LSTA: Learning-Based Static Timing Analysis for High-Dimensional Correlated On-Chip Variations

Song Bian   Michihiro Shintani
Masayuki Hiromoto   Takashi Sato
Department of Communications and Computer Engineering, Kyoto University
Yoshida-hon-machi, Sakyo, Kyoto 606-8501, Japan
paper@easter.kuee.kyoto-u.ac.jp

ABSTRACT

As the transistor process technology continues to scale, the aging effect posits new challenges to the already complex static timing analysis (STA) process. In this paper, we first observe that aging can be thought of a type of correlated dynamic on-chip variations (OCV), and identify the problem introduced by such type of OCV. In particular, we take the negative bias temperature instability (NBTI) as an example dynamic OCV mechanism. We then propose a learning-based STA (LSTA) library to "predict" the timing of gates by capturing the correlation between our designed predictors. In the experiment, we used a linear regressor, support vector regression, and a non-linear method, random forest, to create the prediction model. An ISCAS'89 benchmark circuit is used as a training sample to for the algorithms to learn the aging model of gates, and the accuracies of the model is then tested on two processor-scale designs using the library are evaluated, achieving a maximum absolute error of 3.42%.

1. INTRODUCTION

The ever expanding technological frontier of the semiconductor manufacturing industry is yet to push the scaling of transistors to their limits. The complex physics on this level of scale raises a variety of device-level characteristic variations. These variations can be classified into two groups: static variation and dynamic variation (commonly known as the aging effect). Static variations are determined during or after the fabrication of a chip, while dynamic variations are generated over the lifespan of the chip. Correlations between the variations produce complex high-dimensional interactions that result in both optimism and pessimism, especially when the traditional STA approach is taken.

The high dimensional correlation problem persists across many different fields related to modern-day STA. As a possible solution, learning-based timing characterization is under active research. A study has utilized artificial neural network (ANN) and support vector machine (SVM) for path-based timing correction of non-signal-integrity (non-SI) mode to SI mode error compensation [11]. Others have been considering on predicting the timing failure of SRAM [7] and improving timing correlation between tools from different vendors [10]. However, the basic role that learning plays in these works is being an additional step after the application of STA, or multiple STAs, to improve the overall timing accuracy. The core part of STA is still traditional because it is assumed that a two-dimensional library with dimensions of load capacitance and slew time is sufficient for accurate STA.

Dynamic variations destroy the two-dimensional assumption. In particular, negative bias temperature instability (NBTI) and hot-carrier injection (HCI) are variations that gradually degrade a MOS device over a certain period of time, thus named aging effect [2, 8, 14]. Aging adds new challenges to the existing timing analysis flow, for it complexes the simple variation model assumed by the modern timing libraries by introducing correlated high-dimensional V_{th} variations. We demonstrate an example in Figure 1. While the details of the figure will be explained in Section 2, essentially, the figure is a SPICE simulation of the delays of an OR gate at different degradation levels involving two transistors, M1 and M2. It is observed that in the figure, the slope of the curve changes as M2 degrades, and this is caused by the fact that there is actually three pMOS transistors in an OR gate. The degradation on the third transistor, denoted as ΔV_{M3}, is a function of those of the other transistors, ΔV_{M1} and ΔV_{M2} under NBTI. With traditional *interpolation-based* STA libraries, a dense two-dimensional map is required to minimize the error introduced by M3, since it greatly degrades the linearity between the actually gate delay and the degradations on M1 and M2.

In searching for a more efficient but accurate solution to the high-dimensional correlation problem, we utilize *regression-based* machine learning algorithms. Machine learning algorithms can capture the underlying correlations between variables (predictor variables), and efficiently characterize a much more compact library compared to the traditional methodology. In addition, we want to note that the proposed learning-based static timing analysis, shorted as the LSTA flow, is obviously not limited to aging prediction. The main contribution of this paper is summarized as follows:

- **The identification of difficulties involving high-dimensional STA**: we point out that when NBTI is considered, it is required for the library to deal with high-dimensional correlated data. In addition, the proposed library construction is not limited to NBTI we

