A Fast TCAD-based Methodology for Variation Analysis of Emerging Nano-Devices

Hassan Ghasemzadeh Mohammadi*, Pierre-Emmanuel Gaillardon*, Majid Yazdani†, Giovanni De Micheli*
Integrated Systems Laboratory*, IDIAP Laboratory†
EPFL
Lausanne, Switzerland
Email: hassan.ghasemzadeh@epfl.ch

***Abstract*—Variability analysis of nanoscale transistors and circuits is emerging as a necessity at advanced technology nodes. *Technology Computer Aided Design* (TCAD) tools are powerful ways to get an accurate insight of *Process Variations* (PV). However, obtaining both fast and accurate device simulations is impractical with current TCAD solvers. In this paper, we propose an automated output prediction method suited for fast PV analysis. Coupled with TCAD simulations, our methodology can substantially reduce the time complexity and cost of variation analysis for emerging technologies. We overcome the simulation obstacles and preserve accuracy, using a neural network based regression to predict the output of individual process simulations. Experiments indicate that, after the training process, the proposed methodology effectively accelerate TCAD-based PV simulations close to compact-model-based simulations. Therefore, the methodology can be an excellent opportunity in enabling extensive statistical simulations such as Monte-Carlo for emerging nano-devices.**

I. Introduction

The aggressive downscaling trend in CMOS technology is expected to reach its fundamental physical limits in the near future [1]. In order to discover free rooms for further performance and functionality improvements, a number of emerging nano-devices, e.g., *Doped/Schottky Barrier Silicon NanoWire* FETs (SiNWFETs), various types of Fin-Shaped FETs, and *Carbon Nanotube* FETs (CNTFETs) have been proposed by research community [2]–[5]. Due to their novel and complex geometries, they introduce new challenges such as additional sources of variation which makes the statistical variation analysis challenging [7].

Process Variations (PVs) and fabrication defects are largely dominating the performance and reliability in advanced technology nodes. On the one hand, *Technology Computer Aided Design* (TCAD) simulation can be exploited for accurate PV analysis of emerging nano-devices because it virtually provides an insight view into the physics of semiconductors. Numerical simulations of TCAD are by far too slow to give timely answers to many questions arising from variation of device parameters or manufacturing fluctuations. On the other hand, the use of compact models, in order to get fast prediction of the device electrical behavior does not help anymore. Indeed, developing and verifying a dependable compact model for emerging nano-devices is a costly procedure, requiring mature device fabrication, test, and measurement.

With the lack of mature devices and compact-models availabilities, TCAD appears as the most appropriate candidate for PV analysis of emerging technologies. However, the simulation time requirements of potential TCAD-based PV analysis of nano-devices critically depends on the complexity of the physical models, the size of parameter space, and the underlying optimization algorithms [8], [9]. In addition, convergence of computations for small feature size also represents a key limitation for commercial TCAD softwares. Current TCAD simulators benefit by different types of optimization techniques at the various level of simulation [10]. Nevertheless, impractical TCAD simulation time for nano-devices remains an important barrier, preventing their use for fast PV analysis.

In this paper, we propose a novel methodology intended to effectively speed up the variation analysis for nano emerging devices and circuits. A prediction technique is proposed to estimate the output of TCAD simulation. The technique is based on a "*Feed Forward Neural Network*" (FFNN) and provides a multivariate non-linear regression method for faster transistor I-V prediction as compared to TCAD simulations. This technique learns the fundamental relations between a device parameters and its functionality. In order to optimize the I-V curves fitting, we modify the FFNN structure through adding a number of extra nodes called *Control Nodes*. These nodes modify the weights of the network in a way that the predicted I-Vs have minimum error value corresponding *On Current* (I_{on}), *Off Current* (I_{off}), *Threshold Voltage* (V_{Th}), and *Sub-threshold Slope* (S). This methodology enables us to produce a large enough data set of a target device in presence of variations and handles the convergence issues related to complex numerical models. To evaluate the capability of the developed methodology, *Double-Gate Silicon Nanowire* FETs (DG-SiNWFETs) were used as a target device. Simulation results proved that the proposed methodology makes the time complexity of device simulation comparable to compact-model simulation. After training, the average runtime for I-V estimation of a DG-SiNWFET is 0.12 ms which is considerably lower than TCAD simulation execution time.

