* The accuracy threshold is 85% and the PRI is [75%, 100%].

As the size of compact test subset increases to 3%, the performance of regressor improves (i.e., SD of prediction error decreases) as shown in Fig. 4a. Once the size of compact subset surpasses 3% of the test set, the improvement of regressor saturates and slowly increases. This is due to the MARS regressor used for performance prediction utilizing a finite number of basis functions to fit the signature to predict model accuracy. Larger compact image subsets generate longer signature vector to reach the limit of improvement of this formulation. Test speedup is slowly reduced after its rapid growth to a maximum at 3% image subset in Fig. 4b because

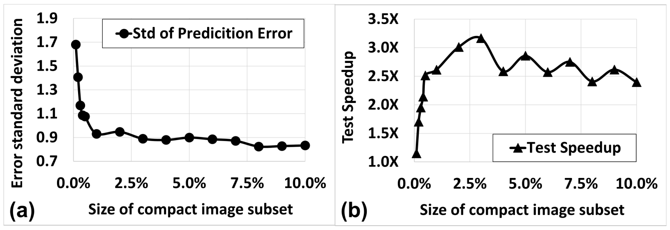

Fig. 4: Sensitivity of compact test image subset.

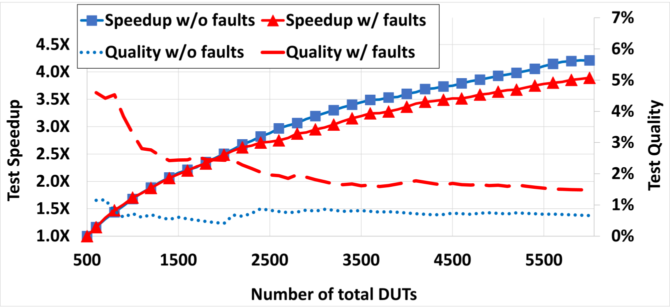

Fig. 5: Improvement under ML kernel retraining.

the increment of test cost dominates the marginal improvement of regressor performance as we increase the test subset size.

B. ML Kernel Retraining Analysis

The testing performance with ML kernel retraining is evaluated using VGG16 on CIFAR-10. In this experiment, a single test module is initially trained on 500 benchmarking devices with a compact image subset size of 3% (i.e., 300 images out of 10K test images). In total, 6000 DUTs are tested with the retraining process under a batch size of 100, which after every 100 DUTs, the ML kernels are retrained for the next test run (see Section IV-D). The accuracy threshold is 84% and the PRI is assigned as [75%, 100%]. *Permanent faults are injected concurrently with process variations to test the scheme in the presence of defects.* Stuck-at-faults are injected in 10% of the initial benchmarking devices and each batch of DUTs. Faults are injected into 10% of the memristors in the RRAM crossbar. Stuck-at-zero (SA0) faults, the RRAM device is always at its low resistance state representing a maximum weight value, and stuck-at-one (SA1) faults, the memristor is stuck at its high resistance state representing a minimum weight are used for a fault injection.

Fig. 5 shows test speedup and test quality for devices with and without injection of permanent faults. For devices without stuck-at-faults, the test speedup (square marker) gradually improves to achieve 4.2× computational efficiency compared to the standard testing. The test quality (dotted) is fine-tuned within 3 iterations and is maintained below 1% throughout the testing process. In the presence of faults, test speedup (triangle marker) shows a comparable trend of improvement to a test case without fault injection, achieving 3.9× test efficiency at the end of the retraining, which is marginally degraded. This is because defective devices, 10% of total DUTs, are detected by the outlier detector and have to be tested by expensive standard testing, which degrades the test speedup. The test quality for faulty devices (dash) shows significant improvement during retraining. At the initial test runs, the test result with the presence of faults has high false positive rate (i.e., initial test quality = 4.6%) because ML kernels of test modules lack statistics of faulty devices. Our retraining process expands the operating space of outlier detector and regressor using the data of fault injected devices within PRI for every iteration. After the 4th iteration, the test modules start to learn the statistics of moderate faulty devices (i.e., outliers within PRI), as shown by a decline in test quality in Fig. 5. The testing modules for faulty devices are simultaneously trained on nonidealities and permanent faults.

VII. Conclusion

This paper presents an alternative testing framework to accelerate evaluation of DNN models realized on resistive crossbar arrays. Our proposed alternative testing significantly reduces the computational cost of DNN testing. The viability of this work is verified in the presence of permanent faults under recursive retraining and over multiple test cases.

Acknowledgment

This research was supported by the U.S. National Science Foundation under Grant: 2128149.

References

[1] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 260–285, 2018.

[2] S. Jain, A. Sengupta, K. Roy, and A. Raghunathan, "Rxnn: A framework for evaluating deep neural networks on resistive crossbars," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 2, pp. 326–338, 2021.

[3] S. Roy, S. Sridharan, S. Jain, and A. Raghunathan, "Txsim: Modeling training of deep neural networks on resistive crossbar systems," *IEEE Transactions on Very Large Scale Integration(VLSI) Systems*, vol. 29, no. 4, pp. 730–738, 2021.

[4] P. N. Variyam, S. Cherubal, and A. Chatterjee, "Prediction of analog performance parameters using fast transient testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 3, pp. 349–361, 2002.

[5] R. Voorakaranam, S. S. Akbay, S. Bhattacharya, S. Cherubal, and A. Chatterjee, "Signature testing of analog and rf circuits: Algorithms and methodology," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 5, pp. 1018–1031, 2007.

[6] S. Kundu, S. Banerjee, A. Raha, S. Natarajan, and K. Basu, "Toward functional safety of systolic array-based deep learning hardware accelerators," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 3, pp. 485–498, 2021.

[7] C.-Y. Chen and K. Chakrabarty, "Efficient identification of critical faults in memristor-based inferencing accelerators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[8] Y. Long, X. She, and S. Mukhopadhyay, "Design of reliable dnn accelerator with un-reliable reram," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1769–1774.

[9] P.-Y. Chen and S. Yu, "Compact modeling of rram devices and its applications in 1t1r and 1s1r array design," *IEEE Transactions on Electron Devices*, vol. 62, no. 12, pp. 4022–4028, 2015.

[10] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[11] P. J. Rousseeuw, "Least median of squares regression," *Journal of the American statistical association*, vol. 79, no. 388, pp. 871–880, 1984.

[12] J. H. Friedman, "Multivariate adaptive regression splines," *The annals of statistics*, vol. 19, no. 1, pp. 1–67, 1991.

[13] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, 1908.

[14] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] Nanoscale Integration and Modeling (NIMO) Group, ASU, "Predictive Technology Model," http://ptm.asu.edu, 2011, online; accessed 8 April 2022.