Electrical Behavior Prediction Of An Inverter Using Machine Learning Algorithms

Gabriel Lima Jacinto and Cristina Meinhardt
Department of Informatics and Statistics, Federal University of Santa Catarina
gabriellimajacinto@gmail.com and cristina.meinhardt@ufsc.br

***Abstract*—With the advance of technology nodes, more corners must be consider during the electrical characterization to cover challenges like process variability effects. The adoption of machine learning techniques came to assist digital design in many abstraction levels. Thus, the main objective of this paper is evaluating machine learning regression algorithms (multiple linear regression, support vector regression, decision trees, and random forest) as an alternative to electrical simulation on the cell design. This paper presents the results for a first case study: a CMOS inverter. Specifically, this work will separately predict the values of the energy and the propagation times of the circuit. A comparative analysis is made for each variable between all four models in order to understand which one is the best regression model suited for the task. The algorithm with the lowest cost function proved to be the Multiple Linear Regression for all the predicted variables.**

***Index Terms*—Cell characterization, Power, Delay, Machine Learning**

I. INTRODUCTION

As technologies advance, more challenges are emerging. The variability and design rules become more extensive and more complex. Consequently, design tools have to solve increasingly complex problems. This problem has been growing exponentially, and these difficulties directly reflect on the cost of developing electronic device designs [1]. One of the major challenges on nanometer nodes is the process variability. Process variability affects the expected normal behavior of the cells, changing the delay and power consumption observed at nominal conditions. This expands the cell characterization to enclose now multiple corners, despite the traditional fast, low and nominal corners. To explore a large range of possible corners on a target technology, traditionally designers have adopted electrical simulations on the timing and power measurement. Thus, increasing the number of electrical simulations to cover multiple corners could became a large time demand task. Machine learning techniques are presented as good alternatives as an attempt to reduce costs for the development of electronic devices.

The bibliographic review shows that there is still a lot of space to be explored regarding the use of machine learning for microelectronic tools. Its present usage is strongly linked to quality prediction of candidate solutions given by existing tools, or attempting to guide them to a better quality solution by realizing metric predictions [1].

With the increasing availability of huge quantities of manufacturing data, and the pressures of continuous process improvement and scrap reduction, machine learning techniques can be adopted to analyze, classify, and predict the quality of metal etch [2] or investigate process variability effects on 3D NAND Flash Memory Cell [3]. In the Placement and Routing steps, we found works that use machine learning to predict path-based slack from graph-based timing analysis [4] and pre-routing timing prediction [5].

Furthermore, we have found one work that explores machine learning on the electrical characterization of devices, however, specifically for Magnetic Devices [6]. Thus, this indicates that there is room to also explore machine learning algorithms to electrical characterization of logic cells in bulk CMOS, SOI or FinFET technologies.

This work explore machine learning techniques to predict the electric behavior of circuits. As a first step, we are considering as a case-study the CMOS inverter circuit, and predicting the energy and propagation timing. From the machine learning techniques available, we chose to start investigating with four regression methods, exploring Multiple Linear Regression (MLR), Support Vector Regression (SVR), Decision Tree (DT) and Random Forest (RF). We are considering voltage, temperature and process variability. The prediction methodology presented in this paper will be extend to consider other circuits and technologies in the following steps of this project.

II. BACKGROUND ABOUT MACHINE LEARNING TECHNIQUES

There is a multitude of machine learning algorithms for different tasks, ranging from predicting the values of the stock market to decision-making in a Mars rover. Since, in this work, our prediction target is the energy and delay behavior of a CMOS inverter circuit - which are all continuous values - our approach will be to use a supervised learning technique known as Regression, which will give us the correlation between our different independent variables [**?**]. There are many regression algorithms, hence we will delimit this analysis to the four most common models to see how they fit and compare to each other.

A. Multiple Linear Regression

Linear regression is a statistical technique widely used when one wants to predict the behavior of a target value. Since in this work our dependent variables are predicted based on the values of many independent variables, we will be using Multiple Linear Regression (MLR) instead of simple linear regression. Our predicted value $\hat{y}$ is described in Eq. 1, where θ_0 is the intercept term and $\theta_1 ... \theta_n$ are the weights of each value of the input features (independent variables) [?]. What is desired is to find the values of each θ_n weights that minimize the cost function of the model [?], which in this work will be the root mean square error (RSME) and the mean absolute error (MAE).

