2022 23rd International Symposium on Quality Electronic Design (ISQED) | 978-1-6654-9466-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISQED54688.2022.9806233

Machine Learning Approach to Characteristic Fluctuation of Bulk FinFETs Induced by Random Interface Traps

Rajat Butola[1,2], Yiming Li[1-6, *], and Sekhar Reddy Kola[1,2]

[1]Parallel and Scientific Computing Laboratory, [2]Electrical Engineering and Computer Science International Graduate Program, [3]Institute of Communications Engineering, [4]Department of Electrical Engineering and Computer Engineering, [5]Institute of Biomedical Engineering, and [6]Center for mmWave Smart Radar System and Technologies, National Yang-Ming Chiao Tung University, Hsinchu 300, Taiwan, 1001 Ta-Hsueh Rd., Hsinchu 300, Taiwan; [*]Tel: +886-3-5712121 ext. 52974; Fax: +886-3-5726639; Email: ymli@nycu.edu.tw

Abstract

Interface traps are of particular concern for highly scaled-down semiconductor devices. They cause trapping and de-trapping of charge carriers and have an adverse effect on device characteristics and variability. Therefore, in this work, the influence of randomly generated interface traps (RITs) on device characteristics of 16-nm-gate high-κ/metal gate bulk fin field-effect transistors (FinFETs) is investigated for experimentally validated simulated data. A machine learning (ML) model is proposed here to imitate the device simulation results. The impact of variation of these multi-point defects is analyzed by generating RITs at the interface of gate-oxide and silicon channel of the explored bulk FinFETs. The statistical fluctuations induced by RITs are analyzed by predicting the variations in threshold voltage (V_{TH}), subthreshold slope (*SS*), drain-induced barrier lowering (*DIBL*), off-state current (I_{OFF}), and transconductance (g_m) using the proposed ML model with high accuracy and small error, in much less computational cost. This work shows the possibility of accelerating the random defects analysis using the technique of machine learning.

Keywords

Bulk FinFET, characteristic fluctuation, interface trap, machine learning, random forest regressor.

1. Introduction

In the semiconductor fabrication industry, the ever-increasing demand for higher density, better performance, and low power consumption led the semiconductor planar devices to continuous scaling [1], [2]. However, non-stop scaling of planer device technology encountered hard challenges that leads to objectionable effects such as short channel effects (SCE), increased leakage current, and fluctuation of characteristics of these devices [3]. Control of SCE is central since it allows lower operating voltages and shorter channel lengths. The planar devices don't scale efficiently beyond 30-nm lengths, so the FinFETs emerged as their substitute. The FinFET devices because of their unique three-dimensional (3-D) gate structure overcome these issues up to a great extent [4]. FinFET blocks SCE and has lower switching times as well as higher current density than a planar transistor and makes further scaling of transistor possible. But the continuous shrinking of FinFET devices makes them vulnerable to random fluctuations such as work function fluctuations (WKF), line edge roughness (LER), random dopant fluctuations (RDF), and interface traps. In past, these fluctuations are studied in detail. For example, WKF-induced variability in GAA FETs has been reported [5]. Similarly, in [6] the impact of different types of LER has been analyzed for 14-nm-gate high-κ/metal gate (HKMG) trapezoidal bulk FinFET. The effect of RDF on variability of threshold voltage of InGaAs nMOSFETs has been reported in [7]. Impact of RITs at the interface of SiO_2/Si on the electrical characteristic of the 16-nm HKMG bulk FinFET devices is investigated [8].

Recently, ML has been gaining popularity in the semiconductor fabrication industry [9]-[11]. The idea, however, is not new [12]. There is always a motivation for using ML in semiconductor manufacturing. The application of ML has also explored the effects of random fluctuations in FinFET transistors. An artificial neural network is developed to investigate the effect of LER-induced random variation on the input/output transfer characteristics of 5-nm FinFETs [13]. Similarly, the authors in [14] have analyzed the impact of WKF on gate-all-around silicon nanosheet MOSFET using three different deep learning algorithms.

In this work, the impact of RITs on device characteristics of 16-nm-gate HKMG bulk FinFETs is examined. The 3-D interface traps are generated randomly at the gate-oxide/silicon channel interface of FinFETs. These traps affect the drain current due to the capture and emission of charge carriers in the channel [15]. A regression-based ML model is proposed here which predicts the RITs induced key characteristics fluctuations. The rest of the paper is outlined as follows: Section 2 presents the bulk FinFET device structure, simulation methodology, and generation of RITs. Section 3 demonstrates the machine learning modeling. In Section 4, results obtained from the proposed ML model are reported and at last, Section 5 concludes this study.

