

## Article

# Compact Modeling of Advanced Gate-All-Around Nanosheet FETs Using Artificial Neural Network

Yage Zhao <sup>1</sup>, Zhongshan Xu <sup>1</sup>, Huawei Tang <sup>1</sup>, Yusi Zhao <sup>1</sup>, Peishun Tang <sup>1</sup>, Rongzheng Ding <sup>1</sup>, Xiaona Zhu <sup>1</sup>, David Wei Zhang <sup>1,2</sup> and Shaofeng Yu <sup>1,2,\*</sup>

<sup>1</sup> School of Microelectronics, Fudan University, Shanghai 200433, China; 20112020044@fudan.edu.cn (Y.Z.); 21112020147@m.fudan.edu.cn (Z.X.); 23112020147@m.fudan.edu.cn (H.T.); 22212020147@m.fudan.edu.cn (P.T.); xnzhu@fudan.edu.cn (X.Z.); dwzhang@fudan.edu.cn (D.W.Z.)

<sup>2</sup> National Integrated Circuit Innovation Center, Shanghai 201203, China

\* Correspondence: shaofeng\_yu@fudan.edu.cn

**Abstract:** As the architecture of logic devices is evolving towards gate-all-around (GAA) structure, research efforts on advanced transistors are increasingly desired. In order to rapidly perform accurate compact modeling for these ultra-scaled transistors with the capability to cover dimensional variations, neural networks are considered. In this paper, a compact model generation methodology based on artificial neural network (ANN) is developed for GAA nanosheet FETs (NSFETs) at advanced technology nodes. The DC and AC characteristics of GAA NSFETs with various physical gate lengths ( $L_g$ ), nanosheet widths ( $W_{sh}$ ) and thicknesses ( $T_{sh}$ ), as well as different gate voltages ( $V_{gs}$ ) and drain voltages ( $V_{ds}$ ) are obtained through TCAD simulations. Subsequently, a high-precision ANN model architecture is evaluated. A systematical study on the impacts of ANN size, activation function, learning rate, and epoch (the times of complete pass through the entire training dataset) on the accuracy of ANN models is conducted, and a shallow neural network configuration for generating optimal ANN models is proposed. The results clearly show that the optimized ANN model can reproduce the DC and AC characteristics of NSFETs very accurately with a fitting error (MSE) of 0.01.

**Keywords:** gate-all-around (GAA) Nanosheet FETs (NSFETs); compact model; artificial neural network (ANN); TCAD simulation



**Citation:** Zhao, Y.; Xu, Z.; Tang, H.; Zhao, Y.; Tang, P.; Ding, R.; Zhu, X.; Zhang, D.W.; Yu, S. Compact

Modeling of Advanced

Gate-All-Around Nanosheet FETs

Using Artificial Neural Network.

*Micromachines* **2024**, *15*, 218.

<https://doi.org/10.3390/mi15020218>

Academic Editors: Zeheng Wang and

Jing-Kai Huang

Received: 1 January 2024

Revised: 28 January 2024

Accepted: 29 January 2024

Published: 31 January 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In response to market demands, the transistor dimensions have been scaled down proportionally according to Moore's Law. As an alternative to planar metal-oxide-semiconductor field-effect transistors (MOSFETs), fin field-effect transistors (FinFETs), which utilize a three-dimensional architecture with the gate wrapping around vertical fins on top and sides, have been developed and commercialized in 22 nm CMOS technology [1–3]. During the past decades, FinFET technology has been successfully applied to 5 nm and even 3 nm technology nodes through higher aspect ratio and layout optimization [4–8]. However, the scaling of FinFETs has also encountered fabrication- and performance-related obstacles due to fundamental physical limitations and difficulties in developing the required process. The performance improvement is constrained by the severe short channel effects (SCEs), while it is difficult to populate multiple fins in a limited space as CGP is further reduced. As the most feasible solution to extend Moore's Law and Dennard's Law, gate-all-around (GAA) nanosheet FETs (NSFETs) are poised to become a mainstream device architecture for 2 nm node and beyond [9–12]. Compared to traditional FinFETs or planar MOSFETs, GAA NSFETs offer superior electrostatic control, higher driving capability, lower leakage current, and more effective footprint [10]. This is because they not only have the gates surrounding the channel, but also have wider effective widths in the same footprint. Nevertheless, this advancement imposes a challenge to semiconductor device models.

Semiconductor device models are regarded as a bridge between foundry, EDA vendor, and design house, as well as a key-enabler for accurate integrated circuit (IC) simulations. The conventional semiconductor device models include macro models [13,14], compact models [15–18], and look-up table (LUT) models [19,20]. In particular, compact models are the mainstream ones and are composed of physics-based equations, which have been developed for decades. The first industry standard compact model is BSIM (Berkeley short-channel insulated-gate field-effect transistor model), whose genesis can be traced to the 1980s [21], and several versions have been developed and remain in use today [16,22,23]. Generally, analytical equations are used to describe device  $I$ - $V$  and  $C$ - $V$  characteristics in the subthreshold, linear, and saturation regions in a unified way. The accuracy of the compact models is crucial for efficient analysis and design of ICs. However, for advanced transistors, the underlying physics becomes much more complicated, making the models more difficult to fit. In addition, the actual electrical properties of miniaturized transistors are case sensitive due to dimension variations. Since developing suitable analytical compact models is complex and often takes several years, it requires novel modeling methodology to circumvent the high costs of time and labor.

The need for a new technique brings the artificial neural network (ANN) method to the attention of researchers, which has been attempted for planar MOSFETs modeling since the early 1990s and showed good precision [24]. ANNs represent a class of machine learning models inspired by the neuromorphic architecture, and use a set of multilayered perceptrons/neurons, also known as feed-forward neural networks, consisting of an input layer, multiple hidden layers, and an output layer [25,26]. Because of the robust learning capability, they have once been a powerful tool used in the computer science to deal with machine learning issues. The primary objective of ANNs is to learn complex mappings between inputs and outputs by adjusting the weights and biases of interconnected neurons, in a nutshell, is to achieve a good means of solving data fitting problems. This learning process involves the application of mathematical principles, particularly the chain rule in calculus, to update the network parameters and minimize the error between predicted and actual outcomes. In other words, with a reasonable network configuration, ANNs can fit arbitrary nonlinear functions and hence can also be developed as black-box models to address nonlinear systems or more sophisticated internal expressions, such as the compact modeling of semiconductor devices in advanced nodes mentioned earlier. Although the ANN models seemed to be a simple black box, there are many parameters within the neural network that have an impact on the accuracy of models, which will further affect the subsequent circuit simulations. Thereby, an in-depth study of ANN-based compact modeling methodology is necessary for the development and application of GAA devices and even complementary FET (CFET) devices, which are more sophisticated architectures with n-FET folded onto p-FET, in advanced technologies [27,28]. Actually, there are some interesting and meaningful studies on ANN-based device modeling that have been published in recent years [29–32]. However, most of the literature in this field have only superficially studied ANN modeling, focusing instead on its implementation in subsequent circuits or on the unique electrical properties under investigation, and lacking an in-depth understanding and full exploration of the ANNs used for modeling.

