IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 11, 2018, accepted November 27, 2018, date of publication December 4, 2018,
date of current version December 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2885024

Intelligent Manufacturing: TCAD-Assisted Adaptive Weighting Neural Networks

CHIEN Y. HUANG, SZE M. FU, PARAG PARASHAR, CHUN H. CHEN, CHANDNI AKBAR, AND ALBERT S. LIN
Department of Electronics Engineering, National Chiao-Tung University, Hsinchu 30010, Taiwan
Corresponding author: Albert Lin (hdtd5746@gmail.com)

This work was supported by the Ministry of Science and Technology (MOST), Taiwan under grant number MOST 106-2628-E-009 -010 -MY3.

ABSTRACT Using machine intelligence on device and process performance prediction is an emerging methodology in the IC industry. While semiconductor technology computer-aided design (TCAD) has been researched and developed for over 30 years, it should contribute to or be used in conjunction with machine learning algorithms in solution finding procedure. Here, we propose an adaptive weighting neural network (AWNN) model that combines the advantages of statistical the machine learning model and the physical TCAD model. Using aspect ratio dependent etching as an example, our proposed AWNN outperforms conventional artificial neural network in terms of mean square errors in the test set where 5–10 times reduction is observed. The effectiveness of the TCAD AWNN model can be especially effective in the case of sampling over a vast sample space since the under-sampling problem can be compensated by the TCAD model. The large and nearly unbounded sample space is very common in IC technology, where cascaded and repeated process steps exist (∼150 process steps and ∼20 masks for 90-nm CMOS process).

INDEX TERMS Machine learning algorithms, artificial neural networks, semiconductor device manufacture, semiconductor process modeling.

I. INTRODUCTION

In recent years, the research on machine learning has progressed significantly due to its feasibility and broad applicability [1]–[17]. The purpose of machine learning is to predict the results or behaviors of new action from collected experimental data through training procedures. For example, Fabrizio *et al.* demonstrate integrating the data mining cloud service into scalable data analysis workflows [12]. Saman *et al.* propose a reversible watermarking technique for social network data analytics [13]. Yansheng *et al.* utilize deep hashing neural networks in remote sensing [15]. Litjens *et al.* apply machine learning to medical imaging [16]. Sommer *et al.* utilized machine learning to recognize phenotypes in biology [17]. However, in the semiconductor industry, the complicated process steps and a large number of process parameters during the manufacturing process makes machine learning highly suitable to be used in the semiconductor industry. There are already some prior works related to using machine learning in semiconductor manufacturing [18]–[37]. Tello *et al.* propose a deep-structured machine learning model for defect detection [33]. Susto *et al.* demonstrate an anomaly detection approaches in semiconductor manufacturing [34]. Nakata *et al.* propose a comprehensive big-data-based monitoring system for yield improvement in semiconductor manufacturing [35]. Kang *et al.* demonstrate the application for yield management in semiconductor manufacturing [36]. Kim *et al.* try to establish a model for plasma etching process by neural network [37]. In these prior works, standard neural network training and prediction are employed. In many cases in semiconductor manufacturing, only a limited amount of experimental data can be made available since the semiconductor manufacturing processes are time-consuming and expensive. Therefore, we cannot conduct as many experiments as desired, and undersampling is inevitable in this case. As a result, the training dataset can only contain a narrow range of experimental parameters, and thus the output of the neural network will have limited prediction capability. To resolve this issue, we proposed a supervised machine learning neural network which includes the experimental data as well as the TCAD data. By the combination of these two types of training dataset, we look forward to establishing a more accurate and applicable model using a limited amount of precious experiment data points.

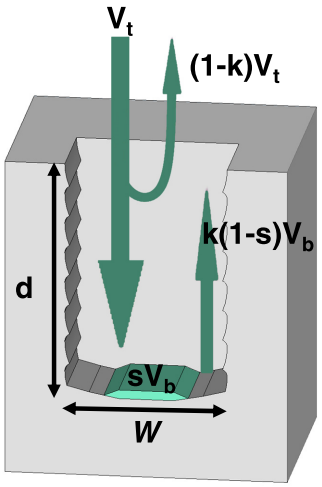

FIGURE 1. Schematic diagram of modeling aspect-ratio dependent etching (ARDE) using gas kinetics, proposed by IBM [38].

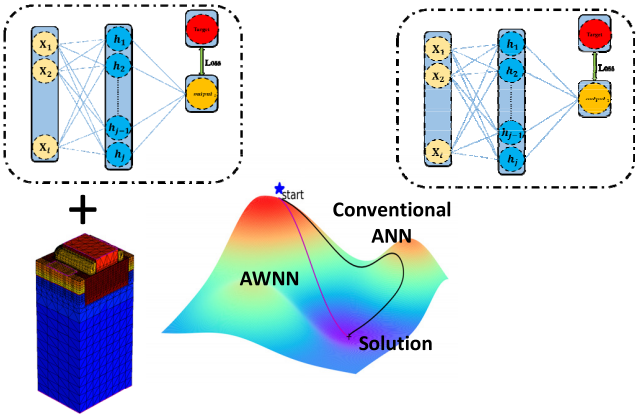

FIGURE 2. Illustration of the AWNN vs conventional ANN methods in the optimal process parameters locating.

In this work, we propose a supervised machine learning model called adaptive weighting neural networks (AWNN) is proposed. The concepts of this model are demonstrated in Fig. 2. In this model, the TCAD data and experimental data are both utilized to improve the accuracy of the prediction. Initially, we will simulate the etching process by using TCAD. Varied complexity level of TCAD model can be found in literature, and there is abundance of commercial software packages. In regular Sentaurus^{TM} Process, the aspect-ratio dependent etching (ARDE) cannot be modeled, and a more advanced software package Sentaurus^{TM} topography has to be used. Instead of using complex Sentaurus^{TM} topography modeling, we employed a simple ARDE model from IBM based on gas kinetics and chemical reaction as illustrated in Fig. 1.

The outcome of the TCAD calculation is the etching depth, and the parameters used to predict the etching depth are the trench width and the etching time. Apart from TCAD, we conduct the experiment for deep reactive ion etching (DRIE) BOSCH etching process to acquire the experiment data. The etching patterns in the experiment are set the same as in the simulation, and therefore, the input feature values are the same for TCAD calculations and physical experiments. The etching process steps consist of pattern exposure, development, and dry plasma etching. Finally, the actual etching depth is examined with the scanning electron microscope (SEM). After experiment, the experiment data set will be fed into our artificial neural network first. Then the output result of experimental data will be combined with the TCAD data. The weighting for the experimental data and TCAD data are w_{NN} and w_{TCAD}, respectively. Furthermore, w_{NN} and w_{TCAD} are input-parameter-dependent, which is modeled by two additional ANN. By the inclusion of simulation data, it can reduce the required number of experimental data points. Therefore, the proposed model can predict the result more precisely while saving the cost and time in processing.

TABLE 1. Pattern parameters are designed for diverse aspect ratio (AR).

Parameters	Value	Units
Space	5, 1, 0.5	μm
Line width	1, 0.8, 0.5, 0.3	μm

II. METHOD

A. FABRICATION AND TCAD MODEL

The 6 inch, (100) oriented, p-type boron-doped Si test-grade wafer was prepared for samples fabrication. The resistivity of Si wafer ranged between 1.5 and 100 Ω-cm. Before using Track (TEL CLEAN TRACK MK-8) to spin on e-beam positive photoresist (TDUR-P015), Si wafer was cleaned for 600s by SPM solution ($4H_2SO_4$:$1H_2O_2$, 120°C) with DI water rinse and then by DHF solution (1HF:$1H_2O$, 25°C). The 700 nm thick e-beam positive photoresist was coated at 1000rpm and soft baked for 90s at 100°C. The e-beam lithography (EBL) was executed on Leica Weprint 200 E-beam stepper. The exposure beam energy was 40 keV, and the exposure dose, 5μC/cm^2, was selected to form the patterns with 5, 1 and 0.5μm spacing with different sizes of trench widths (W), as listed in Table 1. In our BOSCH process using Oxford^{TM} machine, the micro-loading effect is not very pronounced in our one-dimensional (1D) trench pattern. Therefore, the spacing between adjacent trenches is excluded from the neural network input features. In a more complex two-dimensional (2D) pattern, the micro-loading effect can be non-negligible, but the same AWNN methodology can be applied to 2D patterns with an arbitrary number of input features.

The post-exposure baking was conducted at 120°C for 90s. The patterned wafer was sequentially developed in a 2.38% tetramethylammonium hydroxide (TMAH) solution for 80s and hard-baked for 60s at 115°C. After developing, the wafer was inspected for the different line widths, by using critical dimension scanning electron microscope (CD-SEM, HITACHI S-6280H) and then diced into 36 pieces of 1.2cm×1.2cm samples. The Bosch process for deep Si etching was carried out in ICP-RIE (Oxford^{TM} Estrelas 100). Before etching, the dicing sample was bonded onto 4-inch

TABLE 2. Experimental parameters for main etching step in BOSCH process.

