An accurate neural network model to study threshold voltage variability due to metal grain granularity in Nanosheet FETs

Julian G. Fernandez[a,*], Enrique Comesaña[b], Natalia Seoane[a], Juan C. Pichel[a], and Antonio García-Loureiro[a]
[a]CiTIUS, University of Santiago de Compostela, Spain (*e-mail: julian.garcia.fernandez2@usc.es)
[b]Escola Politécnica Superior de Enxeñaría, Universidade de Santiago de Compostela, Spain

NANOSHEET FETs are currently considered one of the preferred architectures for the next technology nodes [1]. Due to the expensive manufacture of new devices, other solutions, such as technology-aided computer design (TCAD), are needed to evaluate the impact of variability on future transistors. However, the realistic simulation of these devices is computationally demanding. Therefore, exploring new techniques such as the Pelgrom-based predictive model [2] or the application of machine learning techniques [3], [4], [5] is essential.

We present a multi-layer perceptron (MLP) neural network (NN) to estimate the impact of metal grain granularity (MGG), one of the most harmful sources of variability [6], on the threshold voltage (V_{Th}) of a 12 nm gate length nanosheet (NS) FET. The Si NSFET was previously calibrated in [7] against the experimental device reported in [8]. The simulations were carried out using VENDES [9] with the density-gradient quantum-corrected drift-diffusion methodology, and the linear extrapolation method [10] is used to extract the V_{Th}.

MGG consists of the appearance of different metallic grain orientations with different work functions (WF) during the gate deposition process. To implement this source of variability, we have generated random MGG profiles where the grains are created with Poisson-Voronoi diagrams depending on the average grain size (GS) [11]. The TiN metal gate has two grain orientations with WF of 4.4/4.6 eV, and occurrence probabilities of 40/60$\%$, respectively. To have statistical significance we generate around 900 profiles for each GS studied in this work ($3, 5, 7, 10$ nm). Fig. 1 shows an scheme of a NSFET affected by MGG, including its main design parameters. The implementation of these realistic MGG profiles in the NN training is the main novelty of this research, as previous works generate synthetic profiles with fixed-size rectangular (3.92×3 nm^2 [4]) or square grains (2×2 nm^2 [5]). Note that, since the GS depends on the annealing temperature and duration of the gate deposition process [12], it is important to evaluate a variety of GSs.

The MLP-NN was developed using Python 3.9, the Scikit-learn 1.0.2 [13], and the PyTorch Lightning 1.9.0 library. Several hyperparameters, such as the batch size ($bs = 64$), the initial learning rate ($lr = 0.1$), the number of neurons and hidden layers, were calibrated to optimize the MLP performance using the Ray Tune 2.2.0 library [14]. Fig. 2 shows the structure of the MLP-NN, with an input layer corresponding to the number of features of the MGG profile (N_{x_i}), two hidden layers with 234 and 44 neurons, and an output layer corresponding to the V_{Th}. ReLU is used as the activation function, the mean square error (MSE) as the loss function, and an adaptative lr scheduler to avoid divergence in the MSE minimization. The stochastic gradient descent (SGD) optimization algorithm with a momentum $= 0.9$ is implemented. The main issue of using a realistic MGG profile (368×41 maps) is the huge amount of features ($N_{x_i} = 15088$), a value much larger than the sample size ($N_{sample} = 3604$), which causes problems in training, since for a regression $N_{x_i} < N_{sample}$. The sample is split into three subsets (train, validation, test), being the train size $N_{train} = 2306$. The principal component analysis technique (PCA) is applied with a 95$\%$ threshold cut-off of the cumulative variance, to determine the representative N_{x_i} value. With this methodology (see Fig. 3) the train dataset features are reduced from $N_{x_i} = 15088$ to $N_{x_i} = 700$ making $N_{x_i} < N_{train}$. The metrics used to evaluate the training process are the coefficient of determination (R^2) and the mean absolute percentage error ($MAPE$), obtaining for the test values a $R^2 = 0.977$ and a $MAPE = 1.36\%$. Fig. 4 shows the comparison between predicted and simulated V_{Th} test results.

