# Modeling and Predicting Transistor Aging Under Workload Dependency Using Machine Learning

Paul R. Genssler, *Member, IEEE*, Hamza E. Barkam, *Member, IEEE*, Karthik Pandaram, Mohsen Imani, *Member, IEEE*, and Hussam Amrouch, *Member, IEEE*

***Abstract*— The pivotal issue of reliability is one of the major concerns for circuit designers. The driving force is transistor aging, dependent on operating voltage and workload. At the design time, it is difficult to estimate close-to-the-edge guardbands that keep aging effects during the lifetime at bay. This is because the foundry does not share its calibrated physics-based models, comprised of highly confidential technology and material parameters. However, the unmonitored yet necessary overestimation of degradation amounts to a performance decline, which could be preventable. Furthermore, these physics-based models are computationally complex. The costs of modeling millions of individual transistors at design time can be exorbitant. We propose the use of a machine learning model trained to replicate the physics-based model, such that no confidential parameters are disclosed. This effectual workaround is fully accessible to circuit designers for the purposes of design optimization. We demonstrate the model's ability to generalize by training on data from one circuit and applying it successfully to a benchmark circuit. The mean relative error is as low as 1.7 %, with a speedup of up to 20×. Circuit designers, for the first time ever, will have ease of access to a high-precision aging model, which is paramount for efficient designs. In contrast to existing work, our approach takes the full switching activity into account to model recovery effects. This work is a promising step in the direction of bridging the gap between the foundry and circuit designers.**

Manuscript received 28 August 2022; revised 22 December 2022 and 17 June 2023; accepted 17 June 2023. Date of publication 10 July 2023; date of current version 30 August 2023. This work was supported in part by the German Research Foundation (DFG) "ACCROSS: Approximate Computing aCROss the System Stack" under Grant 428566201 and Grant AM 534/3-1; in part by the Advantest as part of the Graduate School "Intelligent Methods for Test and Reliability" (GS-IMTR), University of Stuttgart; in part by the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award; in part by the National Science Foundation under Grant 2127780 and Grant 2312517; in part by the Semiconductor Research Corporation (SRC); in part by the Office of Naval Research under Grant N00014-21-1-2225 and Grant N00014-22-1-2067; in part by the Air Force Office of Scientific Research under Award FA9550-22-1-0253; and in part by Xilinx, Advanced Micro Devices (AMD), and Cisco. This article was recommended by Associate Editor P. A. Beerel. *(Corresponding author: Paul R. Genssler.)*

Paul R. Genssler and Karthik Pandaram are with the Chair of Semiconductor Test and Reliability (STAR), University of Stuttgart, 70569 Stuttgart, Germany (e-mail: genssler@iti.uni-stuttgart.de; pandarkk@iti.uni-stuttgart.de).

Hamza E. Barkam and Mohsen Imani are with the Bio-Inspired Architecture and Systems (BIASLab), UC Irvine, Irvine, CA 92697 USA (e-mail: herrahmo@uci.edu; m.imani@uci.edu).

Hussam Amrouch is with the Chair of Semiconductor Test and Reliability (STAR), University of Stuttgart, 70569 Stuttgart, Germany, also with the Chair of AI Processor Design, Technical University of Munich (TUM), 80333 Munich, Germany, and also with the Munich Institute of Robotics and Machine Intelligence (MIRMI), 80333 Munich, Germany (e-mail: amrouch@tum.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2023.3289325.

Digital Object Identifier 10.1109/TCSI.2023.3289325

***Index Terms*— Circuit reliability, transistor aging, degradation, machine learning.**

## I. Introduction

RELIABILITY is a major concern in today's circuits. As CMOS scaling reaches the atomic level, the impact of degradation effects on the reliability becomes stronger [1]. Aging is the most dominating effect and changes the transistor's properties like the threshold voltage $V_{th}$. Consequently, it can cause permanent failures in a circuit. Even before such failures, aging indirectly impacts the circuit's timing and hinders performance improvements. The negative bias temperature instability (NBTI) aging mechanism is responsible for the highest degradation [2]. During regular transistor operation, Si-H bonds at the Si-$SiO_2$ interface might be broken and annealed. Additionally, charges are captured and emitted in the oxide vacancies at the interface layer. Over time, these defects accumulate and manifest themselves as a shift in $V_{th}$, referred to as $\Delta V_{th}$. The induced increase in the propagation delay of the logic gates can cause timing violations.

To prevent such timing violations and ensure the circuit performs as specified during its entire projected lifetime, timing guardbands are added during the design phase. Such additional slack compensates for the reduced switching speed of aged transistors. The design challenge is to balance such guardbands between too pessimistic, reducing the circuit's performance, and too optimistic, increasing the risk of premature failures. To find an optimal guardband (i.e., small, yet sufficient), the aging-induced $\Delta V_{th}$ has to be accurately estimated. Aging models are required to abstract the underlying physical behaviors, take technology parameters, stress patterns, and voltages into account, and predict the evolution of $\Delta V_{th}$ over time. Only with such models can designers make informed and proper decisions on the guardband of every transistor.

Physics-based aging models capture the *dynamics* of the fundamental physical behavior and chemical reactions inside the transistors. Complex differential equations take the material and technology dependent parameters into account. This makes the model capable of capturing recovery effects, where $V_{th}$ is indeed reduced as shown in Fig. 1. During low-stress phases, the defects are partially healed and $V_{th}$ recovers [3]. The supply voltage $V_{DD}$ is dynamic, creating such phases, changes over time, and is typically defined through the workload of the circuit. To capture these voltage dynamics, an aging model has to process such a voltage waveform.

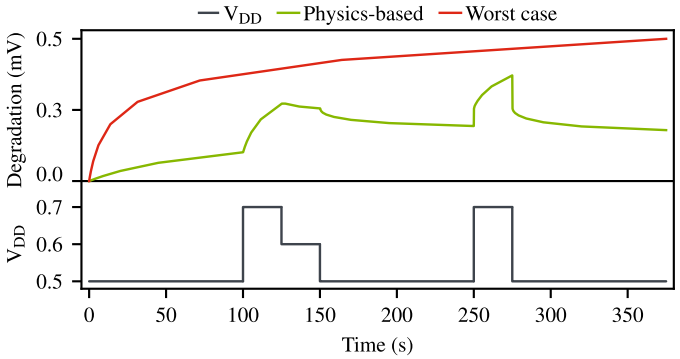


Fig. 1. Worst-case models are typically employed in the industry. For transistor aging, they assume constant stress and thus the highest possible degradation (red). Physics-based models are far more accurate because they take the input waveform and recovery effects into account.

Worst-case aging models are not capable of this. They are created by fitting measurements of constant voltage stress on a transistor. Hence, they cannot model the physics of voltage dynamics and recovery effects. To process a voltage waveform, the highest voltage is applied for the whole duration of the voltage waveform. Consequently, they overestimate the impact of aging significantly. Today's high-end devices are operating at the technological limits and cannot afford the unnecessary performance penalties mandated by such pessimistic predictions, an ideal aging model has to be as precise as possible. While physics-based models achieve such high accuracy, they require parameters specific to the manufacturing process to compute the degradation. Such parameters are a valuable secret of the foundry because they reveal details about their technology through material-dependent parameters. The foundry instead provides a process design kit (PDK) covering various corner cases including the worst case (i.e., the slow-slow corner). In summary, designers have limited options to optimize their circuit, which reduces performance and increases costs. An "ideal" aging model should therefore not expose any confidential information about the underlying technology. At the same time, it should still provide accurate estimations, including recovery.

The foundry only guarantees the slow-slow corner leading to very pessimistic guardbands and hence efficiency losses. With the risk of failure on the designers' side, this pessimism might be reduced. Alternatively, the degradation can be measured during post-silicon validation. However, such test chips are costly and increase time to market. Additionally, the design is almost complete at this stage further increasing costs, which often cannot be afforded by small to medium-sized companies. With an aging model, the impact of the circuit's workloads and voltages on $V_{th}$ can be predicted early in the design phase. For application-specific circuits like a video decoder, the workloads are known from the specifications and can be simulated even before the design phase starts. Starting with the much faster typical-typical corner, an appropriate guardband is added. An ideal aging model is thus available to the designers during design time and allows them to predict the degradation for each individual transistor. During runtime, the remaining guardband can be treated as a resource like remaining battery power. Resource management schemes require a long-term aging model to optimize over the whole lifetime. Physics-based models are not an option, because of their confidential parameters and their high computational complexity. An ideal aging model has a low computational cost to be employed for millions of transistors during design time. At runtime, it provides predictions as a low-overhead background task in the operating system.

### A. Our Main Contributions

Designers require an *accurate and fast* transistor aging model to optimize the performance of their circuit designs depending on the potential workload. Further, simulating millions and billions of transistors is time consuming necessitating a fast aging model. Physics-based models are slow and confidential, i.e., not accessible to designers. Therefore, we propose to employ machine learning (ML) to model transistor aging. As shown in Fig. 2, the foundry employs its confidential physics-based models to train an ML-based model. Such a model is fast and does not reveal the technology and material parameters. Hence, it can be provided to the circuit designers. They employ the model in conjunction with their workloads to generate their workload-specific, aging-aware PDK. With this PDK, guardbands can be reduced increasing performance.

