Machine Learning-Based Schottky Diode Parameter Extraction

Hershel Millman[1], Darren Smith[1], Naazaneen Maududi[1], and Olivia Li[1] (Group 3)

Arizona State University
Ira A. Fulton Schools of Engineering
School of Electrical, Energy, and Computer Engineering

ABSTRACT

In this work, we explore the feasibility of using machine learning techniques to extract Schottky diode parameters from IV and CV characteristic curves. The principal goal is to determine an algorithm which can extract multiple Schottky diode parameters at once with high accuracy, and works with minimal data. The parameters of interest for this project include substrate material, contact material, substrate doping concentration, guardring doping concentration, contact length, electron affinity and the work function of the metal. The programming language Python is used for two tasks. First, it is used to generate random combinations of parameters, run a Schottky diode simulation with those parameters utilizing Silvaco ATLAS, and save the results. This process is repeated thousands of times to obtain data for the next step. Secondly, Python is used to clean up the data into a more easily processable format, develop a neural network using the PyTorch deep learning framework, and train and test the network. The network is scored for its accuracy and the results are compared to a set of hand calculations.

Keywords: Schottky Diode, ML Semiconductors, Parameter Regression

---

I. INTRODUCTION

Knowing the parameters that characterize a semiconductor device is essential when developing circuit designs. Given a sample device, the process of extracting material parameters is often a long and tedious process. This process is referred to as semiconductor characterization, and involves measuring the responses of the device to various stimuli and analyzing the resulting behavior to extract material or behavioral parameters. In this work, we explore the characterization of Schottky diodes.

I.a. Schottky Basics

A Schottky diode is the formation of a potential barrier between a metal and a semiconductor. Some metals used to construct these types of molybdenum, chromium, and tungsten[1]. These metals all have different work functions, and as such, each material has a unique impact of the behavior of the device it helps make up. Figure 1 on the following page shows a cross-section representation of a Schottky diode.

Forward conduction occurs when electrons pass over the potential barrier from the n-type

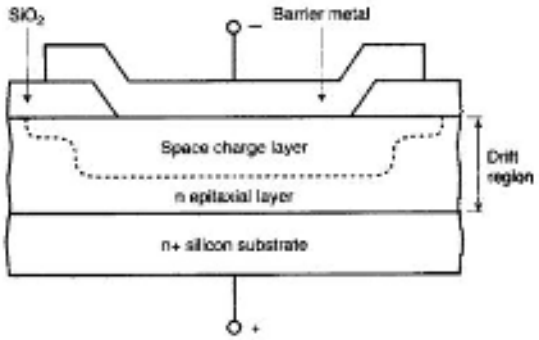

Figure 1: Cross section of Schottky Diode

semiconductor to the barrier metal, and vice versa for conduction. Reverse conduction is impeded by the guard rings, which reduce the elector field at the edge of the metal contact area. In Figure 3, the guard rings are not shown, but they are positioned directly beneath the intersection of the silicon oxide and the barrier metal on either side of the space charge ayer. Compared to a typical p-n junction diode, Schottky diodes have a very low forward voltage drop, a high current density, and fast reverse transit times [2].

Schottky diodes are widely used for high frequency and quick switching applications such as radio frequency circuitry, mixers, and rectifier in power applications etc. Therefore, the accurate extraction of the device parameters which further characterize the metal-semiconductor junction is vital. We have developed a basic machine learning algorithm for extraction of Schottky diode metal-semiconductor junction device parameters from easily obtainable data. The data that that has been attained for each device includes IV and CV characteristic curves, which are generated with Silvaco simulations. The device parameters of interest for the purpose of this study were semiconductor type, metal contact material, doping concentration of the n-type region, doping concentration of the guard rings, size in the x direction, size in the y direction, electron affinity and metal work function.