In this work, we conduct a comprehensive evaluation of the compact modeling of advanced GAA NSFETs based on ANN, with the datasets from finely calibrated TCAD simulations. Referring to [10] and IRDS 2022 [33], an N-channel GAA NSFET was built as the nominal transistor for the modeling study. The applied voltages on terminals and 3-D nanosheet dimensions were set as input parameters and varied to obtain datasets, some of which were used for training data feeding into the ANN and the others were used for testing data for the final test. Appropriate data preprocessing and neural network configurations, as well as  $L2$  regularization were adopted to improve model accuracy. Without considering the physical characteristics of real transistors, high fitting accuracy can be achieved by using transistor data for model training. The DC and AC characteristics are well mapped with the five input variants, including applied voltages and geometrical dimensions.

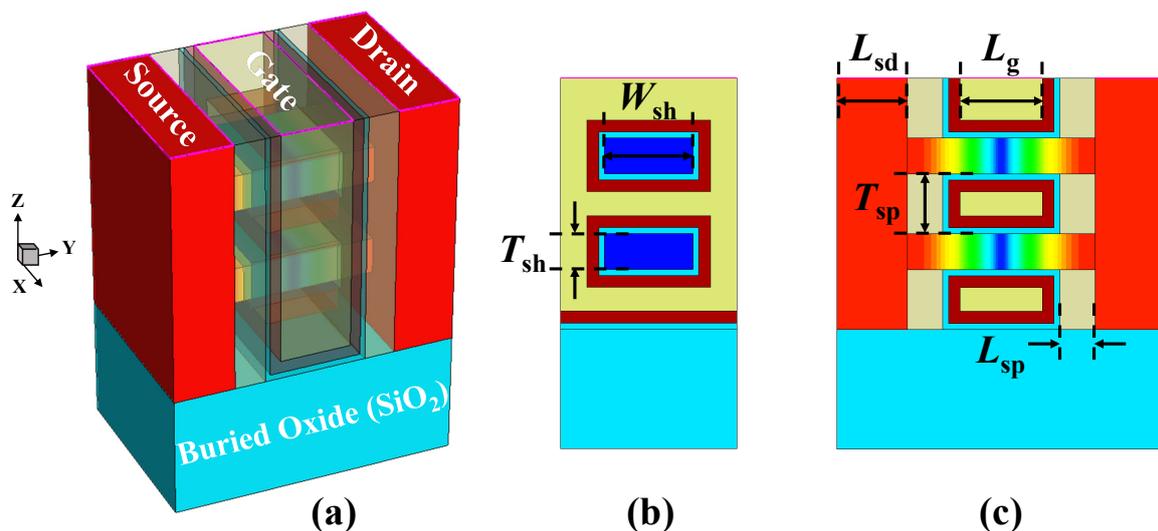
## 2. Device Structure, TCAD Simulation Calibration, and Dataset Generation

### 2.1. Device Structure

The Sentaurus Technology Computer-Aided Design (TCAD) [34] tool is exploited to construct the GAA NSFET devices and generate physical electric characteristic data for subsequent studies. Figure 1a–c shows the 3-D schematic of nominal GAA NSFET structure and 2-D cross-sectional along and across the channel views, respectively. Detailed parameters of a nominal highly scaled device at 2 nm technology node are specifically listed in Table 1 following IRDS 2022 [33], where the physical gate length ( $L_g$ ) of 14 nm, nanosheet width ( $W_{sh}$ ) of 15 nm, nanosheet thickness ( $T_{sh}$ ) of 6 nm, the spacer length ( $L_{sp}$ ) of 6 nm, and the sheet-to-sheet spacing ( $T_{sp}$ ) of 10 nm are adopted. For n-type MOS, the in-situ uniform doping profiles for channels and source/drain regions were performed with  $1 \times 10^{10} \text{ cm}^{-3}$  of boron doping concentration and  $5 \times 10^{20} \text{ cm}^{-3}$  of arsenic doping concentration, respectively. As for the high-k/metal gate (HKMG) stack, the equivalent oxide thickness (EOT) is 1.35 nm, which consists of  $\text{HfO}_2$  of 2 nm and interfacial oxide  $\text{SiO}_2$  of 1 nm. The work-function metal used in the gate stack is TiN and the effective work-function ( $WF$ ) is set to 4.4 eV. Note that for high-performance devices, the geometric parameters are the same as above except for the nanosheet width being wider.

**Table 1.** Detailed parameters of nominal device at 2 nm technology node [33].

Parameters	Value
Physical gate length ( $L_g$ )	14 nm
Source/drain length ( $L_{sd}$ )	12 nm
Spacer length ( $L_{sp}$ )	6 nm
Nanosheet width ( $W_{sh}$ )	15 nm
Nanosheet thickness ( $T_{sh}$ )	6 nm
Sheet-to-sheet spacing ( $T_{sp}$ )	10 nm
Equivalent oxide thickness (EOT)	1.35 nm
Source/drain doping concentration ( $N_{sd}$ )	$5 \times 10^{20} \text{ cm}^{-3}$
Channel doping concentration ( $N_{ch}$ )	$1 \times 10^{10} \text{ cm}^{-3}$
Metal gate work-function ( $WF$ )	4.4 eV



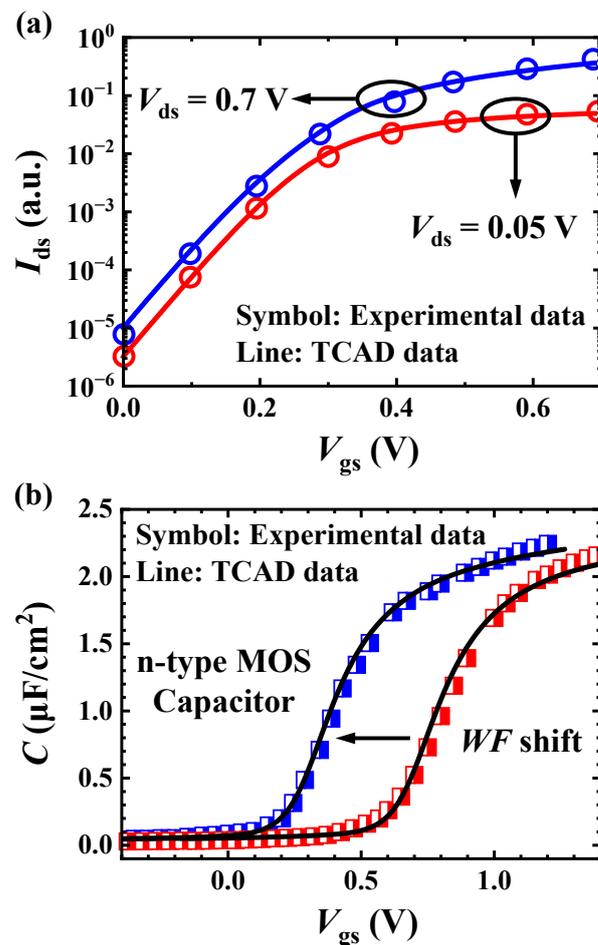
**Figure 1.** An illustration of nominal gate-all-around nanosheet FET (GAA NSFET) and details of device structure : (a) Entire 3-D schematic; (b) X–Z cut plane and (c) Y–Z cut plane of the nominal device structure.

