Nuclear Science and Techniques (2023) 34:192
https://doi.org/10.1007/s41365-023-01340-x

A machine learning approach to TCAD model calibration for MOSFET

Bai-Chuan Wang[1,2,3] · Chuan-Xiang Tang[1,2] · Meng-Tong Qiu[3] · Wei Chen[3] · Tan Wang[3] · Jing-Yan Xu[3] · Li-Li Ding[3]

Received: 1 January 2023 / Revised: 8 May 2023 / Accepted: 27 July 2023 / Published online: 7 December 2023
© The Author(s), under exclusive licence to China Science Publishing & Media Ltd. (Science Press), Shanghai Institute of Applied Physics, the Chinese Academy of Sciences, Chinese Nuclear Society 2023

Abstract
Machine learning-based surrogate models have significant advantages in terms of computing efficiency. In this paper, we present a pilot study on fast calibration using machine learning techniques. Technology computer-aided design (TCAD) is a powerful simulation tool for electronic devices. This simulation tool has been widely used in the research of radiation effects. However, calibration of TCAD models is time-consuming. In this study, we introduce a fast calibration approach for TCAD model calibration of metal–oxide–semiconductor field-effect transistors (MOSFETs). This approach utilized a machine learning-based surrogate model that was several orders of magnitude faster than the original TCAD simulation. The desired calibration results were obtained within several seconds. In this study, a fundamental model containing 26 parameters is introduced to represent the typical structure of a MOSFET. Classifications were developed to improve the efficiency of the training sample generation. Feature selection techniques were employed to identify important parameters. A surrogate model consisting of a classifier and a regressor was built. A calibration procedure based on the surrogate model was proposed and tested with three calibration goals. Our work demonstrates the feasibility of machine learning-based fast model calibrations for MOSFET. In addition, this study shows that these machine learning techniques learn patterns and correlations from data instead of employing domain expertise. This indicates that machine learning could be an alternative research approach to complement classical physics-based research.

Keywords Machine learning · Radiation effects · Surrogate model · TCAD model calibration

1 Introduction

Technology computer-aided design (TCAD) is a powerful simulation tool for electronic devices. This simulation tool has been widely used in the research of radiation effects [1–4]. To obtain reliable simulation results, TCAD models should be calibrated in advance [5–8]. Calibration of TCAD models is essential for all simulation studies [9–11]. Generally, the structure and doping parameters of TCAD models are adjusted to make the simulated current–voltage curves consistent with the process design kit (PDK) results, while some parameters should be in accordance with the PDK information [12–14]. Calibration is time consuming because TCAD simulations are slow and need to be performed iteratively. The calibration procedure typically requires several weeks or more for manual adjustments.

Evolutionary methods, such as genetic algorithms, are possible approaches for automatic calibration [15–18]. However, such methods require a cold start for each task. Even a small change in the calibration goal requires repeating all the simulations in the evolution process.

Currently, machine learning methods provide another possible approach for fast calibration. Once trained, the machine learning-based surrogate model can serve as a quick tool for a variety of tasks within its scope. This method has been adopted to accelerate time-consuming scientific simulations in many research fields, such as partial differential equation solving [19], nanostructure design [20], thermal

This work was supported by the National Natural Science Foundation of China (Nos. 11690040 and 11690043).

✉ Li-Li Ding
lili03_ding@126.com

[1] Department of Engineering Physics, Tsinghua University, Beijing 100084, China

[2] Key Laboratory of Particle and Radiation Imaging (Tsinghua University),, Ministry of Education, Beijing 100084, China

[3] State Key Laboratory of Intense Pulsed Radiation Simulation and Effect, Xi'an 710024, China

metamaterial design [21], and diode failure troubleshooting [22]. The trained machine learning-based surrogate models are typically several orders of magnitude faster than the original scientific simulators. However, to the best of our knowledge, machine learning-based TCAD model calibration for metal–oxide–semiconductor field-effect transistors (MOSFETs) has not been reported in the literature. We believe that machine learning-based fast tools will be widely adopted in the future. In this paper, we propose a machine learning approach for fast calibration of the TCAD model and provide a corresponding calibration tool for MOSFET using Python script. MOSFETs are basic components of modern CMOS integrated circuits. We took N-type MOSFET (NMOS) as an example to demonstrate the potential of machine learning methods for fast model calibration.

Three issues need to be addressed when calibrating MOSFETs using machine learning approaches. First, the machine learning-based surrogate model should be widely applicable. Otherwise, every single task requires considerable time to build a new model, and the speed advantage is negated. Second, the validity of the parameter combinations for MOSFETs should be determined to avoid invalid calculations. Third, the important MOSFET parameters for calibration should be identified and focused on.

In our approach, we developed possible solutions to these issues. First, to make the proposed surrogate model more widely applicable, a fundamental model of typical planar MOSFET was introduced. Second, classifiers were introduced to address the validity issues of the parameter combinations. Finally, important parameters were identified using the random forest technique. Their influence on the current–voltage curves was analyzed. A calibration tool based on Python script was developed and tested with different calibration goals for different PDKs. The results indicated that the proposed tool could achieve the desired calibration parameters within several seconds.

We demonstrated a machine learning approach to TCAD model calibration for MOSFET and demonstrated its great advantage in terms of speed. We believe that this approach will become popular in solving similar problems in the near future. In addition, we demonstrated that this data-driven approach could be a new method of identifying valid parameter combinations and important parameters without the help of domain expertise. These results could be referenced for further physical analyses.

2 TCAD simulations and datasets

2.1 TCAD model for MOS transistors

A fundamental TCAD model containing 26 parameters was introduced to represent the typical structure of MOSFETs. As depicted in Fig. 1, a common planar MOSFET includes doping distributions in various regions. The calibration results of this TCAD model can be referenced for further detailed calibrations or directly applied in preliminary simulations of radiation effects. The TCAD model includes source/drain doping (SD), low-doped drain (LDD), halo doping, and channel doping. Specifically, channel doping consists of three parts: the doping concentration is homogeneous in the middle part and Gaussian in the top and bottom parts. The Gaussian peaks of the top and bottom parts are located at their respective boundaries with the middle part. Their peak values are equal to the concentration in the middle part. The 26 parameters listed in Table 1 are used to describe the MOSFET model. These parameters control the key dimensions and doping concentration. Six of these can be obtained from the PDK information: gateLen, gateWidth, tox, sd_peak, sd_depth, and V_{dd}. During calibration, these parameters should be assigned according to the PDK information, and the other 20 parameters need to be adjusted.

2.2 TCAD simulations and calibration goals

The physical models listed in Table 2 were used in TCAD simulations. The calibration targets were the $I_d - V_g$ curves provided by the PDK. Figure 2 shows typical $I_d - V_g$ curves for an NMOS transistor. V_d was set to a low voltage ($V_d = 0.1$ V) and the working voltage. Both the linear and semi-logarithmic scales of the curves should be well calibrated. This is because the conduction characteristics of $I_d - V_g$ curves are easy to check on a linear scale, whereas the subthreshold characteristics are easy to check on a semi-log scale.