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \tag{1}$$

B. Support Vector Regression

Even though Support Vector Machines (SVM) are usually applied to classification problems, it is a versatile algorithm that can also perform regressions, called SVR [?]. The idea is, given a ϵ value, the SVR calculates a margin around the hyper-plane of the prediction function, where we try to fit as many observations as possible between the area set by the margins and reduce the values that are off that same area.

The decision function represented by Eq. 3 computes similarly to the linear regression, however, here we are putting the equation on a vectorized form [?]. The SVR has many kernels (linear, Gaussian RBF, polynomial), which are functions that calculate the dot product $\phi(a)^T\phi(b)$ without calculating the transformation ϕ, therefore only using the vectors a, b for that computation [?]. In this work we will use the linear kernel represented by the Eq. 2.

$$K(a,b) = a^T b \tag{2}$$

$$\hat{y} = w^T x + b \tag{3}$$

C. Decision Trees Regression

Decision Tree (DT) is another algorithm more famously used in classification problems. However, it is also possible to use it in regression tasks [?]. The algorithm works by creating a tree structure with internal nodes with tests that are made in order to predict to which leaf node the feature will be assigned to, returning, in our regression case, a continuous value.

D. Random Forest Regression

The Random Forest (RF), which is a ensemble learning algorithm [?], is a similar algorithm to the decision trees, the difference being that instead of computing only one decision tree to make a decision for the value, it actually calculates a large number of decision trees with different depths and then it computes the average of all predictions to estimate the value. That makes this method one of the most powerful supervised algorithms. We can represent the joined n decision trees as described in Eq. 4.

$$g(x) = f_0(x) + f_1(x) + f_2(x) + f_3(x) + ... + f_n(x) \tag{4}$$

E. Performance Measurements

In order to assess the fitness of each model, we need to evaluate the error between the predicted value and the desired value. In this work, we will be using the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) as performance measurements since they are the most commonly used in regression problems. The RMSE gives higher weights for large errors, which means that it is more sensitive to outliers [?]. They are calculated using the Euclidean norm and the Manhattan norm, as presented in Eq. 5 and Eq. 6 (in vectorized form), respectively, where m is the number of instances on the data set, $x^{(i)}$ and $y^{(i)}$ are vectors of all features values and the desired value of the i^{th} feature of the data set, $\mathbf{X}$ is the matrix of the feature vectors and h is the model function for the prediction [?].

$$RMSE(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2} \tag{5}$$

$$MAE(\mathbf{X}, h) = \frac{1}{m}\sum_{i=1}^{m}|h(x)^{(i)} - y^{(i)}| \tag{6}$$

III. Feature Extraction

In order to predict the tp$_H L$, the tp$_L H$, and the energy of the CMOS inverter circuit, the first step is to define what are the dependent variables and the independent on this experiments. Table I summarizes these information. We will be simulating the values of the following nine independent variables: The threshold voltage for NMOS and PMOS devices by the Vth0 parameter on the device model; the temperature, the voltage, the width (W) and channel length (L) for PMOS and NMOS devices, the Energy and the propagation times low-to-high (Tp$_{LH}$) and high-to-low (Tp$_{HL}$). The last three variables are also the observable variables of the experiment.

TABLE I: Dependent and Independent Variables

Dependent Variable	Tp$_{HL}$	Tp$_{LH}$	Energy
Independent Variables	NMOS Vth0 PMOS Vth0 temperature voltage W PMOS W NMOS L Energy Tp$_{LH}$	NMOS Vth0 PMOS Vth0 temperature voltage W PMOS W NMOS L Tp$_{HL}$ Energy	NMOS Vth0 PMOS Vth0 temperature voltage W PMOS W NMOS L Tp$_{LH}$ Tp$_{HL}$

In this experiment, we are adopting the 16 nm High Performance Bulk CMOS predictive model provided by the PTM [7]. The Vth0 parameter is varied to simulated the process variability impact due to geometric variability and random dopant variability. These parameters were randomly assigned by a Gaussian distribution with 3σ and 10% of standard deviation in a Monte Carlo Simulation [8]. All electrical simulations adopt the Hspice simulation software. We preset the values for the width, length, temperature and voltage, following these variations:

- **Temperature** (^oC): -25, 0, 50, 75, and 100
- **Width of PMOS (nm)**: 140, 70, 280, 350, 420
- **Width of NMOS (nm)**: 70 and 140
- **Length of PMOS and NMOS (nm)**: 32, 20, and 40
- **Voltage (V)**: 0.6, 0.7, 0.8, and 0.9

The values for the energy and propagation times (Tp$_H L$ and Tp$_L H$) were all obtained by the electrical simulations based on all these variables. At the end, we ran over 1000 transient simulation of 20 ns with a step of 0.1 ns, therefore we had a total of 168,000 observations. However, due to outliers removal that otherwise would give us wrong results, our final number of observations was 120,954.

