Signal Processing and Renewable Energy
December 2021, (pp. 67-81)
ISSN: 2588-7327
eISSN: 2588-7335

A Machine Learning-based Model for predicting Stochastic BTI Effects

Siavash Esshaghi *[1], Mohammad Bazli [1], Arash Esshaghi [2]

[1] Electrical Engineering Department, Islamic Azad University, Birjand Branch, Birjand, Iran.
[2] Department of Mathematical Sciences, Payam Noor University, Tehran, Iran.

Received: 07-Jun-2021, Revised: 09-Jul-2021, Accepted: 10-Jul-2021.

Abstract
BTI is a major reliability concern in nanoscale digital design, and addressing it during design space exploration in high levels of abstraction is essential to enhance reliability. Aging prediction model appropriate for these levels should have short runtime. In addition, the model must predict the new-observed stochastic effects of aging. A machine learning (ML)-based model for predicting stochastic aging effects is proposed in this paper. First, a large enough training set is obtained by Monte Carlo (MC) simulations, and then, the ML-based model is trained and developed to predict aging statistical characteristics. Various ML algorithms, such as Ridge, Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest, and stacked generalization are evaluated. Results show that ensemble algorithms have high efficiency in aging prediction. When compared to the MC-based approach, the proposed technique shows that the aging prediction runtime is reduced by more than 99%, while accurate prediction of the statistical properties of stochastic aging is obtained with an accuracy of up to 98%. This improvement is achieved by offline data collecting and model training which needs a noticeable runtime. However, it is a one-time offline task and has no impact on prediction runtime.

Keywords: BTI, Stochastic Aging, Machine Learning, Ensemble Algorithms, Monte Carlo.

1. INTRODUCTION

Reliability has become more challenging as technology has advanced into deep nanometer nodes, and therefore, reliability analysis and lifetime optimization of nanometer digital systems have been manifested as an important necessity in the design flow of modern digital circuits. A complementary metal-oxide-semiconductor (CMOS) circuit may fail due to reliability effects, which are either spatial or temporal.

*Corresponding Authors Email: esshaghi@gmail.com

Fabrication-induced reliability issues resulting from the growing complexity required for fabricating nanoscale CMOS circuits, combined with aggressive scaling into atomic sizes, are known as spatial effects. Temporal effects, which are observed during circuit operation, may be permanent or transient. For example, soft errors induced by cosmic rays cause transient errors. Due to their importance, techniques for hardening the circuits against soft errors are under special investigation today. Permanent effects on the reliability of circuits, such as accelerated transistor aging, often arise from the aging effects on circuit components [1].

After the production of circuits and after the beginning of their operation, aging effects are observed on the behavior of circuit components as run-time deviations from their ideal behavior. Interconnects wear out due to electron migration effect. In addition, transistors suffer from aging phenomena. Transistor aging effects impose serious constraints and challenges in the design of emerging Nano-sized digital systems.

Transistors wear out due to aging phenomena, which includes time dependent dielectric breakdown (TDDB), bias temperature instability (BTI), and hot carrier injection (HCI), among which BTI is a serious mechanism of transistor aging. BTI increases the switching delay of the transistors and hence the delay of combinational paths. Finally, circuit fails due to timing violation [2].

Aging effects and phenomena were first observed in the 1970s. Since then, many studies have been conducted to identify these phenomena and their mechanisms; and research is still ongoing. Efficient aging related models have been developed based on these studies. Initially, the models predicted the aging-induced degraded delay of circuits as a deterministic function of operating conditions such as workload and temperature. However, as the dimensions of the components continued to shrink, stochastic behaviors appeared in the aging effects in circuits. This means that two exactly identical transistors may show different aging rates even under exactly the same operating conditions.

Various methods have been proposed to deal with the destructive effects of aging on circuit performance [3]. Initially, most aging optimization methods focused on low levels of abstraction, which are close to the physical source of transistor aging. An accurate understanding of aging mechanisms is available, and therefore, effective models for predicting and evaluating the aging severity may be developed in these low levels [4]. Over time, high levels of abstraction came to the attention of designers to deal with the complexities of designing modern systems, which led to a shift in the design paradigm of complex digital systems. Design criteria such as reliability, as a major challenge in era of nanometer digital circuits, should be considered in the earliest stages of design. However, transistor-level models for evaluating and predicting the aging of circuits are not suitable for use at high levels of abstraction due to their large execution time and complexity. In addition, aging effects are stochastic in new technology nodes, and therefore, aging models must be upgraded to be able to predict these stochastic behaviors [5]-[7].

