Activity-aware prediction of Critical Paths Aging in FDSOI technologies

K. Senthamarai Kannan, Michele Portolan, Lorena Anghel

Version of Record: https://www.sciencedirect.com/science/article/pii/S0026271421002274
Manuscript_252de1f0242bbd7ebf80d92731f8839e

Activity-Aware Prediction of Critical Paths Aging in FDSOI Technologies

Kalpana Senthamarai Kannan, Michele Portolan, Lorena Anghel[2]

University Grenoble Alpes, CNRS, Grenoble INP[1], TIMA, INAC-Spintec[2], 38000 Grenoble, France

{name.surname}@univ-grenoble-alpes.fr

Abstract*— Modern CMOS technologies such as FDSOI are affected by severe aging effects that do not only depend on physical issues related to nanoscale technologies, but also on the circuit environment and its run-time activity. Therefore it is extremely difficult to reliably establish a-priori guard bands for Critical Path estimations, usually leading to both large delay penalties (and therefore loss of performances) or too short operating lifetime. In this paper, we propose an approach that uses Machine Learning techniques to obtain reliable predictions of the aging of the Near Critical Paths. Starting from a limited set of measurements and simulation data, our framework is able to accurately estimate Critical Path delay degradation in time depending on physical parameters, environment conditions and circuit activity. Further to that, the corresponding regression models are applied to obtain dynamic aging-aware Operating Performance Point selection strategies.

Keywords—Circuit Aging, Prediction, Machine Learning, FDSOI

I. INTRODUCTION

Due to technology scaling and transistor size getting smaller and closer to atomic size, the last generation of CMOS technologies present important variability of several physical parameters. As a consequence, it becomes more and more difficult to guarantee circuit functionality for all Process, Voltage, Temperature (PVT) corners and in turn, to compensate for different sources of variability. Moreover, circuit wear-out degradation leads to additional temporal variations, potentially resulting in timing and functional failures. Under normal operation conditions, a transistor can be affected by various aging effects such as Hot Carrier Injection (HCI), Negative/Positive Bias Temperature Instability (NBTI,PBTI), and Time-Dependent Dielectric Breakdown (TDDB). In advanced technologies, such as FDSOI, local and global variability, NBTI and HCI phenomena are considered as critical reliability issues. Hence, taking into account these phenomena as early as possible in the design steps (i.e. during the standard cells characterization step, or at the circuit and system design) are mandatory, especially for high reliable application such as automotive applications, or mixed critical applications[1][2].

Indeed, the above-mentioned reliability threats can severely degrade performances, and in the worst case, provoke system failures, affecting safety goals of critical reliable systems. Accurate simulations with physical degradation models of aging phenomena combined with actual silicon measures are, de facto, necessary to better understand and assess the reliability impact on complex digital designs. To handle such problems, one conventional method consists in providing more safety margins (also called guard bands) at design-time. Adding pessimistic timing margins (or their equivalent voltage margins) to guarantee all Operating Points under worse case conditions is not possible anymore due to the huge impact on design costs, with an upward trend as technology moves further. Therefore, the usage of delay violation monitors, usually placed at the end of potential critical paths, becomes necessary. Placing the monitors in a given design is a critical task: the designer has to select the endpoints that will age the most, as it may become a potential point of failure. Monitor warnings signals can trigger adaptive techniques, such as Adaptive Voltage Scaling (AVS) or Dynamic Voltage Frequency Scaling (DVFS). They are then used to adapt dynamically the frequency and the voltage according to the operating conditions and the application needs [3][4]. In addition to the reduction of design margin, monitors also help compensate performance and power degradation. Sometimes, the circuit's lifetime can be extended. It is worth noticing that the area overhead induced by the monitor placement should be carefully considered and should remain reasonable. The number of selected endpoints for monitor insertion should be as small as possible, but still cover the most important critical endpoints of the design [5]. However, endpoint selection is an extremely complex task which requires a deep knowledge of both the target technology and the final workload.

To alleviate these restrictions, in the paper we propose a Machine Learning method that starting from a limited set of technological parameters is able to efficiently predict the delay degradation of paths depending on a given workload and available Operating Performance Points (OPP) expressed in terms of Voltage and Frequency. The aim is to obtain a lightweight, embeddable solution that can be used in conjunction with delay violation monitors in order to alleviate monitor insertion complex task, but also and to identify the best OPPs following different optimization strategies. The paper is organized as follows: Section II presents the state of the art, while Section III introduces the theoretical framework underlining the Machine Learning approach. Section IV presents the ML framework, which is validated and compared with the State-of-the-Art data in Section V. Section VI then proposes System-level innovative applications of the Method to compose Path Slack Ranking and proposes an aging-ware OPP adaptation strategy, while lastly Section VII draws conclusions and points future developments.

II. STATE OF THE ART

Adaptive Voltage and Frequency Scaling architectures (AVFS) or Adaptive Body Bias techniques (ABB) have been used since late 2000s to decrease safety margins and compensate for variations [6][7]. Such techniques use

embedded performance monitors inserted at strategic points within the design to track circuit timing fluctuations. The monitors are combined with adaptive voltages and/or frequency management schemes to reduce energy consumption and avoid timing errors [7][8]. The efficiency of these approaches is directly impacted by the quality and performances of these monitors: [8][9][10][11] use Razor like Flip flop as monitors implementations, while [12] uses an indirect form of monitor implementation. The insertion of monitors in a given design is done once the monitor has been designed, modeled, characterized, and validated.

The selection of the endpoints for monitor insertion is a time-consuming work, as designers have to consider all worst-case critical paths corresponding to all corner cases, including PVT variations, aging degradation, and workload influence on the delay degradation. To avoid critical paths delay overestimation, path delay degradation is sometimes simulated in SPICE, increasing, even more, the design and validation time [13].

The conventional method used for monitor insertion in a given design is to obtain a list of critical paths from static timing analysis (STA) after physical synthesis, usually during the Place & Route stage (i.e. after the placement of gates and after the clock-tree synthesis CTS optimization, and once the verifications step of the detailed timing analysis are done)[10]. For the chosen functional corner, a decision is made to target the worst critical paths for monitor insertion, and to regenerate connectivity and delay calculation for all selected critical paths. For the updated netlist, timing and power estimation are performed and afterwards the standard flow is executed with detailed routing and optimization (timing, power, IR drop, signal integrity, etc.).