DAC '17, June 18-22, 2017, Austin, TX, USA
© 2017 ACM. ISBN 978-1-4503-4927-7/17/06...$15.00
DOI: http://dx.doi.org/10.1145/3061639.3062280

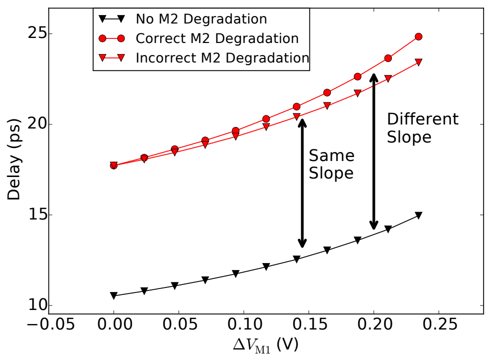

Figure 1: The $\Delta V_{\rm th}$-Δdelay relationship for the falling edge of A1 input to an OR gate simulated by SPICE.

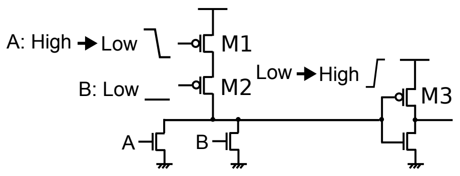

Figure 2: An OR gate demonstrates the stacking effect and the multi-stage correlation.

focus in this paper. Essentially, any identifiable non-Gaussian correlated OCV requires the library to be multi-dimensional.

- **A learning-based STA flow**: we propose a fundamentally novel approach to STA that utilizes learning (regression) instead of interpolation. Learning-based approach is known for its capability of capturing high-dimensional correlations efficiently. Hence, to solve the previously described challenge of high dimensionality, LSTA is an essential choice.
- **A thorough evaluation on different learning setups**: we experiment on different learning algorithms on setups for the LSTA library, and carefully examine the errors of timing predictions on both large and small designs. We achieved a maximum absolute error of 3.45% in larger designs, and around 4% across all tested designs.

The rest of this paper will be organized as follows. First, in Section 2, we explain the NBTI effect and point out the challenge it introduces to STA. Second, we introduce our proposed timing characterization flow in Section 3, and discuss the experimental result in Section 4. Finally, we conclude our paper in Section 5.

2. PRELIMINARIES

2.1 NBTI and Existing Research

Negative bias temperature instability has been studied extensively as it introduces non-Gaussian threshold voltage ($V_{\rm th}$) variations to a pMOS device over a long period of time (e.g., 10 years). Accurate STA on dynamic variations such as HCI and NBTI is essential for efficient mitigation techniques, and is currently under active research [4, 9, 12]. The fundamental approach for solving NBTI during an STA run is by adding extra dimensions to the STA library expressing the amount of degradation. For example, [9] adds n dimensions to the traditional 2D LUT, where n is the number of transistors in the particular gate. As explained in the next subsection, this approach faces two severe problems: i) a full characterization of a complex gate results in an exponential growth in the size of the library and ii) interpolation is inherently weak towards problems with high dimensionality. In [12], essentially the same approach has been taken where n dimensional dynamic variations are considered in a degradation equation. Apart from the dimensionality problem, [12] also does not provide standard SPICE results on the aged delays, leaving the accuracy of their method to be untested. To avoid the dimensionality problem, in [4], only a single dimension is added to the STA library, and signal probability is used to index the new dimension where NBTI-induced delay degradation is taken into account. This method achieves lower library characterization overhead and faster runtime, but suffers when multi-stage pMOS transistors are presented in the gate to be characterized.

2.2 High-Dimensional Correlation of NBTI

One important observation here is, for the n dimensions added to the STA library as in [9, 12], unnecessary points will be characterized, for that the variables are correlated. There are three types of correlations: i) delay correlation due to process-parameter correlations, which is static and thus not related to this work, ii) delay correlation, represented by the stacking effect, and iii) multi-stage correlation. In what follows, we will describe the latter two correlations, which have not yet been well handled, using specific examples.

