Received July 15, 2020, accepted August 3, 2020, date of publication August 5, 2020, date of current version August 17, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.3014470*

# Improvement of TCAD Augmented Machine Learning Using Autoencoder for Semiconductor Variation Identification and Inverse Design

**KASHYAP MEHTA[1], SOPHIA SUSAN RAJU[1], MING XIAO[2], BOYAN WANG[2], YUHAO ZHANG[2], (Member, IEEE), AND HIU YUNG WONG[1], (Senior Member, IEEE)**

[1]Electrical Engineering Department, San Jose State University, San Jose, CA 95192, USA
[2]Center for Power Electronics Systems, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

Corresponding author: Hiu Yung Wong (hiuyung.wong@sjsu.edu)

This work was supported in part by the San Jose State University College of Engineering (SJSU COE) New Faculty Startup Fund.

**ABSTRACT** A machine learning (ML) model by combing two autoencoders and one linear regression model is proposed to avoid overfitting and to improve the accuracy of Technology Computer-Aided Design (TCAD)-augmented ML for semiconductor structural variation identification and inverse design, without using domain expertise. TCAD-augmented ML utilizes TCAD simulations to generate sufficient data for ML model development when experimental data are inadequate. The ML model can then be used to identify semiconductor structural variation for given experimental electrical measurements. In this study, the variation of layer thicknesses in the p-i-n diode is used as a demonstration. An ML model is developed to predict the diode layer thicknesses based on a given Current-Voltage (*IV*) curve. Although the variations of interest can be incorporated easily in TCAD simulations to generate ML training data, the TCAD-augmented ML model generally is overfitted and cannot predict the variations in experiment well due to hidden variables which also alters the *IV* curves. We show that by using an autoencoder, this problem can be solved. To verify the effectiveness, another set of TCAD simulation data is generated with hidden variables (dopant concentration variation) to emulate experimental data. Testing on the second set of data shows that the proposed model can avoid overfitting and has up to 15 times improvement in accuracy in thickness prediction. Moreover, this model is used successfully to perform inverse design and can capture an underlying physics that cannot be described by a simple physical parameter.

**INDEX TERMS** Autoencoder, defect identification, inverse design, machine learning, TCAD simulation, variability.

## I. INTRODUCTION

Machine learning (ML) has been widely used in semiconductor fabrication to improve manufacturing yields at various manufacturing stages [1]–[5]. They have been used for tool anomaly detection [1], [2], lithographic hotspot detection [3], and optical proximity correction [4], etc.. These problems usually have a large amount of data for effective ML model development. Semiconductor defect and yield troubleshooting have also been benefited by using ML to speed up classification and data analysis on the final wafer map [5]. However, we believe the most efficient and cost-effective defect identification is to use ML to pinpoint the structural defect and variation sources using the final device electrical characteristics as inputs. The electrical characteristics can be Current-Voltage (*I*-*V*) curves, Capacitance-Voltage (*C*-*V*) curves, or others. The result can then be confirmed with physical failure analysis such as Transmission Electron Microscopy (TEM), which is very expensive in terms of cost and turn-around time. If the ML model is good enough, the need for expensive physical failure analysis can be alleviated to enable more rapid and inexpensive development of semiconductor technologies.

However, to the best of our knowledge, this ideal strategy has not been successfully deployed as can be seen in the lack of relevant publications. We believe this is because of the lack

The associate editor coordinating the review of this manuscript and approving it for publication was Tao Zhou.

of data to train a successful ML model. Firstly, in the experiment, it is difficult to accurately and confidently associate defects, such as oxide weak spots and traps concentrations, and variations such as layer thicknesses, contact resistance, etc. to electrical characteristics. Secondly, a large amount of training data is required to develop a good ML model However, for immature technology, there not enough wafers; while for mature technology, there not enough defective dies.

Therefore, in [6], we have proposed to use Technology Computer-Aided Design (TCAD) to generate defective and variation data for ML. A Similar idea was also proposed by another group in [7]. TCAD models are calibrated to experiment. Defects and variations can be introduced in the TCAD simulation to generate enough data for ML. The ML model can then be used to pinpoint the sources of defects and variations for any given experimental curve. In [8], we showed that the TCAD generated ML model cannot be used on experimental data directly because experimental data contains more variations (including noise) than in the TCAD data used to train the ML model. This is equivalent to the overfitting of the ML model when it is trained by TCAD data. A methodology was proposed to reduce overfitting but it was difficult to verify the effectiveness because of the small experimental data size and lack of confidence in the exact physical values in the experiment.

In this article, we propose a new model based on autoencoder and linear regression to make TCAD trained ML model more robust when it is presented with data with hidden variations. The TCAD training data has one set of variations of interest (layer thicknesses in p-i-n diode). Another set of TCAD data with a different type of variations (doping concentration) are then used for validation. This validation dataset emulates the experimental data with additional variations that the ML has not seen before. TCAD dataset is used for validation because the additional variations can be more precisely controlled and many more data can be generated compared to experiments. Moreover, this new model requires no domain expertise. This is important to minimize data preprocessing for more complex systems.

The paper is organized as follows. In Section II, we discussed how the datasets are generated, the setup in TCAD, and the characteristics of the datasets and their implications. In Section III, overfitting of TCAD augmented machine learning is presented. We then discuss how autoencoder based algorithm can effectively avoid overfitting. In Section IV, the autoencoder based algorithm is used for reverse engineering and its performance is presented.

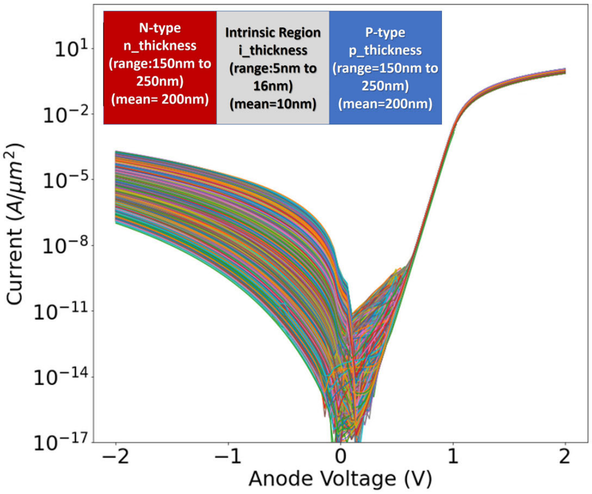

**FIGURE 1.** ***Set 1 IV*s of the 2000 devices simulated. The concentration of the n$^+$-doped and p$^+$-doped layers are both $10^{20}$ cm$^{-3}$ and the donor concentration of the unintentionally-doped layer (i-layer) is $10^{17}$ cm$^{-3}$.**

## II. TCAD SIMULATION AND DATASETS DISCUSSION

TCAD simulations are performed using TCAD Sentaurus with the structures created by SProcess [9] and electrical simulations performed by SDevice [10]. There are two sets of simulations. The first set (*Set 1*) is for ML model training with about 2000 1D p-i-n diodes with layer thicknesses being varied independently and uniformly (Fig. 1). The means and ranges of the thicknesses of n$^+$-doped ($t_n$), intrinsic ($t_i$), and p$^+$-doped ($t_p$) layers are showed in Fig. 1. The doping concentrations of n$^+$-layer ($N_n$), i-layer ($N_i$) and p$^+$-layer ($N_p$) are fixed at $10^{20}$cm$^{-3}$, $10^{17}$cm$^{-3}$ and $10^{20}$cm$^{-3}$, respectively. Fig. 1 also shows the simulated *IV* curves. These settings suggest that the ML model trained by *Set 1* will understand how *IV* is affected by layer thicknesses but has no knowledge of how the doping will affect the *IV*. A good ML model should capture the features in the *IV* curves that are correlated to the thicknesses and should not be affected by any new features introduced by new variables (e.g. doping concentration).