The computational time (t_{comp}) to reduce the features with PCA and train the MLP network is 14 min, with the advantage of being usable for future predictions without any extra computational cost. Considering that in an Intel Core i9-10850K CPU 3.60 GHz processor each quantum corrected drift-diffusion simulation takes 7.5 hours, decreasing N_{train} will lead to a huge reduction in t_{comp}. Therefore, Fig. 5 shows the effect, after PCA features reduction, of decreasing the fraction of the training dataset from 1 ($N_{train} = 2306$) to 0.2 ($N_{train} = 462$). The performance metrics for a fraction of 0.6 ($N_{train} = 1384$) are $R^2 = 0.972$, $MAPE = 1.56\%$, very similar values to those of the complete dataset. Note that, even for a fraction of 0.2, we still obtain a good accuracy ($R^2 = 0.937$, $MAPE = 2.73\%$).

In conclusion, we presented an MLP-NN to estimate the MGG-induced V_{Th} variability on a 12 nm NSFET, with an accuracy of $R^2 = 0.977$. We demonstrated that this NN could obtain an accuracy of $R^2 = 0.937/0.972$ by using only the 20/60$\%$ of the training dataset, reducing the computational time $5.0\times/1.6\times$ with respect to the total train dataset. Also, once the NN is trained, it can accurately predict the impact of realistic MGG variability on V_{Th} with no further simulations.

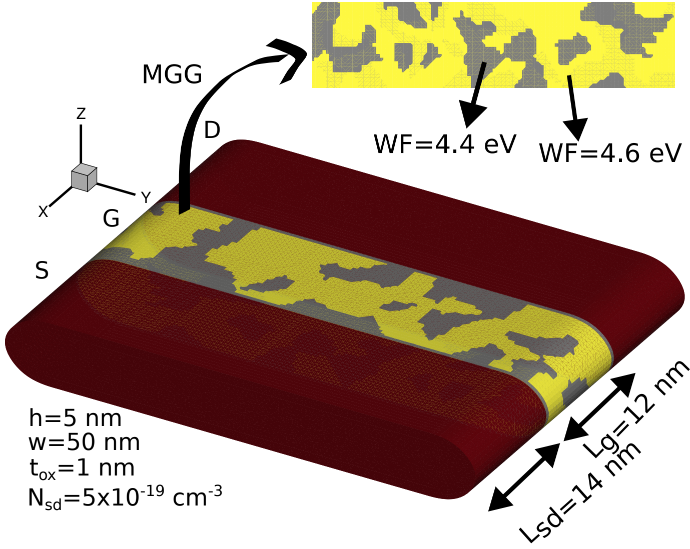

Figure 1: 12 nm gate length (Lg) nanosheet FET with MGG. Regions: source (S), gate (G), and drain (D). Lsd and Nsd are the length and doping of S and D. w and h are the channel width and height. tox is the effective oxide thickness. The TiN work functions (WF) are 4.4 eV (40%) and 4.6 eV (60%).

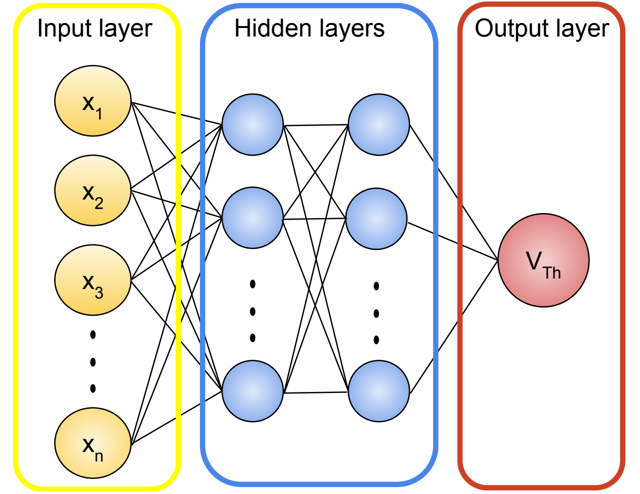

Figure 3: A multi-layer perceptron neural network with an input layer, two hidden layers and an output layer. x_1 to x_n (input) are the MGG features. V_{Th} (output) is the threshold voltage.

