Probabilistic Error Models for Machine Learning Kernels Implemented on Stochastic Nanoscale Fabrics

Sai Zhang, *Student Member, IEEE,* and Naresh R. Shanbhag, *Fellow, IEEE*
Dept. of Electr. & Comput. Eng., Univ. of Illinois at Urbana-Champaign, Urbana, IL, USA

***Abstract*—Presented in this paper are probabilistic error models for machine learning kernels implemented on low-SNR circuit fabrics where errors arise due to voltage overscaling (VOS), process variations, or defects. Four different variants of the additive error model are proposed that describe the error probability mass function (PMF): additive over Reals Error Model with independent Bernoulli RVs (REM-i), additive over Reals Error Model with joint Bernoulli random variables (RVs) (REM-j), additive over Galois field Error Model with independent Bernoulli RVs (GEM-i), and additive over Galois field Error Model with joint Bernoulli RVs (GEM-j). Analytical expressions for the error PMF is derived. Kernel level model validation is accomplished by comparing the Jensen-Shannon divergence D_{JS} between the modeled PMF and the PMFs obtained via HDL simulations in a commercial $45\,\mathrm{nm}$ CMOS process of MAC units used in a 2^{nd} order polynomial support vector machine (SVM) to classify data from the UCI machine learning repository. Results indicate that at the MAC unit level, D_{JS} for the GEM-j models are 1-to-2-orders-of-magnitude lower (better) than the REM models for VOS and process variation errors. However, when considering errors due to defects, D_{JS} for REM-j is between 1-to-2-orders-of-magnitude lower than the others. Performance prediction of the SVM using these models indicate that when compared with Monte Carlo with HDL generated error statistics, probability of detection p_{det} estimated using GEM-j is within 3% for VOS error when the error rate $p_\eta \leq 80\%$, and within 5% for process variation error when supply voltage V_{dd} is between $0.3\,\mathrm{V}$ and $0.7\,\mathrm{V}$. In addition, p_{det} using REM-j is within 2% for defect errors when the defect rate (the percentage of circuit nets subject to stuck-at-faults) p_{saf} is between 10^{-3} and 0.2.**

I. Introduction

There is great interest in implementing machine learning (ML) kernels on silicon [1], [2] in order to provide *in-situ* data analytics in emerging sensor-rich platforms. Such implementations need to be energy-efficient in order to operate with energy constrained sources. Dedicated implementations of ML kernels have been shown to achieve 144× energy reduction and a 37.5-49× throughput enhancement, when applied to cardiac-arrhythmia detection [1] and a MRF-based stereo matching engine [2] as compared to implementations on general purpose platforms.

ML algorithms have a unique property that can be exploited to enhance energy efficiency and throughput - robustness to statistical errors in data and computation [3]. Indeed, techniques such as approximate computing [4] do exploit this inherent robustness to trade off accuracy for improved energy/performance. ML algorithms' robustness to computational errors can be exploited in another way, which is by implementing such kernels on circuit fabrics that operate at the limits of energy efficiency and hence robustness. We refer to such fabrics as *low-SNR* or *stochastic nanoscale fabrics* as the behavior of computational kernels on such fabrics is non-deterministic. Examples of stochastic nanoscale fabrics include near/subthreshold voltage (NTV) CMOS [5] and the emerging nanoscale devices such as CNFET [6], spin [7], and others. It is possible to exceed the limits of intrinsic robustness of ML kernels by the use of stochastic nanoscale fabrics exhibiting very high error rates. In such cases, extrinsic robustness can be introduced at the algorithmic and architectural levels via error resiliency techniques such as RAZOR [8], error detection sequential (EDS) [9], confidence driven computing (CDC), data driven hardware resilience (DDHR) [10] and statistical error compensation (SEC) [11], to ensure that application level quality is maintained.

Systematic design of ML kernels on stochastic nanoscale fabrics requires one to efficiently predict the behavior of such implementations. For this, high-level error models of key ML building blocks need to be developed. The error models of such kernels need to capture the stochastic behavior of the underlying fabric such as voltage overscaling (VOS), process variations, and defects. Compact analytical models of kernel behavior in the presence of errors are much desirable as these can be employed to: 1) characterize the inherent error resiliency of ML algorithms, and 2) evaluate the effectiveness of error resiliency techniques in compensating for these errors.

The error behavior of computational kernels can be fully captured in terms of their joint probability mass functions (PMFs). To date, not much work has been done on this topic. Analytical models for logic errors [12], transient errors [13], and timing errors [14] have been proposed. These models focus on obtaining expressions for the error rate/magnitude. In approximate computing, theoretical models have been proposed to model the inaccuracy of circuits [15]. However, the models are architecture specific. Interval-based approaches (interval arithmetic or affine arithmetic) [16] have been proposed to model and propagate PMFs. The approaches need to store the entire error PMF. A lookup table based technique [17] has been proposed to characterize the statistical properties of approximate hardware. These models only capture the standard deviations of basic circuit building blocks rather than the error PMF. In signature analysis based testing, symmetrical error model [18] and independent error model [19] are employed to model the output error PMF. Additionally, in all the models for approximate computing, the errors are due to imprecise but deterministic circuits, not dynamic errors due to VOS and process variations.

