Modeling Emerging Technologies using Machine Learning: Challenges and Opportunities

Invited Talk

Florian Klemme*, Jannik Prinz*, Victor M. van Santen*, Jörg Henkel*, Hussam Amrouch†
*Karlsruhe Institute of Technology, {klemme, victor.santen, henkel}@kit.edu
†University of Stuttgart, amrouch@iti.uni-stuttgart.de

***Abstract*—Compact models of transistors act as the link between semiconductor technology and circuit design via circuit simulations. Unfortunately, compact model development and calibration is a challenging and time-intensive task, hindering rapid prototyping of a circuit (via circuit simulations) in emerging technologies. Moreover, foundries want to protect their confidential technology details to prevent reverse engineering. Hence, they limit access to compact transistor models of commercial technologies (e.g., with Non-Disclosure-Agreements). In this work, we propose Machine Learning (ML) to bridge the gap between early device measurements and later occurring compact model development. Our approach employs a Neural Network (NN) that captures the electrical response of a conventional FinFET transistor without knowledge of semiconductor physics. Additionally, our approach can be applied to emerging technologies, using Negative Capacitance FinFET (NC-FinFET) as an example for a (challenging to model) emerging technology. Inherently, the black-box nature of ML approaches keeps technology manufacturing details confidential. Furthermore, we show how using solely R2 score as our fitness function is insufficient and instead propose fitness based on key electrical characteristics or transistors like threshold voltage. Our NN-based transistor modeling can infer FinFET and NC-FinFET with an R2 score larger than 0.99 and transistor characteristics within 5% of experimental data.**

***Index Terms*—Machine learning, Neural Network, Compact model, Transistor model, FinFET, Negative Capacitance FinFET**

I. Introduction to Compact Models

Circuit simulations are the key to the evaluation of a semiconductor technology at the circuit level. The performance, power, and efficiency of a specific circuit design in a specific technology are evaluated in circuit simulators such as SPICE. The transistors in such circuit simulators are implemented through compact models such as BSIM-CMG [1]. Compact models are an abstracted high-performance implementation of physical equations, which represent the electrical behavior of a transistor. A semiconductor manufacturer (foundry) can then use a set of parameters, called a *model card* to calibrate this behavior to the observed behavior (measurements or TCAD simulations) of a specific technology (e.g., 22 nm FinFET). However, this standard approach of circuit simulations faces three key challenges in the current nano-era of semiconductors.

Foundry Secrecy: First, the foundries are more and more reluctant to share transistor model cards, as this could allow reverse engineering of their commercial product. This is evident by the frequent use of *arbitrary units* (a.u.) in measurement data across publications [2]–[4] and the Non-Disclosure-Agreements (NDA) guarding Process Design Kits (PDK) access.

Innovation requires new Models: Secondly, with the end of Dennard scaling and the current age of innovation in semiconductor technologies, transistors are not simply scaled anymore but altered in their fundamental structure. For example, 32 nm technology introduced high-k metal gates and 22 nm introduced 3D FinFET. These innovations enable the continuous advancement of computing systems in terms of low power and high performance. Yet, for compact models, this innovation poses a serious challenge. With each innovation, it requires a new compact model instead of updated parameters in model cards. For example, BSIMv4 was the compact model for all planar MOSFETs from 1 µm to 45 nm, but since then we needed BSIM-SOI, BSIM-IMG, and BSIM-CMG to model various new transistor types. With Nanosheets, Negative Capacitance Transistors, and TunnelFET just around the corner, more demand for additional compact models is anticipated. However, compact model development can take years. Development can only start once the underlying technology is mature and its physics are understood. This delay in modeling availability hinders the use of standard EDA tools to evaluate circuit designs in such new technologies, which hinders market entry and prolongs time to market for the emerging technology.

Early Evaluation of Technology: Thirdly, semiconductor technology is divergent and frequently various competing transistor technologies exist simultaneously. For example, FinFET currently competes against FDSOI, Nanowire, and Nanosheet transistors in traditional CMOS and NC-FinFET, TFET, and other transistors in emerging technologies. Currently, each technology has prototypes that are measured as well as mate-

ICCAD '20, November 2–5, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8026-3/20/11
https://doi.org/10.1145/3400302.3415770

rial and physics simulations in TCAD. However, in order to evaluate if a particular transistor technology is suitable for a specific circuit currently in the design phase, we need support within the EDA standard flow (e.g., in SPICE simulators). Therefore, traditional evaluation mandates the development of various compact models and the calibration in various model cards. This a complex and thus costly and labor-intensive process, as transistors are calibrated for different voltages, temperature, variants (high-Vth, low-Vth, high power, etc.), and geometries to just name a few variables. Therefore, before a technology can be evaluated to determine commercial or academic interest, considerable investments are necessary to develop and calibrate compact models.

In this work, we propose Machine Learning (ML), specifically Neural Network (NN) based transistor models to act as an intermediate between early measurements from early prototypes and the accurate yet complex compact models. Our goal is to provide NN-based transistor models, tackling the three challenges outlined above. First, NN-based transistor models are by its very nature black-box modeling approaches and thus cannot expose manufacturing details about the transistors, which protects the intellectual property of the foundries. Furthermore, NN-based transistor models are generic, i.e. a single approach can apply to different transistor technologies (as we show in Section IV-H) and thus do not require continuous development of new models. Lastly, NN-based modeling can act without domain knowledge, i.e. model something without grasping the underlying fundamentals (physics, materials, etc.). This allows for the quick development of these NN-based models in the presence of transistor innovation. Combined with the ability to model a transistor based on limited data from early prototype measurements, NN-based transistor models are the rapid prototyping for the early design-space exploration of a technology (e.g., rough performance and power evaluations). Naturally, these limited data sets result in inaccurate and constrained models at such an early stage. This, in turn, can then either drive or deter further compact model development for the technology.