Parameters	Value	Units
Pressure	40	mTorr
ICP RF power	1250	W
RF Bot power	0	W
SF_6	200	sccm

Si maintaining wafer, which was deposited 2um thick silicon dioxide on it, with vacuum grease in order to enhance thermal conductance between the sample and the maintaining wafer. During whole etching process, the bottom of the maintaining wafer was held at 25°C under helium cooling. The top solenoidal coil for ICP source generation was operated with power between 1000 and 1750W. The bottom of electrode was used to accelerate ion to bombard surface of the sample and the power was below 60 W for etching process. In the Bosch process, there are three steps in a cycle, and the range of experimental parameters of main etching process are shown in Table 2. Additionally, the distinct total etching time for a sample were set to 50, 20, 15, and 10 cycles. A scanning electron microscope (SEM, Hitachi SU-8010) was used for checking etching line widths and depth for the processed samples. Fig. 3 shows the micrograph of the samples after Bosch etching process. The experimental conditions were at 40 mTorr pressure, 1250 W ICP power, 200 sccm SF_6 flow rate, and 50-cycle etching time.

The calculation of the etching depth taking into account the aspect ratio dependent etching can be done by considering the conservation of gas flow [38]

$$v_t - (1 - k(\frac{d}{W}))v_t - k(\frac{d}{W})(1 - s)v_b = sv_b. \quad (1)$$

where k is the transmission probability or Knudsen coefficient, v_t is flux incidence of gas at the top of etched feature, v_b is the flux species at the bottom of etched feature, and s is the reaction probability on the bottom surface of etched feature. The etching rate when etching started can be calculated by

$$R(\frac{d^{(t)}}{W}) = R(0)\frac{k(\frac{d^{(t)}}{W})}{k(\frac{d^{(t)}}{W}) + (1 - k(\frac{d^{(t)}}{W}))s} \quad (2)$$

where $R(0)$ and $R(d^{(t)}/w)$ is the etching rate at the top and the bottom of the etched feature respectively. The etching depth after certain period of time can be known by integration as in

$$d^{(t+dt)} = d^{(t)} + R\left(\frac{d^{(t)}}{W}\right)dt \quad (3)$$

where $d^{(t+dt)}$ is the etching depth at time of t+dt, and $d^{(t)}$ is the depth of the feature at time of t.

B. CONVENTIONAL ARTIFICIAL NEURAL NETWORK (ANN) MODEL

ANN is considered as a "black box" connecting inputs with outputs through an explicit set of non-linear functions shown in Fig. 4. Generally, ANN model consists of the following

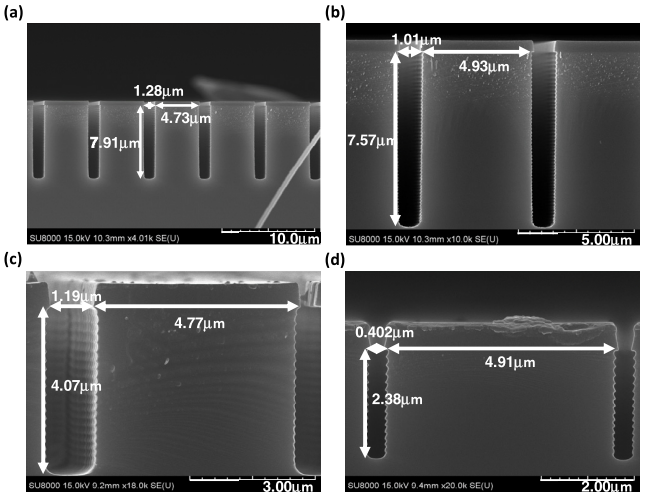

FIGURE 3. Cross-sectional SEM micrograph of the etched pattern: (a) and (b) are carried out at 40 mTorr pressure, 1250 W ICP power, 200 sccm SF_6 flow rate, and 50 cycles etching time but different line widths (1μm and 0.8μm respectively). (c) and (d) are under the same etching process, but the line widths and etching time are 1μm, 20 cycles and 0.3μm, 15cycles individually.

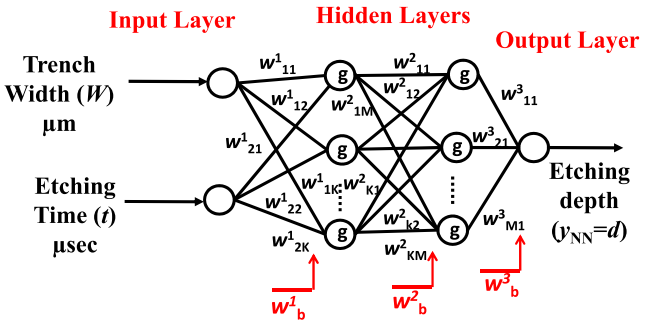

FIGURE 4. General ANN representation, i.e. input layer, two hidden layers, and output layer. Trench width (μm), etching time (μsec) and etching depth (μm) are input and output parameters respectively.

steps: accumulating of experimental data, adjudicating of input and output parameters, pre-processing of collected data, training of ANN model, testing the trained model, and evaluating the performance of ANN model. ReLU activation functions have lower cost value and converge faster as compare to sigmoid and hyperbolic tangent function due to no exponential function in ReLU. Escaping from local optimization and obtaining sparse representation can be easily achieved by ReLU activation functions. Its derivative also considered as left-hard-saturation i.e. $x < 0$, so that ANN with ReLU activation function, is converged only with positive values of the input dataset.

Considering the 4-layers ANN with $X = \{x_1, x_2, \ldots, x_n\}$ as the input parameters, and the activation function is g. The mathematical notation of output h_j^1 from any arbitrary 'j' node in first hidden layer is given as.

$$h_j^1 = g(\sum_{i=1}^{k} x_i \times w_{ij}^1 + w_b^1) \quad (4)$$

wherew_o is the bias in hidden layer, w_{ij}^1 represents the weight between any arbitrary ith node in input layer and jth node in hidden layer respectively. x_i is the ith input from 'n' dimensional input vector X. The definition of ReLU as an activation function is given as

$$g(x) = \max(0, x) = \begin{cases} x, & if\ x \geq 0 \\ 0, & if\ x \leq 0 \end{cases} \tag{5}$$

ReLU activation function is non-saturated and its derivative is given as

$$g'(x) = \begin{cases} 1, & if\ x \geq 0 \\ 0, & if\ x < 0 \end{cases} \tag{6}$$

The output of the arbitrary pth node in the second hidden layer h_p^2 is represented by taking input, as the output of each neuron from the first hidden layer h^1 with their corresponding weights is given as.

$$h_p^2 = g(\sum_{i=1}^{k} h_i^1 \times w_{ip}^2 + w_b^2) \tag{7}$$

where k represents the number of neurons in the first hidden layer, w_o is the bias and g is the activation function.

The output of ANN model, also known as feed forward propagation, is calculated by the summation of activation function with their corresponding weights is given as

$$y_{NN} = \sum_{j=1}^{M} h_j^2 \times w_{j1}^3 + w_b^3 \tag{8}$$

where M is the number of neurons or nodes in the second hidden layer. The loss of ANN can be determined by

$$Error = \frac{1}{2}\sum_{t=1}^{s}(y_{exp,t} - y_{NN,t})^2 + \frac{\beta}{2}\sum_{k=1}^{3}\sum_{i}\sum_{j} w_{ij}^k \tag{9}$$

where β is the regularization hyper-parameter, and s is the size of training dataset. y_{exp} and y_{NN} is the original and the predicted value respectively..

C. ADAPTIVE NEURAL NETWORK MODEL

This section consists of a description of adaptive weighting neural network (AWNN). The training procedure is illustrated in Fig. 5, and the AWNN structure is illustrated in Fig. 6. The weights w_{NN} and w_{TCAD} are used to achieve a balance between machine learning prediction and physical models based on chemistry and fluid dynamics. Moreover, the proposed AWNN is not very complicated and thus provides an optimally balanced and controlled adaptive neural-network-based model with low complexity.