In this paper, we investigate for the first time how physics-based models can be abstracted through ML methods. In contrast to existing work, our approach takes the full switching activity into account to model recovery effects. ML algorithms like deep neural network (DNN) or long short-term memory (LSTM) have a high computational complexity but can achieve in high accuracy in many applications. As a less computational-intense algorithm, lightweight brain-inspired ML methods have attracted the interest of the community in recent years. Brain-inspired hyperdimensional computing (HDC) does not utilize networks of neurons but is built around large randomly-generated hypervectors [4]. The accurate yet complex equations of physics-based models have to be replaced by a trained ML model. To this end, we investigate two challenges. First, the capability to constructed a $\Delta V_{th}$ trace from a voltage activity waveform. Such traces and waveforms are typically in the range of nanoseconds to minutes and model short-term aging [5]. Second, predict only the last degradation $\Delta V_{th}$ value for a single transistor based on a given short voltage activity waveform. This prediction is essential for an extrapolation to ten years until the end of lifetime (EoL) of the device. We investigate the accuracy of the ML models not only on their prediction of this $\Delta V_{th}$ value. We also employ the predicted $\Delta V_{th}$ further to extrapolate the circuit delay after ten years and compare the impact on the delay. The performance of the models is evaluated by training on the transistors of standard cells and an 8-bit adder. The test set are the transistors of a 32-bit MAC unit with which we also evaluate the prediction of the delay after 10 years.

## II. Related Work and Background

Transistor aging has been studied for many years and the impact is well understood. This sections aims at summarizing this research briefly.

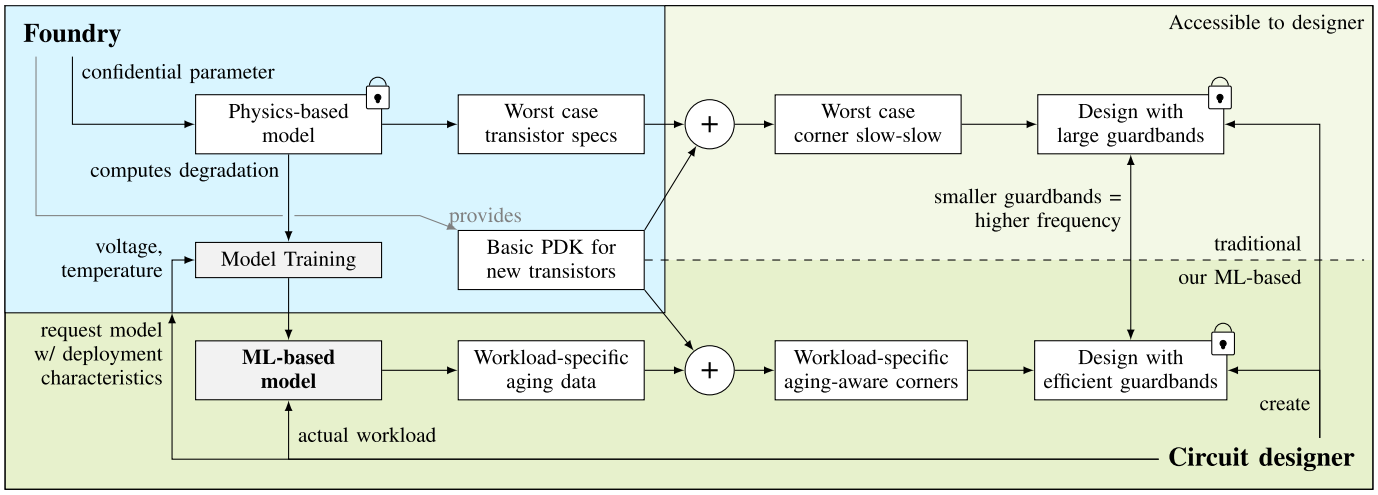


Fig. 2. Typically, circuit designers do not have access to accurate physics-based aging models to estimate efficient (i.e., small, yet sufficient) guardbands. A machine learning-based aging model are free sensitive material and process parameters of the foundry and can thus be shared with designers. Now, circuit designer can create workload-specific aging data for efficient guardbands.

### A. Transistor Aging Models

Since manufacturing technology has moved past 45 nm, new materials had to be used [6]. Hafnium Oxide ($HfO_2$) is used as a high-$\kappa$ dielectric and replaced the traditional silicon dioxide. A drawback of $HfO_2$ is its higher number of pre-existing defects in the material itself, making it more susceptible to degradation and thus less reliable. Hence, transistor aging has become a major consideration in modern circuits.

In this work, we focus on NBTI as the primary aging mechanism [2]. Note that our method can be applied analogously to other aging mechanisms like hot carrier degradation (HCD). NBTI aging occurs when the pMOS transistor is turned on. During the on-time, two effects come into play. First, positively charged holes are trapped inside the $HfO_2$ dielectric. This increases the $V_{th}$ of the transistor. If the stress is reduced, i.e., the voltage lowered or the transistor completely turned off, then the holes can be removed and the initial $V_{th}$ can be recovered over time. Due to the second effect, new traps are generated in the interface material. If the transistor is turned on, these traps are positively charged increasing the $V_{th}$. Similar to the first effect, some of these traps may be deactivated once the stress is reduced or removed partially restoring $V_{th}$. In both cases $\Delta V_{th}$ is dictated by the applied voltage.

Most models (especially analytical models) consider recovery only at 0 V. However, measurements have proven that even a reduction in the voltage starts the recovery [3]. The phenomenon is demonstrated in Fig. 1, in which a physics-based NBTI model is employed to calculate the transient trap occupancy, among others [2]. Hence, it is indispensable to consider the dynamics of different voltage levels when modeling aging [7]. In this work, the physics-based NBTI aging model "BAT" is employed, which has been calibrated with measurements from various technologies [2].

ML-based methods to model and predict the impact of aging have been investigated at different levels of the stack. At the system level, reinforcement learning-based methods have been used to schedule threads on a multi-core CPU to reduce aging [8]. At the circuit level, the increase in path delay due to an increased $\Delta V_{th}$ has been modeled with multivariate adaptive regression splines and compared against support vector machine (SVM) and recurrent neural network (RNN) [9]. Their model takes changing operation conditions, like different voltages, into account. At the gate level, the generation of reliability-aware cell libraries through ML has been demonstrated [10]. This allows a circuit designer to quickly generate cell libraries specifically for their workloads and optimize their design accordingly.

The authors in [11] propose a method to predict aging-induced delays for 28 nm FDSOI technology. Their approach considers different voltage and frequency settings as well as workloads, although abstracted as percental switching activities. They train an ML model for two standard cells and use a conversion scheme to extend the prediction of other standard cells. In contrast to this work, their model only considers an abstracted switching activity workload and thus cannot include recovery effects. Further, the focus on standard cells limits its applicability in custom circuits.

In [12], at device level, a single transistor is subjected to constant voltage stress and the $V_{th}$ curve is fitted with a regression model. Such voltage dynamics are taken into account in [13]. They performed TCAD simulations of a single transistor until time $t$ and afterwards continue with an ML model. Since TCAD simulations are still required initially, their approach cannot relieve the confidentiality concerns of the foundry. The physics-based aging model in TCAD includes sensitive technology parameter and reveals details about the manufacturing process, an essential trade secret of any foundry. In contrast to [12], we include voltage dynamics and recovery effects. In contrast to [13], our models do not rely on physics-based models during inference. Hence, the input to our model is not a single fixed voltage or a statistical assumption of on/off times, but a trace representing workloads and operating conditions for an individual transistor.

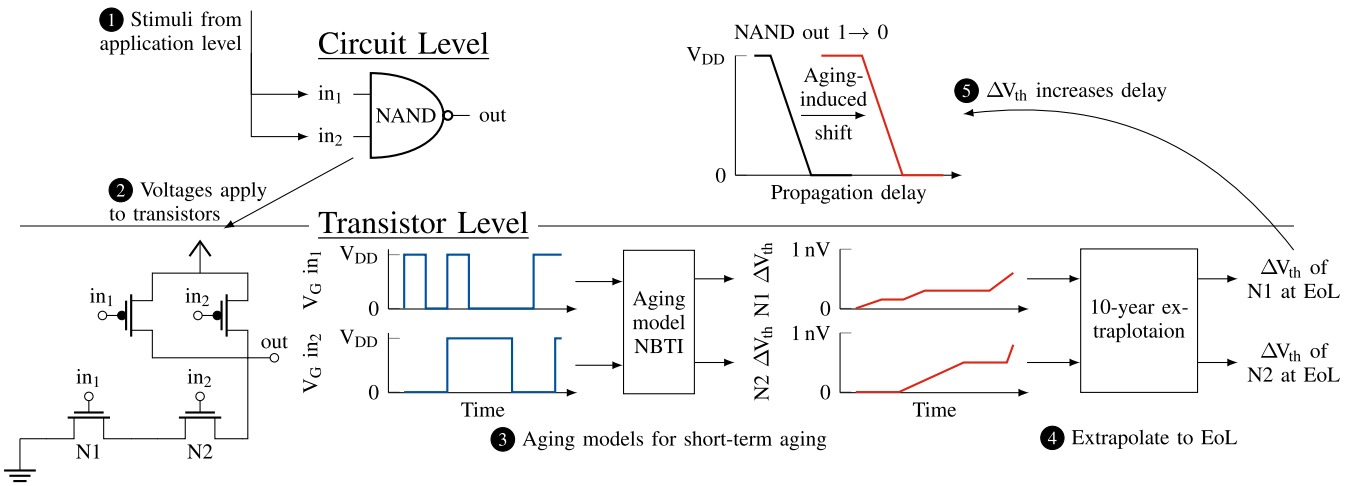


Fig. 3. In the experimental setup, stimuli are applied at circuit level ❶ and voltage waveforms for each transistor extracted ❷. Those are passed to the aging models ❸ to generate the ground truth for the training of the machine learning models. Then, their prediction is extrapolated to the end of lifetime (EoL) ❹. Finally, the degradation is applied again at circuit level for efficient guardband estimation ❺.