### 2.2. TCAD Simulation Calibration

Since nanoscale devices typically exhibit size-dependent behavior, the corresponding physical model parameters built-in in the TCAD simulator may not be accurate enough

with the scale shrinking, which affects the validity of the device characteristics resulted from TCAD simulations. Therefore, in order to ensure the accuracy of the subsequent simulations to generate more physically accurate datasets for the subsequent ANN model, it is essential to calibrate the simulator against experimental data to lay a solid ground for the ANN modeling work. In this calibration work, both DC and AC characteristics were covered, comprehensively demonstrating the exactitude of the simulation platform.

Besides the nominal GAA NSFET structure illustrated in the previous section for DC calibration, an n-type MOS capacitor was generated according to the device description for AC calibration [35]. TCAD calibrations against experimental data of Refs. [10,35] and were performed in the framework of drift-diffusion (DD) transport model with quantum correction in electrostatics. The results are shown in Figure 2a,b, where the calibrated simulator closely matches the experimental  $I_{ds}$ - $V_{gs}$  and  $C$ - $V$  characteristics after adjustments of the relevant model parameters. The physical models used include the Philip unified mobility, thin-layer mobility and high-field saturation models, as well as Shockley–Read–Hall (SRH) recombination, Auger recombination and band-to-band tunneling models in the drift-diffusion (DD) framework. Physically more correct, Fermi-Dirac statistics are used for high doping concentrations. Furthermore, the density-gradient and kinetic velocity models are considered to account for quantum confinement and ballistic effects.



**Figure 2.** TCAD simulation calibrations against the experimental data under the same simulation environment. (a) Calibrated  $I_{ds}$ - $V_{gs}$  characteristics of n-type NSFET versus experiment data from Ref. [10], and (b)  $C$ - $V$  characteristics of n-type MOS capacitor versus experimental data from Ref. [35].

### 2.3. Dataset Generation

Based on the previously calibrated simulation environment, a number of GAA NSFETs were designed by altering the nanosheet dimensions ( $L_g$ ,  $W_{sh}$ , and  $T_{sh}$ ) of the nominal device. And the ranges of dimensional variants were designed to cover the specifications of IRDS roadmap organized for 3 nm to 1 nm nodes [33]. Here,  $L_g$  ranges from 10 to 20 nm,  $W_{sh}$  ranges from 15 to 30 nm, while  $T_{sh}$  has a smaller movable range between 4–7 nm. Then, the DC and AC characteristics were extracted to create dataset when  $V_{ds}$  and  $V_{gs}$  were set at 0–0.7 V. For the C-V model, the AC characteristics were obtained with a frequency of  $10^6$  Hz. In the practical simulation experiments, we can flexibly control the number of points taken in the electric characteristic curves. Considering that too much data generated by TCAD are very likely to cause overfitting and waste of computing resource, we finally randomly selected 4000 sets of data to form the dataset used for the subsequent study.

## 3. Development and Optimization of ANN Model

### 3.1. Development of ANN Model

Figure 3 shows the proposed schematic diagram of developing a regression ANN model, which is executed in the following steps: (1) accepting the input data, (2) fine-tuning the input and output parameters while training the model, (3) testing and (4) evaluating the trained model using the testing data. A complete five parameters are used as input variants, including gate-to-source voltage ( $V_{gs}$ ), drain-to-source voltage ( $V_{ds}$ ), physical gate length ( $L_g$ ), nanosheet width ( $W_{sh}$ ), and nanosheet thickness ( $T_{sh}$ ). The training/testing data, which comprising DC and AC characteristics for various input parameters, is obtained from physical TCAD simulations. The hidden layers consist of two layers, with  $k$  ( $k = 10$ ) and  $s$  ( $s = 5$ ) neurons respectively. The number of neurons in the output layer is  $p$  ( $p = 4$ ), one is used for the  $I$ - $V$  model, and the other three are used for the C- $V$  model. Besides, we define the conversion function for mapping the output values of the ANN model to the real current  $I_{ds}$  and the capacitance  $C_{g,g}$ ,  $C_{g,d}$  and  $C_{g,s}$ . The training of the ANN model is realized using python with the assistance of the PyTorch package.

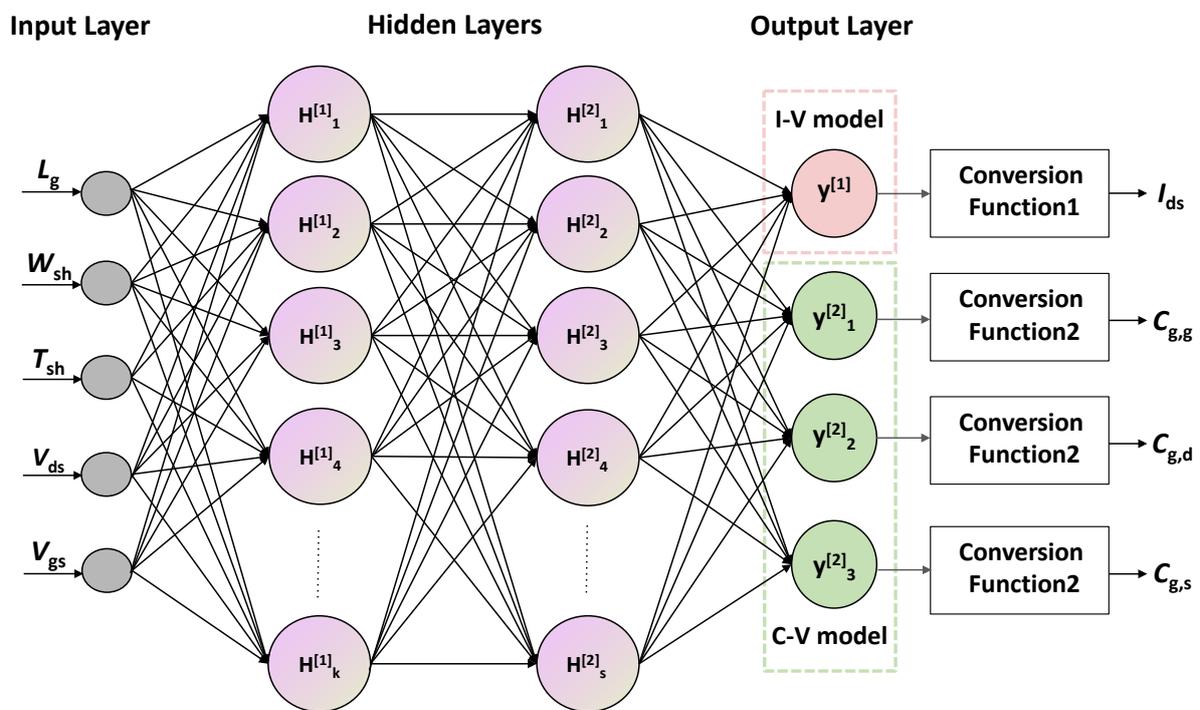


Figure 3. The regression neural network topology framework.