Three metrics were extracted to describe the $I_d - V_g$ curves: threshold voltage V_t, transconductance G_m [23],

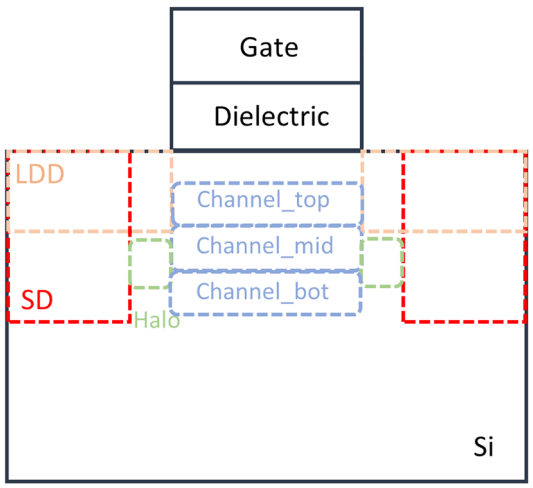


Fig. 1 Schematic of the TCAD model to be calibrated

Table 1 Parameters of the TCAD model to be calibrated

Number	Parameter	Description
1	workF	Work function of gate material
2	sub_const	Doping concentration of substrate
3	well_const	Doping concentration of well
4	ch_const	Doping concentration of channel middle part
5	ch_depth_a	Depth of channel top part
6	ch_depth_b	Depth of channel bottom part
7	ch_factor_a	Gaussian factor of channel top part
8	ch_factor_b	Gaussian factor of channel bottom part
9	ch_position_a	Beginning position of channel top part
10	ch_depth_const	Depth of channel middle part
11	ldd_peak	Peak doping concentration of LDD
12	ldd_depth	Depth of LDD
13	ldd_factor	Gaussian factor of LDD
14	sd_peak	Peak doping concentration of source and drain
15	sd_depth	Depth of source and drain
16	wellc_peak	Peak doping concentration of well contact
17	halo_peak	Peak doping concentration of halo
18	halo_depth	Depth of halo
19	halo_factor	Gaussian factor of halo
20	halo_position_z	Beginning position of halo
21	sd_position	X position of source or drain position
22	halo_position_x	X position of halo position
23	gateLen	Gate length
24	gateWidth	Gate width
25	tox	Thickness of the gate dielectric
26	V_d	Drain voltage

Table 2 Physical models used in TCAD simulations

Physical model	Value
Hydrodynamic	eTemperature
Mobility	DopingDep, HighFieldSaturation, CarrierCarrierScattering, Enormal
EffectiveIntrinsicDensity	BandGapNarrowing
Recombination	SRH, Auger, Avalanche
Temperature	300

and subthreshold slope S [24]. The threshold voltage V_t was defined using the constant-current method [25]. Specifically, in this study, V_t refers to the gate voltage when the drain current reaches 1×10^{-7} A. Transconductance G_m refers to the ratio of the drain current to the gate voltage above the threshold voltage. Subthreshold slope S is computed as $(\mathrm{d}(\log I_d)/\mathrm{d}V_g)^{-1}$, which is the reciprocal of the slope of the I_d–V_g curves in the subthreshold region with the semi-log scale. S^{-1} was computed in this study. The metrics at low and working drain voltages V_d were extracted as the calibration goals.

2.3 Techniques in dataset generation

Generally, the training set is created using simulations with randomly generated parameters [20, 21]. However, two problems should be solved in our case. First, the MOSFET may not function when the parameters are randomly generated. A large number of invalid samples would waste considerable computing time. Second, the number of parameters for the MOSFET model is large, indicating that many training samples are needed to build a machine learning model.

If the validity of parameter combinations could be identified before computation, invalid calculations could be avoided. On the other hand, if the importance of every parameter could be determined, excluding the unimportant parameters could reduce the dimensions of the search space, thereby decreasing the number of required training samples.

Generally, device experts are needed to identify the validity of parameter combinations or key parameters. In this study, machine learning methods were utilized instead of domain expertise. Specifically, classification models were trained to identify the validity of the parameters, and a random forest-based feature selection technique was utilized to determine the important parameters.

2.3.1 Classification for valid combinations of parameters

A TCAD model with randomly generated parameters may not function well or have a threshold voltage outside the range of concern, which cannot produce effective samples. We randomly generated 1000 TCAD samples and found that only 321 samples were valid for creating the dataset. This indicates that the efficiency of training sample generation was only approximately 32%. To improve efficiency, we trained the classification models to predict the validity of the parameters. The parameters had two possible validity values: positive and negative. Positive values indicated that the related parameters were valid for creating the dataset. Only the parameters predicted as valid were sent to the TCAD simulation.

Five types of popular classifiers were trained and compared using the aforementioned 1000-sample dataset. The classifiers include gradient boosting (GB) [26], multilayer perceptron (MLP) [27], random forest (RF) [28], support vector (SV) [29], and stochastic gradient descent (SGD) classifiers [30]. They were implemented using the Scikit-learn Python library [31], which provides off-the-shelf machine learning methods. To make the different features of the dataset comparable in value, the logarithm of the doping concentrations was used, and each feature was normalized

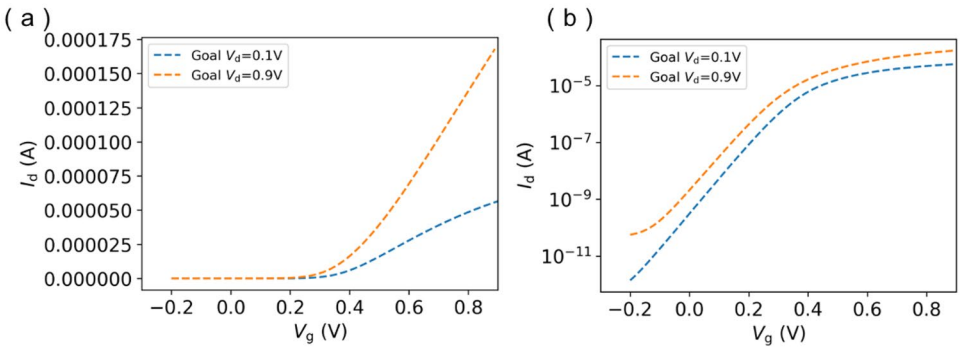


Fig. 2 (Color online) Target $I_d - V_g$ curves calculated by the PDK. **a** Target $I_d - V_g$ curves on linear scale. **b** Target $I_d - V_g$ curves on semi-log scale

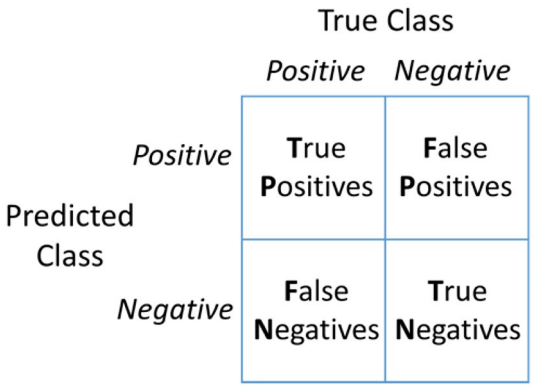


Fig. 3 Schematic of confusion matrix

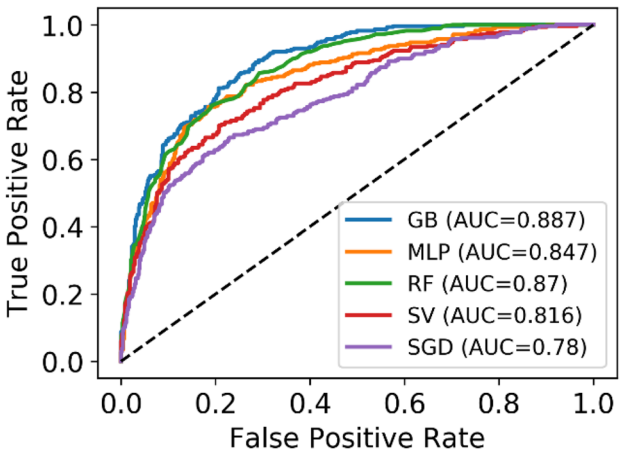


Fig. 4 (Color online) Performance of five classifiers. The dashed line corresponds to the results of a random classifier

to a mean of 0 and standard deviation of 1. The dataset was randomly split into training and test sets in proportions of 90% and 10%, respectively. Generally, the influence of class imbalance starts to become significant when the minority class is less than 10% [32, 33]. Therefore, class imbalance was not considered in this study.

The confusion matrix, shown in Fig. 3, is widely used to assess the performance of classifiers [34]. Some valuable metrics can be calculated from the confusion matrix, such as receiver operating characteristic (ROC) curves [34] and the area under the ROC curve (AUC) [35]. In the confusion matrix, true positives (TPs) refer to correctly predicted positives, true negatives (TNs) refer to correctly predicted negatives, false positives (FPs) refer to negatives which incorrectly classified as positives, and false negatives (FNs) refer to positives which incorrectly classified as negatives.