The second set (*Set 2*) is for testing the ability of the trained ML model to predict the thickness when unseen variations (in this case, it is doping concentrations) are introduced. This is to emulate the experimental data which typically contains multiple variation sources. *Set 2* also contains about 2000 *IV* curves of p-i-n diodes. The diode structure has constant thicknesses ($t_n$ = 200nm, $t_i$ = 10nm, and $t_p$ = 200nm) but $N_n$, $N_{i,}$ and $N_p$ are varied as shown in Fig. 2. This set of data emulates the experimental data to be tested because the thicknesses are the same as the mean in *Set 1* while it has 3 extra doping concentration variables that the ML model trained by *Set 1* has not seen before. If the ML model trained by *Set 1* is effective, it should predict the layer thicknesses of the *Set 2* data well.

In the simulations, essential physics models are turned on, including Fermi-Dirac statistic, doping dependent and high field saturation models for carrier mobilities, Schottky-Reed-Hall Recombination (SRH) and non-local Band to Band tunneling (BTBT). 80-bit ExtendedPrecision is used to avoid noisy reversed curves. Poisson equations as well as the electron and hole continuity equations are solved self-consistently. The anode voltage ($V_a$) is swept from -2V to 2V and anode current ($I$) is measured, as shown in Figs. 1 and 2.

From device physics [11], it is known that the reverse current, $I_{off}$, at $V_a$ < 0V depends exponentially on $t_i$ and

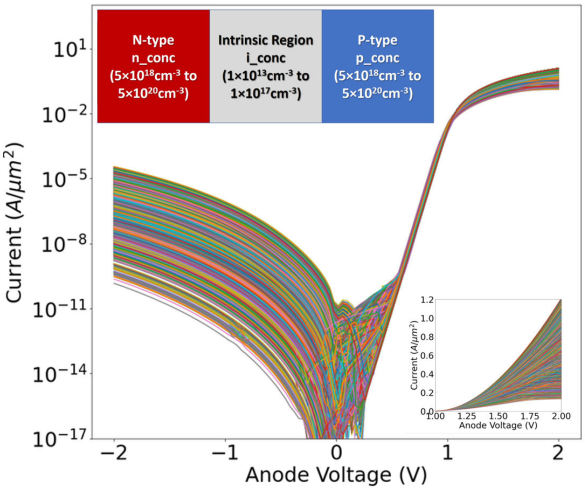

**FIGURE 2.** ***Set 2 IV*****s of the 2000 devices simulated. The thicknesses of n+/i/p+ doped layers are 200nm/10nm/200nm. The upper left inset shows the structure and the range of variation. The lower inset shows the forward *IV*s in linear scale.**

forward current, $I_{on}$, at $V_a > V_{bi}$, depends linearly on $1/t_n$, $1/t_p$, $N_n$ and $N_p$, where $V_{bi}$ is the built-in potential. Since the ML model trained by *Set 1* does not know the existence of possible concentration variations, the most *naïve model* will treat the effect of the $N_n$ and $N_p$ concentration variations on *IV* in *Set 2* as a result of $t_n$ and $t_P$ variations. Since $N_n = N_p = 10^{20}\text{cm}^{-3}$ in *Set 1* and *Set 2* has $N_n$ and $N_p$ variations from $5 \times 10^{18}\text{cm}^{-3}$to $5 \times 10^{20}\text{cm}^{-3}$, the *naïve model* should predict the $t_n$ and $t_p$ thicknesses to be between $200\text{nm}\times(10^{20}\ \text{cm}^{-3}/5 \times 10^{20}\text{cm}^{-3}) = 40\text{nm}$ and $200\text{nm}\times(10^{20}\ \text{cm}^{-3}/5 \times 10^{18}\text{cm}^{-3}) = 4000\text{nm}$, or predict $t_n + t_p$ to be between 80nm and 8000nm. A well-trained model, on the other hand, should be able to identify the features from the *I-V* curves and predict all *Set 2* data to have $t_n = t_p = 200\text{nm}$ or $t_n + t_p = 400\text{nm}$. However, since ML is a statistical tool, it is expected to have variations and its performance should be gauged by the means and ranges of the predictions of $t_n$, $t_i$, $t_p$, and $t_n + t_p$.

## III. MACHINE LEARNING MODELS

### A. DATA PREPARATION

Unlike other applications of ML in TCAD ([7], [12], [13]), domain expertise (i.e. knowledge in semiconductor physics and fabrication) is not used in our approach. This means that physical quantities such as sub-threshold slope, transconductance, turned-on voltage, currents at particular voltages (such as currents at $V_a = -2\text{V}$ or 2V, which are the typical definitions of OFF/ON currents) and other quantities will not be extracted from the raw data (*IV* curves).

Each *IV* curve is discretized into 102 current values ($I_0$ - $I_{101}$). Since the currents have almost a range of 16 orders of magnitude (Fig. 1), both the currents, *I* and logarithmic values of the currents, *log(I)* are used as the input futures, to best capture the variations in all regions of the curves. The outputs are $t_n$, $t_{i,}$ and $t_p$. 90% of the curves are used for training ($X_{train}$)and 10% are for validation ($X_{test}$).

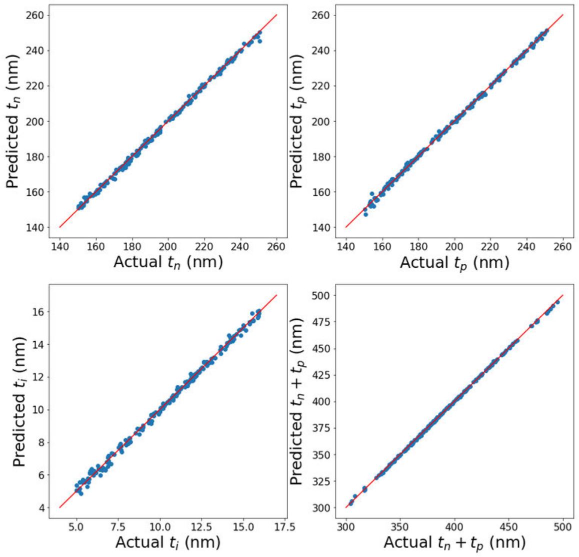

**FIGURE 3.** **The 4 sub-plots show the prediction of *Set 1* testing data thicknesses by using *Model-R-1.***

The *I* and *log(I)* are then scaled based on the means and standard deviations of the training data using the following equations

$$X_{trainscaled} = \frac{X_{train} - \mu_{train}}{\sigma_{train}} \tag{1}$$

$$X_{test\,scaled} = \frac{X_{test} - \mu_{train}}{\sigma_{train}} \tag{2}$$

where $\mu_{train}$ and $\sigma_{train}$ are the mean and the standard deviation of the training data, respectively.

### B. LINEAR REGRESSION AND OVERFITTING

Linear Regression is one of the basic supervised machine learning algorithm. The basic linear regression equation is represented as:

$$\hat{y} = \beta_0 + (\beta_1 * X_{input}) \tag{3}$$

where $\beta_0$ is the linear regression intercept, $\beta_1$is the coefficient of linear regression, $X_{input}$ is the input feature and $\hat{y}$ is the predicted output. If the input consists of multiple features then the model is called a multiple linear regression model, where $\beta_0$, $X_{input}$, and $\hat{y}$ are vectors and $\beta_1$is a matrix.