This method may generate a huge number of monitors. In fact, current SOC designs have hundreds of thousands of critical paths and their corresponding endpoints. Thus, a careful selection of endpoints to be monitored has to be performed. One solution would be to extract from the first number of endpoints only a subset of endpoints and to monitor a few selected sub-critical paths [15]. Reducing the subset to 20% or 10% of the most critical paths can be a potential solution. This is what is done in numerous designs today in the attempt to use a smaller subset of paths by combining all design, PVT and environment conditions [16]. However, the number of endpoints can still be quite high, generating significant area overheads. In addition, monitor output signals collection in reasonable time is a real challenge and can turn this approach into an unpractical technique, with difficult or impossible scalability. It is therefore mandatory to select only meaningful sensitive critical and subcritical paths to be monitored for setup delay violations, while considering all realistic combinations of PVT, aging and workload while taking physical level aging-related phenomena.

Authors of paper [17] have studied the impact of aging on logic gates and introduced an aging analysis flow at gate level and an aging-aware analytical timing model for macro cells. Their methodology is based on a graph theory, whose combinatorial complexity can quickly become unbearable in several applications, such as in embedded setups were prediction time is critical to assure a correct reaction. Machine Learning (ML) techniques have been widely implemented to represent complex linear and non-linear relationships between the inputs and outputs of mathematical models with limited computation complexity. They are widely used in signal processing and pattern recognition, and in general in all situation where computers have to deal with specific problems through data and experience learning. Machine Learning can be seen as a combination of three parts: tasks (classification, regression, clustering, anomaly detection, sorting, etc), models (linear models, SVM, tree models, neural networks, etc) features (such as statistical, business and automatically extracted features). Machine Learning techniques have been used in physical design to assist placement and routing tasks [18] and in congestion prediction of physical design detailed routing [19]. Machine Learning techniques have also been used in the testing domain, to improve the quality of testing and the overall test coverage of photonic integrated circuits [20], or analog and RF circuit [21] .The framework proposed in the following section is based on Machine Learning (ML) Algorithm used for path delay degradation prediction.

III. TIMING ERRORS PREDICTION FRAMEWORK

The estimation of aging-induced path delay is usually done by characterizing the NBTI and HCI aging effects at physical and transistor levels, extracting the meaningful parameters [22] and injecting them into higher abstraction levels (e.g. circuit or gate level). System and product-level derating can be obtained by STA back annotated with timing information extracted after circuit fabrication and testing [23]. The gate-level evaluations are based on physical SPICE- level simulations which require a huge set of fabrication parameters, which are usually quite difficult to obtain from foundries. Moreover, to limit the complexity of the analysis, the activity of gates is usually set to 50%, reducing the workload-aware analysis to a single point. While accurate analysis is indeed possible [8, 14], the difficulty in obtaining data and effectively processing it, limits its application to only a set of simple gates, with small number of conditions, making the application to real-world cases difficult.

To overcome this problem, we propose a two-step modelling approach:

- First of all, we will develop a precise Machine Learning model of representative gates, INV, NOR and NAND, in the target FDSOI technology considering all dependencies of the gate delay on PVT, aging and activity variations. This reference

data is usually available from foundries, and will be used to train ML model. The SPICE simulations are able to consider any UDRM (User-Defined Reliability Model) established based on the complexity of aging phenomena and for a given technology.

- In the second step we will exploit the inherent correlations between base cell implementations in a given technology to extend the ML prediction tool from base cells to all other gates, thanks to approach known as Logical Effort Conversion [27].

Thanks to this model, we are therefore able to predict aging degradation of all standard cell gates for a target technology, and apply it to the gates composing the Near Critical Paths as extracted from STA. The two frameworks are detailed in the next sections.

A. *Aging Delay Degradation Model*

The Aging Delay Degradation model aims at providing delay degradation computation while considering the impact of PVT, duty cycle and aging-induced parameters. The principle is to use a training set of technology and design parameters, on which both simulations and the measurements were performed, in order to derive the underlying correlations. The functions have been approximated using adaptive polynomial regression functions. A model for delay estimation of a MOSFET has been proposed and defined by T. Sakurai and R. Newton early in 1990 [24]. The propagation delay of a cell dependence on the voltage and output capacitance is expressed by the alpha-power law such as:

$$Delay_{Cell}\ \alpha\, C_{out}\frac{VDD}{I_d} \tag{1}$$

Where, C_{out}- output load capacitances, V_{DD} – supply voltage and I_d- Drain current. Starting from equation (1) , the Inverter propagation delay dependence of the voltage, temperature and time is given in equation (2) as shown in [25].

$$Delay(V,T,t) = p_\beta + p_{\mu-1}(T)\ \frac{V}{V - \left(pv_{th}(T)\right) + \Delta pv_{th}(V,T,t)^{p_\alpha}} \tag{2}$$

Where, p_β, p_α are constant while $p_{\mu-1}(T)$ shows exponential dependence on temperature and pv_{th} , related to transistors mobility and threshold voltage, respectively. The delay model equation including PVT variations consists of 8 parameters. They are specified in [25],[26] by the extended equations (3), (4), and (5).

$$p_{\mu-1}(T) = C_1 + k_1 T^{n1} \tag{3}$$

$$p_{vth}(T) = C_2 + k_2 T^{n2} \tag{4}$$

$$\Delta_{pvth} = V^\gamma * e^{-\frac{E_a}{kT}} * (C_1 * t^{n1} + C_2 * t^{n2}) \tag{5}$$

C_1, C_2, k_1, k_2, are fit parameters for a given technology (i.e., FDSOI 28 nm in our case) obtained from extraction of fit measurements curves with a very high degree of confidence; γ - voltage acceleration factor; E_a- temperature activation energy; k- Boltzmann's constant; (n_1, n_2) - two different exponents of temperature dependence. Each corner has its specific set of parameters for a given technology. NBTI and HCI aging induced degradations impact mostly the threshold voltage of transistors, but also the carrier mobility. This is captured in Equation 2 by the factor Δpv_{th}, which shows the dependence on temperature, voltage and time. Paper [22] presents a framework for transistor and gate-level modeling of NBTI and HCI aging phenomena, where the aging model is obtained by applying different stress levels to reference technology cells. . Note that equation (2) does not incorporate workload dependencies.