### *B. Machine-Learning Methods*

As for our predictive models, we used different strategies and analyzed what were the trade-offs between each one of them. The multilayer perceptron (MLP) model is one of the simplest neural network models and this practicality has caused its increase on popularity. On the other hand, ML focused on the maximization (support) of separating the margin between classes (vector), also called SVM learning, is a powerful classification tool that has been used widely on many applications and achieved great results.

RNNs are frequently used in application involving sequential data, which fits the temporal nature of aging. However, RNNs frequently fail to learn the important information from the input data involving learning long-term dependencies. By introducing gate functions into the cell structure, the LSTM is able to handle the problem of long-term dependencies well [14]. Since its introduction, almost all the results based on RNNs have been achieved by LSTMs. The many applications include machine translation, time series prediction, natural language processing, and Computer Vision among others [15]. Because of the influence of previous voltages on aging, LSTM's ability to successfully train on data with long-term temporal dependencies makes it natural choice for this application [16].

### *C. Brain-Inspired Hyperdimensional Computing*

Brain-inspired HDC is a lightweight alternative to traditional ML approaches. It is a rapidly emerging concept that has been successfully applied to voice recognition [17], and hand gesture identification [18], seizures detected [19], image classification [20], pattern recognition for wafer defect maps [21], circuit reliability estimation [22], [23], and others. Implementations range from low-power embedded devices [24] to high-power GPUs [25]. HDC is based on the concept of hypervectors, vectors with thousands of dimensions. The hypervectors can consist of simple bits, integers, real numbers, or other symbols.

Hypervectors representing real-world values (e.g., 0.7 V) are generated once and stored in the item memory. If the same value has to be mapped into hyperdimensional space again, the previously generated item hypervector is retrieved from the item memory. Due to the high dimension, it is very likely that two randomly-generated hypervectors are orthogonal to each other. For binary hypervectors, this similarity metric is computed with the Hamming distance, for integer-based hypervectors using the cosine similarity.

Multiple item hypervectors are combined into a class hypervector through the basic operations of bundling and binding [4]. This process is also called encoding. A voltage waveform is encoded into a single hypervector which then represents said waveform. If a similar waveform is encoded, then its resulting hypervector has a high similarity to the first hypervector. Each operation is executed on the individual independent components of the hypervector making them trivial to parallelize.

Traditional ML methods such as DNN require huge amounts of data and lots of processing power for training [17]. HDC promises to reduce these requirements. Learning from few samples has been demonstrated for the example of seizure detection [26]. The distributed design of hypervectors makes HDC very robust against failures in the underlying memory and thus well suited for less reliable low-power emerging memories [27]. The design makes it also robust against noise in the data, e.g., from low-quality aging monitors embedded in the circuit. Additionally, HDC operations are trivial to parallelize to make use of multiple processing units. All these properties suggest that an ideal aging model can be implemented with HDC.

## III. Methodology and Experimental Setup

To evaluate the impact of transistor aging on a circuit, the analysis starts at application level. The activities of the application generates the stimuli for the inputs of the circuit (a NAND gate in this example) as shown in Fig. 3 ❶. Those stimuli are then propagated to the individual transistors in ❷. In larger circuits, not every transistor is connected to an input and thus its stimulus depends on the logic inside the circuit. Therefore, the circuit has to be simulated to extract

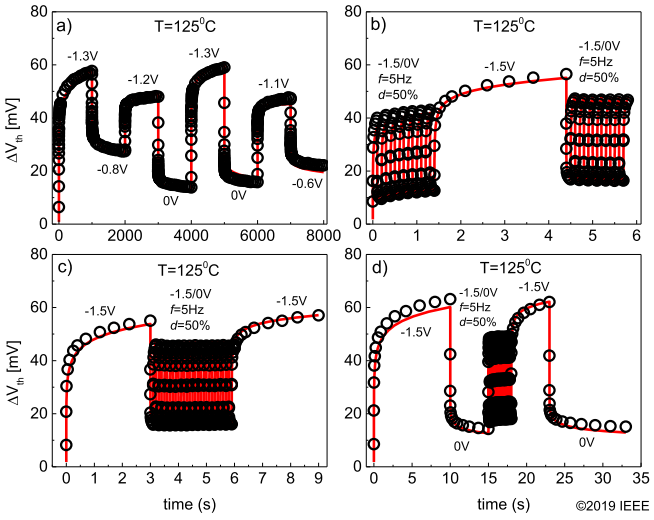


Fig. 4. Circles represent NBTI-induced $\Delta V_{th}$ measurements and the lines the physics-based aging model. (a) $\Delta V_{th}$ from different gate voltages. (b) High/low/high switching frequency experiment. (c) Low/high/low, and (d) High frequency switching during a low-frequency switch. From [28].

the voltage waveforms. In ❸, the waveforms are provided as an input to the aging models which generate the corresponding degradation trace. Based on this short-term trace, the EoL degradation is extrapolated, typically to ten years ❹. The resulting EoL $\Delta V_{th}$ for each transistor is applied to the circuit ❺ and causes an increase of the propagation delay or latency. Only if this aging-induced shift is considered during design can the system continue functioning properly over its whole lifetime.

This work builds on top of the CARAT framework [29] to simulate circuits with SPICE, extract the voltage waveforms, run the aging models, and simulate again to determine the additional propagation delay. A circuit designer can have access to such a framework except for the aging models, which contain sensitive parameters that the foundry does not share. Consequently, the whole flow does not benefit the designer because they do not know how much guardband each transistor requires. To explore the problem space, the state-of-the-art physics-based BTI Analysis Tool (BAT) framework [2] is employed. It estimates the impact of NBTI on different transistor technologies and manufacturing processes. BAT has been validated against measurements from several technologies including FinFET, FD-SOI, and nanosheets. The validation results are shown in Fig. 4 and demonstrate that the physics-based model captures recovery effects [28].

BAT models the generation of interface and bulk oxide traps as well as hole trapping and other aging effects, including recovery. The model has been calibrated with experimental measurements to obtain the otherwise confidential parameters. Such an effort is infeasible for most designers and not possible for technologies in the early prototype stage. A foundry can employ their confidential models and apply the proposed methodology analogously for new technology nodes.

Process variation in the transistors is not investigated in this work because it is orthogonal to aging and can be included independently. Using the $\Delta V_{th}$ value predicted by this work, the transistors are either annotated at the circuit level in simulators like SPICE or after cell library characterization with static timing analysis, which additionally considers process variation. If the design is based on standard cells, previous work has proposed an ML-based flow for extremely fast cell library characterization. Such a flow allows the designer to include the individual transistor aging information into an dedicated standard cell library for their design and workload [30].

In this work, an LSTM-based neural network, a 3-layer MLP, traditional SVM, and the emerging brain-inspired HDC are investigated. The input to all models is the waveform of switching activity and the ML models predict the corresponding $\Delta V_{th}$ trace, with the exception of the 3-layer MLP only employed to predict the last $\Delta V_{th}$ value. However, only the LSTM model supports the direct generation of a $\Delta V_{th}$ trace. The SVM and HDC require an iterative history-based approach in which a part of the waveform and previous $\Delta V_{th}$ values are employed to predict the next $\Delta V_{th}$ value. The next section introduces the employed ML models, Section III-B details the dataset generation, and Section III-C describes the general data representation and pre-processing in more detail. The history-based approach is described in Section IV.

### A. Description of the ML Models

SVM is based on statistical learning frameworks. Training samples are assigned to one of two groups. To support more classes (i.e., more fine-grained $\Delta V_{th}$ values), the problem is mapped to multiple binary classifications. The employed Scikit-learn library provides an SVM written in C. An SVM can be extended to a nonlinear classifier using the kernel trick. We perform a grid search to find the best model parameters and utilize the SVM implementation of the Scikit-learn library [31]. The core parts have been implemented in C. To predict only the last $\Delta V_{th}$ value and not the full trace, the SVM is employed as a regressor (SVR). The methodology for training remains the same.

The recently-proposed OnlineHD is selected as an HDC implementation [25]. It uses the MAP hypervector architecture [32], in which real numbers are the hypervector components. The distance between two hypervectors is computed with the cosine similarity. OnlineHD supports retraining to increase the prediction accuracy. During retraining, the model is queried with the training dataset and if the prediction is incorrect, the class hypervector is slightly altered to be more similar to the query hypervector. In this work, the number of retraining iterations (epochs) is set to 50 and the learn rate to 0.01. Similar to SVM, major parts of OnlineHD have been implemented in C through PyTorch.

