MACHINE LEARNING BASED AUTONOMOUS QUALITY CHECK AND CHARACTERISTICS EXTRACTION FOR CHIP RESEARCH

(ÇİP ARAŞTIRMALARI İÇİN MAKİNE ÖĞRENMESİ TEMELLİ OTONOM KALİTE KONTROLÜ VE KARAKTERİSTİK PARAMETRE ÇIKARIMI)

by

Hüsnü Murat KOÇAK, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Supervisor: Assist. Prof. Dr. Ahmet Teoman NASKALI

July 2022

This is to certify that the thesis entitled

MACHINE LEARNING BASED AUTONOMOUS QUALITY CHECK AND CHARACTERISTICS EXTRACTION FOR CHIP RESEARCH

prepared by **Hüsnü Murat KOÇAK** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering** at the **Galatasaray University** is approved by the

Examining Committee:

Assist. Prof. Dr. Ahmet Teoman NASKALI (Supervisor)
Department of Computer Engineering
Galatasaray University ----------

Assist. Prof. Dr. Günce Keziban ORMAN
Department of Computer Engineering
Galatasaray University ----------

Assist. Prof. Dr. Bahri Atay ÖZGÖVDE
Department of Computer Engineering
Bogazici University ----------

Date: ----------

ACKNOWLEDGMENTS

I owe a debt of gratitude to both of my supervisors; Assist. Prof. Dr. Ahmet Teoman Naskali and Dr. Jerome Mitard for being more than an advisor, providing continuous motivation and full support at every single moment. This project would not have been done without their expertise and support.

Many thanks to my dear flatmate Anja Ulrich for all her help and social motivation, to IMEC and KU Leuven for giving me the opportunity to work in one of the best facilities in the world, to all the scientists in the Memory Department at IMEC for their valuable contributions to my research, and to the fellows of lovely Galatasaray University for the excellent education they have taught. Finally, I especially thank my family for their support throughout my education life.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

TABLE OF CONTENTS v

LIST OF FIGURES vii

LIST OF TABLES x

ABSTRACT xi

RÉSUMÉ xiii

ÖZET xv

1 INTRODUCTION 1

2 DEFINITION OF THE PROBLEM 5

2.1 MOSFETS 7

3 LITERATURE REVIEW 11

3.1 RNN BASED TRANSISTOR MODELLING 11

3.2 CNN BASED PASS FAIL/CHECK AND KEY PARAMETER EXTRACTION 12

4 EXPERIMENTS PREPARATION 15

4.1 ENVIRONMENT SETUP 15

4.2 DATASET SETUP 15

4.3 OVERFITTING PREVENTION 18

5 RECURRENT NEURAL NETWORKS BASED TRANSISTOR MODELLING 20

5.1 DATA REPROCESSING 23

5.2 MODEL DETAILS 23

5.3 EXPERIMENT RESULTS 24

6 2-STEP CONVOLUTIONAL NEURAL NETWORKS BASED APPROACH 26

6.1 THE CONVOLUTION OPERATION 27

6.2 SUMMARY OF WORK 29
7 1ST STEP : DEVICE CHARACTERISTICS CLASSIFIER 30
7.1 DATA REPROCESSING FOR TRAINING 31
7.2 CHOOSING THE RIGHT ACTIVATION FUNCTION 31
7.3 MODEL DETAILS 32
7.4 EXPERIMENT RESULTS 33
8 2ND STEP : THRESHOLD VOLTAGE EXTRACTOR 37
8.1 TOWARDS BUILDING AN ACCURATE EXTRACTOR MODEL 38
8.1.1 CHOOSING THE RIGHT ML ARCHITECTURE . . 39
8.1.2 CHOOSING THE RIGHT LOSS METRIC 41
8.2 MODEL DETAILS OF FINALIZED EXTRACTOR MODEL . 42
8.3 EXPERIMENT RESULTS 43
9 VISUALIZATION OF RESULTS 46
9.1 WAFER VIEW 46
9.2 DEVICE VIEW 48
10 FUTURE WORKS 49
11 CONCLUSION 50
REFERENCES 52
BIOGRAPHICAL SKETCH 57

LIST OF FIGURES

Figure 1.1 Illustration of the motivation of this work. 2

Figure 2.1 An n-type MOSFET illustration. Gate is completely isolated from the rest of the circuit by the insulator oxide. The voltage on the gate creates a field effect where an increment of gate voltage will repel the holes in the substrate away from the area between the source and drain. This area is called the channel region (x) where an inversion layer of electrons starts to form at the source, and as the voltage increases, expands toward to drain. L represents the length of the channel region. After gate voltage passes the threshold voltage, electrons from the source and drain flow in, and form an inversion layer of electrons that connect the source and drain regions. Finally, a voltage that is applied to the source, a current will flow. 8

Figure 2.2 4-terminals I-V characteristics of a FinFET and at two inputs drain voltages (V_{ds}) (0 is V_{ds}= 50mV, and 1 is V_{ds}= 800mV). Healthy, not-healthy, and not-sure labels evaluated by the same expert. 9

Figure 4.1 A few examples of very advanced measuring equipment that includes nanometric level sensitivity for electrical and material characterization is widely used in research centers. 16

Figure 4.2 Wafer images. 16

Figure 4.3 Each chip includes multiple modules, and each modules comprised of multiple devices that consists 4 terminal ; drain, source, gate, and body (or bulk). 17

Figure 4.4 Illustration of dataset operations for creating training dataset for each ML model. 18

Figure 5.1 Our BiLSTM method combines two techniques (Graves et al., 2013). 21

Figure 5.2 A few example from the research towards building an accurate LSTM architecture. Green lines represent gate currents, red lines represent drain currents, yellow lines represent body currents, and finally, blue lines represent source currents. The structure as is follows ; (a) : upper is single layered bidirectional LSTM prediction, lower is actual values ; (b) : upper is single layered univariate convolutional LSTM prediction, lower is actual values ; (c) : upper is single layered multivariate convolutional LSTM prediction, lower is actual values ; (d) : upper is classic LSTM prediction, lower is actual values. 22

Figure 5.3 Preprocessing of the raw dataset for our LSTM architecture. 23

Figure 5.4 An example of transistor modeling result by bidirectional LSTM architecture. Green dotted plots represent actual gate currents, yellow dotted plots represent actual body currents, blue dotted plots represent actual source currents, and red plots represent drain currents while dotted plots represent actual drain currents, line plots represent predicted drain currents. (a) : predicted and actual results in linear scale (b) : predicted and actual results in log10 scale 24

Figure 6.1 Finalized system structure and data flow. 26

Figure 6.2 An example of 2D kernel convolution on 2D input data. The upper-left element of the output tensor is created by applying the kernel to the matching upper-left portion of the input tensor, as shown by the box with arrows. 28

Figure 7.1 Our classifier architecture is visualized by NN-SVG (LeNail, 2019) technique. 32

Figure 7.2 Training and validation loss over the epochs, while the y-axis represents loss and the x-axis, represents the number of epochs. 34

Figure 7.3 Comparative accuracy results of experts and our CNN architectures. 35

Figure 8.1 An illustration of how our ML method can be used to overcome the shortcomings of the peak-gm method. (a) : Device routine outputs in linear scale, (b) : V_{th} over Vds outputs obtained from peak-gm and our extractor. Since the peak-gm method does not follow any sequence information among Vds, It can generate outputs that can lead to confusion, while our ML approach considers a correlation between Vds and V_{th}. 38

Figure 8.2 Training and validation loss of 8 different CNN architectures. 40

Figure 8.3 Training and validation loss with respect to 3 different loss functions 42

Figure 8.4 ML Architecture illustration of CNN-based extractor model. 43

Figure 8.5 The error of each prediction. Max tolerated error voltage of 20mV is specified and non-complied devices are classified as not-healthy 44

Figure 9.1 An example of wafer view of modules. The green border represents the modules that have no leaks, while the red border represents the modules have leaks. The squares inside modules represent devices and their leaks statues with their color. 47

Figure 9.2 Visualization of classifier outputs and extractor models on Id-Vg curve of each device. 48

LIST OF TABLES

Table 7.1 Confusion matrix scores of single CNN classifier, including the mean of all test cases. 33

Table 7.2 Confusion matrix scores of multi model ensemble of CNNs approach. 35

Table 8.1 Table of CNN architectures that have been used. 39

Table 8.2 Table of convergence rate and losses of selected architectures 41

ABSTRACT

The flawless and correct functioning of high-end electronic devices is of utmost importance for consumers as well as for manufacturers. Although Metal-Oxide Semiconductor Field Effect Transistor (MOSFET) manufacturers use a variety of protocols and procedures in the development and production of high-tech devices, the integrated circuits are usually manufactured by third parties. The performance of the transistors used in integrated circuits can vary from batch to batch during manufacturing, and even samples within the same batch can have different performance characteristics. For applications in life-critical domains, the selection of the most fit for purpose components is vital. In addition, long quality control processes make it difficult for research institutes to reach faster and lower energy consumption chips. While this time-cost factor is reflected as a potential loss of revenue for research institutes, it may cause them to fall back in cutting-edge chip technology competition in today's world and cause a decrease in market size on a global scale. Therefore, while the developments for time-cost reductions directly affect positively on the industry and customers, they also pave the way for collaborations and academic contributions.

In this thesis, we propose a new approach for the semiconductor industry to verify the quality of transistors and the extraction of characteristics of transistors. The I-V graphs of the components are evaluated by multiple Convolutional Neural Networks using visual data in a manner similar to expert evaluation, and then these Machine Learning architectures use a multi-model ensemble technique in which one architecture providing a negative output overrules the vote of the other architectures to ensure very stringent quality control. After the filtering process, we implemented a second model that we created in the CNN technique to extract the threshold voltage (V_{th}, the voltage point at which transistors switch from non-conducting state to conductive state), which is an important characteristic parameter to allow the researcher to get a general overview of devices. With these techniques, we can conduct a performance assessment of the printed chips and get fast estimation of experimental chip architectures with their response to the input voltage. Our 2-step ML approach has been tested on

approximately 2500 filtered devices. Our technique has achieved senior expert-level accuracy in filtering, and 50mV total error rate in not filtered dataset, 8mV total error rate in the filtered dataset by V_{th} extraction.

Keywords : Machine Learning, Semiconductor, Characterization, Convolutional Neural Networks, Chip Research

RÉSUMÉ

Le fonctionnement correct et infaillible de composants électroniques de haut niveau est de la plus grande importance pour les consommateurs ainsi que pour les fabricants. Bien que les fabricants de transistors de type MOSFET utilisent des protocoles et procédures variés pour le développement et la production de composants de haute technologie, les circuits intégrés restent généralement produits par des tiers partis. La performance des transistors utilisés dans les circuits intégrés peut varier d'une série de production à l'autre lors de la fabrication, cela allant même à des variations de performance entre différents échantillons d'une meme série. Pour des applications dans des domaines critiques, la sélection des composants les plus adaptés est donc particulièrement importante et nécessaire. De plus, de longs processus de contrôle qualité peuvent entraver le développement de puces plus rapides et moins consommatrices d'énergie par les instituts de recherche. Cette perte de temps et de possibilité de développement peut se répercuter sur les revenus des instituts de recherche et peut leur causer un retard dans la compétition un niveau de l'innovation et la création de technologies de pointe à l'échelle mondiale. Ainsi, la mise en place de solutions réduisant les possible pertes de temps affectent positivement les industriels et les consommateurs et peut ouvrir la voie à des collaborations et contributions académiques.

Au cours de cette thèse, nous proposons une nouvelle approche pour vérifier la qualité des transistors et extraire leurs caractéristiques. Les courbes I-V des composants sont évaluées par de multiple réseau neuronal convolutif (CNN) utilisant des données visuelles d'une façon similaire à celle de l'évaluation d'un expert. Ensuite, ces architectures de apprentissage automatique (ML) utilisent une technique d'ensemble multi modèles dans laquelle une architecture donnant une sortie négative surpassera les autres architectures afin d'assurer un contrôle qualité strict. Après, le processus de filtrage, nous avons implémenté un deuxième modèle créé en technique CNN pour extraire la tension limite des transistors (Vth, est la tension pour laquelle les transistors passent d'un état semi-isolant à un état semi-conducteur) qui est un paramètre important pour le chercheur afin d'obtenir une compréhension générale du composant. Avec ces techniques, nous pouvons réaliser une évaluation des performances des puces, et obtenir

une rapide estimation de l'architecture expérimentale des puces avec leur réponse à une tension d'entrée. Notre approche ML en 2 étapes a été testée sur approximativement 2500 composants filtrés. Notre technique a atteint un niveau d'expert confirmé en filtrage, avec un ratio d'erreur de 50mV pour les donnés non filtrées et de 8mV pour les donnés filtres par extraction de tension limite.

