This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079981, IEEE Access

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Deep Learning Algorithms for the Work Function Fluctuation of Random Nanosized Metal Grains on Gate-All-Around Silicon Nanowire MOSFETs

Chandni Akbar[1,2,6,7], Yiming Li[1-10,*] and Wen Li Sung[1,4,6,9]

[1] Parallel and Scientific Computing Laboratory, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan
[2] Electrical Engineering and Computer Science International Graduate Program, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan
[3] Department of Electrical Engineering and Computer Engineering, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan
[4] Institute of Communications Engineering, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan
[5] Center for mmWave Smart Radar System and Technologies, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan
[6] Parallel and Scientific Computing Laboratory, National Chiao Tung University, Hsinchu 300, Taiwan
[7] Electrical Engineering and Computer Science International Graduate Program, National Chiao Tung University, Hsinchu 300, Taiwan
[8] Department of Electrical Engineering and Computer Engineering, National Chiao Tung University, Hsinchu 300, Taiwan
[9] Institute of Communications Engineering, National Chiao Tung University, Hsinchu 300, Taiwan
[10] Center for mmWave Smart Radar System and Technologies, National Chiao Tung University, Hsinchu 300, Taiwan

[*]Corresponding author: ymli@faculty.nctu.edu.tw

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

ABSTRACT Device simulation has been explored and industrialized for over 40 years; however, it still requires huge computational cost. Therefore, it can be further advanced using deep learning (DL) algorithms. We for the first time report an efficient and accurate DL approach with device simulation for gate-all-around silicon nanowire metal-oxide-semiconductor field-effect transistors (MOSFETs) to predict electrical characteristics of device induced by work function fluctuation. By using three different DL models: artificial neural network (ANN), convolutional neural network (CNN), and long short term memory (LSTM), the variability of threshold voltage, on-current and off-current is predicted with respect to different metal-grain number and location of the low and high values of work function. The comparison is established among the ANN, CNN and the LSTM models and results depict that the CNN model outperforms in terms of the root mean squared error and the percentage error rate. The integration of device simulation with DL models exhibits the characteristic estimation of the explored device efficiently; and, the accurate prediction from the DL models can accelerate the process of device simulation. Notably, the DL approach is able to extract crucial electrical characteristics of a complicated device accurately with 2% error in a cost-effective manner computationally.

INDEX TERMS Work function fluctuation, Nanosized metal grain, Gate-all-around, Nanowire, MOSFET, Statistical device simulation, Deep learning, Convolutional neural network, Artificial neural network, Long short term memory, Root mean squared error.

I. INTRODUCTION

Owing to the low-power consumption and straightforward fabrication procedure with high flexibility, silicon-based transistors are acknowledging as the most favorable technology [1-4]. Silicon (Si) transistors offer distinguished functionalities, such as high scalability, high integrity and low-power consumption, etc. [5]. Besides these capabilities, Si transistors have suffered due to various limitations serious issues, such as high leakage current, significant fluctuation of threshold voltage and poor subthreshold slope (SS), etc. [2]. To keep up the continuous downscaling of Si transistors for high-performance applications, gate-all-around (GAA)

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

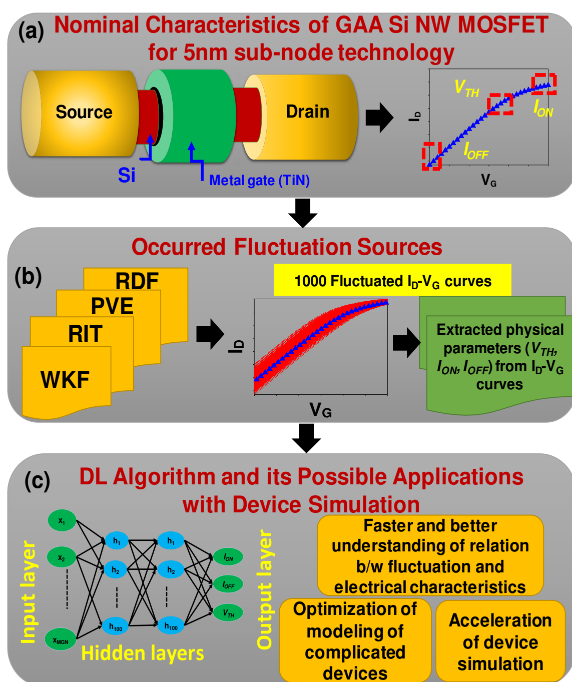

FIGURE 1°. (a) An illustration of the nominal GAA Si NW MOSFET and I_D-V_G curve which depicts the crucial parameters. (b) Various random sources and fluctuated I_D-V_G curves. Device parameters can be extracted from these curves. (c) A general DL model that takes device fluctuations and electrical characteristics as input and target values, respectively. It also demonstrates the possible application of concatenation of device simulation technology with DL algorithms.

Si nanowire (NW) metal-oxide-semiconductor field-effect transistor (MOSFETs) are considered as an arising aspirant among the nanodevices due to predominant electrical characteristics [6]. The fluctuation of electrical characteristic plays a significant role in the field of nanoelectronics. There are various variability sources such as the random dopant fluctuation (RDF), work function fluctuation (WKF), random interface trap (RIT) and the process variation effect (PVE) [7-14]. Moreover, the variability source, the model ambiguity, and the manufacturing tolerance are playing vital roles in chip production and yield optimization [15]. From the prior work [16], three-dimensional (3D) statistical device simulation depicted that the RDF in GAA Si NW MOSFETs greatly affects the device variability. In [17], characteristic fluctuation induced by the WKF on GAA Si NW MOSFETs was studied; and, the results have shown that the reduction of variation of the threshold voltage (V_{TH}) affects the reduction of variation of the frequency. Similarly, in [18], the timings and the power fluctuations were determined by considering the various random discrete dopants (RDDs) on GAA Si NW complementary metal-oxide-semiconductor (CMOS); and, it concluded that the timing fluctuation and the power consumption in CMOS are directly dependent on the variation of the V_{TH}. The variability of the V_{TH} induced by titanium nitride (TiN) metal-gate WKF on GAA Si NW device was examined in [19]; and, comparison of the induced V_{TH} between WKF and RDF indicated that the variability of V_{TH} dominated by WKF has more impact than that of RDF. According to these points of views, electrical characteristics of GAA Si NW MOSFET induced by WKF can be further investigated by using deep learning (DL) algorithms integrated with device simulation.

Recently, machine learning (ML) has been growing prominently in every field due to its scalability and wide range of algorithms and applications [20-23]. ML/DL algorithms are implementing to forecast the unknown future based on known experimental data. For example, in [24], the effect of climate change on urban buildings was studied with the help of ML algorithms. Similarly, in [25], the integration of ML with metabolic engineering was discussed. Likewise, ML was utilized in [26] for the optimization of signal processing algorithms. Moreover, semiconductor and integrated circuit manufacturing industry is highly suitable to be integrated with DL techniques because the semiconductor manufacturing process encounters a large number of parameters and a various number of procedures that are inevitable to be performed manually by engineers. There is some prior research based on the integration of ML with semiconductor manufacturing [27-30]. In [31], a ML algorithm was reported for defect detection. Similarly, a ML algorithm was explored in [32] to yield improvement in semiconductor manufacturing. Nowadays, the integration of ML with the study on GAA Si NW MOSFETs is considered to be significantly feasible and broadly applicable [33-38]. The purpose of applying ML is to predict the characteristics of semiconductor devices. Due to rapid prediction, while maintaining the performance accuracy, ML is being applied in many research areas [39-48]. Moreover, in the variability of the V_{TH} of the GAA Si NW MOSFETs induced by WKF, the random metal grain (MG) and the positional sequence of MG are the complicated factors that motivate the utilization of ML models in MOSFETs. Furthermore, the estimation of the total number of MG in a GAA Si NW, the estimation of an appropriate width of the MG, and the adjustable position of the MG with respect to the different value of WK are intricate processes and vary a lot in determining their parameters. There is some prior work related to the PV integrated with ML. In [49], ML algorithm (artificial neural network; ANN) was applied to predict the V_{TH} of Si junctionless NW transistor by feeding the neural-network model with three input parameters including the off-current (I_{OFF}), the on-current (I_{ON}), and the subthreshold slope (SS). Similarly, in [50], Ko et al. proposed ANN to predict the characteristics of ultra-scaled GAA vertical FET device by using PV. The model was trained by using five variables, four obtained from the PV and one was from the dimension of the device. Then, these input variables were coupled with the target variable which represents the characteristics of VFET device. In [51], Kyul et al. proposed the ANN ML algorithm for 3D NAND flash memories. Due to a simple mathematical model, neural-network was implemented in all these prior work.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079981, IEEE Access