The organization of this paper is as follows. Section II describes the motivation and background. Section III presents the proposed methodology for fast variation analysis. Section IV provides the simulation results, and finally Section V concludes the paper.

II. Background and Motivation

This section focuses on the hurdles encountered during PV analysis of emerging nano-devices. We discuss the time

978-1-4799-1585-9/13/$31.00 ©2013 IEEE

complexity issues of TCAD for the purpose of PV analysis and briefly review the related work for TCAD optimization.

A. Problem

Statistical information for transistors can be obtained by the three following ways: device fabrication, TCAD simulation, or compact-model based simulation. One important issue of nano-devices fabrication, in addition to the lack of maturity, resides in the increase of processing complexity. Moreover, the contribution of each device parameter to the functionality variation cannot be determined through cross-section analysis. On the other side, reliance on device compact models is a hardly feasible solution since the development of accurate models mainly depends on tests and extensive measurements on mature devices.

Thanks to their precise physical models, TCAD provides comprehensive information about devices behavior and alleviates expensive fabrication and inspection process. Hence, TCAD simulations are an invaluable solution for the evaluation and minimization of PV impacts, thus enabling technologists to early identify the main sources of parametric yield loss in manufacturing. However, it suffers from several significant challenges such as computational complexity of the DC/AC device simulations, and convergence problem for complex models. Thus, device level *Monte Carlo* analysis of nano-structures using TCAD also will result in prohibitive time complexity. Therefore, a fast methodology for PV analysis is required, especially when new processes or devices are introduced. In the following, a novel methodology which can be integrated with TCAD to enhance simulation performance of PV analysis is introduced.

B. Previous Work

TCAD simulation are extensively exploited for evaluating the fabrication process and the electrical behavior of semiconductors [12]. During the recent years, TCAD has been used for the simulation of emerging nano-devices such as *Multigate FETs* [14] and CMOS/Flash devices [13]. Moving towards deep submicron technologies, the process simulation and physical model of devices becomes more complex. Hence, optimization technique have been utilized for improving TCAD performance.

Process simulation requires geometrical information and mesh generation for the model under test. Moreover, it accurately estimates structural layers and active dopant distributions at the end of a procedure run. Techniques to enhance process simulation precision and complexity have been investigated in [16], [17].

Physical models are based on complex equations, describing the semiconductor conduction mechanisms. In order to reach desirable accuracy at advanced technology nodes, the complexity of model keeps increasing. Due to computational complexity, several optimization techniques such as cluster computation and gradient-based optimizations have been proposed [15]. However, the downscaling of semiconductors are correlated to increasingly complex models and therefore the efficiency of the mentioned techniques are counteracted.

Most recently, [18] proposed a methodology in which redundant physics computations are removed for common parts of the model and accordingly repetitive 3-D simulations are discarded for various parts of a circuits/layouts. This methodology accelerates TCAD simulations and makes it reachable for analysis of small circuits and logic blocks. However, it is still very limited for large layouts or for the repetitive simulations in the context of PV analysis. For example, one TCAD simulation run of a 6T SRAM cell in FinFET technology takes more than 17 hours in this methodology.

III. Learning-based Methodology for Fast Process Variation Analysis of Emerging Devices

In this section, we introduce the proposed methodology and its key component: a prediction module. The prediction module is used, after training, to predict the device variations instead of using TCAD simulator, thereby improving significantly the simulation process speed.

A. PV Analysis Using Neural Networks

To model the underlying relations of multivariate parameters in TCAD simulations, precise definition of the functional relations among the variables is required. These functional relations are based on solid-state physics and quantum-mechanics equations. Thus, the TCAD simulation for only a single device is very slow ($O(hours)$) [18], and also the simulation does not converge for a non-negligible number of data points.

To overcome this issue, our method uses a predictor which learns how to approximate the simulator results. The predictor is used for data generation and deal with the missing data resulting from failed simulation runs. Without loss of generality, we will focus on single device simulation in this work. Circuit-level modeling using this method is out of this paper's scope.

