Fast and Low-Error Prediction of Logic Gate Cell Characterization

Gabriel L. Jacinto[1], Cinthia S. Schneider[2], Alexandra L. Zimpeck[2], Mateus Grellert[3] and Cristina Meinhardt[1]
[1]*Department of Informatics and Statistics, Federal University of Santa Catarina (UFSC), Brazil*
[2]*Social Science and Technological Center, Catholic University of Pelotas (UCPEL), Brazil*
[3]*Federal University of Rio Grande do Sul (UFRGS), Brazil*
gabriellimajacinto@gmail.com, cinthia.schneider@sou.ucpel.edu.br, alexandra.zimpeck@ucpel.edu.br,
mateus.grellert@gmail.com, cristina.meinhardt@ufsc.br

***Abstract*—This work proposes a machine learning approach for cell characterization of logic gates. Traditional electrical simulation-based characterization faces challenges related to foundry secrecy and runtime. The proposed framework addresses these challenges by utilizing ML models to estimate power consumption and propagation times. The experiments demonstrate the potential of the framework to predict different logic gate functions in different technology models, showing the feature extraction differences for bulk CMOS and FinFET devices. Results demonstrate the effectiveness of the Decision Tree algorithm in fast and accurately predicting cell behavior, with inference times almost a thousand times faster than the traditional electrical simulation and coefficient of determination superior to 95%.**

***Index Terms*—Cell characterization, FinFET devices, Machine Learning**

I. Introduction

Cell characterization is a time consuming task however fundamental in the availability of cell libraries for standard cell synthesis. Traditional cell characterization tools rely on electrical simulations, such as from SPICE simulators, to evaluate the performance, power and efficiency of a given circuit designed in a specific technology, described through a compact model that abstracts the electrical behavior of a transistor. As technology has advanced to the nanoscale, the parameters affecting circuit operation have increased considerably, expanding the number of cases to be evaluated for a complete cell characterization combining the effects of process variability, temperature, different voltage levels with traditional characterization corners. Moreover, there is a foundry reluctance to share transistor model cards limiting the early evaluation of technologies to choose which device is suitable for a specific circuit, for example, if FinFET is a good alternative compared to traditional CMOS [1].

The electrical characterization of FinFET circuits is very similar with planar devices. The key differences from planar devices are 1) the fin width (W) and fin length (L) parameters are fixed according to the technology adopted, and the sizing of FinFETs is given by the number of fins, and, 2) the main sources of process variability. In bulk CMOS devices, the random dopant fluctuation was considered the most critical source. As the channel of FinFETs is lightly doped, these devices suffer less from dopant-induced variations. However, FinFETs main variability source is due to the work-function fluctuation (WFF) derived from the metal gate granularity [2].

In recent years, we have observed new approaches exploring machine-learning (ML) for modeling emerging technologies behavior, mainly to define: 1) transistor models that learn and reproduce the I-V curve (electrical response) of a transistor [3] [4], or 2) predict the effects process variability on the IDS-VGS characteristics to reduce the conventional TCAD simulation running time [5] [6]. However, to the best of our knowledge, few works look at the problem at the gate level. In [7], it is proposed a ML approach that quickly generates cell libraries performing automatic parameter optimization on transistor parameters. In [8], we evaluate machine learning regression algorithms as an alternative to exhaustive electrical simulation in cell characterization, presenting a first case study for an Inverter using 16 nm bulk CMOS technology. The preliminary evaluation shows the advances of adopt a decision tree model concerning accuracy, and fast time to estimate the cell behavior. In this paper, we proposed a generic machine learning-based framework for predicting power and propagation times. This environment can preserve the foundry secrecy about the device models, and the ML model acts as a light and suitable black-box model. The evaluation shows the framework training and validation considering bulk CMOS and FinFET based standard cells.