In this paper, a model for predicting the aging severity of circuits is proposed which is suitable for high levels of abstraction, e.g. high-level synthesis (HLS). This model predicts the aging severity of circuits according to workload and operating conditions, such as temperature, based on machine learning (ML) approaches. The model predicts the distribution of the circuit aging by means of a mean value and standard deviation. The key point of the model is its short runtime while gives precise results. Therefore, it is suitable for high levels of abstraction, in particular the design space exploration in HLS. To the best of our knowledge, it is the first time that such a model is developed.

The rest of the paper is organized as follows. Related works and background topics are reviewed in Section 2. The proposed approach and experimental results are introduced in Section 3 and Section 4, respectively. The paper concludes with Section 5.

2. BACKGROUND AND RELATED WORKS

Analyzing and predicting the severity of aging is an important part of designing circuits in an age-aware manner. In this section, we classify and review related works.

2.1. Transistor-Level Aging Prediction

BTI gradually degrades the threshold voltage of the transistors and thus their ON current. As a result, the delays of the combinational paths increase over time. Whenever the increased delays cause the timing constraints to be violated, the circuit fails. Therefore, aging effects restrict the operational lifetime of circuits. BTI consists of two similar phenomena, NBTI (negative BTI) and PBTI (positive BTI), which affect PMOS and NMOS transistors, respectively.

NBTI is observed when a negative gate-source voltage (so called Negative BTI) is applied to a PMOS transistor. PBTI is activated by applying a positive voltage to the gate-source of a NMOS transistor. In other words, when a MOSFET transistor turns on, its threshold voltage is degraded due to BTI effects. This is called the stress phase. When the transistor is turned off (by applying the appropriate gate voltage), the BTI effect enters the recovery phase, where a peculiar property of this phenomenon is observed. In the recovery phase, the threshold voltage degradation is partially compensated. However, the recovery rate does not keep pace with the degradation rate, and therefore, after consecutive stress and recovery phases (which is called dynamic BTI), the threshold voltage of the transistor increases, Fig. 1. This long-term degradation of the threshold voltage depends on the duty cycle of the input signal, named also as Signal probability (SP), and is computed deterministically by Equation 1 [4].

$$\Delta V_{th} = [\sqrt{K_v^2 \alpha T_{clk}} \,/\, (1-\beta_t^{-2n})]^{2n} \qquad (1)$$

In the above equation, K_v is a technology-dependent constant. α is workload specification that denotes the input duty cycle or SP. T_{clk} is the clock period. Also, β_t is calculated based on the technology and workload specifications and n is a constant equal to 1/4 or 1/6 depending on technology.

Static BTI is the case that a transistor is under constant stress (duty cycle = 1). It is clear that the static BTI effect has a higher severity when compared to the dynamic BTI since constant stressing is applied and there is no time to recover the degradation in the static NBTI, Fig. 1. Threshold voltage degradation due to the static BTI is computed deterministically as follows [4]:

$$\Delta V_{th} = A[(1+\delta)t_{ox} + \sqrt{C\,t}\,]^{2n} \quad (2)$$

where A and C are technology-dependent constants, t_{ox} is oxide thickness, t is the total stress time, and n is a technology-dependent factor equals to 1/4 or 1/6. δ is a constant and equals to 0.5.

The above equations are true for older technology nodes where the effects of BTI on those relatively large transistors are somewhat deterministic. In the era of deep nanometer technologies, the amount of threshold voltage shift is stochastic and should be described by statistical distributions.

The increase in the threshold voltage (during the stress phase) is due to the charging of individual defect traps which already exist in the gate oxide and the compensation of this degradation (in the recovery phase) is due to the discharging of these traps. On the one hand, the number of traps in individual transistors in advanced technology nodes is relatively small (proportional to the decreasing gate area), and on the other hand, the charging and discharging of these traps is a stochastic process. Consequently, a drastic time-dependent variation in the BTI effects in these deeply scaled devices (and hence in the circuit lifetime) is observed [8].

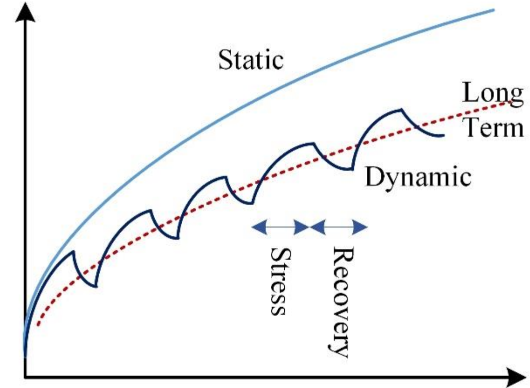

Fig. 1. Static and Dynamic NBTI.