Firstly, data is collected from the experiment and TCAD simulation. The data consists of two parts i.e. experiment dataset by semiconductor processing and the TCAD dataset by TCAD physical models. For training and verification

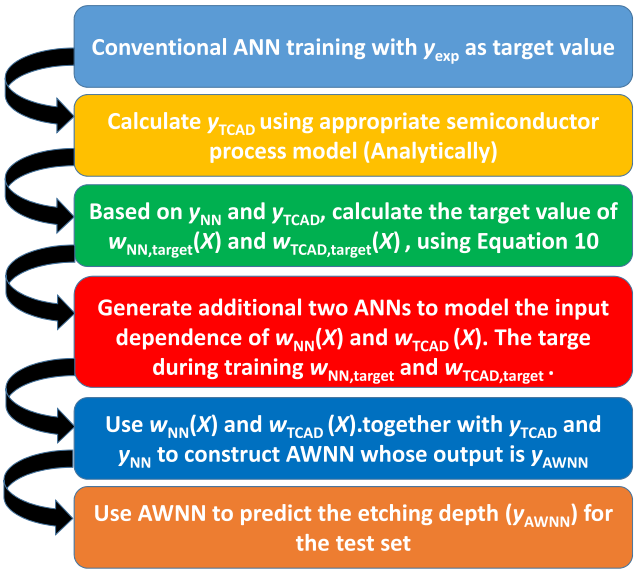

FIGURE 5. The flow chart of the adaptive weighting neural network (AWNN) implementation.

purpose, the experiment or TCAD datasets are split into the training and the test dataset using parameter *partition*as a boundary. In this work, we set the*partition* equal to 15, 30, 45 for a comprehensive comparison. By increasing the *partition* value, the training dataset expands and test dataset contracts, respectively. Both datasets have two independent inputs parameters, namely etching time (sec) and trench width (μm), and one output parameter etching depth (μm).

The Python and Scikit Learn are used in this paper. Collected data is cleaned and pre-processed by scaling and normalization. Scikit Learn is an efficient toolbox of Python for Machine Learning (ML), whose functionality includes classification, regression, clustering, etc. For the continuous variable problems, multi-layer perceptron (MLP) regression model can be employsed. The main hyper-parameter specification is (50, 50) for the neuron number in each hidden layer, 0.9 momentum in stochastic gradient descent for neural network weight locating, maximum iteration of 10000 with tolerance of 0.00001, and initial learning rate of 0.0001. The proposed AWNN structure mainly consists of three ANN, including 1) Traditional ANN trained by the experiment data set 2) ANN for w_{NN} 3) ANN for w_{TCAD}.

The baseline for comparison is a primary artificial neural network (ANN), and the training procedure is the same as in the literature where a training set data points are firstly fed into the ANN. Afterward, by using stochastic gradient descent (SGD), a maximum likelihood (MLE) neural network weights can be evaluated. In the control group basic ANN model, an experimental dataset is used solely. During the training stage, the weights and bias are being optimized in SGD iteration to minimize the loss function, as expressed in (9). In contrast, the AWNN algorithm is a little bit more complex, and both experiment and TCAD data sets

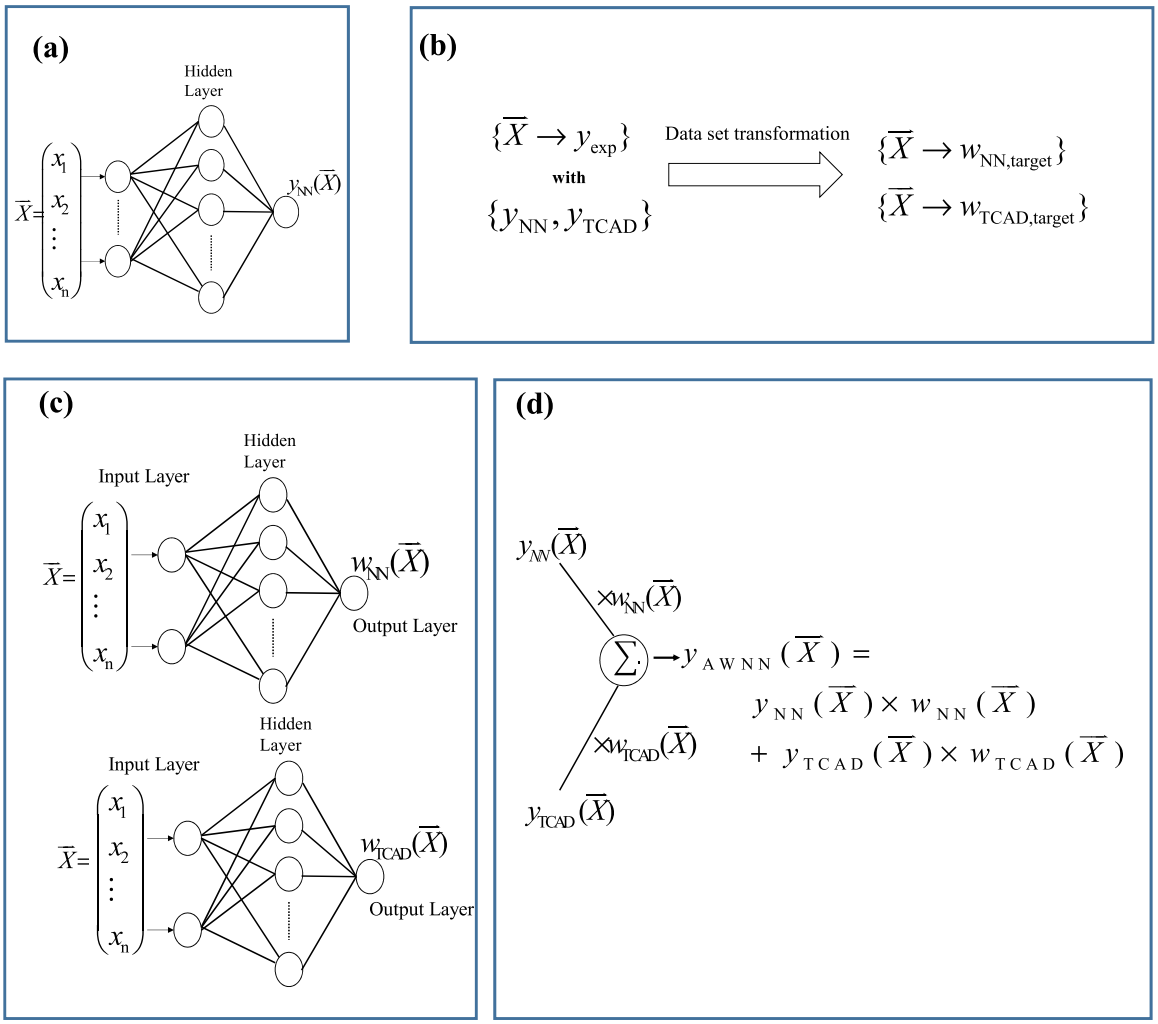

FIGURE 6. The proposed adaptive weighting neural network (AWNN) model with three ANNs. (a) Represents the basic ANN to predict the y_{NN} from the experimental training set. (b) Illustrates the mapping of the dataset to generate $w_{NN,target}$ and $w_{TCAD,target}$, the target values for training the neural network for w_{NN} and w_{TCAD} as illustrated in part (c). (c) represents two additional ANNs to model the input dependence of w_{NN} and w_{TCAD}. (d) illustrates the calculation of final predicted value y_{AWNN} by y_{NN} and y_{TCAD}.

are utilized. As illustrated in Fig. 6, the first step is quite similar to basic ANN training where we use a training set to calculate the MLE weights in the network. The second step is the construction of the w_{NN} and the w_{TCAD} neural network, in order to model the input dependence of the adaptive weighting scheme. Before evaluate the MLE weights and biases in these two ANNs for w_{NN} and w_{TCAD}, we need to know the target values of the w_{NN} and w_{TCAD}. With the information of y_{NN}, y_{TCAD}, and y_{exp} in hand, the target values of w_{NN} and w_{TCAD} are calculated as

$$w_{NN,target} = \begin{cases} \left|\dfrac{y_{NN} - y_{TCAD}}{y_{exp} - y_{TCAD}}\right|, & y_{TCAD} > y_{exp} > y_{NN} \\ & y_{TCAD} < y_{exp} < y_{NN} \\ 0, & y_{exp} > y_{TCAD} > y_{NN} \\ & y_{exp} < y_{TCAD} < y_{NN} \\ 1, & y_{exp} > y_{NN} > y_{TCAD} \\ & y_{exp} < y_{NN} < y_{TCAD} \end{cases} \tag{10a}$$

$$w_{TCAD,target} = \begin{cases} \left|\dfrac{y_{NN} - y_{TCAD}}{y_{exp} - y_{TCAD}}\right|, & y_{TCAD} > y_{exp} > y_{NN} \\ & y_{TCAD} < y_{exp} < y_{NN} \\ 1, & y_{exp} > y_{TCAD} > y_{NN} \\ & y_{exp} < y_{TCAD} < y_{NN} \\ 0, & y_{exp} > y_{NN} > y_{TCAD} \\ & y_{exp} < y_{NN} < y_{TCAD} \end{cases} \tag{10b}$$