2. Device structure and simulations

A 3-D schematic of the bulk FinFET structure used in this study is shown in Fig. 1(a). The detailed structural information of the device is as follows: the channel length (L_G) 16-nm, the fin height (H_{FIN}) 32-nm, the fin width (W_{FIN}) 8-nm, the aspect ratio 4, and the effective oxide thickness (*EOT*) is 0.95 nm and calculated by the expression:

$$EOT = t_{SiO_2} + t_{\mathrm{HfO}_2} \times \frac{\varepsilon_{SiO_2}}{\varepsilon_{\mathrm{HfO}_2}} \quad (1)$$

where t_{SiO_2} and t_{HfO_2} are thicknesses of SiO_2 gate oxide and high-κ/metal gate. Similarly, ε_{SiO_2} and ε_{HfO_2} are the dielectric constants for SiO_2 gate oxide and HfO_2. In addition to this, the

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on May 03,2024 at 02:54:55 UTC from IEEE Xplore. Restrictions apply.

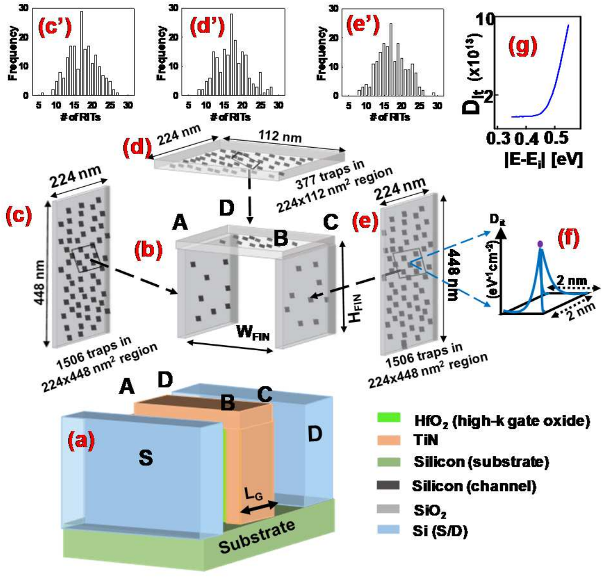


Figure 1: (a) A 3-D schematic of the explored Si bulk FinFET with the high-k-metal-gate. (b) RITs generated on the interface of gate oxide/channel. (c)-(e) show three different sides with RITs. The concentration of each RIT is approximately 1.5×10^{12} cm^{-2} ev^{-1} for all planes, and is driven by the Poisson distribution. (c')-(e') show the Poisson distribution of the statistically generated RITs for each plane. (g) The density of each trap (D_{it}) on the plane is assigned according to distribution of trap's energy.

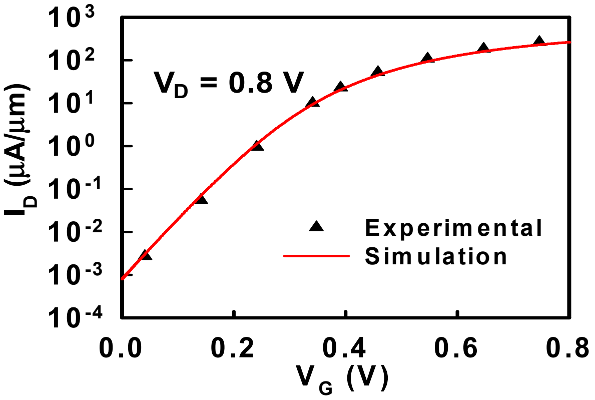


Figure 2: An illustration of validation of device simulated data (solid red line) with experimentally measured data (black triangle dots) through proper calibration.

channel doping concentration is 5×10^{17} cm^{-3} and the source/drain doping concentration of device is 1×10^{20} cm^{-3}. Table I lists all the structural and simulation parameters used for the bulk FinFET device. The RITs are generated arbitrarily at the HfO$_2$/SiO$_2$ and Si channel interface using the statistical generation simulator [Fig. 1. (b)], to study their impact on the electrical characteristics of the bulk FinFETs. The simulation flow of RITs is as follows: A total of 1506 traps are generated in two large areas of 224 x 448 nm^2 2-D planes for side gates of bulk FinFET, as depicted in Figs. 1(c) and (e). Similarly, in Fig. 1(d), 377 traps are generated in 224 x 112 nm^2 area for the top fin. Each 3-D trap is of dimension 2-nm (width), 2-nm (length), and 0.5-nm (depth) and the density of trap is 1.5×10^{13} cm^{-2} eV^{2}. The energy of each trap is varied randomly and it is distributed according to trap density. The standard deviations for the side fins are 4.0681, 4.4327 and for the top fin, it is 2.0241, as depicted in Figs. 1(c'), (e'), and (d') respectively. The top fin and side fins of bulk FinFET are partitioned into small sub-planes. The top and side fin sub-planes are of size 16 x 8 nm^2 and 32 × 8 nm^2 respectively. In this way, the 4500 devices are generated for statistical device simulations to measure the impact of RITs variabilities on DC characteristics.