To describe the stochastic behavior of BTI effects, in the atomic trap-based model [6], each transistor is characterized by parameters n (number of defects), τ_c (defects capture time), and τ_e (defects emission time). The time it takes to charge (in the stress phase) and discharge (in the recovery phase) a defect is described as τ_c and τ_e, respectively. These time constants are dependent on voltage and temperature. The probability density function (PDF) of widely distributed defect capture and emission times, as well as their correlations, are defined by capture/emission time (CET) maps [7, 8], as shown in Fig. 2a. Experimental data can be used to build CET maps [9]. By integrating the CET map over the entire time domain, defect density (N_{TD}) (per unit area) of available traps is determined. Consequently, the average number of defects n (in a transistor with the size of W and L) is calculated as follows [6]:

$$n = N_{TD} \cdot WL \quad (3)$$

Depending on workload, only a fraction of available traps will be involved (occupied) and participate in the degradation of the threshold voltage. By alternative stress and

recovery phases by applying high and low digital voltages, the probability of occupancy of traps, denoted by P_{OCC}, (Fig. 2b) can be computed as a function of the frequency (f), duty factor (α), and total time (t), Equation (4). In this equation, τ_c is the capture time and τ_e is the emission time of defects.

$$P_{occ} = \frac{1-e^{\frac{-\alpha}{f\,\tau_C}}}{1-e^{\frac{-X}{f}}}(1-e^{-tX}) \qquad X = \frac{\alpha}{\tau_C} + \frac{1-\alpha}{\tau_e} \tag{4}$$

The initial CET map is multiplied with the occupancy probability map to generate CET-active map (Fig. 2c). This map describes the distribution of the occupied traps as a function of the applied stress waveform. By taking advantage of this map, the ratio of active (occupied) traps to total traps which is denoted by ρ (corresponding to the applied workload) is easily computed as follows [7]:

$$\rho = \frac{\iint f(\tau_c,\tau_e) P_{occ}(\tau_c,\tau_e,\alpha,t,f)\, d\tau_c\, d\tau_e}{\iint f(\tau_c,\tau_e)\, d\tau_c\, d\tau_e} \tag{5}$$

Then, for a specific device and stress waveform, the mean number of occupied traps, n_T, is computed simply by multiplying the mean number of available traps in that device (n) with the ratio of occupying of traps (ρ); i.e. $n_T = \rho \cdot n$. Each occupied trap has a certain contribution to threshold voltage degradation. The average impact of defects that is denoted by η can be determined experimentally. By knowing n_T and η, the total Cumulative Distribution Function (CDF) for ΔV_{th} can be obtained from the following equation [6,7]:

$$H(\Delta V_{th}) = \sum_{i=0}^{\infty} \frac{e^{-n_T} n_T^i}{i\,!}[1 - \frac{i}{i\,!}\Gamma(i, \Delta V_{th}/\eta)] \tag{6}$$

The mean and the standard deviation of the distribution can be used for describing transistor-aging features.

2.2. Gate-Level BTI

Timing analysis is an inevitable step in the design of digital circuits. Transistor-level simulations are able to analyze the timing of circuits with high accuracy. However, these methods are very time consuming and are not applicable to large circuits. Therefore, static timing analysis (STA) is widely used in timing analysis of large digital circuits, which includes thousands or even millions of gates. In order to perform timing analysis, STA takes advantage of gate delay models. Various methods have been proposed to incorporate aging effects into the gate delay model. Methods based on lookup tables (LUTs) are common in STA [10]. The effects of aging can also be easily incorporated into these LUTs. However, since the aging severity of a gate depends on various factors (especially workload and temperature), in practice, LUT-based aging-aware STA approaches require a large number of lookup tables which is not desirable.

An alternative method is to obtain the delay degradation of a gate as a function of aging-induced shift in the parameters of its internal transistors. These methods first calculate the threshold voltage degradation of the transistors (based on workload) and then

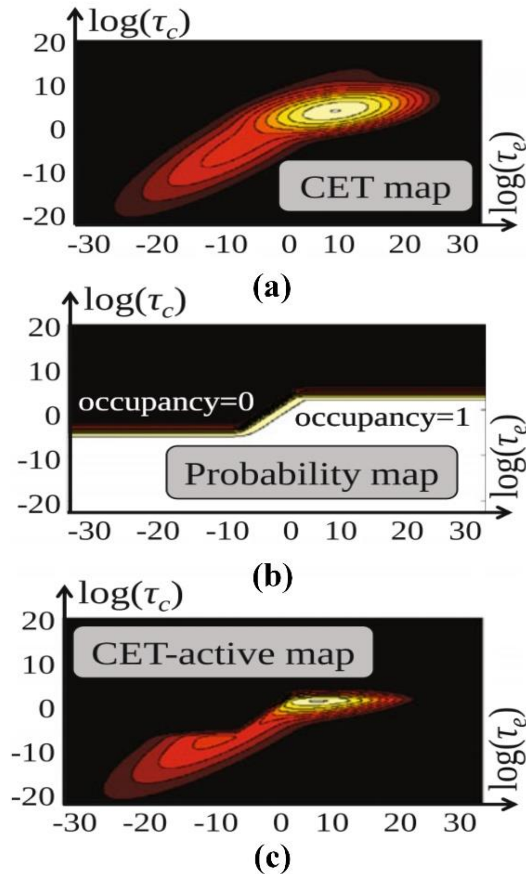

Fig.2. CET and CET-active maps [7].