To provide a proof of concept, our NN-based transistor model is calibrated to a commercial $14\,\mathrm{nm}$ FinFET technology [5], as well as NC-FinFET [6] as our example of an emerging technology. The transistor model is able to match NC-FinFET just as well as conventional FinFET despite the material and structural innovations in NC-FinFET (ferroelectric gates, which acts as a negative capacitance providing voltage amplification and negative drain induced barrier lowering). Additionally, we show the capability of NN-based transistor models to provide reasonable accuracy in early evaluations, i.e. training with limited prototype data sets. Furthermore, we explore how to properly validate an NN-based transistor model. Our Section IV-F shows that regular ML fitness functions are insufficient for accurate transistor modeling and instead key transistor metrics should be used.

Our novel contributions within this paper are as follows:

- We present an ML approach that can capture and reproduce the electrical behavior of an industrial configured FinFET compact model. This enables circuit design evaluation via circuit simulations without the need for a compact model.
- We demonstrate that our ML approach also supports the modeling of NC-FinFET. Unlike established FinFET, NC-FinFET shows a more complex behavior due to a negative DIBL effect. Our approach can adjust to this new behavior without any manual intervention or change in configuration. *This hints to possible ML-based transistor modeling for new emerging technologies.*
- Finally, our solution incorporates secrecy in its fundamental structure. This protects the intellectual property of the foundries and enables a wide community to access the less-restricted technology modeling.

II. Related Work

Related work can be categorized into works using transistor knowledge and ones that do not.

A. ML-based Transistor Models with Domain Knowledge

Early works in ML-based transistor modeling had to work with limited resources. The number of neurons and the number of layers in the NN was limited by the computational power and tool support of the time. Therefore, various works used domain knowledge to augment the structure and weights of the NN, to improve its accuracy. While this augmentation leads to better performance, it cannot be applied to emerging technologies (which are not yet fully understood) and customizes (as well as constraints) the model for a particular technology (removes generality of the model).

An early approach of modeling transistors with Neural Networks (NN) is shown in [7]. In 1996, ML techniques were still in their infancy. For example, a MOSFET NN to predict the current-voltage (I-V) curve consists of only 12 neurons. Certain characteristics such as monotonicity were guaranteed by the training algorithms. Domain knowledge of the device is incorporated to mitigate errors and reduce the demand for training data.

A recent application of NN for transistor modeling is presented in [8]. Once again, small NNs with less than 15 nodes are used and transistor-specific knowledge is employed. Thus, nodes are connected manually, considering the physical dependencies within the transistor. This results in specific nodes to model the sub-threshold area and nodes which model just the area above the threshold voltage. To reduce required data to the minimum, they use a sparse, non-uniform data set.

The NN-based approach in [9] is heavily focused on incorporating device physics, targeting the emerging Thin-TFET transistor. Two separate NNs for V_{ds} and V_{tg} (top gate voltage) with less than 10 nodes each form the core structure. The V_{ds}-connected NN uses a tanh activation function to follow the transistor's linear behavior in the sub-threshold region and saturation behavior above the threshold voltage. Bias terms

are purposefully removed to force $I_d = 0$ at $V_{ds} = 0$. All these measures are taken to prevent the learning of physically impossible behavior.

B. ML-based Transistor Models without prior Knowledge

Approaches, which do not rely on prior knowledge can be generic (apply to many technologies), but require more computational resources.

A combination of NN and genetic algorithms is used in [10]. In addition to the regular training, the genetic algorithm is used to find the best structure of the NN. However, only regular planar MOSFET transistors are evaluated and only the current is inferred. Hence, the work lacks information if this could be applied to emerging technologies is missing as well as a sufficient validation (see Section IV-F for details).

[3] presents an NN-based surrogate model as a compact model alternative for novel transistors. They demonstrate their framework for FinFET as an established technology and TFET as an emerging technology with no available compact model. However, prediction accuracy is not discussed on transistor I-V curves. Therefore, their validation is lacking (see Section IV-F) and we cannot estimate their achieved modeling accuracy with respect to the transistor parameters (Vth, Ion, Ioff, etc.). Additionally, the traditional and emerging technologies are evaluated on different circuits, which does not allow for direct comparisons or any claim about the generality of their approach.

Last but not least, commercial tools using ML techniques for compact model generation are recently available. Silvaco TechModeler [11] uses ML techniques to build Verilog-A models from input samples. Unfortunately, we could not compare it to our work due to limited publicly available information.

C. Distinction from existing state of the art

In summary, existing works offer similar approaches but differ in these key points.

- We focus on a general-purpose NN approach without incorporating transistor knowledge. Therefore, our approach is generic and can model at least two different transistor technologies without any adaption.
- We highlight for the first time, the foundry secrecy concern and how ML-based approaches act as a suitable black-box model.
- Our validation does no rely on solely the R^2 score, which can be misleading when judging ML accuracy and fitness. Instead, we additionally consider the error on key transistor characteristics like on-current, off-current, threshold voltage, and sub-threshold slope, which are stricter fitness and accuracy constraints.
- We explore how our approach can work on limited data sets, e.g. for early technology evaluations, where limited experimental data is available.

III. Background

A. Compact models

Compact models are the interface to bring a transistor implementation into the SPICE simulator. They abstract the underlying physical equations to electrical behavior. They do not attempt to model the underlying physics directly but try to provide high accuracy with the minimum number of equations and parameters (to improve performance). Actual physical modeling of the transistors materials is done in TCAD simulators, which are very computational complex and thus time-consuming but also highly accurate. Unfortunately, they are not suited for circuit simulations due to their computational complexity [12]. Typically TCAD simulations (simulated transfer curves) and selected experimental data sets (measured transfer curves) are used to calibrate the parameters (the transistor model card) of a particular transistor (e.g., Intel FinFET) to a particular transistor model (e.g., BSIM-CMG). BSIM-CMG has over 100 parameters in the model card, spanning geometry, dopant concentrations, material, and electrical properties. Therefore, a compact model can only be developed when the physics is fully understood and the technology reached sufficient maturity so that most properties are fixed and a TCAD model is available. Then the labor-intensive compact model development can begin.