The $w_{NN,target}$ and $w_{TCAD,target}$ is specified such that the TCAD output, y_{TCAD}, supplements the inaccuracy of the ANN output y_{NN}. In case the true value of etching depth y_{exp} is out of the bounds, i.e. outside the interval between y_{TCAD} and y_{NN}, the value of (1,0) or (0,1) are assigned to $w_{TCAD,target}$ and $w_{NN,target}$, respectively. If y_{exp} is in the interval between y_{TCAD} and y_{NN}, then weighted-averaged scheme is used to assign proper values to $w_{NN,target}$ and $w_{TCAD,target}$. It is important to distinguish between (w_{NN}, w_{TCAD}) and ($w_{NN,target}$, $w_{TCAD,target}$). As in regular NN training using

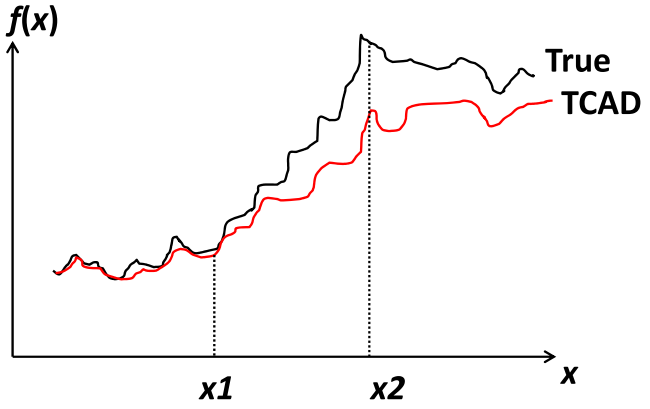

FIGURE 7. Illustration of the true function $f(X)$ and TCAD calculated function $f_{TCAD}(X)$. Initially, the function values are quite flat. After $x = x1$, $f(x)$ begins to increase, and after $x = x2$, $f(x)$ begins to saturate.

SGD, the target values are fixed during network training. As a result, $w_{\mathrm{NN,target}}$ and $w_{\mathrm{TCAD,target}}$ are calculated based on Eq.10, and their values are fixed during SGD iterations. On the other hand, what needs to be updated are w_{NN} and w_{TCAD}, and their values will change at each SGD iteration until the values converge. All of w_{NN}, w_{TCAD}, $w_{\mathrm{NN,target}}$, and $w_{\mathrm{TCAD,target}}$ are functions of input features $\vec{X}$ while the explicit dependence is not included in expressions every time to prevent wordiness.

As far as the relationship between $w_{\mathrm{NN}}(X)$ and $w_{\mathrm{NN,target}}(X)$ is concerned, in the training set, $w_{\mathrm{NN}}(X)$ and $w_{\mathrm{NN,target}}(X)$ will be quite similar if the neural network for $w_{\mathrm{NN}}(X)$ converges acceptably well. In the test set, there is no such a thing as $w_{\mathrm{NN,target}}(X)$, and $w_{\mathrm{NN}}(X)$ constructed by the adaptive weighting neural network can be very useful for test set prediction. Similar argument can be made for $w_{\mathrm{TCAD}}(X)$ and $w_{\mathrm{TCAD,target}}(X)$.

The ANNs of w_{NN} and w_{TCAD} in the proposed AWNN algorithm have two hidden layers. The input layer of both ANNs is also the input feature X, i.e. the trench width and the etching time, and the output is w_{NN} or w_{TCAD}. Each hidden layer has 50 neurons with ReLU activation functions, and the learning rate is set as 0.0001. In general, the adaption in weighting should depend on the input features, which means the degree of the correction by TCAD model depends on the trench width and the etching time. This is a common scenario since the accuracy of TCAD and conventional ANN model varies with input parameters X. Without constructing these additional two NN for w_{NN} and w_{TCAD}, the adaptive weighting correction becomes ineffective.

Mathematically, the w_{NN} and w_{TCAD} are a function of input parameters

$$w_{\mathrm{TCAD}} = f_1(\vec{X}) = f_1(W, t) \tag{11a}$$

$$w_{\mathrm{NN}} = f_2(\vec{X}) = f_2(W, t) \tag{11b}$$

where f_1 and f_2 represent the two neural networks for w_{NN} and w_{TCAD}, respectively. In this simple example of ARDE problem, the input feature array X consist of only two input parameters, i.e. trench width W and etching time t.

After the construction of the additional two neural networks for w_{NN} and w_{TCAD}, we should proceed to calculate the final predicted value by proposed AWNN scheme. This is illustrated in (d), and the calculation is simply by

$$y_{\mathrm{AWNN}}(\vec{X}) = y_{\mathrm{NN}}(\vec{X}) \times w_{\mathrm{NN}}(\vec{X}) + y_{\mathrm{TCAD}}(\vec{X}) \times w_{\mathrm{TCAD}}(\vec{X}) \tag{12}$$

where $y_{\mathrm{NN}}(\vec{X})$ and $y_{\mathrm{TCAD}}(\vec{X})$ are the predicted values from the traditional ANN and from the TCAD model, respectively.

Essentially, the error associated with ANN and AWNN algorithms, err_{ANN} and err_{AWNN}, can be expressed as

$$err_{ANN} = \int \left\| f_{ANN}(\vec{X}) - f(\vec{X}) \right\|^2 d\vec{X} \tag{13a}$$

$$\begin{aligned} err_{AWNN} &= \int \left\| f_{AWNN}(\vec{X}) - f(\vec{X}) \right\|^2 d\vec{X} \\ &= \int \Big\| [f_{ANN}(\vec{X}) w_{NN}(\vec{X}) \\ &\qquad + f_{TCAD}(\vec{X}) w_{TCAD}(\vec{X})] - f(\vec{X}) \Big\|^2 d\vec{X} \end{aligned} \tag{13b}$$

where $\vec{X}$ is input variable vector, f_{ANN} is the baseline ANN prediction. f_{TCAD} is the TCAD model prediction, f_{AWNN} is the proposed AWNN prediction, and f is the true function mapping input $\vec{X}$ to output. Theoretically, we want to prove that $err_{\mathrm{ANN}} > err_{\mathrm{AWNN}}$. Though it is difficult to conduct a generic proof since both $\mathrm{err}_{\mathrm{ANN}}$ and $\mathrm{err}_{\mathrm{AWNN}}$ are highly dependent on the sample space, $f(\vec{X})$. As a result, the effectiveness or any algorithms can only be defined after dataset or sample space is known. $\mathrm{err}_{\mathrm{ANN}}$ and $\mathrm{err}_{\mathrm{AWNN}}$ also depend on $f_{\mathrm{ANN}}(\vec{X})$ in terms of neuron number, hidden layer number, hyper-parameter selection. $\mathrm{err}_{\mathrm{AWNN}}$ also depends on $f_{\mathrm{TCAD}}(\vec{X})$, which is related to TCAD model selection. With so many factors affecting err_{ANN} and err_{AWNN}, it is very difficult to conduct a generic proof on $err_{\mathrm{ANN}} > err_{\mathrm{AWNN}}$, for all cases.

Graphically we can understand the enhancement of AWNN:

Consider we have a function $f(\vec{X})$ as shown below. The True function $f(\vec{X})$ and TCAD prediction $f_{\mathrm{TCAD}}(\vec{X})$ are all shown in Fig. 7. Initially the function values are quite flat, and after x=x1, $f(\vec{X})$ begins to increase, and after $x = x2$, $f(\vec{X})$ begins to saturate. This $f(\vec{X})$ is used as an example to highlight the effectiveness of AWNN. When we just start the experiment as illustrated in Fig. 8, insufficient experimental data points are collected. In this case, we have training set sampled until $x = x1$, and the prediction using baseline ANN will lead to alarge error in the test set. On the other hand, the AWNN algorithm provides enhancement in prediction since the TCAD fucntion $f_{\mathrm{TCAD}}(\vec{X})$ sees the trend of $f(\vec{X})$ for $x > x1$, as shown in Fig. 8(c). With $f_{ANN}(\vec{X}) w_{ANN}(\vec{X}) + f_{TCAD}(\vec{X}) w_{TCAD}(\vec{X})$, we predict better values using AWNN.