Fig. 1 represents the general flow of the mentioned methodology. First, our methodology considers the TCAD model for the target device, using the nominal values of the device parameters. This set of parameters is determined to precisely reproduce the normal I-V characteristics of the target device. In the next step, the profile of model parameter variations such as gate length, channel length, and oxide thickness, is applied to the TCAD model generator, and then a large data set of various device models is created. Only a small subset of these TCAD models is randomly selected and simulated. The cardinality of this subset represents the minimum necessary TCAD simulations for training the predictor. Predictor uses the obtained I-V curves as a ground truth and learns the underlying relations between the device parameters and its functionality. In order to speed up the estimation of I-V curves, the prediction module uses a regression technique to prevent repetitive time-consuming TCAD simulations. After training, the TCAD simulator can be replaced by the predictor, in order to speed up the PV simulations. In the following subsections, we explain the predictor structure and the training algorithm.

B. Neural Network Structure

Among nonlinear regression techniques, *Feed-Forward Neural Network* (FFNN) [19] is mostly used in practice. Compared to its competitors, such as *Super Vector Regression* (SVR) [21], FFNN has smaller number of hyper parameters

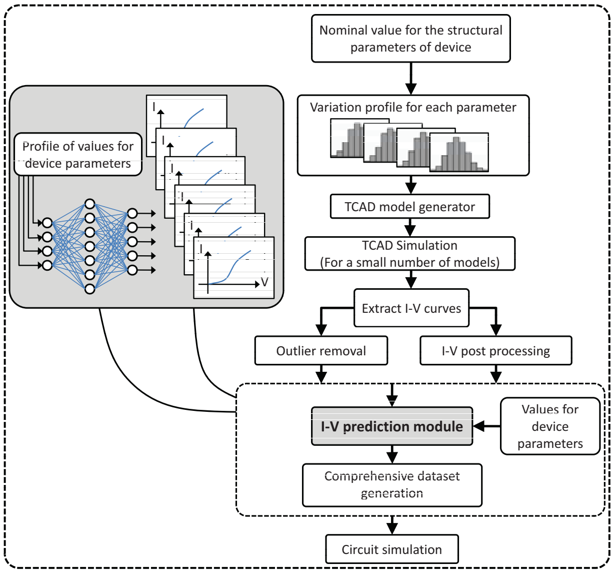
Nominal value for the structural parameters of device
Variation profile for each parameter
TCAD model generator
TCAD Simulation (For a small number of models)
Extract I-V curves
Outlier removal
I-V post processing
I-V prediction module
Values for device parameters
Comprehensive dataset generation
Circuit simulation
Profile of values for device parameters
I
V

Fig. 1: Flowchart of the proposed methodology.

and smaller model-size. Moreover, FFNN is fast while SVR becomes very slow when many support vectors are created. Although we utilize FFNN for prediction purpose here, it should be noted that other non-linear regression methods can also be applied to estimate the $I-V$ values. The performance comparison for these methods is out of this paper's scope.

Fig. 2 provides a graphical representation for the proposed three layer FFNN. The set of device parameters such as *Channel Length*, *Oxide Thickness*, and *Gate Length* makes the FFNN input ($\mathbf{x} = \{x_1, x_2, \cdots, x_J\}$). The set of $h = 1, 2, \cdots, H$ also represents the hidden layer nodes. The FFNN output vector ($\hat{\mathbf{i}} = \{\hat{i_1}, \hat{i_2}, \cdots, \hat{i_V}\}$) approximates the observed values of the I-V curves (sampled *Drain Current* I_d set) for the given input. An estimated I_d value ($\hat{\mathbf{i}}_v$) for the given input is calculated by:

$$\hat{i}_v(\mathbf{x}) = b'_v + \sum_{h=1}^{H} w'_{hv} \cdot \phi(b_h + \sum_{j=1}^{J} x_j \cdot w_{jh})$$

$\{b', b\}$ and $\{\mathbf{W}', \mathbf{W}\}$ are the network parameters and are called bias set and weight set respectively. These parameters are learned during the training step. $\mathbf{W}$ is a $J \times H$ matrix which transforms the input data into the hidden space, and $\mathbf{W}'$ is a $H \times V$ matrix which transforms data from the hidden space to the output space. The activation function ϕ is conventionally chosen as *Sigmoid logistic function* ($\frac{1}{1+e^{-x}}$) or *hyperbolic tangent* ($tanh(x)$) in non-linear regression problems [23]. It is known that FFNN is a universal approximator, which is expressed in the following theorem reported from [25].