Workload influence on the gate delay

The workload influence or the Duty cycle (DC) is defined as a fraction of the input pulse width with respect to the total period of time the data is applied at the input. It is usually expressed as a percentage. In digital design, timing library files for each library cell are available to the designer. In these files, the input signal duty cycle of 50% is usually used for delay estimation. Starting from the delay computation with for 50% duty cycle, the delay for any other duty cycle can be computed using equation (6) as reported in [16],

$$Delay(DC) = Delay(0.5) * \frac{\tanh(x^\alpha)}{\tanh(1)} \tag{6}$$

Where, DC is the Duty Cycle or input signal probability at the inputs of standard cells, *x*: stands for the expression DC/(1-DC), α is a cell-dependent fit parameter depending on input signal slope and the output capacitance and Delay (0,5) is the delay measured for 50% input switching activity, computed with equation (2).

B. *Linear Delay Model Conversion*

Delay calculation by means of equation (6) can be done for one gate, but extending this computation to all possible gates form the standard library is extremely time-consuming. In addition, we need to be able to evaluate the total delay of a path, not only on individual cells. Therefore we use the well-known approach called Linear Delay Model [27] to get accurate and approximate propagation delay for most of the logic gates with an error rate lower than 2%. With this model, each logic gate is expressed in terms of its Effort delay, dependent on both the gate complexity and topology (gate effort) and on the electrical effort due to the gate fan-out and the Parasitic Delay. With a minimum number of inverter cell parameters physically characterized and

measured after fabrication, one can easily estimate the effort delay and the parasitic delay for any specific gate.

Finally, the aging delay model for specific gate is estimated by a combination of the Inverter aged delay expressed by equation (6) and, and the effort delay (LE) computed with the methodology presented in [27].

$$d_g = (Inverter\ Delay) * (Effort\ Delay)_g \qquad (10)$$

$$'(V,T,t)_g \qquad (11)$$

$$_g + p_{\mu-1}(T)\ \frac{V}{V - (pv_{th}(T)) + \Delta pv_{th}(V,T,t)^{p_\alpha}}\Bigg) * (LE)_g$$

This method has the advantage of combining delay evaluations for OPP-related switching activity along with corner analysis with sufficient accuracy.

IV. Machine Learning (ML) Delay Prediction

Machine Learning has been receiving growing interest, as many current applications need to process vast and massive amount of data. Modern Deep Learning and ML algorithms can give approximate and accurate results even with a smaller or incomplete set of data. In our study, the first step in the setup of a Machine Learning framework is to define a model based on the theoretical framework introduced in Section III, and then apply the chosen learning mechanism exploiting the few available data from foundries for the target FDSOI 28nm technology.

A. Machine Learning Flow

Our goal is to evaluate the delay of a basic logic gate based on a set of physical and electrical parameters, as well as the activity and aging over time. One of the supervised machine learning algorithms which better suits this application Linear Regression (LR). The prediction of delay degradation will be performed while minimizing the minimum root mean square error (RMSE) with respect to the Test data set. The ML model takes the relationship between the dependent and independent variables (X1, …Xn) and creates a generalized continuous output function [28], [29] as shown in equation (12).

$$y = \alpha + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * x_3 + \cdots + \beta_n * x_n \qquad (12)$$

Where α and $\beta_{1\text{-}n}$ are also called coefficients or parameters. A regression with more than one variable is called multiple regression. The sum of squared errors (SSE) is evaluated between the observed and the predicted results [28], with the aim of minimizing it. To proceed with the target design, the system is trained with features such as PVT, Duty Cycle, Workload activity, and time while performing gate or path delay prediction.

Training and testing flows are depicted in Figure 1. Training variables used in equation (11) were extracted from the standard cell library User-defined Reliability Models or measures, such as Process, Voltage, Temperature, Capacitances, Threshold voltage degradation and time. Each data set contains at least 10 degradation points for 10 or 12 years of gate operation. The extracted features have been split into training and testing data according to the usual 80/20 percentage ratio.

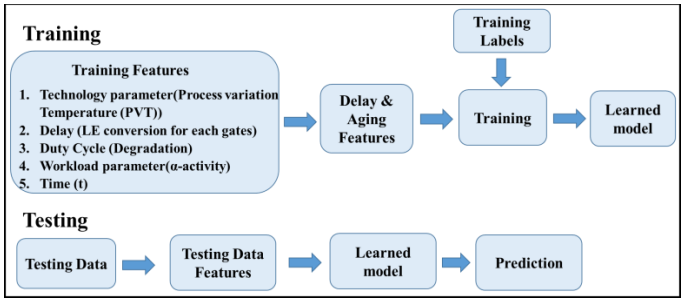

Figure 1: ML-based Delay Estimation flow

B. Application to Path Aging

As explained in Section III, we had detailed data only for NAND and NOR gates in FDSOI 28nm technology, so we applied Machine Learning to it in order to obtain a model able to predict their delay degradations based on Operating Points (OPP) conditions and activity parameters. Afterwards, we developed a complete Logical Effort framework able to extend its results to different gates configurations.

To obtain the delay degradation due to aging for the whole circuit we need three additional steps:

- Static Timing Analysis is performed to obtain all Near-Critical Paths delays as a cumulative effect of individual gates delays
- The Switching Activity of each gate of the circuit is extracted by combining Value Change Dump (VCD) simulation traces with timing analysis.
- Finally, ML is applied to each gate in a NCP, obtaining the overall aged Path delay.

V. Experimental Validation

To validate the approach, we need to validate the capacity of each step to predict data known from independent sources. This analysis is necessary to assess the confidence of the setup before applying it to new cases.

A. NAND/NOR Aging Validation

First of all, trained and validated the ML Model with data set taken from FDSOI 28nm foundry data, containing detailed measures and characterization for NAND and NOR gates: we applied a classical 80%/20% partitioning for training and test set, obtaining the results of Figure 2. We obtain aging induced degradations for the reference gates with respect to different Switching Activities (e.g. 10%, 20%, 50%, 80%, 90%).