For example, the development of BSIM-SOI and BSIM-CMG compact models took more than two years until their first release. [13] And the development of compact models does not start immediately after the new technology is discovered. While silicon data for sub-20nm FinFET was already available in early 2000s [14], [15], it took over a decade for BSIM-CMG became widely available as "the world's first industry-standard compact model for the FinFET". [16] For emerging technologies, developing a compact model can be even more challenging as underlying physics is often not yet fully understood. For example, a Tunnel FET (TFET) was already created and measured in 2004 [17], yet compact models are still in an early development stage [3].

B. Negative Capacitance FinFET

NC-FinFET is an emerging transistor technology with a ferroelectric layer integrated into the gate stack of a FinFET transistor [18]. The ferroelectric layer provides a voltage amplification of the gate voltage applied to the transistor, and thus changes the electrical behavior of the FinFET. For example, its sub-threshold slope is below $60\,\mathrm{mV/decade}$, making NC-FinFET a promising technology to succeed FinFET. The sub-threshold slope is an important characteristic to determine the switching speed of a transistor. A steeper sloped can provide a higher on-currents or less leakage when the transistor is turned off. Sub-threshold slopes of both, FinFET and NC-FinFET, are shown in Figure 1.

Another characteristic of NC-FinFET is reduced short channel effects due to a low or even negative Drain-Induced Barrier Lowering (DIBL) effect [19]. Additionally, the ferroelectric gate increases overall gate capacitance C_{gg}. With the voltage-dependent voltage amplification, negative DIBL, C_{gg}-increase,

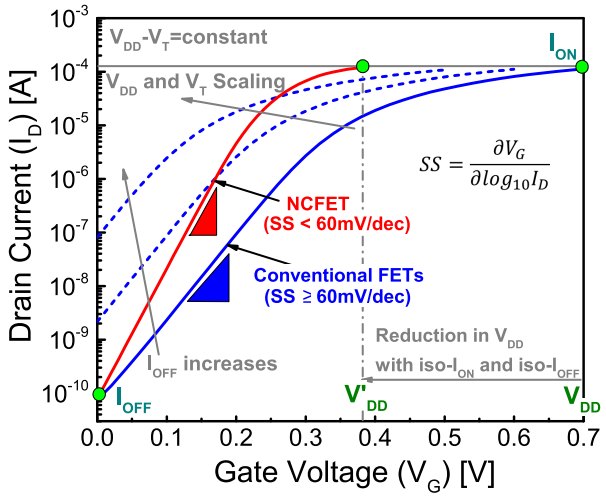


Fig. 1. Transfer curves of FinFET and NC-FinFET in comparison. The sub-threshold slope SS is an important characteristic for the potential of the technology. Image from [19]

and steep sub-threshold slope, NC-FinFET shows an increased complexity in its behavior compared to conventional FinFET. Therefore, it is a prime candidate to test if our approach is applicable to emerging technologies without modifications.

C. Transistor characteristics

Transistor characteristics are used to evaluate and compare the performance of the transistor with regard to certain properties. Due to their importance in evaluation, we use the following transistor characteristics as additional accuracy metrics in our ML approach.

- I_{off} the off-current at $V_{gs} = 0\,\text{V}$,
- I_{on} the on-current at $V_{gs} = V_{DD}$,
- V_{th} the threshold voltage, i.e., voltage at which the transistor turns on or off,
- SS the sub-threshold slope, i.e., fast a transistor turns on or off.

These transistor characteristics are also visualized in Figure 1 and Figure 2.

IV. Our Machine Learning Approach

In this work, we propose an NN to learn and reproduce the I-V curve (electrical response) of a transistor. If the NN-based transistor model can accurately predict the drain current of a transistor, then it is sufficient to be employed in SPICE simulations. SPICE model circuit components (resistors, capacitors, transistors, diodes, etc.) based on their conduction (inverse resistance) from each terminal (gate G, source S, drain D, bulk B for transistors) to each other terminal. Since SPICE knows the voltage at the terminals and the NN-based model provides the currents, the conductances can be calculated with $G = \frac{1}{R} = \frac{I}{V}$. Therefore, for the rest of this work, we solely discuss the I-V curve, also called the transfer curve.

A. Experimental Setup

Our NN-based transistor model is a prototype application in C++ using PyTorch for ML functionality. We use a fully connected feed-forward NN with 2 layers, 500 nodes in each layer, and PReLU as the activation function. The NN is trained using back-propagation with stochastic gradient descent. As established in ML, input values are normalized before training and inference, so that an adaptation of the learning rate to the training data is not required. For validation, 30% of the training data is preserved to be used as the validation data set.

For the generation of training data as well as for validation, existing compact models are used. BSIM-CMG [1] is used for conventional FinFET. A compact model [6] based on BSIM-CMG is used for emerging NC-FinFET. Details on the used NC-FinFET modeling and FinFET device calibration with industrial measurements are available in [20] and [21], respectively.

The hyperparameters of the NN are optimized based on the training data from the conventional FinFET model, as commonly done in ML. However, the NN has been purposely up-scaled slightly (to 2 layers and 500 nodes each), to increase the chance of better adaptability to different transistors. To allow a comparison of our work against previous works, we express accuracy in the traditional R^2 score, as well as our own metrics. The R^2 score expresses the mean error of the prediction in relation to the total variance. The R^2 score is defined as

$$R^2 = 1 - \frac{\sum (Y_{\text{true}} - Y_{\text{pred}})^2}{\sum (Y_{\text{true}} - \bar{Y}_{\text{true}})^2} \tag{1}$$

where Y_{pred} is the predicted value, Y_{true} is the actual value (from the test set), and $\bar{Y}_{\text{true}}$ is the mean value of the test set. An R^2 score of 1 indicates perfect accuracy, whereas a score around 0 represents randomly guessing a value around the mean. The interpolation of two individual NNs as described in the next section IV-B is implemented with a custom Python script, first invoking the individual NNs and interpolating the results afterward.