In this paper, we propose a probabilistic additive error model capable of modeling errors due to sources such as VOS, process variations, and defects. Four different variants of the additive error model are studied: additive over **R**eals **E**rror **M**odel with **i**ndependent Bernoulli RVs (REM-i), additive over **R**eals **E**rror **M**odel with **j**oint Bernoulli RVs (REM-j), additive over **G**alois field **E**rror **M**odel with **i**ndependent Bernoulli RVs (GEM-i), and additive over **G**alois field **E**rror**M**odel with **j**oint Bernoulli RVs (GEM-j). Analytical expressions for the error PMFs are derived. Kernel level model validation is accomplished by comparing the Jensen-Shannon divergence D_{JS} between the modeled PMF and the PMFs obtained via HDL simulations in a commercial $45\,\mathrm{nm}$ CMOS process of MAC units used in a support vector machine (SVM) to classify the UCI machine learning dataset [20].

Results indicate that at the MAC unit level, D_{JS} for the GEM models are 2-orders-of-magnitude lower (better) than the REM models for VOS, and 1-order-of-magnitude lower for process variation errors. However, when considering errors due to defects, D_{JS} for REM-j is between 1-to-2-orders-of-magnitude lower than the others. Performance (probability of detection p_{det}) prediction of a 2^{nd} order polynomial SVM classifier is conducted using the proposed model and compared with Monte Carlo simulation using HDL generated error statistics. We find that p_{det} estimated using GEM-j is within 3% for VOS errors when the error rate $p_\eta \leq 80\%$, and within 5% for process variation errors when supply voltage V_{dd} is between $0.3\,\mathrm{V}$ and $0.7\,\mathrm{V}$. In addition, p_{det} using REM-j is within 2% for defect errors when the defect rate (the percentage of circuit nets subject to stuck-at-faults) p_{saf} is between 10^{-3} and 0.2.

The rest of the paper is organized as follows. Section II describes the framework for the error analysis and the distance measure employed to compare models. Section III presents the proposed models, and derives analytical expressions for the PMF of the errors. Section IV presents the error characterizatio/simulation methodology, and model validation results at kernel level and system level. Conclusions are presented in Section V.

II. Modeling Framework and Accuracy Measure

A. Error Modeling Framework

In this paper, we employ bold symbols to denote random variables (RVs). The proposed error modeling framework (see Fig. 1) captures the spatio-temporal distribution of errors. This is required as certain error sources such as defect and process variations result in an error RV whose PMF is determined by the statistics of the input $\mathbf{x}$ and the spatial distribution across physical instantiations of the computational block. The following notation is employed in this paper: let I_k $(k = 1, 2..., M)$ denote the k^{th} instance of the system/kernel subject to errors, and let $\mathbf{x}_k[n]$, $\mathbf{y}_{o,k}[n]$, $\boldsymbol{\eta}_k[n]$, and $\mathbf{y}_{a,k}[n]$ denote the random process corresponding to the input, error free output, error, and the final output of I_k, with time index n, respectively.

B. Additive Error Models

For notational simplicity, we drop the index n and k. We consider the additive error model as shown in Fig. 1:

$$\mathbf{y}_a = \mathbf{y}_o \oplus_F \boldsymbol{\eta} \tag{1}$$

where $\oplus_F$ denotes addition over field F, $\boldsymbol{\eta}$ is the error RV with PMF $P_{\boldsymbol{\eta}}(\eta)$. The models (REM-i,j and GEM-i,j) proposed in this paper focus on modeling $P_{\boldsymbol{\eta}}(\eta)$.

In REM-i,j, addition in (1) is taken over the field of reals $\mathbb{R}$. Thus, (1) can be written as

$$\mathbf{y}_a = \mathbf{y}_o + \boldsymbol{\eta} \tag{2}$$

where $\mathbf{y}_a, \mathbf{y}_o$ and $\boldsymbol{\eta}$ are reals expressed in the *2's complement form.* For example, $\boldsymbol{\eta}$ is written in the 2's complement form as:

$$\boldsymbol{\eta} = -\boldsymbol{\eta}_0^b + \sum_{i=1}^{B_\eta - 1} \boldsymbol{\eta}_i^b 2^{-i} \tag{3}$$

and $\boldsymbol{\eta}_i^b \in \{0, 1\}$ and B_η are the i^{th} bit and bit precision of $\boldsymbol{\eta}$, respectively. The 2's complement form of $\mathbf{y}_a$ and $\mathbf{y}_o$ can be expressed similarly.