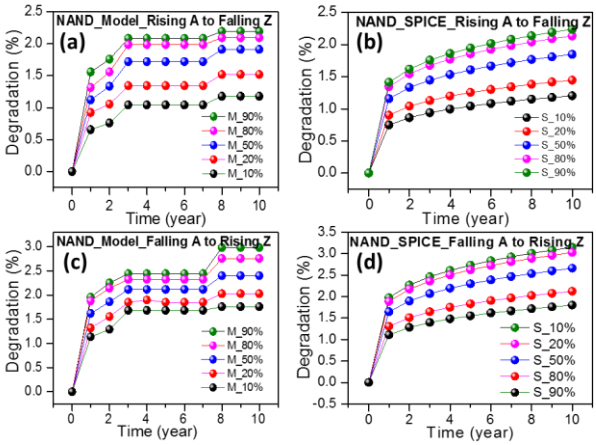

Figure 2 ML vs SPICE Aging degradation for NAND gates

The results for the ML Model presented on the left side of Figure 2 are extremely close to SPICE simulations shown on the right-hand side. Apart from some discontinuity, typical of MV models, the tendency is clearly the same. The results are the same for the NOR gate. A close analysis of the Root Square Mean Error (RMSE) degradation shows errors rates in the order of 1%, as depicted in Figure 3 for NAND and NOR gates for 10% and 90% switching activity.

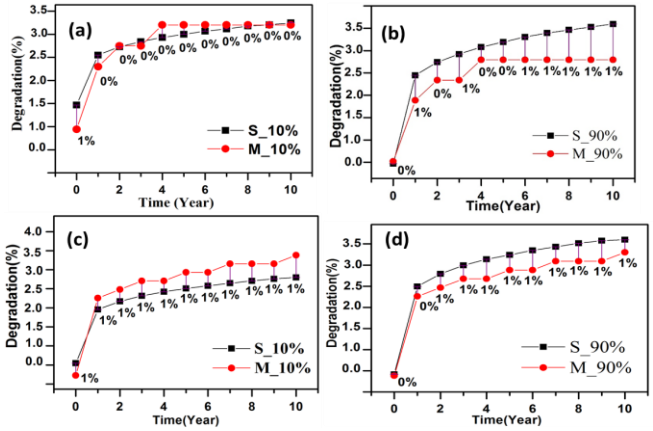

Figure 3 RMSE prediction error for NAND (a,b) and NOR (c,d)

Our Machine Learning Model is therefore able to accurately predict the aging of the reference gates NOR/NAND under different PVT corners and working conditions (switching activity).

B. Logical Effort Validation

Classically, Logical Effort [27] is applied using the Inverter as the reference gate, but our data set is relative to NAND and NOR. We therefore adapted the methodology and equations presented in the previous paragraph using these gates as references. The results of Figure 4 demonstrate our capability of extending the prediction to generic gates in our target technology. Even though we do not have reference results for all these gates, they are obtained by combining the ML prediction framework validated in the previous Paragraph and the Logical Effort approach from the literature [27]. This allows us to obtain reasonable predictions even in the absence of full foundry data.

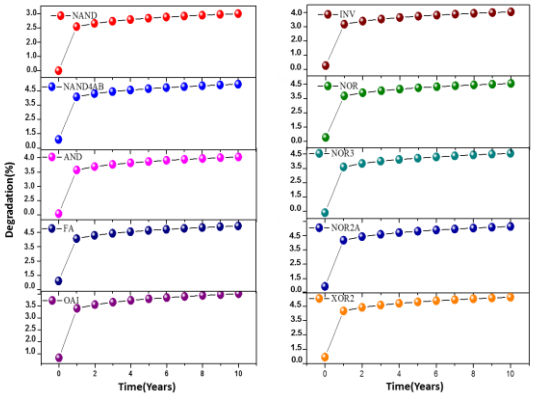

Figure 4 Gate Aging prediction using Logical Effort

C. Effect of Machine Learning Algorithm

As stated in Section V.A, the results were obtained using Linear Regression, which is computationally simple and easy to setup. To verify this assumption, we reproduced the same flow using the Random Forest algorithm [29] and we obtained similar results with a 1% error margin. This validates the independence of the proposed approach from the chosen ML algorithm, and justifies the choice of the less computationally-intensive Linear Regression.

VI. System-Level Application

A. Critical Path Aging Validation

The next step consists in Critical Paths delay computation as the sum of the age-induced delays of each gate composing it. To achieve this, we chose a System Under Test (SUT) to develop the prediction setup consisting in a digital FIR filter. Through a Gate-level simulation we monitor data changes in the Near-Critical Paths and save them in standard formats such as) VCD. A post-processing step can then extract the Activity values needed in ML Model to obtain the results of Figure 5.

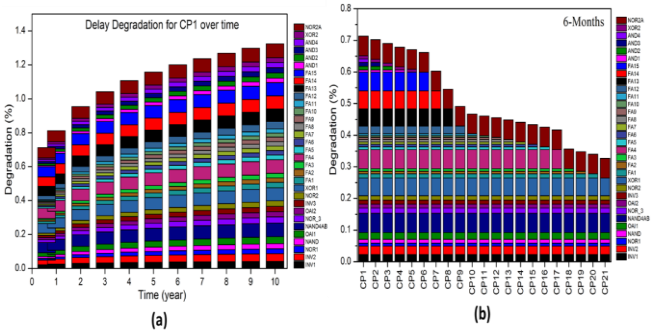

Figure 5 Aging Prediction of Critical Path for the FIR example

Figure 5-a depicts the Degradation of the Most Critical Path (CP1) over time, with each color highlighting the contribution of one of the gates composing it. Figure 5-b shows the degradation of the first 20 critical paths for a given aging time (i.e. 6 months). Paths are ordered by a decrease of their Slack Rank, with CP1 being the most critical (i.e. with the shortest slack) which determines the maximum operating frequency. These results are coherent with an architectural analysis of the Filter: the near-critical paths (NCPs) correspond with the most active parts of the circuit (the Least Significant Bits of the Multiplier/Accumulator), so they age accordingly. Moreover, CP are quite unbalanced, with great differences between them.

To prove the capacity of our approach to predict workload-based aging, we selected a more complex SUT: a 256-bits AES crypto-processor. This design presents a more uniform set of NCPs, and their activity is not correlated, as proven by Figure 6, obtained with our setup. As NCPs age differently and their absolute difference is small, this setup is an ideal candidate to observe the Ranking Inversions, i.e. paths that were not critical at Time 0 becoming problematic after some time due to aging. This phenomenon reported in [13]by the authors of the paper is extremely difficult to observe and predict using traditional flows because it depends on both low-level physical phenomena and high-level setups such as the workload.