An LSTM model is implemented as an alternative method to the history-based approach with SVM and HDC. LSTM models have been show to work well in sequence to sequence learning applications such as translation tasks [33]. In some tasks, they have shown to perform better than gated-RNNs [34]. In this work, an an LSTM encoder-decoder model is trained to predict the full trace based on the input waveform. The encoder contains two layers of stacked LSTMs, each with 256 units, which learn to map the input waveforms to an internal fixed-size vector representations of size 256. The decoder is a one layer LSTM with 256 units and trained to map the fixed internal vector to the degradation trace. Similar to [33], the performance of the LSTM model is improved

by reversing the input waveforms. The LSTM model's performance improved as the number of layers and units in each layer increased, as did the model's complexity. It was observed that model tends to overfit when the number units is increased above 256. The LSTM model's performance tends to deteriorate when the number of segments in the input waveform is greater than 32.

The 3-layer MLP is also implemented with PyTorch. Its first and second layer are fully connected each with 128 neurons. The third layer has 32 neurons fully connected to the single output neuron, which outputs the final $\Delta V_{th}$ value for a waveform. Rectified linear unit (ReLU) is selected as the activation function for all layers. The 3-layer MLP is trained for 1000 epochs.

To allow for a fair comparison of the computational demands of the ML methods and against the physics-based BAT, all experiments are executed on an AMD Ryzen 9 3950X.

### B. Training and Test Dataset Generation

A total of three datasets are generated, from (a) standard cells, (b) an 8-bit adder circuit, and (c) a 32-bit MAC unit. The workloads are generated randomly by applying stimuli to the input terminals of the circuits. As depicted in Fig. 3, these stimuli are propagated to the internal transistors. Through SPICE simulations, the analog waveform for each transistor can be extracted, one for each transistor. The physics-based aging model is employed to generate the $\Delta V_{th}$ trace as the label for a waveform. In the evaluation, these datasets are employed in different combinations for training and test of the ML models. Note that an ML model does not memorize all the training samples to reproduce them but learns the underlying patterns in the data. In other words, the ML model is trained to mimic the behavior of the physics-based aging model.

The first dataset is generated from 62 standard cells (e.g., XOR, full adder). The cells employed in this work have at most five input terminals and no internal state. With the design of digital circuits in mind, those input terminals are either at 0 V or at $V_{DD}$. Depending on the type of the cell, each standard cell contains between 4 and 27 pMOS transistors. In total, all standard cells contain 414 pMOS transistors. Thus, 414 waveform-trace pairs, the samples, can be generated.

While the design of the standard cells is well known, the designer's circuits is their intellectual property that cannot be shared with third parties like the foundry. Therefore, we mimic the application scenario for a circuit designer and generate datasets from transistors in larger circuits. In this work, two circuits are explored. First, an adder for two 8-bit numbers with 111 transistors. Second, a 32-bit MAC unit, that multiplies an 8-bit weight with an 8-bit input and accumulates the result with a 32-bit partial sum. The circuit contains 1395 pMOS transistors. The inputs of each circuit are stimulated with random data for an unbiased evaluation. A circuit designer would simulate their typical workload patterns. Similar to the standard cells, the inputs propagate through the circuit and waveforms for each transistor are extracted. In other words, the designer extracts waveforms representing their workload. For evaluation purposes, the physics-based aging model is employed again to compute the traces as a ground truth. The number of consecutive addition or MAC operations can be set to generate waveforms of various lengths. The longer the trace, the more it challenges the ML model since more features (input voltages) have to be considered.

Although this work focuses on circuits with well-known workloads, e.g., a 32-bit MAC unit, it can be deployed for more versatile circuits, such as a CPU, as well. A CPU comprises many simpler functional units, e.g., an ALU or a floating point unit. For such units, representative workloads can be generated at the design stage and the impact of aging on the individual transistors estimated. The utilization of these units and thus their overall aging is not captured by this unit-wise approach but relies on the circuit designer's knowledge about the potential use of the overall system. At this point, the system-wide workload can have a significant impact.

### C. Data Representation and Preprocessing

Circuit designers have access to foundry-provided PDKs to create and tune their systems. Typically, the foundry publishes an additional set of PDKs with aging data under worst-case conditions, which lead to an overestimated guardband. Actual workloads are far from such worst-case conditions. Therefore, aging models take the workload into account to predict the expected degradation at EoL for a single transistor. The input to the aging model is a waveform $(V_1, \ldots, V_l)$ which is a sequence of $l$ segments where each segment $V_i$ with $i \in \{1, \ldots, l\}$ represents the gate voltage applied to the transistor. The supply voltage can be any of the voltage corners provided by the foundry $V_i \in V_{corners}$. The time component is included in the waveform through the segment index, with each segment lasting the same amount of time.

Physics-based models can take the whole waveform and compute the expected $\Delta V_{th}$ for each point in time. To make such a model accessible to the designer, it has to be replaced with a similarly behaving ML-based model to not disclose the confidential technology parameters. Physics-based models retain the state of the transistor (e.g., the number of defects in the material) during the prediction, which is the basis for their powerful predictive capabilities. In contrast, lightweight ML-based methods do not have such an internal state and have to predict $\Delta V_{th}$ iteratively.

The waveform is provided to the aging model producing a $\Delta V_{th}$ trace $(\Delta V_{th,1}, \ldots, \Delta V_{th,l})$, i.e., a $\Delta V_{th}$ value for each segment. The effect of the input voltage is reflected in the output trace $V_i \rightarrow \Delta V_{th,i}$. However, simply using this mapping as a model does not reflect the voltage dynamics and cannot capture recovery effects. The $\Delta V_{th,i}$ of segment $V_i$ depends also on the previous segment's $V_{i-1}$, as show in Fig. 5. To capture this with the light-weight SVM and HDC models, a history-based approach is proposed and described in Section IV.

For all ML models, the data of the waveforms and $\Delta V_{th}$ traces is pre-processed to make it easier to learn. The pre-processing includes a normalization of the input voltage and the $\Delta V_{th}$ values from the minimal and maximum values found in the training dataset to a range between zero and one. Furthermore, the waveform has to be sampled because the CARAT framework only extracts the time and new voltage for each transition. For example, a transistor constantly turning

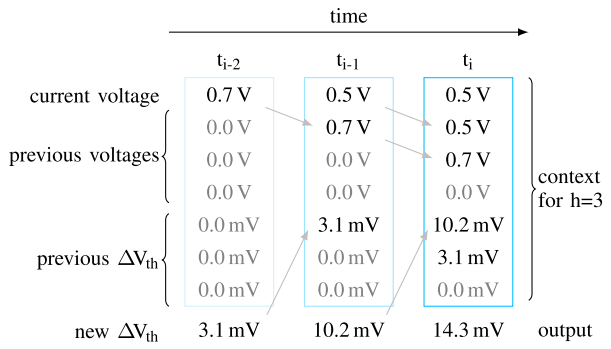


Fig. 5. Some history is added to the current input voltage to better capture the voltage dynamics. In this example $h = 3$, i.e., the input voltage and $\Delta V_{th,i}$ from $t_{i-1}$, $t_{i-2}$, and $t_{i-3}$ are included.

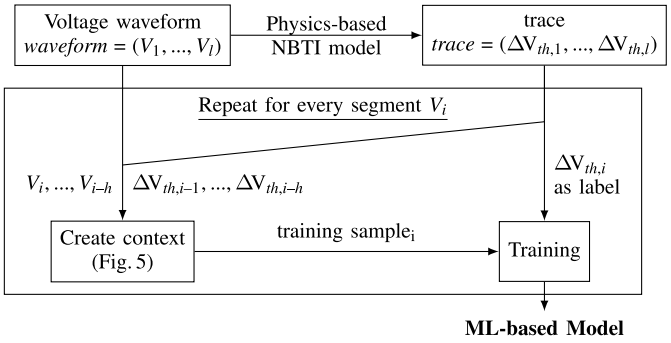


Fig. 6. Voltage waveforms derived from circuit-level stimuli are supplied to the physics-based transistor aging model to create training data for the machine learning-based models. Once they are trained, they take voltage waveforms and predict the degradation trace.

on and off produces a waveform with more samples than a transistor that is only turned on once. Since the employed ML models have a fixed input size, they require the waveforms to have a fixed number of samples. Hence, the waveforms are resampled to always have 32 samples, even if those samples are all the same. For the HDC model, the analog waveform voltages are additionally quantized into 50 levels.

## IV. Scenario 1: Predicting a Full Trace

The objective is to predict a $\Delta V_{th}$ for each segment of the waveform. In contrast to an LSTM, an SVM or HDC cannot directly convert a sequence to another. Hence, the waveforms have to be processed to make them learnable by the latter models. The training procedure for one waveform is sketched in Fig. 6. Since the current state of the transistor is not available for training, the voltage dynamics have to be captured with a history of $h$ previous waveform segments. However, such a snippet of the waveform sequence is not bound to a specific point in time or, more importantly, to the current internal state of the transistor. Setting $h = l$ (i.e., include all segments) is not viable due to the prohibitively large parameter space. Thus, a history of the $h$ previous $\Delta V_{th}$ values is included as well. The combination of voltage and $\Delta V_{th}$ provides a more detailed context for training. Fig. 5 visualizes the information contained in three training samples for $h = 3$. The label for each sample at time $i$ is the $\Delta V_{th,i}$ of the segment taken

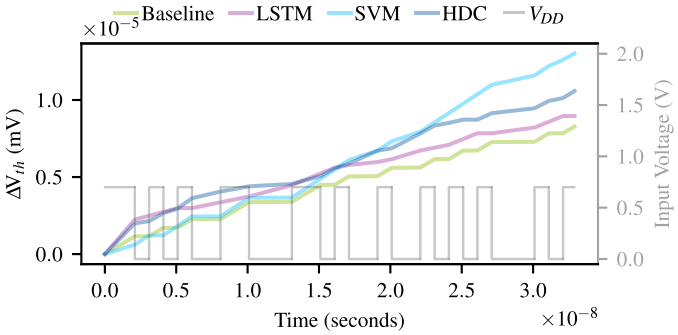


Fig. 7. Example of a waveform (gray), the baseline trace from the physics-based model (green) and the predicted traces from various ML models.

from the trace. The $\Delta V_{th,i}$ is quantized to discrete labels for classification.