II. Proposed framework

This work proposes a framework for characterizing cells from a standard cell library adopting the SPICE netlists. This workflow was designed so it can be generalized to work with other circuits and technologies. Fig. 1 summarizes the training process. The electrical characterization target the energy and propagation time prediction, considering process variability effects. The proposed flow is used to characterize cells adopting bulk CMOS and FinFET device models. The input variables considered were voltage, temperature, output capacitance, number of fins and work-function fluctuation. The predicted variables are the energy and all the propagation times Tp_{HL} (propagation time high-to-low, i.e., the fall delay) and Tp_{LH}(propagation time low-to-high, i.e., the rising delay) for the logic functions under evaluation. Thus, the process is divided into five main steps, starting with the electrical simu-

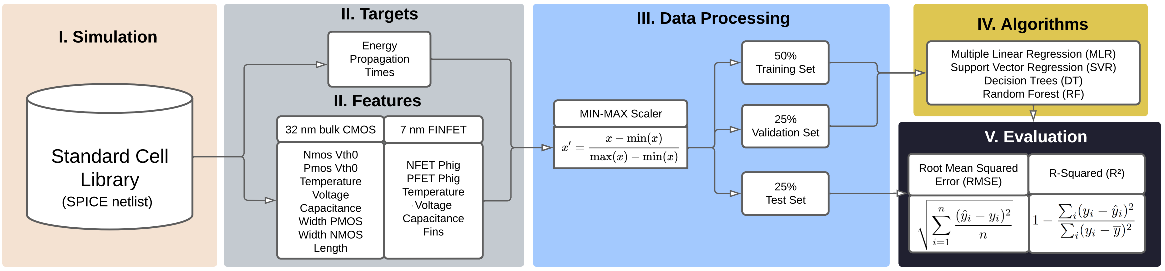

Fig. 1. Training flow of the Proposed Framework

lations to the data set generation followed by the treatment of the target variables, performing the normalization, and also the separation of the data set into training, validation, and testing sets for the training of the evaluated regression algorithms. Stage four contemplates the development and adjustment of the considered algorithms. Finally, step five performs the analysis of the algorithms.

To accurately predict the logic cell performance by the regression models, the data set should include samples of the cell under evaluation under various conditions and affected by process variations. The samples are generated by electrical simulations with Cadence's Spectre observing the energy and all propagation time arches to define the delay of the circuits. These variables are influenced by several parameters of the configuration of the circuit depending on the device model adopted, such as the number of fins, output capacitance, temperature, and operating voltage. Also, variability effects impact these variables depending on the parameters adopted. The first step was to define clearly which input variables are relevant for the output values being modeled. This experiment adopts the 32 nm bulk CMOS model [9] and the 7 nm ASAP FinFET model [10]. The machine learning models are evaluated individually for each of the device model. We define for both device models evaluated the input variables of temperature, voltage, output capacitance. For bulk CMOS devices, we define also the threshold voltage for NMOS and PMOS devices by the parameter $Vth0$ in the device model, the channel length, the width of PMOS devices, and the width of NMOS devices. For FinFET devices, we define the number of fins and the work-function fluctuation by the NFET and PFET $Phig$ model parameter. The variability of the process is simulated using the Monte Carlo method [2]. The input variable values were predefined, following these range of values:

- **Temperature** (^oC): -25, 0, 25, 50, 75, and 100
- **Voltage (V)**: 0.6, 0.7, 0.8, and 0.9
- **Output Capacitance (f)**: 1, 4, 8, and 16
- **NMOS Vth0**: gaussian distribution with mean 0.5088, 3σ, and 10% deviation
- **PMOS Vth0**: gaussian distribution with mean -0.450, 3σ, and 10% deviation
- **PMOS Width (nm)**: 70, 140, 280, 350, and 420
- **NMOS Width (nm)**: 70, and 140
- **CMOS channel Length (nm)**: 20, 32, and 40
- **FinFET number of fins**: 1, 2, 3, 4, and 5
- **NFET work-function (Phig)**: gaussian distribution with mean 4.372, 3σ, and 3% deviation
- **PFET work-function (Phig)**: gaussian distribution with mean 4.8108, 3σ and 3% deviation

To evaluate the proposed flow and identify the best regression algorithm, we choose two basic logic gates: the Inverter and the NAND2 gate complementary network topologies. In the end, more than 100 transient simulations of 20 ns are executed for FinFET and 1000 transient simulations of 20 ns are executed for CMOS, obtaining a total of 96,000 for FinFET and 852,480 observations for CMOS. However, the data processing step removes outliers values that would confound the model results. After that, the final number of observations stood at 91,134 cases for FinFET and 731,705 for CMOS. It should be noted that these outlier values may occur due to internal errors in the electrical model used by the simulator considering edge values in the variables.