B. Data Scaling

Modeling the transfer curve is a challenge since this I-V curve spans multiple orders of magnitude in terms of currents (small leakage current, yet million times stronger drive currents). Therefore, applying standard ML techniques when using the mean squared error (MSE) as the fitness function during training is problematic. While this works fine for large current values, smaller values exhibit high relative errors. The MSE value is dominated by the mismatch in the high-value region and the errors in small values are not weighted enough. However, key transistor characteristics like the sub-threshold slope and leakage current I_{off} is determined within this lower-value range. Thus key transistor parameters are susceptible to error due to inadequate training. This is a common problem also faced by other works [7]. Figure 2 illustrates this problem. In the linear representation on the left side, the I-V curves of SPICE and the NN clearly overlap. In

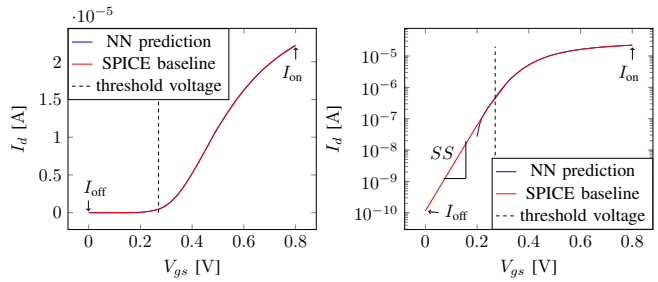


Fig. 2. I_d-V_{gs} curve with $V_{ds} = 0.1\,\text{V}$. To determine the sub-threshold slope (SS) visually, the plot is repeated in logarithmic scale on the right side.

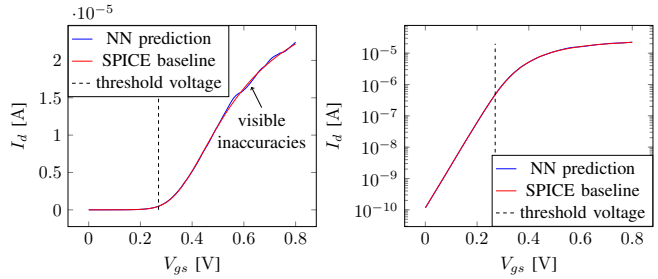


Fig. 3. With logarithmically scaled training data, the I-V curves show a good match. However, at linear scale, inaccuracies appear at higher values.

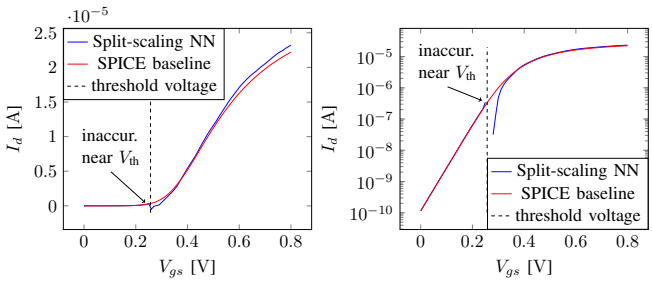


Fig. 4. The I-V curve with split-scaling around the threshold voltage. Hard to see: In the lower-value range, both curves match quite well. However, there are obvious errors around the threshold voltage.

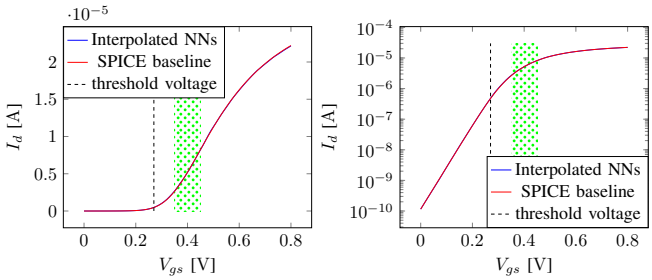


Fig. 5. Interpolation of two independently trained NNs.

the logarithmic plot on the right side, the problem becomes apparent: For lower values, the NN curve diverges from the SPICE baseline, eventually disappearing when negative values are predicted. The sub-threshold slope and leakage current cannot be derived properly.

Our solution to this problem is to scale the data before training. By applying a logarithmic scaling to the training data, the range of values is more equally weighted, i.e. formerly small values are taken more into account. Unfortunately, errors on formerly large values are less considered. Thus, the linear representation of the data shows similar high relative errors for the large values, as shown in Figure 3.

To solve this issue and get acceptable error figures both small and large values, a first approach is to use logarithmic scaling only for the values below the threshold voltage and linear (i.e. no) scaling for the rest. To connect the two different ranges, the normalization is split into two parts, too. The logarithmic part is normalized in the interval $[-1; 0]$ and the linear part is normalized in the interval $[0; 1]$. This split of the normalization range prevents overlapping input ranges (due to different scaling) and prevents non-determinism in the training data (see Figure 4). However, with the two scalings, the NN has problems approximating the function around the threshold voltage (V_{th}) where the differently scaled value ranges connect. From the perspective of the NN, the function features a discontinuity at V_{th} (connecting the two data sets), which leads to different values compared to the SPICE baseline right at V_{th}.

To make the mixed-scaling work and resolve inaccuracies at the corners, the NN could be duplicated with an additional node that decides which NN to query, depending on the input. However, an easier way is to use two different NNs independently and interpolate the results outside of the NNs. This approach can be observed in Figure 5. The interpolation takes place in the green highlighted area (not V_{th}) which is at $V_{gs} = 0.4\,\text{V}$ with an width of $0.1\,\text{V}$. With this configuration, the difference between the combined output of the NNs and the baseline output from SPICE is minimized.