In our ANN model, each hidden layer consists of multiple neurons, and the connections between neurons are characterized by weights  $w$  and biases  $b$ . The mathematical

foundation of ANNs relies on the activation function, often denoted as  $f$ , which introduces non-linearity into the model. The training of the network includes two steps: the forward process and the backward process. The forward process of the network involves calculating the weighted sum of inputs, applying the activation function, and passing the result to the next layer. This process is repeated layer by layer until the final output is obtained. The mathematical representation of the forward process can be expressed as follows:

$$net_j^{(k)} = \sum_{i=1}^{n^{(k-1)}} w_{j,i}^{(k)} * y_i^{(k-1)} + b_j^{(k)} \quad (1)$$

$$y_j^{(k)} = f(net_j^{(k)}) \quad (2)$$

Here,  $net_j^{(k)}$  represents the weighted sum of inputs for neuron  $j$  in layer  $k$ ,  $w_{j,i}^{(k)}$  denotes the weight connecting neuron  $i$  in layer  $k - 1$  to neuron  $j$  in layer  $k$ ,  $y_i^{(k-1)}$  is the output of neuron  $i$  in layer  $k - 1$ ,  $b_j^{(k)}$  is the bias for neuron  $j$  in layer  $k$ , and  $f(net_j^{(k)})$  is the activation function. Especially, the hyperbolic tangent function  $\tanh(x)$  was used as the activation function [36,37]. The output of  $\tanh(x)$  lies within the range of  $[-1, 1]$ , which, compared to the  $[0, 1]$  range of the sigmoid function, makes  $\tanh(x)$  advantageous in zero-centering. This helps mitigate the exploding gradient problem during gradient descent.

The training process involves minimizing a predefined loss function, typically the mean squared error (MSE), which measures the discrepancy between the predicted and actual outputs. MSE is calculated by taking the average of the squared differences between predicted and actual values, which is a simple and easily differentiable form. This simplicity facilitates the updating of weights in optimization algorithms like gradient descent. In addition, as MSE involves squaring the errors, it is less sensitive to outliers (samples with significantly different actual values). This means that individual outliers do not have a disproportionately large impact on the overall loss function, enhancing the robustness of the model.

The backward process, also known as backpropagation, is a crucial step in updating the network parameters. The gradients are propagated backward through the network, and the weights and biases are adjusted using optimization algorithms such as stochastic gradient descent (SGD) [38]. The chain rule is applied iteratively to compute the gradients of the loss ( $L$ ) with respect to the network parameters:

$$\frac{\partial L}{\partial w_{j,i}^{(k)}} = \frac{\partial L}{\partial net_j^{(k)}} * \frac{\partial net_j^{(k)}}{\partial w_{j,i}^{(k)}} \quad (3)$$

$$\frac{\partial L}{\partial b_j^{(k)}} = \frac{\partial L}{\partial net_j^{(k)}} * \frac{\partial net_j^{(k)}}{\partial b_j^{(k)}} \quad (4)$$

These gradients guide the parameters update during the training process, gradually optimizing the network to improve its predictive capabilities. The iterative nature of backpropagation allows the network to learn complex patterns and relationships within the data.

Thus, a four-layered regression ANN involves intricate mathematical formulations, including the forward pass equations for computing neuron activations and the backward pass equations for updating weights and biases during training. The application of the chain rule in calculus is fundamental to these computations, enabling the network to learn and adapt to complex patterns in the data.

### 3.2. Optimization of ANN Model

Before the training process, we noticed that the orders of magnitude of the outputs are too small,  $10^{-13} \sim 10^{-3}$  for the  $I-V$  model and  $10^{-18} \sim 10^{-17}$  for the  $C-V$  model, which

are not favorable for data fitting. So, we preprocessed the outputs ( $I_{ds}$ ,  $C_{g,g}$ ,  $C_{g,d}$ , and  $C_{g,s}$ ) in order to achieve the accurate fitting through a linear preprocessing method. Here, we multiplied the output currents and capacitances by factors of  $1 \times 10^6$  and  $1 \times 10^{18}$ , respectively, thereby converting the units from A and F to  $\mu\text{A}$  and aF.

Since then, the processed dataset was utilized for training, but another problem was identified, namely, the ANN model had overfitting, which means it performs well during training but fails to generalize effectively to the test samples. In other words, our model has a significant gap between the model's performance during training and its performance when making predictions on new data. The network excels in fitting the training data but struggles to make accurate predictions on unseen examples.

Overfitting often leads to excessively complex neural network models. These models tend to capture noise and outliers in the training data, making them less suitable for generalization. Moreover, the loss function used during training may not accurately reflect the network's performance on new data. The model might minimize the training loss, giving a false sense of success, while failing to minimize the loss on validation or test data. Let  $L_{\text{train}}$  denotes the training loss,  $L_{\text{val}}$  the validation loss, and  $L_{\text{test}}$  the test loss. Overfitting occurs when  $L_{\text{train}}$  is significantly smaller than both  $L_{\text{val}}$  and  $L_{\text{test}}$ .

$$L_{\text{train}} \ll L_{\text{val}}, L_{\text{test}} \quad (5)$$

To address this issue, we adopt  $L2$  regularization (also known as weight decay) [39], which is a widely adopted technique to address overfitting by adding a penalty term to the loss function. The regularized loss function is given by:

$$L = \frac{1}{2} \|Xw - y\|^2 + \lambda \|w\|^2 \quad (6)$$

Here,  $X$  is the input matrix,  $w$  is the weight vector,  $y$  is the target vector, and  $\lambda$  is the regularization parameter that controls the strength of the regularization. The first term  $\frac{1}{2} \|Xw - y\|^2$  represents the MSE (described in Section 3.1), aiming to minimize the difference between the predicted and actual values. The second term  $\lambda \|w\|^2$  is the  $L2$  regularization term. It penalizes large weights by adding the squared magnitude of the weight vector. The regularization parameter  $\lambda$  controls the trade-off between fitting the training data and preventing overfitting.

From the viewpoint of convex optimization, the introduction of the  $L2$  regularization term transforms the optimization problem into a constrained optimization problem. The regularization term induces a constraint on the magnitude of the weight vector, effectively defining a hypersphere in the weight space. This transformation has a smoothing effect on the optimization landscape, making it more convex. The regularization term adds a regularization force that discourages the weights from reaching extreme values, leading to a more stable and generalizable model.

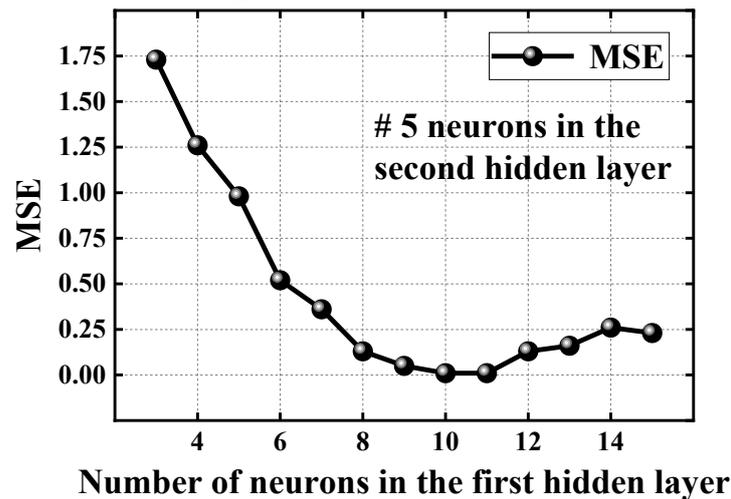
In summary,  $L2$  regularization mitigates overfitting by penalizing large weights in a linear regression model. The mathematical formulation introduces a balance between fitting the training data and controlling the complexity of the model. From a convex optimization perspective, the regularization term induces a constraint that shapes a more well-behaved optimization landscape.