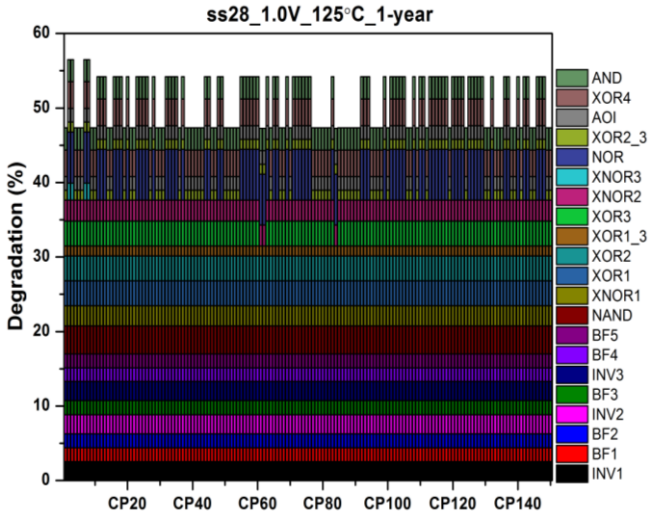

Figure 6 1-Year NCP Degradation for AES Cryptoprocessor for the first 150 Near-Critical Paths

The computational complexity of such simulations and their analysis is by itself a show-stopper. On the other hand, our ML framework is extremely lightweight and we proved its ability to efficiently predict Path Aging. Figure 7 shows the result of our ML age-induced delay degradation prediction flow to the AES core under two different workloads, one per column. For each Path we plotted its Slack Ranking: the larger time (i.e. the smaller slack) the path has, the higher its column is. At Time=0, the distribution is of course linear: paths are ordered by the results of the STA, and as the circuit has not been used yet the workload has no impact. Rows b) and c) show the effect of aging after 6 and 12 months respectively. As the ordering of paths on the X axis is unchanged, the effect of workload on path aging is clear. Path distribution is almost chaotic: each gate aged differently depending on its activity and the aging profile is extremely different between the two workloads.

This observed phenomenon is in fact one of the greatest drawbacks of Monitor Insertion flow: in each aged distribution, we highlighted in Red the 10 Most Critical Paths, on which Aging Monitors should be inserted. From this figure, it is obvious to understand that a choice made at Time 0 based on STA evaluation would not be coherent with an Aged system, making most inserted monitors unnecessary. Similarly, a purely architectural choice with no consideration of the Workload effect would be unable to capture the correct set of aged paths.

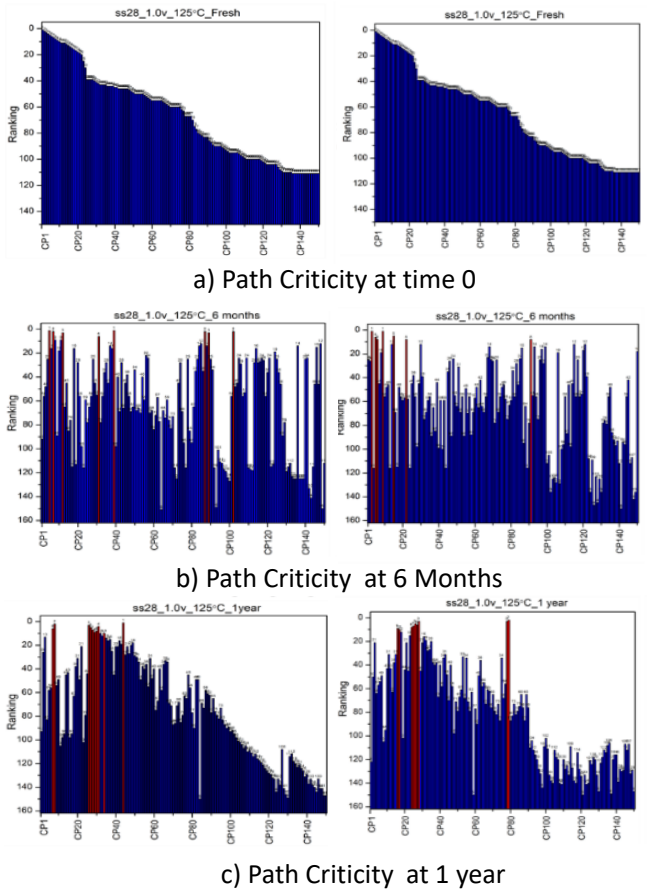

Figure 7 Evolution of Path Slack Ranking over Time

The size of the subset of paths to be considered as NCP is somewhat arbitrary and depends on target technology yield and product reliability targets: a larger set will provide a better coverage of delay faults, but with a higher area overhead.

B. Aging-Aware OPP Adaptation

When defining the Operating Performance Point (OPP) of a system, what matters most is not the absolute value of the propagation delays on the NCPs, but rather the absence of Delay Faults, i.e. data sampling faults caused by larger propagation delay exceeding the Working Frequency period. A Delay Monitor on a given endpoint will raise a flag when such fault happens: for an OPP to be viable, it is therefore mandatory that the Flag Count (i.e. the sum of monitors detecting a delay fault) be zero. Unfortunately, these conditions depend not only on the OPP, but also on the manufactured characteristics of the circuit (resumed as its fabrication corner), on the environment (e.g. temperature, VDD drop) and on its aging. Figure 8, obtained using our model and considering all Paths in the design, depicts the FlagCount for the FIR filter depending on the chose OPP and the impact of the different Corners (Fast, Slow and Typical).

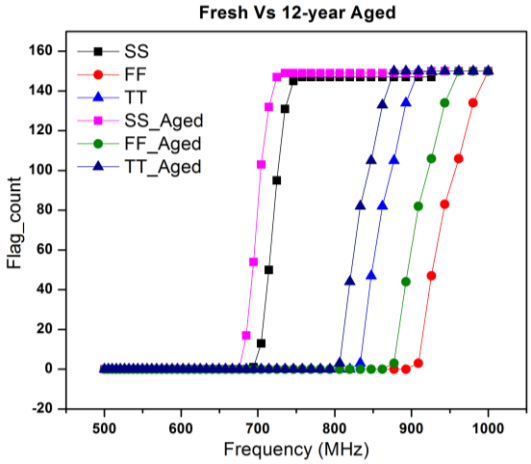

Figure 8 Corner Impact on Flag Count for FIR Filter synthetized at 500MHz for different Corners

First of all, the impact of process characteristics can be easily observed: at Time 0, circuits in the faster (FF) corner can work at almost 900 MHz without any Delay Fault, while the same circuit in the slower one (SS) cannot run at frequencies higher than 700 MHz. Moreover, the impact of aging is also clear: as time passes, flags arise at smaller frequencies, limiting the working frequency. As a result, the circuit will be deployed using an OPP guaranteeing a reasonable margin at the end of its service lifetime (e.g. 12 years), losing potential performance in its early life. This is even more pronounced for setups whose aging is unpredictable, as in the AES cryptoprocessor: in these cases, the aging margin might be very high due to the many parameters that need to be considered.