Theorem 1: If $\phi(.)$ be a bounded, and monotonically-increasing continuous function and I_m denote the m-dimensional unit hypercube $[0, 1]^m$. The space of continuous functions on I_m is denoted by $C(I_m)$. Then, given any function $f \in C(I_m)$ and $\varepsilon > 0$, there exist an integer N and real constants $\alpha_i, b_i \in R$ and $w_i \in R^m$, where $1 \leq i \leq N$ such that: $F(x) = \sum_{i=1}^{N} \alpha_i \phi(w_i^T x + b_i)$ and we have $|F(x) - f(x)| < \varepsilon$.

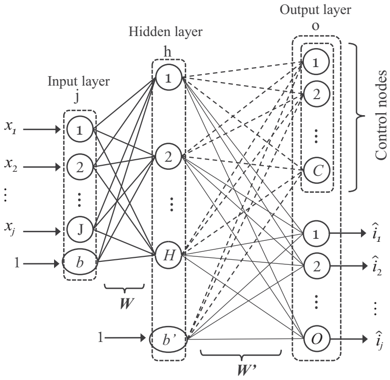

Fig. 2: Graphical representation of the FFNN for I-V curve regression

The hidden space dimension, H, determines the learning capability of the network. An FFNN with a sufficiently large single hidden layer can approximate the function showing the relations among the parameters and their impacts on device functionality. It should be noted that, the excessive number of hidden nodes makes networks more prone to learn noise and memorize the training data, therefore the prediction ability of the network is reduced. In the next section, we discuss how to determine the number of hidden units practically.

Control Nodes: Information obtained from I-V curves are not equally valuable and a number of key statistical information such as I_{on}, I_{off}, V_{Th}, and S are more important. Therefore, we introduce a set of control nodes within the output layer in order to approximate specifically these valuable information. Note that this information is already presented in the output vector, but we distinctively add control nodes to insist on the accuracy of I-V estimation and modify the degree of fitting. By utilizing control nodes, the weights of the nodes which contribute to estimate this information are modified in a way that they have more impact on the regression. Control nodes prevent model from overfitting and reduce the impact of flat data on the I-V prediction. Our experimental results verify that control nodes enhance the performance of regression.

C. Training of the Neural Network

In order to make our ground truth for training purpose, the TCAD generated I-V curves are discretized. The *Gate Voltage* (V_G) value is limited to $[0, V_{dd}]$ along a TCAD simulation. This interval is divided to a number of equidistant subintervals, and then current values are sampled at the end point of these subintervals. Since the V_G values are alike for the whole I-V curves, the correspondent I_d values have to be estimated through the prediction unit. Let the set $Tr = (\mathbf{x}_1, \mathbf{i}_1), (\mathbf{x}_2, \mathbf{i}_2) \cdots (\mathbf{x}_N, \mathbf{i}_N)$ represents the training set which N is the number of simulations used in the training set. The set of the device input parameters is $\mathbf{x}_k$ and $\mathbf{i}_k$ is the observed values of the sampled I_d set for the given input. We minimize the loss function which is defined as the least squares error between the approximation values and the observed values, over the training set:

$$Min\ L = \sum_k (\mathbf{i}(\mathbf{x}_k) - \hat{\mathbf{i}}(\mathbf{x}_k))^2 + (\mathbf{w}^T(\mathbf{c}(\mathbf{x}_k) - \hat{\mathbf{c}}(\mathbf{x}_k)))^2$$

where $\mathbf{c}(\mathbf{x}_k)$ represents the vector of control nodes in output layer and $\mathbf{w}$ is a vector of size C which shows the importance of each control nodes. The backpropagation training algorithm plus *Levenberg-Marquardt* algorithm [24] is used to find the weights of the network.

Cross validation is a practical method to find the number of hidden units and prevent overfitting. To perform cross validation, a part of data set is randomly selected to test the generalization of the network during learning process. The error value of cross validation can be used a criterion for the number of training cycles and the network size. As the network size increases, the error value on the training set becomes smaller. However, the error value on the test set reach to a minimum value for the optimized network size and training iterations. In the following section, we study the experimental results of the cross validation.