The results show that the SVM and HDC models have a bias in their predictions. Although their predictions follow the traces in general, the nominal $\Delta V_{th}$ values often deviate. A multiplier can reduced this offset. After the model training is complete, it is used on the training set itself to predict the traces. The disagreement between the ML-based and the physics-based model is analyzed and the resulting average deviation is used as a multiplier during inference.

### A. Inference

During inference, the same data representation, described above, is used for SVM and HDC. This representation includes the $h$ previous $\Delta V_{th}$. However, only the waveform is available during inference. Hence, the $\Delta V_{th}$ values have to be predicted online during inference. They are then adjusted with the multiplier to be directly used to predict the next segment. For the first segment $i = 1$, the initial $\Delta V_{th}$ and the "previous" $\Delta V_{th}$ is set to 0 mV, as shown in Fig. 5. In effect, only the input voltage $V_1$ determines $\Delta V_{th,1}$. The predicted $\Delta V_{th,1}$ is then used to create the context for segment $i = 2$, $\Delta V_{th,1}$ and $\Delta V_{th,2}$ for segment $i = 3$, and so forth. Due to this recursive process, prediction errors multiply requiring high precision.

### B. Evaluation

The performance of the ML-based models under a variety of different aspects is evaluated. The datasets are generated and split into training and testing set with a 70 % split. As a metric, the relative error per segment $RE_i = (ML_i - BAT_i)/BAT_l * 100\,\%$ is used. $ML_i$ and $BAT_i$ are the predictions for segment $i$ from the ML-based and physics-based models, respectively. The difference is divided by $BAT_l$, the final $\Delta V_{th}$. Overestimating the degradation results in a positive $RE$, underestimating it in a negative. The results in Fig. 8 show balanced models with a tendency for overestimation.

### C. Dimension of the HDC Model

The dimension of the hypervectors determines their capacity to store information. The higher the dimension, the higher the expected accuracy. This increase levels off at an application-specific point, which is not known a priori. A higher dimension

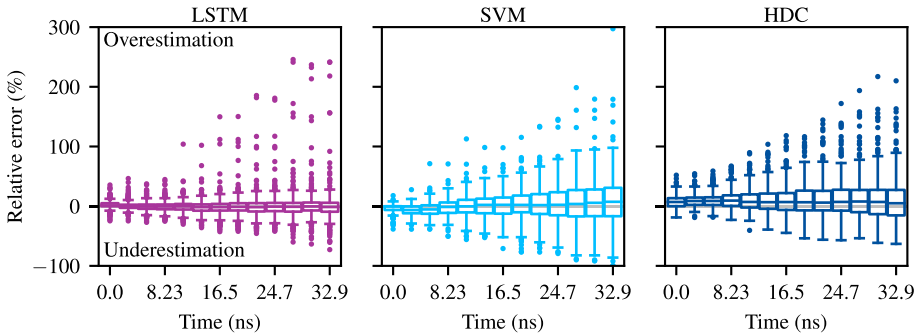


Fig. 8. The SVM and HDC models rely on their own previous outputs for the next prediction. Hence, the error accumulates, which is represented by the increase in the relative error. The LSTM directly translates the whole sequence and achieves a higher accuracy although outliers are as bad as in other models. Training and test are preformed on the adder circuit.

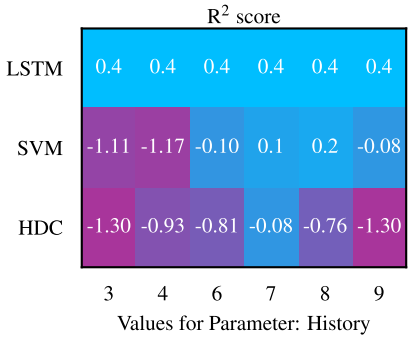


Fig. 9. Mean $R^2$ scores for different histories $h$ with an HDC dimension of 20k for the adder dataset.

TABLE I
EXECUTION TIMES FOR THE 8-BIT ADDER CIRCUIT

| Task | | Wall-clock time |
|---|---|---|
| Training set generation | (total) | 409.0 s |
| HDC training | (total) | 34.2 s – 152.3 s |
| SVM training | (total) | 407.7 s |
| Physics-based trace prediction | (mean) | 602.2 ms |
| HDC trace prediction | (mean) | 28.7 ms – 88.1 ms |
| SVM trace prediction | (mean) | 624.1 ms |
| LSTM trace prediction | (mean) | 1006.7 ms |

also correlates with more costly operations and higher memory requirements. Both costs are not the primary concern during design time. Therefore, HDC-based models with high dimensions above 10 000 are feasible. In this work, dimensions from 1000 to 20 000 vector elements are explored. Contrary to the initial assumption, the highest dimension does not necessarily result in the highest accuracy.

### D. Impact of History h

The parameter $h$ determines how many previous segments are taken into account to predict the next segment. This parameter describes how many previous voltages and $\Delta V_{th}$ values should be considered in the prediction of the next $\Delta V_{th}$ value. Recovery effects, as shown in Fig. 4, have a crucial impact on the degradation and depend on the change in the gate voltage of the transistor. Hence, considering such previous values is indispensable to capture recovery effects, which are omitted in previous related works. However, if too much history is considered, the input to the ML models becomes to complex to capture. Hence, a sweet spot between not considering recovery effects (low history parameter) and the complexity of the input (high history parameter) has to be found. Fig. 9 visualizes this exploration. The presented $R^2$ scores confirm the intuition.

At low history values of three to four, recovery effects are not captured resulting in lower $R^2$ scores. The SVM performs best with $h = 8$. For HDC, the combination with the dimension has to be considered. More history requires a higher capacity of the model to contain the information. While this capacity is available with the high dimensions, the results suggest an oversaturation of the query hypervector with the same voltage hypervector. A different encoding is expected to mitigate this issue. The overall best performances for HDC are achieved with $h$ of seven. For the LSTM, the history parameter does not have any impact because the LSTM is trained directly on the full waveform.

The hyperparameters dimension and $h$ can be selected based on the model's performance on the training data. Our analysis of circuits, discussed in Section V, shows that different settings are required depending on the workload characteristics. The best model is selected by the foundry and send to the designer.

### E. Reduction of Model Execution Time

In the HDC model, the complex differential equations of the physics-based model are replaced with simple operations on integer vectors. The performance advantages are reflected by a reduced execution time shown in Tab. I. Predicting a 32-segment trace for the 8-bit adder takes 29 ms to 88 ms for a dimension of 1000 and 20 000, respectively. This is up to 30X faster compared to the physics-based model with 602 ms or the SVM with 624 ms. The time for training varies with $h$, but it is consistently lower for the HDC model compared to the SVM. OnlineHD utilizes multiple CPU cores to reduce the training time. The LSTM takes the most time, even longer than the physics-based model, but achieves the best accuracy.

## V. SCENARIO 2: END-OF-LIFETIME AGING

Recreating the degradation trace is useful in evaluating short-term aging effects [5]. To predict the degradation at the EoL of the device, and thus for circuit designers to add

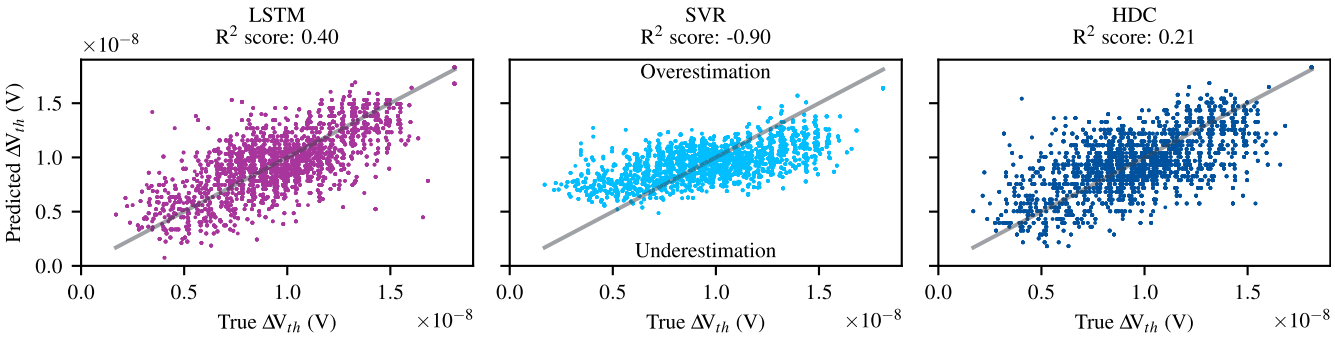


Fig. 10. Training and testing on the same circuit provides a baseline for the complexity of the problem to predict the final ΔVth value based on the trace. The LSTM predicts the whole trace but only the last value is considered in the evaluation.

sufficient guardbands, the whole trace is not necessary. The extrapolation model for NBTI considers the waveform as well as the last $\Delta V_{th}$ value. Hence, an ML model is sufficient for EoL $\Delta V_{th}$ estimation if it can predict this last value. Consequently, the challenge transforms from a recursive trace reconstruction to a simpler regression. With the focus on long-term aging, the final impact of inaccurate predictions from ML models can be evaluated at circuit level. The physics-based BAT is replaced with an ML model to provide the short-term aging value. This result is then processed further by the CARAT framework to predict the aging-induced change of the circuit's delay.