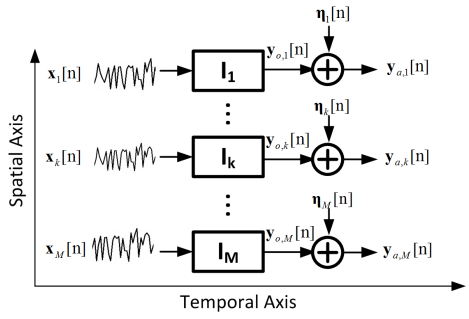
In GEM-i,j, addition in (1) is taken over the Galois field of 2 (GF(2)). Thus, (1) can be written as:

$$\overrightarrow{\mathbf{y}}_a = \overrightarrow{\mathbf{y}}_o \oplus \overrightarrow{\boldsymbol{\eta}} \tag{4}$$

where $\overrightarrow{\mathbf{y}}_a$, $\overrightarrow{\mathbf{y}}_o$ and $\overrightarrow{\boldsymbol{\eta}}$ are the bit vectors representing $\mathbf{y}_a, \mathbf{y}_o$ and $\boldsymbol{\eta}$, respectively, and $\oplus$ is the bitwise XOR operator. For example, $\overrightarrow{\boldsymbol{\eta}}$ is given by the *vectorized form* as:

$$\overrightarrow{\boldsymbol{\eta}} = [\boldsymbol{\eta}_0^b, ...\boldsymbol{\eta}_{B_\eta - 1}^b]^T \tag{5}$$

The vectorized form of $\mathbf{y}_a$ and $\mathbf{y}_0$ can be expressed similarly. Note that the 2's complement form and the vectorized form of $\boldsymbol{\eta}$ are equivalent.



(a)

Fig. 1. Error modeling framework.

C. Model Accuracy Metric

To quantify model accuracy, we employ the commonly employed Jensen-Shannon (JS) divergence D_{JS} [21] as the measure of the distance between two distributions. The JS divergence between two PMFs P and Q is defined as:

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M) \quad (6)$$

where $M(\cdot) = \frac{1}{2}P(\cdot) + \frac{1}{2}Q(\cdot)$, and D_{KL} is the Kullback–Leibler (KL) divergence defined as:

$$D_{KL}(P||Q) = \sum_i P(i) log_2 \left(\frac{P(i)}{Q(i)}\right) \quad (7)$$

The reason for choosing the JS divergence as the distance measure is that it is symmetric ($D_{JS}(P||Q) = D_{JS}(Q||P)$) and bounded ($0 \leq D_{JS} \leq 1$) [21], unlike KL divergence.

III. Error Model Derivation

The challenges in modeling $\boldsymbol{\eta}$ lie in the fact that: 1) $\boldsymbol{\eta}$ is a discrete RV and is restricted to certain error magnitudes (especially when error rate is low), and 2) the PMF is not smooth. Instead of modeling the error magnitude directly, we propose to model the bits $\boldsymbol{\eta}_i^b$ of $\boldsymbol{\eta}$ as joint RVs. Four different variants of the additive error model are studied: REM-i, REM-j, GEM-i, and GEM-j.

A. REM-i: additive over Real Error Model with independent Bernoulli RVs

In REM-i (see (2)), $\boldsymbol{\eta}_i^b$ ($i = 0, 1..., B_\eta - 1$) is modeled as a Bernoulli RV so the PMF of $\boldsymbol{\eta}_i^b$ can be written as:

$$P_{\boldsymbol{\eta}_i^b}(x) = \begin{cases} p_i & \text{if } x = 1 \\ 1 - p_i & \text{if } x = 0 \end{cases} \quad (8)$$

and $\boldsymbol{\eta}_i^b$ ($i = 0, 1..., B_\eta - 1$) are assumed to be independent. Thus, under REM-i, the PMF $P_{\boldsymbol{\eta}}(\eta)$ can be obtained from (8) as:

$$P_{\boldsymbol{\eta}}(\eta) = \prod_{i=0}^{B_\eta - 1} p_i^{\eta_i^b}(1 - p_i)^{1-\eta_i^b} \quad (9)$$

Statistical metrics such as the mean and variance of $\boldsymbol{\eta}$ can be easily derived from (3) and (9).

B. REM-j: additive over Real Error Model with joint Bernoulli RVs

The pairwise covariance between $\boldsymbol{\eta}_i^b$ and $\boldsymbol{\eta}_j^b$ can be included to improve the modeling accuracy. In REM-j, the PMF of $\boldsymbol{\eta}$ is parametrized by the mean vector $\overrightarrow{\boldsymbol{\mu}}_\eta = [p_0, p_1, ..., p_{B_\eta - 1}]^T$ and the covariance matrix $\mathbf{C}_\eta$, where $\mathbf{C}_\eta(i, j) = cov(\boldsymbol{\eta}_i^b, \boldsymbol{\eta}_j^b)$ is the covariance between $\boldsymbol{\eta}_i^b$ and $\boldsymbol{\eta}_j^b$ for $i, j = 0, 1, ..., B_\eta - 1$.