Firstly, training data in *Set 1* is used to train an ML model (dubbed as *Model-R-1*) using linear regression in the Scikit-learn library [14]. Here $t_n$, $t_i$, and $t_p$ are $\hat{y}$ and $X_{train\_scaled}$ are $X_{input}$. This model is then validated by the test data in *Set 1*. Fig. 3 shows that the model has an excellent prediction on the test data $t_n$, $t_i$, and $t_p$. However, when the model is used to predict the thicknesses of the devices in *Set 2*, the performance is very bad (Fig. 4). It may wrongly predict thickness as large as $10^6$nm or even predict negative thicknesses (e.g. $t_n + t_p\ <$ -50000nm). This indicates the *Model-R-1* trained by *Set 1* is probably over-fitted in the sense that when there are extra variables (e.g. variation of doping concentration in this case), the *Model-R-1* is confused by the

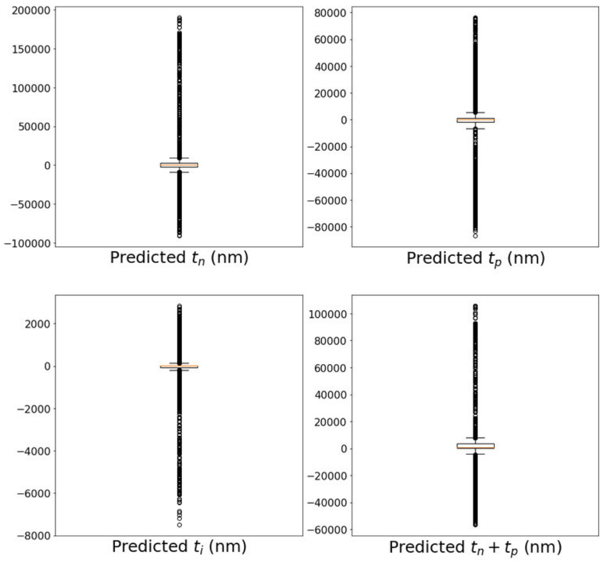

**FIGURE 4.** **Box plots of the thicknesses prediction (in nm) of *Set 2* by using the model trained by *Set 1 (Model-R-1)*.**

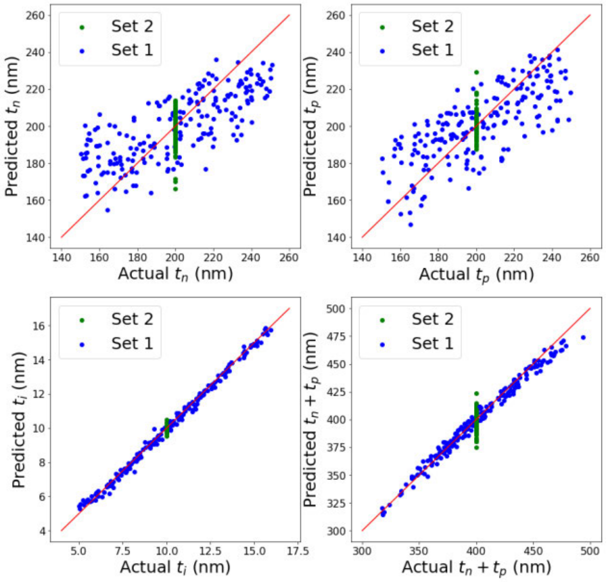

**FIGURE 5.** **The 4 sub-plots show the prediction of *Set 1* and *Set 2* testing data thicknesses by using *Model-R-1-2*. Blue and green dots are data points from *Set 1* and *Set 2* respectively.**

new features in *IV* curves. We also want to note that even if *Model-R-1* does not have the knowledge of the existence of doping concentration, if it is as good as the *naïve model* discussed in Section II, it should predict $t_n + t_p$ to be between in the range of 80 nm to 8000 nm. However, due to overfitting, it performs much worse than the *naïve model*.

It is also useful to investigate how the model will improve if it is allowed to ''see'' *Set 2* (i.e. to be aware of the existence of doping concentration variation) during training. *Set 1* and *Set 2* are combined and 90% of the data are used to train a new model (dubbed as *Model-R-1-2*). 10% of the combined dataset are used for validation. Note that the outputs are still only $t_n$, $t_{i,}$ and $t_p$. Doping concentration is not used in the training. Therefore, the new model still does not have information on how doping concentration will modify the *IV* curves. Fig. 5 shows the validation result of *Model-R-1-2* by using the testing data of the combined dataset. The model now gives much more reasonable predictions on $t_n$, $t_{i,}$ and $t_p$ for *Set 2* curves, which are supposed to be 200 nm, 10 nm, and 200 nm respectively. Although it cannot predict them exactly correctly, it has learned that the variation in *IV* may be caused by some hidden variations. It is also interesting to see that the thicknesses prediction of the data from *Set 1* has more variations than the prediction by *Model-R-1* but the variation ranges are similar to those of *Set 2*. If it is allowed to make an analogy to how a human engineer learns, we may say now the engineer is less confident on the engineer's ability to predict thicknesses based on *IV* curves because data from *Set 2* has trained the engineer that even for the same thickness, *IV* curves are not the same due to hidden variables. Therefore, there is less overfitting.

Although by combing *Set 1* (TCAD data) and *Set 2* (emulated experimental data), a better model is obtained, this defeats the purpose of using TCAD data to augment machine learning. This is because it still requires a large amount of data from the experiment (*Set 2*) in the training. Moreover, and most importantly, new variations (including new defects) can be introduced in new experiments. As a result, the trained model cannot be used in general. Indeed, we will confirm this in Section IIID when inverse design results are discussed.

### C. AUTOENCODER FOR VARIATION IDENTIFICATION

To solve the aforementioned problem, we proposed to use autoencoder. Autoencoder is a special kind of unsupervised deep learning feed-forward neural network which has encoder and decoder, developed using a conventional neural network used for transforming / encoding input data and retrieving back / decoding to the original form. Autoencoders are majorly divided into two main categories namely under complete and overcomplete autoencoder. If the number of neurons in the hidden layer is less the number of neurons in the input layer then it is called under complete autoencoder. If the number of neurons is higher in the hidden layer as compared to the input layer then it is called overcomplete autoencoder. One of the properties of autoencoder includes the number of neurons in the input layer is equal to the number of neurons in the output layer. Autoencoders are majorly used as a regularization technique to overcome overfitting of the model as well as for denoising the signal.

Autoencoder is known to be able to perform efficient coding for signals. In other words, it can find the dominant underlying features of the signals by minimizing the input and output difference [15]. Therefore, it can be used to discover the underlying physics without using domain expertise and has been used in mechanical structure layout optimization [16].

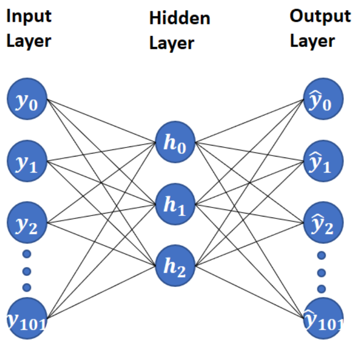

**FIGURE 6.** Autoencoder network used. The input features can be either the original currents ($y_i = I_i$) or their logarithmic values ($y_i = log(I_i)$).

Fig. 6 represents undercomplete autoencoder wherein the input layer consists of 102 current values in the form of neurons ranging from ($y_0$to $y_{101}$), while hidden layer consists of 3 neurons ($h_0$, $h_1$and$h_2$). The output of the hidden layer is the predicted 102 input neuron values ranging from ($\hat{y}_0$ to $\hat{y}_{101}$). The three hidden layer neurons are able to reconstruct 102 input features using trained autoencoder with the help of weights and biases of autoencoder.