#### 4. Results and Discussion

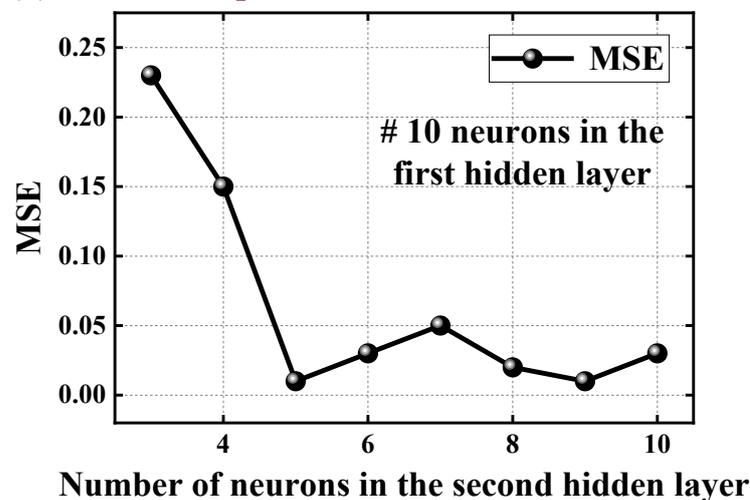
A total of 4000 samples for ANN training and testing are obtained from  $I_{ds}$ - $V_{gs}$  and  $C$ - $V$  data generated by previous TCAD simulations. We randomly split these samples into a training set (a total of 3200 samples) and a testing set (a total of 800 samples) in a 4:1 ratio. Theoretically, as the number of hidden layers and neurons increases, the ANN model becomes more capable of extracting the non-linear mapping relationship between input and output. However, in practice, too many hidden layers or number of neurons can also bring about overfitting problems. And in most cases, the fitting accuracy is determined

synergistically by both the number of hidden layers and neurons. For most fitting cases with limited input and output variants, a shallow neural network is sufficient, which is easier to be trained and converges to the optimal solution faster, with a more favorable computational and memory footprint. Thus, to obtain an optimal network, we studied the impact of network sizes on the errors (MSE) for shallow neural networks with two hidden layers. As shown in Figure 4, we find that the MSE of the testing set tends to decrease and then increase as the number of neurons increases. The minimum MSE is 0.01 with ten neurons in the first hidden layer and five neurons in the second hidden layer.

(a) **The impact of network sizes on MSE**



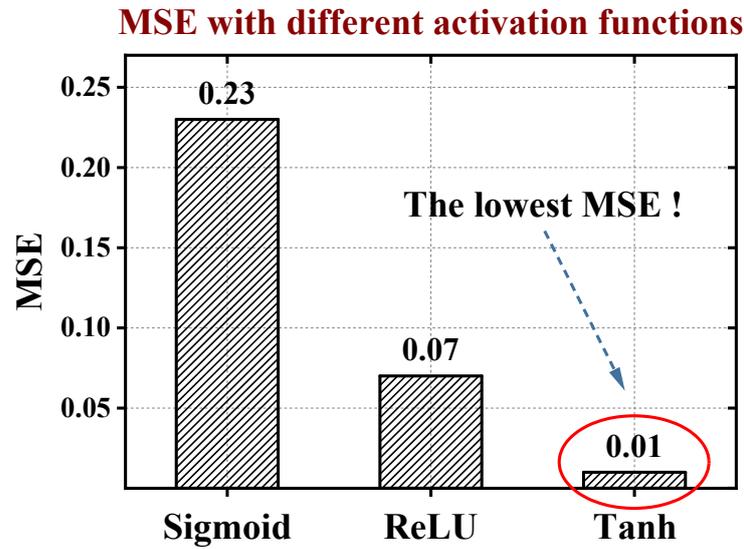
(b) **The impact of network sizes on MSE**



**Figure 4.** The MSE for all the test samples with different numbers of neurons in the first hidden layer (a) and second hidden layer (b).

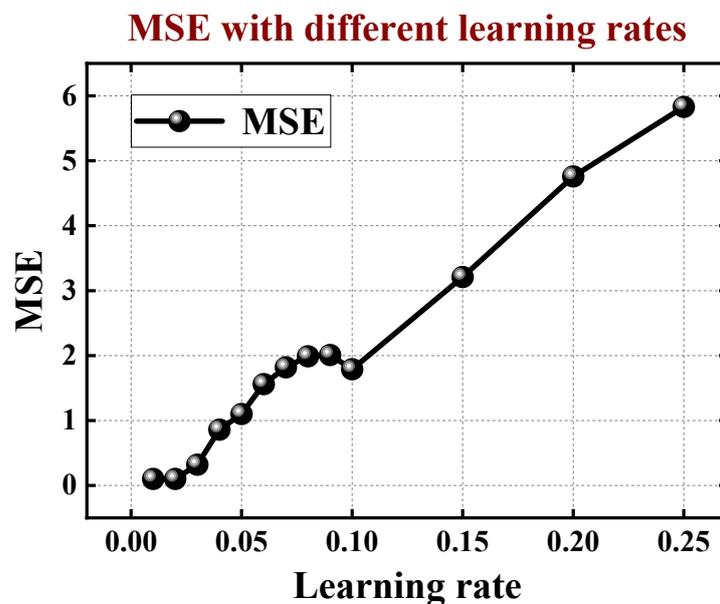
Besides, we investigated the impact of different types of activation functions of the neurons on MSE for the test dataset, as depicted in Figure 5. Using the hyperbolic tangent function  $\tanh(x)$  has the lowest MSE. Through our analysis, the  $\tanh(x)$  function is zero-centered, meaning its mean is zero. This is beneficial for optimization algorithms such as gradient descent, as it helps prevent the gradient updates from consistently favoring a particular direction, thus improving the convergence speed of the model. The derivative of the  $\tanh(x)$  function is non-zero in most regions, aiding in the propagation of gradients during backpropagation. Unlike the sigmoid function, the gradient of  $\tanh(x)$  does not

approach zero in regions of large or small inputs, reducing the risk of the vanishing gradient problem.



**Figure 5.** The MSE with different activation functions of the neurons. Popular activation functions: sigmoid, relu, and tanh.

Moreover, we have studied the impact of the learning rate on MSE. In our ANN model, the learning rate is a crucial hyper parameter in training neural networks. It controls the magnitude of updates applied to the weights during the training process. A higher learning rate means larger updates, leading to faster convergence but with the risk of overshooting the optimal weights. Conversely, a lower learning rate allows for smaller weight updates, potentially resulting in slower convergence but increased precision in finding the global minimum. Through our experiments, we think learning rate of 0.02 is the best solution for the ANN model as shown in Figure 6.