The AUC was used to measure the performance of the classifiers. AUC refers to the area under the ROC curve. Figure 4 shows the ROC curves and corresponding AUCs for different classifiers. Five-fold cross-validation (CV) was used to generate the ROC curves. In this technique, the training set was randomly divided into five parts. The classifiers were trained by four parts and iteratively validated using the remaining part. The abscissa of the ROC curve is the false positive rate, and the ordinate is the true positive rate. The true positive rate (also called recall) refers to the proportion of correctly predicted positives among all positives, and is computed as $\mathrm{tpr} = TP/(TP + FN)$. The false positive rate refers to the proportion of incorrectly predicted negatives among all negatives, which is computed as $\mathrm{fpr} = FP/(FP + TN)$. A good classifier has an ROC curve that lies at the upper left of the figure, whereas the dashed straight line in Fig. 4 corresponds to the results of a random classifier. The AUC is 1 for an ideal classifier and 0.5 for a random classifier. The performance of the classifiers is influenced by the choice of their hyperparameter values. The suitable hyperparameter values vary for different tasks. The optimal hyperparameter values for the different classifiers after manual tuning are listed in Table 3. Figure 4 shows that the GB classifier had the greatest AUC, indicating that it is more suitable for our task than the other classifiers.

For a given classifier, the precision can be improved by specifying a higher threshold. However, the recall decreases simultaneously. Precision refers to the proportion of correct predictions among all positive predictions, which is computed as $TP/(TP + FP)$. The recall rate was the same as the previously defined true positive rate. The correlation

Table 3 Optimal hyperparameter values for each classifier. The key hyperparameters are listed, whereas the others are equal to the default Scikit-learn values

Classifier	Hyper parameter	Value
GB	n_estimators	200
	max_depth	4
MLP	hidden_layer_size	100
	batch_size	16
RF	n_estimators	200
SV	Kernel	RBF
	Gamma	0.03
	C	2
SGD	Alpha	0.03
	Penalty	L1

between the precision and recall for our GB classifier is shown in Fig. 5. The threshold controls the balance between the two metrics. High-precision classifiers are helpful in improving the efficiency of training sample generation.

A new GB classifier was trained using the entire training set and tested using the test set. The confusion matrix for the test set is presented in Table 4. A precision of 78.1% and recall of 73.5% were achieved.

The final GB classifier was trained using the entire 1000-sample dataset. We randomly generated 40,000 samples, and the GB classifier predicted 11,908 samples to be valid. We calculated these samples using TCAD and found that 9430 samples were valid for creating the dataset. This indicates that the efficiency of the generation of samples is improved to approximately 79.2%, which is approximately equal to the precision of the GB classifier on the test set. The slight improvement in precision may be due to an increase in the number of training samples. These results show that

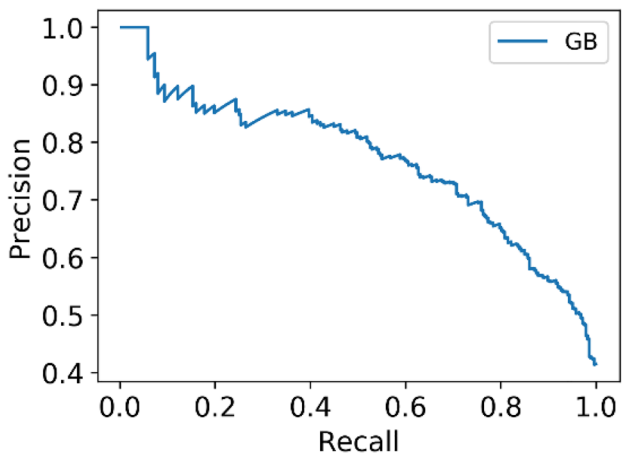


Fig. 5 Precision versus recall curve for GB classifier

Table 4 Confusion matrix of GB classifier on the test set. Precision: 78.1%. Recall: 73.5%

Predicted class	True class		Total
	Positive	Negative	
Positive	25	7	32
Negative	9	59	68
Total	34	66	100

the proposed classifier functions well and saves a significant amount of time in generating the dataset.

2.3.2 Feature selection

We utilized a feature selection technique to identify the important parameters and decrease the dimensions of the TCAD model. The number of parameters for the TCAD model was 26, indicating that a large number of training samples were required to build a machine learning model. Excluding unimportant parameters could reduce the dimensions of the search space, thereby decreasing the required number of training samples.

Random forest regression has proven to be useful in feature selection. This method is helpful in shedding light on the important parameters that govern the output [33]. Random forest refers to a combination of decision trees. Each decision tree in the forest is trained using randomly selected subspaces of the feature space [36]. The final prediction result is obtained by combining the outputs of every tree in the forest [37, 38]. Random forest regression can provide the importance of each input feature. The importance of each feature is typically measured by its influence on the reduction of Gini impurity when training the trees [39]. In our study, the sensitive parameters of the TCAD model were identified using random forest regression. The six parameters specified by the PDK were maintained, and the other 20 parameters were investigated.

The 9430-sample dataset was used to train the RF regression models to evaluate the importance of each parameter for different regression targets, namely V_t, G_m, and S^{-1}. The RF regressions were performed using the Scikit-learn Python library. The importance values of each parameter obtained for the different regression targets are depicted in Figs. 6, 7, 8 and 9. The sum of the importance of each regression target is 1. A higher importance value indicates that the related parameter is regarded as more important by the RF regression. The parameter importance values for different regression targets differed slightly. For threshold voltage, workF and well_const were the most important parameters. For transconductance, well_const and ldd_depth were the most important parameters. For subthreshold slope, ldd_factor and ldd_depth were the