The inputs to the autoencoder are either the original current values, $y_i = I_i$ or their logarithmic values, $y_i = log(I_i)$. The outputs are the corresponding predicted values, $\hat{y}_i = \hat{I}_i$ or $log(\hat{I}_i)$. The hidden layer uses RELU for activation. Therefore, this is not a linear autoencoder and is not the same as Principal Components Analysis (PCA) [17]. Adam algorithm is used for optimization with 500 epochs. The performance metric is given by the Mean Squared Error (MSE) defined by

$$MSE = \frac{1}{n}\sum\nolimits_{i=0}^{n-1}(y_i - \hat{y}_i)^2 \tag{4}$$

where $n = 102$. Different numbers of hidden layers are tested (5, 3, and 1). Since only 3 variables are used to generate *Set 1*, it is expected that only 3 neurons are enough to capture the dominant features. Therefore, the middle layer has only 3 neurons. Fig. 7 shows that 1-, 3- and 5- hidden layers give similar loss values. Therefore, 1 hidden layer is used to avoid overfitting.

To construct the proposed model, two autoencoders in Fig. 6 are trained using the training data in *Set 1*. One is trained using $y_i = I_i$ as the input features (*Model-Auto-Linear-1*) and the other is trained using $y_i = log(I_i)$ as the input features (*Model-Auto-Log-1*). Each of them has 3 hidden layer outputs ($h_{L0}$, $h_{L1}$, $h_{L2}$, $h_{G0}$, $h_{G1}$, $h_{G2}$). Linear regression is then used to correlate these 6 hidden layer outputs to $t_n$, $t_{i,}$ and $t_p$. They are then combined to form the model (*Model-Auto-Combined-1*) showed in Fig. 8.

Fig. 8 represents the overall architecture of the machine learning model developed, that transforms linear and log scaled IV curve using autoencoder and predicts thickness using hidden layer neuron of the autoencoder. The scaler transformation of the IV curve to the autoencoder input in

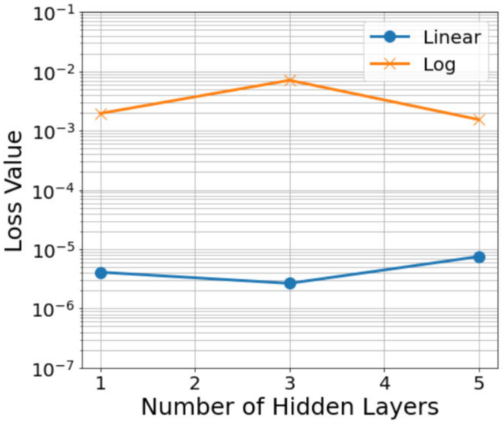

**FIGURE 7.** Loss values of the autoencoder as a function of the number of hidden layers for $I_i$ (linear) and $log(I_i)$ (log) inputs.

*Model-Auto-Linear-1* model is obtained using the following equation:

$$\sigma U_{Lj} = \frac{I_j - \mu_{train}}{train} \tag{5}$$

where $\mu_{train}$ and $\sigma_{train}$ is the mean and standard deviation of all the training data points. Similarly, for *Model-Auto-Log-1* the equation is given as follows:

$$\sigma U_{Gj} = \frac{log(I_j) - \mu'_{train}}{'_{train}} \tag{6}$$

where $\mu'_{train}$ and $\sigma'_{train}$ is the mean and standard deviation of all the logarithmic training data points. The output of the hidden layer is obtained by following equation, where $\sigma$ represents RELU activation function, while $\mathrm{w}_{ij}$ and $w'_j$ represents weight and biases for j$^{\mathrm{th}}$neuron.

$$h_j = \sigma\left(\sum_{i=0}^{101}(\mathrm{y}_i * \mathrm{w}_{ij}) + w'_j\right)$$

As represented, the output of the hidden layer for *Model-Auto-Linear-1* and *Model-Auto-Log-1* are merged and used to train the linear regression with thickness as labels which is represented as below:

$$\begin{aligned} y_{pred} &= \beta_0 + (\beta_1 * h_{L0}) + (\beta_2 * h_{L1}) + (\beta_3 * h_{L2}) \\ &\quad + (\beta_4 * h_{G0}) + (\beta_5 * h_{G1}) + (\beta_6 * h_{G2}) \end{aligned}$$

where $\beta's$ represent the linear regression intercept and coefficients, $y_{pred}$ is the predicted n, p and i-thickness labels.

*Model-Auto-Combined-1* is tested using the testing data from *Set 1*. Fig. 9 shows that the training model can predict the testing data well and achieve similar performance as the linear regression model trained by both *Set 1* and *Set 2* data in Fig. 5. This means that *Model-Auto-Combined-1,* without being exposed to *Set 2*, is able to identify the features related to thicknesses and probably will not be confused if there are hidden variables that will alter the *IV* curves. To confirm this, this model is used to predict *Set 2* data which has $t_n$, $t_{i,}$ and $t_p$, being 200 nm, 10 nm, and 200 nm respectively.