The dichotomized Gaussian (DG) distribution [22] can be used to obtain the PMF of $\overrightarrow{\boldsymbol{\eta}}$. It is shown that [22] for any $\overrightarrow{\boldsymbol{\eta}}$, there exists a latent multivariate Gaussian $\overrightarrow{\mathbf{u}} = [\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{B_\eta - 1}]^T$ with mean vector $\overrightarrow{\boldsymbol{\mu}}_u$ and covariance matrix $\mathbf{C}_u$ such that after dichotomizing $\overrightarrow{\mathbf{u}}$, i.e.

$$\hat{\boldsymbol{\eta}}_i^b = \begin{cases} 1 & \mathbf{u}_i \geq 0 \\ 0 & \mathbf{u}_i < 0 \end{cases} \text{ for } (i = 0, 1..., B_\eta - 1) \quad (10)$$

The mean and covariance of $\overrightarrow{\hat{\boldsymbol{\eta}}} = [\hat{\boldsymbol{\eta}}_0^b, \hat{\boldsymbol{\eta}}_1^b .., \hat{\boldsymbol{\eta}}_{B_\eta - 1}^b]^T$ are $\overrightarrow{\boldsymbol{\mu}}_\eta$ and $\mathbf{C}_\eta$, respectively, i.e., $\overrightarrow{\hat{\boldsymbol{\eta}}}$ and $\overrightarrow{\boldsymbol{\eta}}$ have identical first and second order statistics. Therefore, as shown in the Appendix, REM-j can be obtained as:

$$\begin{aligned} P_{\boldsymbol{\eta}}(\eta) &= P_{\overrightarrow{\boldsymbol{\eta}}}([\boldsymbol{\eta}_0^b, ...\boldsymbol{\eta}_{B_\eta - 1}^b]^T = [\eta_0^b, ..., \eta_i^b]^T) \\ &= \Phi_{\widehat{\overrightarrow{\mathbf{u}}}}([0, ...0]^T; \mathbf{D}\overrightarrow{\boldsymbol{\mu}}_u, \mathbf{D}\mathbf{C}_u\mathbf{D}^T) \end{aligned} \quad (11)$$

where

$$\mathbf{D} = \begin{pmatrix} (-1)^{\eta_0^b} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & (-1)^{\eta_{B_\eta - 1}^b} \end{pmatrix} \quad (12)$$

and $\Phi_{\widehat{\overrightarrow{\mathbf{u}}}}([0, ...0]^T; \mathbf{D}\overrightarrow{\boldsymbol{\mu}}_u, \mathbf{D}\mathbf{C}_u\mathbf{D}^T)$ is the CDF of the joint Gaussian $\widehat{\overrightarrow{\mathbf{u}}}$ with mean $\mathbf{D}\overrightarrow{\boldsymbol{\mu}}_u$ and covariance matrix $\mathbf{D}\mathbf{C}_u\mathbf{D}^T$ evaluated at $[0, 0...0]^T$.

The mean and variance of $\boldsymbol{\eta}$ can be calculated using (3) and (11).

C. GEM-i and GEM-j

The main difference between GEM-i,j (see (4)) and REM-i,j (see (2)) is that in GEM-i,j, the error $\overrightarrow{\boldsymbol{\eta}}$ is defined using addition over GF(2) instead of real addition. Since the vectorized form $\overrightarrow{\boldsymbol{\eta}}$ and the 2's complement form $\boldsymbol{\eta}$ are equivalent, GEM-i,j can be derived in the same manner as REM-i,j. Therefore, the PMF of $\boldsymbol{\eta}$ under GEM-i and GEM-j has the same form as in (9) and (11), respectively. Note that the independent error model [19] is a special case ($p_i = p, \forall i$) of the GEM-i model.

IV. Model Validation

The proposed models are validated and compared at both the kernel and system levels. Kernel level validation aims at comparing the JS divergence D_{JS} between the proposed models and the PMFs obtained via HDL simulation. System level validation aims at validating the accuracy of the proposed models in predicting system level performance metric S. As shown in Fig. 1, S can be obtained via averaging over the spatio-temporal domain. However, we employ the following procedure in order to evaluate the performance yield: 1) For each instance I_k, the system level performance metric for I_k is obtained by averaging over the input $\mathbf{x}$, i.e., $\mathbf{S}_k = E_{\mathbf{x}}(S|I_k)$, where $\mathbf{S}_k$ is a RV, and 2) statistical measures such as the mean and standard deviation of $\mathbf{S}_k$ can be obtained by performing spatial averaging.

A. Error characterization and injection methodology

Figure 2(a) shows the error characterization and injection methodology for VOS, process variation, and defect errors in a common framework. In this paper, an SVM classifier as shown in Fig. 2(b) is employed to validate the models. The SVM classifier consists of two types of multiply accumulator (MAC) kernels: MAC1 is an $8\,\mathrm{b}$ input, $8\,\mathrm{b}$ coefficient, and $22\,\mathrm{b}$ output MAC used in the first stage, and MAC2 is a $10\,\mathrm{b}$ input, $8\,\mathrm{b}$ coefficient, and $24\,\mathrm{b}$ output MAC used in the second stage. Simulation results are obtained using a commercial $45\,\mathrm{nm}$ CMOS process.