C. Advantages of NN-based Transistor Modeling

Using ML techniques to model the behavior of a transistor has multiple advantages. First and foremost, only exemplary measurement data is needed to train the NN. No details about manufacturing or underlying physics are required. This is a core principle of ML, however, especially helpful for emerging technologies as their underlying physics are often still investigated and not yet fully understood.

a) Domain Knowledge: Many ML applications incorporate domain knowledge in their structures and algorithms. Forcing an NN to follow certain assumptions about the transistor can help improving stability, training time, and correctness. However, falsely applied assumptions can make it impossible for the NN to fit certain data. For example, looking at conventional FinFET, one would assume monotonicity in the I_d-V_d curve. However, novel NC-FinFET experiences a negative DIBL effect that breaks this assumption (shown in Figure 7). A strictly monotonic NN would not be able to learn this behavior properly. Therefore, to provide a generic NN-based transistor model, applicable to multiple technologies, no domain knowledge can be used to augment the NN.

b) Development Time: While the development time of sophisticated compact models is usually in the order of years,

ML training can be done within hours. As soon as early measurements from silicon become available the NN can be trained. However, a certain amount of data is required to achieve meaningful accuracy. TCAD simulations can support measured data, especially when larger data sets under varying conditions (temperature, voltages, etc.) should be trained for. In this work, for simplicity but without loss of generality, we take an existing compact model as the source for training data and validation.

c) Secrecy of Neural Networks: An NN is generally conceived as a black-box function. Even with the weights at hand, the contribution of individual parts to the output is often unclear. In fact, comprehending and tracing the decisions of a NN is a challenge and a research topic in its own [22]. Moreover, the structure of the NN does not reflect the structure of the mathematical equations of the underlying physics. Therefore, even if contributions would be understood, all variables are arbitrary parameters with no traceable relation to the real technology parameters such as geometries or dopant concentrations. Also, unlike in traditional compact modeling, there is no separation of technology (compact model) and calibration (model card) in the NN. The transistor calibration is implicit and indistinguishable from the fundamental technology. This adds an additional layer of abstraction to the NN model. *With all this complexity and abstractions in the NN, the extraction of technology details is at least unfeasible, if not impossible.* This offers great advantages in confidentiality and secrecy with regard to emerging and commercial technologies. NN-based transistor models can be shared without the risk of leaking confidential information or reverse engineering.

D. Disadvantages of NN-based Transistor Modeling

Unfortunately, NN-based transistor models also feature some considerable disadvantages, i.e. cannot fully replace compact models and only act as an intermediate.

a) Scalability: A noteworthy disadvantage of ML is the prediction accuracy outside of the range of the training data. Compact models are built on physical equations which allows them to be scalable beyond available data from silicon [12]. For example, with calibration between 20-50°C, the compact model would provide decent accuracy at 80°C, while the NN-based approach would struggle severely. Therefore, NN-based approaches should be trained for wide ranges to ensure that the inference always occurs within the trained range. Additionally, this highlights why NN-based transistor modeling is just an intermediate before the availability of actual compact models for the technology.

b) Learning Challenges: Although the introduction of linear/log mixed-scaling in section IV-B improved the accuracy of the NN, the logarithmic scaling also created a new issue. Comparing the training effort for data on linear and logarithmic scaling, a big discrepancy can be observed in Figure 6. This is caused by troublesome data in the training set. For $V_{ds} = 0\,\mathrm{V}$, SPICE reports a current of exactly $0\,\mathrm{A}$. Since $\log 0$ is not defined, all values $\leq 10^{-30}$ are set to 10^{-30} during scaling. From a logarithmic point of view, this creates a gap between values with $V_{ds} = 0\,\mathrm{V}$ and $V_{ds} = 0.01\,\mathrm{V}$ which is perceived as a discontinuity by the NN. As a workaround, all data with $V_{ds} = 0\,\mathrm{V}$ could be removed from training and test sets. As we can see in Figure 6, the learning curve improves greatly as soon as these data points are removed.

TABLE I
R^2 SCORES FOR DIFFERENT NNS. INCLUDING n_{FIN} INCREASES TRAINING EFFORT AND THUS, R^2 SCORE FOR A GIVEN TRAINING TIME.

	Scale	Num. of steps			R^2 score of output parameters			
		T	V_{ds}	n_{fin}	I_d	c_{gb}	c_{gs}	c_{gd}
NN 1	lin	24	80	–	0.9999	0.9997	0.9997	0.9993
	log	24	80	–	0.9998	0.9993	0.9998	0.9989
NN 2	lin	24	80	7	0.7863	0.9894	0.9626	0.9834
	log	24	80	7	0.9856	0.9448	0.9265	0.9072
NN 3	lin	6	12	7	0.9996	0.9997	0.9997	0.9998
	log	6	12	7	0.9989	0.9995	0.9996	0.9996

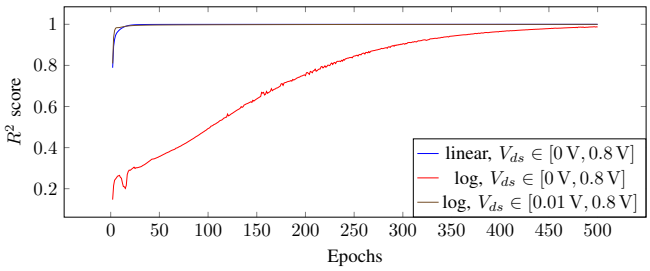


Fig. 6. Learning curves for different scalings. As soon as $V_{ds} = 0\,\mathrm{V}$ is removed from the logarithmic data set, the learning curve improves significantly.

E. Inference Accuracy and Training Time

Training data is generated by sweeping input parameters with a certain step size in SPICE simulations. All input parameter permutations and their corresponding outputs are collected in CSV files to serve as training and validation data sets for the NN. After 500 epochs of training, the R^2 scores for the training set, test set, and complete data set validation are 0.999772, 0.999764, and 0.999769 respectively. The R^2 score of the training set is very close to 1, meaning that the NN was able to match the training data excellently. Likewise, the R^2 score of the test set is only slightly lower than the R^2 score of the training set which indicates good generalization of the NN to the learned function (no overfitting).

Table I shows the R^2 score of different NNs after $48\,\mathrm{h}$ of training. Depending on the number of input and output parameters, the required training time changes. Without the number of fins in the FinFET (n_{fin}) configurable, NN 1 reaches an R^2 score of 0.999 for all output parameters with both linear and logarithmic scaling.

But when adding number of fins (n_{fin}) as an additional input parameter, the precision significantly drops, most notably for I_d with the linear scaling and an R^2 score of 0.786. The reason behind this is the large amount of input data, which, in this case, contains 24 different temperatures (20 °C

TABLE II
RELATIVE ERROR OF TRANSISTOR CHARACTERISTIC QUANTILES DEPENDING ON V_{ds}, V_{gs}, AND TEMPERATURE STEPS. HIGHLIGHT MARKS MINIMUM DATA POINTS SATISFYING THE ERROR LIMIT OF 5%.