**Figure 6.** The MSE with different learning rates.

Epochs represent the number of times the entire dataset is fed forward and backward through the neural network during the training process. The choice of the number of epochs plays a pivotal role in determining how well the model generalizes to unseen data. Too few

epochs may result in under fitting, where the model fails to capture the underlying patterns in the data. On the other hand, an excessive number of epochs may lead to overfitting, causing the model to memorize the training data but perform poorly on new, unseen data. Moreover, the relationship between epoch and learning rate is interdependent. A higher learning rate may require fewer epochs to converge, as each iteration leads to more substantial weight updates. Conversely, a lower learning rate might necessitate a higher number of epochs to allow the model to converge gradually. Finally, as shown in Figure 7, we choose the lowest MSE (0.01) scheme with the epoch of 5000 and the learning rate of 0.02.

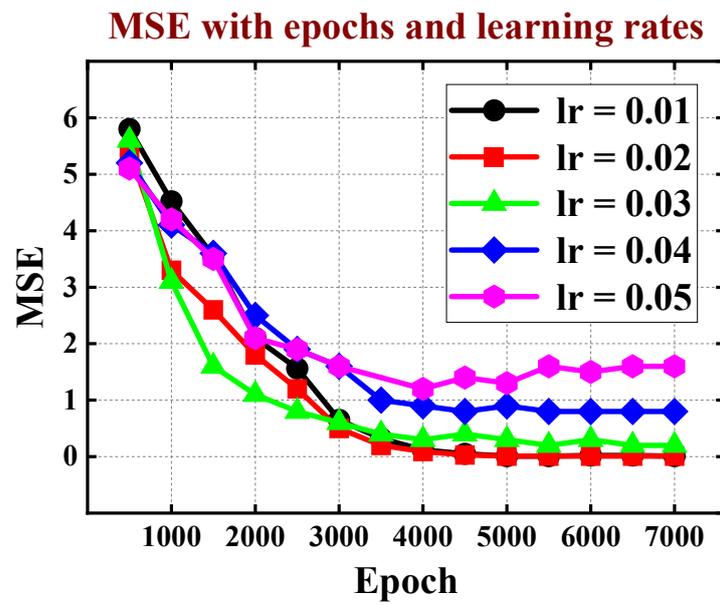
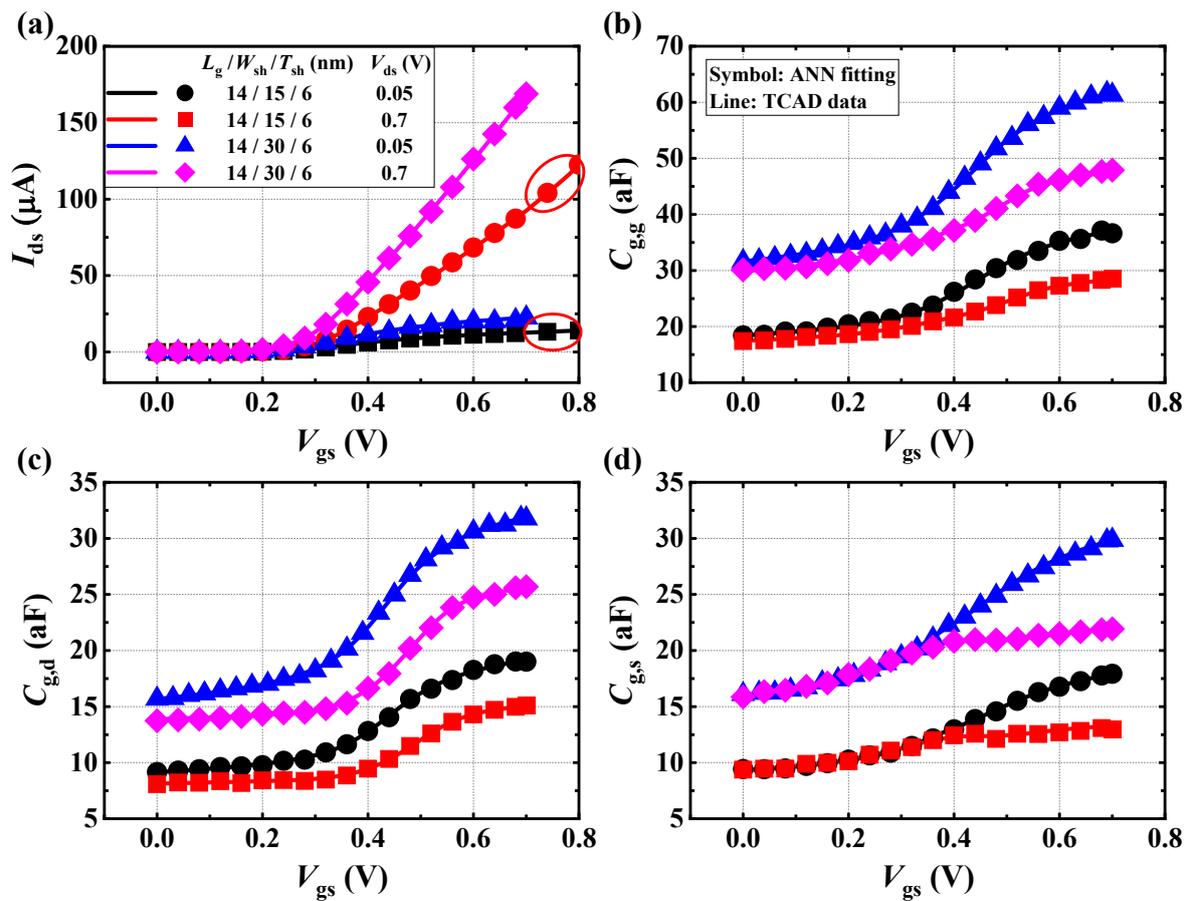


Figure 7. The MSE decline process with different learning rates as epoch increases.

Here, we summarize the primary parameters of the proposed ANN model, as shown in Table 2. Based on the experiments and analysis, we can utilize the ANN model to predict the current and capacitance output based on input data ( $L_g$ ,  $W_{sh}$ ,  $T_{sh}$ ,  $V_{gs}$  and  $V_{ds}$ ) with MSE of 0.01. As shown in Figure 8, the example results of both the DC and AC characteristics of ANN model are fitted against the TCAD data of high-density and high-performance GAA NSFETs at 2 nm technology node. It can be seen that the output  $I-V$  and  $C-V$  performances generated by ANN fit well with the TCAD results. In addition, we predicted the  $I-V$  performance of the nominal NSFET device at high gate bias ( $V_{gs} = 0.7-0.8$  V) to examine the model scalability. The extrapolation behavior also fits well to simulation results. The optimistic results reveal that the proposed network is capable of handling the electrical characterization of advanced GAA NSFETs with great accuracy.

Table 2. The primary parameters of the ANN model.

Parameters	Features
Network size	5-10-5-4
Activation function	Hyperbolic tangent function
Learning rate	0.02
Epoch	5000
#Training samples	3200
#Test samples	800
Task	Regression
MSE	0.01
Regularization	L2 Regularization



**Figure 8.** Example ANN model fitting results of the simulated DC and AC characteristics for high-density and high-performance GAA NSFETs at 2 nm technology node. (a)  $I_{ds}$ - $V_{gs}$ . (b)  $C_{gg}$ - $V_{gs}$ . (c)  $C_{gd}$ - $V_{gs}$ . (d)  $C_{gs}$ - $V_{gs}$ .