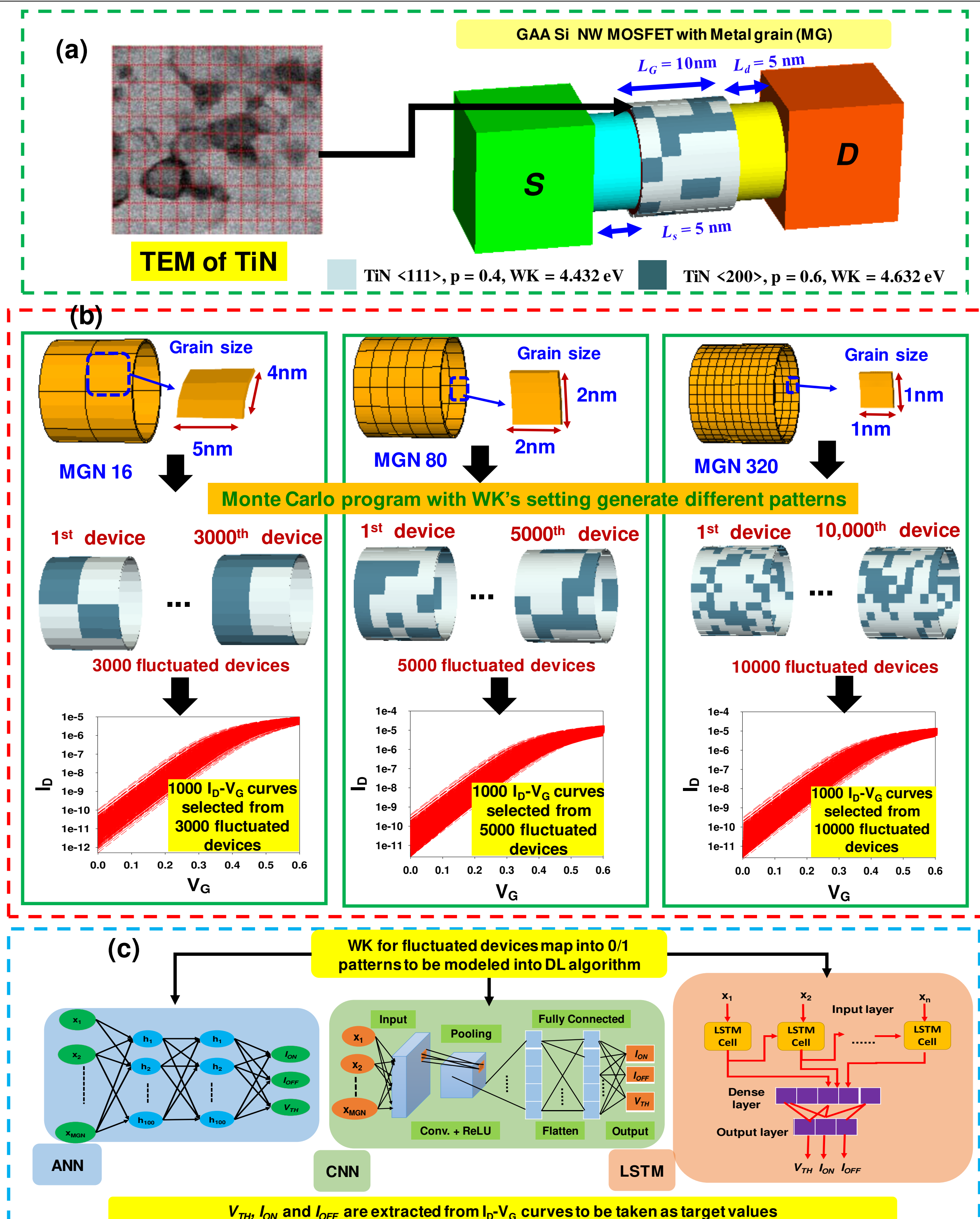

FIGURE 2°. A TEM of TiN which cannot be formulated ideally. Using cuboid grain method, 3D device simulation is performed. (b) The cuboid grain method segregates 3 different grain sizes: MGN = 16, 80 and 320. Corresponding to the cases of MGN = 16, 80 and 320, more than 3000, 5000 and 10000 fluctuated devices are generated and simulated, respectively, and these simulated I_D-V_G curves are illustrated. (d) The extracted device parameters (V_{TH}, I_{ON} and I_{OFF}) and WKF patterns, are feed into three different DL algorithms, i.e., ANN, CNN and LSTM.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079981, IEEE Access

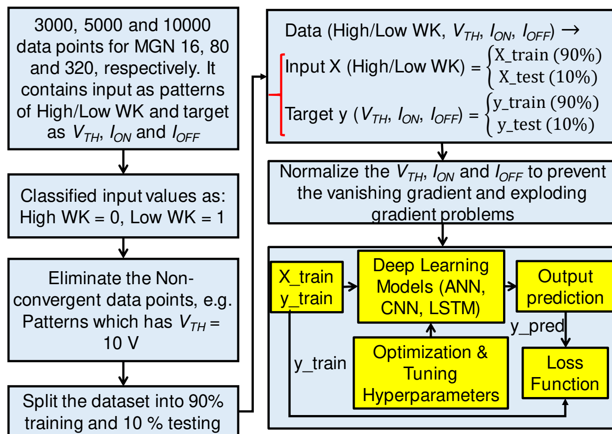

Figure 3°. A comprehensive flow chart of the proposed methodology including data generation, data labeling, noise reduction, splitting the dataset into 90% training and 10% testing, data preprocessing and training and testing the DL models on the basis of optimization algorithm and loss function.

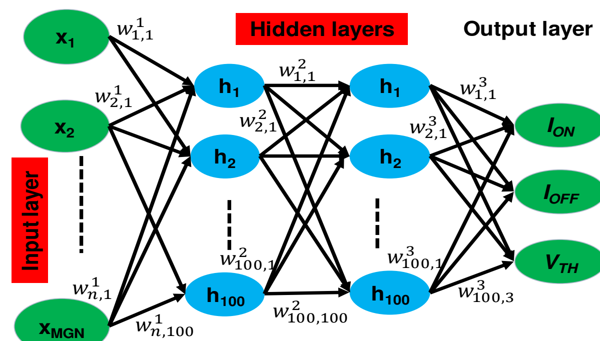

FIGURE 4°. An illustration of the basic architecture of ANN by considering MGN = 16, 80 and 320. It consists of 4 layers including one input layer, 2 hidden layers and one output layer.

To examine the effect of the variability of the WKF on emerging MOSFETs, the cuboid grain method and the averaged WKF method, etc., [52] have been proposed and proven remarkable. In [17], the cuboid grain method was adopted to study the effect of the metal-grain number (MGN) and metal-grain location (MGL) with the low and high work functions on the variability of the V_{TH}. However, with the increase of MGN, the requirement for a large number of samples of WK also increases. Therefore, the highly computational cost is obligatory for accurate prediction. Each 3D device simulation with a given grain pattern takes 1-2 hours and more than 3,000 3D device simulations should be conducted to explore the WKF-induced variability. To overcome these computational resource intensive issues, we advance our recent work on the studies of the WKF by applying three different supervised ML models to study characteristic fluctuation induced by the WKFs.

To understand the concatenation between two emerging technologies, the overview of the integration of the 3D device simulation with DL technology is shown in Fig. 1. Notably, in Fig. 1(a), the nominal GAA Si NW MOSFET and its electrical characteristics for the sub-5-nm technology node is shown. Similarly, in Fig. 1(b), four major variability sources are illustrated such as RDF, PVE, RITs and WKF. By using any of these variability sources, many fluctuated devices can be generated and crucial electrical characteristics can be extracted from a bundle of I_D-V_G curves. Prominently, the WKF is considered one of the most significant variability sources in semiconductor devices. Besides, the variability source and the extracted parameters are taken as input into the DL algorithm, as shown in Fig. 1(c) and it also exhibits the possible applications that arise due to the conjunction of device simulation and DL technology.

In this work, three different DL algorithms including ANN, convolutional neural network (CNN) and long short term memory (LSTM) are proposed to predict characteristic fluctuation of V_{TH}, I_{ON} and I_{OFF} affected by various MGNs, i.e., MGN = 16, 80 and 320. They are induced by low and high work functions and between the source (S) and the drain (D) at different locations. The results of this work conclude that the device simulation based DL algorithms can largely accelerate the prediction process of device simulation data by minimizing the computational cost and can easily enhance the prediction accuracy.

This paper is structured as follows. Section II presents the statistical device simulation and deep learning methodology. Section III demonstrates the comparison of different DL models. Section IV reports the different techniques to evaluate the DL algorithms. Section V illustrates the results and the detail discussion for different MGNs. Section VI presents the emerging applications of DL with WKF for GAA Si NW MOSFET and finally in the section VII, the conclusion is drawn and the direction for future research is given. In Appendices, the mathematical formulations of the applied DL models are mentioned.

II. STATISTICAL DEVICE SIMULATION AND DEEP LEARNING METHODOLOGY

The WK fluctuated devices are simulated by 3D device simulation. As shown in Fig. 2(a), a transmission electron microscope (TEM) of TiN gate from a realistic fabrication is irregular shape generally which is difficult to formulated ideally in device simulation. Therefore, the cuboid grain algorithm is implemented to position the MG with an acute angle. Moreover, the simulated structure consists of a 10-nm diameter (*d*) of horizontal cylindrical Si channel with a 10-nm gate stack of HfO_2 having a 0.6-nm effective oxide thickness (*EOT*) with TiN gate having WK of 4.552 eV, as shown in Fig. 2(a). Moreover, simulating through higher threshold voltage for low-power devices, TiN gate yields 0.2 eV offset between the low and high WKs. Therefore, low and high WKs are defined as TiN<111> having 0.4 probability of occurrence on MG with WK = 4.432 eV and TiN<200> having 0.6 probability of occurrence on MG with WK = 4.632 eV, respectively. Fig. 2(b) illustrates that three different sizes of MGs deposited on the TiN gate. The number of MGs is

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

proportional to the grain size. The number of MGs are 16, 80 and 320 for grain size = 5 nm x 4 nm, $(2\ nm)^2$ and $(1\ nm)^2$, respectively. For each MG, random patterns are generated by the Monte Carlo (MC) method and the simulated I_D-V_G curves are obtained by using 3,000 to 10,000 fluctuated devices. From these I_D-V_G curves, the device parameters are extracted. These WK fluctuated patterns for different MGNs with their corresponding extracted parameters are preprocessed and fed into different DL algorithms, as shown in Fig. 2(c). Three different DL algorithms: ANN, CNN and LSTM, are implemented. The brief demonstration of simulated device parameters is listed in Table I. In this section, we introduce the dataset and its preprocessing. The basic knowledge of ANN, CNN and LSTM is stated to master the relation between the sequence of high and low WKs and the effect of the random WK on the magnitudes of variability of the V_{TH}, I_{ON} and I_{OFF}.

A. DATASET AND PREPROCESSING

The dataset is consisting of random patterns of high and low WKs that are being mapped into 0 and 1 values, respectively. Consequently, the studied DL algorithms can be fed using discrete input data, where any divergent value is eliminated; and, V_{TH} = 10 V is set. The divergent data is insignificant and is considered to be noise. Then, the normalization of all features is performed to scale down the difference between the minimum and the maximum values in the dataset. The most common method to normalize the dataset is MinMaxScaler from Scikit-Learn Python's Library [53]. The final output from all these preprocessing steps is taken as input to the DL models. The flow chart of data collection, noise reduction, data preprocessing, splitting the dataset into training and testing set and feeding the DL model with input and target dataset, all are illustrated in Fig. 3.