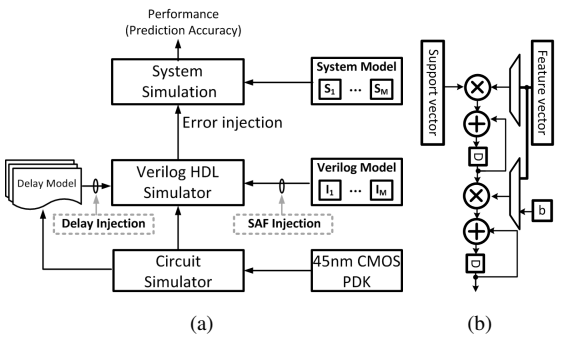


Fig. 2. Model validation: a) error characterization and injection methodology, and b) 2^{nd} order polynomial kernel SVM classifier.

VOS error characterization and injection is done as follows:

1) Characterize delay vs. V_{dd} of basic gates such as AND and XOR using HSPICE for $0.3\,\mathrm{V} \leq V_{dd} \leq 1.2\,\mathrm{V}$.
2) Develop structural Verilog HDL models for the SVM classifier.
3) HDL simulations of the MAC kernels were conducted at different voltages by including the appropriate delay obtained in Step 1 into the HDL model. The errors were characterized as error PMFs.
4) During kernel level validation, analytical models were built using REM-i,j and GEM-i,j (see (9) and (11)). The JS divergence between the models and the characterized error PMFs were calculated according to (6).
5) During system level validation, the system performance (i.e., probability of detection) was evaluated by performing error injection using the HDL error PMF and the models in (9)-(11).

Process variation error characterization and injection is done as follows:

1) Characterize the delay distributions at different V_{dd} of basic gates such as AND and XOR using Monte Carlo Simulation in HSPICE for $0.3\,\mathrm{V} \leq V_{dd} \leq 1.2\,\mathrm{V}$.
2) Develop structural Verilog HDL models of key kernels needed in the SVM classifier.
3) During HDL simulation, multiple instances were generated, and random delays were injected into gates by sampling the characterized delay distribution. HDL simulations of the kernels were conducted to characterize error PMFs for each of the instances.
4) Kernel and system level model validations were conducted following the same procedure as in the VOS methodology.

Defect error characterization and injection is done as follows:

1) Develop structural Verilog HDL models of key kernels needed in the SVM classifier.
2) During HDL simulation, multiple instances of the kernels were generated, and defects (stuck-at-one and stuck-at-zero errors) with different defect error rate p_{saf} were injected to randomly selected nets in the Verilog netlist using custom scripts.
3) Kernel and system level model validations were conducted following the same procedure as in the VOS methodology.

B. Kernel level model validation

Figure 3 shows the JS divergence comparison at different voltage overscaling factor $K_{vos} = V_{dd}/V_{dd,crit}$ where $V_{dd-crit}$ is the minimum voltage needed for error free operation. It shows that for VOS errors, GEM-j achieves the lowest D_{JS} which is below 10^{-2} for $0.5 \leq K_{vos} \leq 0.95$, and both GEM-i and GEM-j achieve up to two-orders-of-magnitude smaller D_{JS} compared with REM-i and REM-j. Additionally, Figure 3 shows that the D_{JS} of the symmetrical error model [18] and independent error model [19] is higher than the GEM models. For all modeling methods, the PMF modeling accuracy decreases as K_{vos} decreases (thus error rate p_η increases).

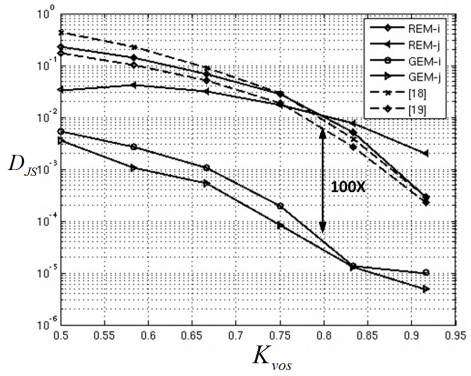


Fig. 3. JS divergence comparison of the proposed models for VOS errors for MAC1 used in the SVM classifier. The JS divergence is calculated between the proposed models and the error PMFs obtained via HDL simulation. The results for MAC2 are similar.

Figure 4 shows that in the case of process variation errors, GEM-j achieves the lowest D_{JS} which is below 0.03 for $0.3\,\mathrm{V} \leq V_{dd} \leq 0.7\,\mathrm{V}$, and both GEM-i and GEM-j achieve up to $10\times$ smaller D_{JS} compared with REM-i and REM-j. Additionally, Figure 4 shows the D_{JS} of the symmetrical error model [18] and independent error model [19] is higher the GEM models. This is expected due to the fact that process variation errors are indeed timing errors. For all modeling methods, the PMF modeling accuracy decreases with V_{dd}.