In order to understand how each variable relates to each other, we calculated their linear correlation as shown on Figure 1. As we can see, the strongest linear correlation in our dependent variables are found between them and the width of the transistor and the temperature of the circuit, so they will certainly have a higher weight on the models' decision function.

IV. Prediction Methodology

Our approach to predict the three dependent variables (tp$_H L$, tp$_L H$, energy) will be to create all four models to each of them, exploring Multiple Linear Regression, Support Vector Regression, Decision Trees and Random Forest regression techniques. Thus, in the end, we sum up a total of 12 trained models. Therefore we created three copies of the 120,954 observations data set and then used sci-kit learn model selection module to split our copies of the data sets between the training set and the test set. Our chosen ratio was the standard 80/20.

Scaling was done on every training set because the SVR is sensitive to value scale due to its boundaries calculation. We used the standardization method, that scales our data around the mean with a unit standard deviation. Mathematically speaking, it adopts the Eq. 7.

The tools explored to manipulate, visualize, clean, train and test the data were Google's Colab and the python libraries sklearn, pandas, numpy, matplotlib and seaborn. After we trained our models on 80% of the data, we tested its performance on our test set and calculated each cost function as described on Table II.

$$X' = \frac{(X - \mu)}{\sigma} \tag{7}$$

TABLE II: MAE and RSME results for tpHL, tpHL and Energy (CMOS inverter)

Model	tpHL (ps)		tpLH (ps)		Energy (fJ)	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
MLR	1.26	0.94	4.33	3.14	0.20	0.15
SVR	12.16	11.68	22.08	20.54	0.44	0.37
DT	3.46	2.73	10.13	7.78	0.42	0.34
RF	3.46	2.73	10.13	7.78	0.42	0.34

(Dependent Variable spans tpHL, tpLH and Energy columns.)

V. Propagation Delay High-to-Low Prediction

As we can see in Table II, for the Tp$_H L$, the smallest values of the RMSE and the MAE are $1.26ps$ and $0.94ps$ respectively, which are the values for the errors of the Multiple Linear Regression model. This is a good result given the fact that our values for the Tp$_H L$ range from $6.34ps$ to $42.94ps$. However, when we analyze the results from the other algorithms we get different results. The errors for the Decision Tree and the Random Forest Regressions don't seem to differ a lot from each other, but increase significantly from the Multiple Linear Regression. The highest error was given by the SVR, which means that this algorithm needs more fine tuning to better describe and fit the data.

VI. Propagation Delay Low-to-High Prediction

The results for the Tp$_L H$ are very similar to the Tp$_H L$, however they differ significantly in magnitude. With a range from $4.24ps$ to 67.20 an RMSE of $4.33ps$ and a MAE of $3.14ps$ from the MLR model can be considered a high error value, meaning that the model didn't fit so well to our data. Nevertheless, these error results are the lowest when compared to the other models', meaning that the MLR is still the best method to predict our desired variable. The behavior similarities from the error values of the DT and the RF algorithms remain the same from the Tp$_H L$ prediction with a significantly increase in magnitude. The same case can be observed for the SVR model, with the highest value of the whole table for the RMSE: $22.08ps$.

VII. Energy Prediction

Our energy values range from $-2.68fJ$ to $-0.83fJ$, so the error values for the Multiple Linear Regression model (RMSE: $0.2fJ$; MAE: $0.15fJ$) can be considered satisfactory to predict the variable to a certain accuracy. We can reach better results by fine tuning the model by performing an analyzes with the cross validation technique. For the energy the best algorithm was the MLR just like for the other dependent variables. However this time we didn't see such substantial increase in the MAE and RMSE for the SVR, DT and RF models, with their cost function performing very similarly to the MLR.

VIII. Conclusion

In this work we analyzed four types of machine learning algorithms for the linear regression task of predicting the electrical behavior of a CMOS inveter circuit. We trained the algorithms using the data from Monte Carlo simulation using the Hspice software. The results show that, for all of the three dependent variable (tp$_H L$, tp$_L H$, and energy), the best performing algorithm was the Multiple Linear Regression. We can try to justify this result by understanding that the other algorithms used (SVR, Decision trees and Random Forest) are most noticeably used in classification tasks, henceforth they require more fine tuning in order to better work in regression tasks such as the one in this work. The slight difference between the values of the RMSE and the MAE are as expected of the formulas, since the Euclidean norm will give more weight to higher values due to its quadratic nature.