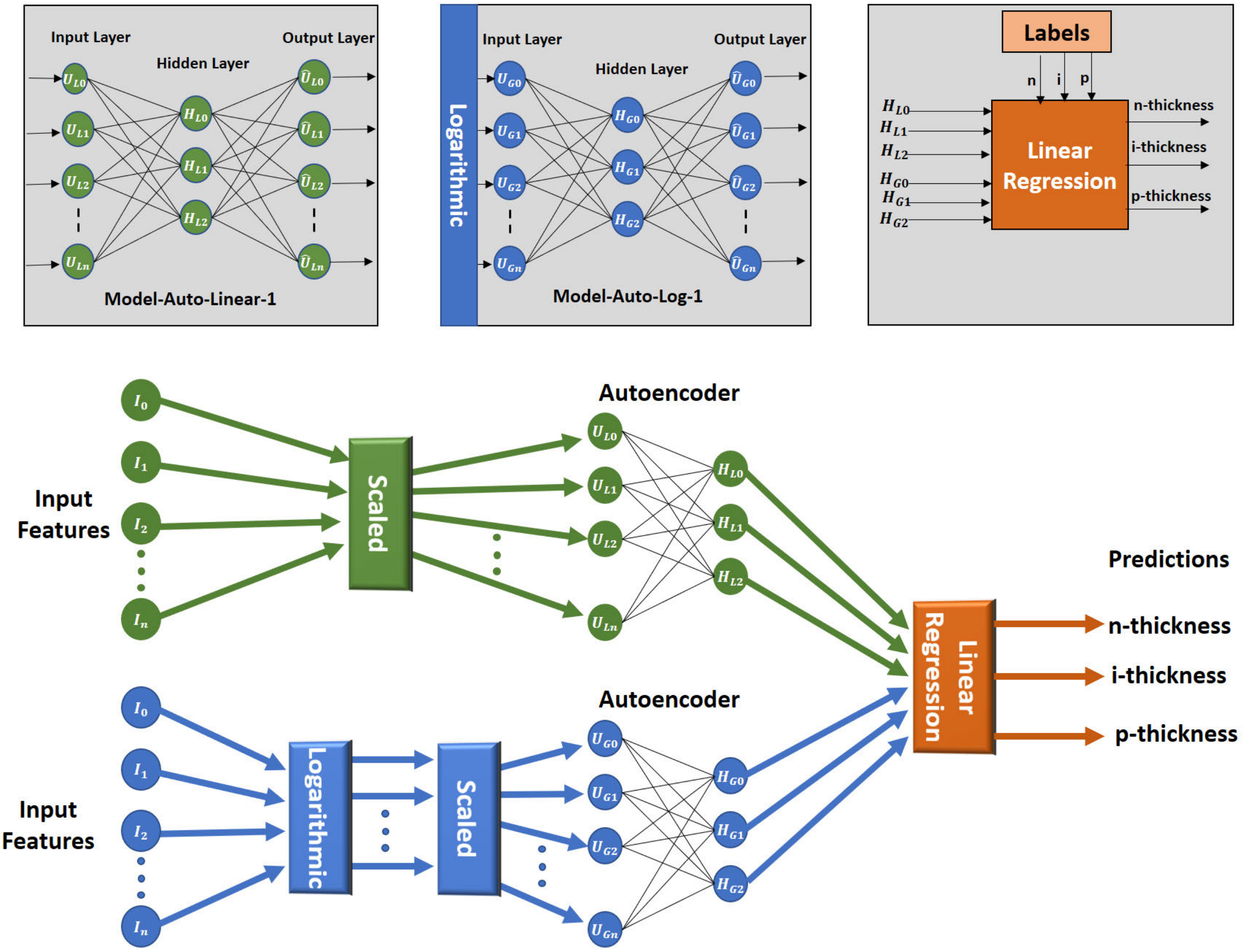

**FIGURE 8.** Autoencoder model (*Model-Auto-Combined-1*) used to predict $t_n$, $t_i$ and $t_p$based on IV curves. Top: The networks used for training the various components of *Model-Auto-Combined-1*. Bottom: The final model used for the prediction of layer thicknesses.

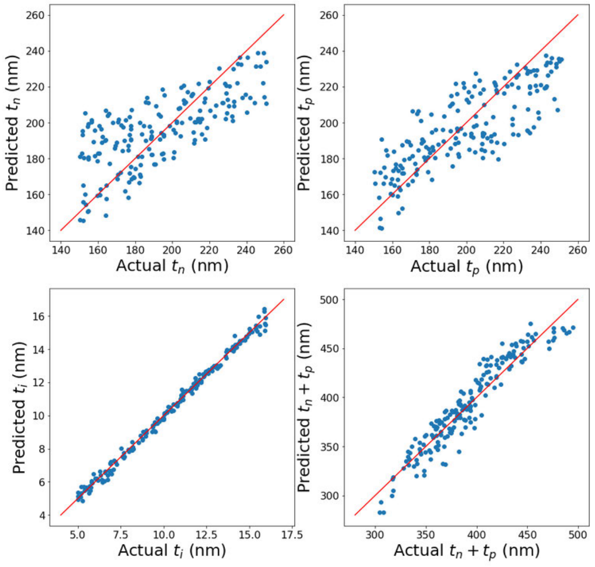

**FIGURE 9.** The 4 sub-plots show the prediction of *Set 1* testing data thicknesses by using *Model-Auto-Combined-1*.

As shown in Fig. 10, the performance of *Model-Auto-Combined-1* has a substantial improvement when compared to *Model-R-1* (Fig. 4). It predicts the averages of $t_i$ and $t_n + t_p$ in the *Set 2* data to be 12.9 nm and 410 nm, respectively, which are very closed to the actual values.

*Table 1* shows a summary of the performance statistics of *Set 2* data layer thicknesses prediction by various algorithms. One can see that *Model-Auto-Combined-1* not only can avoid the spurious thickness prediction in *Model-R-1* but also can predict the actual thicknesses with high accuracies. It predicts all ~2000 $t_n$+ $t_p$ values in *Set 2* to be between 101 nm and 598 nm (i.e. range = 497nm), while the *naïve model* predicts $t_n$ + $t_p$ values in *Set 2* to be between 80nm and 8000nm (i.e. range = 7920nm). Therefore, in terms of prediction range, *Model-Auto-Combined-1* is 15 times better than in the *naïve model*. Moreover, 97% of the thicknesses predicted by *Model-Auto-Combined-1* lie between 240 nm and 560 nm (i.e. +/- 40% of the actual value) (Fig. 10 and Table 1). The mean of the *naïve model* is also skewed severely and depends on how *Set 2* doping is randomly generated, which does not occur in *Model-Auto-Combined-1*.

**TABLE 1.** Summary of the performance statistics of *set 2* data layer thicknesses prediction by various algorithms.

| | *Actual Value* | | *Model-R-1* | | *Naïve Model*+ | *Model-Auto-Combined-1* | |
|---|---|---|---|---|---|---|---|
| | *i* | *n+p* | *I* | *n+p* | *n+p* | *i* | *n+p* |
| **Mean (nm)** | 10 | 400 | -174 | 1833 | 158 | 13 | 410 |
| **Standard Deviation (nm)** | 0 | 0 | 1335 | 26868 | 280 | 3 | 72 |
| **Minimum (nm)** | 10 | 400 | -7496 | -56728 | 80 | 8 | 101 |
| **Maximum (nm)** | 10 | 400 | 2826 | 105625 | 8000 | 21 | 598 |
| **Within +/- 40% of the target** | N/A | N/A | 22% | 25% | 19% | 65% | 97% |
| **Range (max-min) (nm)** | 0 | 0 | 10322 | 162353 | 7920 | 13 | 497 (15X better than *Naïve* model) |

+Naïve model is only for n+p layer thickness predication based on $I_{ON}$. Therefore, i-layer thickness is not included. The statistics are calculated by assuming the doping concentration are distributed uniformly in the *Set 2* data in Fig. 2.

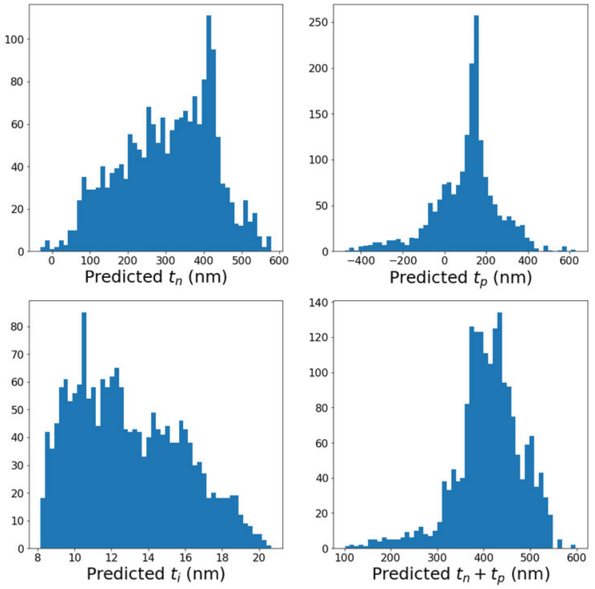

**FIGURE 10.** The 4 sub-plots show the histograms of the layer thickness prediction of *Set 2* data using *Model-Auto-Combined-1.*

For the predictions of $t_n$ and $t_p$, they are not as accurate as desired. However, it is still much better than *Model-R-1*. This is also understandable because, in *Set 1*, n and p regions are almost indistinguishable in device physics because they have the same doping concentration ($N_n = N_p = 10^{20}\text{cm}^{-3}$). The only difference is the electron and hole mobilities. However, this information does not have a strong signal in the *IV* curves. In [6], we showed that it is possible to distinguish the n and p regions in high order linear regression but it requires the training dataset containing different doping concentrations. Therefore, it is already expected that *Model-Auto-Combined-1* does not predict $t_n$ and $t_p$ individually well.