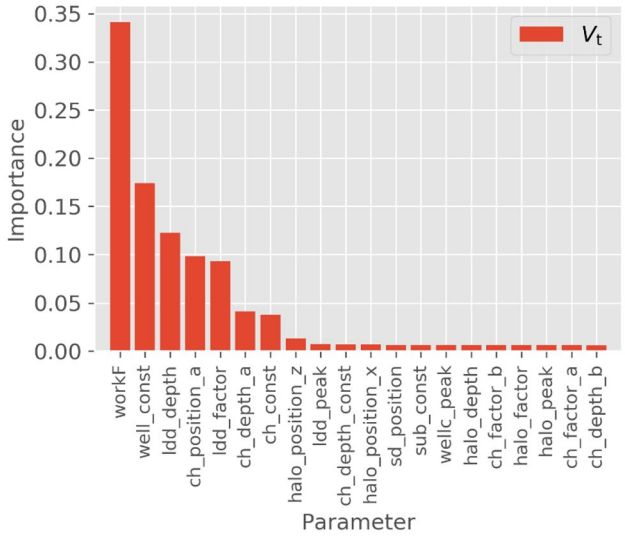


Fig. 6 Parameter importance for threshold voltage regression

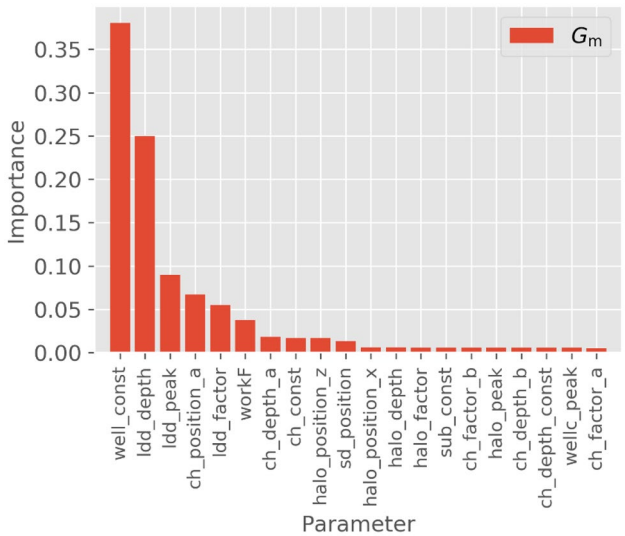


Fig. 7 Parameter importance for transconductance regression

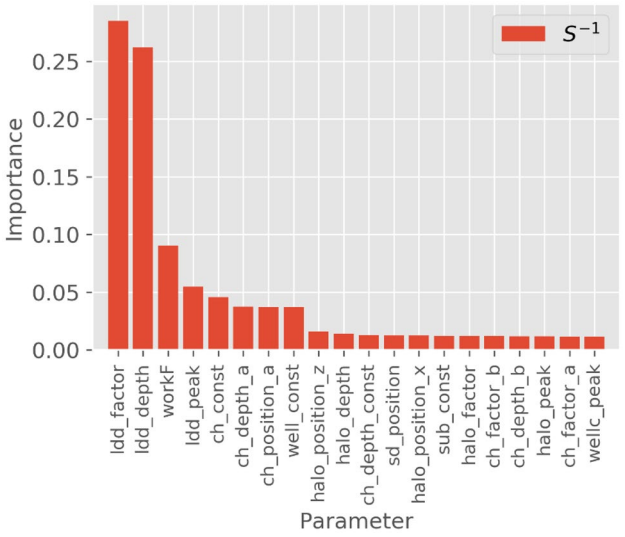


Fig. 8 Parameter importance for subthreshold slope regression

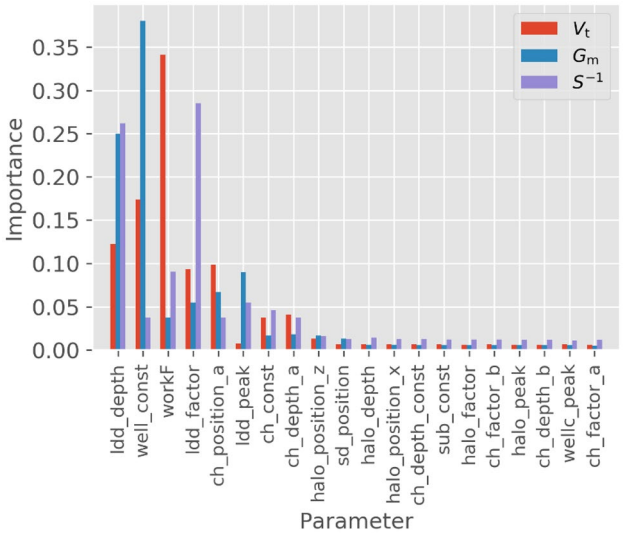


Fig. 9 (Color online) Comparison of parameter importance for different regression targets

most important parameters. The importance of the different regression targets is summarized in Fig. 9. The important parameters were approximately in the same group for the different regression targets. Therefore, the ten most important parameters in Fig. 9, together with six PDK-specified parameters, were selected to build the final model.

The importance of the parameters obtained by the machine learning method is consistent with the results of classical semiconductor theory. In semiconductor theory, when V_d is small, the threshold voltage V_t can typically be approximated as follows [40]:

$$\begin{aligned} V_t = {} & \phi_m - \chi - E_g/(2q) \\ & + kT\ln\left(N_a/n_i\right)/q \\ & + \sqrt{4\varepsilon_{Si}N_a kT\ln\left(N_a/n_i\right)}/C_{ox}, \end{aligned} \tag{1}$$

where ϕ_m is the work function of the gate material, which is equal to the parameter workF in our model. χ is the electron affinity, E_g is the bandgap, q is the electronic charge, k is Boltzmann's constant, T is the absolute temperature, N_a is the acceptor impurity density, n_i is the intrinsic carrier density, ε_{Si} is the silicon permittivity, and C_{ox} is the oxide capacitance per unit area.

Clearly, the threshold voltage V_t is related to the work function of the gate material and the doping density below the dielectric. Therefore, it is consistent with classical semiconductor theories that the workF and well_const parameters are important for controlling the threshold voltage V_t.

However, noting that Eq. (1) applies to simple MOSFETs without channels or LDD doping, it is difficult to obtain an analytical expression for a more complex practical MOSFET. For complex structures, theoretical analysis can qualitatively identify the parameters that may have an impact; however, their impact patterns and importance are difficult to determine. As shown in Fig. 1, MOSFETs typically have channel, LDD, and halo doping, making it difficult to obtain an analytical expression for V_t. Many parameters such as well_const, ch_position_a, ch_depth_a, and ch_const influence the doping distribution below the dielectric. Physics-based analyses can preliminarily determine whether these parameters are related to V_t, but it is difficult to determine which parameter is more important or has greater influence.

For transconductance, the RF regression suggested that well_const was the most important parameter. A possible reason for this is that the transconductance is strongly influenced by the electron mobility [40], and the electron mobility is influenced by the doping concentration [40, 41].

For the subthreshold slope, the RF regression suggested that LDD doping was the most sensitive part. The subthreshold slope is associated with the ability of the gate voltage to control the surface potential [42]. LDD doping has a significant influence on the electric field near the drain [43–45]. Ldd_factor and ldd_depth are the most important parameters, which is consistent with semiconductor theory. In addition, some experimental results have confirmed the significant influence of LDD doping on the subthreshold slope [46].

The importance of the parameters obtained by the machine learning method is consistent with theoretical analyses of semiconductors. For complex devices in which it is not easy to obtain an analytical expression, machine learning methods are helpful in determining the key parameters. These results can be used as a reference for further physical research.

3 Machine learning-based calibration framework

We built a fast 16-dimension NMOS calibration framework using a machine learning-based surrogate model. The surrogate model was several orders of magnitude faster than the original TCAD simulation, and the desired calibration parameters were obtained within several seconds.

3.1 Surrogate model

As shown in Fig. 10, the proposed surrogate model relates the 16-dimension NMOS parameters with the metrics of the $I_d - V_g$ curves: V_t, G_m, and S^{-1}. Considering that the TCAD model may not function with certain parameters, a classifier was utilized before the typical regressor to judge whether the input parameters were valid. If the input parameters were valid, the related metrics were calculated using the regressor. Therefore, for any set of input parameters, the surrogate model could predict the related metrics or its failure in function.

We built a new 16-dimensional dataset to train the surrogate model. Classification techniques were employed again to improve the efficiency of sample generation. This procedure is similar to that described in Sect. 2.3.1. First, we random generated 1398 samples and found that 685 samples were valid for creating the dataset, which accounted for 49.0% of the total calculations. Second, an MLP classifier with a hidden_layer_size of 100 and batch_size of 16 was trained using the 1398-sample dataset. A precision of 83.3% was achieved. Third, we randomly generated 16,200 samples, and the MLP classifiers predicted 7710 of them to be valid. We calculated these samples using TCAD and found that 6473 samples were valid. These accounted for 84.0% of the total calculations. The results show again that the classifier successfully improved the efficiency of generating valid samples and saved time.

The 6473 valid samples, together with the previous 685 valid samples, were used to train the regressor of the surrogate model. The training was performed using a Python library called Keras [47], which was developed for deep learning. We built a three-layer artificial neural network (ANN). This type of machine learning model has been widely adopted in many scientific studies [48–51]. The inputs were the 16-dimensional parameters. Considering the differences in magnitude between the dimensions, we normalized the inputs and outputs. The inputs were normalized using the following two steps: First, the doping parameters were processed using a logarithm to limit the