Before training the algorithms, it was necessary to resize each variable, because, in addition to some values being very small, some algorithms (such as the Support Vector Regression) are sensitive to different scales. For this rescaling, the method known as normalization was used, which scales the data to the range [0,1].

We evaluate four machine learning models for each output variable considering the regression algorithms Multiple Linear Regression (MLR), Support Vector Regression (SVR), Decision Trees (DT), and Random Forest (RF), totaling 64 models trained. We opt to keep the same algorithms from the previous evaluation for 16 nm bulk CMOS devices presented in [8]. Copies of the data sets were created and then the *model selection* module of *sci-kit learn* [11] was used to divide the data set between the training (50%), validation (25%) and test (25%) groups.

The tools used to manipulate, visualize, clean, train and test the data were Jupyter Notebooks and the python language libraries sklearn, pandas, NumPy, matplotlib, and seaborn. All algorithms were trained using *cross-validation* with 10 folds, so that the algorithm makes more general predictions and avoids overfit or bias in the data that was fed during its training. For this, *cross-validation* uses the training set and the validation set to first evaluate how the model is behaving in the 10 folds of these subsets. Furthermore, in this process, several copies of the model being trained are created and its hyperparameters are varied to determine which is the best model with the best configuration of these hyperparameters. For both the Decision Tree and the Random Forest, *Depth* is the maximum depth each tree will have. The Random Forest *# est.* values are how many estimators, that is, how many trees will be created for the model. The hyperparameters are configured in the cross-validation to evaluate DT Depth values of $1, 2, 3, 4, 5, 6, 8, 10, 25, 50$. For RF, the Depth values were the same and the number of estimators were $5, 25, 50, 75, 100, 150$. The SVR models were not included in the cross-validation because their training proved to be very time-extensive.We decided to use the default hyperparameter values: $C = 1$ and $Gamma = 1/(N_{features} \times Var_X)$. At the end of cross validation, the values from Table I were obtained.

In Table I, we observe a reduction in the complexity of the models for FinFET technology compared to the models for bulk CMOS technology. The bulk CMOS models use higher values of maximum depth for random forest estimators, which means these models need to perform more tests due to the larger number of input features in the bulk CMOS data set. In contrast, there is a reduction in dimensionality for the FinFET data set, requiring fewer tests.

TABLE I
CROSS-VALIDATION RESULTS

Model	Gate	Variable	RF		DT
			Depth	# est.	Depth
32 nm	INV	Tp_HL	5	25	5
		Tp_LH and Energy	10	150	10
	NAND	Tp_HL	5	100	5
		Tp_LH and Energy	5	150	5
7nm	INV	Tp_HL	4	50	4
		Tp_LH	4	75	4
		Energy	8	50	8
	NAND	All Tp	4	50	5
		Energy	2	50	2

III. EVALUATION OF THE REGRESSION MODELS

The evaluation metrics are presented in Table II and Table III. The best models for the Inverter were tree-based models, with RF achieving the highest R^2 score. The worst algorithm was MLR due to its linear nature, which fails to learn from data of varying complexity. SVR performed best in terms of R^2 score for energy, while it fell below 85% for delay. The same behavior was observed for the NAND2, where RF and DT algorithms had the highest R^2 scores for both delay and energy. MLR remained the worst model, followed by SVR. The only exception was found in the performance metrics for the falling delays (Tp$_{HL}$ A and Tp$_{HL}$ B), where SVR had a worse performance than MLR, with R^2 scores of 53% and 31%, respectively. It is worth noting that similar metrics for rising and falling delays of each input of the gate varied only from the third decimal place. Thus, tree-based algorithms showed the best performance across evaluation metrics, with a very small difference between DT and RF.