B. ARTIFICIAL NEURAL NETWORK

ANN is the most common and widely used algorithm in science and engineering [54-58] which has been of great interest due to the multilayered network having the capability to extract features [59, 60]. The dataset enters through the input layer, passes by the hidden layer for the extraction of the useful features. Then, the output from the previous hidden layer is considered as the input to the next hidden layer (the mathematical insight of ANN is further explained in Appendix) and afterward, the output is predicted from the output layer. The activation function in the hidden layer performs the non-linear complexity. Generally, there are many different architectures as well as various optimization algorithms of the ANN model; however, we merely focus on one neural-network architecture, as shown in Fig. 4. The implementation of the ANN model is consisting of four layers: one input layer, two hidden layers, and one output layer. The input layer consists a number of neurons equal to the number of features utilize in one batch. Similarly, the number of neurons in the hidden layers are adjustable in the range of 32~100, depending upon many factors such as the length of

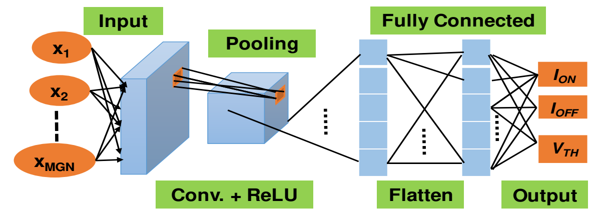

FIGURE 5°. An illustration of the basic architecture of CNN for MGN = 16, 80 and 320, consisting of input layer, one convolutional layer with ReLU activation layer, one pooling layer, two fully connected layers and one output layer.

TABLE I.
LIST OF DEVICE SIMULATION PARAMETERS FOR GAA SI NW MOSFETS

Parameters	Value
L_G (nm)	10
EOT (nm)	0.6
d (nm)	10
L_s *and* L_d (nm)	5
S/D Depth (nm)	14
Channel doping (cm^{-3})	$5x10^{17}$
S/D Doping (cm^{-3})	$1x10^{20}$
S/D Extension Doping (cm^{-3})	$5x10^{18}$
Nominal EWK	4.552
V_{TH} (mV)	272.0
I_{ON} (A)	$6.71x10^{-6}$
I_{OFF} (A)	$6.65x10^{-12}$

the input sequence, the nature of the output (digit or numerical value), etc. Here, the type of target data is numeric continuous values. As shown in Fig. 4, consider the regression ANN model for 16-dimensional input sequence, i.e., $\{x_1, x_2, \ldots, x_{16}\}$, the weight corresponding to the first layer, the second layer and the third layer is represented as a matrix W_{16x100}, $W_{100x100}$ and W_{100x1}, respectively. Moreover, rectified linear unit (*ReLU*) activation function is explored because it converges faster as compared to other activation functions, such as sigmoid, leaky *ReLU* and hyperbolic tangent [53]. The mathematical notation of *ReLU* activation function is given as:

$$ReLU(x) = max(0, x), \tag{1}$$

where x is considered as the input to the neuron and *ReLU* is the activation function depend on the maximum value of 0 and x. In the feedforward direction, the output from any arbitrary neuron S_r is given as:

$$S_{r=}ReLU(\textstyle\sum_{i=1} w_i x_i + b_i), \tag{2}$$

where x, b and w represent the input, the corresponding bias and the weight of a given neuron, respectively. Bias and weight are considered as the hyperparameters which are tunable variables and through backpropagation, these hyperparameters can be optimized. The output obtains from the last layer is contaminated due to the randomness of weights and biases in each layer. To minimize the error between the target and the predicted values, the optimization function is utilized. The following loss function is taken into account to

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079981, IEEE Access

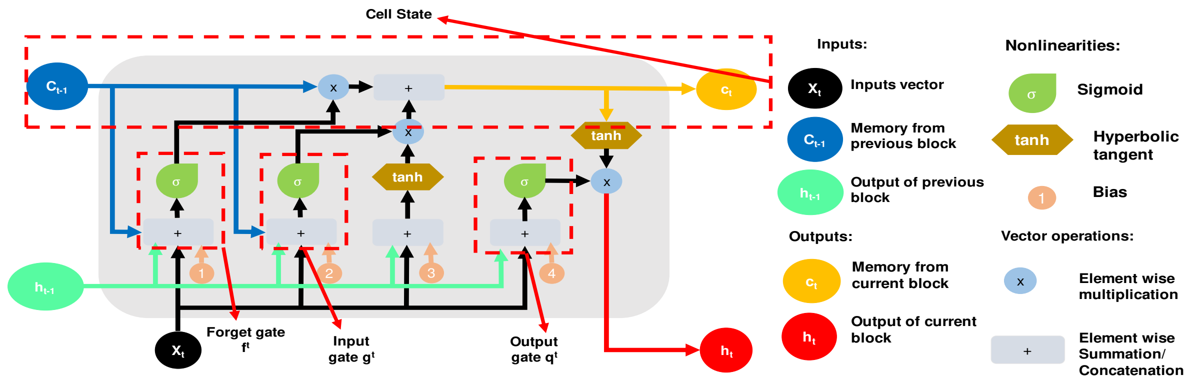

Figure 6º An illustration of the internal structure of LSTM cell consisting of input gate, forget gate, cell state and output gate. Sigmoid activation function is used in input, forget and output gate. In addition, tangent hyperbolic activation function is implemented in cell state.

calculate the error between the predicted and the target values.

$$Loss\ function = \frac{1}{N}\sum_{i=1}^{N}(y_{target} - y_{pred})^2, \quad (3)$$

where y_{target} and y_{pred} represent the exact and the predicted values obtained from the trained ANN model; N represents the total number of samples in the train dataset. Notably, the ANN has some imperfections while dealing with optimization algorithms. Because it takes a longer time to converge and to find the global minima [60]. Moreover, the number of features are 16, 80 and 320 which is infeasible for ANN to deal with the longer input sequences efficiently without facing the curse of dimensionality.

C. CONVOLUTIONAL NEURAL NETWORK

Primarily, the CNN model is considered a useful algorithm in various applications such as object detection, computer vision and pattern recognition due to its ability to extract features from the input data very efficiently [61-64]. Fig. 5 shows the basic architecture of CNN as a regression model and the mathematical perception is given in Appendix. Consider the input for MGN = 16 such that $\{x_1, x_2, \ldots, x_{16}\}$, the input feed into CNN model is converted into a 2D matrix so that it can be manipulated by using a convolutional layer which is based on the processing of 2D input matrix and kernel. The kernel allows us to extract the useful information from the input matrix; for example, some specific kernels can extract the information around the boundary of the input matrix [65-69]. Traditionally, the stack of convolutional layers can be increased as many times as to extract a large number of input features. The mathematical notation of 2D convolution $S(i,j)$ between the input matrix and kernel is expressed as:

$$S(i,j) = (I \times K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n), \quad (4)$$

where K and I are the kernel and filter that represent the square matrix of any size and the 2D input data consisting of a matrix with m-by-n dimensions, respectively. Similarly, i and j represent the number of rows and number of columns of the kernel. Notably, the number of training samples propagates into the DL model in one forward and backward pass is known as batch size. The output of the convolutional layer then undergoes through pooling layer to reduce the input size, therefore, this layer can prevent the various DL dilemmas including (i) the curse of dimensionality, (ii) overfitting during the DL model training and (iii) cost reduction computationally. Lastly, a fully-connected layer is implemented (also known as the Dense layer) that works similar to the feedforward ANN (see Appendix). Besides, the number of epochs handle the number of times the backpropagations process is performed in DL model. In short, the CNN model focuses on each small feature of the input data by sharing parameters using the same kernel (may use different kernels for same input data) and develop a systematic process to predict the output by considering those explored features. Similar to ANN, CNN model is applied to three continuous numerical outputs, i.e., I_{ON}, I_{OFF} and V_{TH}.

D. LONG SHORT TERM MEMORY

It is known that LSTM is popular for forecasting data that depends on the time intervals. Therefore, it is implemented for predicting recurrent input. e.g., time-series data, natural language processing and weather forecasting, etc. LSTM is a special type of neural-network that gains attention due to the memory blocks. These memory blocks have self-connection which memorizes the flow of information [70-74]. Consider LSTM as a regression model and the input given to a LSTM cell is for MGN = 16, i.e., $\{x_1, x_2, \ldots, x_{16}\}$. Before feeding into the LSTM model, the input array is reshaped into batch size, the number of samples passes in one-time step and the number of input features, e.g., (20, 1, 16). The number of features varies with different MGNs. The LSTM cell contains three gates: the input, forget, and output gates, respectively. The architecture of the LSTM cell is shown in Fig. 6 and its comprehensive pseudo-code is listed

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079981, IEEE Access

in Appendix. The mathematical notation of forget gate $f_i^{(t)}$ by considering the t time step of the ith cell, is given as:

$$f_i^t = \sigma(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}), \quad (5)$$

where $x^{(t)}$ is the input vector at time t and $h^{(t)}$ is the hidden vector at time t. b^f, U^f, and W^f are the bias, input weight and the recurrent weight for the forget gate, respectively. The LSTM cell internal state c is updated with the self-loop weight and by considering hyperbolic tangent (*tanh*) as an activation function, the mathematical expression of the cell state is given by:

$$c_i^t = f_i^t c_i^{(t-1)} tanh(b_i^c + \sum_j U_{i,j}^c x_j^{(t)} + \sum_j W_{i,j}^c h_j^{(t-1)}), \quad (6)$$

where b, U and W, respectively, represent the bias, input weight and the recurrent weight of the LSTM cell state. The input gate unit g is computed similarly to the forget gate. The mathematical notation of the input gate g is:

$$g_i^t = \sigma(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}), \quad (7)$$

where b, U and W represent the bias, the input weight and the recurrent weight of the input gate, respectively. Similarly, x and h are the input matrix and the hidden unit, respectively. The output h of the LSTM is obtained via the output gate q:

$$h_i^t = tanh(c_i^t)\, q_i^t \quad (8)$$

and

$$q_i^t = \sigma(b_i^q + \sum_j U_{i,j}^q x_j^{(t)} + \sum_j W_{i,j}^q h_j^{(t-1)}, \quad (9)$$

where c and q represent the cell state and the output gate, respectively. Similarly, b, U and W represent the bias, the input weight and the recurrent weight of the output gate, respectively. These weights and biases are considered as the hyperparameters of the LSTM cell and these hyperparameters can be updated by taking the gradient with respect to the weights and the biases of the input gate, the output gate and the forget gate. The main advantage of the LSTM model is that it can regulate the gradient by overwhelming the most common crisis of DL model, i.e., the vanishing gradient and the exploding gradient. The output obtains from each LSTM cell consists of two arrays, i.e., the cell state c and hidden layer output h. It is not mandatory to forward both arrays into the next LSTM cell. Notably, to limit the size of the output array (three variables), only hidden layer output is considered as the final output from each LSTM cell and cell state is omitted.