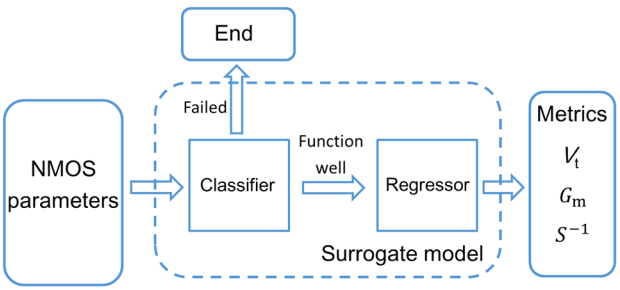


Fig. 10 Schematic of surrogate model

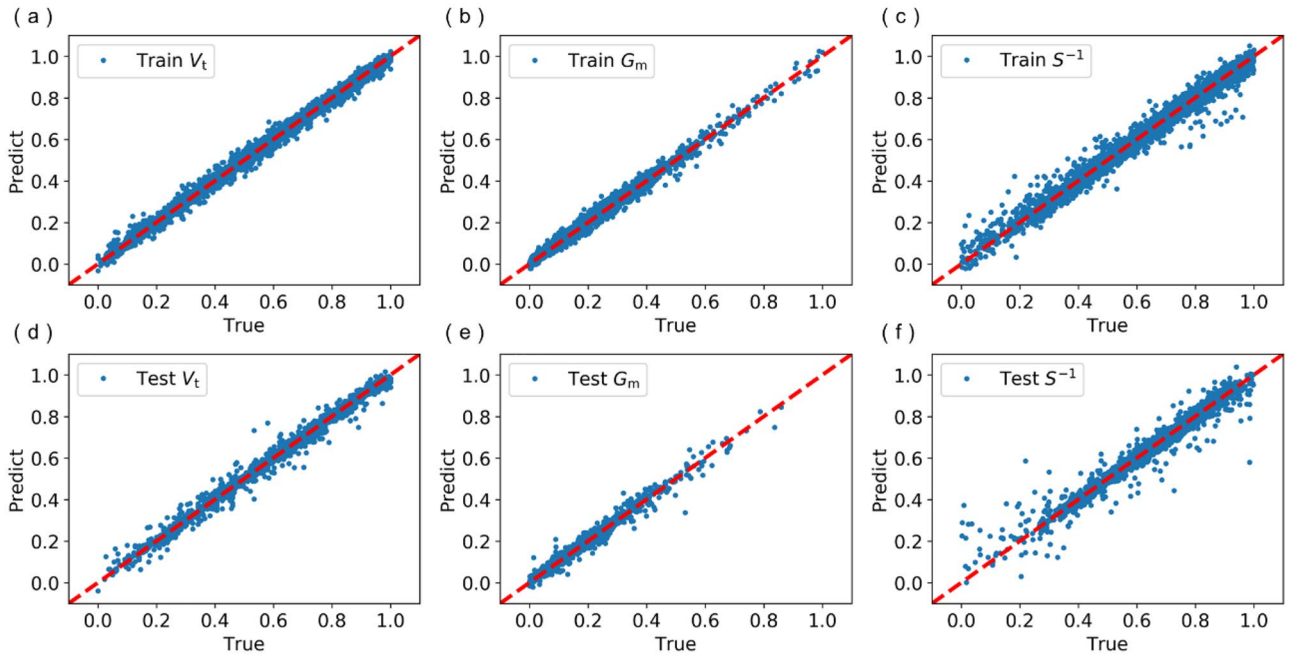


Fig. 11 (Color online) Performance of the ANN on the training and test sets. The abscissa is the true value of TCAD simulations while the ordinate is the predicted value

changing range. Second, all parameters were normalized to a mean of 0 and standard deviation of 1 over the training set. The outputs were three neurons, corresponding to V_t, G_m, and S^{-1}. Each regression target was normalized between 0 and 1 using a linear transformation. The activation functions were softmax [52], rectified linear unit (ReLU) [53], and linear functions for the first to the last layers, respectively. 128 neurons were used in the first and second layers. The loss function was the mean square error. Adam [54] with a default learning rate was adopted as the optimizer for training the ANN. A batch size of 32 was used for stable training [55]. 80 training epochs were used in this study. The dataset was divided into portions of 80% and 20%; 80% of the dataset was used to train the ANN and the remaining part was used for test. The performance on the training and test sets is shown in Fig. 11. The results indicate that the ANN could predict the simulation results for the test set. The mean absolute errors were 0.016, 0.011, and 0.023 for V_t, G_m, and S^{-1}, respectively, in the test set.

The first 1398 calculated samples, together with the 7710 calculated samples, were used to train the classifier of the surrogate model. The dataset was randomly split into training and test sets in proportions of 80% and 20%, respectively. The training procedure was similar to the previous procedures. The ROC curves shown in Fig. 12 suggest that the MLP classifier with a hidden_layer_size of 100 and batch_size of 16 performed the best. The confusion matrix for the test set is presented in Table 5. A precision of 91.0% and recall of 93.8% were achieved for the surrogate model.

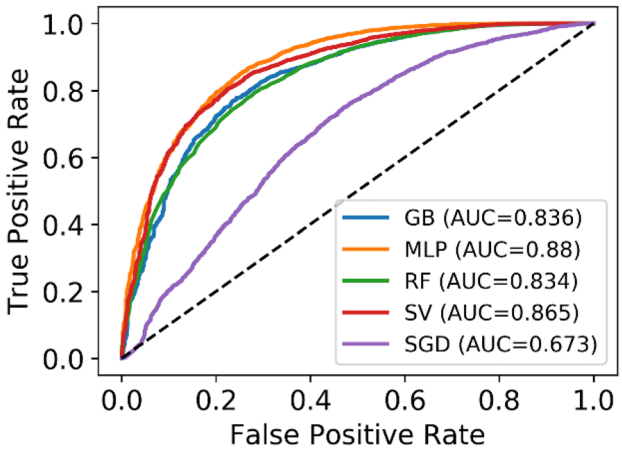


Fig. 12 (Color online) ROC curves for the surrogate model

Table 5 Confusion matrix of MLP classifier on the test set. Precision: 91.0%. Recall: 93.8%

Predicted class	True class		Total
	Positive	Negative	
Positive	1350	134	1484
Negative	90	248	338
Total	1440	382	1822

3.2 Calibration procedure

The surrogate model was utilized for NMOS calibration. The calibration goal is described by six metrics: V_t, G_m, and S^{-1} at a low voltage ($V_d = 0.1$ V) and at the working voltage ($V_d = V_{dd}$). For a perfect set of calibrated NMOS parameters, all six metrics should be identical to their target values. The dimensions and doping of the NMOS are described by 15 parameters, with the exception of V_d. The parameter V_d is used to describe the drain voltage, which is flexible for a given NMOS. Different V_d values are related to different metrics of $I_d - V_g$ curves for the NMOS. The calibration procedure is illustrated in Fig. 13. To check the given NMOS structure described by the first 15 parameters, V_d was set to low and working voltages respectively. The corresponding metrics were predicted and their differences from the goals were calculated. If the difference was sufficiently small, this set of parameters was selected as one calibration result.

When calibrating a PDK, six parameters are specified by the PDK, and the other ten parameters are searched to obtain $I_d - V_g$ curve metrics similar to the goals. The PDK specifies the values of gateLen, gateWidth, tox, sd_peak, sd_depth, and V_{dd}. For the other 10 parameters, we randomly generated values to search the best parameter values.

A Python script was written to implement the calibration procedure. First, the script generates a large number of random parameters. Second, the script utilizes the surrogate models to identify valid sets of parameters and calculate their related metrics at low voltage and working voltage. Third, the script compares the metrics with the goals and computes the differences between them. Considering that the surrogate model contains errors in predicting the metrics, the script outputs the five most consistent parameters. Finally, the output parameters are checked using TCAD calculations. The most consistent parameter set was selected as the calibration result.

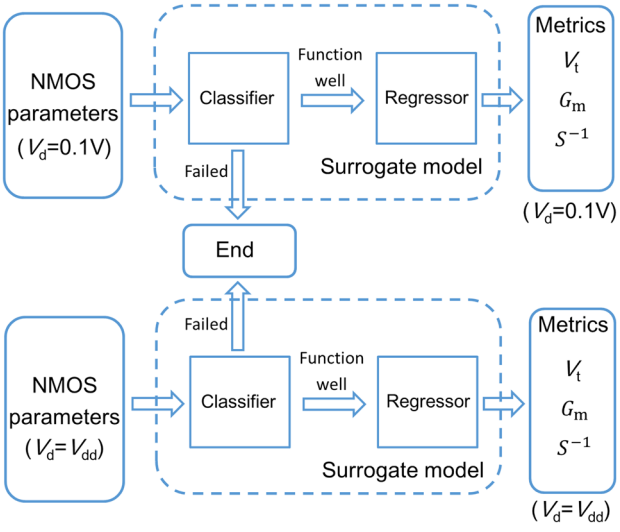


Fig. 13 Proposed calibration procedure

3.3 Performance

We tested the performance of the calibration method with 3 PDKs: 28 nm, 40 nm and 65 nm. Different PDKs have distinct values of tox, sd_peak, sd_depth, and V_{dd}. In addition, each PDK can specify different gateLen and gateWidth values within its allowed range. For each PDK, we selected one gate length value and one gate width value to generate the calibration goal. The PDK information and selected gate dimensions are listed in Table 6. As shown in Fig. 14, the calibration goals are quite different for the different PDKs.

For each of the calibration goals, the Python script required approximately 8 s to find the five most consistent parameter sets in 500,000 random inputs. The output parameter sets were then sent for TCAD calculations to determine the most consistent set. The calibration results are shown in Figs. 15, 16. The calibrated parameters are listed in Table 7. The calibration results matched the goals, and the parameters required by the PDKs were satisfied.