For the FinFET model, the Inverter also showed the best predictions using tree-based models, with RF and DT achieving R^2 scores of 97% and 94% for rising and falling delays. The poorest performer for the Inverter's delays was MLR, scoring below 73%. However, it is worth noting that SVR exhibited better performance for the 7 nm FinFET technology compared to the 32 nm bulk CMOS. The same pattern did not hold for energy, as SVR had worse performance (R^2 = 47%) than MLR (R^2 = 59%) for the Inverter. The best models for energy continued to be tree-based methods, with RF and DT showing higher differences now, with R^2 scores of 98% and 95% respectively. For the NAND gate, the tree-based models outperformed the others once again, with delays scores above 90%. There was more variation among the delay types in the tree-based models, but not as consistent as MLR and SVR. Overall, the Tp$_{LH}$ A and Tp$_{HL}$ B delays had worse predictions, with R^2 scores of 95% and 97% respectively. In terms of energy, the R^2 scores ranked as follows in ascending order: 97.7%, 97.8%, 99.7%, and 99.8%, respectively for SVR, MLR, DT, and RF. Thus, tree-based models remained the best performers, with RF having a slightly more significant R^2 value than DT. Therefore, it is evident that for both technologies, the conclusion regarding the models and their performances remains the same, with RF being the best-performing model in terms of overall r2-score, followed by DT.

Tree-based models experienced a slight decrease in performance for both gates when transitioning to FinFET compared to CMOS, although both RF and DT remained above 90% and remained the best option among SVR and MLR models for all variables. This reinforces the idea that was discussed previously regarding the cross-validation results, that showed that we had simpler models for the 7 nm FinFET technology, which was a direct consequence of the dimensionality reduction caused by the lower number input features when compared to the 32 nm bulk CMOS data set. Therefore, with a simpler data set for FinFET, the tree-based models have few tests to realize so to reach a prediction. It is not enough for the model to be low in error, but also fast if we want to propose a method that is better than exhaustive simulation methods. To further deepen the discussion, Table IV shows the different values for the inference time for each model and the Spectre simulation (Sim.). The the most time-consuming model (RF) take less than 10% of the simulation time to predict the values. We can also see that DT has a smaller inference time than RF. Therefore, even though RF has a slight better performance, DT has a faster inference time and since its performance for all target variables is not so different from RF, DT is the best model to predict electrical behavior from simulation data for the logic gates NAND and NOT.

IV. CONCLUSION

The Decision Tree proved to be the fastest and as high in the R^2 score as the Random Forest, therefore being the best algorithm for this work. There was a slight decrease

TABLE II
EVALUATION OF THE REGRESSION MODELS ON THE PREDICTION OF PROPAGATION TIMES AND ENERGY FOR 32 NM BULK CMOS

Gate	Variable	**32 nm bulk CMOS**							
		MLR		**SVR**		**DT**		**RF**	
		RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
INV	Tp_HL	0.0956	0.595	0.0599	0.8465	0.0038	0.9996	0.0032	0.9997
	Tp_LH	0.0822	0.7311	0.0545	0.882	0.0035	0.9995	0.0028	0.9996
	Energy	0.1778	0.4732	0.0569	0.9327	0.0135	0.9969	0.0103	0.9983
NAND2	Tp_LH A	0.0483	0.5893	0.0431	0.673	0.0012	0.9997	0.0009	0.9998
	Tp_HL A	0.0396	0.5338	0.0480	0.3128	0.0015	0.9993	0.0012	0.9996
	Tp_LH B	0.0472	0.5928	0.0442	0.6425	0.0015	0.9996	0.0011	0.9998
	Tp_HL B	0.0396	0.5363	0.0482	0.3127	0.0017	0.9991	0.0012	0.9996
	Energy	0.0354	0.973	0.0351	0.9734	0.0033	0.9997	0.0022	0.9998

TABLE III
EVALUATION OF THE REGRESSION MODELS ON THE PREDICTION OF PROPAGATION TIMES AND ENERGY FOR 7 NM FINFET

Gate	Variable	**7 nm FinFET**							
		MLR		**SVR**		**DT**		**RF**	
		RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
INV	Tp_HL	0.0793	0.7291	0.0441	0.9162	0.0259	0.9711	0.0254	0.9722
	Tp_LH	0.0615	0.7064	0.0412	0.8683	0.0272	0.9425	0.0268	0.9443
	Energy	0.0431	0.5951	0.0493	0.472	0.0147	0.9529	0.0096	0.9798
NAND2	Tp_LH A	0.0611	0.7132	0.0412	0.8695	0.019	0.9721	0.0184	0.9739
	Tp_HL A	0.0629	0.7153	0.0418	0.8745	0.0020	0.9997	0.0012	0.9999
	Tp_LH B	0.0610	0.7134	0.0412	0.8695	0.0021	0.9996	0.0015	0.9998
	Tp_HL B	0.0629	0.7153	0.0417	0.8748	0.0263	0.9503	0.0257	0.9527
	Energy	0.0383	0.9783	0.0395	0.9769	0.0129	0.9975	0.0128	0.9976