Figure 5 shows that unlike timing errors caused by VOS or process variations, in the case of defects, REM-j achieves the lowest D_{JS} which is below 0.05 for $10^{-3} \leq p_{saf} \leq 0.2$ compared with other models. This indicates that the error statistics of defect errors is different from timing errors, and different model should be employed. In addition, Figure 5 shows the D_{JS} of the symmetrical error model [18] and independent error model [19] is higher than any of the proposed models. Figure 5 also shows that PMF modeling accuracy decreases at higher p_{saf} for all modeling methods.

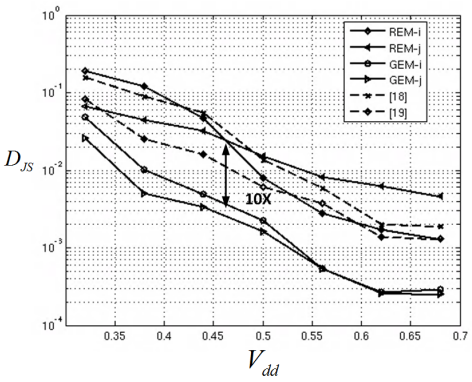


Fig. 4. JS divergence comparison of the proposed models for process variation errors for MAC1 used in the SVM classifier. The JS divergence is calculated between the proposed models and error PMFs obtained via HDL simulation for $M = 30$ instances. The mean D_{JS} is shown in the figure. The results for MAC2 are similar.

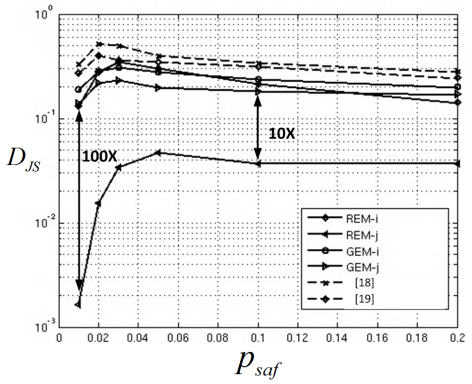


Fig. 5. JS divergence comparison of the proposed models for defect errors for MAC1 used in the SVM classifier. The JS divergence is calculated between the proposed models and error PMFs obtained via HDL simulation for $M = 30$ instances. The mean D_{JS} is shown in the figure. The results for MAC2 are similar.

C. System level simulation

To evaluate the model, we employ the probabilistic models in system simulation and compare it with the conventional error injection methodology following the procedure in IV-A. We employ the Breast Cancer Wisconsin dataset from UCI machine learning repository [20] which consists of labeled feature vectors (benign vs. malignant) constructed from digitized images of fine needle aspirates (FNA) of patient tissue, and use the SVM classifier to perform classification.

In this application, the performance metric $\mathbf{S}_k = \mathbf{p}_{det}$. Figure 6 plots the $\mathbf{p}_{det}$ in presence of VOS errors, and shows that GEM-i and GEM-j are more accurate than REM-i and REM-j. The difference between the estimated $\mathbf{p}_{det}$ using GEM-j and HDL error statistics is within 3% for error rate $p_\eta \leq 80\%$.

Figure 7 shows the distribution of $\mathbf{p}_{det}$ in presence of process variation errors, and demonstrates that GEM-i and GEM-j are more accurate than REM-i and REM-j, similar to the case of VOS errors. The difference between the estimated

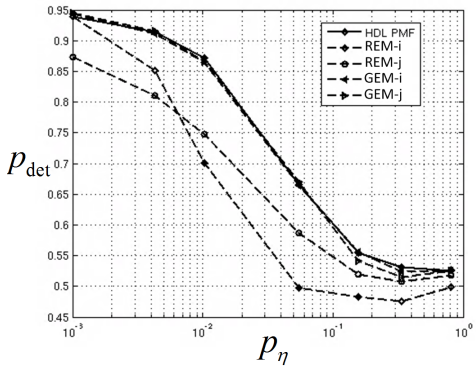


Fig. 6. System simulation results in presence of VOS errors the SVM classifier comparing the proposed models with HDL error statistics.

$\mathbf{p}_{det}$ using GEM-j and HDL error statistics is within 5% for $0.3\,\text{V} \leq V_{dd} \leq 0.7\,\text{V}$.