I.b. Semiconductor Material

Semiconductor type is an important parameter to consider in order to accurately characterize the metal-semiconductor junction. The semiconductor type will change the bandgap of the material and as a result the carrier concentration. Therefore, the different semiconductor types will showcase different carrier densities for each applied voltage. Electron affinity will change for various semiconductor types. Seen from equation (1), barrier height will vary as a result of electron affinity [3]. In order to determine the semiconductor type, the I-V curve will be analyzed within the algorithm for the turn on voltage.

$$\Phi_B = \Phi_m - \chi \qquad (1)$$

I.c Contact Material

Another important parameter within Schottky diodes is the metal material contact. Different metals will showcase higher or lower work functions. The change in work function will result in an increase or decrease in barrier height seen in (1). Contact resistivity will differ for specific metal contacts. In order to determine the metal material contact, the I-V will be analyzed for two regions. The first region is the turn on voltage which will help determine the barrier height. High diode currents are the second region that is analyzed. In this region, contact resistance will be prominent and determined from the I-V curve.

In addition to semiconductor type and metal material contact, the doping concentration is vital in describing the electrical characteristics of Schottky diodes. Due to thermionic emission, current is limited

within Schottky diodes [3]. When the doping concentration is higher, the depletion width is smaller as seen by (2). As the depletion width shrinks, current through the diode changes

$$W \approx \sqrt{\frac{2\epsilon_s\epsilon_0(Vbi+V)}{qN_A}} \quad (2)$$

from thermionic emission to field emission. Shown by (3) it can be seen that as depletion width is smaller, current increases through the area of junction.

$$qN_A A\frac{dW}{dV} = \sqrt{\frac{qN\epsilon_s\epsilon_0 N_A A}{2(V_{bi}+V)}} = \frac{\epsilon_s\epsilon_0 A}{W} \quad (3)$$

As the doping concentration becomes large enough, field emission is the dominating current which is tunneling through the barrier. This can be determined by analyzing the change in I-V characteristics for various doping concentrations. A representation of how an increase in doping concentration can affect the I-V characteristic of the diode is shown in Figure 2. In order to determine the doping concentration and dopant type, the C-V curves from Silvaco are going to be analyzed. Expressed in equation (4), the slope within the C-V curves is inversely

$$N_A = \frac{2}{A^2 e\epsilon_s\epsilon_0 \frac{d(1/C^2)}{dV}} \quad (4)$$

proportional to the doping concentration. In order to determine dopant type such as a p or n type, the capacitance will either increase or decrease depending on the voltage applied being positive or negative.

I.d Active Area

Lastly, the active area of the Schottky diode is another parameter that helps define electrical characteristics of the metal-semiconductor junction. The active area is directly proportional to the current produced within Schottky diodes as seen in (5) [4]. By changing the active area, the current produced will be changed for each potential applied. This parameter is determined at the end and is used to reach the desired current.

$$I = AA^*T^2 exp(\frac{-q\Phi_b}{kT})[exp(\frac{qV_D}{nkT})] \quad (5)$$

In (5) A is the area of the junction, A* is the Richardson constant, T is the temperature in Kelvin, Φ_B is the Schottky barrier height, and n is the ideality factor.

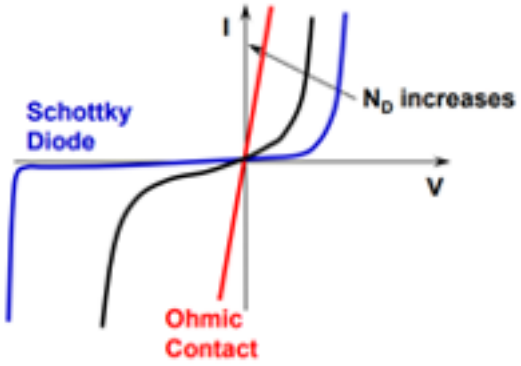

Figure 2: Current versus voltage curve for different doping concentrations

III. Analysis and Methods

Determining semiconductor parameters by hand is sometimes possible, but is often inaccurate or slow, and sometimes both. Additionally, solving for multiple parameters is often difficult given a limited amount of data to work with from any individual semiconductor device. The ideal method would be one which can extract many parameters at once from limited data, and is both fast and accurate. Recent advances in machine learning methods offer the ability to make accurate inferences about the properties of a subject (whether that subject be an