determine the gate delay according to these parameter degradations using gate delay models. In [4], a compact model is proposed to predict the upper bound of the threshold voltage drift. An efficient model is proposed in [11] that considers the aging effects of individual transistors of a gate on its delay. In this method, aging effects on the transition time (slope) of the gate output are also considered. A method for analyzing the aging of circuits by taking into account the stochastic effects of aging is proposed in [8]. A machine learning-based aging prediction method is proposed in [12], which decreases the estimation runtime significantly and is suitable for situations where extensive aging analysis is required, e.g. in design exploration. However, it does not consider stochastic effects.

3. PROPOSED APPROACH

3.1. Problem Statement

Aging effects in circuits are manifested as an increase in their delay. Therefore, by incorporating the aging effects into timing analysis approaches, the aging severity of a circuit can be obtained as an increasing delay of the circuit over time. Transistor-level timing simulations are accurate but very time consuming. In addition, the simulation needs to be repeated for each new workload (input SP). Therefore, these methods are not applicable to large digital circuits. Due to their short execution time, aging-aware STA-based approaches can be used as a remedy for analyzing large circuits.

To meet the challenging requirements of digital system design, technology scaling has been introduced as an effective solution. Nowadays, manufacturing a complex system on a chip (SoC) is possible by taking advantage of Nano-sized transistors. In order to increase design productivity, a paradigm shift in digital design flow has emerged, with a greater emphasis on higher levels of design abstraction. Consequently, computer-aided design (CAD) tools now widely support design methodologies based on high-level synthesis (HLS) algorithms.

HLS explores the design space by taking advantage of processes such as scheduling and binding [12]. Due to the increasing importance of accelerated transistor aging, the lifetime of circuits should be addressed in the early stages of design. To do this, on the one hand, aging prediction models must be

fast enough for efficient exploration of the design space; and on the other hand, the (newly discovered) stochastic nature of Nano-sized transistor aging effects must be included in the analysis of circuit life.

3.2. Proposed ML-based Stochastic Aging Model

Transistor aging causes circuit delay to increase over time. The aging rate (or just aging for simplicity) is the rate of delay degradation in comparison to the zero-time (fresh) delay; i.e.:

$$aging = \frac{D(aged) - D(fresh)}{D(fresh)} \quad (7)$$

In this equation, D(fresh) and D(aged) denote the fresh and the aged delay, respectively. The aging rate is calculated at the end of year 10 in this paper. The aged delay is determined by a number of variables, including workload characteristics, effective temperature, and supply voltage [1]. It has been shown that the signal probability (SP) of an input signal (workload) is sufficient for aging analysis [11]. A vector of SPs of individual input signals (bits) can characterize the workload of a multi-input (multi-bit). For example, an 8-bit adder has two 8-bit ports, A(a7-a0) and B(b7-b0), and one bit for input carry (ci). Hence, workload features of this FU are represented by a vector of the primary input SP values as SP= (SP(ci), SP(b7), …, SP(b0), SP(a7), …, SP(a0)) [12].

For deep-nanometer transistors, the aging rate should be described as a statistical effect by means of a probability distribution with a mean value (μ) and standard deviation (σ). For a given primary input SP vector, the aging severity probability distribution of a circuit can be computed statistically by exploiting a Monte Carlo simulation. In the Monte Carlo simulation, the circuit is simulated iteratively and, in each iteration, different samples of the threshold voltage for individual transistors are chosen. The threshold voltage for each transistor is sampled according to its (aging-induced) degradation distribution. It should be noted that the distribution of the degraded threshold voltage of a transistor depends on the input SP for that transistor, which is calculated by propagating the primary SP vector towards that transistor. At the end of the Monte Carlo, the distribution, the mean value, and standard deviation of the aging severity (for the given primary SPs) can be determined.

In each Monte Carlo iteration, the number of occupied defects for each internal transistor (which depends on workload) is determined according to the Poisson distribution (with an average of n). The effect of each occupied defect on the degradation of the threshold voltage is determined according to an exponential distribution (with an average of η). The value of η has been determined experimentally and has been reported in research papers for a specific technology node [6]. The value of η for the intended technology can be determined by appropriate scaling. Finally, by summing the effects of all occupied defects, the total aging-induced degradation of the threshold voltage of each internal transistor is determined [7]-[8].