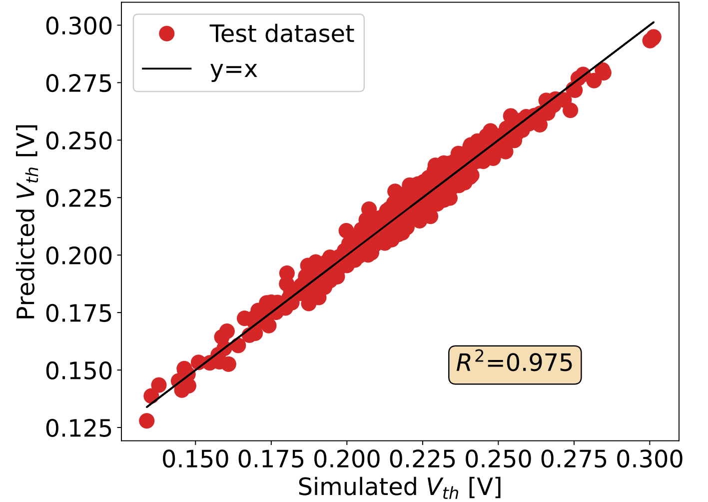

Figure 4: Comparison between simulated TCAD threshold voltage (V_{Th}) and MLP predictions.

References

[1] "More moore," in *IEEE IRDS*, 2022.
[2] J. G. Fernandez *et al.*, 2022. doi: 10.1109/JEDS.2022.3214928
[3] H. Carrillo-Nuñez *et al.*, 2019. doi: 10.1109/LED.2019.2931839
[4] R. Butola *et al.*, 2022. doi: 10.1109/TMTT.2022.3198659
[5] C. Akbar *et al.*, 2022. doi: 10.1016/j.compeleceng.2022.108392
[6] N. Seoane *et al.*, 2021. doi: 10.1109/LED.2021.3109586
[7] D. Nagy *et al.*, 2020. doi: 10.1109/ACCESS.2020.2980925
[8] N. Loubet *et al.*, 2017. doi: 10.23919/VLSIT.2017.7998183
[9] N. Seoane *et al.*, 2019. doi: 10.3390/ma12152391
[10] A. Ortiz-Conde *et al.*, 2013. doi: 10.1016/j.microrel.2012.09.015
[11] G. Indalecio *et al.*, 2016. doi: 10.1109/TED.2016.2556749
[12] H. Dadgour *et al.* doi: 10.1109/IEDM.2008.4796792
[13] F. Pedregosa *et al.*, 2011. doi: 10.48550/arXiv.1201.0490
[14] R. Liaw *et al.*, 2018. doi: 10.48550/arXiv.1807.05118

Aknowledgment

This work was supported by the Spanish MICINN, Xunta de Galicia, and FEDER Funds under Grant RYC-2017-23312, Grant PID2019-104834GB-I00, Grant ED431F 2020/008, PLEC2021-007662, and Grant ED431C 2022/16.

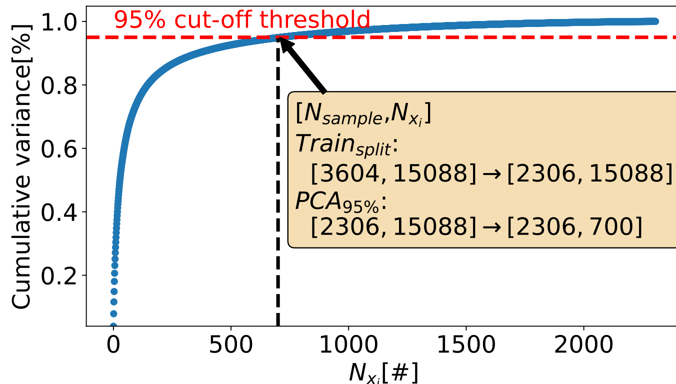

Figure 2: Cumulative variance against the number of features (N_{x_i}), with the data reduction process explained in the box.

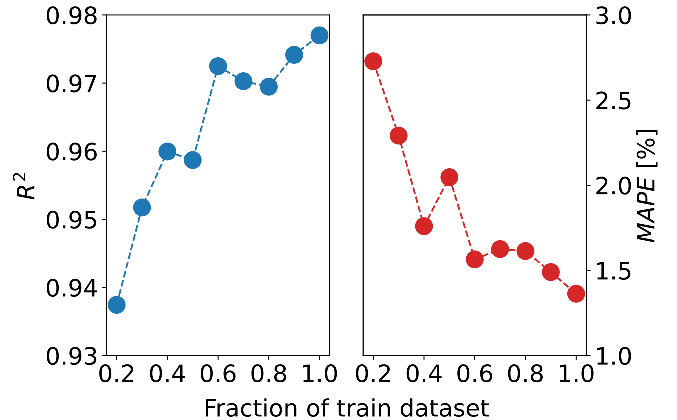

Figure 5: The coefficient of determination (R^2) and the mean square percentage error ($MAPE$) as a function of the training data size.