## 5. Conclusions

In summary, the ANN-based compact modeling methodology has been thoroughly investigated for advanced GAA NSFETs. Here, the impacts of ANN size, activation function, learning rate, and epoch on the accuracy of ANN models were systematically evaluated. Based on the precisely calibrated simulation environment, various GAA NSFET devices were constructed by varying the nanosheet dimensions, and their DC as well as AC characteristics were extracted. The generated dataset contains five input variants ( $V_{gs}$ ,  $V_{ds}$ ,  $L_g$ ,  $W_{sh}$ , and  $T_{sh}$ ) and four output quantities ( $I_{ds}$ ,  $C_{gg}$ ,  $C_{gd}$ , and  $C_{gs}$ ). Before the training process, the output data were preprocessed to circumvent unnecessary fitting mistakes using a linear preprocessing method. By adopting the  $L_2$  regularization, the overfitting issue was perfectly resolved with the addition of a penalty term to the loss function. The optimized ANN model fully demonstrates its superior fitting properties under various conditions with a low fitting MSE error of 0.01. Furthermore, the scalability was also validated. This work contributes to the development of ANN-based compact models, holding great promise for adoption in advanced fast turn-around design and technology co-optimization (DTCO) as well as large-scale product-design-oriented circuit simulations.

**Author Contributions:** Conceptualization, Y.Z. (Yage Zhao); methodology, Y.Z. (Yage Zhao) and S.Y.; investigation, Y.Z. (Yage Zhao), S.Y. and Z.X.; writing—original draft preparation, Y.Z. (Yage Zhao); writing—review and editing, Y.Z. (Yage Zhao), S.Y., Z.X., H.T., Y.Z. (Yusi Zhao), P.T., R.D., X.Z. and D.W.Z.; supervision, S.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by a platform for the development of next generation integrated circuit technology and Shanghai Sailing Program, 20YF1401700.

**Data Availability Statement:** The data and code are available from the corresponding authors upon reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Hisamoto, D.; Lee, W.C.; Kedzierski, J.; Takeuchi, H.; Asano, K.; Kuo, C.; Anderson, E.; King, T.J.; Bokor, J.; Hu, C. FinFET-a self-aligned double-gate MOSFET scalable to 20 nm. *IEEE Trans. Electron Devices* **2000**, *47*, 2320–2325. [[CrossRef](#)]
2. Jan, C.H.; Bhattacharya, U.; Brain, R.; Choi, S.J.; Currello, G.; Gupta, G.; Hafez, W.; Jang, M.; Kang, M.; Komeyli, K.; et al. A 22 nm SoC platform technology featuring 3-D tri-gate and high-k/metal gate, optimized for ultra low power, high performance and high density SoC applications. In Proceedings of the 2012 International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 10–13 December 2012; pp. 3.1.1–3.1.4.
3. Sell, B.; Bigwood, B.; Cha, S.; Chen, Z.; Dhage, P.; Fan, P.; Giraud-Carrier, M.; Kar, A.; Karl, E.; Ku, C.J.; et al. 22FFL: A high performance and ultra low power FinFET technology for mobile and RF applications. In Proceedings of the 2017 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2–6 December 2017; pp. 29.24.21–29.24.24.
4. Yeap, G.; Lin, S.S.; Chen, Y.M.; Shang, H.L.; Wang, P.W.; Lin, H.C.; Peng, Y.C.; Sheu, J.Y.; Wang, M.; Chen, X.; et al. 5 nm CMOS Production Technology Platform featuring full-fledged EUV, and High Mobility Channel FinFETs with densest 0.021  $\mu\text{m}^2$  SRAM cells for Mobile SoC and High Performance Computing Applications. In Proceedings of the 2019 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 7–11 December 2019; pp. 36.37.31–36.37.34.
5. Liu, J.C.; Mukhopadhyay, S.; Kundu, A.; Chen, S.H.; Wang, H.C.; Huang, D.S.; Lee, J.H.; Wang, M.I.; Lu, R.; Lin, S.S.; et al. A Reliability Enhanced 5nm CMOS Technology Featuring 5th Generation FinFET with Fully-Developed EUV and High Mobility Channel for Mobile SoC and High Performance Computing Application. In Proceedings of the 2020 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 12–18 December 2020; pp. 9.2.1–9.2.4.
6. Ding, Y.; Luo, X.; Shang, E.; Hu, S.; Chen, S.; Zhao, Y. A Device Design for 5 nm Logic FinFET Technology. In Proceedings of the 2020 China Semiconductor Technology International Conference (CSTIC), Shanghai, China, 26 June–17 July 2020; pp. 1–5.
7. Chang, C.H.; Chang, V.S.; Pan, K.H.; Lai, K.T.; Lu, J.H.; Ng, J.A.; Chen, C.Y.; Wu, B.F.; Lin, C.J.; Liang, C.S.; et al. Critical Process Features Enabling Aggressive Contacted Gate Pitch Scaling for 3 nm CMOS Technology and Beyond. In Proceedings of the 2022 International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 3–7 December 2022; pp. 27.21.21–27.21.24.
8. Chung, S.S.; Chiang, C.K.; Pai, H.; Hsieh, E.R.; Guo, J.C. The Extension of the FinFET Generation Towards Sub-3 nm: The Strategy and Guidelines. In Proceedings of the 2022 6th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), Oita, Japan, 6–9 March 2022; pp. 15–17.
9. Feng, P.; Song, S.; Nallapati, G.; Zhu, J.; Bao, J.; Moroz, V.; Choi, M.; Lin, X.; Lu, Q.; Colombeau, B.; et al. Comparative Analysis of Semiconductor Device Architectures for 5-nm Node and beyond. *IEEE Electron Device Lett.* **2017**, *38*, 1657–1660. [[CrossRef](#)]
10. Loubet, N.; Hook, T.; Montanini, P.; Yeung, C.W.; Kanakasabapathy, S.; Guillom, M.; Yamashita, T.; Zhang, J.; Miao, X.; Wang, J.; et al. Stacked nanosheet gate-all-around transistor to enable scaling beyond FinFET. In Proceedings of the 2017 Symposium on VLSI Technology, Kyoto, Japan, 5–8 June 2017; pp. T230–T231.
11. Das, U.K.; Bhattacharyya, T.K. Opportunities in Device Scaling for 3-nm Node and Beyond: FinFET versus GAA-FET versus UFET. *IEEE Trans. Electron Devices* **2020**, *67*, 2633–2638. [[CrossRef](#)]
12. Ritzenthaler, R.; Mertens, H.; Eneman, G.; Simoen, E.; Bury, E.; Eyben, P.; Bufler, F.M.; Oniki, Y.; Briggs, B.; Chan, B.T.; et al. Comparison of Electrical Performance of Co-Integrated Forksheets and Nanosheets Transistors for the 2nm Technological Node and Beyond. In Proceedings of the 2021 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 11–16 December 2021; pp. 26.22.21–26.22.24.
13. Strohhahn, K.; Martin, M.N. SPICE macro models for annular MOSFETs. In Proceedings of the 2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720), Big Sky, MT, USA, 6–13 March 2004; pp. 2370–2377.
14. Oh, J.H.; Yu, Y.S. Macro-Modeling for N-Type Feedback Field-Effect Transistor for Circuit Simulation. *Micromachines* **2021**, *12*, 1174. [[CrossRef](#)] [[PubMed](#)]
15. Song, J.; Yuan, Y.; Yu, B.; Xiong, W.; Taur, Y. Compact Modeling of Experimental n- and p-Channel FinFETs. *IEEE Trans. Electron Devices* **2010**, *57*, 1369–1374. [[CrossRef](#)]
16. Ding, J.; Asenov, A. Reliability-Aware Statistical BSIM Compact Model Parameter Generation Methodology. *IEEE Trans. Electron Devices* **2020**, *67*, 4777–4783. [[CrossRef](#)]
17. Wu, T.; Luo, H.; Wang, X.; Asenov, A.; Miao, X. A Predictive 3-D Source/Drain Resistance Compact Model and the Impact on 7 nm and Scaled FinFETs. *IEEE Trans. Electron Devices* **2020**, *67*, 2255–2262. [[CrossRef](#)]
18. Jung, S.G.; Kim, J.K.; Yu, H.Y. Analytical Model of Contact Resistance in Vertically Stacked Nanosheet FETs for Sub-3-nm Technology Node. *IEEE Trans. Electron Devices* **2022**, *69*, 930–935. [[CrossRef](#)]
19. Wang, J.; Xu, N.; Woosung, C.; Keun-Ho, L.; Youngkwan, P. A generic approach for capturing process variations in lookup-table-based FET models. In Proceedings of the 2015 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD), Washington, DC, USA, 9–11 September 2015; pp. 309–312.
20. Thakker, R.A.; Sathe, C.; Sachid, A.B.; Baghini, M.S.; Rao, V.R.; Patil, M.B. A Novel Table-Based Approach for Design of FinFET Circuits. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2009**, *28*, 1061–1070. [[CrossRef](#)]