After determination of the (degraded) threshold voltage of all of the internal transistors of a gate, the degraded delay of the gate is determined (based on the new

parameters of its internal transistors). This is performed by exploiting the gate delay model. The gate delay model determines the degraded delay by considering the original delay, input transition time, and the delay sensitivity to threshold voltage shift. Sensitivity values are calculated by simulation in the phase of standard cells characterization. For a more accurate timing analysis, the degradation of the transition time (slope) of the gate output is also calculated (in a similar approach as delay determination) [11]. The total delay of the circuit is determined by exploiting STA methods. STA is performed in a block-based approach. Aging-induced degraded delays and output transitions of the gates are used in STA. Finally, at the end of the Monte Carlo simulation, the probability distribution of aging severity is determined, and the statistical characteristics of the aging severity are obtained.

By repeating the Monte Carlo simulation for different workloads (primary SPs) and operating conditions, a large-enough set of (SP, aging distribution) pairs can be obtained. Due to the relatively short runtime of the STA-based aging analysis (compared to transistor-level methods), a large set of these pairs can be obtained in a reasonable time. Block-based STA has a time complexity of O(n), with n being the number of gates. Therefore, it can be applied to large circuits.

After obtaining the training set, a machine learning (ML) regression technique is developed to estimate statistical characteristics of stochastic aging severity. In other words, the mean value and standard deviation of the distribution of stochastic aging-induced delay degradation are predicted by the model. The detailed steps of the workflow are depicted in Fig. 3.

3.3. ML-based Estimation Models

Due to their efficiency, ML techniques are becoming popular in digital design and related CAD tools. In this section, it is shown that ML regression models provide a fast and precise aging estimation. Various ML-based models can be used for developing aging estimation framework, where each of them has different efficiency. Here, a number of effective methods for ML-based regression are introduced.

Ridge regression model falls into the category of linear regression algorithms and is fitted by using the L2 regularization method. This model can be used to check the linearity of the relationship between features. This model has a hyper-parameter, called alpha, which is set manually to tune the regularization.

Artificial Neural Network (ANN) discovers the non-linearity relationship in data. The ANN-based nonlinear model is relatively complex and shows very high efficiency in many applications. For developing this model, we should decide about the number of hidden-layers and the number of neurons of each hidden-layer.

Support Vector Machine (SVM) for regression (SVR) can be used to develop models with lower complexity compared with ANN. SVM techniques are very popular in ML, where using kernel tricks, such as Radial Basis Function (RBF)extend their efficiency.

Random Forest (RF) approach constructs a multitude of decision trees that are trained

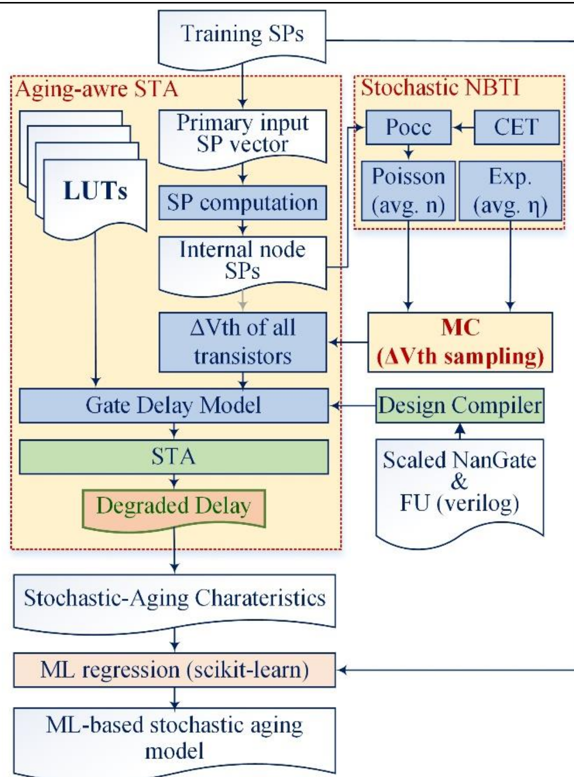

Fig. 3. Workflow.

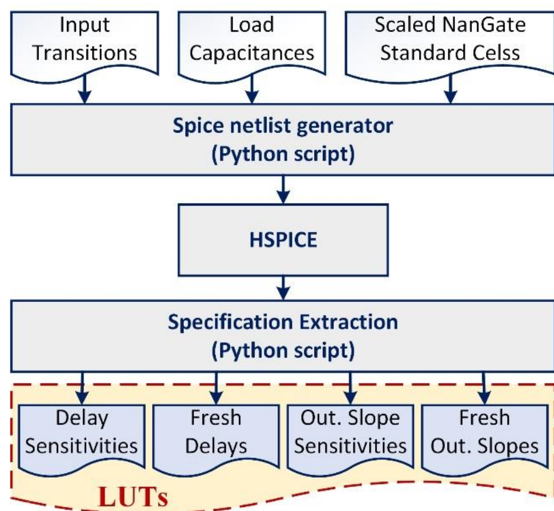

Fig. 4. Standard cell characterization.

using a training set. RF discovers the importance of different features in a regression model. This model falls into the category of ensemble algorithms.