### A. Model Training and Evaluation

Many ML algorithms exist to solve regression problems. The input is a waveform, where each segment acts as a feature and the predicted output is the last $\Delta V_{th}$ value.

An SVM can also be used for regression and is then referred to as an **SVR**. The implementation is based on the Scikit-learn library [31] and a grid search is done for hyperparameter tuning. The SVR has an Radial Basis Function kernel, a gamma value of 0.001, and a C of 100. An **MLP** is implemented with PyTorch [35]. It has a total of three layers with 128 neurons in the hidden layer. The output layer is a single neuron. In contrast to classification, this single neuron returns a floating point value representing the last $\Delta V_{th}$. An **HDC** classifier can be used for regression by quantizing the $\Delta V_{th}$ values and treating those as classes. For comparison, a **worst-case model** is created. With NBTI, the pMOS transistor ages if no gate voltage is applied. Hence, the worst case assumes that the transistor is turned off and only turns on at the end of the simulated time frame to maximize the aging effect.

Each model is trained and evaluated on three circuits. The dataset generation is described in detail in Section III-C. The aging extrapolation models for NBTI depend on the last $\Delta V_{th}$ value and the waveform. But instead of the physics-based aging model, the ML models are employed. The predictions are compared with the output of the physics-based model as a baseline. As an accuracy metric, $R^2$ score is select, with a value of '1' as a perfect match.

### B. Results at Circuit Level

To judge the complex of the problem, the models are trained and tested on the adder circuit. Three sets of random inputs are generated for training, hyperparameter tuning, and testing. The results are presented in Fig. 10 and show the correlation between the baseline physics-based last $\Delta V_{th}$ values and the ML-based ones for all transistors. The $R^2$ scores are given above the plots and show that the best ML approach is the LSTM model. An $R^2$ score of 0.37 was achieved with a training for 500 epochs, two hidden layers with 25 units per layer, and the L1 loss function. Although there is some spread around the baseline, the model's ability to predict the $\Delta V_{th}$ is clear. A similar picture is given by the MLP, the predictions are correlated with the baseline values. For HDC, the spread is even larger but still follows the baseline. The model is trained with a dimension of 4000 for 50 epochs.

TABLE II
AGING-INDUCED DELAY FOR TRAIN ON ADDER AND TEST ON MAC

| Delay (ps) | Baseline | LSTM | SVR | HDC | Worst Case |
|---|---|---|---|---|---|
| min | −2.08 | −2.08 | −0.40 | −1.81 | −2.12 |
| mean | 4.88 | 4.66 | 5.33 | 4.88 | 31.36 |
| max | 12.60 | 11.70 | 11.70 | 13.90 | 61.10 |

While HDC has the highest spread, the mean aging-induced shift in the propagation delay at circuit level is equal to the baseline. Tab. II compares the different models and also includes the worst-case model with constant aging stress. Both, HDC and MLP, overestimate the impact overestimate aging, which is preferable to the LSTM, which underestimates the impact and thus could lead to insufficiently small guardbands. However, even with doubling the ML-based predictions to save guard against underestimation, the ML-models still outperform the worst-case model by a factor of three.

Training and predicting for the same circuit would require that the circuit designers share details with the foundry, which would train the model. To minimize data sharing, the foundry can train a model on their standard cells and provide those models to designers. However, the results plotted in Fig. 11 show a significant degradation of the quality of the predictions. The $R^2$ scores drop below zero and the models struggle to generalize. Worse, the LSTM and the MLP predict low $\Delta V_{th}$ values although the baseline values are close to the maximum (prediction in lower right corner). While the spread of the SVM has increased compared to a training with the adder, the maximum prediction errors are smaller than with other ML models. The HDC model has failed to generalize and is not included in the results.

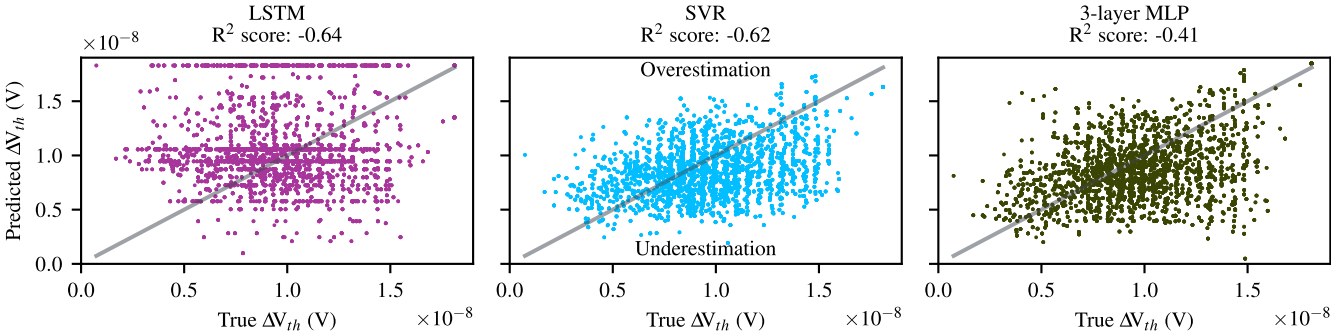


Fig. 11. Standard cells are provided by the foundry and the base for many circuit designs. However, the training dataset generated from them is too small for the ML methods to sufficiently learn and generalize. Hence, the inference results with the adder circuit are worse compared to Fig. 10.

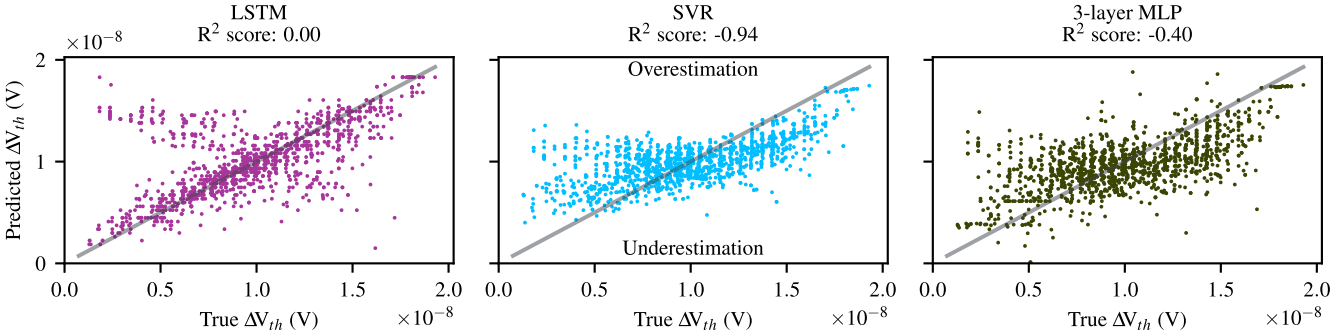


Fig. 12. The models are trained on the adder circuit and used to predict the degradation in the MAC circuit. The large dataset from the adder allows the LSTM to train and provide adequate results. The outliers are analyzed in Section V-C.

Similar and better results are shown in Fig. 12 for training on the adder and testing on the MAC circuit. First, the LSTM has sufficient data to train and can predict most samples with a low error. Nevertheless, outliers can cause incorrect guardband estimations. Second, while the SVM's $R^2$ is the lowest, it underestimations the least preventing severely incorrect guardband estimations. Overestimations are limited to smaller $\Delta V_{th}$ values and in total the SVM model achieves a mean relative error of 1.7 % compared to the LSTM's 3 %. Finally, the largest $\Delta V_{th}$ value in the MAC dataset is higher than in the adder and this behavior is not captured by the ML models, they are limited by their training. This is evident by the horizontal line-like cluster in the top right of the plots.