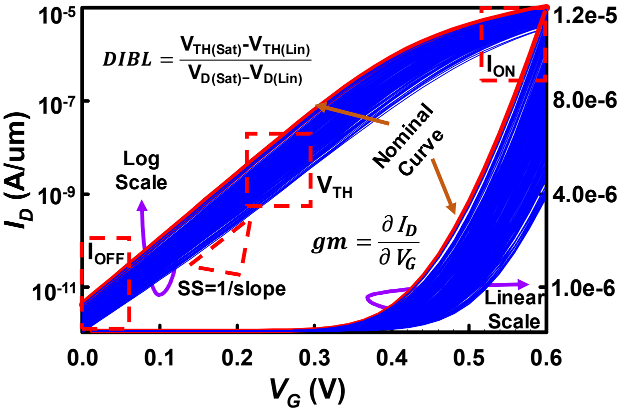


Figure 3: The I_D-V_G characteristics of the 4500 simulated FinFET devices. The red curve shows the nominal case (i.e., the FinFET without traps) and the blue curves show the FinFET devices with traps, in logarithmic and linear scales.

Table I: List of bulk FinFET device parameters, regions, and their specified values used in the 3-D device simulation.

Device parameter	Value
Channel length (nm) (L_G)	16
Channel doping (cm^{-3})	5×10^{17}
Fin height (nm) (H_{FIN})	32
Fin width (nm) (W_{FIN})	8
Aspect ratio (H_{FIN} / W_{FIN})	4
Work function (eV)	4.52
S/D doping (cm^{-3})	1×10^{20}
Density of interface trap (cm^{-2} eV^{-1}) (D_{it})	1.5-8.5 x 10^{12}
Interface trap energy (eV)	0.35-0.55
effective oxide thickness (nm) (*EOT*)	0.95
S/D extension doping (cm^{-3})	4.8×10^{18}

Fig. 2 shows the validation of the I_D-V_G characteristic of the simulated data with experimental Si data before generating the large dataset for machine learning. Fig.3 shows the I_D-V_G characteristics for the nominal device (without traps) and devices with traps, in linear and logarithmic scale. In the figure, the fluctuations in I_D-V_G characteristics can be clearly observed. The figures of merit (FoM) analyzed in this paper, are also marked in curves. The I_{OFF} is the current where V_G=0V and I_{ON} is the current at $V_G = V_D + V_{TH}$. The slope below the threshold voltage region is called sub-threshold slope (*SS*). The formulas used to calculate two important FoM, *DIBL* and g_m are also shown in the figure.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on May 03,2024 at 02:54:55 UTC from IEEE Xplore. Restrictions apply.