For a given Corner, an OPP is composed of two parameters: Frequency and Voltage. Up to now we considered only the former, but the choice of Voltage is also fundamental: on the one hand, higher voltages allow for higher frequencies, but on the other hand they can also seriously stress the circuit and cause a faster degradation. So, and aged circuit will need to operate at higher voltage to guarantee correct functional timing behavior at a given frequency, as shown in Figure 9, depicting the Flag Count on a 12-Year AES depending on Voltage.

Based on these result, for instance, to guarantee the circuit is operational at 400 MHz for its whole lifetime, a designer will be forced to choose a Voltage of at least 0,69V to be sure that the Flag Count at after12 years will be zero. However, at Time=0 this is a clear case of overdesign. In fact, the circuit might have worked at a lower Voltage for some time, but with the risk of a Delay Fault due to aging, inducing higher power consumption in early life, and a potential smaller lifetime. In fact, in a real scenario what matter is the Working Slack, i.e. the margin between the Critical Path and the Working Frequency Period at a given time point.

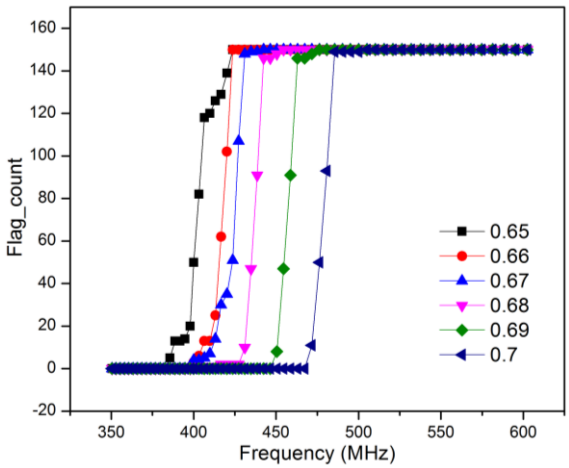

Figure 9 Impact of Voltage over aging for AES (12 years)

In a static OPP selection scenario as the one depicted in Figure 9, the Minimal Working Slack (MWS) is computed with the Worst-Case Scenario, i.e. the aged circuit at the end of its lifetime. Let's call it WS (12y) .This means that at any given time t < 12y, WS(t) >> WS(12y) = MWS. So, the margin is actually much higher than what it really needed.

To overcome this issue, we propose to apply the ML Prediction Framework to the choice of OPP, based on the principle of Time Window (TW): instead of considering the status of the circuit at the end of its lifetime (here we take 12 years), at any time 't' we predict its status at time 't+TW'. This Time Window (TW) reflects the aging timing

characteristics for a given technology, and by applying our Model we can predict its effect on the Working Slack and guarantee that WS (t+TW) > MWS: this is the minimal condition that guarantees that there will be no timing faults, i.e. the count FlagCount of monitors will remain 0 . The idea is to predict the effect of a given OPP periodically, rather than on the long period, and react more quickly to any possible failure risk without resorting to overdesigned margins. The key value is the Flag Count at time t+TW: if FlagCount (t+TW) = 0, the OPP is viable and can be used, otherwise it must be discarded and another one from the OPP stack will be analyzed. Once the OPP has been chosen, we move the analysis time step to t+TW, and repeat the process. This implies that at each step, we need to compute a new prediction: this is possible only thanks to the lightweight nature of our ML approach. The sheer combinatorial complexity would make this unfeasible using traditional simulation-based approaches, while our Model is lightweight enough to be executed on embedded processors to allow online periodic prediction and further adaptation.

1) Power Minimization for Fixed-Frequency OPP

Different OPP selection choices can result in different optimization schemes, obtaining great flexibility and adaptability. The most typical setup is targeting power optimization: while keeping performances stables with a fixed Working Frequency, select the lowest possible Voltage which assures the respect of the Minimal Working Slack. With reference to Figure 9, it means that for a given Frequency (X axis), we need to select the curve which guarantees Flag_Count=0. But as explained in the previous paragraph, this is a Worst-Case Scenario. By applying our ML model, we can guarantee the respect of MWS in all Time Windows by verifying that flag_count=0 at any time t+TW. Figure 10 depicts the results of OPP adaptation with a target TimeWindow of 4 months.

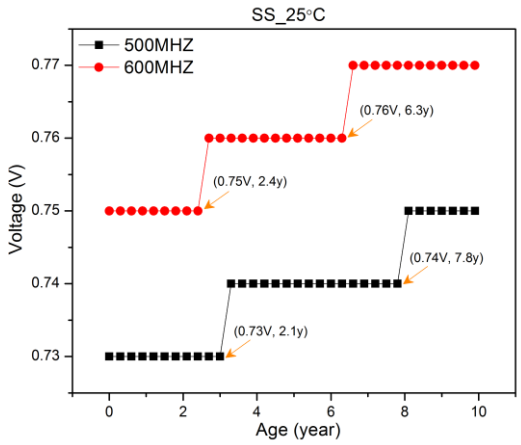

Figure 10 Voltage Minimization for Fixed-Frequency

In a traditional setup, we should have selected the OPP for the end of service (10 Years in this example): 0,75V for 500 MHz or 0,77V for 600MHz. Thanks to our approach, we only need to reach this value of the end of the lifetime, while for most of its mission modes we were able to work at lower voltages, with significant gains in terms of total power dissipation. We can see that for lower frequencies the circuit ages slower, so we can maintain a more efficient OPP even longer.

2) Incremental Time Window

In the previous example, we chose a fixed Time Window, arbitrarily set at 4 months, based on an appreciation of the aging behavior of the FDSOI 28nm technology. Anyway, this is a trade-off that does not reflect the reality of the aging at a given time 't': at the beginning of the lifetime a circuit ages much faster that towards its end-of-life. Therefore, a fixed Time Window is suboptimal: it is too large at the beginning, losing precision, and too small at the end, generating potential performance losses. As a solution, we implemented an Incremental Time Window, that is extremely small at the beginning of lifetime (i.e. 2 weeks) and gradually increases (up to roughly 5/6 months), as depicted in Figure 11.