Mots Clés : Apprentissage Automatique, Semi-conducteur, Caractérisation, Réseau Neuronal Convolutif, Recherche de Puces

ÖZET

Metal Oksit Yarı İletken Alan Etkili Transistör (MOSFET) cihazlarının doğru ve kusursuz çalışması, müşteriler ve yarı iletken araştırmaları için büyük önem taşımaktadır. MOSFET cihaz üreticileri, yüksek teknolojili cihazların geliştirilmesi ve üretimi sırasında çeşitli protokoller ve prosedürler kullanmasına rağmen, entegre devreler genellikle üçüncü parti üreticiler tarafınca üretilmektedir. Entegre devrelerde kullanılan yarı iletkenlerin imalatı sırasında bileşenlerin performansı üretilen partiden partiye değişebilmekte ve hatta aynı parti içindeki numuneler bile farklı performans özelliklerine sahip olabilmektedir. Özellikle hayati öneme sahip uygulamalarda amaca en uygun bileşenlerin seçimi hayati önem taşır. Ayrıca uzun kalite kontrol süreçleri, araştırma enstitülerinin daha hızlı ve daha düşük enerji tüketiminin sahip çiplere ulaşmasını zorlaştırmaktadır. Bu durum, araştırma merkezlerinin günümüz dünyasındaki üst seviye teknoloji çip mimarisi rekabetinde geri kalmalarına yol açabilir ve küresel ölçekte pazar hakimiyetlerinin azalmasıyla birlikte potansiyel gelir kaybı yaşamalarına sebebiyet verebilir. Bu nedenle zaman-maliyet azaltımına yönelik gelişmeler sektörü ve üçüncü parti cihaz üreticilerini doğrudan olumlu etkilerken, Oluşabilecek iş birlikleriyle de akademik katkıların önünü açmaktadır.

Bu tezde, transistörlerin kalitesini doğrulamak ve karakteristiklerinin çıkarımını sağlamak için yarı iletken endüstrisi için yeni bir yaklaşım öneriyoruz. Negatif oy veren bir mimarinin diğer mimarilerinin oylarını geçersiz kılacağı çok modelli bir topluluk tekniği kullanılarak bir araya getirilen üç farklı ve birbiriyle haberleşebilen Evrişimli Sinir Ağı (Convolutional Neural Networks) modeli, cihazların I-V grafiklerini -uzman değerlendirmesine benzer olarak görsel veriler üzerinden- değerlendirerek oldukça sıkı kalite kontrolü sağlar. Filtreleme aşamasından sonra önemli bir karakteristik parametresi olan eşik voltajı (V_{th}, transistörlerin iletken olmayan halden iletken hale geçtikleri voltaj noktası) tespiti için CNN mimarisinde oluşturduğumuz ikinci bir model kullandık. Bu sayede oluşturulmuş çip mimarisinin performans ölçümlemesi yapılabilirken aynı zamanda anomali tespitiyle deneysel mimarileri hakkında hızlı tahminleme almamızı sağlamaktadır. V_{th} tespiti modelimiz filtrelenmiş yaklaşık 2500 adet cihaz üzerinde test edilmiştir. Alaninda uzman bilim insani seviyesinde filtreleme yapmistir ve filtre-

lenmemis verisetinde 50mV error, filtrelenmis verisetinde 8mVluk hata ile eşik voltajı üretimi başarısı göstermiştir.

Anahtar Kelimeler : Makine Öğrenmesi, Yarı İletkenler, Karakterizasyon, Evrişimli Sinir Ağı, Çip Araştırması

1 INTRODUCTION

Integrated circuits are used in some form or another in almost all high-tech electronic equipment. For essential applications such as scientific, aeronautical, automotive, and medical equipment, the performance quality of these components is critical. These integrated circuits, which are made up of sophisticated and interconnected transistor components, allow us to do precise computations. With the advancement of manufacturing tools and fab equipment, substantial research is being done to lower the size of the transistors to gradually increase the number of transistors included on the chips after a long, time-consuming period of design and modeling activities. Today, 7nm silicon logic architectures are common, while advanced manufacturing facilities can shrink them to 3nm (Radamson et al., 2020).

Chip foundries and governments are building more and more capacity into supply chains for increasingly innovative electronics products as demand for chips continues to grow and see an uptake as more industries transform digitally. This push, if not handled carefully, may have an impact on the final product's quality control. On the R&D front, increased use of new materials and new chip architectures are currently being seriously considered (Horiguchi et al., 2020) in order to maintain the rapid performance growth of compute-oriented systems. From a massive amount of experimental data, new physics-based mechanisms or process-induced phenomena should be captured as soon as possible. Failure to do so could put a stop to disruptive, smart, and sustainable solutions in a variety of fields, including healthcare, clean energy, and Industry 4.0 to name a few (Singer, 2020).

Manufacturers have been able to reduce costs, reduce testing time, increase performance, and improve product performance thanks to advancements in semiconductor technology. However, as products become more demanding, high-power optimization and high-frequency operation, as well as the increased number of semiconductors in a single package, have resulted in an increase in the number of defective components. These flaws not only cause quality inconsistencies but also lead to low yields, causing manufacturing costs and a loss of brand reputation. Currently, device performances are examined using a Wafer Map (WM) visualization, which provides information of the

device performances (Yuan et al., 2011). Even when all clean room standards are met and cutting-edge technology is used in the manufacturing process, problems in WM are unavoidable (Wang et al., 2006).

One tool which can be used to quickly, accurately, and predicatively impact awareness of semiconductor devices is the Neural Network (NN) based Machine Learning (ML) method through the relationship between inputs and outputs that predicts outputs with respect to new inputs. The use case which will be given in this thesis is the autonomous classification and threshold voltage extraction of output characteristics of the key building blocks of the semiconductor industry : the Metal Oxide Semiconductor Field Effect Transistor (MOSFET). These transistors exist in almost all manufactured integrated circuits. As a result, following a brief description of the data to be analyzed, we will place this work in the context of similar research papers and introduce our novel approach.

Hence, after a rapid description of the data to be analyzed, we will place this work in the context of similar research papers and introduce our new approaches.

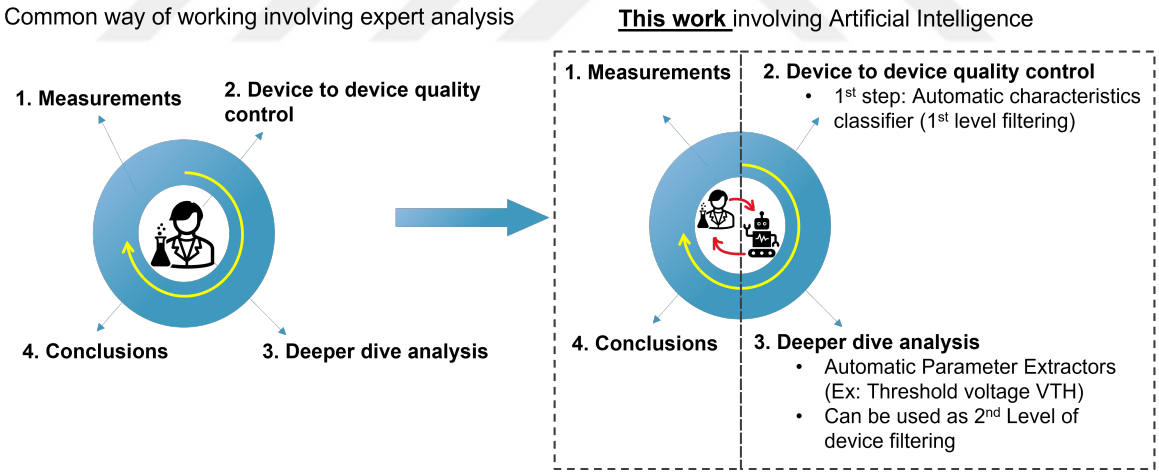


FIGURE 1.1 – Illustration of the motivation of this work.

In this thesis, we propose a new approach for the semiconductor industry to verify transistor quality and extract the on-off switch voltage, namely V_{th}. Traditional methods for evaluating transistor quality involve calculating on-off switch currents for each device in the wafer using relative complex computations which contain lots of inclusive and exclusive parameters (such as the temperature of the environment as an exclusive feature, silicon width as an inclusive feature) one by one without any filtering operations. After these time-consuming and costly procedures are completed, they are classified as healthy (no or minimal anomaly) or not-healthy (anomaly level

exceeds an acceptable threshold) based on their statistical distribution by experts eye in the field. Devices classified into these two classes not only measure the success of the architecture being developed but also aid in the development of a road map for next-generation experimental architectures. In short, conducting transistor quality control and classification is critical not only for current but also for future architectures.

For quality control of these components, some samples are currently taken from each batch and tested by experts. Expert evaluation is by no means a gold standard in this field, as even the same data evaluated by the same expert at different times can yield different results. These factors necessitate multiple evaluations of the same data which is a tedious and costly process. Automated methods, such as the one presented in this thesis, can enable rapid and accurate testing of all manufactured components.

Since these processes need to be interpreted individually for each device, computations can take hours or even days depending on the number of transistors in the wafer maps. Therefore, speeding up and automating the evaluations greatly reduces both time and source costs, and accelerates the processes of the end-user product.

The structure of this thesis is as follows : The definition of the problem that we have dealt with and the importance of the work are presented in chapter 2. Descriptions and processes of producing MOSFETs are described in section 2.1. In the chapter 3, previous attempts regarding machine learning are described. Section rnnbased focuses on RNN-based approaches, while section 3.2 focuses on CNN-based approaches. Chapter 4 covers the experiments preparation steps and following ; environment setup is presented in section 4.1, collection and processes of datasets are presented in section 4.2, and our operations for solving the overfitting problem are presented in section 4.3. We in-

troduced our attempts for RNN-based transistor modelling in chapter 5 and following; data preprocessing and representation are described in section 5.1, model details of the LSTM ML model architecture are described in section 5.2, and our complete analysis of the RNN-based approach is described in section 5.3. Then our main work which is a 2-step CNN-based approach is presented in chapter 6. Section 6.1 briefly describes general convolution operation and adaptation for the CNN architecture. The section 6.2 composes all the operations. Our first step which is device characteristics classifier method is introduced in chapter 7 and following; section 7.1 describes our operations for fitting data to an ML model, section 7.2 describes our research towards choosing the right activation function, section 7.3 presents our final CNN model, and lastly, our experiment results and inferences are presented in section 7.4. Next, our second step which is a threshold voltage extractor is introduced in chapter 8 and following; our research for building the best possible ML extractor model is described in section 8.1, while section 8.1.1 covers research for ML models, and section 8.1.2 covers research for choosing loss function. Model details are described in section 8.2, and our in-depth review of the CNN based approach, results, and inferences is presented in section 8.3. Visualization of obtained results is presented in chapter 9, while section 9.1 includes method and example on wafer-based visualization, section 9.2 presents method and renders the results on devices. In the chapter 10 our future works are introduced, and finally the conclusion of the thesis is presented in the chapter 11.

2 DEFINITION OF THE PROBLEM

In chapter 1, some challenges that chip foundries and research centers are facing in the development of faster and higher quality chips and their potential solutions using ML are presented.

Using AI for the entire process, from design to final product would be the ultimate goal of automation. However, in today's environment, the exploration of different chip architectures and the lack of sufficient data has prevented ML algorithms from achieving acceptable accuracy. Therefore, defining the problem and developing solutions for each step is of vital importance.

The production of an experimental chip can be divided into the 6 following stages :

1. **Analysis :** Current digital chip applications provide an important market revenue for manufacturers, such as usage cases for end-users, marketing strategies, etc. Utilizing well-researched components and tested architectures is often the preferred and the least risky path. During this phase, experts evaluate other electrical components, consider the shortcomings of current architectures, and try to come up with solutions to these problems and find an optimal solution for clients (most often mobile phone or computer manufacturers). The requirements for experimental architectures are defined in this phase. The choice of materials and the architecture of the transistors are made during this phase.
2. **Projection of Architecture :** The two recutsive steps during this phase are repeated several times.
   (a) **Design :** After the number of layers necessary for the number of cores of the chip are determined, the design of the chip is performed. In this, the material composition of the transistors and the positioning of the transistors are designed by experts. The aim of this design phase is to have the smallest silicon footprint of the chip and the highest density of transistors with minimal heating and almost no or very limited leakage.
   (b) **Simulation :** The chip design is optimized using external tools and then simulated for different operating conditions. This phase involves the use of

SPICE programs and special software tools developed for in-house use to generate usability reports. To obtain acceptable results from new architectures, hundreds of designs have to be made and the simulation results need to be evaluated.