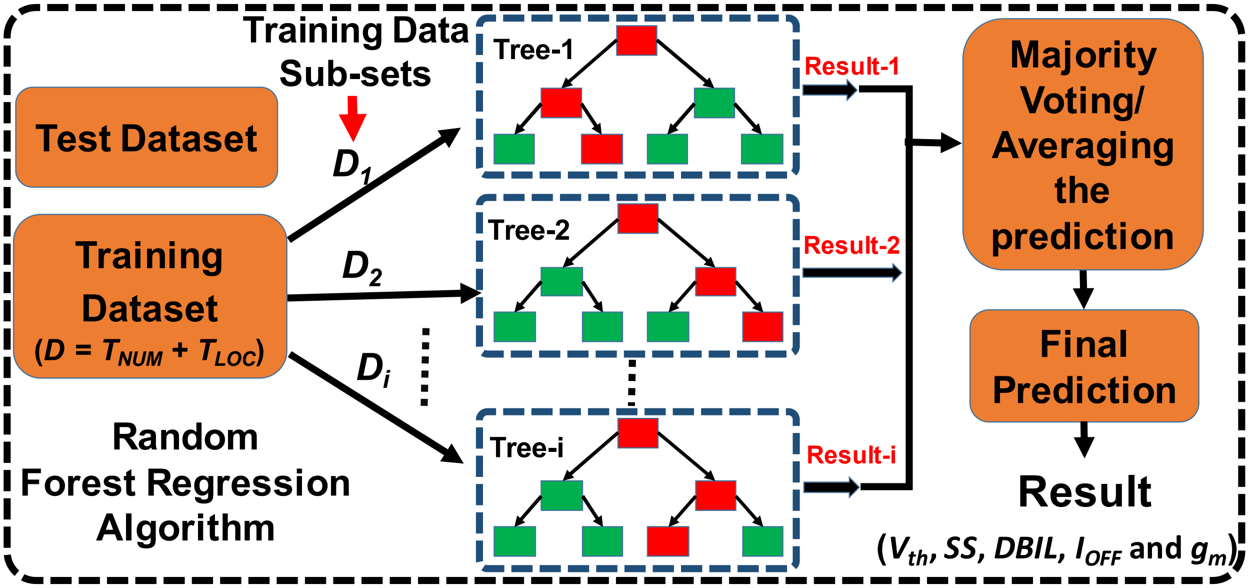

Training Data Sub-sets
Test Dataset
Training Dataset
($D = T_{NUM} + T_{LOC}$)
D_1
D_2
D_i
Random Forest Regression Algorithm
Tree-1
Tree-2
Tree-i
Result-1
Result-2
Result-i
Majority Voting/ Averaging the prediction
Final Prediction
Result
(V_{th}, SS, $DBIL$, I_{OFF} and g_m)

Figure 4: The workflow demonstration of the random forest regression model building and its implication on data. The model operates by constructing several decision trees during training time and it utilizes the ensemble learning technique that uses averaging to improve the predictive accuracy. The training data consists of random numbers (T_{NUM}) and random locations (T_{LOC}) of RITs as inputs and the electrical parameters (V_{th}, SS, $DIBL$, I_{OFF} and g_m) of the explored device as the outputs.

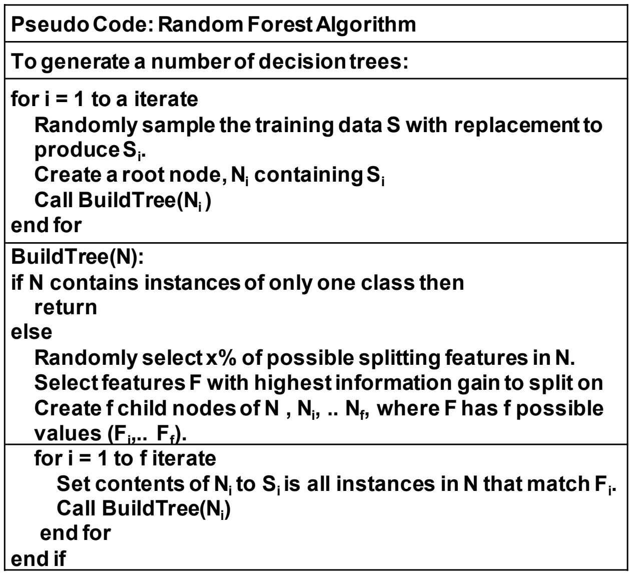
Table II: List of hyperparameters with values utilized in the random forest regressor algorithm.

RFR Hyperparameters	Value
n_estimators	250 (No. of trees)
random state	42
min_samples_split	2
min_samples_leaf	1
max_depth	None

3. Machine learning modeling

For this task, we use a regression-based supervised machine learning algorithm which is mainly used for predicting continuous values. The purpose of the ML model is to develop an automatic, fast, and reliable data-driven model that can associate the device parameters to device characteristics. The regression model, random forest regressor (RFR) [16], is used in this work from the scikit-learn library [17]. It is a flexible, reliable, and easy-to-use ML algorithm that generates great results with minimal hyperparameter tuning requirements. RFR is based on the ensemble learning method and consists of several decision trees also called "forest". It considers the outputs of these trees individually and takes one (which gets the majority of votes) as a selected prediction as shown in Fig. 4. This prediction is the most accurate and stable output.