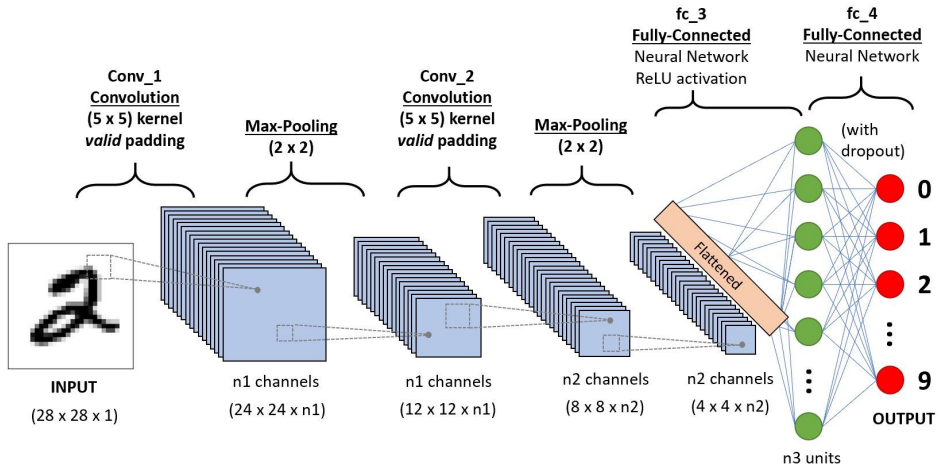

Figure 3: Diagram of Convolutional Neural Network

image, a video, a book, etc.) given limited information about that subject. Classic examples include algorithms to recognize faces in images, classify objects into different categories, and fill in missing areas of images, to name a few.

Many of the tasks undertaken within machine learning employ some type of convolutional neural network. A neural network is a collection of layers of neurons, into which an input is passed, a nonlinear function is applied to those inputs, and the results are passed along through the layers until the end of the network. A convolutional neural network makes use of an operation called convolution, which involves sliding a square window of values (called a filter) over a matrix. A diagram of a convolutional neural network is shown in Figure 3. At each point, the values in the filter are multiplied with their corresponding values in the matrix, and these products are summed. The values inside the filter determine what kind of information is extracted. In an image, for example, there are filters which extract edges from an image or recognize corners, and more complex filters can recognize features including human faces or other complicated features. When training a convolutional neural network, one is trying to determine what values to put in the filters, how many filters to use, and how many to apply at a time. Because the network has layers, and each layer operates on the output of the previous layer, neural networks are often able to detect meaningful information that is not easily discernible from the original data. It is because of this property of neural networks that this method was chosen for characterization. Convolutional neural networks are very good at extracting relationships between neighboring points, because all convolution operations are local to one small region at a time. The multi-layer perceptron (Figure 4) is good at assigning weights to features, i.e. determining which features are important and deciding how to combine them to form a final representation of the data. Training a neural network refers to the algorithm used to determine the values of the filters in the non-linear function parameters convolutional neural network.

The process of training a network is complex, and will not be covered at great length here, but one important aspect of

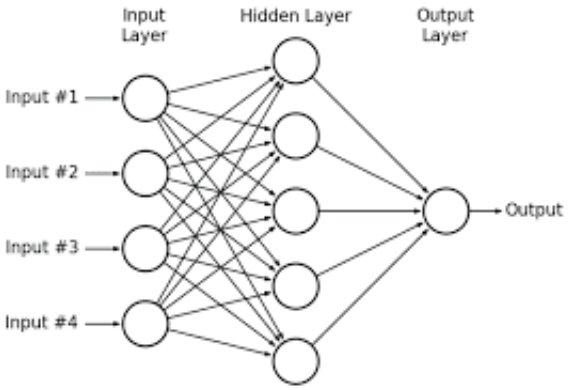

Figure 4: Diagram of Fully Connected Neural Network

training will be discussed. First, the quantity and quality of available data must be high. If the model is complex, it will require hundreds or even thousands of pieces of training data. The model needed for this project is very complex, so many samples were necessary to train the network.