We introduced the root-mean-square error (RMSE) to measure the differences between the calibration results and goals. The RMSE is computed as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}, \tag{2}$$

where y and $\hat{y}$ represent the goal and the calibration results, respectively. The ratios of the RMSE relative to the average value of y are shown in Fig. 17. The relative RMSE of S^{-1} was also computed to evaluate its performance in the subthreshold region. Generally, an RMSE of approximately 10% is sufficient for TCAD model calibration for radiation effects. Our results show the feasibility of fast NMOS model calibration with the help of machine learning. The TCAD simulation required approximately 600 s to calculate one NMOS case on a personal computer with an Intel i7-12700 CPU. Generally, only one case can be calculated

Table 6 Information about PDKs to be calibrated and selected gate dimensions

	28 nm PDK	40 nm PDK	65 nm PDK
tox (nm)	3	2.42	2.35
sd_peak (cm^{-3})	1×10^{20}	1×10^{20}	1×10^{20}
sd_depth (nm)	64	70	115
V_{dd} (V)	0.9	1.1	1.2
gateLen (nm)	35	40	65
gateWidth (nm)	200	120	200

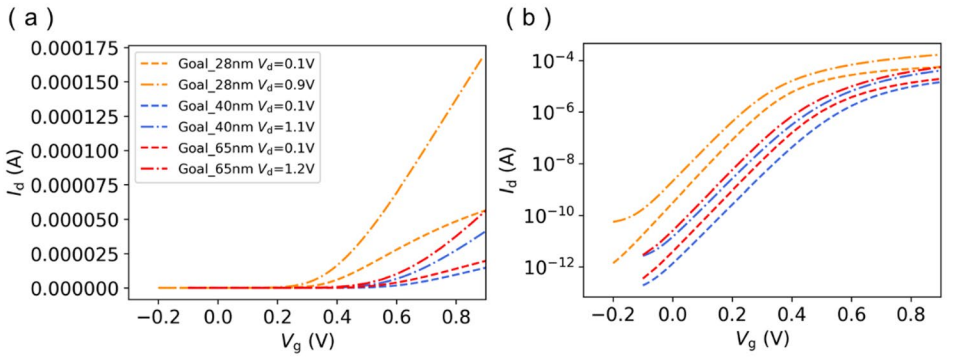


Fig. 14 (Color online) Calibration goals for different PDKs

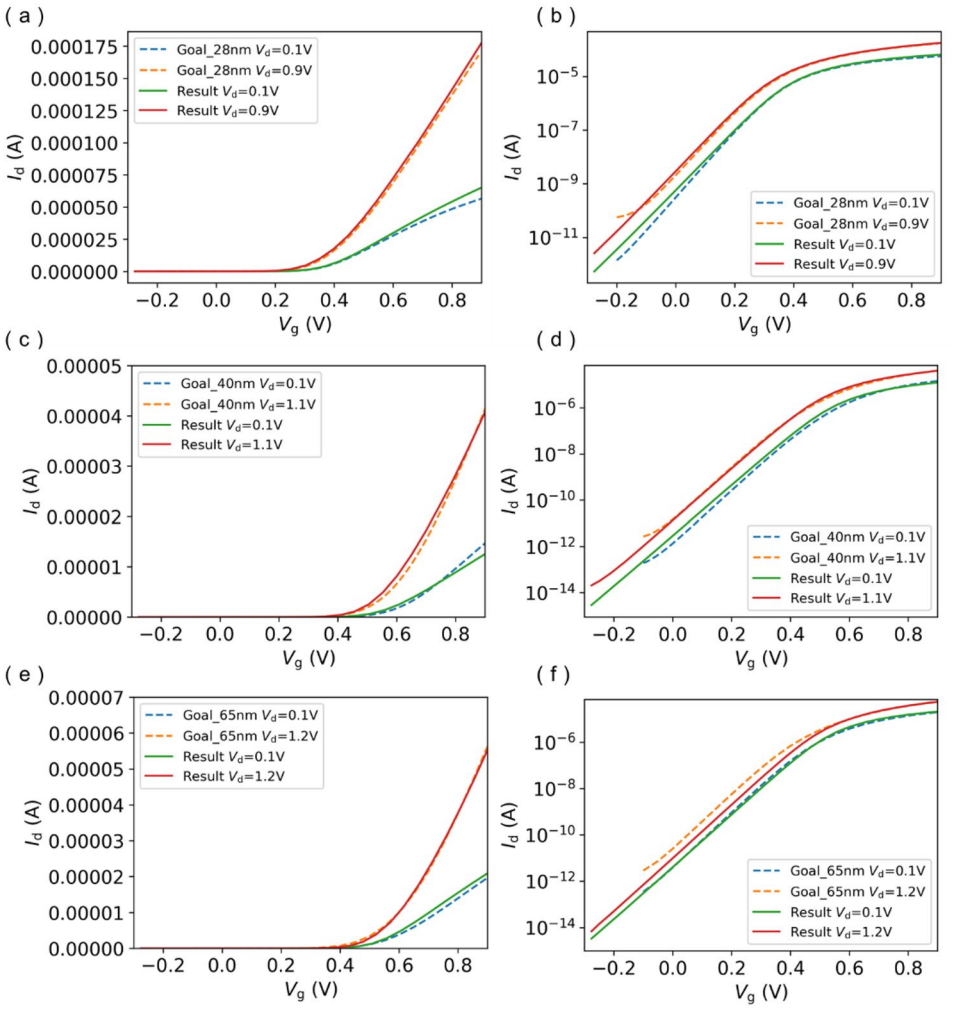


Fig. 15 (Color online) Calibration performance for 3 different PDKs

at a time. For comparison, 500,000 cases could be evaluated in approximately 8 s using the surrogate model on the same computer. The proposed method is approximately 10^7 times faster than the original TCAD simulation.

For the 40 nm and 65 nm PDKs, the gate material is typically polysilicon. In these cases, the workF of the gate material is set to 4.09 eV as an approximation of polysilicon with N+ doping concentration of 2×10^{20} cm^{-3}. The other nine parameters are searched during calibration. As shown in Fig. 18, the $I_d - V_g$ curves remained almost the same when the gate material changed to the TCAD built in polysilicon with arsenic doping of 2×10^{20} cm^{-3}.

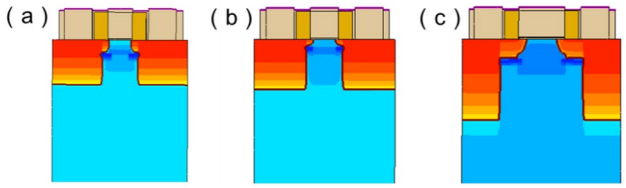


Fig. 16 (Color online) Schematic of calibrated results. **a** 28 nm PDK. **b** 40 nm PDK. **c** 65 nm PDK

Table 7 Calibrated parameters for three different PDKs

	28 nm PDK	40 nm PDK	65 nm PDK
workF (eV)	4.20	4.09	4.09
well_const (cm^{-3})	4.76×10^{17}	6.89×10^{17}	8.00×10^{17}
ch_const (cm^{-3})	2.03×10^{18}	1.88×10^{18}	5.80×10^{18}
ch_depth_a (nm)	3.63	10.67	26.05
ch_position_a (nm)	9.01	0.65	0.93
ldd_peak (cm^{-3})	4.82×10^{19}	2.18×10^{19}	4.53×10^{19}
ldd_depth (nm)	17.14	9.18	30.08
ldd_factor	0.05	0.48	0.41
halo_position_z (nm)	19.88	10.85	28.21
sd_position (nm)	9.76	7.36	27.48