III. COMPARISON OF MODELS

Each algorithm has some limitations such as ANN requires a huge amount of dataset as well as its ability to compute the gradient is also computationally expensive. Typically, for the ANN model, there is no criterion for the minimum number of samples. However, the smaller number of training samples as compared to the larger input features is not an adequate approach. Moreover, the LSTM model is more suitable for predicting the data having a larger input sequence and it has effective regularization over the vanishing gradient and the exploding gradient. Nonetheless, the LSTM model requires a comparatively larger amount of training samples. The mathematical notation for the minimum number of training samples required for the LSTM model is given as:

$$4(n \times m + n^2), \quad (10)$$

where n and m correspond to the size of the input features and the output features, respectively. In the cases of MGN = 16, 80 and 320, the minimum number of training samples required for a well-trained LSTM model is 1,088, 25,920 and 410,880, respectively. Therefore, for MGN = 80 and 320 above-mentioned number of training samples are implausible to achieve through device simulation of GAA Si NW MOSFETs. As we know, the CNN model is prominently used in the field of computer vision and object detection. Besides, it has a powerful endowment to deal with more complex problems. Due to this reason, the CNN model is more suitable for dealing with a large number of input features. Moreover, the minimum number of training samples is dependent on the depth of the CNN model and its mathematical formula is shown below.

$$(N_{in} + 1) \times N_h + (N_h + 1) \times N_{out} < N_s, \quad (11)$$

where N_{in}, N_h and N_{out} are the number of input features, number of hidden units and number of output features, respectively. N_s represents the minimum number of training samples. Therefore, by comparing the model architecture among ANN, CNN and LSTM, the CNN model is considered to be more efficient for the dataset obtain from the device simulation of GAA Si NW MOSFETs.

The main challenge in developing an optimal DL algorithm is to set the hyperparameters of the model to minimize the aforementioned loss function and acquire the convergence rapidly. Majorly, three experiments, for MGN = 16, 80 and 320, are carried out for each DL model (ANN, CNN and LSTM). In the conducted experiments, three datasets are utilized to predict the three crucial electrical characteristics (i.e., V_{TH}, I_{ON} and I_{OFF}), the hyperparameter configurations are illustrated in Table II. It is proven that DL algorithms have been performing well as compared to ML algorithms due to the dense number of layers and their ability to deal with the curse of dimensionality.

IV. EVALUATION OF DEEP LEARNING MODELS

While dealing with GAA Si NW MOSFETs, there are two major challenges: (i) highly computational cost and (ii) precision of electrical characteristics. The DL models are evaluated by using the root mean squared error (RMSE) value as well as the error rate. Generally, the RMSE values are calculated for the regression and the numerical problems. The RMSE value is the difference between the true and predicted values from the DL models. The mathematical expressions of the RMSE values for the train and the test dataset are given by the following two equations, respectively:

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079981, IEEE Access

TABLE II
LIST OF THE HYPERPARAMETERS CONFIGURATION FOR ANN, CNN AND LSTM MODEL USING MGN = 16, 80 AND 320

MGN	DL Model	Hidden (H), Activation (Act), Dense (D), Batch Size (BS), Epochs (E), Filter (F), Kernel (K)
16	*LSTM*	H = 64, Act= ReLU, D = No, BS = 20, E = 100
	CNN	Act= ReLU, D = 32, BS = 20, E = 100, F = 64, K = 2
	ANN	H = 20, Act= ReLU, D = 10, D = 1 BS = 20, E = 100
80	*LSTM*	H = 64, Act= ReLU, D = 1, BS = 20, E = 100
	CNN	F = 32, Act= ReLU, D = 1, BS = 20, E = 100, K = 3
	ANN	H = 20, Act= ReLU, D = 10,D = 1, BS = 1, E = 100
320	*LSTM*	H = 64, Act= ReLU, D = 1, BS = 20, E = 100
	CNN	H = 64, Act= ReLU, D = 1, BS = 20, E = 100, K = 3
	ANN	H = 64, Act= ReLU, D = 10, D = 1, BS = 20, E = 100

TABLE III.
THE CALCULATED RMSE VALUES FOR TRAIN AND TEST OF THE ANN, CNN AND LSTM MODELS USING MGN = 16

DL Model	RMSE Value for Train	RMSE Value for Test
LSTM for V_{TH}	0.00632	0.03694
CNN for V_{TH}	0.00474	0.0350
ANN for V_{TH}	0.00638	0.04058
LSTM for I_{ON}	0.00441	0.02291
CNN for I_{ON}	0.00340	0.01417
ANN for I_{ON}	0.00550	0.02382
LSTM for I_{OFF}	0.00717	0.06562
CNN for I_{OFF}	0.00689	0.06472
ANN for I_{OFF}	0.00741	0.07194

$$RMSE_{train} = \sqrt{\frac{\sum_{i=1}^{N_{train}}(y_{train}-y_{train_pred})^2}{N_{train}}} \quad (12)$$

and

$$RMSE_{test} = \sqrt{\frac{\sum_{i=1}^{N_{test}}(y_{test}-y_{test_pred})^2}{N_{test}}}, \quad (13)$$

where N_{train} and N_{test} is the total number of the train and the test data samples, respectively. Similarly, y_{train} and y_{test} represent the exact values from the train and the test data samples, respectively. Likewise, y_{train_pred} and y_{test_pred} depict the predicted values by using its corresponding train and the test dataset, respectively.

Similarly, the performance of the explored DL models can be determined by calculating the error rate in terms of the variance of the actual dataset and the predicted output. The mathematical formula of error rate is shown below:

$$Error\ Rate\ \% = \frac{\sigma_{test}-\sigma_{pred}}{\sigma_{test}} \times 100, \quad (14)$$

where σ_{test} and σ_{pred} represent the standard deviation of the test dataset and the predicted values collected from the trained DL model, respectively.

V. RESULTS AND DISCUSSION

In our prior work [17], it has been observed that the high-κ/metal-gate (HKMG) technology with GAA Si NW MOSFETs is considered to be a more effective technology and the magnitude of electrical characteristics induced by the WKF depend on two factors: the random number of MGs and the random position of MGs. The most crucial electrical characteristics induced by low and high WKs is the variability of the threshold voltage (σV_{TH}) defined by:

$$\sigma V_{TH} = \sqrt{\frac{\sum_{i=1}^{n} SDM_i}{n-1}} \quad (15)$$

and

$$SDM_i = (V_{TH_i} - V_{TH_mean})^2, \quad (16)$$

where *i* and *n* are the index number of the fluctuated devices and the total number of the fluctuated device, respectively; *SDM* is the square of deviation from the mean value and V_{TH_mean} is the mean of V_{TH}. From (15), it is clarified that σV_{TH} is directly proportional to the sum of SDM and inversely proportional to the total number of fluctuation devices.

The integration of HKMG technology with the emerging DL methodology demonstrates the accurate prediction of electrical characteristics of GAA Si NW MOSFETs and to determine the relationship of induced electrical characteristics with the randomly generated fluctuated devices. The prediction of continuous numeric values, i.e., V_{TH}, I_{ON} and I_{OFF}, appears in the domain of regression problems. The output from the explored DL models is considered as the predicted value estimated through the target values obtain from HKMG WK's effective electrical characteristics including V_{TH}, I_{ON} and I_{OFF}. While developing the DL models, the various fluctuated devices are considered as the number of input data. For example, for MGN = 16, 80 and 320, the total number of fluctuated devices is 3000, 5000 and 10000, respectively, which is equivalent to the total number of samples for MGN = 16, 80 and 320.

The term 'train' in the DL model is referred to as feeding the model to estimate its hidden hyperparameters and optimize them using various algorithms, such as the stochastic gradient descent and the adaptive moment estimation, etc. Whereas, the term 'test' refers to predicting the trained DL model with the new dataset. Usually, in the DL algorithms, the split between train and test datasets is 90% and 10%, respectively. Initially, the DL models are trained after randomly shuffling the dataset, so that, in each epoch, every data point enlightens the trainable parameters of the adapted DL model; otherwise the RMSE value will be too high. Moreover, it can be observed in Table III that the RMSE values obtain through all explored DL models (ANN, CNN and LSTM) during training and testing. It is observed that the RMSE value during the training of all DL models is outperformed, whereas the reduction of RMSE value for testing dataset obtained through CNN among all the DL models is persistent. Moreover, due to sizeable variation in I_{OFF} dataset, the RMSE value for both train and test datasets is comparatively higher than V_{TH} and I_{ON}. Likewise, from Tables IV and V, it can be seen that for MGN = 80 and 320, the RMSE values for the test dataset are decreased by the CNN

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

TABLE IV
THE CALCULATED RMSE VALUES FOR TRAIN AND TEST OF THE ANN, CNN AND LSTM MODELS USING MGN = 80

DL Model	RMSE Value for Train	RMSE Value for Test
LSTM for V_{TH}	0.00934	0.03297
CNN for V_{TH}	0.00529	0.0298
ANN for V_{TH}	0.01318	0.05087
LSTM for I_{ON}	0.00219	0.01207
CNN for I_{ON}	0.00522	0.01060
ANN for I_{ON}	0.00798	0.01694
LSTM for I_{OFF}	0.01355	0.05285
CNN for I_{OFF}	0.00664	0.03518
ANN for I_{OFF}	0.01569	0.03735

TABLE V
LIST OF COMPUTED RMSE VALUES FOR TRAIN AND TEST OF THE ANN, CNN AND LSTM MODELS USING MGN = 320

DL Model	RMSE Value for Train	RMSE Value for Test
LSTM for V_{TH}	0.09421	0.08707
CNN for V_{TH}	0.00733	0.0513
ANN for V_{TH}	0.0840	0.0535
LSTM for I_{ON}	0.00903	0.0717
CNN for I_{ON}	0.00579	0.05612
ANN for I_{ON}	0.01410	0.07475
LSTM for I_{OFF}	0.01249	0.06739
CNN for I_{OFF}	0.01475	0.05764
ANN for I_{OFF}	0.01003	0.06400

TABLE VI
LIST OF CALCULATED ERROR RATE FOR TEST DATASET OF THE ANN, CNN AND LSTM MODELS USING MGN 16, 80 AND 320

MGN	Electrical Characteristics	ANN	CNN	LSTM
16	V_{TH}	0.9%	0.5%	0.8%
	I_{ON}	1.9%	0.95%	1.3%
	I_{OFF}	1.7%	1.5%	1.6%
80	V_{TH}	1.3%	0.5%	1.5%
	I_{ON}	0.9%	0.8%	1.3%
	I_{OFF}	1.1%	0.8%	1.3%
320	V_{TH}	1.6%	1.1%	1.8%
	I_{ON}	1.5%	1.3%	1.9%
	I_{OFF}	1.8%	1.3%	1.7%

model. The reason behind the reduction of RMSE value via the CNN model is that it can handle properly a large number of input features, i.e., 80 and 320. The DL model having a large number of input features demand two major properties: (i) the number of training data points should be at least four times larger than the number of input features and (ii) the optimized deep model architecture having a large number of parameters. In the case of LSTM and ANN, it is architecturally impossible to fulfill the above two criteria for such long input sequences, i.e., 80 and 320. Nonetheless, in GAA Si NW MOSFETs with HKMG technology, it takes highly computational cost to collect a large number of simulated samples, which is a critical issue in the field of GAA Si devices.