Obtaining the data was straightforward. For each parameter to be changed, a range of acceptable values was decided. These values are shown in Table 1a. Once these ranges were determined, a Python script was used to perform simulations. The script would randomly choose a value for the substrate and contact material from the proper lists, and would sample from a uniform distribution over the correct ranges for that parameter. The script would write a Silvaco input file with the proper parameter values, mesh definitions, and voltage sweep. The range of the voltage sweep was from 0V to 10V, but a small variation was added in each run to ensure that the voltage values were not identical between runs, to account for similar inaccuracies in real-world measurements. Once the simulation had finished running, the python script would parse the output file for the IV and CV curves, and store them in a .csv file. The script performed roughly 60,000 simulations with different parameter values. Unfortunately, some of those simulations did not converge, and some of the data files were corrupted. After post-processing of the data, there were around 40,000 sets of curves to use for our network where 80% of the data was used for training, and 20% was used for validation.

When passing the data to the convolutional neural network, it was “stacked” to form a two dimensional array. This was done so that the relationships between the different curves could be extracted. Before training, the data was normalized to fit on the range from 0 to 1. This was done because it generally results in faster training times and higher accuracy in the network. The network took in a two-

Table 1 - Parameter Ranges

Parameter	Acceptable Range
X Scale Factor (from 12 um)	.5 - 2
Y Scale Factor (from 5 um)	.5 - 2
Contact Length (um)	1 - (12 * scale - 1)
Log Guard Ring Doping Concentration	19 - 23 (non-log is 1e19 - 1e23 cm^-2)
Log Substrate Doping Concentration	18 - 22 (non-log is 1e18 - 1e22 cm^-2)

dimensional array of data, performed 5 layers of convolutions, and had two fully connected layers. The stacking scheme for input to the network is shown in Figure 5. The training process took about two hours.

[	V1	V2	V3	V4	…	Vn]
[	Ia1	Ia2	Ia3	Ia4	…	Ian]
[	V1	V2	V3	V4	…	Vn]
[	Ib1	Ib2	Ib3	Ib4	…	Ibn]
[	V1	V2	V3	V4	…	Vn]
[	C1	C2	C3	C4	…	Cn]
[	V1	V2	V3	V4	…	Vn]

Figure 5: Stacking scheme for data to pass into the network

III. Results

The network's loss during training (a measurement of how far from correct it is) is plotted in Figure 6. The lower the loss, the more accurate the network. At first, the loss function was simply the mean squared error of all of the parameters, however, this function was not very informative for the network, and it was not able to obtain very high accuracy. Thus, the loss function was split into 4 parts. The loss became the classification error of the semiconductor material, the classification error of the contact material, the mean squared error of the parameters involving length (x scale factor, y scale factor, and contact length), and the mean squared error of the rest of the parameters. Each of these errors were given relative weights, so the network could factor in their relative importance to each other when updating its parameters. When the original loss function was used, the errors were as high as 50% for some of these parameters. However, this decreased drastically with the new loss function. The log losses during training are plotted below in Figure 6, showing how the accuracy of the different groups of parameters improved as training progressed.

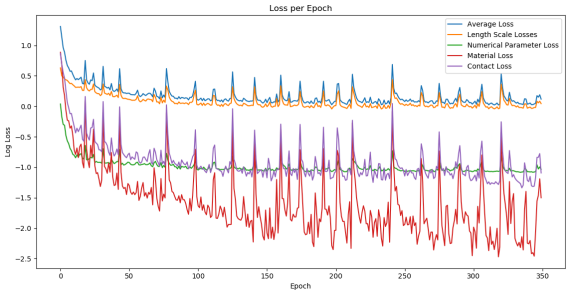

Figure 6: Log loss during training

Table 2 - Performance on Unseen Data

Parameter	Average Error	Average % Error (%)
X Scale Factor (from 12 um)	0.18204	16.89697%
Y Scale Factor (from 5 um)	0.15691	14.41294%
Contact Length (um)	2.45805	40.29127%
Log Guard Ring Doping Concentration	0.34675	1.62811%
Log Substrate Doping Concentration	0.34675	1.555%
Electron Affinity (eV)	0.03290	0.88138%
Work Function (eV)	0.04014	0.88045%