IV. Result and Discussion

In this section, the experimental results are presented. First, the setup of the experiments is explained and, then, the simulation results are discussed.

A. Setup of Experiments

As a case study, *Double-Gate Silicon Nanowire* FETs (DG-SiNWFETs) [5] is selected among various emerging devices as a target model in our experiments. This device represents a natural evolution of FinFET structure and can be dynamically configured between *n*- and *p*-type through an additional terminal, called *Polarity Gate* (PG). In-field reconfiguration can be used for compact implementation of XOR based circuits [6]. Fig. 3 shows the structure of the DG-SiNWFET as used in the TCAD model, while Table I presents the main geometrical parameters of device and their nominal values.

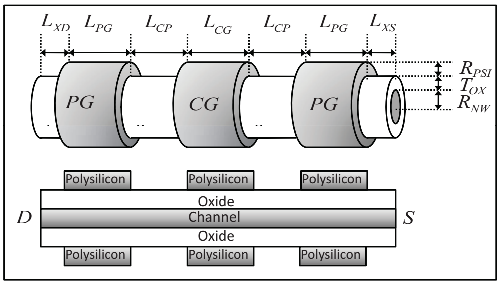

Fig. 3: 3-D TCAD structure of the DG-SiNWFET with all related geometrical parameters.

For the sake of experiments, a 30% variation along normal distribution is used for each geometrical parameters ($\sigma = 30\%$), in order to show the impact of PVs involved in the device fabrication. In our experiments, *Sentaurus* [11] was used as the TCAD simulator. As a case study, 2300 TCAD simulations were performed to provide enough statistical information, and then analytical metrics such as the on-current (I_{on}), the off-current (I_{off}), the threshold voltage (V_{Th}) and the sub-threshold slope (S) were extracted through I-V curves post processing. Fig. 5 illustrates the statistical distribution of I_{on}, I_{off}, and V_{Th} for DG-SiNWFETs. In order to find the minimum number of data points for training, we also trained the predictor with the various size of train set. Fig. 8 depicts the training performance when the size of train set is increases. Our study revealed that almost 600 input data is enough for training the predictor.

The proposed FFNN is implemented in *MATLAB*. In our experiments, all the input vectors are applied to the FFNN in each iteration which is called epoch and then the estimation error is computed. The new values of the weights are computed by applying gradient descent on the error function. The learning method utilizes *Mean Square Error* (MSE) error function. In order to validate the proposed prediction I-V module, we divided the data set, i.e., the outputs of TCAD simulations into three parts: 75% for FFNN training, 15% for validation, and 15% for test.

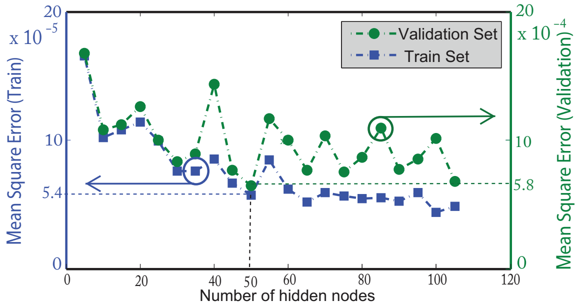

Fig. 4: Mean square error of train and test sets for networks with various number of hidden nodes

Before the learning process, we need to determine the hyper parameters of the model such as error function, activation function and the number of neurons in hidden layer. The selection of the first two hyper parameters was discussed before. In order to find the optimum number of nodes in the hidden layer, we performed the following procedure. The number of hidden nodes is increased until the performance of the FFNN on the test set starts to decrease. Fig. 4 depicts the error value of the network on the test and train sets. As shown in the figure, the minimum error on the test set is achieved when hidden layer has 50 nodes. The figure clearly depict that how the performance of the larger networks are degraded on the validation set.

B. Experimental Results

In the following, the simulation results of the proposed methodology are represented. Precision and time complexity are the main aspects of our study.

1) Precision of Prediction: To show that the control nodes can enhance the results of regression and prevent overfitting over the training set, we built two separate FFNNs: one with control nodes and the other one without control nodes. Fig. 6 illustrates the error distribution for the two FFNNs. The best performance of the FFNN with control nodes is bounded to the MSE of 4.3×10^{-4}, while the ordinary FFNN reaches the error bound of 6.0×10^{-4}. Thus, the addition of control nodes, controls the error bound of estimator.