For ii), a simple example of the MOSFET stacking effect related to NBTI degradation is illustrated in Figure 2. In the first stage of the OR gate, suppose that M1 makes a high to low transition. Although M2 stays low and thus is not involved in the transition process, the current going through both pMOS transistors is affected by the $V_{\rm th}$ of the lower transistor, as every standard MOSFET model describes [22]. If the $V_{\rm th}$ of that lower pMOS is degraded, the transition time increases as a function of the amount of threshold voltage degradation, $\Delta V_{\rm th}$. This creates a situation where the delay of a certain cell will be correlated to the amount of $V_{\rm th}$ degradation on each and every pMOS device in the cell. Figure 1, as shortly mentioned in the introduction, is the SPICE simulation of an OR gate that clearly indicates this trend. The curve for the $\Delta V_{\rm th}$-Δdelay relation is shifted upward significantly as M2 degrades.

The above mentioned delay correlation is the essential reason for requiring high-dimensional STA; however, the type-iii) correlation, or the multi-stage correlation appears to be more challenging, as it increases the dimensionality of STA, but in an implicit way. Two curves present in Figure 1, marked as correct and incorrect when transistor M2 is fully degraded. The incorrect curve ignores the effect of the third transistor, M3, in the second stage of the OR gate, as shown in Figure 2. At this point, it seems that it is necessary for the STA library to have three degradation dimensions to accurately characterize the $\Delta V_{\rm th}$-Δdelay relation. This is not entirely true, for that in fact, under NBTI and HCI, the amount of degradation on M3 is perfectly correlated, in a way or another, to the joint probability distribution of M1 and M2. For example, under NBTI degradation, all the three pMOS in Figure 2 cannot suffer from full degradation concurrently, due to the fact that if both M1 and M2 are turned on, M3 is forced to be shut down. In order for an interpolation-based STA library to accurately capture this complex correlation, it has to create the wasteful three-dimensional space to capture the correlation, just as existing works [9, 12] do.

3. LSTA: LEARNING-BASED STATIC TIMING ANALYSIS

To solve the problem introduced by NBTI described in the previous section, we propose a regression-based library construct to solve the high dimensionality problem.

3.1 Problem Formulation

We first describe a high-level formulation of gate delay under dynamic V_{th} variations (i.e., aging). Here we take NBTI as an example. For any given gate, we consider each input pin to the gate to be an independent random variable $X : \{0, 1\} \to \{0, 1\}$, that is, it can take either a logic value of 0 or 1. We can then define the logic 0 probability to be $p_{x_i} = \Pr[X_i = 0]$ for each pMOS attached to the i-th input pin X_i. Next, it is easy to see that any gate node Y_i of a pMOS that is not directly attached to an input pin is a conditioned random variable on the input random variables, namely, $p_{y_i} = \Pr[Y_i = 0|\{X_i\}]$. Following this, we assume an abstract function $v : [0, 1] \to \mathbb{R}$ that maps the signal probability (can be logic 0 probability, up to the definition) p on a pMOS transistor to its ΔV_{th}, the V_{th} degradation. Finally, d, the delay of a gate under dynamic variations, can be abstractly modeled as the following equation.

$$d = d_0 + g(v(p_0), \cdots, v(p_{m-1}), \cdots, v(p_{n-1})), \qquad (1)$$

where d_0 is the delay without degradation, m is the number of pMOS attached directly to m distinct inputs, n is the total number of transistors in the gate, and $g : \mathbb{R}^n \to \mathbb{R}$ maps a set of ΔV_{th} values to a Δdelay value. Equation (1) points out our dilemma: correlations present in the probability space that makes the maximum dimension of the gate m; nevertheless, the ΔV_{th}-Δdelay relationship is n-dimensional.

To cope with the dimensionality dilemma, we can create the linear-step grid in V_{th} space, and find an inverse function for v to calculate the probability correlation. Concretely, let $v^{-1} : \mathbb{R} \to [0, 1]$, for each V_{th} step, we can calculate

$$\begin{aligned} p_m &= corr_m(v^{-1}(\Delta V_{\mathrm{th}_0}), \cdots, v^{-1}(\Delta V_{\mathrm{th}_{m-1}})) \\ &\vdots \\ p_{n-1} &= corr_{n-1}(v^{-1}(\Delta V_{\mathrm{th}_0}), \cdots, v^{-1}(\Delta V_{\mathrm{th}_{m-1}})) \end{aligned} \qquad (2)$$

where $corr_i$ is the i-th correlation function that maps the independent variables $(p_0, \cdots, p_{m-1})$ to the dependent variables $(p_m, \cdots, p_{n-1})$. After calculating the correlations, we can use v to map the dependent variables back to V_{th} space, and use Equation (1) to calculate the actual delay at this ΔV_{th} step. Namely,

$$d = d_0 + g(\Delta V_{\mathrm{th}_0}, \cdots, \Delta V_{\mathrm{th}_{m-1}}, v(p_m), \cdots, v(p_{n-1})) \qquad (3)$$