3. **Production :** The designs that fail the simulation or exhibit inadequate performance go back to the previous step. The designs that show potential are moved on to the production phase. The production is performed under cleanroom conditions, mostly autonomously by robots and equipment that have nanometer resolution.

4. **Test :** The real chips are electrically tested with use case scenarios to determine their performance metrics such as heating or leakage spot detection. Id-Vg traces (more info for Id-Vg is presented in section 2.1) are extracted for different use cases such as high-frequency use and transistor output, reliability tests, stress tests, and process compatibility. Given the variables, unexpected results are not uncommon. The reasons for this may range from the quality control phases at early steps in the process fabrication not catching potential errors to unexpected influences of the testing design environment within the chip (difficult to simulate the entire chips composed of billions of transistors). The diligence of the experts and the speed of the tools used during this process have great importance.

5. **Evaluation :** The test results are further submitted to the evaluation of experts. One of these steps is the processing of the Id-Vg graphs using certain methods to extract characteristics such as on-off switching voltages and the frequency analysis of the transistors that the authors explained in (Schroder, 2015; Ortiz-Conde et al., 2002; Fleury et al., 2008). These characteristics give insight into the function of the transistor as well as enable the characterization of the performance of the chip. In this phase, mathematics-based software that is often developed in-house is utilized. The semiconductor expert prepares his findings with the help of these tools. This phase of the process is also very time-consuming.

6. **Deployment :** After these long and often repetitive steps, the chip design that now meets market demands is ready for production. The chip design can be deployed in a limited segment for real-world testing or shared with manufacturers for possible partnerships. Many chip research centers work with partners for mass production, packaging, and shipment similar to IMEC where the experiments presented in this thesis are conducted.

During the production phase each step is specialized and uses state of the art too. These tools often rely on complex calculations and physics based applications (such as Id-Vg characteristics extractions) and can take hours or even days to complete. Even slight improvements in these tools can yield significant profits and enable the creation of new technologies. Therefore, manufacturers invest heavily in their development (Tuv et al., 2018).

In our work, we aim to create a more centralized architecture for the "evaluation" phase of development i.e. create an AI-based solution to minimize the discrepancies between the experts while also expediting the process increasing its accuracy and efficiency.

The evaluation step is one of the most time consuming phases because relies heavily on expert observations. Even several pass experiment consumes a lot of time and it is often necessary to perform them several times. Due to these factors that are costly time wise, we aim to speed up this process to accelerate the development of new technologies.

2.1 MOSFETS

Metal Oxide Semiconductor Field Effect Transistor (MOSFET) is a semiconductor device that is widely utilized in electrical devices for switching purposes and signal amplification. It is a four-terminal device with source (S), gate (G), drain (D), and body (B) terminals, and its operation is dependent on the electrical fluctuations that occur in the channel width as well as the flow of carriers (either holes or electrons). If the charge carriers that enter the channel via the source terminal (Is) do not properly exit via the drain terminal (Id), the MOSFET is called not-healthy or defective. The output of terminals (Id-Vg traces) is stored for evaluation in this step, which is also known as electrical characterization. Fig. 2.2 reflects some examples of I-V (electric response) sample curves taken from routine operations of transistors. Green plots represent the current of the gate, red plots represent the current of the drain, blue plots represent the current of the source, and yellow plots represent the current of the body. Dashed plots represent the first routine where the low voltage is applied, while straight line plots represent the second routine where the high voltage is applied. Because a single experiment may not correctly characterize the component, a secondary experiment with a different voltage level is done, and the two results are interpreted together.

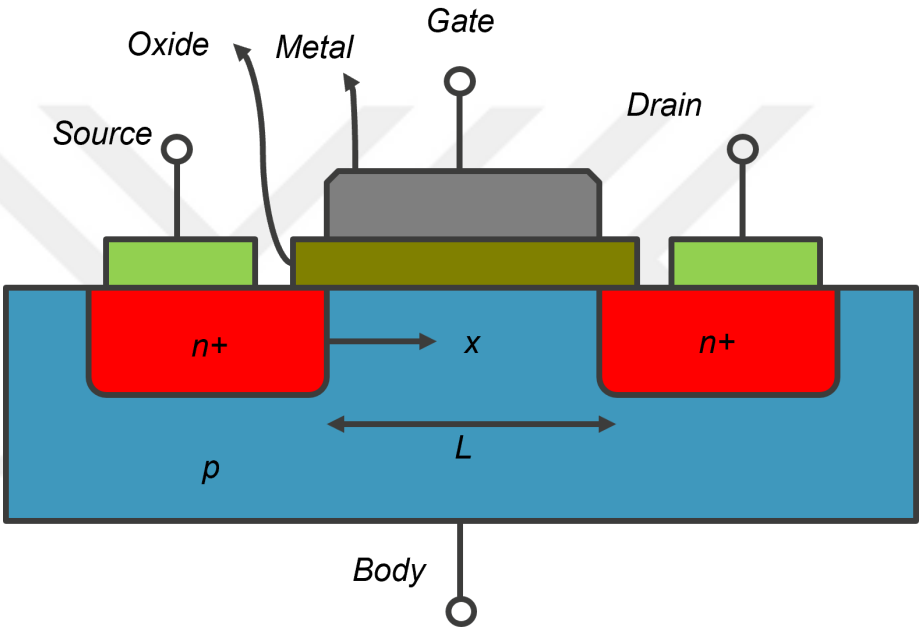


FIGURE 2.1 – An n-type MOSFET illustration. Gate is completely isolated from the rest of the circuit by the insulator oxide. The voltage on the gate creates a field effect where an increment of gate voltage will repel the holes in the substrate away from the area between the source and drain. This area is called the channel region (x) where an inversion layer of electrons starts to form at the source, and as the voltage increases, expands toward to drain. L represents the length of the channel region. After gate voltage passes the threshold voltage, electrons from the source and drain flow in, and form an inversion layer of electrons that connect the source and drain regions. Finally, a voltage that is applied to the source, a current will flow.

The measured Id-Vg data is used to extract parameters that define the functionality and the performance of the chip. The most important parameter is the threshold switch voltage (V_{th}) which defines the voltage level at which the semiconductor switches from a conducting state (on or 1) to a nonconducting state (off or 0). The extraction of these values is performed by a process composed of complex calculations on the Id-Vg curves. As the number of devices increases so does the time and cost of this process.

In the light of the presented information, the time it takes to evaluate the components continues to be a burden for research centers. One of the current solutions to these problems is to increase the computing power of the system. However, as the compute modules do not distribute the workload evenly and because the software is not optimized and complicated, so it is not very suitable for parallelization. Therefore computation time does not decrease linearly with the compute power of the system.

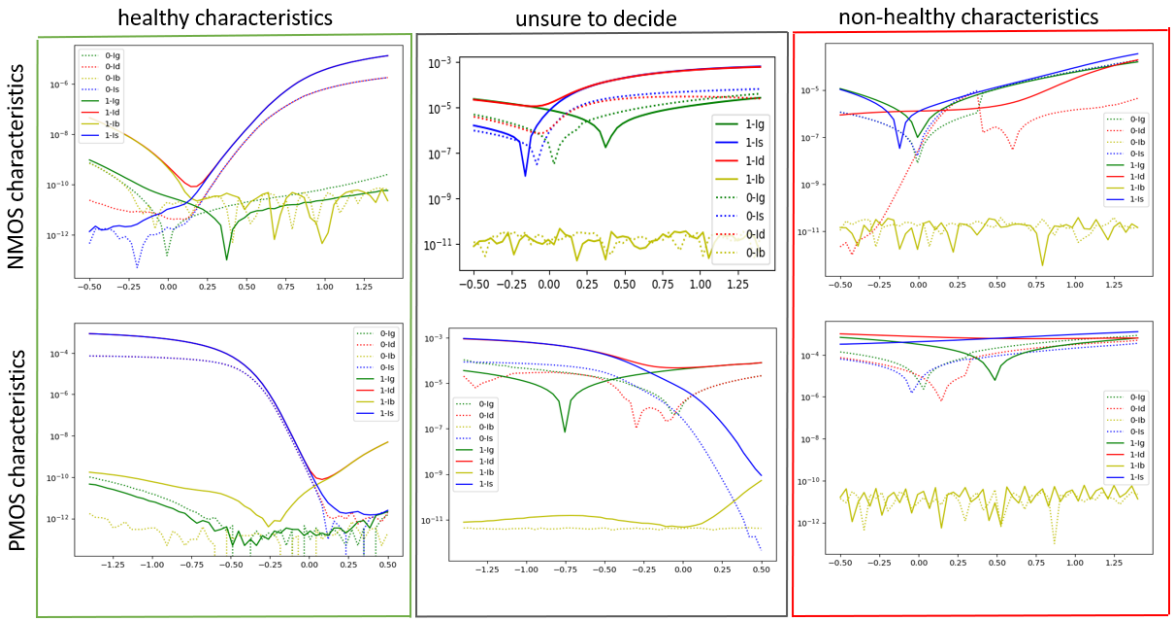


FIGURE 2.2 – 4-terminals I-V characteristics of a FinFET and at two inputs drain voltages (V_{ds}) (0 is V_{ds}= 50mV, and 1 is V_{ds}= 800mV). Healthy, not-healthy, and not-sure labels evaluated by the same expert.

For these reasons, we propose an AI-based system that does not rely on the defined mathematics and the physical constraints of the components. An exhaustive study on new AI architectures, and various AI architectures utilized for similar problems enabled us to select a few candidates. When we examine the chip data, the Id-Vg curves are represented as a single-dimensional signal time series and it is observed that computer-aided simulation of transistors (or transistor modeling) where AI model will be trained with outputs of 4 terminals and will generate new drain current can

fit our purposes. This method can be extended to generate other terminals as well. Due to this observation, initially, we chose to use Long-Short Term Memories (LSTM) algorithm which is a member of the Recurrent Neural Networks family (more details in section 3.1). Another solution is using the Id-Vg curves in signal format from the other terminals to create a two-dimensional time series and use a Convolutional Neural Network architecture (more details in section 3.2).

Both architectures are examined and research for the implementation to this domain is started.

3 LITERATURE REVIEW

We conducted our literature review and related work search for two problems ; RNN-based transistor modeling for simulation of transistors based on their outputs, and a 2-step CNN approach, a multi-model ensemble of CNNs for characteristics filtering, and a CNN model for threshold voltage extraction.

3.1 RNN BASED TRANSISTOR MODELLING

RNNs first emerged from a Ph.D. thesis (Rumelhart et al., 1986) in 1986 and are an ML algorithm based on artificial neural networks. This architecture has been gaining popularity since its introduction. Currently, RNN algorithms are well researched and have proven themselves in multiple domains (Karpathy, 2015), and Long Short-Term Memories (LSTM) is one of the most effective algorithms in the RNN family (details of RNN family algorithms are clearly described in section 5). Until 2019 about 1000 different LSTM-based algorithms have been created and more than 900 of these have been created after 2015 (Smagulova and James, 2019). Therefore LSTM based algorithms and their variants provide accurate results and they have proven their strength.

In general, RNN algorithms have also been implemented in the semiconductor domain for various purposes. However, for the semiconductor domain, their usage is limited and merits further research. An RNN-based approach has been proposed in (Nawaz et al., 2013). The network is trained with vibration data coming from the device. In (Fu et al., 2018), RNNs have been used to detect supply chain problems and predict market demand. In both cases, the RNNs are trained with the data output of other devices and do not provide insight into the internal functions of the semiconductors. In our work, we aim to use time-series data obtained from testing the transistors and simulate a virtual transistor (or transistor modeling).

The authors proposed one of the first ML approaches to transistor modeling in (Meijer, 1996). The key idea is to predict the current voltage curve using only 12 neurons, which are most likely constrained by computational memory. Because it has very limited

memory, some domain knowledge (e.g., gate length between source and drain, silicon width) was coded inside the model during the training process to improve accuracy. In 2017, another approach to transistor modeling and classification using neural networks was presented (Zhang et al., 2017). Physical dependency information and transistor-specific knowledge are used to train a 15-neuron NN model. In (Li et al., 2016) authors used two separate NNs, one for V_{ds} and another for V_g (gate voltage), each with less than 10 neurons. In other words, because their NN models are comprised of a limited number of neurons, this limited number of neurons is insufficient to learn complex information, and the system necessitates manual coding of domain knowledge, limiting the capabilities of the ML.