In the DL algorithms, it is a commonly practice to evaluate the regression model using the RMSE value. Even the smallest difference between the attained RMSE values are considered to be significantly important. While pondering towards the RMSE values for train data, all the DL models outperform. The prevailing goal is to accomplish a nominal RMSE value for the test dataset. In Tables III, IV and V, the minimum RMSE value, for both train and test datasets, is obtained by the CNN model due to its capacity to deal with higher dimensional data. Specifically, by comparing Table V with Tables III and IV, the increment of the RMSE value for the test dataset obtained through the ANN, CNN and LSTM models using the MGN = 320 are due to various reasons such as (i) the larger input length, i.e., 320, (ii) the curse of dimensionality encountered by the DL models, (iii) the exploding and the vanishing gradient during the backpropagation of the explored DL models. From our earlier work [13], it is clear that electrical characteristics are induced by the WKF, which majorly depends on the random position of the MG. By conquering the issue of the random position, the CNN model performs better as compared to the ANN and the LSTM model. The comparison among three DL models in terms of the RMSE values depicts that for the MGN =16, all the DL models outperform. However, the CNN model surmounts the result due to its sparse property. The training and the testing of the ANN, the CNN and the LSTM model through the electrical characteristics, i.e., V_{TH}, induced by the WK having the MGN = 16 is illustrated in Figs. 7(a) and (b), respectively; whereas, for the MGN = 16, the training and the testing of all DL models using I_{ON} is illustrated in Figs. 7(c) and (d), respectively and in Figs. 7(e) and (f), the training and testing of all DL models using I_{OFF} is shown, respectively. From these given plots, it is difficult to distinguish the most significant DL model performance. So, the error rate is calculated. As listed in Table VI, for the MGN = 16, the test error rate for the ANN, the CNN and the LSTM models, by considering V_{TH} is 0.9%, 0.5% and 0.8%, respectively. Similarly, the error rate for the ANN, the CNN and the LSTM model by using I_{ON} is 1.9%, ANN, the CNN and the LSTM models using the explored electrical characteristics, i.e., V_{TH}, I_{ON} and I_{OFF}. Likewise, Figs. 9(a)-(f) illustrate the training and the testing for MGN = 320 by implementing the all explored DL models using V_{TH}, I_{ON} and I_{OFF}. Apart from the MGN = 16 and 80, the MGN = 320 is considered as a different case due to its excessive feature size, i.e., 320. As it has been already discussed, the larger input length may affect the training and testing of the DL models because (i) it may produce the curse of dimensionality, (ii) it requires more training dataset which is an inevitable problem in the field of device manufacturing and the simulation, and (iii) it can explode or vanish the gradient of the DL models during their backpropagation. Along with these issues, our explored DL models predict better than the anticipation.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079981, IEEE Access

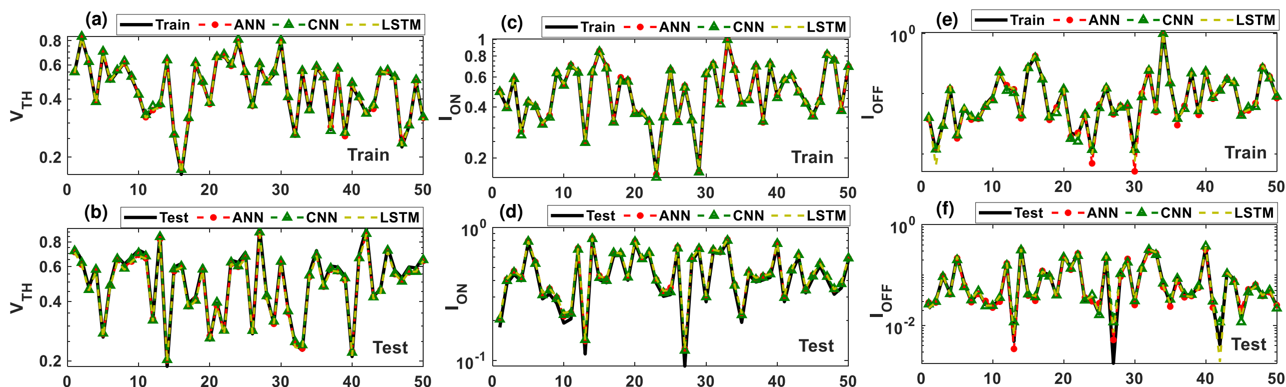

Figure 7° Comparison among the ANN, CNN and LSTM models by predicting electrical characteristics induced by WKF using MGN = 16 having grain size = 5 nm x 4 nm on GAA Si NW MOSFETs. (a) and (b) illustrate the train and test of the explored DL models, respectively. All these adapted DL algorithms predict the V_{TH} induced by WKF. (c) and (d) depict the train and test procedure of the DL models, respectively. These adapted DL algorithms predict the I_{ON} induced by WKF. (e) and (f) represent the train and test of all the DL models, respectively. These adapted DL algorithms predict the I_{OFF} induced by WKF.

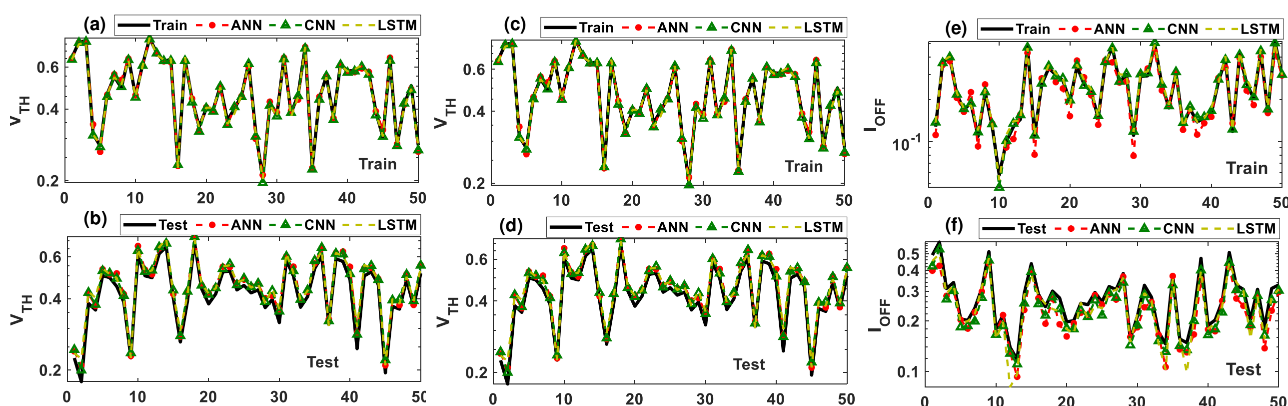

Figure 8° Comparison among the ANN, CNN and LSTM models by predicting electrical characteristics induced by WKF using MGN = 80 having grain size = (2 nm)² on GAA Si NW MOSFETs. (a) and (b) illustrate the train and test of the explored DL models, respectively. These adapted DL algorithms predict the V_{TH} induced by WKF. (c) and (d) depict the train and test procedure of the DL models, respectively. These adapted DL algorithms predict the I_{ON} induced by WKF. (e) and (f) represent the train and test of the DL models, respectively. These adapted DL algorithms predict the I_{OFF} induced by WKF.

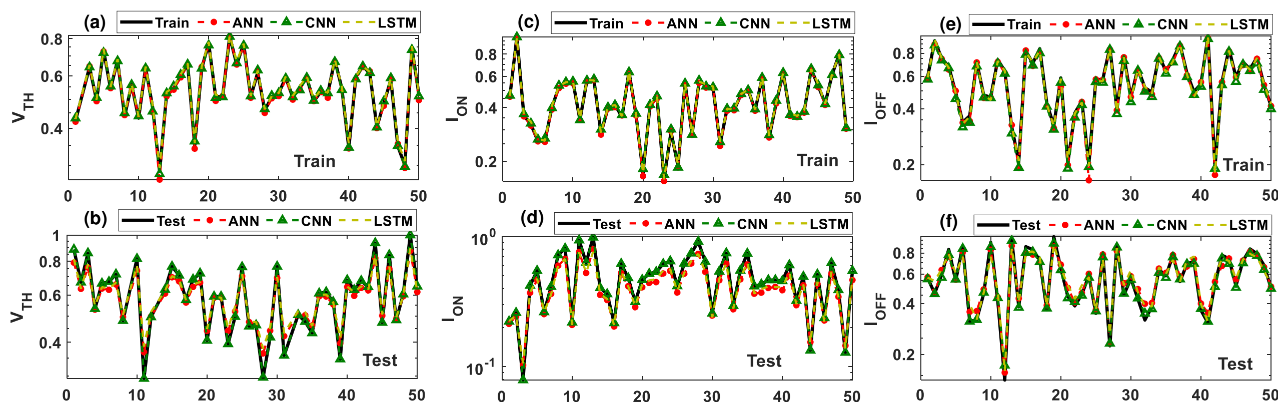

Figure 9° Comparison among the ANN, CNN and LSTM models by predicting electrical characteristics induced by WKF using MGN = 320 having grain size = (1 nm)² on GAA Si NW MOSFET. (a) and (b) illustrate the train and test of the explored DL models, respectively. These adapted DL algorithms predict the V_{TH} induced by WKF. (c) and (d) depict the train and test procedure of the DL models, respectively. These adapted DL algorithms predict the I_{ON} induced by WKF. (e) and (f) represent the train and test of the DL models, respectively. These adapted DL algorithms predict the I_{OFF} induced by WKF.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

```
Pseudo Code: Artificial Neural Network
1.   Procedure train
2.   X ← Training dataset of size mxn
3.   y ← Labels for training data X
4.   w ← Weights for respective layers
5.   L ← The number of layers in model
6.   E ← The error for all L
7.   for i = 1 to m
8.      a^L ← forward(x^(i), w)
9.      E^L ← a^L – y^(i)
10.     D ← min (dE/dw)
11.     w^(i) ← w^(i-1) + η*D
12.  end for
```

FIGURE 10°. An illustration of the pseudo-code of ANN focuses more on training procedure because the testing procedure is straightforward.