21. Sheu, B.; Scharfetter, D.; Hu, C.; Pederson, D. A compact IGFET charge model. *IEEE Trans. Circuits Syst.* **1984**, *31*, 745–748. [[CrossRef](#)]
22. Duarte, J.P.; Khandelwal, S.; Medury, A.; Hu, C.; Kushwaha, P.; Agarwal, H.; Dasgupta, A.; Chauhan, Y.S. BSIM-CMG: Standard FinFET compact model for advanced circuit design. In Proceedings of the ESSCIRC Conference 2015-41st European Solid-State Circuits Conference (ESSCIRC), Graz, Austria, 14–18 September 2015; pp. 196–201.
23. Singh, S.K.; Gupta, S.; Vega, R.A.; Dixit, A. Accurate Modeling of Cryogenic Temperature Effects in 10-nm Bulk CMOS FinFETs Using the BSIM-CMG Model. *IEEE Electron Device Lett.* **2022**, *43*, 689–692. [[CrossRef](#)]
24. Litovski, V.B.; Radjenović, J.I.; Mrčarica, Ž.M.; Milenković, S.L. MOS transistor modelling using neural network. *Electron. Lett.* **1992**, *28*, 1766–1768. [[CrossRef](#)]
25. Jain, A.K.; Jianchang, M.; Mohiuddin, K.M. Artificial neural networks: A tutorial. *Computer* **1996**, *29*, 31–44. [[CrossRef](#)]
26. Saha, N.; Swetapadma, A.; Mondal, M. A Brief Review on Artificial Neural Network: Network Structures and Applications. In Proceedings of the 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 17–18 March 2023; pp. 1974–1979.
27. Ryckaert, J.; Schuddinck, P.; Weckx, P.; Bouche, G.; Vincent, B.; Smith, J.; Sherazi, Y.; Mallik, A.; Mertens, H.; Demuynck, S.; et al. The Complementary FET (CFET) for CMOS scaling beyond N3. In Proceedings of the 2018 IEEE Symposium on VLSI Technology, Honolulu, HI, USA, 18–22 June 2018; pp. 141–142.
28. Subramanian, S.; Hosseini, M.; Chiarella, T.; Sarkar, S.; Schuddinck, P.; Chan, B.T.; Radisic, D.; Mannaert, G.; Hikavy, A.; Rosseel, E.; et al. First Monolithic Integration of 3D Complementary FET (CFET) on 300 mm Wafers. In Proceedings of the 2020 IEEE Symposium on VLSI Technology, Honolulu, HI, USA, 16–19 June 2020; pp. 1–2.
29. Ko, K.; Lee, J.K.; Kang, M.; Jeon, J.; Shin, H. Prediction of Process Variation Effect for Ultrascaled GAA Vertical FET Devices Using a Machine Learning Approach. *IEEE Trans. Electron Devices* **2019**, *66*, 4474–4477. [[CrossRef](#)]
30. Butola, R.; Li, Y.; Kola, S.R.; Chen, C.Y.; Chuang, M.H. Artificial Neural Network-Based Modeling for Estimating the Effects of Various Random Fluctuations on DC/Analog/RF Characteristics of GAA Si Nanosheet FETs. *IEEE Trans. Microw. Theory Tech.* **2022**, *70*, 4835–4848. [[CrossRef](#)]
31. Qi, G.; Chen, X.; Hu, G.; Zhou, P.; Bao, W.; Lu, Y. Knowledge-based neural network SPICE modeling for MOSFETs and its application on 2D material field-effect transistors. *Sci. China Inf. Sci.* **2023**, *66*, 122405. [[CrossRef](#)]
32. Wei, J.; Wang, H.; Zhao, T.; Jiang, Y.L.; Wan, J. A New Compact MOSFET Model Based on Artificial Neural Network with Unique Data Preprocessing and Sampling Techniques. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 1250–1254. [[CrossRef](#)]
33. IRDS. International Roadmap for Devices and Systems 2022 (IRDS 2022). Available online: <https://irds.ieee.org/> (accessed on 31 July 2022).
34. Synopsys, Inc. *Sentaurus TCAD User's Manual*; Synopsys Inc.: Mountain View, CA, USA, 2019.
35. Arimura, H.; Ragnarsson, L.Å.; Oniki, Y.; Franco, J.; Vandooren, A.; Brus, S.; Leonhardt, A.; Sippola, P.; Ivanova, T.; Verni, G.A.; et al. Dipole-First Gate Stack as a Scalable and Thermal Budget Flexible Multi-Vt Solution for Nanosheet/CFET Devices. In Proceedings of the 2021 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 11–16 December 2021; pp. 13.15.11–13.15.14.
36. Zamanlooy, B.; Mirhassani, M. Efficient VLSI Implementation of Neural Networks With Hyperbolic Tangent Activation Function. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2014**, *22*, 39–48. [[CrossRef](#)]
37. Lau, M.M.; Lim, K.H. Investigation of activation functions in deep belief network. In Proceedings of the 2017 2nd International Conference on Control and Robotics Engineering (ICCRE), Bangkok, Thailand, 1–3 April 2017; pp. 201–206.
38. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
39. Gupta, S.; Gupta, R.; Ojha, M.; Singh, K.P. A Comparative Analysis of Various Regularization Techniques to Solve Overfitting Problem in Artificial Neural Network. In Proceedings of the Data Science and Analytics, Singapore, 8 March 2018; pp. 363–371.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.