A compact model of the transistor is usually described as the interface to bring a transistor implementation into the simulators. There are commercial compact modeling tools that use ML techniques and classification based on compact models. One of these is Silvaco TechModeler (Blaesi, 2018). Due to the limited availability of information, we could not get information about the manner of software work.

Contrary to the abovementioned work, we aim to train our model without apriori knowledge and obtain a more generalized system that can be implemented efficiently. Because we intend on obtaining a universal model without hard coded transistor constraints relying solely on Id-Vg outputs. These constraints contain noteworthy inputs about the transistors that result in loss of information if it is not used. Therefore it is important to have a robust algorithm that is capable of functioning with limited data. To the best of the authors' knowledge, an LSTM-based approach is not available and the creation of a generalized LSTM-based approach will be a significant contribution.

3.2 CNN BASED PASS FAIL/CHECK AND KEY PARAMETER EXTRACTION

Early works of Machine Learning-based categorization of transistor characteristics had to be done with very limited resources. The number of applicable ML algorithms was limited, and neural network techniques were incapable of learning very complex tasks due to a lack of computational power and tool support, resulting in the use of fewer neurons and layers in ML architecture. As a result, domain knowledge was used in a number of works to improve ML accuracy and performance, the same as transistor

modeling approaches. While incorporating domain knowledge into the development of NNs improves their accuracy, this type of manual intervention into an NNs function prohibits many of its benefits (e.g. generalizability, autonomous operation).

The authors described the usage of a random forest algorithm combined with domain knowledge to detection of transistor defects in (Teo et al., 2019). This approach, however, could be a disadvantage in this application due to the difficulties of updating itself coming from the nature of the random forest algorithm. Authors presented in (Klemme et al., 2020) a NN with 250 neurons solution that predicts signal outputs without domain knowledge and categorizes transistors based on their outputs. The use of such NNs architectures places constraints on the system's scalability. Furthermore, the data must be preprocessed, which adds additional constraints in the training phase because data limitations reduce the range of variants and the requirement of scaling data creates conflicts between real-world and reprocessed data, as the authors point out.

After the classification process of characteristics, we aim to perform key parameter extraction and there are several ML approaches that extract features from time series data such as Nanopoulos et. al. (Nanopoulos et al., 2001) used one type of NN algorithm, multi-layer perception (MLP) architecture to extract and classify features obtained from time-series data. Kampouraki et. al. (Kampouraki et al., 2008) used a support vector machine (SVM) to time-series data and the authors presented more accurate results than the NN approach. Hsu et. al (Hsu et al., 2021) created Convolutional Neural Network (CNN) model and applied convolutions to time-series data for fault diagnosis and detection. In these 3 studies, new ML algorithms are proposed and their differences are explained.

ML applications have emerged as a widely used optimization method in almost every domain in the past decades including semiconductor manufacturing. In (Irani et al., 1990) experts have used this method as a tool to automate the production line and save time on manual work for the production of devices in ultra-clean environments. Over time, it has also found applications in other areas of the semiconductor industry (Luo, 2013; Teo et al., 2019; Ding et al., 2009). Many semiconductor companies have started to use computer vision-based automated defect detection, manufacturing optimization, and process automation systems to replace manual systems (Tuv et al., 2018). Chip foundries are now researching ways to have different forms of ML using advanced

algorithms in many other parts of the factory (Göke et al., 2021). An example of intense research in this domain can be found in a recent paper from Intel (Tuv et al., 2018) where the defects are detected automatically with high accuracy using ML and computer vision techniques. As a result, ML has become very popular in industrial environments for automatic defect classification, electrical testing, anomaly detection, and performance prediction.

An ML architecture that efficiently uses the dataset is an important aspect of this research and appropriate algorithm creation can be achieved using ML. One of these approaches is a genetic algorithm combined with NNs to find the best architecture of NNs proposed in 2016 (Lamamra and Berrah, 2016). However, the usage of this ML solution is very limited because it can be only applied to very specific transistor characteristics, which would penalize the accuracy and performance (Bousquet et al., 2003) if the same technique was implemented to a different type of transistor.

In summary :

— We propose here a multi model ensemble approach with increased sensitivity for internal circuits of systems operating in common applications.

— We explore the capabilities of our approach on limited data sets such as limited experiments.

— Domain knowledge is not incorporated into the models. Therefore, the solution is more generic and is capable of working with devices ranging from very conventional up to very disruptive ones.

— General purpose i.e., physics agnostic and no need for detailed chip information. Therefore Non-Disclosure Agreements and IP of customers are protected,

4 EXPERIMENTS PREPARATION

For the performance and accuracy analysis of the developed ML architecture, it is important to conduct the experiments using the same datasets under the same condition. Therefore, we researched the standards and best practices for a roadmap.

4.1 ENVIRONMENT SETUP

All experiments that are presented in this thesis are executed in a 4-core 1.7 GHz Intel CPU with no GPU computation support. Our RNN-based and CNN-based ML approaches are coded in Python using tensors in Tensorflow 2 framework for ML functionality. To preprocess, prepare and visualize the data, open source libraries of python were used.

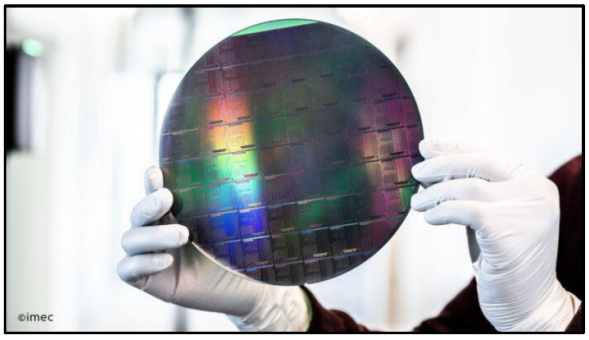
4.2 DATASET SETUP

In this work, the input device characteristics are obtained from a research and development prototype line and the MOSFETs that we have used are manufactured using conventional materials. The 16nm-node (most used today) MOSFET characteristics used in this thesis are represented by Id contained at 5 different measured drain to source voltages (Vds = 50mV, 200mV, 400mV, 600mV, 800mV) and 4 terminals (Id as drain voltage, Is as source voltage, Ig as gate voltage, Ib as body voltage) outputs recorded from the terminals of the transistor. Current of each terminal is recorded every 0.5 seconds and the total record consists of 51 data points (25 seconds total).

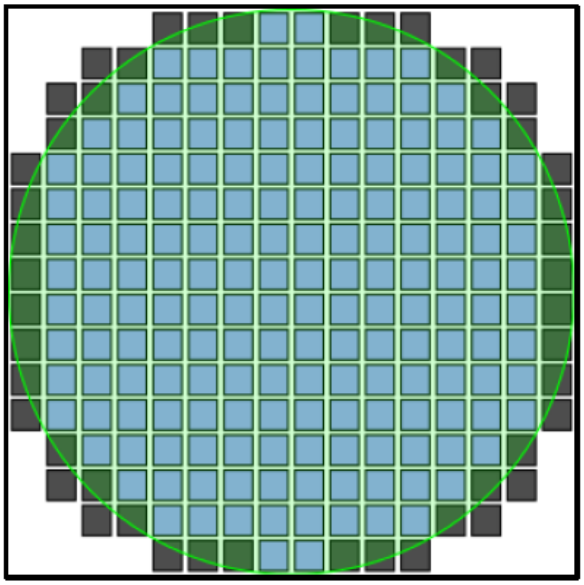
To evaluate the performance of the CNN model the data was evaluated by experts observing the Id-Vg graphs of the transistors and classified as healthy (no leaks or leaks that can be tolerated) or not healthy (leaks that can not be tolerated). Only transistors that were unanimously labeled as healthy were selected as training data. Then the V_{th} values of the transistors were calculated to form a second dataset to be used by our V_{th} extractor network.

The entire dataset consists of 59 devices with different designs (LG arrays), each device

FIGURE 4.1 – A few examples of very advanced measuring equipment that includes nanometric level sensitivity for electrical and material characterization is widely used in research centers.



(a) A photo of 300mm wafer.



(b) A schema of a wafer map visualized by software. A wafer includes multiple chips, and blue squares reflect chip locations on the wafer.

FIGURE 4.2 – Wafer images.

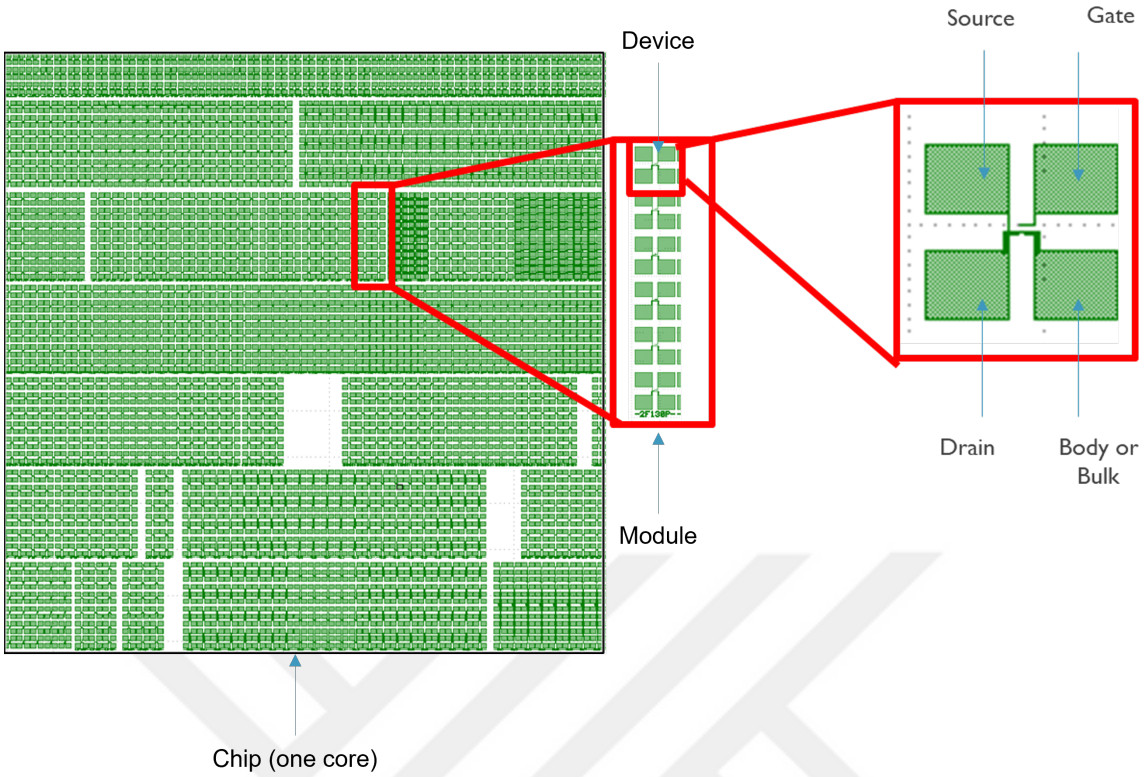


FIGURE 4.3 – Each chip includes multiple modules, and each modules comprised of multiple devices that consists 4 terminal ; drain, source, gate, and body (or bulk).

design comes from 38 different dies within the 300mm wafers and finally each transistor is electrically characterized at 5 different drain to source voltages (Vds). Our dataset, therefore, consists of 11210 samples of which 75% (8408 samples) are used for training and 25% (2802 samples) is used for testing. To assure the accuracy of the model by providing inaccurate labels, the dataset was cleansed of false data, also dubious data points were also eliminated from the dataset.

The training data set dropped to 6760 samples after a first expert evaluation. For each iteration of training (epochs), we decided to use random cross-validation to create a different validation dataset and measure the performance of the current epoch. Cross-validation is a resampling method that trains and tests a model on multiple iterations using different portions of the data chosen randomly. It's used for measuring how well a model prediction will perform after each iteration. Feature extraction and validation were performed multiple times on different parts of the dataset reducing overfitting (Hawkins, 2004).

After the cleaning and reprocessing phases, the dataset was augmented to 3 datasets. The first dataset which doesn't have any expert evaluation separated for the RNN-

based approach. For the second, unanimous expert votes were used to classify the data into binary classes (0 as healthy, 1 as not-healthy) containing an equal number of healthy and non-healthy data to have a balanced dataset. For the third dataset, the V_{th} values are calculated to be used as prediction values. As the V_{th} values are widely used in the industry, they were obtained using the peak-gm method (Schroder, 2015).