In the DL algorithms, the curse of dimensionality can be overcome by implementing the different strategies for feature extraction and feature selection. However, each input feature holds equal significance as other features. Thus, it is worthless to use different algorithms to reduce the size of input features for the explored DL models. It is important for us to discuss more about the effectiveness of the explored DL models for predicting the V_{TH}, I_{ON} and I_{OFF}. By analyzing the effect of different MGNs (i.e., 16, 80 and 320), it is demonstrated that by increasing MGN, the RMSE value increases which depicts that it greatly affects the DL model performance. For the MGN = 16 and 80, all explored DL models such as ANN, CNN and LSTM, outperform but in the case of MGN = 320, even its noticeable concern with a long sequence of patterns, the performance of all explored DL models is acceptable.

In short, the well-trained DL models are generic, i.e., a single model architecture can be applied to different MGNs, if the dataset for any value of MGN is exhibiting similar variability characteristics (for example, the ratio of I_{ON}/I_{OFF} and the V_{TH} lie in the range of MGN with which DL model has been already trained), then the same trained DL model is enough for any MGN value. Otherwise, to obtain the converged and optimized DL model for prediction, it is necessary to repeat the training process every time with datasets having different MGNs.

VI. POTENTIAL APPLICATIONS OF DEEP LEARNING MODELS WITH WKF FOR GAA Si NW MOSFET

Nowadays, in the semiconductor industry, due to the excellent electrical characteristics of GAA Si NW MOSFETs with HKMG technology, various innovations have been launching simultaneously. In these innovations, various factors are affecting nodes below 10 nm. However, both MGL and MGN are the two most important factors that are influenced by WKF. Moreover, due to the laborious process of generating different grain sizes having different MGN and MGL, device simulation is considered the most powerful tool in semiconductor manufacturing industries. For 40 years, researchers have been investing efforts to accomplish a simple and accessible method for model designing. Technology computer-aided design (TCAD) is an authorized computer simulation tool for semiconductor manufacturing technologies and devices. Despite TCAD simulation tool success, device simulation has some limitations while dealing with WKF for GAA Si NW MOSFETs, i.e., high computational cost and the optimized solution, etc.

There are various methodologies to examine the effect of different MGNs and MGLs on the variability of GAA Si NW MOSFET. Analytical model, averaged WKF, cuboid grain method and Voronoi method [75], all are popular technologies. Therefore, it is necessary to explore the most optimized and converge methodology for MG to study the effect of the WKF. Furthermore, studying the relationship between various sizes of MG and electrical characteristics induced by WKF has significant importance in nano-device technology.

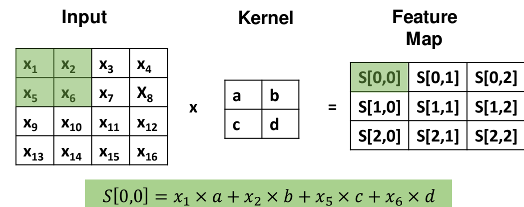

FIGURE 11°. An illustration of the convolutional layer by multiplying the input matrix with appropriate kernel and the feature map that represents the output of the convolutional layer.

TABLE VII
THE DIFFERENCE BETWEEN VARIOUS ACTIVATION FUNCTIONS IN TERMS OF THEIR INPUT AND OUTPUT RANGE AND DOMAIN

	Sigmoid	*Tanh*	*ReLU*
Range	0 and 1	-1 and 1	Max(0,x)
Outcome	A small change in input would result large change in output.	Output is centered around zero.	Computationally inexpensive as compared to sigmoid and tanh.

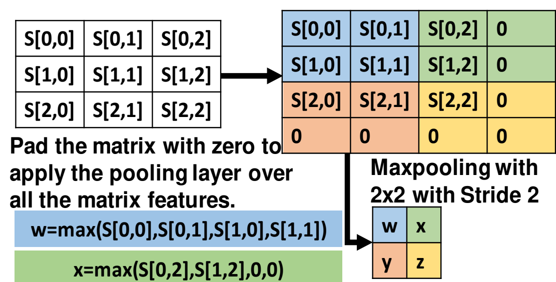

FIGURE 12°. An illustration of the max-pooling layer by padding the convolutional layer's output with zeros. Padding can be utilized at any layer to get the desired output dimensions. Stride represents the step size which moves the filter at a time.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

```
Pseudo Code: Convolutional Neural Network
1.  Procedure train
2.  X ← Train dataset of size mxn
3.  y ← Labels for training data X
4.  w ← Weights for respective layers
5.  Call Convolutional_layer
6.  Call Pooling_layer
7.  Call Fully_connected_layer
8.  end
9.  Define Convolutional_layer
10.    for i = 1 to m
11.       for j = 1 to n
12.          temp = 0
13.          for ii=1 to K
14.             for jj = 1 to K
15.                temp = temp + K[ii][jj] * X[i-ii,i-jj]
16.             end for
17.          end for
18.       end for
19.    end for
20. Define Pooling_layer
21. for i = 1 to m
22.    for j = 1 to m
23.       matrix= max(temp[i,j],... temp[m,m])
24.    end for
25. end for
26. Define Fully_connected_layer
27. for i = 1 to m
28.    temp = 0
29.    for j = 1 to n
30.       temp = temp + W[i][j] x X[j]
31.    end for
32. y[i] = temp
33. end for
```

FIGURE 13°. An illustration of the pseudo-code of CNN including three main layers, i.e., the convolutional layer, the pooling layer and the fully-connected layer.

To study the challenges that appear in WKF for GAA Si NW MOSFETs, different DL algorithms conquer all the limitations. Moreover, the integration of DL with WKF for GAA Si NW MOSFETs imply various advantages, such as (1) DL can estimate the model specification induced by WKF for GAA Si NW MOSFETs very accurately which can accelerate the device simulation with less computational cost; (2) DL can extract crucial electrical characteristics induced by WKF for GAA Si NW MOSFETs that can be extended to model the complicated device; and (3) by studying the effect of different MGNs and MGLs on the variability of GAA Si NW MOSFETs through the DL algorithm, the device simulation can be conducted easily. Based on these points of view, the integration of DL with physically-sound device simulation can be considered an auxiliary technique in design, simulation and optimization of emerging device technologies.

VII. CONCLUSIONS

In this paper, for the first time, three DL algorithms have been implemented, i.e., ANN, CNN and LSTM, which have shown sufficiently efficient and accurate performance. For the effect of MGN = 16, all the explored DL models outperform due to an optimal number of features. However, for the cases of MGN = 80 and 320, the CNN performs better than ANN and LSTM in terms of testing error rate and the RMSE value. The improvement in the predicted values using the CNN model attribute to the fact that the CNN model has the property of sparse interaction which can extract features using the minimum number of parameters. In the evidence of the testing error rate as summarized in Table V, it is concluded that the CNN model is a more optimal approach to estimating the electrical characteristics of GAA Si NW MOSFETs.

Furthermore, it is accomplished that by the integration of the DL algorithms with 3D device simulation, various achievements have been observed. such as the more complicated device simulation can be modeled and the device simulation process can be accelerated. Therefore, more complex data structures obtain through the device simulations, e.g., for MGN = 480, the electrical characteristics can be accurately predicted by using the well-trained DL models.

APPENDIX

Artificial Neural Network

Traditionally, a single artificial neuron having an input layer, activation function and output layer is known as the perceptron. A stack of these neurons having multiple layers is known as a multi-layer perceptron (MLP). Furthermore, the activation function is utilized to introduce non-linearity in the model. There are various types of activation functions, the difference between them is depicted in Table VII. However, there is one constraint regarding the selection of appropriate activation function, i.e., it should be differentiable so that while estimating the loss function it does not get vanished during the backpropagation. Conventionally, MLP/ANN consists of two main working strategies, i.e., (i) the forward propagation and (ii) the backward propagation. In the forward propagation, all the features are multiplied with their corresponding weights and biases and pass through the activation function after the summation of weight and bias, which is expressed as:

$$y = \sigma(w \times x + b), \tag{17}$$

where x and y represent the input and the output of single perceptron, respectively; σ represents the appropriate activation function and w and b depict the hyperparameters, i.e., weight and bias, respectively.

Moreover, in the backward propagation, the error is minimized and weights are updated by using the chain rule method as given below:

$$\frac{dE}{dw} = \frac{dE}{dA} \times \frac{dA}{dy} \times \frac{dy}{dw}, \tag{18}$$

where dE/dw represents the derivative of the error function with respect to the weight. Similarly, dE/dA, dA/dy and dy/dw represent the derivation of the error function with respect to the activation function, the derivative of the activation function with respect to the forward propagation, and the derivative of the forward propagation function with respect to the weight, respectively. Once all derivatives are calculated, then the weights are updated with respect to the corresponding

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Pseudo Code: Long Short Term Memory
1. **Procedure train**
2. **X ← Training dataset of size mxn**
3. **y ← Labels for training data X**
4. **w ← Weights for respective layers**
5. **Calculate the output from Input, output and forget gate from Eqs. (5) to (9).**
6. **Calculate the error function**
7. **Calculate derivative from output from all the gates with respect to their corresponding weights**
8. **Update the weights through back propagation**
9. **end**

FIGURE 14° An illustration of the pseudo-code of LSTM having input gate, output gate and forget gate. (For (5)-(9), see the section II statistical device simulation and deep learning methodology)

to these derivatives and the small factor known as a learning rate is given by:

$$w_{new} = w_{old} - \eta \frac{dE}{dw}, \tag{19}$$

where w_{new} and w_{old} are the updated and the previous weights, respectively. η is the learning rate that corresponds to the step size for each iteration towards the minimization of the loss function. Also, the comparatively smaller learning rate corresponds to the slow convergence with the optimal solution and vice versa. An estimated value of the optimum learning rate in a neural-network is 0.001. A list of the pseudo-code of ANN is illustrated in Fig. 10.