It is also noted that, once the distribution for all the independent variables are determined, this procedure becomes strictly a static timing analysis problem, instead of statistical STA.

While the procedure described above may seem viable, to create an STA library using this method, we face two problems: the $corr_i$ function may not be well-defined, and v^{-1} can simply be non-existent. To make things less abstract, take NBTI and the OR gate we have been using as an example again. The relationship between ΔV_{th} and logic 0 probability p can be formulated as $\Delta V_{\mathrm{th}} = v(p) \approx \left(\frac{0.001n^2K_v^2pCt}{0.81t_{\mathrm{ox}}^2(1-p)}\right)^n$ according to [3, 13]. The v^{-1} here can be written as $p = v^{-1}(\Delta V_{\mathrm{th}}) = \frac{0.81t_{\mathrm{ox}}^2 + \log_n \Delta V_{\mathrm{th}} - 0.001n^2K_v^2pCt}{\log_n \Delta V_{\mathrm{th}} - 0.001n^2K_v^2pCt}$ The correlation between p_3, the logic 0 probability on $M3$ and p_1, p_2 is given by the simple equation $p_3 = 1 - p_1p_2$. We then can use the previously described v^{-1} to calculate p_3, and the final ΔV_{th_3} is then given by

$$\Delta V_{\mathrm{th}_3} = v\left(1 - v^{-1}(\Delta V_{\mathrm{th}_1}) \cdot v^{-1}(\Delta V_{\mathrm{th}_2})\right), \qquad (4)$$

which is non-linear and thus time-consuming to solve (the p-ΔV_{th} relation is an approximated equality in the first place) on a per-gate scale.

As the number of degradation mechanisms increases, even a simple cell like half-adder becomes quite impossible to model. It is absolutely tedious to manually adjust each v and v^{-1} functions on a per-gate level to obtain a good accuracy in calculating the steps, and this becomes the main motivation for us in proposing an alternative learning approach to the timing under such complex variation with high-dimensional correlations, which will be explained in the next section.

3.2 Learning the Correlations

In this section, we first describe the overall timing flow, and then present the regression construct. We use the term "fresh" to refer to the state where the gate/design without aging-induced dynamic variations, and "aged" for the gates that are degraded under the aging effect.

3.2.1 Library Characterization and Application

We first present an overall flow for the library characterization, delaying the discussion on regression setup, parameter selection and algorithm adoption in later sections. Figure 3 illustrates the library characterization step. First of all, a traditional 2D LUT, shown in the bottom-left of the figure, will be characterized as usual. Meanwhile, a set of training designs will be prepared and the *fresh* and *aged* gate delays in these designs are characterized, as indicated on the right-hand side of the figure. The fresh delay works as one of the predictor variables, and the aged delay is the predicted variable. The tool used in the characterization step should be the correct timing value, either derived from complex mathematical expression or careful real-chip measurement. Finally, the aged and fresh gate delays, along with other parameters, are setup to be the training samples for the learning algorithm, from which the aging models will be created. These learned models will be integrated into the STA library.

Figure 4 summarizes the library application process. The application of the proposed library is virtually the same as the traditional one, except a set of parameters required by the learning algorithm, as described in the next section.