Fig. 7 depicts a sample of I-V regression for the ordinary and proposed FFNNs. The error value of V_{Th} and I_{on} are much smaller when control nodes are added to FFNN. Therefore,

TABLE I: DG-SiNWFET geometrical parameters

Parameters	L_{CG}	L_{CP}	L_{PG}	L_{XD}	L_{XS}	R_{NW}	R_{PSI}	T_{OX}
	Control Gate Length	Gate Spacer Length	Polarity Gate Length	Drain/Pillar Extension Length	Source/Pillar Extension Length	Radius of NW	Polysilicon Thickness	Oxide Thickness
Nominal value (nm)	22	18	22	2.5	2.5	7.5	2	12

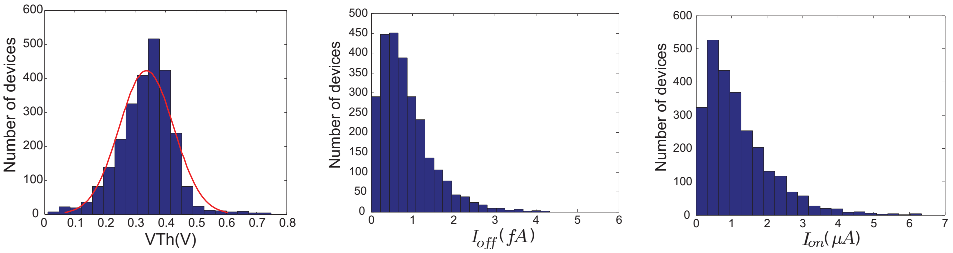

Fig. 5: Distribution of V_{Th}, I_{off}, and I_{on} for DG-SiNWFET (σ=30% for structural parameters). Only the variation of V_{Th} follows a Gaussian distribution.

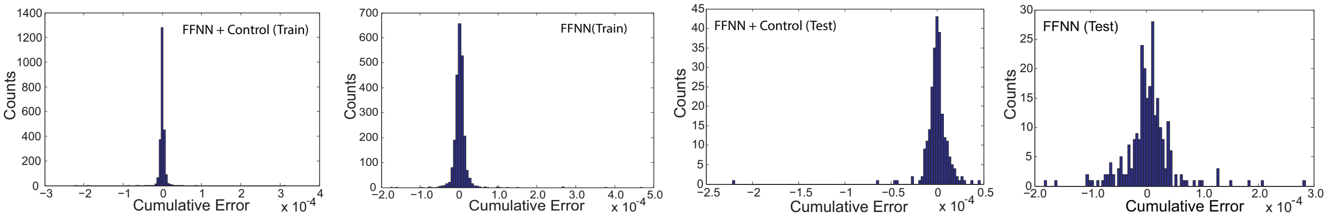

Fig. 6: Error distribution for the networks with and without control nodes. The figures depict the cumulative error of the train set for the FNN with control nodes and regular FFNN.

control nodes enable the FFN to learn the shape of the output while, simultaneously, the error of the important outputs is kept minimum. This approach of estimation can be useful for the statistical analysis that use the information of post proceed I-Vs such as I_{on}, I_{off}, V_{Th}, and S for the purpose of PV analysis.

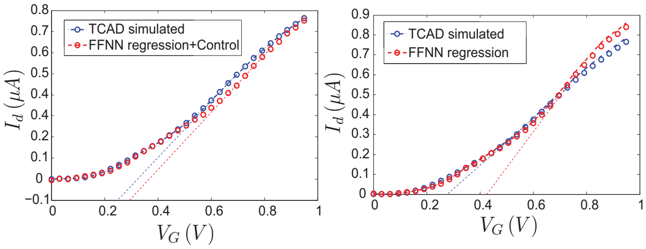

Fig. 7: Two sample I-V regression through different FFNNs. Adding control nodes to FFNNs results in more precise prediction.

In order to verify our methodology, we used it for PV study of DG-SiNWFET by applying a 30% Gaussian random variations for each studied parameters. The same experiments were performed with TCAD simulations for comparison purpose. Table III represents the effect of R_{NW} ,T_{OX}, and L_{CG} variations on I_{on}, I_{off} and V_{Th}. Looking at the obtained values, our methodology demonstrates a good correlation (average error of less than 2% in mean values of I_{on}, I_{off} and V_{Th}) with the results of TCAD simulation.