The RFR model is trained with the bagging method. This method combines the learning of models to increase the overall result. The simulated dataset created for this task has 4500 samples. The DC characteristics are extracted from simulated data. The dataset is split into the train (80%) and test (20%) components. The number of RITs (T_{NUM}) and their locations (T_{LOC}) are chosen as input features for the ML model to investigate their impact on the electrical parameters (V_{th}, SS, $DIBL$, $I_{OFF,}$ and g_m). The fluctuations induced due to RITs are predicted as outputs. The training data is first normalized between the range of zero and one to increase the convergence speed and to make the model output more precise. Then, the training is done to teach the ML model from the normalized data and to enhance its predictive power for ensuring high accuracy. The hyperparameters that are selected to optimize the ML model are listed in Table II and are selected manually. The pseudocode for the RFR algorithm is explained in Fig. 5. After successful learning, the well-trained model is verified on unseen test data in order to evaluate accuracy of RFR model.



```
Pseudo Code: Random Forest Algorithm
To generate a number of decision trees:
for i = 1 to a iterate
    Randomly sample the training data S with replacement to
    produce $S_i$.
    Create a root node, $N_i$ containing $S_i$
    Call BuildTree($N_i$ )
end for
BuildTree(N):
if N contains instances of only one class then
    return
else
    Randomly select x% of possible splitting features in N.
    Select features F with highest information gain to split on
    Create f child nodes of N , $N_i$, .. $N_f$, where F has f possible
    values ($F_i$,.. $F_f$).
    for i = 1 to f iterate
        Set contents of $N_i$ to $S_i$ is all instances in N that match $F_i$.
        Call BuildTree($N_i$)
     end for
end if
```

Figure 5: A list of the pseudocode is used by trained RFR algorithm to perform prediction. Each random tree predicts a different target for same test feature and then each target is calculated and considered as vote to select final output.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on May 03,2024 at 02:54:55 UTC from IEEE Xplore. Restrictions apply.