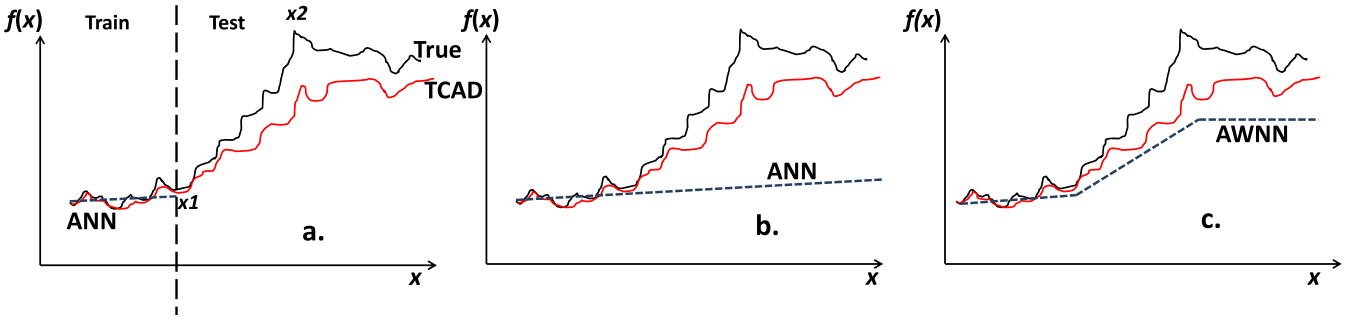

FIGURE 8. Initial trial-and-error stage with few experimental data points. The red and black lines are TCAD dataset and experimental dataset, respectively. (a) The training using baseline ANN. The dashed line indicates the separation between the training and the test set. (b) The prediction using baseline ANN. (c) The prediction using AWNN with TCAD assistance.

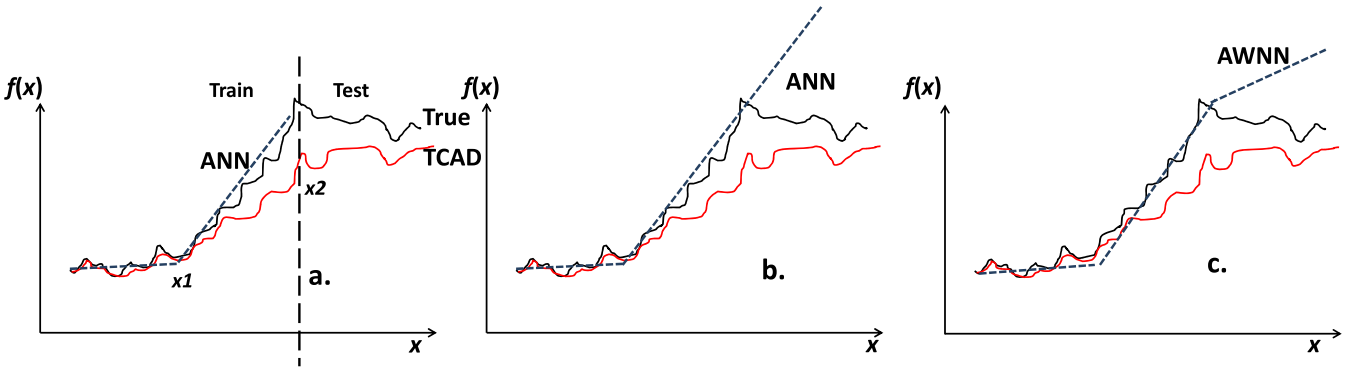

FIGURE 9. Intermediate trial-and-error stage with more experimental data points. The red and black lines are TCAD dataset and experimental dataset, respectively. (a) The training using baseline ANN. The dashed line indicates the separation between the training and the test set. (b) The prediction using baseline ANN. (c) The prediction using AWNN with TCAD assistance.

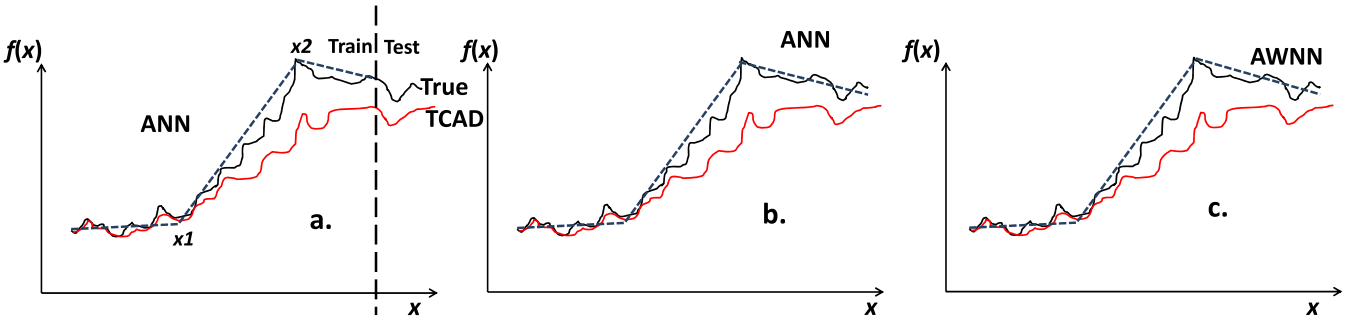

FIGURE 10. Late trial-and-error stage with abundant experimental data points. The red and black lines are TCAD dataset and experimental dataset, respectively. (a) The training using baseline ANN. The dashed line indicates the separation between the training and the test set. (b) The prediction using baseline ANN. (c) The prediction using AWNN with TCAD assistance.

After more experiments are conducted, we can have a better, more expressive training set as shown in Fig. 9. Now the training set sampling has reached $x = x2$. Similar to the previous case in Fig. 8, the AWNN effectiveness will now be reflected in the portion of dataset in $x>x2$. In Fig. 10, we demonstrate the situation where the training set sampling is complete and over the entire sample space, and it can be seen that AWNN is now not better than ANN. Nonetheless, we have to emphasize that if sampling is dense and thorough, we do not even need machine learning algorithms of any kind. In the case of Fig. 11 where the TCAD model is inaccurate to some extent, the AWNN can be worse than ANN. Normally when TCAD model trend is wrong, AWNN cannot provide advantages in prediction.

One aspect worth to mention is that a more advanced training method for AWNN is a two-training-subset approach. The training set is firstly divided into two subsets. The first training subset is used to train y_{NN} (baseline neural network output, Fig. 6), and the second subset is used to train $w_{NN}(X)$ and $w_{TCAD}(X)$. This can be a more robust method for difficult problems though in our case it is not necessary to use this complex two-training-subset scheme to

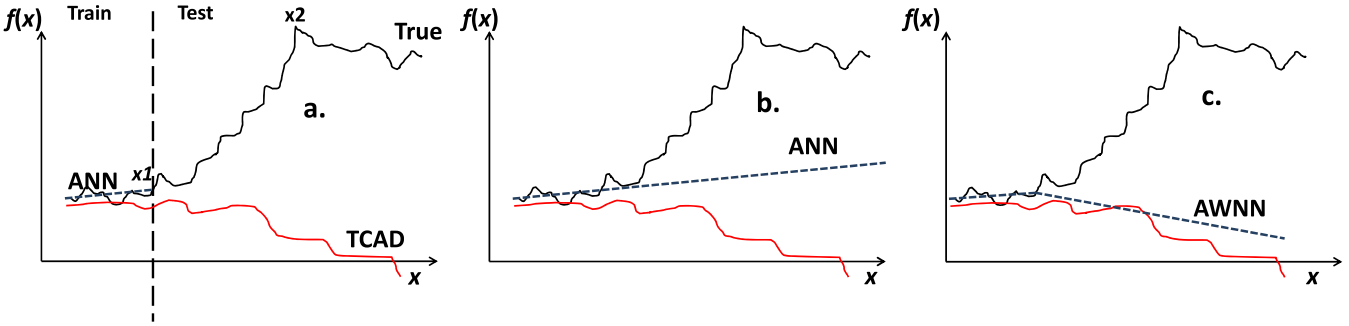

FIGURE 11. Initial trial-and-error stage with few experimental data points but an inaccurate TCAD model. The red and black lines are TCAD dataset and experimental dataset, respectively. (a) The training using baseline ANN. The dashed line indicates the separation between the training and the test set. (b) The prediction using baseline ANN. (c) The prediction using AWNN with TCAD assistance.

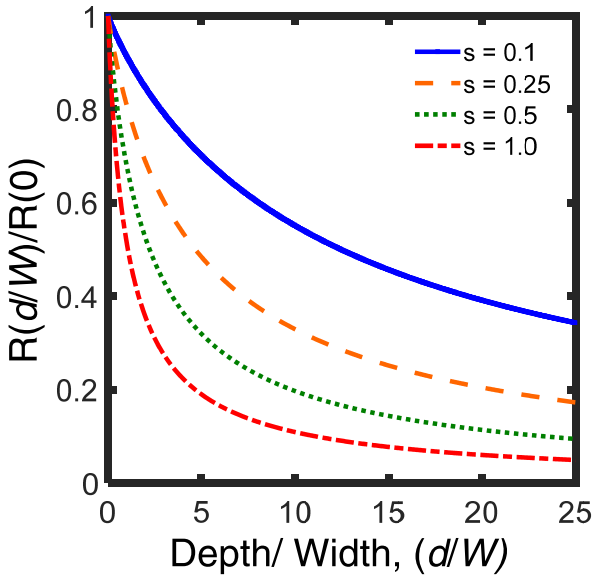

FIGURE 12. The simulated etching rate versus the aspect ratio under different reaction probability (s=0.1, 0.25, 0.5, 1).