Stacked Generalization, also called stacking, is an ensemble machine-learning model that combines some ML algorithms. Some base estimators pass their individual predictions to a final regression model. This final regression model computes the final prediction based on the results of base estimators. Stacking regression takes advantage of multiple ML algorithms, and therefore, mitigates over-fitting and under-fitting problems.

4. EXPERIMENTS

4.1. Experimental Setup

Standard cells of the library are characterized by exploiting extensive sensitivity analysis. By performing this analysis, the delay sensitivity of each gate to the threshold voltage shift of all its internal transistors is obtained. At the same time, the fresh (zero-time) delay and the degradation of the output transition time of gates are determined. All of these values are obtained using extensive HSPICE simulations, with the 7nm predictive technology model (PTM) [13], for different values for load capacitances and operating conditions such as temperature. To do this, the required netlists are generated by a Python script. The results are then read and the corresponding LUTs are filled with another Python script, Fig. 4. These LUTs are necessary for aging-aware STA which is used for data collection and training set generation.

To perform STA analysis, a Verilog description of the circuits is first developed, and then synthesized by Synopsys Design Compiler under simultaneous constraints of speed and area. In order to perform the

synthesis flow, a 7-nm standard-cell library is developed by scaling the cells of NanGate 45nm Open Cell Library [14], an open-source standard-cell library based on 45nm bulk CMOS technology. However, the proposed method is not limited to any particular standard-cell library and technology node.

For a given workload, the SP values of all internal nodes of the gate-level description of the circuit are obtained by propagating the SP values of the primary inputs of the circuit. Based on the SP values of the gate inputs, the mean value of the occupied traps of each internal transistor of the gate is determined (Equation 3). The average impact of each defect on threshold voltage degradation is also specified for the technology node. A Monte Carlo analysis, combined with aging-aware STA, is performed to obtain the statistical characteristics of the degraded delay of the functional units (FUs) and some known benchmarks. These statistical values are obtained based on the distribution of the degraded threshold voltage of each internal transistor. Most of these tasks are performed using python scripts as depicted in Fig. 3.

Data collection for training the model is essential to ML techniques. By repeating the MC for different workloads and temperatures, a large enough training set for a ML-based prediction model is obtained. Two separate training sets, one set in form of (Primary SPs, the mean value) pairs and the other set in form of (Primary SPs, the standard deviation) pairs, are used to develop two regression models.

Scikit-learn [15], a free Python library that features various ML-based regression algorithms, is used to develop and evaluate the estimation models. RidgeCV method, which consists of the Linear Model package of the Scikit-learn, implements ridge regression with a built-in grid search and cross-validation for tuning the model parameters, such as the alpha parameter. In this work, we use three-fold cross validation and grid search of different alpha values (0.001, 0.01, 0.1, 1).

To find the best parameters of ANN, SVR, and RF models, the built-in GridSearchCV function is used. This function implements the grid search and cross validation for tuning parameters of any ML-based model. In this work, three-fold cross validation is used. ANN efficiency for aging estimation is evaluated using MLPRegressor function, which is implemented in the Neural Network package. SVR regression with the RBF kernel is used in this work. It is an implementation for support-vector-machines regression algorithms and exists in the sklearn.svm package. In addition, RandomForestRegressor method existing in the sklearn.ensemble package is used for evaluating the efficiency of Random Forest (RF) algorithm.

StackingRegressor function as an implementation of stacking regression algorithm in scikit-learn is used in this work. Random forest and SVR are used for the base estimators and a neural-network model performs the final prediction. The parameters of the base models are equal to the parameters of these models in previous experiments.

The efficiency of the proposed approach and estimation models is evaluated in the aging prediction of some FUs such as adder and subtractor. In addition, some of the ISCAS85 benchmark [16] circuits are used.

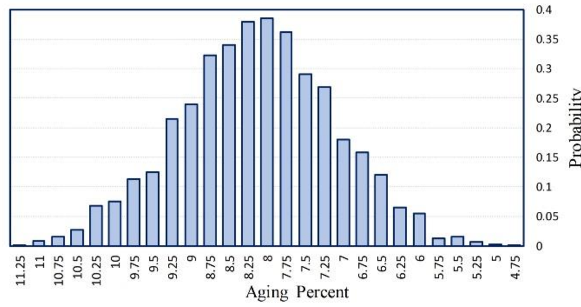

Fig.5. PDF of aging severity for an Adder corresponding to a given SP.