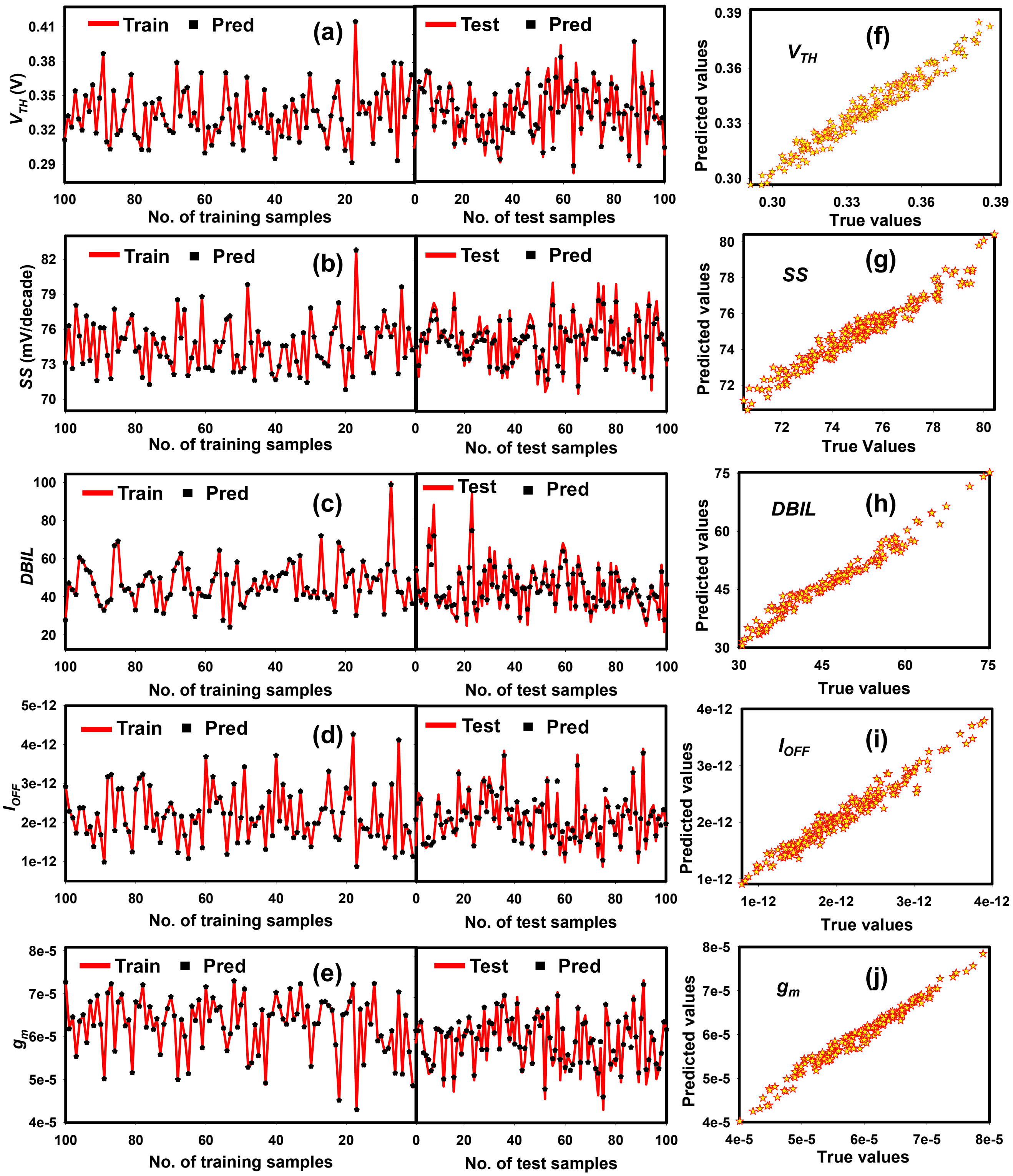


Figure 6: A comparison of the simulated data and the ML model predicted output values for five different key electrical parameters, i.e., V_{TH}, SS, $DIBL$, I_{OFF} and g_m is illustrated, (a)-(e) for training and testing datasets, respectively. The solid line represents the simulated values and the markers represent the predicted values. (f)-(j) show that the ML model predicted test outputs are in straight line between the two axis where simulated values (Y_i) = predicted values ($\hat{Y}$).

4. Result and discussion

Here, the use of the ML algorithm is demonstrated to predict the variations in the electrical parameters (V_{TH}, SS, $DIBL$, $I_{OFF,}$ and g_m). The T_{NUM} and T_{LOC} are given as inputs to the ML model. After successful training, the model is evaluated for the test dataset. Root mean squared error (*rmse*) is considered as the loss function in a regression problem, which is defined as the root of the average squared difference between the

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on May 03,2024 at 02:54:55 UTC from IEEE Xplore. Restrictions apply.

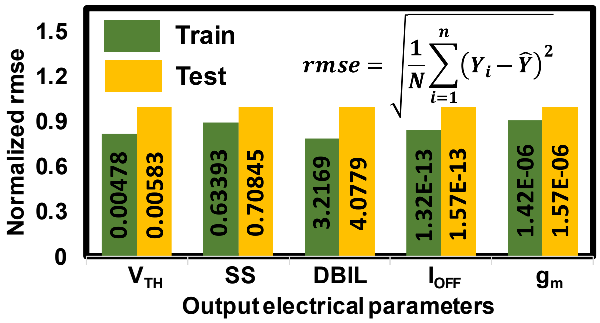


Figure 7: The quantitative assessment using *rmse* values. A comparison between training and test datasets of all the electrical parameters (V_{th}, *SS*, *DIBL*, I_{OFF} and g_m) is shown using histograms after normalizing values by test values.

predicted values and true simulated values. In regression problems, *rmse* measures the deviation of the predicted values with respect to the true values given as:

$$\text{rmse} \;=\; \sqrt{\frac{1}{N}\sum_{\text{i}\,=\,1}^{\text{n}}\left(Y_i - \hat{Y}\right)^2}, \qquad (2)$$

where Y_i *and* $\hat{Y}$ are the true simulated and the predicted values, respectively. The ML model is considered well trained if the *rmse* value is very small which means the predicted regression values of the model are close to the real values.

The ML algorithms generally have a preconceived degree of randomness in the training process and initializing procedure. Therefore, a ML model may produce different output patterns, even if it runs with the same input data, depending on the random initial condition [18]. To improve the performance of the model so that its convergence can lead to the global minimum, a random starting point is often necessary. To ensure the known randomness of the model, a quantity, "random seed" is chosen. In our case, it is set to 35. It makes possible the reproducibility of the result. This is an initial point for a sequence, and it guarantees that the trained model will predict the same output every time if it starts from the same seed number.

There are a lot of causes that affect the accuracy of any ML model. Overfitting is one of those causes in ML that hampers the performance as well as the accuracy of the model. To counter overfitting, in the RFR ML model, there are some hyperparameters that must be selected carefully to minimize the risk of overfitting. One such hyperparameter is "n_estimators" which selects the number of trees in the forest. The higher number of trees gives better performance and makes the model less likely to overfit but they also make the code slower. So, it must be selected according to the processor used. To ensure that model does not overfit, we select "n_estimators" as 250 before taking the maximum voting or averages of predictions. This makes the predictions of the RFR model stronger and more stable.