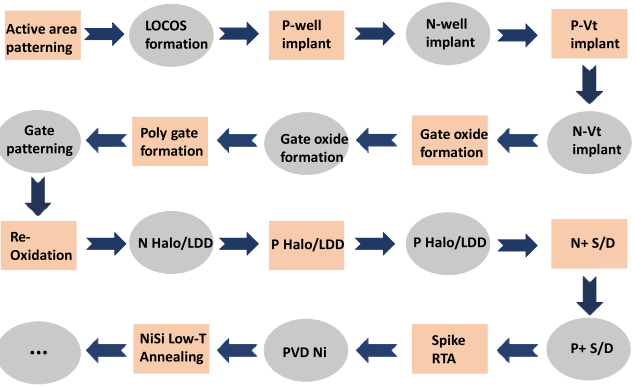

FIGURE 13. The standard 90nm CMOS process requires 150 steps, and there are 1000+ parameters used during the whole process.

attain satisfactory results. The convergence of ANN is not difficult in our case. In a difficult convergence learning problem, more neurons or more layers can be used.

III. RESULT AND DISCUSSION

The simulated etching depths using the TCAD model described in section II. In Fig. 12, it can be seen that the aspect ratio dependent etching (ARDE) exists in a standard Bosch process, and from Fig. 3 in the previous section, ARDE indeed exists in our Oxford^{TM} machine. The etchant gas flowing into the deep and narrow trench can be retarded, resulting in a decreased etching rate. Now we are going to use a proposed joint TCAD and machine learning method, to demonstrate its effectiveness in prediction.

In a regression problem, a mean squared error (MSE) is the summation of the square of the difference between the true values and the predicted values, divided by the degree of freedom. It measures the deviation of the predicted values with respect to the true values that is by Bosch process in our case. In our experimental and simulated data set, a *partition* parameter divides the data set into the training set and the test set. For the data point index from 1 to *partition*(data [:*partition*]), we treat them as the training set. For data point index from *partition* to the end of the data set (data [*partition* :]), we treat them as the test set. Defining partition parameters is used because in real semiconductor fabrication, more and more batches of experimental data are being collected when experiment is progressed. With varied *partition* values, we can observe the effect of AWNN model from the beginning of the experiment, in which very few data points are available, to the later stages of the experiment, in which more data points are collected.

It can be observed in Table 3 that in all cases, the test set MSE values are reduced by AWNN. The reduced MSE values are due to the effectiveness of the incorporated TCAD model in the machine learning prediction, and a more in-depth reasoning will be discussed in the following paragraphs. It may be argued that if we can collect a lot of data points, the TCAD assistance model is unnecessary and pure statistical algorithms such as ANN can be sufficient for the prediction. Nonetheless, as far as the semiconductor processing is concerned, collecting enough data points for the training purpose can be very difficult due to a very large sam-

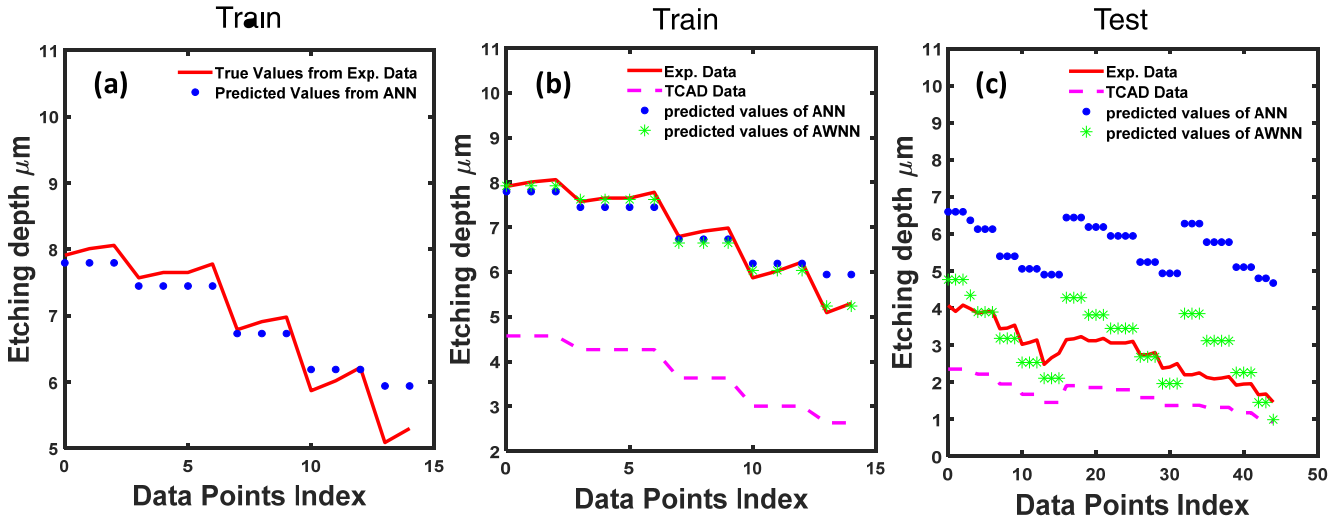

FIGURE 14. Comparison between AWNN and ANN for *Partition* = 15. (a) Training of ANN model using experiment dataset. (b) Training of ANN with experiment data set and the training of AWNN with both experiment and TCAD data set. (c) Testing of AWNN and ANN models in the test set.

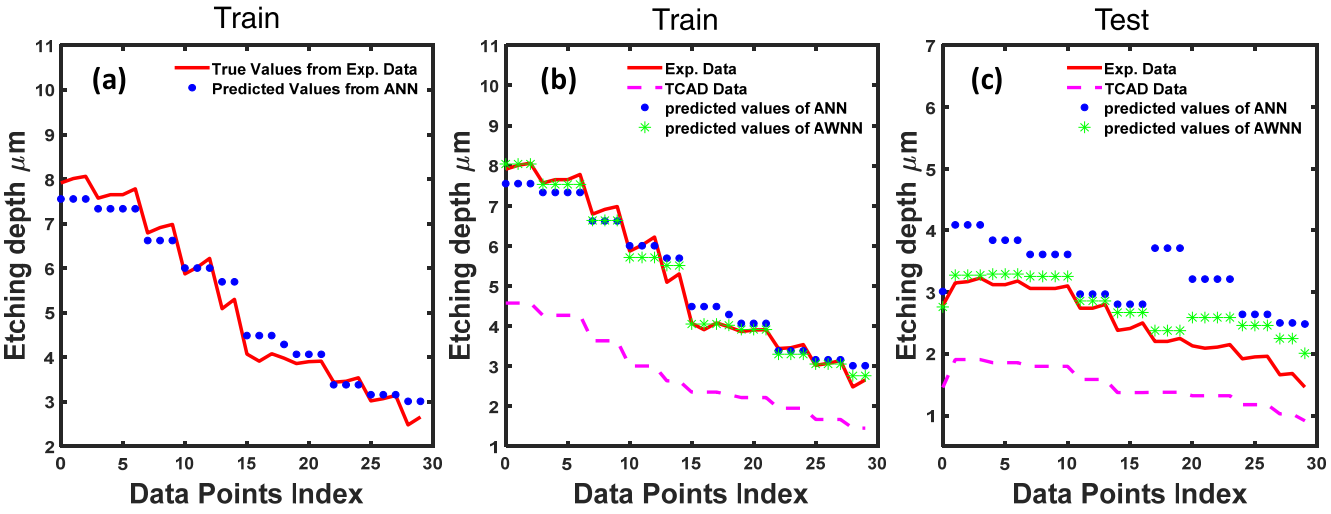

FIGURE 15. Comparison between AWNN and ANN for *Partition* = 30. (a) Training of ANN model using experiment dataset. (b) Training of ANN with experiment data set and the training of AWNN with both experiment and TCAD data set. (c) Testing of AWNN and ANN models in the test set.

ple space commonly encountered in this field, as illustrated in Fig. 13.

In Fig. 13, we see that even for a standard 90nm CMOS process, it requires ∼20 photolithography masks, ∼150 process steps. Considering that in average we have 20 parameters for each process steps, the total number of input parameters (features) will be ∼3000. If we coarsely divide each input feature into 10 divisions during training, the total number of data points in the sample space will be 10^{3000}, a number that definitely leads to severe under-sampling in any cases.