3.2.2 Regression Setup

In general, a regression analysis can be seen as a series of function optimization in a certain function class $f \in \mathcal{F}$ to minimize some error function $\mathcal{E}$ over a set of data pairs (x_i, y_i). The problem can be formulated into the following equation.

$$\min_{f \in \mathcal{F}} \mathcal{E}(y_i, f(x_i)) \qquad (5)$$

While different machine learning algorithms have different

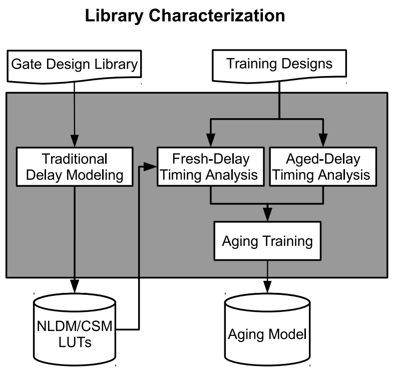

Figure 3: Overview of the proposed library characterization process with learning-based training step.

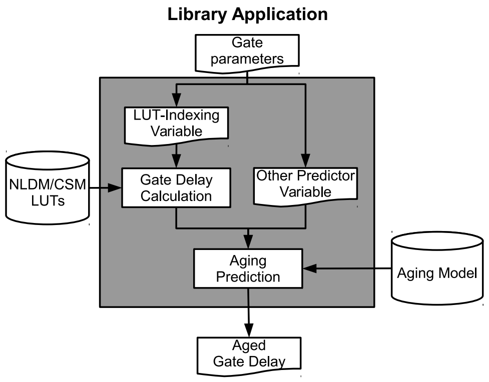

Figure 4: Overview of the proposed library being applied in a timing step.

approaches to $\mathcal{F}$ and $\mathcal{E}$, it is noted that the exact function we want to learn actually has already been abstractly sketched in Section 3.1 as Equations (2) and (3). Thus, assuming the v^{-1} and $corr_i$ functions can be analytically defined, Equation (5) can be perfectly solved. Nonetheless, in reality, this is not the case, and we seek an approximation to the equation.

3.3 Parameter Selection

Parameter selection is the core of every application of machine learning algorithms. In particular, the problem we face in a learning-based STA library is unique, in the sense that the difficulty lies in using what we have in hand to obtain what we want to know. For example, although a thoroughly characterized CSM waveform is almost certainly better as a predictor variable than a single delay value from an NLDM library, this piece of information can be costly to obtain in the library characterization step and thus unavailable to the learning algorithm.

In this study, we target on an NLDM implementation, and selected the parameters accordingly. An NLDM implementation feeds the learning algorithm with the least amount of information and the most amount of non-linear errors. Thus, we consider this implementation to be a worst-case scenario. The aging-prediction formulation is demonstrated in Equation (6).

$$d = f(d_0, \Delta V_{\mathrm{th}_0}, \ldots, \Delta V_{\mathrm{th}_{m-1}}, t_{\mathrm{slew}}, C_{\mathrm{load}}), \qquad (6)$$

where t_{slew} and C_{load} are the traditional timing slew and capacitive load used to index the NLDM libraries. Equation (6) shares the same notation as described in Equation (1), for that virtually they are the same function. The difference here is that, instead of explicitly writing out the correlations functions in Equations (2) and the inverse function v^{-1} as in Equation (3), only m ΔV_{th} values are used in Equation (6). In essence, what we are doing here is to use Equation (5) to *learn* the best f that embeds the correlation functions of the gate and the inverse ΔV_{th} function v^{-1}, instead of manually adjusting the parameters and functions.

On a separate note, to train the prediction model, a set of training samples each containing the corresponding values for the variables in Equation (6) is required in the training process. In this work, we only extract these values from a fixed experimental design. An efficient way generating high-quality training design remains as an interesting open problem.

3.4 Learning Algorithm Examination

When considering a learning problem, we have truly a plethora of algorithms at hand. In this work, we focus on two fundamental types of algorithms: linear regressors and decision tree regressors. While linear regressors are widely used, their robustness relies heavily on the distribution of the data. We use the support vector regression (SVR) with a radial basis function (RBF) kernel to examine the capability a linear regressor. On a different note, decision trees (DTs) are known for their efficiency and capability of capturing non-linear data [15]. However, a single decision tree generally suffers from overfitting, since it tries to capture all sorts of non-linearities that reside in the data [15]. Hence, we apply two typical methods to improve the performance of DT: namely, bagging [5] (with random forest (RF)) and boosting [16] (with AdaBoost).