## IV. AUTOENCODER FOR REVERSE ENGINEERING

To test the usefulness of *Model-Auto-Combined-1*, it is used to perform reverse engineering. An arbitrary *IV* curve is manually drawn with Fig. 1 as the background. This imitates a device designer is trying to sketch a desired *IV* curve to an integration/process engineer but sketches it so that it is within the capability of the known fabrication process. It is then discretized and fed into the *Model-Auto-Combined-1* (Fig. 8) to find the corresponding $t_n$, $t_{i,}$ and $t_p$. The result is $t_n$ = -768 nm, $t_i$ = 13.7 nm and $t_p$ = 2700.8 nm and thus $t_n + t_p$ =1932 nm. As mentioned earlier, the physical distinction between n- and p-regions is very small. So, it is not surprising to get negative $t_n$. A structure with $t_i = 13.7$nm and $t_n + t_p$ =1932 nm (by setting $t_n = 1000$nm and $t_p = 932$nm) is thus constructed and simulated in TCAD and compared to the hand-drawn curve as shown in Fig. 11. The TCAD simulated *IV* emulates a device fabricated with the dimension predicted by *Model-Auto-Combined-1*.

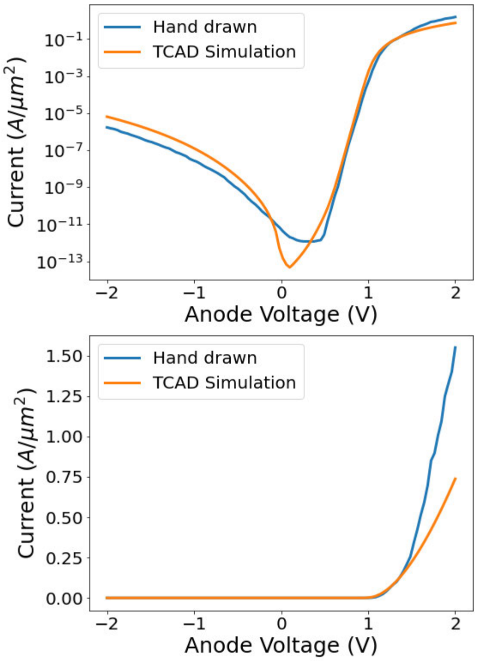

**FIGURE 11.** The hand drawn *IV* curve and TCAD simulated *IV* curve using the structure predicted by *Model-Auto-Combined-1.* Top: in log scale. Bottom: in linear scale.

As shown in Fig. 11, the simulated *IV* has very similar reversed leakage current as the drawn *IV*. It only has about 4 times difference in a large portion of the negative

**TABLE 2.** Predicted thicknesses of the hand drawn curves in fig. 12.

| | Model-R-1 | | | | Model-Auto-Combined-1 | | | |
|---|---|---|---|---|---|---|---|---|
| | i (nm) | n+p (nm) | n (nm) | p (nm) | i (nm) | n+p (nm) | n (nm) | p (nm) |
| HD0 | 76308 | -312542 | -533545 | 221002 | 10 | 185 | -182 | 367 |
| HD1 | 1956 | -484684 | -320278 | -164406 | 13 | 1089 | -164 | 1252 |
| HD2 | -65720 | 310045 | 1162207 | -852161 | 11 | 1132 | 12 | 1120 |
| HD3 | -7815 | 354260 | 847152 | -492892 | 12 | 1032 | -28 | 1060 |
| HD4 | -17872 | 158037 | 1782094 | -1624057 | 7 | 1350 | -342 | 1692 |
| HD5 | -3530 | 72553 | 282827 | -210274 | 18 | 1306 | -283 | 1589 |
| HD6 | 24834 | -13535 | 149291 | -162826 | 6 | 1153 | -154 | 1307 |
| HD7 | 120715 | -1506786 | -2725909 | 1219123 | 7 | 1056 | -224 | 1280 |
| HD8 | -76604 | 6123 | 703635 | -697512 | 12 | 1285 | -285 | 1570 |
| HD9 | 69926 | -485572 | -972805 | 487233 | 13 | 1249 | -287 | 1536 |
| HD10 | -2232 | 396750 | 1732187 | -1335436 | 13 | 925 | 240 | 685 |
| HD11 | 11113 | -15255 | -264677 | 249422 | 3 | -1052 | 900 | -1952 |
| HD12 | 97565 | -169061 | 641431 | -810492 | 11 | 1195 | -174 | 1369 |

$V_a$ regime. Note that the model was trained by *Set 1* which has 4 orders of magnitude difference in leakage current (Fig. 1).

Moreover, in the forward region (positive $V_a$), they match very well up to 1.5V. It does not match well for $V_a > 1.5$V. However, such discrepancy is expected. This is because it is not possible for the simple diode to have a shape like the hand-drawn one in the forward bias. The model actually made a more reasonable choice to match the current at $V_a < 1.5$V because it is more physical to have a smooth transition in the $V_{bi}$ neighborhood. Indeed, this is what the naïve model or a regular engineer will not be able to do as they will typically choose $t_n + t_p$ to match the current at $V_a = 2$V. Therefore, the model proposed is able to capture the underlying hidden physics that is not explicitly used and cannot be easily done through feature extraction with domain expertise.

As a comparison, *Model-R-1* (i.e. the model trained by only *Set 1*) is also used to perform reverse engineering to find the thicknesses of the hand-drawn curve and the results are $t_n = -2.2 \times 10^6$nm, $t_i = -77035$ nm and $t_p = -1.2 \times 10^6$nm and it is impossible to construct a device with negative $t_i$.

Then *Model-R-1-2* (trained by both *Set 1* and *Set 2*) is also used and the results are $t_n = 73402$ nm, $t_i = 3822$ nm and $t_p = 244277$ nm. The result is also obviously bad because all thicknesses are too large and will give very low forward and reverse currents. This confirms that adding experimental data (emulated by *Set 2*) in the training is not good enough for predicting future new datasets because new variations are introduced in the hand-drawn curve (i.e. noise). Therefore, our proposed model is more suitable to improve TCAD-augmented ML.

To further confirm the robustness of *Model-Auto-Combined-1* for reverse engineering, another 13 curves (*HD0 – HD12*) are drawn and showed in Fig. 12. The family of curves has the following characteristics. 1) Due to hand-drawn, it is very noisy. 2) It has 5 orders of magnitude difference in maximum leakage (larger than that in *Set 1*). 3) In

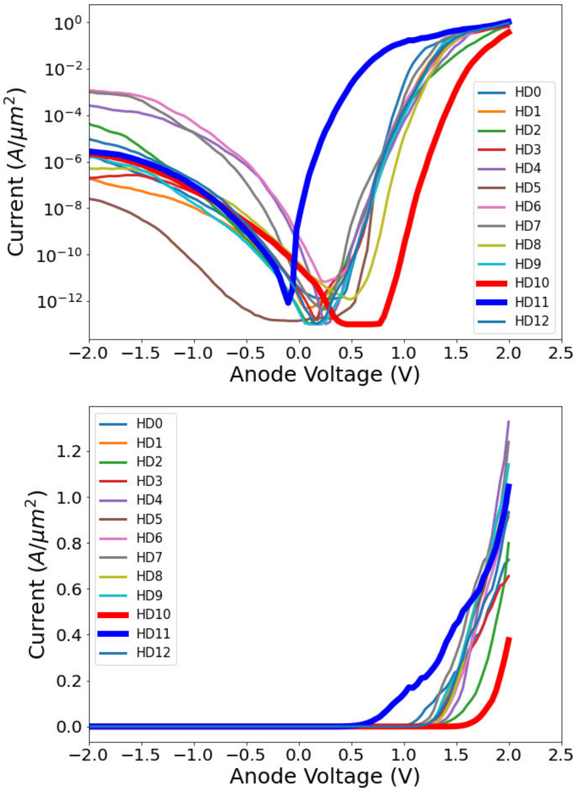

**FIGURE 12.** Another 13 hand-drawn *IV* curves for testing the robustness of performing reverse engineering. *HD10* and *HD11* are highlighted for their huge lateral shifts. Top: in log scale. Bottom: in linear scale.