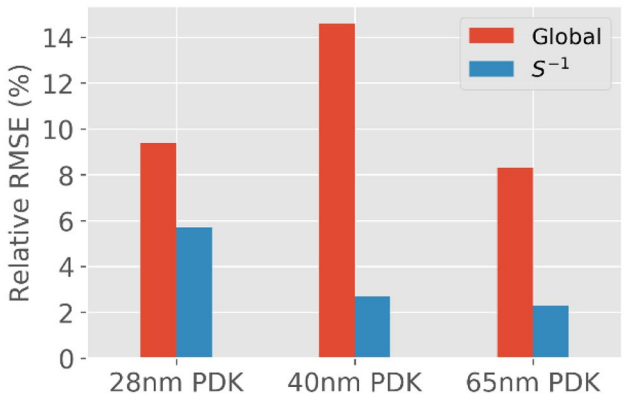


Fig. 17 (Color online) Relative RMSEs for three different PDKs

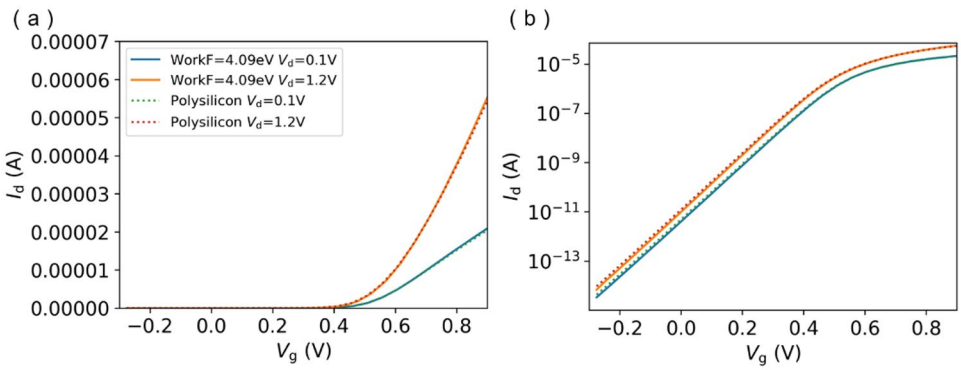
4 Discussion

The calibration method and relevant Python script proposed here can serve as a fast preliminary calibration tool that automatically provides possible calibration results in several seconds. The model contains basic types of doping, and more detailed calibrations can be performed on this basis. The important parameters that govern each metric of the $I_d - V_g$ curves were identified using the machine learning methods in Sect. 2.3.2. Thus, small adjustments are easy to perform. For the 28 nm PDK, the gate dielectric is commonly multilayered. However, the PDK used in this study only provides a tox value for reference. If detailed structural information is available, the calibration can be further improved based on the script outputs.

The key advantage of the proposed approach is that once the model is trained, it can continuously provide fast calibrations for different goals within its scope. The generation of training samples is time consuming. However, this is a one-time process.

The proposed machine learning methods are data-driven, learning patterns and correlations from data. In this study, the classification of valid parameter sets, importance of parameters, and correlations between the parameters and related metrics were obtained using machine learning methods without the need for semiconductor expertise. This data-driven approach complements physics-based research. Classical semiconductor theory is suitable for describing relatively simple structures. However, it is difficult to obtain analytical expressions for complex structures. Machine learning is a data-driven approach. It learns the correlations and importance of parameters from the data but does not fully understand the physical principles behind the results. Its results can serve as a reference for further physical research.

Fig. 18 (Color online) Comparison between polysilicon and material with workF 4.09 eV for the 65 nm MOSFET



5 Conclusion

We presented a machine learning approach for MOSFET model calibration and built a Python script utilizing a machine learning-based surrogate model. The surrogate model was several orders of magnitude faster than the original TCAD simulation, and the desired calibration parameters for the NMOS could be obtained in several seconds. In this study, a fundamental model containing 26 parameters was introduced to represent the typical structure of a MOSFET. Classifications were developed to improve the efficiency of generating training samples by predicting the validity of parameter combinations before the TCAD calculation. Feature selection techniques were used to identify the important parameters and decrease the dimensions of the NMOS model. A 16-dimension surrogate model comprising a classifier and regressor was built. The surrogate model determines the validity of the input parameters and predicts the corresponding threshold voltage, transconductance, and subthreshold slope of $I_d - V_g$ curve. A calibration procedure was proposed and implemented using a Python script. The calibration script was tested using three NMOS calibration goals generated by different PDKs. The results indicated that the calibrated parameter values could be achieved within approximately 8 s. Our work demonstrates the feasibility of machine learning-based fast model calibration. A similar approach could be adopted to develop fast calibration tools for other devices. In addition, this study shows that these machine learning methods learn patterns and correlations from data instead of employing domain expertise. This indicates that machine learning could be an alternative research approach to complement classical physics-based research.

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Bai-Chuan Wang, Tan Wang and Jing-Yan Xu. The first draft of the manuscript was written by Bai-Chuan Wang and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Data availability The data that support the findings of this study are openly available in Science Data Bank at https://doi.org/10.57760/sciencedb.j00186.00311 and https://cstr.cn/31253.11.sciencedb.j00186.00311

Declarations

Conflict of interest The authors declare that they have no competing interests.

References

1. L. Ding, W. Chen, T. Wang et al., Adverse effect of inappropriately implementing source-isolation mitigation technique. At. Energy Sci. Technol. **55**, 2260–2266 (2021).
2. L. Ding, W. Chen, H. Guo et al., Scaling effects of single-event gate rupture in thin oxides. Chin. Phys. B **22**, 640–644 (2013). https://doi.org/10.1088/1674-1056/22/11/118501
3. T. Wang, L. Ding, Y. Luo et al., Physics-based circuit-level analysis of MCU characteristics in bulk CMOS SRAM. At. Energy Sci. Technol. **55**, 2121–2127 (2021).
4. L. Cai, G. Guo, J. Liu et al., Experimental study of temperature dependence of single-event upset in SRAMs. Nucl. Sci. Tech. **27**, 16 (2016). https://doi.org/10.1007/s41365-016-0014-9
5. B. He, L. Ding, Z. Yao et al., Three-dimensional simulation of total dose effects on ultra-deep submicron devices. Acta Phys. Sin. **60**, 544–550 (2011). (**in Chinese**)
6. L. Ding, W. Chen, H. Guo et al., Modeling the impact of well contacts on SEE response with bias-dependent single-event compact model. Microelectron. Reliab. **81**, 337–341 (2018). https://doi.org/10.1016/j.microrel.2017.11.001
7. L. Ding, T. Wang, F. Zhang et al., An analytical model to evaluate well potential modulation and bipolar amplification effects. IEEE T. Nucl. Sci. **70**, 1724–1731 (2023). https://doi.org/10.1109/TNS.2023.3266005
8. J. Xu, S. Chen, R. Song et al., Analysis of single-event transient sensitivity in fully depleted silicon-on-insulator MOSFETs. Nucl. Sci. Tech. **29**, 49 (2018). https://doi.org/10.1007/s41365-018-0391-3
9. J. Li, R. Li, L. Ding et al., TCAD simulation analysis of vertical parasitic effect induced by pulsed γ- ray in NMOS from 180 nm to 40 nm technology nodes. Acta Phys. Sin. **71**, 201–208 (2022). (**in Chinese**)
10. L. Ding, H. Guo, W. Chen et al., Simulation study of the influence of ionizing irradiation on the single event upset vulnerability of static random access memory. Acta Phys. Sin. **62**, 486–493 (2013). (**in Chinese**)
11. X. Cao, L. Xiao, M. Huo et al., Heavy ion-induced single event upset sensitivity evaluation of 3D integrated static random access memory. Nucl. Sci. Tech. **29**, 31 (2018). https://doi.org/10.1007/s41365-018-0377-1
12. L. Ding, W. Chen, T. Wang et al., Modeling the dependence of single-event transients on strike location for circuit-level simulation. IEEE T. Nucl. Sci. **66**, 866–874 (2019). https://doi.org/10.1109/TNS.2019.2904716
13. O.A. Amusan, *Analysis of Single Event Vulnerabilities in a 130 nm CMOS Technology* (Vanderbilt University, Nashville, 2006).
14. C. Xu, Y. Liu, X. Liao et al., Machine learning regression-based single-event transient modeling method for circuit-level simulation. IEEE T. Electron Dev. **68**, 5758–5764 (2021). https://doi.org/10.1109/TED.2021.3113884
15. S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future. Multimed. Tools Appl. **80**, 8091–8126 (2021). https://doi.org/10.1007/s11042-020-10139-6
16. T. Binder, C. Heitzinger, S. Selberherr, A study on global and local optimization techniques for TCAD analysis tasks. IEEE Trans. Comput. Aided Des. **23**, 814–822 (2004). https://doi.org/10.1109/TCAD.2004.828130
17. Z. Dai, Y. Nie, Z. Hui et al., Design of S-band photoinjector with high bunch charge and low emittance based on multi-objective genetic algorithm. Nucl. Sci. Tech. **34**, 41 (2023). https://doi.org/10.1007/s41365-023-01183-6
18. H. Chen, L. Zheng, B. Gao et al., Beam dynamics optimization of very-high-frequency gun photoinjector. Nucl. Sci. Tech. **33**, 116 (2022). https://doi.org/10.1007/s41365-022-01105-y