### C. Error Analysis

While many predictions of the LSTM are within a tolerable error range, there are outliers that are either under- or overestimated as shown in Fig. 12. The same samples are plotted in Fig. 13 for an error analysis. First, overestimated samples have a lower duty cycle and especially fewer voltage transitions in the waveform. In other words, transistors that are off most of the time and change their on/off state seldom. Overestimations have a negative impact on the circuit's timing because guardband are designed unnecessary large. However, they do not lead to failure of the device, in contrast to underestimations. The impact of aging is underestimated for some transistors with a duty cycle above 0.6. Their waveforms have an average amount of transitions. This combination of duty cycle and number of transitions is not a defining feature for underestimations by the LSTM model. Hence, it is impossible

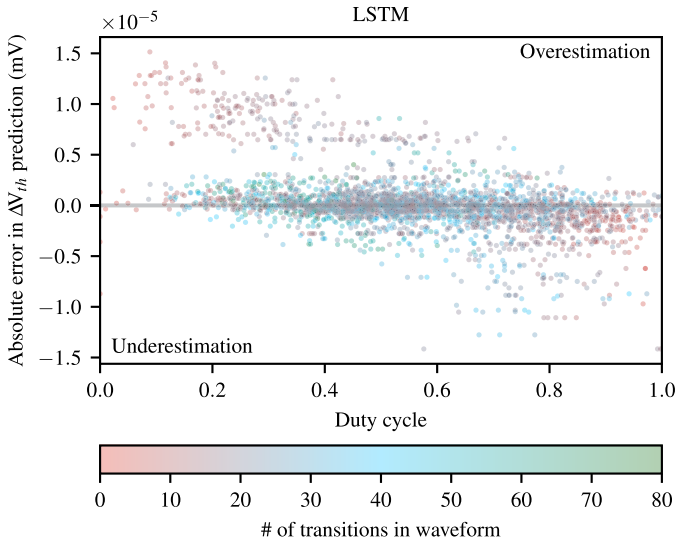


Fig. 13. Analysis of the LSTM model for the MAC circuit. The largest prediction errors are correlated with a low duty cycle and a low number of voltage transitions.

to derive a simple rule-based solution to contain the potential timing errors due to insufficient guardbands.

The SVR shows a similar pattern. Overestimations correlate with a low duty cycle combined with a low number of transitions in the waveform. Underestimations are not as frequent and as pronounced. While they occur mainly above a duty cycle of 0.4, worst-case underestimations do not correlate with the number of transitions in the waveform. Hence, similar to the LSTM, a simple rule-based error reduction cannot

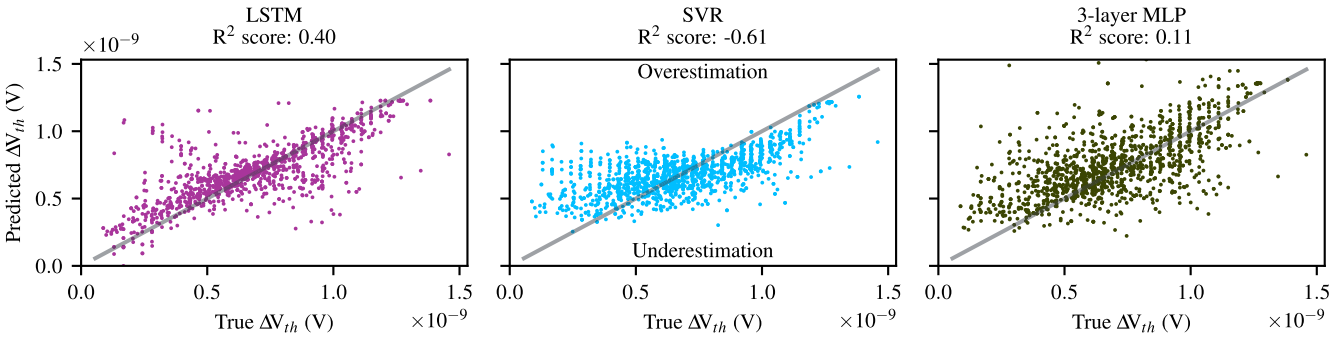


Fig. 14. Training on an 8-bit adder and testing on a 32-bit MAC circuit at 30 °C. The results align with the 90 °C case in Fig. 12 demonstrating the versatility of the proposed approach.

TABLE III
AGING-INDUCED DELAY FOR TRAIN ON STANDARD CELLS AND TEST ON ADDER CIRCUIT

| Delay (ps) | Baseline | LSTM | SVR | MLP | Worst Case |
|---|---|---|---|---|---|
| min | −2.08 | −4.33 | −0.93 | −1.14 | −2.13 |
| mean | 4.76 | 6.76 | 5.06 | 5.19 | 31.52 |
| max | 10.90 | 18.80 | 12.60 | 15.10 | 61.10 |

TABLE IV
AGING-INDUCED DELAY FOR TRAIN ON ADDER AND TEST ON MAC

| Delay (ps) | Baseline | LSTM | SVR | MLP | Worst Case |
|---|---|---|---|---|---|
| min | −1.80 | −2.90 | −2.10 | −3.80 | −70.70 |
| mean | 5.03 | 5.58 | 5.31 | 4.94 | 88.67 |
| max | 15.00 | 14.90 | 12.60 | 11.50 | 450.91 |

be derived. In summary, the models perform well for most samples but outliers, especially underestimations, still pose a challenge.

### D. Impact of Temperature on the Accuracy

The full flow is repeated at 30 °C. The models are trained on the waveforms of the 8-bit adder circuit and employed for the 32-bit MAC circuit, both at 30 °C. The results presented in Fig. 14 align with the results at 90 °C (Fig. 12) and demonstrate that the general approach is flexible and can be applied analogously to different operating conditions, opening a wide field of applications.

## VI. DISCUSSION

The focus of this work is on NBTI, the dominant degradation effect in current transistor technology [2]. Nevertheless, PBTI and HCD also play an important role. Their impact on the transistor has to be considered as well to design the circuit with small yet sufficient timing guard bands. Hence, an investigation into replacing those models with ML-based models is necessary. Preliminary results suggest that the methods explored in this work are challenged by the different types of stimuli driving those degradation effects. In NBTI, the on/off time is the dominant factor whereas in HCD the number of transitions has to be considered, among other stimuli.

Every modeling of a behavior, such as transistor aging, includes some inaccuracies. Modeling another model introduces another layer of inaccuracies, especially if ML methods are employed abstracting the behavior even further. However, the goal of this work is not the most accurate modeling of transistor aging and outperforming physics-based models, which are the most accurate. The goal of this work is to reduce the fundamental barriers of confidentiality for the circuit designers. Without our proposed ML models, they have to fall back to worst-case estimations and overestimating the impact of aging. Tab. IV shows the estimations from the ML models and the worst case. Even after doubling the ML models' prediction to account for the additional inaccuracies introduced by another layer of modeling, our work reduces the $V_{th}$ guardbands by 8× for the mean and 30× for the maximum. The speedup introduced by the HDC-based aging model also enables an exploration of the impact of aging on complex ML accelerators. The resulting timing errors would present a new fault model with which the ML algorithms would be confronted. Although some ML algorithms are robust against incorrect computations [36], timing errors can challenge them [37].

## VII. CONCLUSION

Accurate physics-based aging models include confidential technology and material parameters. Thus, such models are not available to circuit designers to optimize their designs under the actual impact of aging. This work explores the applicability of ML-based methods to train on the physics-based models, in particular traditional SVM, LSTM, and brain-inspired HDC. While ML-based models can predict the impact of aging for most transistors accurately, outliers can be over- or underestimated. Nevertheless, the explored ML-based methods predict the degradation about 3x more precise than available worst-case models. For the first time, circuit designers have access to an accurate aging model which is indispensable for efficient designs. This work opens the door to narrow the boundary between foundry and circuit designers.

## ACKNOWLEDGMENT

The authors would like to thank Souvik Mahapatra and his team for providing the physics-based aging models and Victor M. Van Santen for his help with the CARAT framework and insightful discussions.

## REFERENCES


[1] D. S. Huang et al., "Comprehensive device and product level reliability studies on advanced CMOS technologies featuring 7nm high-k metal gate FinFET transistors," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Mar. 2018, pp. 6F.7-1–6F.7-5.

[2] S. Mahapatra and N. Parihar, "Modeling of NBTI using BAT framework: DC–AC stress-recovery kinetics, material, and process dependence," *IEEE Trans. Device Mater. Rel.*, vol. 20, no. 1, pp. 4–23, Mar. 2020.

[3] S. Satapathy, W. H. Choi, X. Wang, and C. H. Kim, "A revolving reference odometer circuit for BTI-induced frequency fluctuation measurements under fast DVFS transients," in *Proc. IEEE Int. Rel. Phys. Symp.*, Apr. 2015, pp. 6A.3.1–6A.3.5.

[4] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognit. Comput.*, vol. 1, no. 2, pp. 139–159, Jun. 2009.

[5] V. M. van Santen, H. Amrouch, J. Martin-Martinez, M. Nafria, and J. Henkel, "Designing guardbands for instantaneous aging effects," in *Proc. 53nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2016, pp. 1–6.

[6] J. Keane and C. H. Kim, "Transistor aging," *IEEE Spectr.*, vol. 48, no. 5, pp. 28–33, Apr. 2011.

[7] V. M. van Santen, H. Amrouch, N. Parihar, S. Mahapatra, and J. Henkel, "Aging-aware voltage scaling," in *Proc. Design, Autom. Test Eur. Conf. Exhibit. (DATE)*, Mar. 2016, pp. 576–581.

[8] A. Das, R. A. Shafik, G. V. Merrett, B. M. Al-Hashimi, A. Kumar, and B. Veeravalli, "Reinforcement learning-based inter- and intra-application thermal optimization for lifetime improvement of multicore systems," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.

[9] K. Huang, X. Zhang, and N. Karimi, "Real-time prediction for IC aging based on machine learning," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 12, pp. 4756–4764, Dec. 2019.

[10] F. Klemme and H. Amrouch, "Machine learning for on-the-fly reliability-aware cell library characterization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 6, pp. 2569–2579, Jun. 2021.