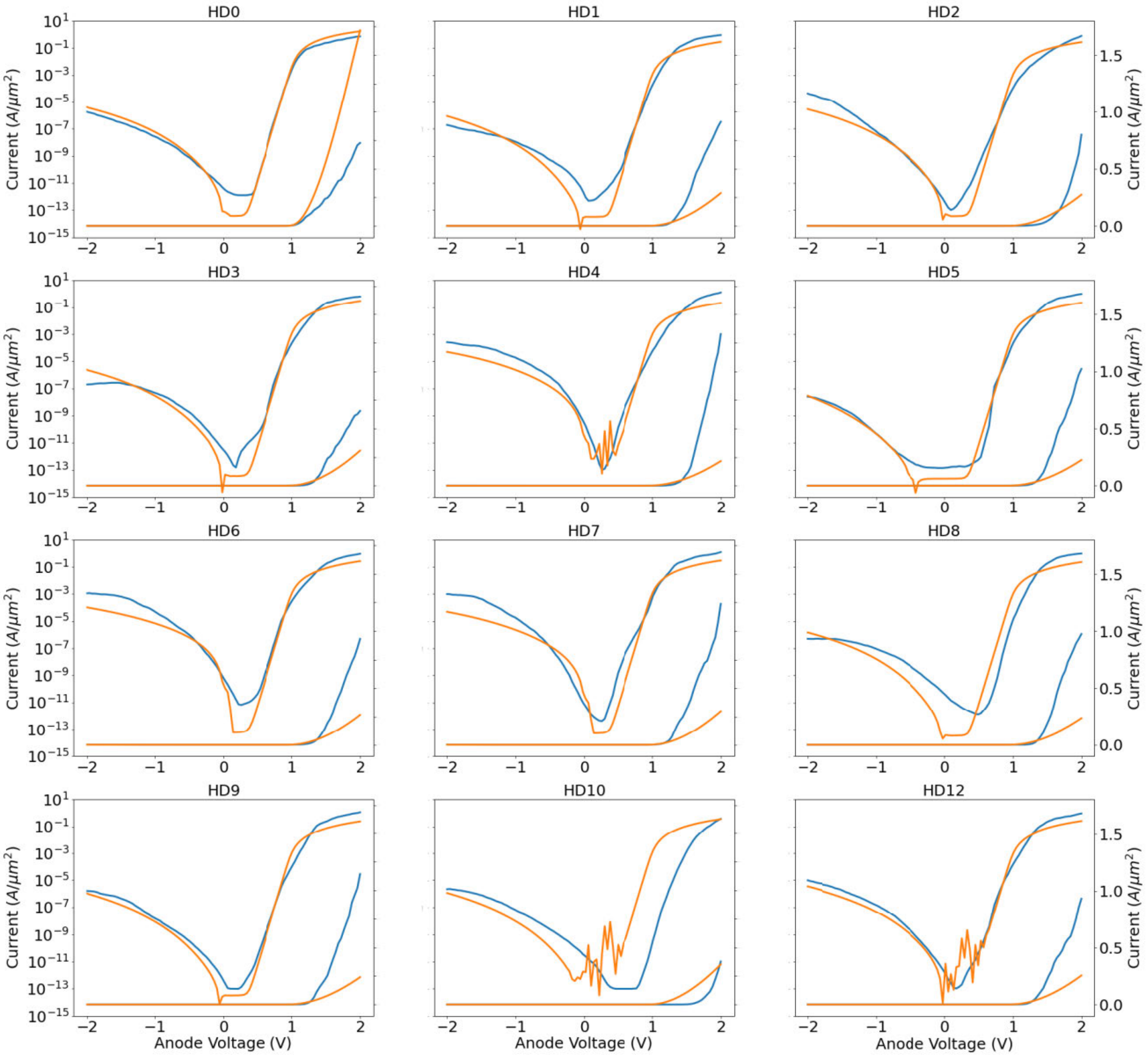

**FIGURE 13.** **The additional hand drawn *IV* curves from Fig. 12 (except HD11) and TCAD simulated *IV* curves using the structure predicted by *Model-Auto-Combined-1.* Left axes: in log scale. Right axes: in linear scale. Blue: Hand-drawn. Orange: TCAD.**

the reverse region, it has different slopes. Some are physical and some are impossible for the givens structure. 4) It has different sub-threshold slopes (SS) before $V_{bi}$. It should be noted that for a diode, SS is about 60mV/dec at 300K and is independent of the structure. 5) $V_{bi}$ is changed substantially for *HD10* and *HD11* (highlighted). For a given material, $V_{bi}$ should be almost constant as it depends on the bandgap of the material. Silicon has bandgap = 1.12eV. So *HD10* and *HD11* correspond to a material with bandgap $\sim$ 1.6eV and 0.6eV respectively. 6) The maximum forward current range is 5X and is much larger than the 2X in *Set 1*.

These curves are then fed into *Model-Auto-Combined-1* for reverse engineering to find the required thicknesses (Table 2). TCAD simulations are performed using the predicted thicknesses (Fig. 13). It can be seen that the machine can predict the thicknesses very well so that the corresponding structure can deliver the required leakage and forward currents. Since hand drawing was performed on log scales, most of the time, it requests an unphysical forward current (too low in *HD0* and too high in other cases). Again, as discussed early, the machine is able to match the part around $V_{bi}$ and disregard the unphysical request.

In particular, the machine can find a structure to deliver the maximum leakage and forward currents in *HD10*, although this curve corresponds to a diode with much larger bandgap material.

*HD11* is not shown in Fig. 13. This is because the machine has predicted $n + p$ thickness to be negative (Table 2 ). This shows a limitation of *Model-Auto-Combined-1*. On the other hand, this can be useful in real applications because the

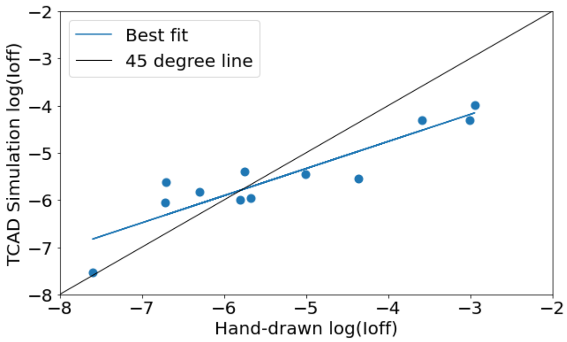

**FIGURE 14.** Scatter plot of TCAD simulation current against hand-drawn current at $V$ = *-2V* of the cases in Fig. 12.

device designer can be warned of the inappropriateness of the device specifications being requested (i.e. Silicon device cannot deliver such curve trivially).

Since hand-drawn curves are drawn arbitrarily, there is no "standard" answer (e.g. the "correct" thicknesses correspond to any hand-drawn curve) or simple measure to benchmark the performance of *Model-Auto-Combined-1* in its inverse design capability. It depends on which part of the curve is more important in the application. Usually, $I_{off}$ ($V$ = *-2V*) and $I_{on}$ ($V = 2V$) are two important quantities for a diode. Therefore, the TCAD predicted $I_{off}$ (resembling manufactured device created through inverse design) is compared against hand-drawn $I_{off}$ (Fig. 14). Ideally, they should be identical (correlation = 1). Although this is not the case, it is found that their correlation is 91% in log scale. Therefore, the performance of the inverse design is very good, provided the noises and abnormalities of the hand-drawn curves aforementioned, and the absence of domain expertise. Correlation in $I_{on}$ is not a suitable measure in this case because the machine tries to fit the curves so it retains the physically meaningful part around $V_{bi}$. If $I_{on}$ matching is important even it is unphysically large or small, a new machine needs to be developed and domain expertise is required (as $I_{on}$ is identified to be something important by experts) and this is out of the scope of this article.