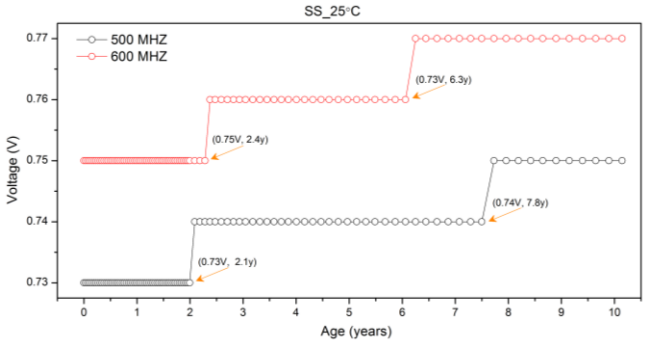

Figure 11 Fixed-Frequency OPP adaptation with Incremental Time Window

3) Aging-Aware OPP Overclocking

The fine-grain control of our approach allows for innovative optimization strategies: thanks to the TimeWindow, it is possible to dynamically select OPPs and change them before any Delay Fault arises. A possible application is overclocking: by applying high Voltages it is possible to set very high working Frequencies, with significant performance boosts. Unfortunately, these off-spec working conditions result in an accelerated aging: due to the difficulty of controlling these phenomena, these solutions are rarely applied.

However, thanks to our framework, we can define an overclocking OPP selection: at any TimeWindow, choose the OPP with the highest possible working frequency. The results are plotted in Figure 13.

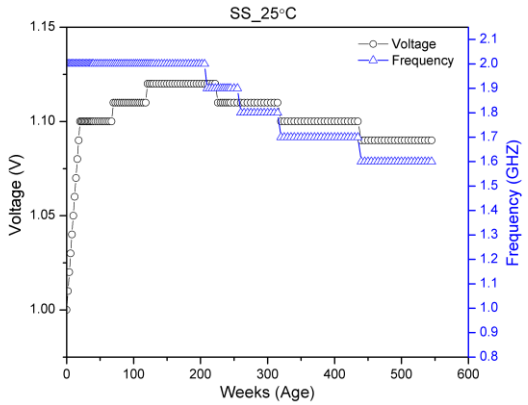

Figure 12 Overclocking OPP Optimization

Thanks to this very aggressive optimization strategy, we were able run an AES crypto processor synthetized for 500MHz/1V at a frequency 4 times higher than the nominal values for almost 4 years (200 weeks). Anyway, from this Figure we can also observe the stress this solution puts on the system: the aging curve is extremely steep, so that in the second half of the lifetime we are forced to lower both frequency and voltage to avoid delay errors. Even though this use case is purposefully extreme, it might be useful for performance-hungry setups with short lifetime (ex: consumer electronics) or on specific short-time high performance tasks and workloads.

4) Maximum Performance with OPP Cap

To show the flexibility of our approach, we implemented a more nuanced optimization strategy: we still look for the maximum performance at a given time step, but we defined a maximum limit (cap) for Voltage and Frequency to avoid over-stressing the system. The results, plotted in Figure 13 show how our prediction approach allows for fine-grain tuning.

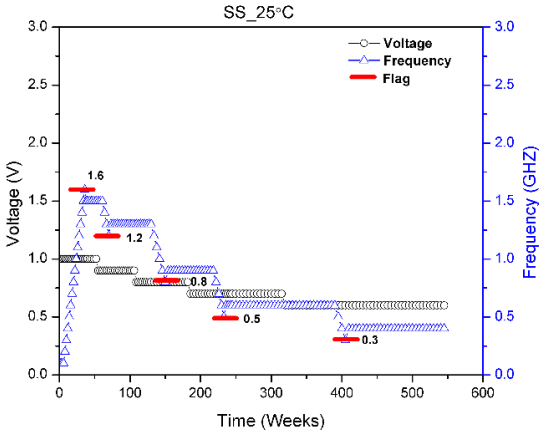

Figure 13 OPP optimization capped at 1V/1,5GHz

C. Convergence with Delay Monitor Insertion

The Machine Learning approach we proposed is complementary to traditional Monitor Insertion Flow, and presents several possible convergence points.

For starters, one of the main limitation of Monitors is the selection of insertion endpoints: because it is practically unfeasible to instrument all paths, it is necessary to efficiently choose a subset. As shown in Section VI.A and most notably in the results depicted in Figure 7, our approach allows us to efficiently identify the NCR with respect to both aging and workload. By selecting the Paths to instrument based on their aging profile, we can sensibly reduce the number of inserted monitors and therefore their area overhead.

Monitors can also be used to precisely measure Path propagation time: when one of them flags, it means the degradation has exactly the value set by its delay buffer. This can be used to palliate the problem all models share: drift. Our Model is built from design-time parameters like STA and foundry-based parameters and measures, so its precision is maximal at time=0. But as time passes, prediction errors might accumulate and the model might drift away from the real circuit. In this context, strategically placed monitors within the design can provide delay

propagation measures of aged paths, and therefore allow fine-tuning online calibration of the Model. In addition, dynamically adjustable Timing Window can also help adjusting OPP accordingly and help extending the lifetime at a manageable power consumption and performance.

VII. Conclusion and perspective

In this paper, we proposed an approach that by combining Machine Learning and theoretical analysis is able to accurately estimate the aging behavior of circuits in FDSOI 28nm technology with respect to both Operating Conditions and Workload. After validating it against known data, we proved its capacity to follow the evolution of Critical Paths over time and applied it to obtain a dynamic OPP adaptation framework. In this paper, we applied our methodology to FDSOI technology using well-known literature models: due to its genericity, it would be easy to modify it for other technologies or reliability user-defined models, as long as training data is available.

Future works will include the exploration of new OPP Optimization strategies and the integration with classical Monitor Insertion Flow, most notably to identify an optimal Path selection insertion strategy. We also plan on exploiting its computational simplicity for embedded applications to allow on-line aging-ware OPP adaptation.

Acknowledgment

This work has been partly funded by the French Government under the framework of the PENTA HADES ("Hierarchy-Aware and secure embedded test infrastructure for Dependability and performance Enhancement of integrated Systems") European project.