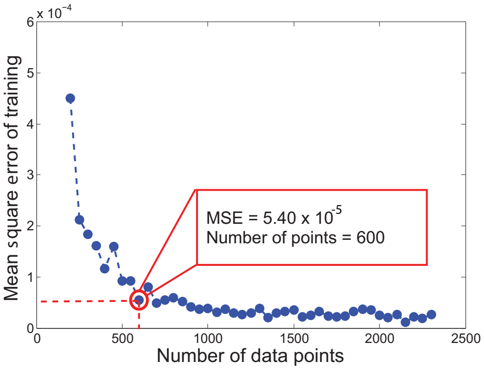

Fig. 8: Necessary number of data points for training the predictor

2) Time Complexity: The runtime of the PV analysis for DG-SiNWFET through the proposed methodology has been shown in Table II. The prediction methodology is clearly faster than TCAD simulation. The time complexity of the FFNN predictor is related to the learning step which is done once. This methodology is useful when a large data set is needed. This is a common case for statistical PV analysis. For example, consider a usual Monte Carlo method for variation analysis of DG-SiNWFET in which 5000 data points is required. Only a small part of the data points (nearly 12%) is necessary to reveal the variation distribution on the device output. In this case, only 600 TCAD simulations can be performed to set up the predictor instead of 5000 overkilling simulations. In the next step, TCAD simulator is replaced by predictor which can quickly produces the necessary data set in $O(sec)$ for 4500 remaining simulation (with average execution time of 0.12 ms per sample). Figure 9 compares the execution time of TCAD and proposed methodology for PV analysis. After a number of TCAD simulation which is necessary for the training, FFNN considerably accelerates the simulations runtime.

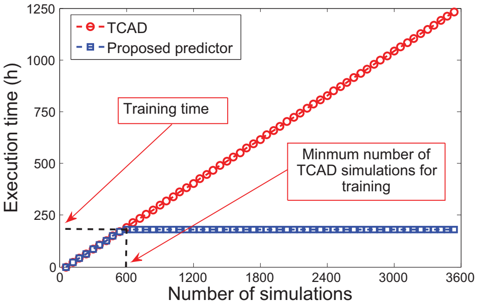

Fig. 9: Runtime comparison of the TCAD simulation with proposed methodology

TABLE II: Execution time comparison

Method	Average execution time for one simulation (CPU: Dual-Xeon X5650, Memory: 24 GB)
TCAD	19 min
Proposed methodology	1.23×10^{-4}sec (Learning time: 18 min)

TABLE III: DG-SiNWFET PV analysis using TCAD and proposed methodology

		TCAD			Proposed methodology		
		R_{NW}	T_{ox}	L_{CG}	R_{NW}	T_{ox}	L_{CG}
I_{on}	$mean\,(\mu A)$	1.28	1.45	1.44	1.29	1.43	1.43
	$std\,(nA)$	528.7	48.7	0.5	536.2	57.7	0.7
I_{off}	$mean\,(fA)$	0.90	0.73	0.72	0.88	0.76	0.72
	$std\,(fA)$	0.53	0.02	0.006	0.49	0.03	0.01
V_{Th}	$mean\,(mV)$	330	331	333	321	346	327
	$std\,(mV)$	55	21	9	59	18	12

V. Conclusion

In this paper, we introduced an efficient methodology for variation analysis of emerging nano-devices. This methodology consists of a prediction module which can be added to TCAD simulator to enhance the time complexity of the PV analysis. The prediction module is based on feed-forward neural network, which is capable of estimating the underlying relation of device parameters and device functionality. Learning this relation enables us to replace repetitive and time consuming TCAD simulation by a fast estimation methodology. Our simulation results revealed that the proposed methodology can effectively be used for PV analysis, while reducing the time complexity of TCAD simulations of device to $O(ms)$ with the maximum error bound of 4.3×10^{-4}.

Acknowledgment

This work has been supported by ERC senior grant NANOSYS ERC-2009-AdG-256810.