Fig. 14(a) illustrates the training results for *partition*=15. It can be seen that the conventional ANN can fit the training set in our ARDE problem. Fig. 14(b) compares the fitting to the training set for the case of using conventional ANN and using AWNN. The MSE values of AWNN and ANN in the case of *partition*=15 are listed in the first section of Table 3. The most pronounced difference in their prediction capability is shown in Fig. 14 (c) in which the prediction using AWNN on the test set is significantly improved over the conventional ANN model. We use different random state values in neural network weight initial guesses in stochastic gradient decent (SGD), and we list the corresponding MSE values in the different columns of Table 3. It can be seen that the MSE values on the test set are always much lower in the case of AWNN, compared to conventional ANN. In general, the etching depth (d) is dependent on the trench width (W), leading to a so-called aspect ratio (d/W) dependent etching effect (ARDE). ARDE effect can be highly chamber-dependent, together with a high dependence on etchant gas, process pressure, and RF plasma power. In fact, many semiconductor pro-

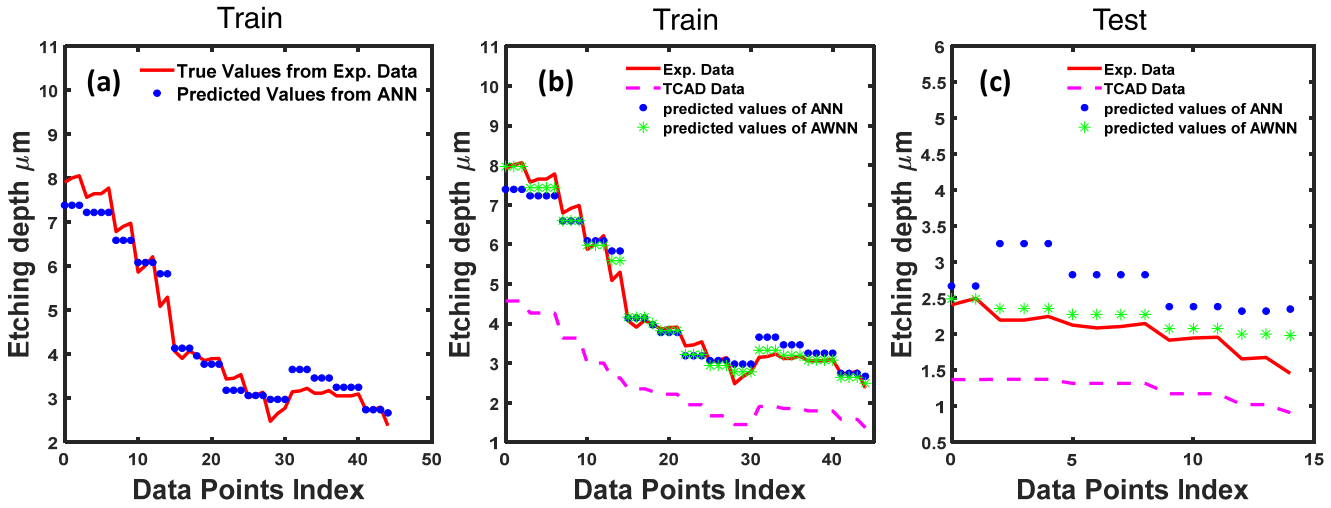

FIGURE 16. Comparison between AWNN and ANN for *Partition* = 45. (a) Training of ANN model using experiment dataset. (b) Training of ANN with experiment data set and the training of AWNN with both experiment and TCAD data set. (c) Testing of AWNN and ANN models in the test set.

TABLE 3. Mean square error (MSE) for the training and the test set. The comparison is made between ANN and AWNN.

	random state	1	2	3	4	5
Partition = 15	Training of ANN	0.0026	0.005	0.004	0.0024	0.006
	Testing of ANN	**0.1895**	**0.2066**	**0.2635**	**0.2863**	**0.2003**
	Training of AWNN	0.0005	0.0007	0.0006	0.0013	0.0135
	Testing of AWNN	**0.0115**	**0.0044**	**0.0076**	**0.0609**	**0.0071**
Partition = 30	Training of ANN	0.0025	0.0041	0.0020	0.0025	0.0028
	Testing of ANN	**0.0161**	**0.0246**	**0.0171**	**0.0151**	**0.0038**
	Training of AWNN	0.0009	0.0015	0.0012	0.0010	0.0014
	Testing of AWNN	**0.0024**	**0.0026**	**0.0030**	**0.0014**	**0.0006**
Partition = 45	Training of ANN	0.0026	0.0036	0.0018	0.0018	0.0026
	Testing of ANN	**0.0116**	**0.0166**	**0.0124**	**0.0090**	**0.0029**
	Training of AWNN	0.0007	0.0082	0.0008	0.0008	0.0023
	Testing of AWNN	**0.0011**	**0.001**	**0.0015**	**0.0017**	**0.0010**

* The random state of neural network for y_{NN} is varied from 1 to 5. The random state of neural network for w_{NN} and w_{TCAD} is fixed at 1.

cesses are highly chamber-dependent, and therefore, machine learning becomes inevitable in semiconductor industry, and the accurate prediction and guidance on future experiments using AWNN can save tremendous amount of cost and time. Fig. 15 and Fig. 16 further illustrate the results for the cases of *partition=30 and partition=45*, and it can be observed that the improvement is always persistent. In Table 3, we lists all MSE values for random states from 1 to 5.

It is worth to discuss more in-depth reasoning behind the effectiveness of AWNN model. Simply speaking, the reason for AWNN improvement lies in the fact that the TCAD model supplements the inaccuracy of conventional ANN, while the degree of the supplement is modeled by additional neural networks to describe its dependence on the input feature values. The effectiveness is especially prominent in two situations that are all very common in the application field of semiconductor manufacturing. The first situation is at the initial trial-and-error stage. In machine learning terminology, this is initial mining stage where we do not have much information on either what the objective function $Y(x)$ looks like or which portion of the searching space the optimal input parameters should be located. The second situation is when searching a vast sample space is inevitable, and thus many data points are not sampled leading to significant under-sampling. Adaptive weighting model uses the relevant weighting scheme, w_{NN} and w_{TCAD}, to resolve this problem, and the information regarding the unsampled sample space can be supplemented by TCAD model values.

In short, in many scenarios, including initial trial-and-error stage and sampling over a vast sample space, the collected dataset does not reflect the unsampled portion of the sample space. Therefore, the conventional model described by ANN, solely based on the limited training set information, is not accurate enough, resulting in test set prediction error. On the other hand, the AWNN model incorporates TCAD information that can somehow model this unsampled sample space. Some errors can indeed exist between the TCAD calculated values based on semiconductor physics and the real experimentally-sampled values by semiconductor fabrication. Nonetheless, with a reasonably accurate TCAD model, the advantages of incorporating TCAD in learning can

easily outperform the fact that deviation can occur between the TCAD model values and the real experiment values.

The additional advantage associated with the proposed AWNN model is that the AWNN model can be useful to supplement the model error between the conventional ANN output y_{NN} and the corresponding target value y_{exp}, during the training procedure. Since we are discussing training set error, the error is not due to the under-sampling problem as discussed the previous paragraphs. This error comes from the neural network modeling itself such as an insufficient neuron or hidden layer number, improper hyper-parameter setting, or insufficient epoch number. Certainly, more neuron can improve the ANN model accuracy, but the computational loading also increases. In a complex, large-scale problem, such as semiconductor processing illustrated in Fig. 13, the compromise inevitably has to be made between neuron number and CPU runtime. This is understandable since the conventional ANN prediction always has some error after training. This error can be large if the sample space is irregular and the problem is overly complex where ANN with reasonable computational demand can only approximate or fit the sample space Y(x) to some extent.

IV. CONCLUSIONS

In this paper, a newly proposed algorithm, adaptive weighting neural network (AWNN) has shown prominently more efficient performance as compared to traditional artificial neural network (ANN). The test results using ARDE as an example illustrate that the etching depth (t) of DRIE process, as a function of trench width and etching time, can be better predicted by AWNN. The MSE value is reduced from 0.19 to 0.01 for *partition*=15, from 0.02 to 0.002 for *partition*=30, and from 0.01 to 0.001 for *partition*=45, in the case of random state = 1. Varied random state does not affect the effectiveness of proposed AWNN model. This is evident in comparing the predicted etching depth and the corresponding mean square error (MSE) in the test data set with varied random state initialization in codes. The improvement of proposed AWNN framework is attributed to the fact that the TCAD compensates the error associated with the prediction using traditional neural networks. The weighting scheme between TCAD and machine learning tends to arrive at an optimized balance between physical models and statistical model.