V_{ds} steps	V_{gs} steps	Temp. steps	Data points	Q5 I_{off}	Q95 I_{off}	Q5 I_{on}	Q95 I_{on}	Q5 V_{th}	Q95 V_{th}	Q95 SS
80	12	6	5760	-5.02%	1.63%	-0.30%	0.51%	-0.50%	0.61%	4.11%
80	9	6	4320	-4.70%	2.07%	-0.41%	0.48%	-0.45%	0.75%	3.67%
80	7	6	3360	-4.92%	1.77%	-0.26%	0.46%	-0.06%	1.65%	4.35%
80	5	6	2400	-3.82%	1.86%	-0.38%	0.44%	-0.31%	2.14%	5.28%
17	17	6	1734	-5.43%	1.68%	-0.39%	0.48%	-0.73%	0.47%	4.30%
17	12	6	1224	-4.88%	1.67%	-0.22%	0.50%	-0.58%	0.49%	4.00%
17	9	6	918	-4.60%	1.77%	-0.43%	0.49%	-0.55%	0.76%	3.46%
12	17	6	1224	-6.62%	1.61%	-0.29%	0.62%	-0.84%	0.65%	5.11%
12	12	6	864	-6.91%	1.44%	-0.29%	0.52%	-0.72%	0.62%	5.92%
12	9	6	648	-5.62%	1.99%	-0.40%	0.56%	-0.63%	0.74%	4.56%

- 135 °C), 80 different V_{ds} steps (0.01 V - 0.8 V), 81 different V_{gs} steps (0 V - 0.8 V), and 7 different number of fins (1 - 7). This accumulates to a total number of 1,088,640 data points, resulting in a much longer training time for a single epoch.

While longer training times can be overcome with more processing power or just waiting longer, longer training cannot guarantee continuous improvement (i.e., converges to a low R^2 score). Alternatively, the training data sets can be trimmed. In our case, there is plenty of training data and a reduction shows no negative impact. NN 3 in Table I shows the results of using only 6 temperature steps and 12 V_{ds} steps for the training. With this optimization, the NN is capable to learn all 4 output parameters depending on all 4 input parameters within 48 h of training time.

F. Traditional Fitness compared to Transistor Metric Fitness

The R^2 score metric provides an impression of the accuracy of an NN. However, inaccurate regions (outliers) might be overshadowed by good overall accuracy. Typically, this is not an issue, as overall accuracy is key. However, for transistors, certain transistor characteristics are critical for the evaluation of the transistor in circuit simulation. For example, the leakage current at $V_{gs} = 0V$ and the on-current at $V_{gs} = V_{DD}$ are ends of the I-V curve, but very important since digital circuits operate a majority of the time in these two extremes. Therefore, inaccurate modeling at these locations might severely alter the circuit simulation results of digital circuits and as such, an R^2 score infers little about circuit simulation accuracy of a transistor model. For this reason, we propose to determine key transistor characteristics instead and use this as our metric to decide after how many training epochs a sufficient accuracy is reached. The comparison between R^2 score and transistor metric fitness is shown in in Table III.

For the training set, temperature ranges from 20 °C to 135 °C, V_{ds} from 0.01 V to 0.8 V, and V_{gs} from 0 V to 0.8 V. For the following tests, we use an enlarged test set with generated data at a smaller step size. As NNs tend to become more inaccurate at the edges of the training data, we narrow the temperature range slightly. For the test set, the ranges change for the temperature from 25 °C to 130 °C, and for V_{ds} from 0.05 V to 0.8 V. Independent of the number of data points used in the training, the validation uses a temperature step size of 1 °C and a voltage step size of 0.01 V.

TABLE III
COMPARISON OF R^2 SCORE AND TRANSISTOR SPECIFIC CHARACTERISTICS DEPENDING ON THE NUMBER OF TRAINED EPOCHS.

Epochs	50	100	200	300	400
R^2 linear	0.99948	0.99968	0.99981	0.99989	0.99991
R^2 log	0.99666	0.99795	0.99896	0.99944	0.99968
Q5 I_{off}	-1.47%	-9.69%	-8.20%	-6.14%	-4.62%
Q95 I_{off}	3.01%	2.71%	2.62%	2.59%	2.93%
Q5 I_{on}	-1.93%	-1.68%	-1.28%	-1.14%	-0.96%
Q95 I_{on}	1.95%	1.80%	1.46%	1.30%	1.09%
Q5 V_{th}	-0.64%	-0.76%	-0.72%	-0.61%	-0.59%
Q95 V_{th}	2.12%	1.57%	1.17%	0.92%	0.67%
Q95 SS	1.25%	9.49%	7.78%	5.85%	4.45%

TABLE IV
DEVELOPMENT OF NN ACCURACY FOR NC-FINFET. COMPARED TO TABLE III, NC-FINFET STARTS WITH HIGHER ERRORS BUT EVENTUALLY CONVERGES TO A SIMILAR PRECISION OF FINFET.

Epochs	50	100	200	300	400
R^2 linear	0.99876	0.99956	0.99984	0.99988	0.99992
R^2 log	0.99552	0.99743	0.99879	0.99937	0.99960
Q5 I_{off}	-1.56%	-0.86%	-7.85%	-5.68%	-4.69%
Q95 I_{off}	4.83%	3.17%	3.22%	2.52%	2.67%
Q5 I_{on}	-5.90%	-4.35%	-2.31%	-1.86%	-1.43%
Q95 I_{on}	3.13%	1.70%	1.79%	1.40%	1.11%
Q5 V_{th}	-1.10%	-0.67%	-0.79%	-0.70%	-0.69%
Q95 V_{th}	3.95%	3.28%	1.94%	1.44%	1.18%
Q95 SS	2.11%	0.55%	7.22%	5.12%	4.44%

For each metric, the relative errors across the test set are calculated. Then, the error values are sorted so that a minimum error at the 5%-quantile (Q5) and a maximum error at the 95%-quantile (Q95) can be determined. This procedure is slightly different for SS as it can be measured at multiple points below the threshold voltage. We measure the SS relative error at multiple points and calculate the average of the absolutes, so that negative and positive relative errors do not cancel each other out. In consequence, there is only a Q95 error for SS. All metrics, as well as the R^2 score are shown Table III.