4. NUMERICAL EXPERIMENT

4.1 Experiment Setup

Numerical experiments are conducted on a set of designs that includes the ISCAS'89 circuit bench [6] and two processors. One is a five-stage pipelined processor from a commercial IP library [21] (named "Shino"), and the other is a modified version of the five-stage pipelined processor [1] that implements the full MIPS32 instruction set (named "Kotori"). In the experiment, we resynthesized the ISCAS'89 benchmark circuit using cells: NAND, NOR, AND, OR, INV, and DFF. Nangate 45 nm Open Cell Library [17] is used for circuit implementation. The designs are synthesized using a logic-synthesis tool [18]. To simplify the evaluation process (also let SPICE to obtain a realistic run time), we selected 25,446 worst-case path candidates from Shino and 24,978 from Kotori using a commercial STA tool [20]. In this experiment, we fixed the year of aging and temperature at 10 years and 400 K for NBTI. These parameters can also be used as a predictor variable to achieve a more complex analysis with virtually the same construct. As mentioned in Section 3, we used [19] as the golden standard for cell timings. The experiment was conducted on a Intel Xeon E5-2630 processor using a single thread.

The experiment flow is summarized in Figure 5. We first use the s38584 circuit in ISCAS'89 bench to train the learning algorithm, and then apply the model to larger designs (processors). The training design contains 18,119 training samples. A sample here is a gate delay associated with a particular set of predictor parameters. The training design of

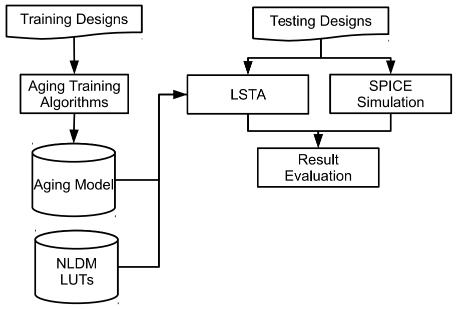

Figure 5: The evaluation flow of path delays utilizing the proposed timing framework.

Table 1: Result of proposed technique (all errors are compared to SPICE simulation)

Design	# Samples	MAE (%)	NRMSE (%)	Runtime (s) RF	Runtime (s) SVR
Shino	5281	3.34	0.93	0.12	0.42
Kotori	5813	2.86	0.24	0.13	0.46
s27	25	2.23	1.3	0.10	0.001
s1494	378	1.58	2.1	0.10	0.03
s5378	1383	4.06	1.3	0.11	0.11
s13207	2262	3.53	1.3	0.11	0.18

s38584 has 11,052 paths and a longest path delay of 2.32 ns, a considerably smaller design when compared to a processor with more than 20k paths. The training and prediction process is conducted for each gate in the designs, whereas the error evaluation is performed on the paths. We report two types of error, the maximum absolute error (MAE) and the normalized root-mean square error (NRMSE) with respect to the path delay calculated by SPICE. The NRMSE formula is given as $NRMSE = \frac{\sqrt{\sum_{i=1}^{n}(D_{\mathrm{p}}-D_{\mathrm{t}})^2}}{\sqrt{n}(D_{\mathrm{t}_{\max}}-D_{\mathrm{t}_{\min}})}$, where D_{p} is the predicted path delay, D_{t} is the correct path delay measured by SPICE, n is the number of paths, and $D_{\mathrm{t}_{\max}}$ and $D_{\mathrm{t}_{\min}}$ are the maximum and minimum path delays measured by SPICE, respectively. In what follows, we will first present the overall result in Section 4.2, and perform an algorithm comparison in Section 4.3.

4.2 Experiment Result: Overall

Figures 6, 7, and 8 compare the accuracy of delay calculation under a specific training-testing setting. The test design here is Shino. The fresh-time versus aged-time delay relationship is plotted in Figure 6. Using an aging-aware 3D LUT as proposed in [4] without the manual adjustment required by the paper, the result is demonstrated in Figure 7 with a significant amount of underestimation, as expected. Finally, the aged delay calculated using the proposed learning algorithm with respect to SPICE is shown in Figure 8, with an MAE of 3.4%.