The depth of the RFR model also causes overfitting if it does not prune accurately. It defines the longest path from the root node to the leaf node in the forest. If this value is too large then the model will perform efficiently for the training dataset but will perform poorly over an unseen test dataset and the model, therefore will not be able to generalize due to overfitting. So, the depth of the RFR model must be limited to a value where it can keep away from the overfitting problem. It also reduces the complexity of the ML model. For this work, the depth is set to 20.

Once the model is trained and optimized, it is used to predict the desired output. The fluctuations in V_{th}, *SS*, *DIBL*, $I_{OFF,}$ and g_m induced by RITs are predicted as output using the ML model according to the random number T_{NUM} and random position T_{LOC} of RITs. The results of the ML model for training and testing datasets of different parameters are shown in Figs. 6 (a)-(e). The curves are plotted between the values of the parameters in the y-axis and training and testing data samples in the X-axis. The first 100 training and testing samples results are plotted for better visualization and understanding. In the left side figures, the training simulated data (red line) and ML predicted training data (black dots) are shown. Similarly, in the right side figures, testing simulated data (red line) and ML predicted test data (black dots) are shown. From the figures, it can be seen that the prediction by the ML model is in good agreement with the simulated values. The prediction accuracy of the ML model for training data is approximately 98-99% for all the parameters. The test results by ML model, on the other hand, specially for V_{th}, I_{OFF} and g_m are very accurate and the predicted values are overlapping with the simulated values. There are just a few outliers when the model is predicting the *SS* and *DIBL* parameter values but the predicted values are still close to the true simulated values.

The predictive capability of the model can be confirmed by Figs. 6 (f)-(j). The plots show that the predicted test values are in a straight line between the two axes where simulated values Y_i= predicted values $\hat{Y}$. It is examined that ML model outputs follow the simulated outputs. From Fig. 7, the quantitative assessment of the ML model using *rmse* can be evaluated. The *rmse* values comparison between train and test dataset of all the electrical parameters is shown using histograms after normalizing the values by test values. it can be seen that the test *rmse* values are close to the train values. It shows that what the model learns during training, it is able to use that learning to deduce the outputs on the test dataset. Hence, the model successfully overcomes the overfitting problem. The real *rmse* values are also shown in the figure.

It is also worth mentioning that, the simulation took approximately 37 days to generate 4500 samples (12 mins/sample) whereas with the same computing resources the RFR model is trained in only 240 seconds. It shows that modeling using ML is faster than simulation. In addition to this, once the training of the ML model is done, a large dataset can be derived quickly without further training process. This confirms the feasibility of using a ML model to predict the semiconductor processing results. The results also show that the ML model, after proper training and optimization, proved to be a powerful tool to obtain accurate results rapidly.

5. Conclusions

In this paper, the regression-based ML model is used successfully to imitate the results of simulated data to investigate the effects of RITs on electrical characteristics (V_{th}, *SS*, *DIBL*, $I_{OFF,}$ and g_m) of 16-nm HKMG bulk FinFETs. This work bridge the gap between past physical model-based

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on May 03,2024 at 02:54:55 UTC from IEEE Xplore. Restrictions apply.

approaches and machine learning models in analyzing the defects that occurred due to random fluctuations at the device level. Also, the ML approach is completely data-driven and device physics-free, therefore, deep knowledge of device physics is not required to apply the ML algorithm. It is shown here that, the effects of random numbers and random locations of traps on various electrical parameters (V_{th}, *SS*, *DIBL*, $I_{OFF,}$ and g_m) variations are predicted successfully.

The model is designed easily using the ensemble technique. Once the model is trained, it is used to predict desired output without additional training of the model and large data can be obtained quickly. A total of 4500 samples are simulated and used for training and testing of the ML model. The results demonstrated that the predicted output values have strong agreement with simulated values. The error function is designed using *rmse*. The low training and test *rmse* values are achieved for V_{th}, *SS*, *DIBL*, $I_{OFF,}$ and g_m which proves that the model performs fairly well and without overfitting on data. The ML approach is very efficient and less time-consuming as compared to its simulation counterpart. With these advantages, the ML provides a great possibility to accelerate semiconductor development technology. In future work, the model can also be further utilized to analyze the effects of other fluctuations such as WKF, LER, and RDF for different semiconductor devices.

Acknowledgement

This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant MOST 110-2221-E-A49-139, Grant MOST 109-2221-E-009-033, and Grant MOST 109-2634-F-009-030, and in part by the "Center for mm Wave Smart Radar Systems and Technologies" under the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan.