For our evaluation, we define an error of $\leq 5\%$ *at each 5%/95%-quantile to be acceptable.* However, please note, that

other values could have been chosen and internal experiments have shown that longer training times or more training data are sufficient for more stringent accuracy constraints.

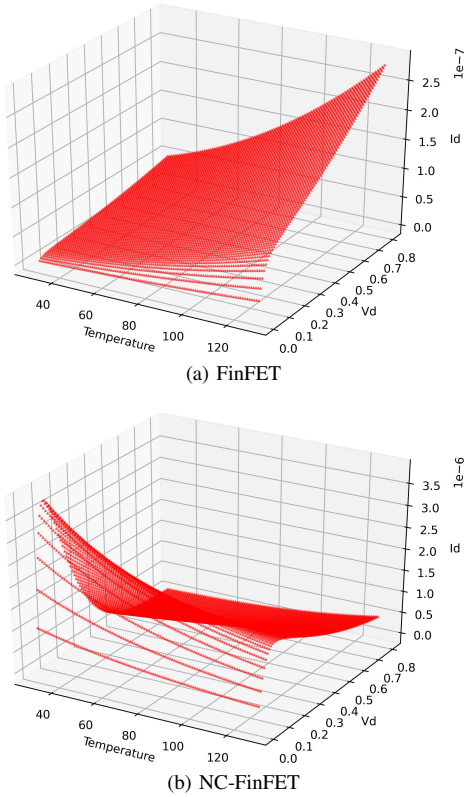
All transistor characteristic results in Table III are created by using two separate neural networks with different scaling where the results are interpolated as described in section IV-B. After 50 epochs, the R^2 score is already > 0.99, however, Q5 I_{off} and Q95 SS still show an error of more than 11%, violating our tolerance. *This shows that the R^2 score of the NN is not sufficient to judge the accuracy of important transistor characteristics.* For the conventional FinFET device, the NN needs 400 epochs of training to reach the demanded precision.

G. Early Evaluation with Limited Data

To be able to reach an error $\leq 5\%$ for all transistor characteristics, a certain amount of training data is needed. The required amount depends on the number of parameters and the number of measurement points for each parameter (i.e., in our case, sweeping steps in SPICE). Table II explores the relative error of each transistor characteristic depending on V_{ds} and V_{gs} steps. For the sake of space in this manuscript, a minimum of 6 temperature steps has been chosen as a suitable minimum based on preceding experiments. Table II shows that for a NN transistor model with temperature dependency, around 900 data points are sufficient to reach our tolerance. Looking at the step sizes, *we can say that around 17 $I_d - V_{gs}$ curves with 9 data points each are enough to start developing an early model that satisfies our error constraint.* This number of points (measurements) is feasible to acquire even for early prototypes. With our selection of step sizes and data ranges, we are able to minimize the training effort to these 900 points.

H. Modeling NC-FinFET with NN-based Transistor Models

To evaluate if our NN can also fit other, previously unseen transistors, we repeat the experiment in section IV-F using an NC-FinFET compact model [6] instead of BSIM-CMG. Comparing conventional FinFET an NC-FinFET, a stark difference in the behavior becomes apparent when looking at the transfer curves in Figure 7. We can observe that compared to FinFET, NC-FinFET shows a clearly different, non-monotonic response in the V_d axis. This additional complexity is added by the negative DIBL effect caused by the ferroelectricity of the gate and suggests an increased difficulty in fitting the behavior. Table IV shows the progress of learning NC-FinFET. When comparing these results with the FinFET results in Table III, we can see that (especially in the first 200 epochs) almost all metrics show a worse error. Some stand out, e.g., Q5 I_{on} with an error of -5.9% after 50 trained epochs. With an increasing number of training epochs, these errors are getting closer to the errors of the FinFET measurements. The desired precision of $\leq 5\%$ relative error for each Q5/Q95 metric is reached after 400 epochs. *Eventually, our NN is able to apply to NC-FinFET in a similar quality as to FinFET.* This hints to the possibility, that NN-based transistor models are indeed generic and thus that additional emerging technologies can be modeled with this approach.



(a) FinFET

(b) NC-FinFET

Fig. 7. Comparison of FinFET and NC-FinFET transfer curves at $V_{gs} = 0.15\,\text{V}$. The negative DIBL effect in NC-FinFET completely changes the shape of the I-V curve.

V. Conclusion

In this work, we presented an ML approach that is able to learn FinFET transfer curves with high accuracy specific to selected transistor characteristics. Without any changes to the NN, it is also able to learn emerging NC-FinFET, although extra complexity is added due to the negative DIBL effect. This hints at the chance that more emerging technologies can be estimated with ML techniques, serving as an intermediate solution until sophisticated compact models are developed.

An ML solution can not only be developed in shorter time frames than the traditional compact model, it is also easier to share with others. Due to the black-box characteristics of the NN, the extraction of technology details is impractical or even impossible. With these benefits at hand, ML provides a great opportunity to speed-up technology development, achieving faster time-to-market, and increase customer acceptance due to easier access.

Acknowledgment

The work is supported in part by the U.S. Office of Naval Research (ONR) under Grant N00014-18-1-2672.

Authors would like to thank Yogesh S. Chauhan from the Indian Institute of Technology Kanpur (IITK), Kanpur, India and his research group as well as Girish Pahwa from University of California, Berkeley for their valuable support in NCFET device modeling and analysis.