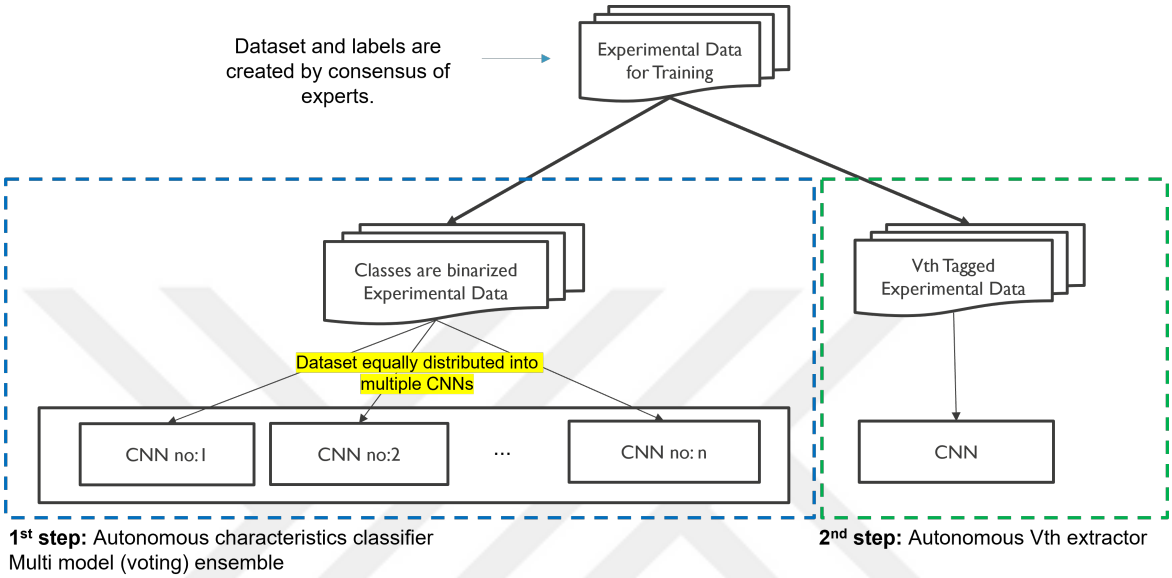


FIGURE 4.4 – Illustration of dataset operations for creating training dataset for each ML model.

4.3 OVERFITTING PREVENTION

The overfitting problem occurs from a lack of variety in data. Overfitting can be described as when a neural network memorizes the training data rather than finding information features. As a result, the model achieves perfect accuracy on training data while performing poorly on unseen test data. This problem was first pointed out in (Hawkins, 2004) as "...the principle of parsimony, calls for using models and procedures that contain all that is necessary for the modeling but nothing more. Overfitting is the use of models or procedures that violate parsimony...". The authors mentioned overfitting types and their solutions in (Ying, 2019).

Overfitting manifests itself as a model with poor generalization. Generalizability refers to the performance metric difference of a model when evaluated on training data (the data it has previously seen) versus testing data (the data it has never seen before). We utilize overfitting prevention methods on our ML models, which are data augmentation, dropout layers and early stopping technique.

Data augmentation is a technique for improving generalization. The augmented data will represent a more comprehensive set of possible data points, reducing the distance between the training and validation sets, as well as any future testing sets (Shorten and Khoshgoftaar, 2019). Yet another method Dropout was introduced for the first time in 2014 (Srivastava et al., 2014). Dropout is a straightforward but effective overfitting prevention technique for neural networks. The basic idea is based on randomly disabling some neuron units and their connections in hidden and visible layers in each epoch (number of passes of the entire training dataset completed by the ML algorithm), so the trained neural network becomes a union of thinned networks with extensive weight sharing, where each thinned network is trained at least once with all of the data.

Finally, we set callback points in all ML models to save the entire network weights and record validation loss after each epoch. To avoid overfitting, we stop training after the validation loss stops decreasing and reset the weights to the most recent callback with the lowest validation loss. While validation loss indicates how well features from training data fit new data, other metrics such as accuracy and training loss can also be used.

These 3 techniques are utilized in all of our approaches with further details provided below the description of each approach.

5 RECURRENT NEURAL NETWORKS BASED TRANSISTOR MODELLING

Our reasoning for selecting the LSTM algorithm which is a form of RNN is explained in section 2, and similar work using RNNs is presented in section 3.1. In this section,

we will explain the principles of the LSTM algorithm and present our results.

RNN works on time variant data and is able to predict the following steps, detect errors or simulate systems. Machine translation (Vaswani et al., 2017), speech recognition (Sak et al., 2014), object detection on video (Dasiopoulou et al., 2005) are some state-of-the-art examples of RNN based approaches in different domains.

Given an input sequence $x = (x_1, ..., x_\mathrm{T})$, a classic RNN computes the hidden vector sequence $h = (h_1, ..., h_\mathrm{T})$ and output vector sequence $y = (y_1, ..., y_\mathrm{T})$ by iterating the following equations from $t = 1$ to T :

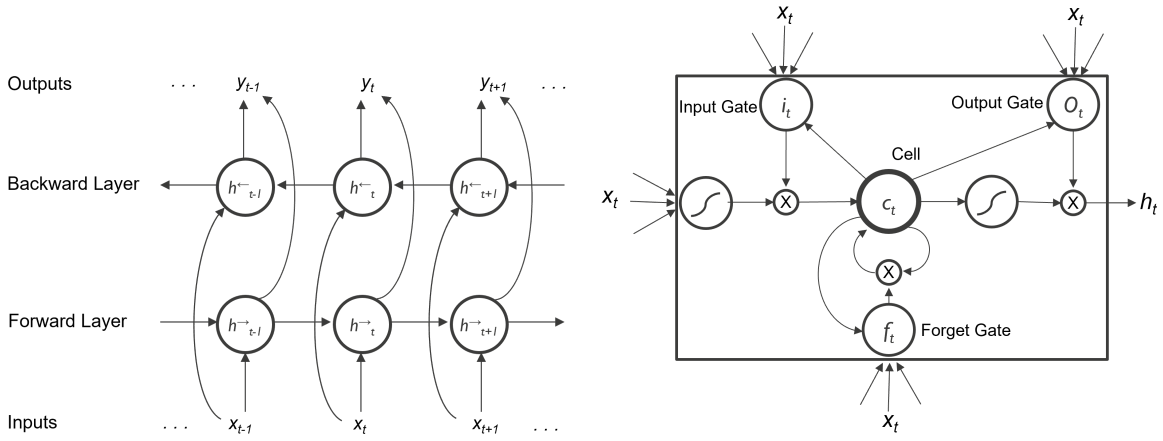
$$h_\mathrm{t} = H(W_\mathrm{xh} x_\mathrm{t} + W_\mathrm{hh} h_\mathrm{t\text{-}1} + b_\mathrm{h})$$

$$y_\mathrm{t} = W_\mathrm{hy} h_\mathrm{t} + b_\mathrm{y}$$

where the W terms represent weight matrices (e.g. W_xh is the input-hidden weight matrix), the b terms represent bias vectors (e.g. b_h is hidden bias vector) and H is the hidden layer function (Graves et al., 2013).

The major distinguishing factor of RNNs from other architectures that use feedforward neurons is its memory module (internal state) that enables it to track the time wise change of the data and extract features from it (Zell, 1994). This enables RNNs to interpret and process dynamic behavior. Different variants of the RNN algorithm such as fully connected RNN (classic RNN), independently RNN (Li et al., 2018), and bi-directional RNN (Graves et al., 2005) are used for different purposes.

As the complexity of time-series data increases, extracting relevant information becomes harder. Research is developing new architectures where classical RNN architectures cease to meet their needs. The classical RNN algorithm can also suffer from the



(a) Bidirectional RNN (b) A long short term memories cell

FIGURE 5.1 – Our BiLSTM method combines two techniques (Graves et al., 2013).

vanishing gradient problem (during each iteration of training, neurons receive vanishingly small weight updates due to its proportional to the partial derivative of the error function) (Hochreiter et al., 2001) because of its finite-precision number usage on gradients. LSTMs, one of the special kinds of RNNs (Hochreiter and Schmidhuber, 1997), has a gate structure that avoids the vanishing gradient problem. In addition, it permits gradients to flow between neurons therefore relevant information in the context and error weights can be updated with avoiding long-term dependency.

Another shortcoming of classical RNNs is that they are only able to make use of the previous context. Bidirectional RNNs (BRNNs) do this by processing the data in both directions with two separated hidden layers, which are then fed forwards to the same output layer. As illustrated in Fig. 5.1a, a BRNN computes the forward hidden sequence $\mathrm{h}^{\rightarrow}$, the backward hidden sequence $\mathrm{h}^{\leftarrow}$ and the output sequence y by iterating the backward layer from $t = T$ to 1, the forward layer from $t = 1$ to T and then updating the output layer. Combing BRNNs with LSTM gives bidirectional LSTM (Graves et al., 2005), which can access long-range context in both input directions.

As described in section 4.2, our dataset consists of voltage data collected during 2 seconds consisting of 51 data points. As it is time consuming to evaluate each experiment with each device for the entire 2 seconds, the first 2.5 seconds (5 data points) are processed with the current technique in the semiconductor domain which is the gm peak method and the remaining 22.5 seconds (46 data points) is simulated using LSTM with a forget-gate. This approach yielded a 10-fold increase in speed reducing

time cost by 91%. Depending on the experimental results this separation is optimized.

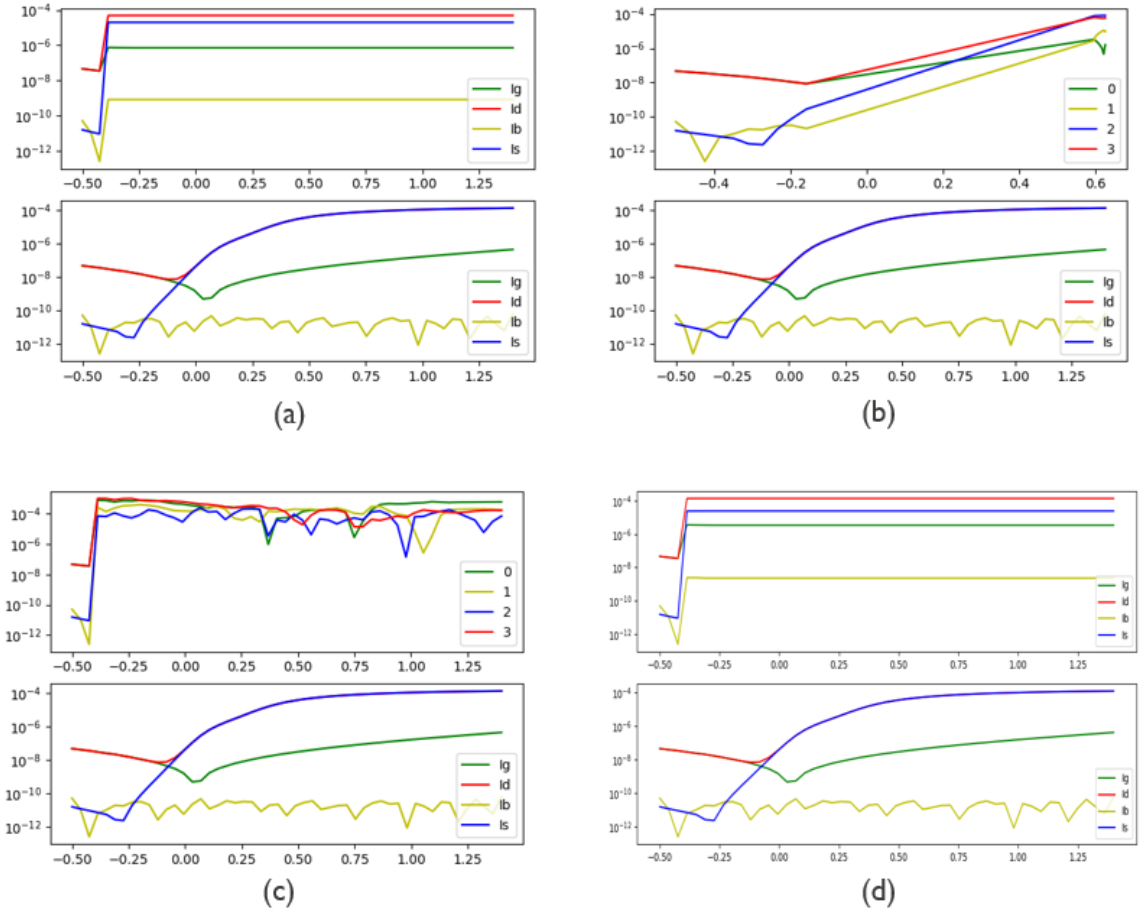


FIGURE 5.2 – A few example from the research towards building an accurate LSTM architecture. Green lines represent gate currents, red lines represent drain currents, yellow lines represent body currents, and finally, blue lines represent source currents. The structure as is follows ; (a) : upper is single layered bidirectional LSTM prediction, lower is actual values ; (b) : upper is single layered univariate convolutional LSTM prediction, lower is actual values ; (c) : upper is single layered multivariate convolutional LSTM prediction, lower is actual values ; (d) : upper is classic LSTM prediction, lower is actual values.