TABLE IV
SIMULATION TIME AND INFERENCE TIME (MS)

Model	**Gate**	**Sim.**	**MLR**	**SVR**	**DT**	**RF**
32 nm	INV	217.0	0.5	0.4	0.4	17.2
	NAND2	248.0	0.4	0.5	0.4	27.1
7 nm	INV	258.0	0.5	0.4	0.4	17.2
	NAND2	300.0	0.6	0.8	0.5	29.5

in performance on the FinFET data set, since it had less features than the CMOS data set. The direct consequence of this complexity reduction is reflected in the quantitative results expressed by the models scores. It can be noticed that the R^2 value is relatively lower for the FinFET gates, although still high and accurate. In addition to the lower number of input features, there is also a trade-off between bias and variance because less complex models have lower bias and better generalization of data. Therefore, to achieve similar values to the CMOS models, it is necessary to add more information to the data set so that the algorithms can learn to make equally precise predictions. In future work, feature extraction will be explored by feeding the positioning/layout of the logic gate transistors into the models to compensate for the lack of information present in the other CMOS models.

ACKNOWLEDGMENT

This work was financed in part by National Council for Scientific and Technological Development – CNPq, the FAPERGS and the Propesq/UFSC.

REFERENCES

[1] Florian Klemme, Jannik Prinz, Victor M. van Santen, Jörg Henkel, and Hussam Amrouch. Modeling emerging technologies using machine learning: Challenges and opportunities. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9, 2020.

[2] Alexandra L. Zimpeck, Cristina Meinhardt, and Ricardo Reis. Evaluating the impact of environment and physical variability on the ion current of 20nm finfet devices. In *2014 24th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, pages 1–8, 2014.

[3] Kumar Sheelvardhan, Surila Guglani, M. Ehteshamuddin, Sourajeet Roy, and Avirup Dasgupta. Machine learning augmented compact modeling for simultaneous improvement in computational speed and accuracy. *IEEE Transactions on Electron Devices*, pages 1–7, 2023.

[4] Huifan Zhang, Youliang Jing, and Pingqiang Zhou. Machine learning-based device modeling and performance optimization for finfets. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 70(4):1585–1589, 2023.

[5] Paul R. Genssler and Hussam Amrouch. Brain-inspired computing for circuit reliability characterization. *IEEE Transactions on Computers*, 71(12):3336–3348, 2022.

[6] Jaehyuk Lim, Jinwoong Lee, and Changhwan Shin. Probabilistic artificial neural network for line-edge-roughness-induced random variation in finfet. *IEEE Access*, 9:86581–86589, 2021.

[7] Florian Klemme, Yogesh Chauhan, Jörg Henkel, and Hussam Amrouch. Cell library characterization using machine learning for design technology co-optimization. In *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9, 2020.

[8] Gabriel Lima Jacinto, Lucas Yuki Imamura, Mateus Grellert, and Cristina Meinhardt. Exploring machine learning for electrical behavior prediction: The cmos inverter case study. In *2022 35th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI)*, pages 1–6, 2022.

[9] Wei Zhao and Yu Cao. New generation of predictive technology model for sub-45nm design exploration. In *7th International Symposium on Quality Electronic Design (ISQED'06)*, pages 6 pp.–590, 2006.

[10] Lawrence T. Clark, Vinay Vashishtha, Lucian Shifren, Aditya Gujja, Saurabh Sinha, Brian Cline, Chandarasekaran Ramamurthy, and Greg Yeric. Asap7: A 7-nm finfet predictive process design kit. *Microelectronics Journal*, 53:105–115, 2016.

[11] David Cournapeau. Scikit-learn Machine Learning in Python. https://scikit-learn.org/stable/, 2007. [Online; accessed 03-May-2022].