References

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

[2]. S. R. Kola, Y. Li, and N. Thoti, "Effects of a dual spacer on electrical characteristics and random telegraph noise of gate-all-around silicon nanowire p-type metal-oxide-semiconductor field-effect transistors," Jpn. J. Appl. Phys. vol. 59, no. SG, pp. SGGA02, Jan. 2020.

[3]. R. H. Dennard, J. Cai, and A. Kumar, "A Perspective on Today's Scaling Challenges and Possible Future Directions," Solid-state Electronics, vol. 51, no. 4, pp. 518–525, Apr. 2007.

[4]. Y. Li, H.T. Chang, C.N. Lai, P.J. Chao, and C.Y. Chen, "Process Variation Effect Metal-Gate Work-Function Fluctuation and Random Dopant Fluctuation of 10-nm Gate-All-Around Silicon Nanowire MOSFET Devices," in IEDM, Washington, DC, pp. 887-890, Dec. 2015.

[5]. E. Mohapatra, T. P. Dash, S. Das, J. Jena, J. Nanda, and C. K. Maiti, "Investigation of Work Function Variation on the Electrical Performance of sub-7nm GAA FETs," Devices for Integrated Circuit (DevIC), pp. 103-106, May 2021.

[6]. C. Chen, W. Huang, and Y. Li, "Electrical characteristic and power consumption fluctuations of trapezoidal bulk FinFET devices and circuits induced by random line edge roughness," International Symposium on Quality Electronic Design (ISQED), pp. 61-64, Mar. 2015.

[7]. N. Zagni, F. M. Puglisi, P. Pavan, and G. Verzellesi, "Random dopant fluctuation variability in scaled InGaAs dual-gate ultra-thin body MOSFETs: Source and drain doping effect," IEEE International Integrated Reliability Workshop (IIRW), pp. 1-4, Oct. 2017.

[8]. S. C. Hsu, and Y. Li, "Electrical characteristic fluctuation of 16-nm-gate high-κ/metal gate bulk FinFET devices in the presence of random interface traps," Nanoscale Res Lett., vol. 9, no. 633, Nov. 2014.

[9]. T. Wu and J. Guo, "Speed Up Quantum Transport Device Simulation on Ferroelectric Tunnel Junction with Machine Learning Methods," in IEEE Transactions on Electron Devices, vol. 67, no. 11, pp. 5229-5235, Nov. 2020.

[10]. J. Kim, S. J. Kim, J. -W. Han and M. Meyyappan, "Machine Learning Approach for Prediction of Point Defect Effect in FinFET," in IEEE Transactions on Device and Materials Reliability, vol. 21, no. 2, pp. 252-257, June 2021.

[11]. M. S. Abrishami, H. Ge, J. F. Calderon, M. Pedram, and S. Nazarian, "NN-PARS: A Parallelized Neural Network Based Circuit Simulation Framework," International Symposium on Quality Electronic Design (ISQED), pp. 452-456, Mar. 2020.

[12]. K. B. Irani, J. Cheng, U. M. Fayyad, and Z. Qian, "Applying machine learning to semiconductor manufacturing," in IEEE Expert, vol. 8, no. 1, pp. 41-47, Feb. 1993.

[13]. J. Lee, T. Park, H. Ahn, J. Kwak, T. Moon, C. Shin, "Prediction Model for Random Variation in FinFET Induced by Line-Edge-Roughness (LER)," Electronics, vol. 10, no. 455, Feb. 2021.

[14]. C. Akbar, Y. Li, and W. L. Sung, "Deep Learning Algorithms for the Work Function Fluctuation of Random Nanosized Metal Grains on Gate-All-Around Silicon Nanowire MOSFETs," in IEEE Access, vol. 9, pp. 73467-73481, May 2021.

[15]. S. R. Kola, Y. Li, and N. Thoti, "Random telegraph noise in gate-all-around silicon nanowire MOSFETs induced by a single charge trap or random interface traps," J Comput Electron, vol. 19, pp. 253–262, Jan. 2020.

[16]. L. Breiman, "Random Forests", Machine Learning, vol. 45, pp. 5–32, Oct. 2001.

[17]. F. a. V. Pedregosa, G. and Gramfort "Scikit-learn: Machine Learning in Python," JMLR, vol. 12, pp. 2825-2830, 2011.

[18]. T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins, and N. Khovanova, "Machine Learning for Predictive Modelling based on Small Data in Biomedical Engineering," IFAC-Papers Online, vol. 48, no. 20, pp. 469–474, Nov. 2015.

Authorized licensed use limited to: National Yang Ming Chiao Tung University. Downloaded on May 03,2024 at 02:54:55 UTC from IEEE Xplore. Restrictions apply.