According to this plan, multiple single-layer and multi-layer LSTM architectures were implemented and mean-squared error results were recorded. For this application, it was observed that single-layer architectures exhibit overfitting despite hyperparameter optimization and the use of different activation functions, loss functions, and optimizers.

Some examples are provided in Fig. 5.2. In Fig. 5.2, the first 5 points of terminals

are given by us, and the rest are fulfilled by predictions. Therefore it is predicted that increasing the models' capacity will yield positive results. An extensive search on Multi-layered Bi-LSTM (Stacked Bi-LSTM) where input processed by two layered Bi-LSTM

modules was done and ML architecture built with respect to our inferences.

5.1 DATA REPROCESSING

The details of the dataset presented in section 4.2 are examined and data flow for the system is conceived. The data is transformed to a stack format for the LSTM where 5 data points will enter the LSTM model and the 6th point will be predicted. The new point is added to the end of the data for prediction and the process is iterated until the 51st point. This enabled us to predict the next step with 2.5 seconds of granularity. This data transformation is illustrated in Fig 5.3.

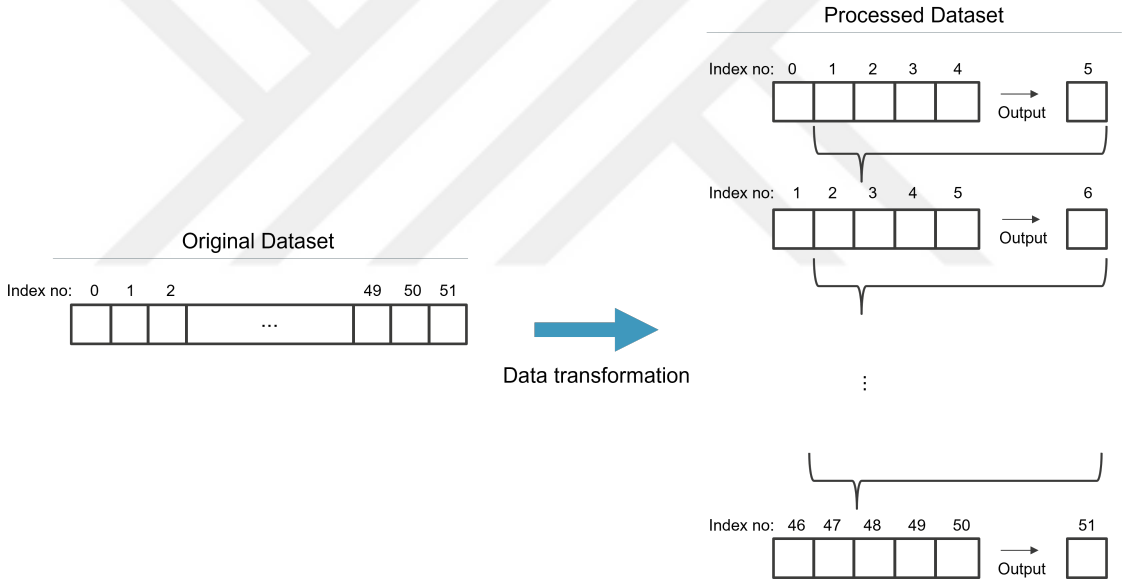


FIGURE 5.3 – Preprocessing of the raw dataset for our LSTM architecture.

5.2 MODEL DETAILS

Section 5 describes our search for the appropriate architecture. Our model has 2 layers consisting of 5 interconnected Bidirectional LSTM modules. Every layer utilizes the ReLU activation function (Glorot et al., 2011). 30% dropout rate is used to avoid overfitting. We train 500 randomly initialized models and select the best performing model. The mean-squared error rate is used as the loss function and ADAM (Kingma and Ba, 2014) optimizer is used as the optimization algorithm. The number of epochs

was limited to 6000 and an early stopping algorithm was utilized to halt training if no progress is observed for 500 epochs. The training process is completed in approximately 2 hours and 5.4 shows an example simulation of transistor modeling.

5.3 EXPERIMENT RESULTS

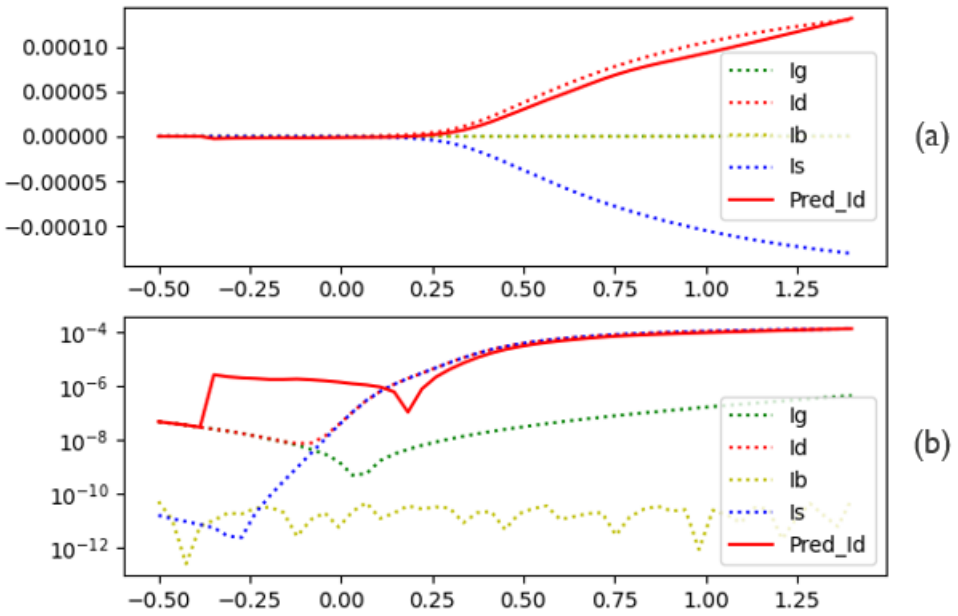


FIGURE 5.4 – An example of transistor modeling result by bidirectional LSTM architecture. Green dotted plots represent actual gate currents, yellow dotted plots represent actual body currents, blue dotted plots represent actual source currents, and red plots represent drain currents while dotted plots represent actual drain currents, line plots represent predicted drain currents. (a) : predicted and actual results in linear scale (b) : predicted and actual results in log10 scale

After the training phase, the performance of the RNN-based transistor model was examined, providing 5 data points to the model and predicting the following 46 points. When the results depicted in Fig. 5.4 are examined using a linear scale. We observe that the critical segment following the first activation of the drain is not simulated or predicted to a satisfactory level. When the same results are viewed in the Log_{10} scale, the prediction of the phase preceding the drain activation is also unsatisfactory. Time wise, the simulation of the 46 data points is in the order of milliseconds and the entire dataset consisting of 2802 samples is completed in 2 seconds.

After evaluating different model architectures, we conclude that the dataset is not large enough or diverse enough to yield satisfactory results. As new data from experimental architectures is obtained, revisiting this method is envisaged. An alternative CNN-based approach will be examined in the following sections.

6 2-STEP CONVOLUTIONAL NEURAL NETWORKS BASED APPROACH

Using traditional methods, transistor functionality is evaluated by analytical approaches driven by physics-based formulas or looked at "visually" by experts. Our approach is a two step evaluation process. First, the IV (MOSFETs current at terminals) graphs of the components are evaluated by multiple Convolutional Neural Networks using images of plotted data similar to what experts view in the first-level evaluation of the transistors. Then, these ML models utilize a multi-model ensemble technique to ensure stringent sanity checks of the functionality of transistors. The prediction accuracy of our CNN-based results is comparable to those given by highly-experienced evaluators. Then we conducted research to create a second CNN-based ML approach to extract the threshold voltage of MOSFETs after the first filtering process. Note that, the first filtering process aims to remove Id-Vg traces of MOSFETs with low ON and OFF state currents or anomalous Id-Vg characteristics (high noise, high gate current, not stable drain and source current, etc.).

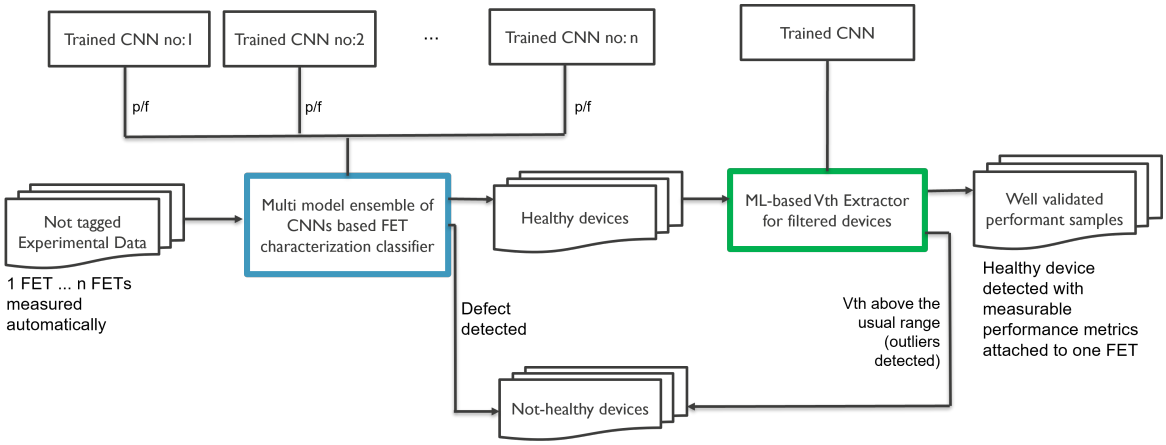


FIGURE 6.1 – Finalized system structure and data flow.

In more detail, our data-set contains multiple output terminals as mentioned in section 4.2. Our choice was to leverage algorithms that are capable of examining multiple channels concurrently. After an extensive literature survey, the implementation of CNN architectures (He et al., 2016; Huang et al., 2020) has emerged as the most appropriate architecture for our task, therefore we opted to utilize CNN on both phases of the

problem. CNN is one of the popular deep learning (DL) algorithms that can take in 2D data (e.g. image, time-series), tune its network automatically with learnable weights and biases from various aspects of the data, and be able to differentiate one from the other using common extracted features (LeCun et al., 1995). Therefore we propose 2 ML approaches based on CNN which does not strictly encode any domain knowledge into the model and works sequentially. The first CNN is configured as a classifier that initially filters the signals and detects not defective (healthy) devices. The second CNN architecture uses the filtered outputs of the first CNN and calculates the threshold voltage (V_{th}) of the devices.

6.1 THE CONVOLUTION OPERATION

Convolution is a mathematical operation on two functions that results in a third function that expresses how the function is affected by the other function. Typically, It is described with an asterisk :

$$F = (I * K)$$

where i and k are functions, and f is produced as the third function. In convolutional neural network terminology, the first function (function I) to the convolution is usually described as the input while the second function (function K) as the kernel. The output is often referred to as the output, or as the feature map (function F).

Our application uses discrete and multidimensional array shaped data as well as the kernel which is adapted by the learning algorithm. These multidimensional arrays of data will be called tensors. The convolution operations run over the tensors more than one axis at a time. The result of using 2D kernel K over 2D time-series data I :

$$F(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

while (i, j) represents the position of input, m represents the number of columns of the kernel, and n represents the number of rows of the kernel. An illustration of typical convolution is demonstrated in fig. 6.2.