Table 1 summarizes the number of samples, MAE, NRMSE, and runtime of different test designs. All results are obtained using s38584 in the ISCAS'89 circuit bench to train a RF with 200 decision trees (each tree has a maximum depth of 20). These hyper-parameters are selected empirically on the training dataset, as no significant accuracy improvement is observed with further increases in the parameter values. RF is used to obtain the results because it is discovered that SVR is not suitable for this type of analysis from the runtime perspective (A more detailed discussion will be carried out in Section 4.3).

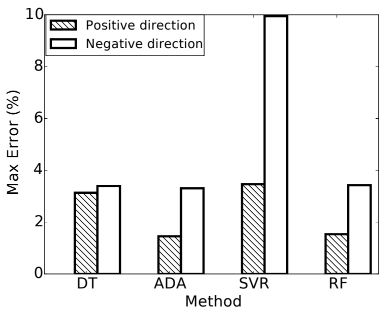

Figure 9: Performance of each algorithm on Shino with dashed bar indicating the maximum error, and blank bar the absolute value of the minimum error.

Table 2: Comparison between the MAEs for tree-based algorithms and linear algorithms

		W/ Data Norm.	W/o Data Norm.
RF	Accuracy	3.42%	3.42%
	Train Time	2.29 s	2.30 s
	Test time	0.12 s	0.12 s
SVR	Accuracy	4.23%	9.94%
	Train Time	11.73 s	18.45 s
	Test time	0.42 s	0.90 s

4.3 Experiment Result: Algorithm Comparisons

In this subsection, we compare the speed of RF and SVR in both training and testing. Here, train time is the time required to train the regressor using the s38584 design, and test time is the time to apply the learned model to the gates in the Shino design. From Table 1, it suggests that the runtime for applying RF regressor is independent of the size of the test set. This can be explained by the fact that 200 trees have to be evaluated regardless of the test size. Meanwhile, due to the fact that the evaluation of each tree is logarithmic in the number of tree nodes (linear in tree height), and that the maximum depth for each tree is fixed, we see almost a constant runtime for the RF regressor. On the other hand, SVR sees a significant performance degradation as the number of test samples increases. Second, as outlined in Table 2, data normalization simplifies the problem SVR needs to solve, and results in a great performance enhancement runtime for both training and testing, as well as an improvement in accuracy. In contrast, it seems that data normalization has virtually no effect on the RF regressor.

Figure 9 shows the maximum error in the positive (overestimation) and negative (underestimation) directions of different learning algorithms. Comparing with a single decision tree, RF and AdaBoost are more stable across designs. In this study, we found RF (bagging) and AdaBoost (boosting) performs equally well. Thus, we conclude here that tree-based regressors are more suitable for our problem for their stability both in terms of training/testing speed and prediction accuracy. As a final note, these errors can be offset to be positive only, if required.

5. CONCLUSION

In this paper, we proposed a learning-based method for predicting the NBTI-induced delay degradation in large de-

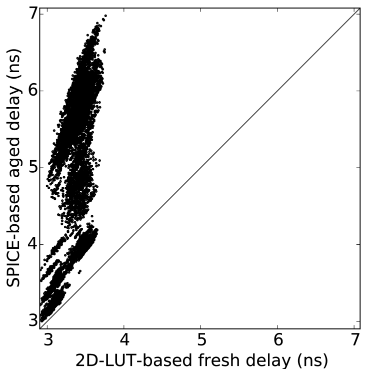

Figure 6: Delay comparison between fresh-time LUT and aged SPICE for Shino.

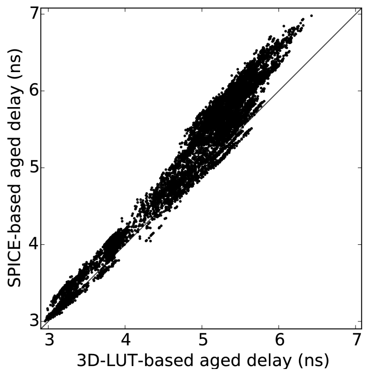

Figure 7: Delay comparison between aged LUT used in [4] and aged SPICE for Shino.