## V. CONCLUSION

An ML model combining autoencoders and linear regression is proposed to improve TCAD-augmented machine learning for variation identification and inverse design. The validation is done by using TCAD simulated data set with new variations to emulate experimental data for better control and more data points to obtain high statistical confidence in the performance. The model can avoid overfitting and perform 15 times better than the *naïve model*. Moreover, it can be used to perform inverse design and has shown excellent results. It has also discovered an underlying physics (smoothness of the curve at $V_{bi}$ neighborhood) which cannot be described by a simple physical parameter.

## ACKNOWLEDGMENT

The authors would thank Dr. Philip Leong of the University of Sydney for the discussion of ML algorithms.

## REFERENCES

[1] D.-Y. Liao, C.-Y. Chen, W.-P. Tsai, H.-T. Chen, Y.-T. Wu, and S.-C. Chang, "Anomaly detection for semiconductor tools using stacked autoencoder learning," in *Proc. Int. Symp. Semiconductor Manuf. (ISSM)*, Dec. 2018, pp. 1–4.

[2] G. A. Susto, M. Terzi, and A. Beghi, "Anomaly detection approaches for semiconductor manufacturing," *Procedia Manuf.*, vol. 11, pp. 2018–2024, Jan. 2017, doi: 10.1016/j.promfg.2017.07.353.

[3] D. Ding, X. Wu, 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, doi: 10.1109/ICICDT.2009.5166300.

[4] R. Luo, "Optical proximity correction using a multilayer perceptron neural network," *J. Opt.*, vol. 15, no. 7, Jul. 2013, Art. no. 075708, doi: 10.1088/2040-8978/15/7/075708.

[5] Y. Yuan-Fu, "A deep learning model for identification of defect patterns in semiconductor wafer map," in *Proc. 30th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, May 2019, pp. 1–6.

[6] Y. S. Bankapalli and H. Y. Wong, "TCAD augmented machine learning for semiconductor device failure troubleshooting and reverse engineering," in *Proc. Int. Conf. Simul. Semiconductor Processes Devices (SISPAD)*, Sep. 2019, pp. 21–24, doi: 10.1109/SISPAD.2019.8870467.

[7] C.-W. Teo, K. L. Low, V. Narang, and A. V.-Y. Thean, "TCAD-enabled machine learning defect prediction to accelerate advanced semiconductor device failure analysis," in *Proc. Int. Conf. Simulation Semiconductor Processes Devices (SISPAD)*, Sep. 2019, pp. 17–20, doi: 10.1109/SISPAD.2019.8870440.

[8] H. Y. Wong, M. Xiao, B. Wang, Y. K. Chiu, X. Yan, J. Ma, K. Sasaki, H. Wang, and Y. Zhang, "TCAD-machine learning framework for device variability analysis with experimental demonstration on $Ga_2O_3$ power diodes," *IEEE J. Electron Devices Soc.*, submitted for publication.

[9] *Sentaurus Process User Guide Version O-2018.06*, Synopsys, Inc., Mountain View, CA, USA, Jun. 2018.

[10] *Sentaurus Device User Guide Version O-2018.06*, Synopsys, Inc., Mountain View, CA, USA, Jun. 2018.

[11] S. M. Sze, *Physics of Semiconductor Devices*. Hoboken, NJ, USA: Wiley, 1981, ch. 2.

[12] H. Carrillo-Nunez, 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 Lett.*, vol. 40, no. 9, pp. 1366–1369, Sep. 2019, doi: 10.1109/LED.2019.2931839.

[13] J. Chen, M. B. Alawieh, Y. Lin, M. Zhang, J. Zhang, Y. Guo, and D. Z. Pan, "Powernet: SOI lateral power device breakdown prediction with deep neural networks," *IEEE Access*, vol. 8, pp. 25372–25382, 2020, doi: 10.1109/ACCESS.2020.2970966.

[14] *Scikit-Learn: Machine Learning in Python*. Accessed: Apr. 15, 2020. [Online]. Available: https://scikit-learn.org/stable/

[15] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, Feb. 1991, doi: 10.1002/aic.690370209.

[16] Y. Zhang and W. Ye, "Deep learning based inverse method for layout design," 2018, *arXiv:1806.03182*. [Online]. Available: http://arxiv.org/abs/1806.03182

[17] E. Plaut, "From principal subspaces to principal components with linear autoencoders," 2018, *arXiv:1804.10253*. [Online]. Available: http://arxiv.org/abs/1804.10253

**KASHYAP MEHTA** was born in Faridabad, Haryana, India, in 1996. He received the B.Tech. degree in electrical engineering from Pandit Deendayal Petroleum University, Gandhinagar, Gujarat, India, in 2017. He is currently pursuing the M.S. degree in electrical engineering with San Jose State University, San Jose, CA, USA.

From 2017 to 2019, he was an Executive Engineer with Torrent Power Ltd. Since 2020, he has been a Research Assistant with San Jose State University. His research interests include applied machine learning and artificial intelligence, data analytics computer vision, and deep learning.

**SOPHIA SUSAN RAJU** was born in India. She received the bachelor's degree in electronics and communication from the Cochin University of Science and Technology, Kerala, India. She is currently pursuing the master's degree in electrical engineering with San Jose State University. She worked with Wipro Technologies for four years.

**MING XIAO** received the B.S. and M.S. degrees in electronic science and technology and Ph.D. degree in electronic science and technology from Xidian University, China, in 2012, 2015 and 2018, respectively. He is currently a Postdoctoral Associate with the Center for Power Electronics Systems, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University.

**BOYAN WANG** received the B.Sc. degree from Fudan University, Shanghai, China, in 2018. He is currently pursuing the Ph.D. degree with the Center for Power Electronics Systems, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University.

His current research interests include design, simulation, fabrication, characterization, and applications of (ultra-) wide band gap power devices.

**YUHAO ZHANG** (Member, IEEE) received the B.S. degree from Peking University, Beijing, China, in 2011, and the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2013 and 2017, respectively.

From 2017 to 2018, he was a Postdoctoral Associate with MIT. Since 2018, he has been an Assistant Professor with the Center for Power Electronics Systems, Bradley Department of Electrical and Computer Engineering, Virginia Tech. His research interests include power semiconductor devices, (ultra-) wide-bandgap semiconductor materials, and power electronics applications.

**HIU YUNG WONG** (Senior Member, IEEE) received the B.Eng. degree in computer engineering and the M.S.E. degree in computer science and engineering from The Chinese University of Hong Kong, in 1999 and 2001, respectively, and the Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, in 2006.

He is currently an Assistant Professor with the EE Department, San Jose State University, USA. From 2006 to 2009, he worked as a Technology Integration Engineer on 45/32nm NOR flash memory in Spansion, Inc. From 2009 to 2018, he was a Senior Staff AE with TCAD simulation in Synopsys, Inc. His research interests include the applications of machine learning in simulations, NBTI, and hot carrier degradation simulation in FinFET/nanowire/nanosheet, wide band gap materials (such as GaN, SiC, Ga2O3, and Diamond) device and reliability/defect simulations, novel semiconductor device design, and design technology co-optimization (DTCO). Part of the activities are reflected in the 70 publications and patents awarded.