REFERENCES

[1] S. Venugopalan, N. Paydavosi, J. Duarte, D. Lu, S. Khandelwal, C.-H. Lin, M. Dunga, S. Yao, A. Niknejad, and C. Hu, "Bsim-cmg 110," 2016. [Online]. Available: http://bsim.berkeley.edu/models/bsimcmg/

[2] S.-Y. Wu, C. Lin, M. Chiang, J. Liaw, J. Cheng, S. Yang, C. Tsai, P. Chen, T. Miyashita, C. Chang *et al.*, "A 7nm cmos platform technology featuring 4 th generation finfet transistors with a 0.027 um 2 high density 6-t sram cell for mobile soc applications," in *2016 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2016, pp. 2–6.

[3] Z. Zhang, R. Wang, C. Chen, Q. Huang, Y. Wang, C. Hu, D. Wu, J. Wang, and R. Huang, "New-generation design-technology co-optimization (dtco): Machine-learning assisted modeling framework," in *2019 Silicon Nanoelectronics Workshop (SNW)*. IEEE, 2019, pp. 1–2.

[4] D. Ha, C. Yang, J. Lee, S. Lee, S. Lee, K.-I. Seo, H. Oh, E. Hwang, S. Do, S. Park *et al.*, "Highly manufacturable 7nm finfet technology featuring euv lithography for low power and high performance applications," in *2017 Symposium on VLSI Technology*. IEEE, 2017, pp. T68–T69.

[5] H. Amrouch, S. Salamin, G. Pahwa, A. D. Gaidhane, J. Henkel, and Y. S. Chauhan, "Unveiling the impact of ir-drop on performance gain in ncfet-based processors," *IEEE Transactions on Electron Devices*, vol. 66, no. 7, pp. 3215–3223, 2019.

[6] G. Pahwa, T. Dutta, A. Agarwal, S. Khandelwal, S. Salahuddin, C. Hu, and Y. S. Chauhan, "Analysis and compact modeling of negative capacitance transistor with high on-current and negative output differential resistance—part i: Model description," *IEEE Transactions on Electron Devices*, vol. 63, no. 12, pp. 4981–4985, Dec 2016.

[7] P. B. L. Meijer, *Neural network applications in device and subcircuit modelling for circuit simulation*. Philips Electronics, 1996.

[8] L. Zhang and M. Chan, "Artificial neural network design for compact modeling of generic transistors," *Journal of Computational Electronics*, vol. 16, no. 3, pp. 825–832, 2017.

[9] M. Li, O. İrsoy, C. Cardie, and H. G. Xing, "Physics-inspired neural networks for efficient device compact modeling," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 2, pp. 44–49, 2016.

[10] K. Lamamra and S. Berrah, "Modeling of mosfet transistor by mlp neural networks," in *International Conference on Electrical Engineering and Control Applications*. Springer, 2016, pp. 407–415.

[19] H. Amrouch, G. Pahwa, A. D. Gaidhane, J. Henkel, and Y. S. Chauhan, "Negative capacitance transistor to address the fundamental limitations in technology scaling: Processor performance," *IEEE Access*, vol. 6, pp. 52 754–52 765, 2018.

[11] T. Blaesi, "Spice model generation by machine learning," 2019. [Online]. Available: https://silvaco-blog.com/circuit_simulation_modeling/spice-model-generation-by-machine-learning-html/

[12] M. V. Dunga, C.-H. Lin, A. M. Niknejad, and C. Hu, "Bsim-cmg: A compact model for multi-gate transistors," in *FinFETs and Other Multi-Gate Transistors*. Springer, 2008, pp. 113–153.

[13] Y. S. Chauhan, S. Venugopalan, M. A. Karim, S. Khandelwal, N. Paydavosi, P. Thakur, A. M. Niknejad, and C. C. Hu, "Bsim—industry standard compact mosfet models," in *2012 Proceedings of the European Solid-State Device Research Conference (ESSDERC)*. IEEE, 2012, pp. 46–49.

[14] Y.-K. Choi, N. Lindert, P. Xuan, S. Tang, D. Ha, E. Anderson, T.-J. King, J. Bokor, and C. Hu, "Sub-20 nm cmos finfet technologies," in *International Electron Devices Meeting. Technical Digest (Cat. No. 01CH37224)*. IEEE, 2001, pp. 19–1.

[15] B. Yu, L. Chang, S. Ahmed, H. Wang, S. Bell, C.-Y. Yang, C. Tabery, C. Ho, Q. Xiang, T.-J. King *et al.*, "Finfet scaling to 10 nm gate length," in *Digest. International Electron Devices Meeting,*. IEEE, 2002, pp. 251–254.

[16] N. Paydavosi, S. Venugopalan, Y. S. Chauhan, J. P. Duarte, S. Jandhyala, A. M. Niknejad, and C. C. Hu, "Bsim—spice models enable finfet and utb ic designs," *IEEE Access*, vol. 1, pp. 201–215, 2013.

[17] J. Appenzeller, Y.-M. Lin, J. Knoch, and P. Avouris, "Band-to-band tunneling in carbon nanotube field-effect transistors," *Physical review letters*, vol. 93, no. 19, p. 196805, 2004.

[18] S. Salahuddin and S. Datta, "Use of negative capacitance to provide voltage amplification for low power nanoscale devices," *Nano letters*, vol. 8, no. 2, pp. 405–410, 2008.

[20] H. Amrouch, G. Pahwa, A. D. Gaidhane, C. K. Dabhi, F. Klemme, O. Prakash, and Y. S. Chauhan, "Impact of variability on processor performance in negative capacitance finfet technology," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2020.

[21] S. Mishra, H. Amrouch, J. Joe, C. K. Dabhi, K. Thakor, Y. S. Chauhan, J. Henkel, and S. Mahapatra, "A simulation study of nbti impact on 14-nm node finfet technology for logic applications: Device degradation to circuit-level interaction," *IEEE Transactions on Electron Devices*, vol. 66, no. 1, pp. 271–278, 2018.

[22] V. Buhrmester, D. Münch, and M. Arens, "Analysis of explainers of black box deep neural networks for computer vision: A survey," 2019.

[23] S. Pagani, S. M. PD, A. Jantsch, and J. Henkel, "Machine learning for power, energy, and thermal management on multi-core processors: A survey," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.