Therefore, in future works we plan to explore other non-linear models in order to take advantage of other correlations presented in the data, using, for example, a polynomial kernel for the SVR or even neural networks architectures. Also we will use other cost functions techniques (cross validation for example) in order to better evaluate the models and gain more

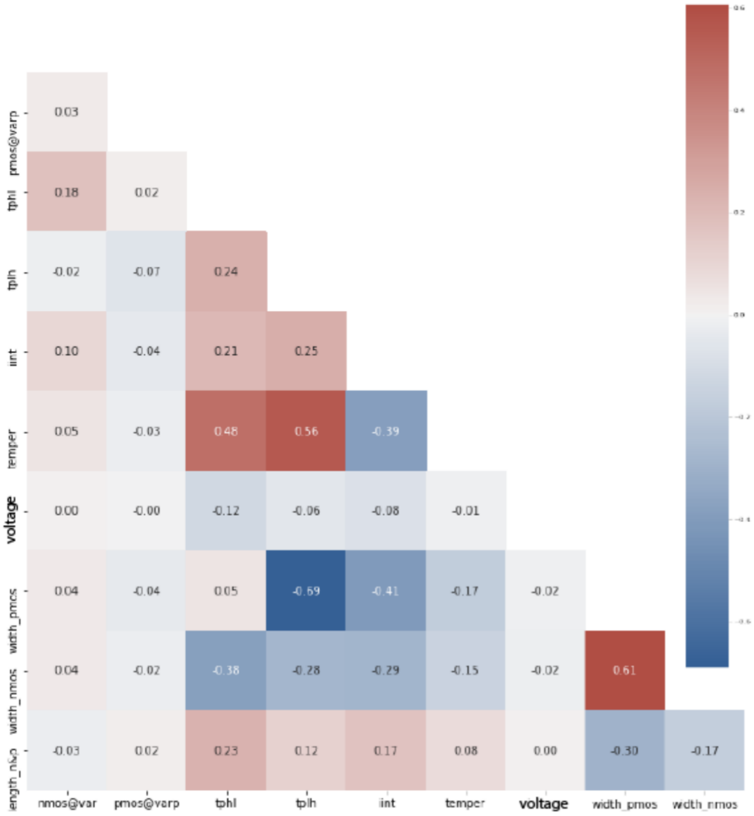

Fig. 1: Linear Correlations (Pearson's R)

insight into what can be done in order to improve its fitness. Furthermore, we will try to better fine tune each model so their cost function is minimized, by, for example, extracting more features from the data set in order to remove redundancies and other types of unwanted dependencies.

Even though some of our variables present a weak linear correlation, therefore indicating that there might be some other type of more complex correlation, our multiple linear regression model did manage to predict the values of our test set satisfactory, which means that machine learning is an interesting and powerful tool to predict the behavior of electrical circuits, even in its most basic technique.

Acknowledgment

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

References

[1] Andrew B. Kahng. Reducing time and effort in ic implementation: A roadmap of challenges and solutions. In *Proceedings of the 55th Annual Design Automation Conference*, DAC 18, New York, NY, USA, 2018. Association for Computing Machinery.

[2] A. Chatterjee, D. Croley, V. Ramamurti, and Kui-Yu Chang. Application of machine learning to manufacturing: results from metal etch. In *Nineteenth IEEE/CPMT International Electronics Manufacturing Technology Symposium*, pages 372–377, 1996.

[3] J. K. Lee, K. Ko, and H. Shin. Analysis on process variation effect of 3d nand flash memory cell through machine learning model. In *2020 4th IEEE Electron Devices Technology Manufacturing Conference (EDTM)*, pages 1–4, 2020.

[4] A. B. Kahng, U. Mallappa, and L. Saul. Using machine learning to predict path-based slack from graph-based timing analysis. In *2018 IEEE 36th International Conference on Computer Design (ICCD)*, pages 603–612, 2018.

[5] E. C. Barboza, N. Shukla, Y. Chen, and J. Hu. Machine learning-based pre-routing timing prediction with reduced pessimism. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6, 2019.

[6] A. Kaintura, K. Foss, I. Couckuyt, T. Dhaene, O. Zografos, A. Vaysset, and B. Sorée. Machine learning for fast characterization of magnetic logic devices. In *2018 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, pages 1–3, 2018.

[7] 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.

[8] Massimo Alioto, Elio Consoli, and Gaetano Palumbo. Variations in nanometer cmos flip-flops: Part i—impact of process variations on timing. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 62(8):2035–2043, 2015.