19. S. Nikolopoulos, I. Kalogeris, V. Papadopoulos, Non-intrusive surrogate modeling for parametrized time-dependent partial differential equations using convolutional autoencoders. Eng. Appl. Artif. Intel. **109**, 104652 (2022). https://doi.org/10.1016/j.engappai.2021.104652
20. Y. Kiarashinejad, S. Abdollahramezani, A. Adibi, Deep learning approach based on dimensionality reduction for designing electromagnetic nanostructures. NPJ Comput. Mater. **6**, 12 (2020). https://doi.org/10.1038/s41524-020-0276-y
21. B. Liu, L. Xu, J. Huang, Thermal transparency with periodic particle distribution: a machine learning approach. J. Appl. Phys. **129**, 65101 (2021). https://doi.org/10.1063/5.0039002
22. Y.S. Bankapalli, H.Y. Wong, TCAD augmented machine learning for semiconductor device failure troubleshooting and reverse engineering, in *2019 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)* (2019).
23. K. Mehta, S.S. Raju, M. Xiao et al., Improvement of TCAD augmented machine learning using autoencoder for semiconductor variation identification and inverse design. IEEE Access **8**, 143519–143529 (2020). https://doi.org/10.1109/ACCESS.2020.3014470
24. H. Dhillon, K. Mehta, M. Xiao et al., TCAD-augmented machine learning with and without domain expertise. IEEE T. Electron Dev. **68**, 5498–5503 (2021). https://doi.org/10.1109/TED.2021.3073378
25. A. Ortiz-Conde, F.J.G. Sánchez, J.J. Liou et al., A review of recent MOSFET threshold voltage extraction methods. Microelectron. Reliab. **42**, 583–596 (2002). https://doi.org/10.1016/S0026-2714(02)00027-6
26. A. Natekin, A. Knoll, Gradient boosting machines, a tutorial. Front. Neurorobot. **7**, 21 (2013). https://doi.org/10.3389/fnbot.2013.00021
27. D.W. Ruck, S.K. Rogers, M. Kabrisky, Feature selection using a multilayer perceptron. J. Neural Netw. Comput. **2**, 40–48 (1990).
28. K. Fawagreh, M.M. Gaber, E. Elyan, Random forests: from early developments to recent advancements. Syst. Sci. Control Eng. **2**, 602–609 (2014). https://doi.org/10.1080/21642583.2014.956265
29. R.G. Brereton, G.R. Lloyd, Support vector machines for classification and regression. Analyst **135**, 230–267 (2010). https://doi.org/10.1039/B918972F
30. F. Kabir, S. Siddique, M. Kotwal, Bangla text document categorization using stochastic gradient descent (SGD) classifier, in *2015 International Conference on Cognitive Computing and Information Processing(CCIP)*, 0003-04-20, pp. 1–4.
31. F. Pedregosa, G. Varoquaux, A. Gramfort et al., Scikit-learn: Machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011).
32. R.C. Prati, G. Batista, D.F. Silva, Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. Knowl. Inf. Syst. **45**, 247–270 (2015). https://doi.org/10.1007/s10115-014-0794-3
33. A. Costine, P. Delsa, T. Li et al., Data-driven assessment of chemical vapor deposition grown MoS2 monolayer thin films. J. Appl. Phys. **128**, 235303 (2020). https://doi.org/10.1063/5.0017507
34. T. Fawcett, An introduction to ROC analysis. Pattern Recogn. Lett. **27**, 861–874 (2006). https://doi.org/10.1016/j.patrec.2005.10.010
35. D.J. Hand, R.J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems. Mach. Learn. **45**, 171–186 (2001). https://doi.org/10.1023/A:1010920819831
36. T.K. Ho, Random decision forests, in *Proceedings of 3rd International Conference on Document Analysis and Recognition, 1995-01-01*, pp. 278–282.
37. L. Breiman, Random forests. Mach. Learn. **45**, 5–32 (2001). https://doi.org/10.1023/A:1010933404324
38. X. Chen, M. Liu, Prediction of protein–protein interactions using random decision forest framework. Bioinformatics **21**, 4394–4400 (2005). https://doi.org/10.1093/bioinformatics/bti721
39. B.H. Menze, B.M. Kelm, R. Masuch et al., A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC Bioinform. **10**, 213 (2009). https://doi.org/10.1186/1471-2105-10-213
40. Y. Taur, T.H. Ning, *Fundamentals of Modern VLSI Devices*, 2nd edn. (Cambridge University Press, Cambridge, 2009).
41. O.D. Restrepo, K. Varga, S.T. Pantelides, First-principles calculations of electron mobilities in silicon: Phonon and Coulomb scattering. Appl. Phys. Lett. **94**, 212103 (2009). https://doi.org/10.1063/1.3147189
42. A. Godoy, J.A. López-Villanueva, J.A. Jiménez-Tejada et al., A simple subthreshold swing model for short channel MOSFETs. Solid State Electron. **45**, 391–397 (2001). https://doi.org/10.1016/S0038-1101(01)00060-0
43. S. Ogura, P. Tsang, W. Walker et al., Design and characteristics of the lightly doped drain-source (LDD) insulated gate field-effect transistor. IEEE J. Solid-St. Circ. **15**, 424–432 (1980). https://doi.org/10.1109/JSSC.1980.1051416
44. A. Klös, A. Kostka, A new analytical method of solving 2D Poisson's equation in MOS devices applied to threshold voltage and subthreshold modeling. Solid State Electron. **39**, 1761–1775 (1996). https://doi.org/10.1016/S0038-1101(96)00122-0
45. L. Chua, P. Liu, Subthreshold current for submicron LDD MOS transistor, in *Proceedings of 36th Midwest Symposium on Circuits and Systems* (1993).
46. D. Zhang, S. Yu, C. Huang, Light-doped drain technology for submicron CMOS. Microelectron. Comput. (1994). https://doi.org/10.19304/j.cnki.issn1000-7180.1994.01.013 (**in Chinese**)
47. Keras Documentation. https://keras.io. Accessed 26 Sept 2023.
48. B. Wang, M. Qiu, W. Chen et al., Machine learning-based analyses for total ionizing dose effects in bipolar junction transistors. Nucl. Sci. Tech. **33**, 131 (2022). https://doi.org/10.1007/s41365-022-01107-w
49. Y. Pan, X. Nie, Z. Li et al., Data-driven vehicle modeling of longitudinal dynamics based on a multibody model and deep neural networks. Measurement **180**, 109541 (2021). https://doi.org/10.1016/j.measurement.2021.109541
50. J. Ma, S. Dong, G. Chen et al., A data-driven normal contact force model based on artificial neural network for complex contacting surfaces. Mech. Syst. Signal Process. **156**, 107612 (2021). https://doi.org/10.1016/j.ymssp.2021.107612
51. Y. Liu, J. Zhu, N. Roberts et al., Recovery of saturated signal waveform acquired from high-energy particles with artificial neural networks. Nucl. Sci. Tech. **30**, 148 (2019). https://doi.org/10.1007/s41365-019-0677-0
52. A. Laha, S.A. Chemmengath, P. Agrawal et al., On controllable sparse alternatives to softmax, in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)* (2018).
53. V. Nair, G. Hinton, Rectified linear units improve restricted Boltzmann machines, in *27th International Conference on Machine Learning (ICML-10)* (2010).
54. D. Kingma, J. Ba, Adam: a method for stochastic optimization, in *3rd International Conference on Learning Representations (ICLR 2015)* (2015).
55. D. Masters, C. Luschi, Revisiting small batch training for deep neural networks. arXiv:1804.07612 (2018).

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.