References

[1] T.-C. Chen, “Overcoming research challenges for CMOS scaling: industry directions,” ICSICT Tech. Dig., 2006.

[2] V. Subramanian, “Multiple gate field-effect transistors for future CMOS technologies,” IETE Tech. Rev., 27(6): 446, 2010.

[3] K. Ryu *et al.*, “CMOS-analogous wafer-scale nanotube-on-insulator approach for submicrometer devices and integrated circuits using aligned nanotubes,” Nano Letters, 9(1):189-197, 2008.

[4] S. Bangsaruntip *et al.*, “High performance and highly uniform gate-all-around silicon nanowire MOSFETs with wire size dependent scaling,” IEDM Tech. Dig., 2009.

[5] M. De Marchi *et al.*, “Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire FETs,” IEDM Tech. Dig., 2012.

[6] M.H. Ben Jamaa, K. Mohanram and G. De Micheli, “An Efficient Gate Library for Ambipolar CNTFET Logic,” IEEE Trans. on CAD, 30(2):242-255, 2011.

[7] R.I. Bahar *et al.*, “Architectures for Silicon Nanoelectronics and Beyond,” IEEE Computer, 40(1):25-33, 2007.

[8] W. Hong *et al.*, “A Novel Dimension-Reduction Technique for the Capacitance Extraction of 3-D VLSI Interconnects,” IEEE Trans. on Microwave Theory and Techniques, 46(8): 1037-1044, 1998.

[9] C. Heitzinger, and S. Selberherr, “An Extensible TCAD optimization framework combining gradient-based and genetic optimizers,” Microelectronics J., 33:61-68, 2002.

[10] J.K. Lorenz *et al.*, “Hierarchical Simulation of Process Variations and Their Impact on Circuits and Systems: Methodology,” IEEE Trans. on Electron Devices, 58(8): 2218-2226, 2011.

[11] [Online]. Available: www.synopsys.com

[12] G. Pei *et al.*, “FinFET design considerations based on 3D TCAD simulation and analytical modeling,” IEEE Trans. on Electron Devices, 49(8):14111419, 2002.

[13] A. Zaka, “Characterization and 3D TCAD simulation of NOR-type flash non-volatile memories with emphasis on corner effects,” Solid-State Electron., 63(1):158162, 2011.

[14] M. Nawaz, “On the device design assessment of multigate FETs (MuGFETs) using full process and device simulation in 3D TCAD,” Microelectronics, 38(12):12381251, 2007.

[15] R. Strasser, and S. Selberherr, “Parallel and Distributed TCAD Simulations using Dynamic Load Balancing,” Simulation of Semiconductor Processes and Devices, 89-92, 1998.

[16] P. Fleischmann, “Grid generation for three-dimensional process and device simulation,” SISPAD Tech. Dig.,1996.

[17] W. Wessner, “Anisotropic mesh refinement for the simulation of three-dimensional semiconductor manufacturing processes,” IEEE Trans. on CAD, 25(10):21292139, 2006.

[18] A.N. Bhoj, R.V. Joshi, and N.K. Jha, “Efficient Methodologies for 3-D TCAD Modeling of Emerging Devices and Circuits,” IEEE Trans. on CAD, 32(1):47-58, 2013.

[19] D.F. Specht, “A general regression neural network,” IEEE Trans. on Neural Networks, 2(6):568-576, 1991.

[20] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” Neural Networks, 2(5):359-366, 1989.

[21] D. Basak, S. Pal, and D.C. Patranabis, “Support vector regression,” Neural Information Processing Lett., 11(10):203-224, 2007.

[22] S. Dreiseitl, L. Ohno-Machado, “Logistic regression and artificial neural network classification models: a methodology review,” Biomedical Informatics, 35(5-6):352-359, 2002.

[23] E.M. Biganzoli, F. Ambrogi, and P. Boracchi, “Partial Logistic Artificial Neural Networks (PLANN) for Flexible Modeling of Censored Survival Data,” IJCNN Tech. Dig., 2009.

[24] L.S.H. Ngia, “Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm,” IEEE Trans. on Signal Processing, 48(7):1915-1927, 2000.

[25] K. Hornik, ”Approximation Capabilities of Multilayer Feedforward Networks”, Neural Networks, 4(2):251-257, 1991.