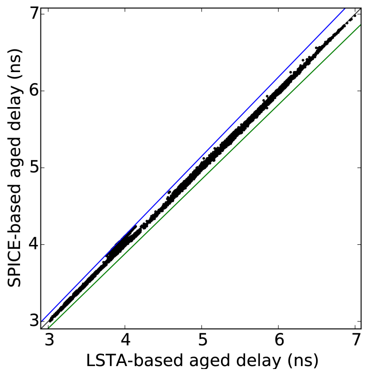

Figure 8: Delay comparison between the learned aged delay based on a traditional NLDM LUT and aged SPICE for Shino. The blue and green lines indicate plus and minus 3% error bar, respectively.

signs like processors. The method utilizes smaller designs, and "learns" the degradations on gates in these designs. In the experiment, the proposed technique is applied to two processor designs with various algorithm construct to verify its effectiveness. We achieved a maximum absolute error of 3.45% in the processors, and around 4% across all designs.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant No. 26280014, 17H01713, 15K15960, and Grant-in-aid for JSPS Fellow (DC1). The authors also acknowledge support from VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

6. REFERENCES

[1] OpenCores. http://www.opencores.org.

[2] M. A. Alam and S. Mahapatra. A comprehensive model of PMOS NBTI degradation. *Microelectron. Reliab.*, 45(1):71–81, 2005.

[3] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula. Predictive modeling of the NBTI effect for reliable design. In *Proc. CICC*, pages 189–192, 2006.

[4] S. Bian, M. Shintani, S. Morita, M. Hiromoto, and T. Sato. Nonlinear delay-table approach for full-chip NBTI degradation prediction. In *Proc. ISQED*, pages 307–312, 2016.

[5] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[6] F. Brglez, D. Bryan, and K. Koiminski. Combinational profiles of sequential benchmark circuits. In *Proc. ISCAS*, pages 1929–1934, 1989.

[7] W. T. J. Chan, K. Y. Chung, A. B. Kahng, et al. Learning-based prediction of embedded memory timing failures during initial floorplan design. In *Proc. ASPDAC*, pages 178–185, 2016.

[8] P. Chaparala, J. Shibley, and P. Lim. Threshold voltage drift in PMOSFETs due to NBTI and HCI. In *Proc. IIRW*, pages 95–97, 2000.

[9] F. Firouzi, S. Kiamehr, M. Tahoori, and S. Nassif. Incorporating the impacts of workload-dependent runtime variations into timing analysis. In *Proc. DATE*, pages 1022–1025, 2013.

[10] S.-S. Han, A. B. Kahng, S. Nath, and A. S. Vydyanathan. A deep learning methodology to proliferate golden signoff timing. In *Proc. DATE*, pages 1–6, 2014.

[11] A. B. Kahng, M. Luo, and S. Nath. SI for free: machine learning of interconnect coupling delay and transition effects. In *Proc. SLIP*, pages 1–8, 2015.

[12] S. Karapetyan and U. Schlichtmann. Integrating aging aware timing analysis into a commercial STA tool. In *Proc. VLSI-DAT*, pages 1–4, 2015.

[13] H. Konoura, Y. Mitsuyama, M. Hashimoto, and T. Onoye. Comparative study on delay degrading estimation due to NBTI with circuit/instance/transistor-level stress probability consideration. In *Proc. ISQED*, pages 646–651, 2010.

[14] S. Mahapatra, P. Bharath Kumar, and M. Alam. A new observation of enhanced bias temperature instability in thin gate oxide p-MOSFETs. In *IEDM Tech. Digest*, pages 337–340, 2003.

[15] M. Pal and P. M. Mather. An assessment of the effectiveness of decision tree methods for land cover classification. *Remote sensing of environment*, 86(4):554–565, 2003.

[16] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of statistics*, pages 1651–1686, 1998.

[17] Si2.org. Nangate 45nm open cell library. http://www.si2.org.

[18] Synopsys, Inc. *Design Compiler I-2013.06.*

[19] Synopsys, Inc. *HSPICE I-2013.12.*

[20] Synopsys, Inc. *PrimeTime Fundamental H-2013.06.*

[21] Synopsys, Inc. *Processor Designer G-2012.09.*

[22] University of California, Berkeley. Bsim4v4.7, University of California, 2011. http://www-device.eecs.berkeley.edu/bsim/Files/BSIM4/BSIM470.