Convolutional Neural Network

Conventionally, CNN consists of a stack of three basic layers, i.e., the convolutional layer, the pooling layer, and the fully-connected layer. CNN also works similar to ANN in terms of the forward and backward propagation strategies. Here, CNN is explained in terms of mathematical equations and also depicted in Fig. 11. The convolutional layer corresponding to the convolution between filter/kernel and input matrix/image is given by:

$$S(i,j) = (I \times K)(i,j) = \sum_m \sum_n I(m,n) K(i-m, j-n), \tag{20}$$

where K is the kernel/filter and I is the 2D input data consisting of a matrix with i-by-j dimensions. The second step in CNN is the pooling layer which extracts the important features or eliminates the noise from the input matrix, as shown in Fig. 12. After the pooling layer, the fully-connected layer is implemented which is nothing but the traditional neural network. In the backward propagation, the same process is established such that the model's parameters are updated by minimizing the loss function. The pseudo-code of CNN is illustrated in Fig. 13.

Long Short Term Memory

LSTM is a DL model and it is an extended version of the recurrent neural network (RNN). RNN is a special type of neural network that can deal with datasets having the characteristics of time dependency, periodicity and sequence, etc. However, there are some limitations of RNN including (i) it cannot deal with long term dependencies means that it is not able to memorize and make a correlation between long sequence of data and (ii) due to the absence of long term dependencies, during the backpropagation, mostly it faces the problem of vanishing gradient and sometimes exploding gradient. Therefore, LSTM is established to overcome these problems by introducing the concept of forget gate and cell state. The pseudo-code of LSTM is illustrated in Fig. 14.

REFERENCES

[1] P. Stallhofer, "Why Are Silicon Wafers as Thick as They Are?," *Ultra-thin Chip Technology And Applications*, 2011, pp. 3-12.

[2] S. Keyvaninia, M. Muneeb, S. Stankovic, P. J. Van Veldhoven, D. Van Thourhout, and G. RoelKens, "Ultra-thin DVS-BCB adhesive bonding of II-V wafers, dies and multiple dies to a patterned silicon-on-insulator substrate," *Optical Materials Express* 3, vol. 3, 2013, pp. 35-46.

[3] C. Landesberger, G. Klink, G. Schwinn, R. Aschenbrenner, "New dicing and thining concept improves mechanical reliability of ultra thin silicon," in *International Symposium on Advanced Packaging Materials APM*, 2001, pp. 92-97.

[4] K. Rim, K. Chan, L. Shi, D. Boyd, J. Ott, N. Klymko, F.Cardone, L. Tai, S. Koester, M. Cobb, D. Canaperi, B. To, E. Ducj, I. Babich, R. Carmthers, P. Saunders, G. Walker, Y. Zhang, M. Steen, M. Leong, "Fabrication and mobility characteristics of ultra-thin strained Si directly on insulator (SSDOI) MOSFETs" in *IEEE International Electron Devices Meeting IEDM*, 2003, pp. 3.1.1-3.1.4.

[5] Z. Dong, Y. Lin, "Ultra-Thin wafer technology and applications: A review," *Journal of Material Science in Semiconductor Processing*, vol. 105, 2020, pp. 1-8.

[6] H. Cheng, T. Liu, C. Zhang, Z. Liu, Z. Yang, K. Nakazato and Z. Zhang, "Nanowire gate-all-around MOSFETs modeling: ballistic transport incorporating the source-to-drain tunneling," *Journal of Applied Physics*, 2020, pp. 074002-1-9.

[7] J. J. Gu, H. Wu, Y. Liu, A. T. Neal, R. G. Gordon, and P. D. Ye, "Size-Dependent-Transport Study of $In_{0.53}Ga_{0.47}As$ Gate-All-Around Nanowire MOSFETs: Impact of Quantum Confinement and Volume Inversion," *IEEE Electron Device Letters*, vol. 33, no. 7, 2012, pp. 967-969.

[8] K. Nayak, "CMOS Logic Device and Circuit Performance of Si Gate All Around Nanowire MOSFET," *IEEE Transactions on Electron Devices*, vol. 61, no. 9, 2014, pp. 3066-3074.

[9] P. Singh, N. Singh, J. Miao, W. Park, and D. Kwong, "Gate-All-Around Junctionless Nanowire MOSFET With Improved Low-Frequency Noise Behavior," *IEEE Electron Device Letters*, vol. 32, no. 12, 2011, pp. 1752-1754.

[10] K. H. Yeo, "Gate-All-Around (GAA) Twin Silicon Nanowire MOSFET (TSNWFET) with 15 nm Length Gate and 4 nm Radius Nanowires," in *IEEE International Electron Devices Meeting IEDM*, 2006, pp. 1-4.

[11] J. J. Gu, "III-V gate-all-around nanowire MOSFET process technology: From 3D to 4D," in *IEEE International Electron Devices Meeting IEDM*, 2012 , pp. 23.7.1-23.7.4.

[12] S. Johansson, E. Memisevic, L. Wernersson, and E. Lind, "High-Frequency Gate-All-Around Vertical InAs Nanowire MOSFETs on Si Substrates," *IEEE Electron Device Letters,* vol. 35, no. 5, 2014, pp. 518-520.

[13] Y. Li, H. Chang, C. Lai, P. Chao, and C. Chen, "Process variation effect, metal-gate work-function fluctuation and random dopant fluctuation of 10-nm gate-all-around silicon nanowire MOSFET devices," in *IEEE International Electron Devices Meeting IEDM*, 2015 , pp. 34.4.1-34.4.4.

[14] H. Nam, Y. Lee, J. Park, and C. Shin, "Study of Work-Function Variation in High-κ/Metal-Gate Gate-All-Around Nanowire

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3079981, IEEE Access

MOSFET," *IEEE Transactions on Electron Devices,* vol. 63, no. 8, 2016, pp. 3338-3341.

[15] N. Paydavosi, "BSIM—SPICE Models Enable FinFET and UTB IC Designs," *IEEE Access,* vol. 1, 2013, pp. 201-215.

[16] K. Nayak, S. Agarwal, M. Bajaj, K. V. R. M. Murali, and V. R. Rao, "Random Dopant Fluctuation Induced Variability in Undoped Channel Si Gate all Around Nanowire n-MOSFET," *IEEE Transactions on Electron Devices,* vol. 62, no. 2, 2015, pp. 685-688.

[17] W. L. Sung and Y. Li, "Effects of random number and location of the nanosized metal grains on the threshold voltage variability of silicon gate-all-around nanowire n-type metal-oxide-semiconductor field-effect transistors," *Journal of Computational Electronics,* vol. 19, 2020, pp. 1478-1484.

[18] Y. Li, C. Hwang, T. Li, and M. Han, "Process-Variation Effect, Metal-Gate Work-Function Fluctuation, and Random-Dopant Fluctuation in Emerging CMOS Technologies," *IEEE Transactions on Electron Devices,* vol. 57, no. 2, 2010, pp. 437-447.

[19] W. L. Sung and Y. Li, "DC/AC/RF Characteristic Fluctuations Induced by Various Random Discrete Dopants of Gate-All-Around Silicon Nanowire n-MOSFETs," *IEEE Transactions on Electron Devices,* vol. 65, no. 6, 2018, pp. 2638-2646.

[20] A. L. Tarca, V. J. Carey, X. W. Chen, R. Romero, and S. Drăghici, ''Machine learning and its applications to biology,'' *PLOS Computational Biology*, vol. 3, no. 6, 2007, pp. 0953-0963.

[21] W. Y. Lin, Y. H. Hu, and C. F. Tsai, "Machine learning in financial crisis prediction: A survey," *IEEE Transactions on System, Man, and Cybernetics, Part C Applications and Review,* vol. 42, no. 4, 2012, pp. 421–436.

[22] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," *SIGCOMM Workshop Mining Network Data*, 2006, pp. 281-286.

[23] L. Yu, S. Wang, K. K. Lai, and F. Wen, "A multiscale neural network learning paradigm for financial crisis forecasting," *Journal of Neurocomputing*, vol. 73, no. 4–6, 2010, pp. 716–725.

[24] S. Fathi, R. Srinivasan, A. Fenner, and S. Fathi, "Machine Learning Applications in Urban Building Energy Performance Forecasting: A Systematic review," *Journal of Renewable and Sustainable Energy reviews*, vol. 133, 2020, pp 1-13.

[25] G. B. Kim, W. J. Kim, H. U. Kim, and S. Y. Lee, "Machine Learning Applications in Systems Metabolic Engineering," *Journal of Current Opinion in Biotechnology*, vol. 64, 2020, pp. 1-9.

[26] E. Tohidi, R. Amiri, M. Coutino, D. Gesbert, G. Leus, and A. Karbasi, "Submodularity in Action: From Machine Learning to Signal Processing Applications," *IEEE Signal Processing*, vol. 27, 2020, pp.1-12.

[27] B. Kim, J. Bae, and B. T. Lee, ''Modeling of silicon oxynitride etch microtrenching using genetic algorithm and neural network,'' *Microelectronic Engineering*, vol. 83, no. 3, 2006, pp. 513–519.

[28] B. Kim, M. Kwon, and S. H. Kwon, "Modeling of plasma process data using a multi-parameterized generalized regression neural network," *Microelectronic Engineering,* vol. 86, no. 1, 2009, pp. 63–67.

[29] D. Braha and A. Shmilovici, "Data mining for improving a cleaning process in the semiconductor industry," *IEEE Transactions on Semicondonductor Manufacturing*, vol. 15, no. 1, 2002, pp. 91–101.

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

[31] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, and U. Lee, "Deep-structured machine learning model for the recognition of mixed-defect patterns in semiconductor fabrication processes," *IEEE Transaction on Semiconductor Manufacturing*, vol. 31, no. 2, 2018, pp. 315–322.

[32] K. Nakata, R. Orihara, Y. Mizuoka, and K. Takagi, "A comprehensive big-data-based monitoring system for yield enhancement in semiconductor manufacturing," *IEEE Transaction on Semiconductor Manufacturing*, vol. 30, no. 4, 2007, pp. 339–344.