Table 1. ***Data collection runtime and number of gates for each benchmark.***

Benchmark	# gates	Data collection runtime (s)
Adder	492	670*1000
Subtractor	511	652*1000
Comparator	251	312*1000
C432	160	236*1000
C880	383	385*1000
C1355	546	692*1000

Experiments were run as single threads on a personal computer with an Intel Core i7 Q740 processor (1.73GHz, 6MB Cache) and 4GB RAM.

4.2. Experimental Results

The probability distribution function (PDF) for aging severity corresponding with a given primary input SP vector is depicted in Fig. 5. This is obtained by exploiting a Monte Carlo simulation. The horizontal axis shows the aging severity (in percent) and the vertical axis denotes the corresponding probability. The mean value and standard deviation of aging severity are computed using this distribution, after completing the MC.

Data collection for producing the training set is performed by MC for each input SP vector. Each iteration of MC is based on the aging-aware STA method, and the MC simulation is completed after (up to) 500 iterations. Finally, MC is repeated for 1300 different input SP vectors to produce the desired pairs for training the model. The Runtime of data collection and number of gates are reported in Table 1 for different circuits. The runtime is shown as a multiplication of a MC simulation for a given SP by the number of repeating the MC (number of SP vectors for training set).

The proposed approach estimates the mean and standard deviation of aging severity distribution for the benchmark circuits and FUs, using various ML-based regression models. Best combination of parameters of the models for each regression algorithm is discovered by exploiting a grid search combined with cross validation. The runtime of training of each model is depicted in Table 2. Parameters of models are also listed in the table.

The runtime of the MC-based aging analysis for each circuit is shown in Table 3, which is approximately equal to the runtime of aging-aware STA (one iteration of MC) multiplied by the number of MC iterations. The runtimes of different ML-based regression methods are also reported in Table 3. The machine learning method produces two separate models, one for estimating the mean value and the other for estimating standard deviation. The runtime shown in the table includes the total execution time of

Table 2. ***Training runtime and parameters of each ML-based regression model***

Estimation model	Training runtime (s)	Model Parameters
Ridge	6.73	Alpha=0.1
ANN	6841	power-t=0.5 alpha=0.001 #neurons=100
SVR	259	gamma=0.001 C=100

Random forest	3184	#estimators=150
		depth=20

Stacking	3252	-

Table 3. The runtime ***of aging analysis for different approaches*****.**

Circuit	Runtime (msec)					
	Rdg	ANN	SVR	RF	Stack	MC
add	0.73	1.68	1.22	1.57	2.45	1341
sub	0.73	1.67	1.21	1.57	2.45	1304
comp	0.72	1.68	1.19	1.57	2.45	624
C432	0.73	1.67	1.19	1.57	2.45	472
C880	0.72	1.68	1.22	1.57	2.45	771
C1355	0.73	1.68	1.22	1.57	2.45	1384

Table 4. ***The mean value of aging severity obtained by MC, and accuracy of ML models*****.**

Circuit	The Mean value of aging severity					
	Accuracy (compared to MC) %					MC %
	Rdg	ANN	SVR	RF	Stack	
Add	94.3	98.1	97.3	97.9	98.5	7.84
Sub	95.2	98.4	97.8	98.2	98.9	7.21
comp	95.7	98.3	98.1	98.5	98.6	8.12
C432	94.9	98.0	97.2	97.8	98.3	7.43
C880	95.3	98.3	97.9	98.4	98.7	8.25
C1355	95.4	98.5	97.8	98.2	98.9	7.49
average	95.13	98.27	97.68	98.17	98.65	NA

Table 5. The standard *deviation of aging severity obtained by MC, and accuracy of ML models.*

Circuit	Standard deviation of aging severity					
	Accuracy (compared to MC) %					MC
	Ridge	ANN	SVR	RF	Stack	
Add	96.8	98.4	97.9	98.1	99.1	0.94
Sub	96.7	98.5	98.2	98.3	98.9	0.76
comp	97.1	98.7	98.3	98.5	99.2	0.81
C432	96.6	98.7	98.1	98.4	99.3	0.89
C880	97.2	98.6	98.1	98.8	99.1	0.68
C1355	96.5	98.7	98.1	98.5	98.9	0.73
average	96.82	98.6	98.12	98.43	99.08	NA

these two models. The results show a significant reduction (more than 99%) in the runtime of aging analysis with ML-based methods compared to the MC method. This improvement is achieved with the cost of offline training runtime.

To evaluate the accuracy of the proposed method, the Monte Carlo method and the proposed ML-based models are used to extract statistical characteristics of stochastic aging severity for different circuits and corresponding to different workloads. The analysis is repeated for 100 different SPs. The average values of the mean value and standard deviation using MC for each circuit are shown in Table 4 and Table 5, respectively. The accuracy of the ML-based models compared to the MC results is also reported in these tables. The results show that the linear model has the weakest efficiency in aging analysis and the stacking method is the most efficient model. Ensemble algorithms such as random forest and stacking regression are effective for predicting aging characteristics.