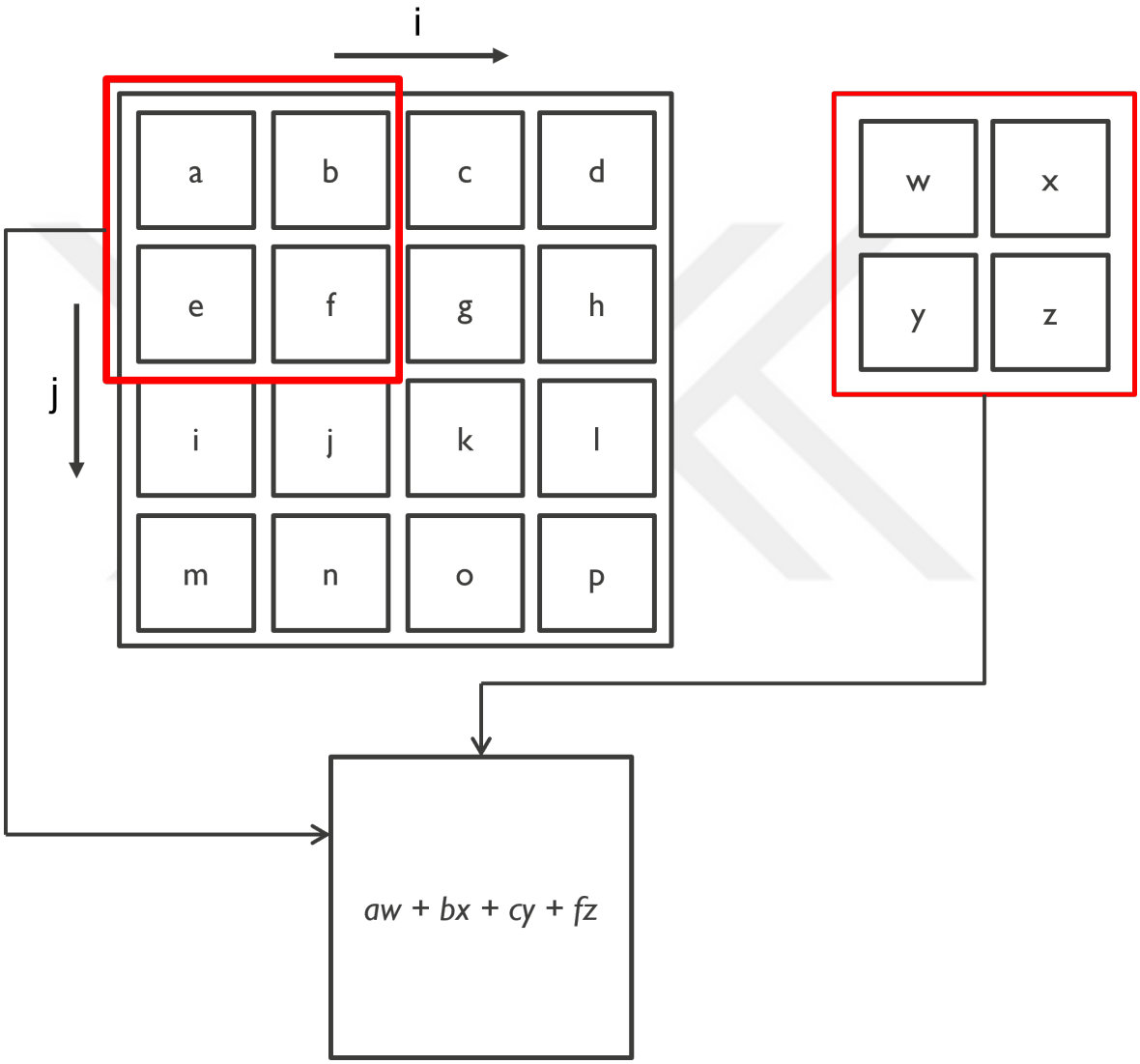


FIGURE 6.2 – An example of 2D kernel convolution on 2D input data. The upper-left element of the output tensor is created by applying the kernel to the matching upper-left portion of the input tensor, as shown by the box with arrows.

6.2 SUMMARY OF WORK

We propose a new methodology for the semiconductor industry to verify the quality of transistors and extract the on-off switch voltage, called threshold voltage (V_{th}). In traditional V_{th} extraction methods, scripting language has to be deployed leveraging basic extraction technique (Ortiz-Conde et al., 2002) or quite advanced one (Fleury et al., 2008). This process is also most of the time used on unhealthy devices like on MOSFET with high source and gate for instance. In the end, due to a lack of pre-data filtering the simple extraction of V_{th} becomes time consuming and drastically slows down the learning curve.

In our flow, the devices are first classified as healthy (no or minimum leaks) or not-healthy (leak is above the acceptable threshold) based on their statistical distribution evaluated by a pool of experienced scientists. The classification of devices into two categories as described before measuring the quality or the success of an experiment facilitates the development of the next generation of MOS-based switches for the semiconductor industry.

7 1ST STEP : DEVICE CHARACTERISTICS CLASSIFIER

We present a multi-model ensemble of CNNs approach that trained with I-V curves of transistors for classifying transistor characteristics based on curve information in order to detect leaks and potential defects. Because of its faster response, if the CNN model can accurately classify transistors as healthy (no or minimal leaks) or not-healthy (leak current above the acceptable threshold), it is sufficient to use as a pass/fail checker in the quality control process.

the lack of time to digest info manifests itself as increased false positives in the classification of the data and this may cause misdirected results during the decision-making phase that we have mentioned in the previous sections. Along with our ML approach, the system was also designed to provide double-check control to the experts as an auxiliary tool that informs the expert when they decide against the results generated by our model.

An approach using multiple models was investigated instead of relying on a single model for component classification, as the detection of substandard components is important for chip manufacturers. This method mimics the quality control of critical systems, in which the same data is evaluated by multiple experts.

The creation of a CNN consortium of the odd number of models is aimed so that the model votes would not be equal. With the growth of the data set, as the number of CNN models increases, prediction sensitivity can be strengthened. When the current requirements and the size of the data set were examined, it was observed that 3 models would be sufficient. For the reasons abovementioned, 3 CNNs were trained and a CNN consortium was formed. The models evaluate the input data and each model casts a vote. A majority vote for this application was not considered a viable option because according to one CNN, claiming that the component is unfit for purpose is enough to discard it. As a final result, only unanimous votes were accepted, and non-unanimous votes were labeled as not-sure since the problem requires certainty. This approach increased the reliability of the categorization results of transistors.

7.1 DATA REPROCESSING FOR TRAINING

I-V curves from transistors are labeled and well-validated by multiple experts in the field. Curves that cause confusion are also excluded by experts. The total number of devices in the dataset is 6760. However, 3200 curves are suitable for this operation. With 70% of the dataset separated for the training process (2560 samples) and 30% of total curves (640 samples) set aside for validation. We use the k-fold cross validation technique to generate validation data and create a new validation dataset for each iteration. To avoid an unbalanced data problem, the entire dataset is divided into binary classes, with half of it labeled as healthy and the other half as not-healthy.

Memory management is a crucial consideration for training neural networks. Optimizing memory usage not only reduces training time, but can also help you to avoid problems like overfitting. Reducing the dimensions of the inputs is one of the most common techniques for reducing memory usage. Bi-linear interpolation (Press et al., 2007) was applied to resize the images to 120px, 160px to achieve a 16 fold size reduction. For further reduction of memory usage, the images were grayscaled to reduce the dimensions by another 2/3.

7.2 CHOOSING THE RIGHT ACTIVATION FUNCTION

Artificial Neural Networks (ANN) are one type of deep learning algorithm and multi-level representation technique that aims to represent raw data at higher abstraction levels. Many of these transformations generate learned complex functions. More complex and multi-layered neural network models that can produce more accurate results in tasks such as regression, classification, and clustering have begun to emerge as the computational power of processors has increased. The difficulty of developing algorithms to effectively extract the patterns in the data is, however, a critical issue of using deep neural network architectures, and studies on these issues related to neural network training have been a key research area so far.

Rectified Linear Unit (ReLU) function is one of the keys to the recent successes of neural networks. Several studies (He et al., 2015; Ramachandran et al., 2017) show us that ReLU outperforms sigmoid, linear, and hyperbolic functions and produces better accuracy due to their vanishing gradient problem. Despite these benefits, ReLU may

also suffer from dying ReLU problem (Lu et al., 2019) and can still be improved. Instead of zeroing out the negative inputs as regular ReLU does, parametric ReLU (PReLU) multiplies the negative input by a small value (slope parameter learned during back-propagation) and keeps the positive input as is. We decided to use PReLU for these benefits as an activation function of Conv2D layers and Dense layers in our architecture.

7.3 MODEL DETAILS

The IV graphs of the transistors are currently evaluated using humans, human vision, and prior experience to be more precise we chose an approach that mimics expert reasoning to classify good or bad IVs transistor characteristics globally. For the task of detecting transistor defects, we chose a CNN architecture which has proven its efficiency in pattern recognition, and image classification tasks, and the models are not strictly encoded with domain knowledge.

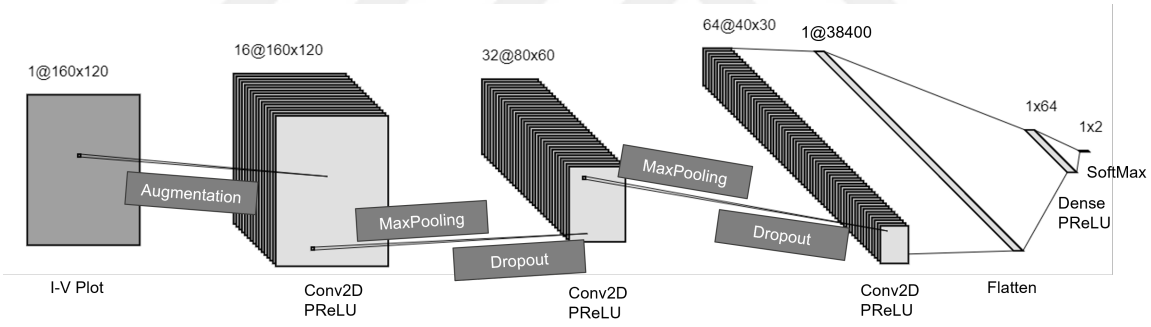


FIGURE 7.1 – Our classifier architecture is visualized by NN-SVG (LeNail, 2019) technique.

We use multi-layered CNNs as a model architecture. The structure of the model starts with a preprocessing layer for scaling the input data to a predefined size and depth, a data augmentation layer for reducing the over-fitting probability and forcing the model to find features, two connected Conv2D layers with Maxpooling, Dropout layers with Parametric Rectified Linear Unit (PReLU) (He et al., 2015) activator, a flatten layer to transform 2D features into 1D shape, a dropout, and a dense layer with softmax activator to get probability result. We used ADAM optimizer and Sparse Categorical Crossentropy as loss metrics. Complete model architecture is depicted in Fig 7.1.

In the data augmentation layer, at least one of the following image manipulation techniques is applied to the images that are randomly selected from training data : flip

(both vertical and horizontal), rotate, zoom-in or zoom-out, and crop random (5px, 5px) piece of the image. We have focused on the model learning not useful features of the image and forcing it to learn the shape feature in IV curves. In addition, we have increased the variance of our data which makes it difficult for the model to memorize the data and allows it to perform more accurate predictions during the testing.

7.4 EXPERIMENT RESULTS

The dataset that we have used for training and testing our ML models is described in detail in section 4.2 and the training details described in section 7.3 and 7.2. Fig. 7.2 shows training and validation loss over epochs of a single CNN model as an example. We see that after 30^{th} epoch there is a sharp decrease in loss and loss is very close to 0 after the 70^{th} epoch, meaning the CNN model was able to match the features with low loss.

Table 7.1 shows the accuracy, precision, sensitivity, and F1 scores over three different wafers and the mean of all wafers after 70 epochs of training which took 12h depending on the number of input samples. While a longer training time can overcome the difficulty of finding features with more processing power, it does not guarantee continuous improvement of the model. After 70^{th} epoch no improvement was seen. Therefore, the training process was finished at the 70^{th} epoch.

TABLE 7.1 – Confusion matrix scores of single CNN classifier, including the mean of all test cases.