[33] J. Abuogo, Z. Zao, and J. Ke, "Linear regression model for screening SiC MOSFETs for paralleling to minimize transient current imbalance," in *IOP Conference Series: Materials Science and Engineering*, 2019, pp. 1-8.

[34] J. O. Abuogo and Z. Zhao, "Machine learning approach for sorting SiC MOSFET devices for paralleling," *Journal of Power Electronics,* vol. 20, no. 1, 2020, pp. 329-340.

[35] M. Baharani, M. Biglarbegian, B. Parkhideh, and H. Tabkhi, "Real-Time Deep Learning at the Edge for Scalable Reliability Modeling of Si-MOSFET Power Electronics Converters," *IEEE Internet of Things Journal,* vol. 6, no. 5, 2019, pp. 7375-7385.

[36] D. McMenemy, W. Chen, L. Zhang, K. Pattipati, A. M. Bazzi, and S. Joshi, "A Machine Learning Approach for Adaptive Classification of Power MOSFET Failures," in *IEEE Transportation Electrification Conference and Expo ITEC*, 2019, pp. 1-8.

[37] F. K. a. I. Partin-Vaisband, "A Single-MOSFET MAC for Confidence and Resolution (CORE) Driven Machine Learning Classification," *Electrical Engineering and Systems Science,* 2019, pp. 1-9.

[38] M. Sarvaghad-Moghaddam, A. A. Orouji, Z. Ramezani, M. Elhoseny, A. Farouk, and N. Arun kumar, "Modelling the spice parameters of SOI MOSFET using a combinational algorithm," *Cluster Computing,* vol. 22, no. 2, 2019, pp. 4683-4692.

[39] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature,* vol. 521, no. 7553, 2015, pp. 452-459.

[40] A. Gosavi, "A Reinforcement Learning Algorithm Based on Policy Iteration for Average Reward: Empirical Results with Yield Management and Convergence Analysis," *Machine Learning,* vol. 55, no. 1, 2004, pp. 5-29.

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

[42] B. S. Kang, J. H. Lee, C. K. Shin, S. J. Yu, and S. C. Park, "Hybrid machine learning system for integrated yield management in semiconductor manufacturing," *Expert Systems with Applications,* vol. 15, no. 2, 1998, pp. 123-132.

[43] Y. Li, Y. Zhang, X. Huang, H. Zhu, and J. Ma, "Large-Scale Remote Sensing Image Retrieval by Deep Hashing Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 56, no. 2, 2018, pp. 950-965.

[44] W. Lin, Y. Hu, and C. Tsai, "Machine Learning in Financial Crisis Prediction: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C Applications and Reviews,* vol. 42, no. 4, 2012, pp. 421-436.

[45] A. H. Neto and F. A. S. Fiorelli, "Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption," *Energy and Buildings,* vol. 40, no. 12, 2008, pp. 2169-2176.

[46] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys & Tutorials,* vol. 10, no. 4, 2008, pp. 56-76.

[47] K. Suzuki, "Survey of Deep Learning Applications to Medical Image Analysis," *Medical Imaging Technology,* vol. 35, no. 4, 2017, pp. 212-226.

[48] A. L. Tarca, V. J. Carey, X. W. Chen, R. Romero, and S. Drăghici, "Machine Learning and Its Applications to Biology," *PLOS Computational Biology,* vol. 3, no. 6, 2007, pp. 0953-0963.

[49] H. Carrillo-Nuñez, N. Dimitrova, A. Asenov, and V. Georgiev, "Machine Learning Approach for Predicting the Effect of Statistical Variability in Si Junctionless Nanowire Transistors," *IEEE Electron Device Letters,* vol. 40, no. 9, 2019, pp. 1366-1369.

[50] K. Ko, J. K. Lee, M. Kang, J. Jeon, and H. Shin, "Prediction of Process Variation Effect for Ultrascaled GAA Vertical FET Devices Using a Machine Learning Approach," *IEEE Transactions on Electron Devices,* vol. 66, no. 10, 2019, pp. 4474-4477.

[51] K. Ko, J. K. Lee, and H. Shin, "Variability-Aware Machine Learning Strategy for 3-D NAND Flash Memories," *IEEE Transactions on Electron Devices,* vol. 67, no. 4, 2020, pp. 1575-1580.

[52] Y. Li, H. Cheng, and M. Han, "Statistical Simulation of Static Noise Margin Variability in Static Random Access Memory," *IEEE Transactions on Semiconductor Manufacturing,* vol. 23, no. 4, 2010, pp. 509-516.

[53] F. a. V. Pedregosa, G. and Gramfort, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825-2830.

[54] F. Ambrosino, C. Sabbarese, V. Roca, F. Giudicepietro, and G. Chiodini, "Analysis of 7-years Radon time series at Campi Flegrei

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

area (Naples, Italy) using artificial neural network method," *Applied Radiation and Isotopes*, vol. 163, 2020, p. 109239.
[55] S. Badger and J. Bullock, "Using neural networks for efficient evaluation of high multiplicity scattering amplitudes," *Journal of High Energy Physics,* vol. 2020, no. 6, 2020, p. 114.
[56] A. Di Piazza, M. C. Di Piazza, G. La Tona, and M. Luna, "An artificial neural network-based forecasting model of energy-related time series for electrical grid management," *Mathematics and Computers in Simulation,* 2020, pp 1-12.
[57] S. Javed, M. Zakirulla, R. U. Baig, S. M. Asif, and A. B. Meer, "Development of artificial neural network model for prediction of post-streptococcus mutans in dental caries," *Computer Methods and Programs in Biomedicine,* vol. 186, 2020, p. 105198.
[58] T. Liu, H. Mei, Q. Sun, and H. Zhou, "Application of neural network in fault location of optical transport network," *China Communications,* vol. 16, no. 10, 2019, pp. 214-225.
[59] R. Asadi and A. C. Regan, "A spatio-temporal decomposition based deep neural network for time series forecasting," *Applied Soft Computing,* vol. 87, 2020, p. 105963.
[60] A. Zeroual, F. Harrou, A. Dairi, and Y. Sun, "Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study," *Chaos, Solitons & Fractals,* vol. 140, 2020, p. 110121.
[61] D. Konstantinidis, V. Argyriou, T. Stathaki, and N. Grammalidis, "A modular CNN-based building detector for remote sensing images," *Computer Networks,* vol. 168, 2020, p. 107034.
[62] R. Rosati, L. Romeo, S. Silvestri, F. Marcheggiani, L. Tiano, and E. Frontoni, "Faster R-CNN approach for detection and quantification of DNA damage in comet assay images," *Computers in Biology and Medicine,* vol. 123, 2020, p. 103912.
[63] C. Tian, Y. Xu, Z. Li, W. Zuo, L. Fei, and H. Liu, "Attention-guided CNN for image denoising," *Neural Networks,* vol. 124, 2020, pp. 117-129.
[64] R. Yang, "CNN-LSTM deep learning architecture for computer vision-based modal frequency detection," *Mechanical Systems and Signal Processing,* vol. 144, 2020, p. 106885.
[65] U. R. Acharya, "A deep convolutional neural network model to classify heartbeats," *Computers in Biology and Medicine,* vol. 89, 2017, pp. 389-396.
[66] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *International Conference on Engineering and Technology ICE-IEEE*, 2017, pp. 1-6.
[67] B. N. Chenglong Zhao, Jian Zhang, Qiwei Zhao, Wenjun Zhang, Qi Tian, "Variational Convolutional Neural Network Pruning," in *Proceedings of Conference on Computer Vision and Pattern Recognition CVPR,* 2019, pp. 2780-2789.
[68] C. Dong, C. C. Loy, and X. Tang, "Accelerating the Super-Resolution Convolutional Neural Network," in *European Conference on Computer Vision ECCV*, 2016, pp. 391-407.
[69] C. Z. Xiaofan Lin, W. Pan, "Towards Accurate Binary Convolutional Neural Network," in *Conference on Neural Information Processing Systems NeurIPS,* 2017, pp. 1-14.
[70] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *European Journal of Operational Research,* vol. 270, no. 2, 2018, pp. 654-669.
[71] B. Zhang, S. Zhang, and W. Li, "Bearing performance degradation assessment using long short-term memory recurrent network," *Computers in Industry,* vol. 106, 2019, pp. 14-29.
[72] K. Zhang, W. L. Chao, F. Sha, and K. Grauman, "Video Summarization with Long Short-Term Memory," in *European Conference on Computer Vision ECCV,* 2016, pp. 766-782.
[73] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long Short-Term Memory Network for Remaining Useful Life estimation," in *IEEE International Conference on Prognostics and Health Management ICPHM*, 2017, pp. 88-95.
[74] Y. Zhou, Y. Huang, J. Pang, and K. Wang, "Remaining useful life prediction for supercapacitor based on long short-term memory neural network," *Journal of Power Sources,* vol. 440, 2019, p. 227149.
[75] W. L. Sung, Y. S. Yang, and Y. Li, "Work- function fluctuation of gate-all-around silicon nanowire n-MOSFETs: A unified comparison between Cuboid and Voronoi Methods," *Journal of Electron Devices Society,* vol. 9, 2021, pp. 151-159

Chandni Akbar received a Master's degree in Electronics in 2016 from Quaid-i-Azam University, Islamabad, Pakistan. Currently, she is pursuing a Ph.D. degree at the Parallel and Scientific Computing Laboratory in National Yang Ming Chiao Tung University, Hsinchu, Taiwan. Her research interests focus on machine learning and deep learning algorithms and their applications in advanced nano-scaled semiconductor device simulation and optimization.

Yiming Li (M'02) is currently a Full Professor of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan. He has authored or co-authored over 350 research papers appearing in international book chapters, journals, and conferences. His current research interests include computational electronics, device physics, semiconductor nanostructures, modeling and parameter extraction, biomedical and energy harvesting devices, and optimization techniques. He has been a program committee of IEDM since 2011.

Wen-Li Sung received a B.S. degree in Physics in 1995 from National Tsing Hua University and a M.S. degree in Electronics in 2000 from National Yang Ming Chiao Tung University (NYCU), Hsinchu, Taiwan. Currently, he is a Ph.D. student at the Parallel and Scientific Computing Laboratory and Institute of Communications Engineering, NYCU. His research interests focus on simulation and optimization of advanced nano-scaled MOSFETs and circuits.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/