Once training was completed, the network's performance was evaluated on the test set (data it had never seen before) to try to infer the parameters used to create those curves. The performance of the network on data it had never seen before is shown in Table 2.

As can be seen from the results, some of the parameters are able to be regressed very accurately, and some are not. The parameters involving length were not able to be found with high accuracy. However, all of the other parameters were.

Most notably, the network was able to classify the substrate material and contact material correctly more than 99% of the time! Once trained, the model is able to run on a set of curves in a few milliseconds for fast, accurate classification of materials and most parameters.

To further evaluate the performance of the network, manual calculations were done for a family of curves for which the parameters were known. The only parameter that was able to be extracted by hand was the n-type doping concentration. The value found was incorrect by a factor of over 100. However, the value obtained by the network was only off by a factor of three. Additionally, the parameters that were not able to be found by hand (not including length, height, and contact length) were found with error of less than 2%.

The accuracy of the network was limited by the amount of training data available, and the amount of time it took to collect that data. If twice as many data points per curve were collected, the network would have likely been much more accurate. Therefore this method could be used by a semiconductor company for rapid characterization of its devices. A company would be able to train their version of this network on millions of IV and CV response curves, and would then have a quick, easy method of Schottky diode characterization. Additionally, the method could be expanded to apply to other semiconductor devices as well, including zener diodes and MOSFETs.

IV. Conclusion

Many scripts using Silvaco and Python were written and simulated to find distinct parameters that created various IV and CV response curves of a Schottky diode. Thousands of input decks were simulated and used, but several were unable to converge and had to be discarded. Further error checking was done by a cleanup script to ensure that the values that did converge were accurate and usable for the machine learning algorithm to learn from, and were realistic curves that could have been captured from a physical diode.

The amount of time spent doing simulations was somewhat of an obstacle for this project. To obtain the data, 16 simulations were run in parallel for approximately 48 hours total. The accuracy of the algorithm could be greatly increased if instead of 48 hours of data generation, we could have simulated for two weeks, or even a month.

All parameters tested and simulated could be manipulated further to better detect parameter changes and teach the algorithm more ways to pinpoint the correct parameter values. Additionally, more careful tuning of the loss function could possibly increase the accuracy further.

Parameters that were not able to be accurately determined could be explored in future projects to improve the algorithm. Overall the project was a great success, and the team learned a lot about Schottky diodes, machine learning algorithms, as well as semiconductor parameter extraction.

REFERENCES

[1] D. F. Warne and M. A. Laughton, “17 - Power Semiconductor Devices,” in *Electrical Engineer's Reference Book (Sixteenth Edition)*, Elsevier, 2003, pp. 17–25-17–27.

[2] S. J. Pearton, F. Ren, and M. Mastro, “11 - Schottky contacts to β-Ga2O3,” in *Gallium oxide: technology, devices and applications*, Amsterdam: Elsevier, 2019, pp. 231–261.

[3] M. O'Hara, “CHAPTER 3 - ACTIVE DISCRETE COMPONENTS,” in *EMC at component and PCB level*, Oxford: Newnes, 2003, pp. 43–60.

[4] S. K. Cheung and N. W. Cheung, “Extraction of Schottky diode parameters from forward current−voltage characteristics,” *AIP Publishing*, 14-Jul-1986. [Online]. Available: https://aip.scitation.org/doi/pdf/10.1063/1.97359@apl.2019.APLCLASS2019.issue-1. [Accessed: 04-May-2020].