It should be noticed that these values are obtained with significantly reduced runtimes, as depicted in Table 3. In summary, the experimental results show that the proposed method obtains the statistical characteristics of stochastic aging with up to 98% accuracy with more than 99% reduction in runtime. Therefore, the proposed approach is suitable for aging analysis for exploring the design space at high levels of abstraction.

5. CONCLUSIONS

In this paper, a SVR-based stochastic aging prediction model was developed. The training set was obtained by extensive Monte Carlo simulations, using different primary SP vectors. The proposed method reduced the prediction runtime up to 99%, while the results were accurate as 99%. Training the model needed significant time. However, it

was performed once as an offline process, before the prediction, and had no impact on prediction runtime. The model satisfied the requirements of aging analysis at high levels of abstraction.

REFERENCES

[1] E. Maricau, G. Georges, “Analog IC reliability in nanometer CMOS”, Springer Science & Business Media, Januery 2013.

[2] S. Es' haghi, M. Eshghi. “Lifetime-aware scheduling in high level synthesis”, Microelectronics Reliability, Vol. 91, pp. 86-97, December 2018.

[3] N. Khoshavi, R. Ashraf, R. DeMara, S. Kiamehr, F. Oboril, M. Tahoori, "Contemporary CMOS aging mitigation techniques: Survey, taxonomy, and methods", Integration, the VLSI Journal, vol. 59, pp. 10-22, September 2017.

[4] W. Wang, V. Reddy, A.T. Krishnan, R. Vattikonda, S. Krishnan, Y. Cao, “Compact modeling and simulation of circuit reliability for 65-nm CMOS technology”, IEEE Trans. Device Mater. Reliab. Vol. 7, No. 4, pp. 509–517, December 2007.

[5] B. Kaczer, T. Grasser, P.J. Roussel, J. Franco, R. Degraeve, L.-A. Ragnarsson, E. Simoen, G. Groeseneken, H. Reisinger, “Origin of NBTI variability in deeply scaled pFETs”, IEEE International Reliability Physics Symposium, IRPS’10, pp. 26–32, Garden Grove, CA, USA, May 2010.

[6] B. Kaczer, S. Mahato, V. V. de Almeida Camargo, M. Toledano-Luque, P.J. Roussel, T. Grasser, F. Catthoor, P. Dobrovolny, P. Zuber, G. Wirth et al., “Atomistic approach to variability of biastemperature instability in circuit simulations”, IEEE International Reliability Physics Symposium, IRPS’11, Monterey, CA, USA, pp. XT–3, April 2011.

[7] P. Weckx, B. Kaczer, M. Toledano-Luque, T. Grasser, P.J. Roussel, H. Kukner, P. Raghavan, F. Catthoor, G. Groeseneken, “Defect-based methodology for workload-dependent circuit lifetime projections-application to SRAM”, IEEE International Reliability Physics Symposium, IRPS’13, Monterey, California, USA, pp. 3A-4, April 2013.

[8] S. Kiamehr, P. Weckx, M. Tahoori, B. Kaczer, H. Kukner, P. Raghavan, G. Groeseneken, F. Catthoor, “The impact of process variation and stochastic aging in nanoscale VLSI”, IEEE International Reliability Physics Symposium, IRPS’16, pp. CR–1, Pasadena, California, USA, April 2016.

[9] H. Reisinger, T. Grasser, W. Gustin, C. Schlunder, “The statistical analysis of individual defects constituting NBTI and its implications for modeling DC- and AC-stress”, IEEE International Reliability Physics Symposium, IRPS’10, pp. 7-15, Garden Grove, CA, USA, May 2010.

[10] Liang-Chi Chen, Sandeep K. Gupta, and Melvin A. Breuer, “A new gate delay model for simultaneous switching and its applications”,

ACM/IEEE Design Automation Conference, DAC'01, pages 289–294, New York, NY, USA, 2001.

[11] D. Lorenz, M. Barke and U. Schlichtmann, "Efficiently analyzing the impact of aging effects on large integrated circuits", Microelectronics Reliability, vol. 52, no. 8, pp. 1546-1552, Augest 2012.

[12] S. Es' haghi, M. "Aging-aware scheduling and binding in high-level synthesis considering workload effects", Microelectronics Reliability, Vol. 106, pp.1135-1149, March 2020.

[13] "Predictive Technology Model (PTM)," Available: http://ptm.asu.edu/.

[14] "Nangate 45 nm Open Cell Library v1.3," [Online] Available: http//www.nangate.com.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al. "Scikit-learn: Machine learning in Python", Journal of machine learning research, Vol. 12, pp. 2825-2830, November 2011.

[16] Web.eecs.umich.edu. ISCAS_HLM. [online] Available: <http://web.eecs.umich.edu/~jhayes/iscas.restore/>