References

[1] V. Huard et al., "Adaptive wearout management with in-situ aging monitors," IEEE Int. Reliab. Phys. Symp. Proc., pp. 6B.4.1-6B.4.11, 2014.

[2] S. Taylor et al, "Power 7+: IBM's Next Generation Power Microprocessor," in Hot chips, 2012, p. Vol 24.

[3] L. Lai, V. Chandra, R. Aitken, and P. Gupta, "SlackProbe : A Low Overhead In Situ On-line Timing Slack Monitoring Methodology," 2013 Des. Autom. Test Eur. Conf. Exhib., pp. 282–287.

[4] M. Wirnshofer, L. Heiß, A. N. Kakade, N. P. Aryan, G. Georgakos, and D. Schmitt-Landsiedel, "Adaptive voltage scaling by in-situ delay monitoring for an image processing circuit," Proc. 2012 IEEE 15th Int. Symp. Des. Diagnostics Electron. Circuits Syst. DDECS 2012, pp. 205–208, 2012.

[5] K. Senthamaraikannan, M. Portolan, and L. Anghel, "Run-Time Aging Prediction Though Machine-Learning,", 2018 International Test Conference (poster presentation), 2018.

[6] J. Tschanz et al., "Adaptive frequency and biasing techniques for tolerance to dynamic temperature-voltage variations and aging," Dig. Tech. Pap. - IEEE Int. Solid-State Circuits Conf., pp. 292–294, 2007.

[7] A. Sivadasan, R. J. Shah, V. Huard, F. Cacho, and L. Anghel, "NBTI aged cell rejuvenation with back biasing and resulting critical path reordering for digital circuits in 28nm FDSOI," Proc. 2018 Des. Autom. Test Eur. Conf. Exhib. DATE 2018, vol. 2018-Janua, pp. 997–998, 2018.

[8] L. Anghel, A. Benhassain, A. Sivadasan, F. Cacho, and V. Huard, "Early system failure prediction by using aging in situ monitors: Methodology of implementation and application results," 2016 IEEE 34th VLSI Test Symp., pp. 1–1, 2016.

[9] D. Ernst et al., "Razor: A low-power pipeline based on circuit-level timing speculation," Proc. Annu. Int. Symp. Microarchitecture, MICRO, vol. 2003-Janua, pp. 7–18, 2003.

[10] A. Benhassain et al., "Timing in-situ Monitors : Implementation Strategy and Applications Results," 2015 IEEE Cust. Integr. Circuits Conf., vol. 33, no. 0, pp. 1–4, 2015.

[11] M. Nicolaidis, "Time redundancy based soft-error tolerance to rescue nanometer technologies," Proc. 17th IEEE VLSI Test Symp. (Cat. No.PR00146), pp. 86–94, 1999.

[12] A. K. Uht, "Uniprocessor performance enhancement through adaptive clock frequency control," IEEE Trans. Comput., vol. 54, no. 2, pp. 132–140, 2005.

[13] A. Sivadasan et al., "Workload dependent reliability timing analysis flow," Proc. 2017 Des. Autom. Test Eur. DATE 2017, pp. 736–737, 2017.

[14] B. Halak, Ageing of integrated circuits : causes, effects and mitigation techniques. Springer, Cham, 2020.

[15] A. Benhassain, S. Mhira, F. Cacho, V. Huard, and L. Anghel, "In-situ slack monitors: taking up the challenge of on-die monitoring of variability and reliability," in 2016 1st IEEE International Verification and Security Workshop (IVSW), 2016, vol. 6, pp. 1–5.

[16] A. Sivadasan et al., "Architecture-and workload-dependent digital failure rate," IEEE Int. Reliab. Phys. Symp. Proc., p. CR8.1-CR8.4, 2017.

[17] D. Lorentz, M. Barke, U. Schlichtmann, « Monitoring of aging in integrated circuits by identifying possible critical paths », Microlectronics Reliability, Volume 54, issue 7, 2014, pages 1075-1082.

[18] David Pan B. Yu, D. Z. Pan, T. Matsunawa, and X. Zeng, "Machine learning and pattern matching in physical design," Proc. of the IEEE/ACM Asian and South Pacific Design Automation Conference (ASPDAC), pp. 19-22, 2015

[19] A Smart Design Paradigm for Smart Chips, Cliff Hou, VP R&D TSMC Hsinchu, Taiwan - ISSCC 2017 Keynote Speech

[20] Khan, M. Chalony, E. Ghillino, et al. "Effectiveness of machine learning in assessing impairments of photonics integrated circuits to reduce system margin", 2020 IEEE Photonics Conference.

[21] H. Stratigopoulos, Y.Makris, "Error Moderation in Low-Cost Machine Learning Based Analog/RF Testing", IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, Vol 27, No 2. 2008

[22] V. Huard, C. R. Parthasarathy, A. Bravaix, C. Guerin, and E. Pion, "CMOS device design-in reliability approach in advanced nodes" IEEE Int. Reliab. Phys. Symp. Proc., pp. 624–633, 2009.

[23] V. Huard et al., "A predictive bottom-up hierarchical approach to digital system reliability," IEEE Int. Reliab. Phys. Symp. Proc., pp. 4B.1.1-4B.1.10, 2012.

[24] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," IEEE J. Solid-State Circuits, vol. 25, no. 2, pp. 584–594, 1990.

[25] M. Altieri, S. Lesecq, E. Beigne, and O. Heron, "Towards on-line estimation of BTI/HCI-induced frequency degradation," IEEE Int. Reliab. Phys. Symp. Proc., p. CR6.1-CR6.6, 2017.

[26] M. A. SCARPATO, "Digital circuit performance estimation under PVT and aging effects", Micro and nanotechnologies/Microelectronics. Université Grenoble Alpes, 2017. English. NNT:2017GREAT093

[27] N. E. H. Weste and D. M. Harris, CMOS VLSI Design: A Circuits and Systems Perspective, vol. 53, no. 9. 2013.

[28] J. Watt, R. Borhani, and A. Katsaggelos, "Machine Learning Refined: Foundations, Algorithms, and Applications" (pp. I-Iv). Cambridge: Cambridge University Press, 2016.

[29] M. Kuhn and K. Johnson, "Applied Predictive Modeling" [Hardcover]. 2013.

[30] T. K. Ho, "Random Decision Forest," 1995