Furthermore, the avantage of AWNN can be more prnounced if the sample space is very large where it is impossible to grasp the required amount of information using the limited sampled data points considering cost and time. Thus, the adaptive weighting scheme and the incorporation of TCAD values correct the prediction error of conventional neural networks, and the degree of correction depends on TCAD model accuracy, weighting (w_{NN} and w_{TCAD}) dependence on input features, the ratio between sampled and unsampled data point number, and the complexity of objective function, i.e. $Y(x)$, equivalently the complexity of the sample space. By the aspect-ratio dependent etching (ARDE) example, we can conclude that AWNN is a more generalized and also more optimized approach, which is effective in semiconductor process mining problem.

ACKNOWLEDGMENT

The authors would like to thank the technical support from National Applied Research Lab (NARL).

REFERENCES

[1] A. L. Tarca, V. J. Carey, X.-W. Chen, R. Romero, and S. Drăghici, "Machine learning and its applications to biology," *PLOS Comput. Biol.*, vol. 3, no. 6, p. e116, 2007.
[2] P. Larrañaga *et al.*, "Machine learning in bioinformatics," *Briefings Bioinf.*, vol. 7, no. 1, pp. 86–112, 2006.
[3] W.-Y. Lin, Y.-H. Hu, and C.-F. Tsai, "Machine learning in financial crisis prediction: A survey," *IEEE Trans. Syst., Man, C, Appl. Rev.*, vol. 42, no. 4, pp. 421–436, Jul. 2012.
[4] L. Yu, S. Wang, K. K. Lai, and F. Wen, "A multiscale neural network learning paradigm for financial crisis forecasting," *Neurocomputing*, vol. 73, nos. 4–6, pp. 716–725, 2010.
[5] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," presented at the SIGCOMM Workshop Mining Netw. Data, Pisa, Italy, 2006.
[6] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.
[7] P. A. Bromiley, N. A. Thacker, M. L. J. Scott, M. Pokrić, A. J. Lacey, and T. F. Cootes, "Bayesian and non-Bayesian probabilistic models for medical image analysis," *Image Vis. Comput.*, vol. 21, no. 10, pp. 851–864, 2003.
[8] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. NIPS*, 2012, pp. 2951–2959.
[9] V. Grau, A. U. J. Mewes, M. Alcaniz, R. Kikinis, and S. K. Warfield, "Improved watershed transform for medical image segmentation using prior information," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 447–458, Apr. 2004.
[10] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, pp. 452–459, May 2015.
[11] A. H. Neto and F. A. S. Fiorelli, "Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption," *Energy Buildings*, vol. 40, no. 12, pp. 2169–2176, 2008.
[12] F. Marozzo, D. Talia, and P. Trunfio, "A workflow management system for scalable data mining on clouds," *IEEE Trans. Services Comput.*, vol. 11, no. 3, pp. 480–492, May 2018.
[13] S. Iftikhar, M. Kamran, E. U. Munir, and S. U. Khan, "A reversible watermarking technique for social network data sets for enabling data trust in cyber, physical, and social computing," *IEEE Syst. J.*, vol. 11, no. 1, pp. 197–206, Mar. 2017.
[14] N. Jothi, N. A. A. Rashid, and W. Husain, "Data mining in healthcare—A review," *Procedia Comput. Sci.*, vol. 72, pp. 306–313, Dec. 2015, doi: 10.1016/j.procs.2015.12.145.
[15] Y. Li, Y. Zhang, X. Huang, H. Zhu, and J. Ma, "Large-scale remote sensing image retrieval by deep hashing neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 26, no. 2, pp. 950–965, Feb. 2018.
[16] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017.
[17] C. Sommer and D. W. Gerlich, "Machine learning in cell biology—Teaching computers to recognize phenotypes," *J. Cell Sci.*, vol. 126, pp. 5529–5539, Dec. 2013, doi: 10.1242/jcs.123604.
[18] A. Gosavi, "A reinforcement learning algorithm based on policy iteration for average reward: Empirical results with yield management and convergence analysis," *Mach. Learn.*, vol. 55, no. 1, pp. 5–29, 2004
[19] K. B. Irani, J. Cheng, U. M. Fayyad, and Z. Qian, "Applying machine learning to semiconductor manufacturing," *IEEE Expert*, vol. 8, no. 1, pp. 41–47, Feb. 1993.
[20] D. Ding, J. A. Torres, and D. Z. Pan, "High performance lithography hotspot detection with successively refined pattern identifications and machine learning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 11, pp. 1621–1634, Nov. 2011.
[21] J.-Y. Wuu, F. G. Pikus, and M. Marek-Sadowska, "Efficient approach to early detection of lithographic hotspots using machine learning systems and pattern matching," *Proc. SPIE, Design Manufacturability Through Design-Process Integr. V*, vol. 7974, p. 79740U, Apr. 2011.

[22] D. Duo, W. Xiang, J. Ghosh, and D. Z. Pan, "Machine learning based lithographic hotspot detection with critical-feature extraction and classification," in *Proc. IEEE Int. Conf. IC Design Technol.*, May 2009, pp. 219–222.
[23] D. Braha and A. Shmilovici, "Data mining for improving a cleaning process in the semiconductor industry," *IEEE Trans. Semicond. Manuf.*, vol. 15, no. 1, pp. 91–101, Feb. 2002.
[24] S.-K. Lee, P. Kang, and S. Cho, "Probabilistic local reconstruction for *k*-NN regression and its application to virtual metrology in semiconductor manufacturing," *Neurocomputing*, vol. 131, pp. 427–439, May 2014.
[25] J. Yu, "Semiconductor manufacturing process monitoring using gaussian mixture model and Bayesian method with local and nonlocal information," *IEEE Trans. Semicond. Manuf.*, vol. 25, no. 2, pp. 480–493, Aug. 2012.
[26] B. Kim, D. W. Kim, and G. T. Park, "Prediction of plasma etching using a polynomial neural network," *IEEE Trans. Plasma Sci.*, vol. 31, no. 6, pp. 1330–1336, Dec. 2003.
[27] A. Chatterjee, D. Croley, V. Ramamurti, and K.-Y. Chang, "Application of machine learning to manufacturing: Results from metal etch," in *Proc. 19th IEEE/CPMT Int. Electron. Manuf. Technol. Symp.*, Oct. 1996, pp. 372–377.
[28] V. M. Jimenez-Fernandez, C. Reyes-Betanzo, M. Angelica-Cerdan, Z. J. Hernandez-Paxtian, H. Vazquez-Leal, and A. Itzmoyotl-Toxqui, "Prediction of silicon dry etching using a piecewise linear algorithm," *J. Chin. Inst. Eng.*, vol. 36, no. 7, pp. 941–950, 2013.
[29] B. Kim *et al.*, "Modeling etch rate and uniformity of oxide via etching in a CHF_3/CF_4 plasma using neural networks," *Thin Solid Films*, vol. 426, pp. 8–15, Feb. 2003.
[30] B. Kim, M. Kwon, and S. H. Kwon, "Modeling of plasma process data using a multi-parameterized generalized regression neural network," *Microelectron. Eng.*, vol. 86, no. 1, pp. 63–67, 2009.
[31] B. Kim, D. W. Lee, and K. H. Kwon, "Prediction of etch microtrenching using a neural network," *J. Appl. Phys.*, vol. 96, no. 7, pp. 3612–3616, 2004.
[32] B. Kim, J. Bae, and B. T. Lee, "Modeling of silicon oxynitride etch microtrenching using genetic algorithm and neural network," *Microelectron. Eng.*, vol. 83, no. 3, pp. 513–519, 2006.
[33] 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 Trans. Semicond. Manuf.*, vol. 31, no. 2, pp. 315–322, May 2018.
[34] G. A. Susto, M. Terzi, and A. Beghi, "Anomaly detection approaches for semiconductor manufacturing," *Procedia Manuf.*, vol. 11, pp. 2018–2024, Sep. 2017, doi: 10.1016/j.promfg.2017.07.353.
[35] K. Nakata, R. Orihara, Y. Mizuoka, and K. Takagi, "A comprehensive big-data-based monitoring system for yield enhancement in semiconductor manufacturing," *IEEE Trans. Semicond. Manuf.*, vol. 30, no. 4, pp. 339–344, Nov. 2017.
[36] 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 Syst. Appl.*, vol. 15, no. 2, pp. 123–132, 1998.
[37] B. Kim, D. Kim, J. Park, and S.-S. Han, "Prediction of surface microtrenching by using neural network," *Current Appl. Phys.*, vol. 7, no. 4, pp. 434–439, 2007.
[38] J. W. Coburn and H. F. Winters, "Conductance considerations in the reactive ion etching of high aspect ratio features," *Appl. Phys. Lett.*, vol. 55, pp. 2730–2732, Dec. 1989.

Authors' photographs and biographies not available at the time of publication.