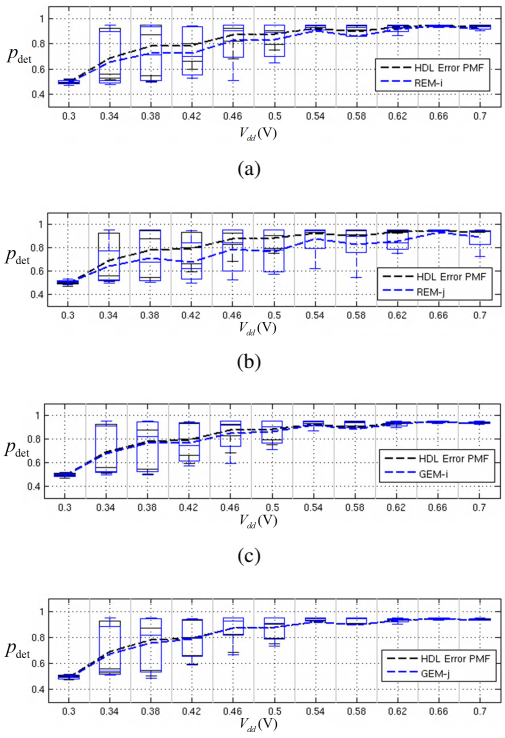
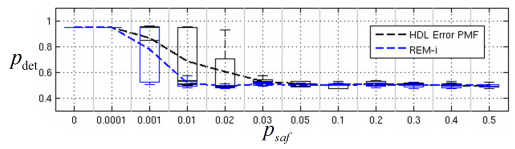
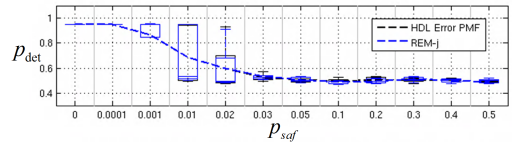


Fig. 7. System simulation results in presence of process variation errors for the SVM classifier comparing the HDL error statistics with a) REM-i, b) REM-j, c) GEM-i, and d) GEM-j. Simulations are performed for 30 instances, the box plot shows the median, 25%, and 75% quartile of the prediction accuracy $\mathbf{p}_{det}$, the dashed line shows the mean $\mathbf{p}_{det}$.

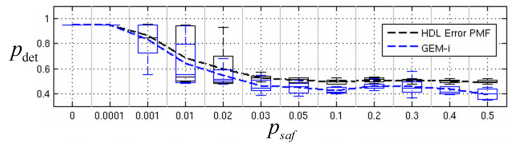
Figure 8 shows the distribution of $\mathbf{p}_{det}$ in presence of defect errors, and demonstrates that REM-j achieves higher accuracy compared with other models, unlike the case of VOS and process variation errors. The difference between the estimated $\mathbf{p}_{det}$ using REM-j and HDL error statistics is within 2% for $10^{-3} \leq p_{saf} \leq 0.2$.



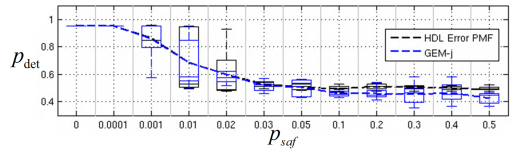
(a)



(b)



(c)



(d)

Fig. 8. System simulation results in presence of defect errors for the SVM classifier comparing the HDL error statistics with a) REM-i, b) REM-j, c) GEM-i, and d) GEM-j. Simulations are performed for 30 instances, the box plot shows the median, 25%, and 75% quartile of the prediction accuracy $\mathbf{p}_{det}$, the dashed line shows the mean $\mathbf{p}_{det}$.

V. Conclusion

In this paper, probabilistic additive models for circuit errors due to VOS, process variation, and defects were proposed to effectively predict the performance of ML kernels in presence of hardware errors. Four models were compared, and analytical expressions for the PMF were derived. In addition, error characterization/injection methodology were proposed and employed to validate the models. Kernel level validation showed that the GEM-j is the most accurate for VOS and process variation errors, but REM-j is the most accurate for defect errors. System level simulation using a 2^{nd} order polynomial SVM classifier further confirms the validity of the models.

Acknowledgment

This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.

Appendix

In this appendix, we derive (11). Use vectorized notation $\overrightarrow{\boldsymbol{\eta}}$ in (5), the PMF of $\boldsymbol{\eta}$ can be expressed as:

$$\begin{aligned} & P_{\boldsymbol{\eta}}(\eta) \\ & = P_{\overrightarrow{\boldsymbol{\eta}}}([\boldsymbol{\eta}_0^b, ...\boldsymbol{\eta}_{B_\eta-1}^b]^T = [\eta_0^b, ..., \eta_i^b]^T) \\ & = P_{\overrightarrow{\mathbf{u}}}((-1)^{\eta_0}\mathbf{u}_0 < 0, .., (-1)^{\eta_{B_\eta-1}}\mathbf{u}_{B_\eta-1} < 0) \end{aligned} \tag{13}$$

where $\overrightarrow{\mathbf{u}} = [\mathbf{u}_0, \mathbf{u}_1, ..., \mathbf{u}_{B_\eta-1}]^T$ is the latent Gaussian that can be dichotomized to obtain $\overrightarrow{\boldsymbol{\eta}}$ according to (10). We further define $\hat{\overrightarrow{\mathbf{u}}} = \mathbf{D}\overrightarrow{\mathbf{u}} = \mathbf{D}[\mathbf{u}_0, ..., \mathbf{u}_{B_\eta-1}]^T$ where $\mathbf{D}$ is defined in (12). Hence, the mean and variance of $\hat{\overrightarrow{\mathbf{u}}}$ can be calculated as $E(\hat{\overrightarrow{\mathbf{u}}}) = \mathbf{D}\overrightarrow{\boldsymbol{\mu}}_u$ and $Cov(\hat{\overrightarrow{\mathbf{u}}}) = \mathbf{D}\mathbf{C}_u\mathbf{D}^T$, where $\overrightarrow{\boldsymbol{\mu}}_u$ and $\mathbf{C}_u$ are the mean vector and covariance matrix of the latent Gaussian $\overrightarrow{\mathbf{u}}$. The PMF of $\boldsymbol{\eta}$ can be obtained as:

$$P_{\boldsymbol{\eta}}(\eta) = \Phi_{\hat{\overrightarrow{\mathbf{u}}}}([0, ...0]^T; \mathbf{D}\overrightarrow{\boldsymbol{\mu}}_u, \mathbf{D}\mathbf{C}_u\mathbf{D}^T)$$

where $\Phi_{\hat{\overrightarrow{\mathbf{u}}}}([0, ...0]^T; \mathbf{D}\overrightarrow{\boldsymbol{\mu}}_u, \mathbf{D}\mathbf{C}_u\mathbf{D}^T)$ is the CDF of the joint Gaussian $\hat{\overrightarrow{\mathbf{u}}}$ evaluated at $[0, 0..., 0]^T$.

References

[1] K. Lee, S. Kung, and N. Verma, "Low-energy formulations of support vector machine kernel functions for biomedical sensor applications," *Journal of Signal Processing Systems*, vol. 69, no. 3, pp. 339–349, 2012.
[2] J. Choi and R. Rutenbar, "Hardware implementation of MRF MAP Inference on an FPGA platform," in *FPL 2012*, 2012, pp. 209–216.
[3] J. Choi, E. Kim, and et al., "Error resilient MRF message passing architecture for stereo matching," in *SiPS 2013*, Oct 2013, pp. 348–353.
[4] J. Han and M. Orshansky, "Approximate computing: An emerging paradigm for energy-efficient design," in *ETS 2013*, pp. 1–6.
[5] R. Dreslinski, M. Wieckowski, and et al., "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, Feb 2010.
[6] H. Wei, M. Shulaker, and et al., "Carbon nanotube circuits: Opportunities and challenges," in *DATE, 2013*, March 2013, pp. 619–624.
[7] K. Roy, M. Sharad, and et al., "Beyond charge-based computation: Boolean and non-boolean computing with spin torque devices," in *ISLPED 2013*, Sept 2013, pp. 139–142.
[8] D. Ernst, N. S. Kim, and et al., "RAZOR: a low-power pipeline based on circuit-level timing speculation," in *MICRO-36*, Dec 2003, pp. 7–18.
[9] J. Tschanz, K. Bowman, and et al., "Resilient circuits; enabling energy-efficient performance and reliability," in *ICCAD 2009*, pp. 71–73.
[10] N. Verma, K. Lee, and et al., "Enabling system-level platform resilience through embedded data-driven inference capabilities in electronic devices," in *ICASSP 2012*, 2012, pp. 5285–5288.
[11] N. Shanbhag, R. Abdallah, and et al., "Stochastic computation," in *DAC 2010*, June 2010, pp. 859–864.
[12] M. Abbas, M. Ikeda, and K. Asada, "Statistical model for logic errors in cmos digital circuits for reliability-driven design flow," in *Design and Diagnostics of Electronic Circuits and systems, 2006 IEEE*, 2006.
[13] K. Lingasubramanian and S. Bhanja, "An error model to study the behavior of transient errors in sequential circuits," in *VLSI Design, 2009 22nd International Conference on*, Jan 2009, pp. 485–490.
[14] Y. Liu, T. Zhang, and K. Parhi, "Computation error analysis in digital signal processing systems with overscaled supply voltage," *VLSI, IEEE Transactions on*, vol. 18, no. 4, pp. 517–526, April 2010.
[15] L. Li and H. Zhou, "On error modeling and analysis of approximate adders," in *ICCAD 2014*, 2014, pp. 511–518.
[16] J. Huang and J. Lach, "Exploring the fidelity-efficiency design space using imprecise arithmetic," in *ASP-DAC, 2011*, pp. 579–584.
[17] W.-T. Chan, A. Kahng, and et al., "Statistical analysis and modeling for error composition in approximate computation circuits," in *ICCD 2013*, Oct 2013, pp. 47–53.
[18] K. Iwasaki and F. Arakawa, "An analysis of the aliasing probability of multiple-input signature registers in the case of a 2m-ary symmetric channel," *CAD, IEEE Transactions on*, vol. 9, no. 4, pp. 427–438, 1990.
[19] T. Williams, W. Daehn, and et al., "Aliasing errors in signature in analysis registers," *Design Test of Computers, IEEE*, vol. 4, no. 2, 1987.
[20] D. N. A. Asuncion, "UCI machine learning repository," 2007.
[21] J. Lin, "Divergence measures based on the shannon entropy," *Information Theory, IEEE Transactions on*, vol. 37, no. 1, pp. 145–151, 1991.
[22] J. Macke, P. Berens, and et al., "Generating spike trains with specified correlation coefficients," *Neural Computation*, vol. 21, no. 2, 2009.