[11] K. S. Kannan, M. Portolan, and L. Anghel, "Activity-aware prediction of critical paths aging in FDSOI technologies," *Microelectron. Rel.*, vol. 124, Sep. 2021, Art. no. 114261.

[12] N. Chatterjee, J. Ortega, I. Meric, P. Xiao, and I. Tsameret, "Machine learning on transistor aging data: Test time reduction and modeling for novel devices," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Mar. 2021, pp. 1–9.

[13] F. Arefaine et al., "Using long short-term memory (LSTM) network to predict negative-bias temperature instability," in *Proc. Int. Conf. Simul. Semiconductor Processes Devices (SISPAD)*, Sep. 2021, pp. 60–63.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[15] A. R. Priatama and Y. Setiawan, "Regression models for estimating aboveground biomass and stand volume using Landsat-based indices in post-mining area," *J. Tropical Forest Manag.*, vol. 28, no. 1, pp. 1–14, Apr. 2022.

[16] S. Hochreiter and J. Schmidhuber, "LSTM can solve hard long time lag problems," in *Advances in Neural Information Processing Systems*, vol. 9, M. Mozer, M. Jordan, and T. Petsche, Eds. Cambridge, MA, USA: MIT Press, 1996.

[17] M. Imani, D. Kong, A. Rahimi, and T. Rosing, "VoiceHD: Hyperdimensional computing for efficient speech recognition," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, Nov. 2017, pp. 1–8.

[18] A. Moin et al., "An EMG gesture recognition system with flexible high-density sensors and brain-inspired high-dimensional classifier," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[19] A. Burrello, S. Benatti, K. Schindler, L. Benini, and A. Rahimi, "An ensemble of hyperdimensional classifiers: hardware-friendly short-latency seizure detection with automatic iEEG electrode selection," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 4, pp. 935–946, Apr. 2021.

[20] Y. Chuang, C. Chang, and A. A. Wu, "Dynamic hyperdimensional computing for improving accuracy-energy efficiency trade-offs," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2020, pp. 1–5.

[21] P. R. Genssler and H. Amrouch, "Brain-inspired computing for wafer map defect pattern classification," in *Proc. IEEE Int. Test Conf. (ITC)*, Oct. 2021, pp. 123–132.

[22] H. Amrouch, F. Klemme, and P. R. Genssler, "Design close to the edge for advanced technology using machine learning and brain-inspired algorithms," in *Proc. 27th Asia South Pacific Design Autom. Conf.*, Jan. 2022, pp. 1–12.

[23] P. R. Genssler and H. Amrouch, "Brain-inspired computing for circuit reliability characterization," *IEEE Trans. Comput.*, vol. 71, no. 2, pp. 3336–3348, Dec. 2022.

[24] M. Imani et al., "Revisiting HyperDimensional learning for FPGA and low-power architectures," in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit. (HPCA)*, Feb. 2021, pp. 221–234.

[25] A. Hernández-Cano, N. Matsumoto, E. Ping, and M. Imani, "OnlineHD: Robust, efficient, and single-pass online learning using hyperdimensional system," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Feb. 2021, pp. 56–61.

[26] A. Burrello, K. Schindler, L. Benini, and A. Rahimi, "One-shot learning for iEEG seizure detection using end-to-end binary operations: Local binary patterns with hyperdimensional computing," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2018, pp. 1–4.

[27] G. Karunaratne, M. Le Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian, "In-memory hyperdimensional computing," *Nature Electron.*, vol. 2020, pp. 1–11, Jan. 2020.

[28] S. Salamin, V. M. Van Santen, H. Amrouch, N. Parihar, S. Mahapatra, and J. Henkel, "Modeling the interdependences between voltage fluctuation and BTI aging," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 7, pp. 1652–1665, Jul. 2019.

[29] V. M. van Santen et al., "BTI and HCD degradation in a complete 32 × 64 bit SRAM array—Including sense amplifiers and write drivers—Under processor activity," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2020, pp. 1–7.

[30] F. Klemme and H. Amrouch, "Scalable machine learning to estimate the impact of aging on circuits under workload dependency," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 5, pp. 2142–2155, May 2022.

[31] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, 2012.

[32] R. W. Gayler, "Multiplicative binding, representation operators & analogy (workshop poster)," in *Advances in Analogy Research: Integration of Theory and Data From the Cognitive, Computational and Neural Sciences*, D. Gentner, K. Holyoak, and B. Kokinov, Eds. Sofia, Bulgaria, Jul. 1998, doi: 10.1006/cogp.1997.0674.

[33] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2014.

[34] R. Cahuantzi, X. Chen, and S. Gttel, "A comparison of LSTM and GRU networks for learning symbolic sequences," 2023, *arXiv:2107.02248*.

[35] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., 2019, pp. 1–12.

[36] S. Thomann, P. R. Genssler, and H. Amrouch, "HW/SW co-design for reliable TCAM-based in-memory brain-inspired hyperdimensional computing," *IEEE Trans. Comput.*, early access, Feb. 23, 2023, doi: 10.1109/TC.2023.3248286.

[37] J. Krautter, P. R. Genssler, G. Sepanta, H. Amrouch, and M. Tahoori, "Stress-resiliency of AI implementations on FPGAs," in *Proc. Int. Conf. Field Program. Log. Appl. (FPL)*, 2023, pp. 1–12.


**Paul R. Genssler** (Member, IEEE) received the Dipl. Inf. (M.Sc.) degree in computer science from TU Dresden, Germany, in 2017. He is currently pursuing the Ph.D. degree with the Semiconductor Test and Reliability (STAR) Chair, Computer Science, Electrical Engineering Faculty, University of Stuttgart. In 2018, he started his Ph.D. research with the Chair for Embedded Systems (CES), Karlsruhe Institute of Technology, Germany. His research interests include emerging technologies, system architecture, and emerging brain-inspired methods for IC test and beyond.

**Hamza E. Barkam** (Member, IEEE) received the Bachelor of Engineering degree in telecommunications engineering from Universitat Politecnica de Catalunya, Spain, in 2020, and the M.Sc. degree from the University of California at Irvine, USA, in 2022. In 2020, he started his Ph.D. research with the Engaging Technology and Application Design Laboratory, CaliT2, with Professor Sergio Gago. Since 2021, he has been with the Bio-Inspired Architecture and Systems (BIASLab). His research interests include theoretical work on vector symbolic architectures, hyperdimensional computing, and secure machine learning.

**Karthik Pandaram** received the Bachelor of Engineering degree in electrical and electronics engineering from the Coimbatore Institute of Technology, India, in 2020, and the M.Sc. degree in information technology with a specialization in embedded systems engineering from the University of Stuttgart, Germany, in 2023. His research interests include deep learning, automated driving, intelligent sensor systems, semiconductor test, and reliability.

**Mohsen Imani** (Member, IEEE) received the Ph.D. degree from the Department of Computer Science, UC San Diego. He is currently an Assistant Professor with the Department of Computer Science, UC Irvine. He is also the Director of the Bio-Inspired Architecture and Systems (BIASLab). His contribution has led to a new direction on brain-inspired hyperdimensional computing that enables ultra-efficient and real-time learning and cognitive support. His research was also the main initiative in opening up multiple industrial and governmental research programs. His research has been recognized with several awards, including the Bernard and Sophia Gordon Engineering Leadership Award, the Outstanding Researcher Award, and the Powell Fellowship Award. He also received the Best Doctorate Research from UCSD and several best paper nomination awards at multiple top conferences, including Design Automation Conference (DAC) in 2019 and 2020, Design Automation and Test in Europe (DATE) in 2020, and International Conference on Computer-Aided Design (ICCAD) in 2020. Furthermore, he received the Best Paper Ward in DATE 2022 Conference.

**Hussam Amrouch** (Member, IEEE) received the Ph.D. degree (summa cum laude) from the Karlsruhe Institute of Technology (KIT) in 2015. He is currently a Professor heading the Chair of AI Processor Design, Technical University of Munich (TUM). He is also with the Munich Institute of Robotics and Machine Intelligence (MIRMI), Germany, and the Head of the Semiconductor Test and Reliability (STAR) Research Group, University of Stuttgart, Germany. Prior to that, he was a Research Group Leader with KIT, where he was leading the research efforts in building dependable embedded systems. He currently serves as an Editor for the *Scientific Reports* (Nature) journal. His main research interests include design for reliability and testing from device physics to systems, machine learning for CAD, HW security, approximate computing, and emerging technologies with a special focus on ferroelectric devices. He holds eight HiPEAC paper awards and three best paper nominations at top EDA conferences, such as DAC'16, DAC'17, and DATE'17 for his work on reliability. He has served in the technical program committees for many major EDA conferences, such as DAC, ASP-DAC, and ICCAD, and a reviewer for many top journals, such as *Nature Electronics*, IEEE TRANSACTIONS ON ELECTRON DEVICES, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and IEEE TRANSACTIONS ON COMPUTERS. He has more than 210 publications in multidisciplinary research areas (including around 90 journals) across the entire computing stack, starting from semiconductor physics to circuit design all the way up to computer-aided design and computer architecture. His research in HW security and reliability has been funded by the German Research Foundation (DFG), Advantest Corporation, and the U.S. Office of Naval Research (ONR).