Product Index	Accuracy	Precision	Sensitivity	F1 Score
1	0.832	0.971	0.554	0.705
2	0.805	0.980	0.536	0.692
3	0.893	0.989	0,625	0.765
Mean :	0.843	0.980	0.572	0.721

Categorizing the FinFET characteristics relies on expertise in this domain and experts can find different features on curves with years of experience and categorize them accordingly. When these categorizations are compared, although they show huge similarity to a great extent, there may be signal outputs that we can describe as gray areas, which

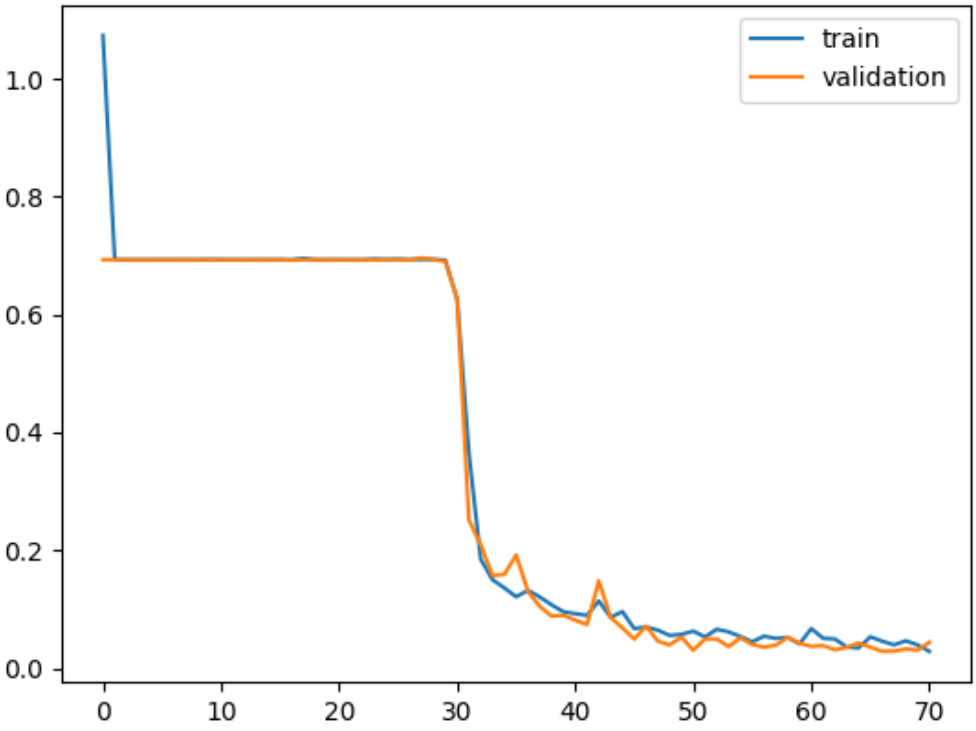


FIGURE 7.2 – Training and validation loss over the epochs, while the y-axis represents loss and the x-axis, represents the number of epochs.

is open to debate regarding the presence of a clear defect. Therefore, It was needed to compare our CNN model with experts with 2 different levels of experience with the same test set.

TABLE 7.2 – Confusion matrix scores of multi model ensemble of CNNs approach.

Wafer Index	Accuracy	Precision	Sensitivity	F1 Score
18	0.903	0.989	0.633	0.771
17	0.871	0.996	0.591	0.742
14	0.940	1.000	0.716	0.834
Mean :	0.904	0.995	0.647	0.782

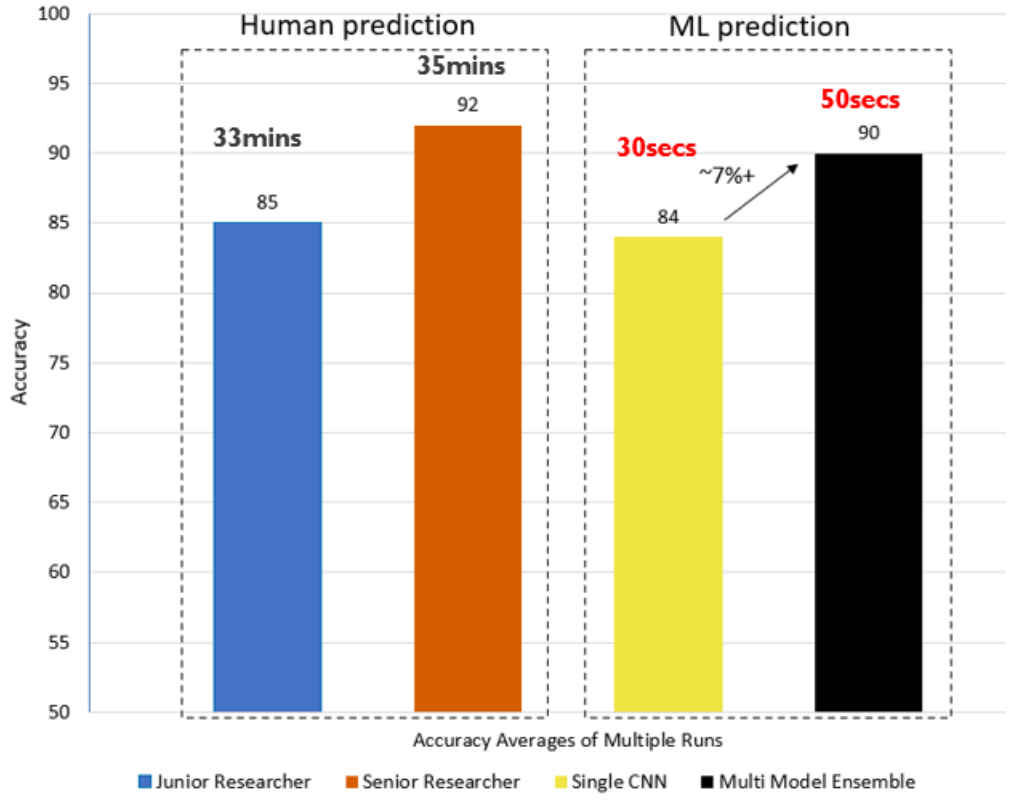


FIGURE 7.3 – Comparative accuracy results of experts and our CNN architectures.

In Fig. 7.3, we present the comparative accuracy performances of the average prediction results of the human predictions, and ML predictions for the I-V curve categorization taken from 3 different wafers can be seen. When the results are investigated, it is seen that a single CNN model (84% accuracy) can categorize similar to a junior-level expert in the field (85% accuracy). Utilizing the proposed multiple CNN model technique, we reached accuracy comparable to a senior-level expert. Although there may be changes in the ratios according to the wafer type, when we examine the average, it can be seen that our multi model ensemble technique generates 90% accuracy, and surpasses the

junior experts whose has 85% accuracy while it is comparable with the senior-level expert whose has 92% accuracy performance.

Time-cost comparison of ML and human prediction demonstrated a different inference. The average time cost of ML approaches for completing their predictions is approximately 85 times faster than the average of human approaches which results in reducing the processes from minutes to seconds. The elapsed time of both experts to complete their prediction task is similar, and there is a 7% accuracy difference between them. When we compare the ML approaches, again, there is a similar time difference, and we can reach the expert level accuracy for an additional 20 seconds with our multi-model ensemble technique. This research has proven that the performance of our classifier is fast and efficient enough for daily industrial usage, while it has shown that using multiple models in a consensus is more effective than the single CNN approach.

8 2ND STEP : THRESHOLD VOLTAGE EXTRACTOR

V_{th} is the minimum needed voltage to create a path between source and drain terminals which means the point of switching the transistor state from off to on, or on to off. While this characteristic provides valuable information about the transistor it also provides insight into the performance of the transistor. There are several methods for calculating V_{th} depending on the manufacturer or customer's preferences. While some of these methods are less computationally intensive and require fewer parameters to operate, generally their precision is also affected. As the mathematical complexity and the device design parameters increase, the computation time can be impacted severely. One of the most common methods for calculating V_{th} is the peak-gm method (Schroder, 2015). This method uses the first derivative of drain or source to calculate the V_{th} value. While this method provides accurate results, it requires somewhat computation power and therefore costs time which is detrimental to the learning of the manufacturer.

Our ML classifier uses curve information as input and detects the healthy i.e functional transistors, completely bypassing the V_{th} calculation step which shortens computation time. This step also provides valuable information about the defect distribution on the wafer. As our approach mimics human expertise, the not functional samples can be eliminated very quickly, in order to quantify the performance of the healthy samples in further detail we propose the use of the V_{th} extractor model. As the extracted V_{th} value is independent of the device, design, or Vds voltage input can be simpler and faster. The calculation of the V_{th} value can be calculated in the order of seconds instead of minutes with this approach, enabling almost instantaneous extraction of one of the most important device parameters which is the threshold voltage. Also, the predictions enable fast performance analysis of the transistors where the V_{th} values can be compared to the ideal V_{th} values to access a better understanding of the performance gap of a technology or an unexpected impact of intrinsic or extrinsic defects owned by the device.

Although one can make a choice between these mathematical analytical models, they sometimes require information about the physical characteristics of the transistors such as gate length, gate width, or the number of fins in advanced device architectures like

FinFETs. This characteristic information is sometimes confidential and may even cause IP issues. Also, V_{th} extraction is not guaranteed for every architecture and Vds. For instance, in the case of the peak-gm technique, there is no maximum transconductance at high Vds making the extraction unreliable and then the gap in the device analysis and time loss. An example presented in Fig. 8.1.

Using our proposed method described in section 7, the defect-free devices can be filtered and key parameter extraction is performed on the healthy devices. In addition to the time savings, our model extracts V_{th} in a fully autonomous manner curve after curve. At the same time, while evaluating extraction with the ML-based model, we significantly reduce the false positive rate of the CNN-based characteristic classifier (e.g., we expect a transistor classified as healthy to not generate abnormal values). This is indicative of the erroneous performance of our CNN-based classifier. This error is rectified by our ML-based model and the error is noted to improve our model in later training phases.

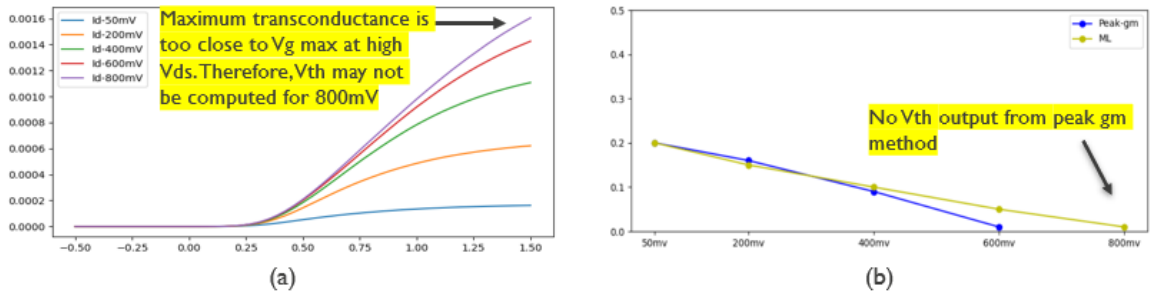


FIGURE 8.1 – An illustration of how our ML method can be used to overcome the shortcomings of the peak-gm method. (a) : Device routine outputs in linear scale, (b) : V_{th} over Vds outputs obtained from peak-gm and our extractor. Since the peak-gm method does not follow any sequence information among Vds, It can generate outputs that can lead to confusion, while our ML approach considers a correlation between Vds and V_{th}.

8.1 TOWARDS BUILDING AN ACCURATE EXTRACTOR MODEL

It is not too realistic to imagine that a single universal architecture will fit all problems in modern machine learning. The architecture needs to be tailored according to the problem, the approach, the data, and the output of the network. Novel architectures are constantly being proposed. Even identical architectures can have vastly different results with slight hyper-parameter changes. In most cases, the same architecture with the

same parameters and the same dataset can yield different results due to the randomness of the training procedure and the features learned from the data. For this reason, we performed 2 independent experiments to obtain the best architecture and loss metric.

8.1.1 CHOOSING THE RIGHT ML ARCHITECTURE

3.2 and 6 sections discuss why we chose to use a CNN architecture. The number of layers and configurations of a CNN architecture can be considered infinite. But a couple of CNN architectures stand out from the others at this time (He et al., 2016; Huang et al., 2020). We chose to implement nomerous different CNN architectures that were capable of utilizing our data type and generating an output of the necessary dimensions. Each architecture with a different layer structure and different hyperparameters was trained on the same data using the same (MSE) loss metric. Some examples of models and the results can be found in section 8.2. Hyperparameters are parameters that are explicitly defined by scientists to control the training process of the ML model. Hyperparameters are used to improve the model's learning, and their values are set before the model's training begins. The number of training iterations (epochs), and train-validation dataset ratio are some common examples of hyperparameters.

During training, the number of passes of the entire training dataset the ML algorithm will be completed was fixed and early stopping was utilized to stop the training process when no further improvement in the loss was observed for 30 epochs. During the entire run of the training phase, the best model was always saved should the training decrease the performance. The number of epochs is a good indication of how fast a certain architecture and optimization function converges to a solution.

The following colors were assigned to each of the architectures :

TABLE 8.1 – Table of CNN architectures that have been used.

Code color	No. of 2D kernels	Dropout	No. of 2D kernels	Maxpooling	Dropout	No. of 1D kernels	No. of NN layers
Blue	32	X	X	✓	X	64	4
Orange	32	X	X	✓	✓	64	4
Green	32	✓	X	✓	✓	64	4
Red	32	X	64	X	X	X	4
Purple	32	X	X	X	X	X	4
Brown	64	X	X	X	X	X	4
Orange	32	X	X	X	X	X	4 (NNs are halved)
Gray	32	X	X	X	X	X	3 (NNs are quartered)

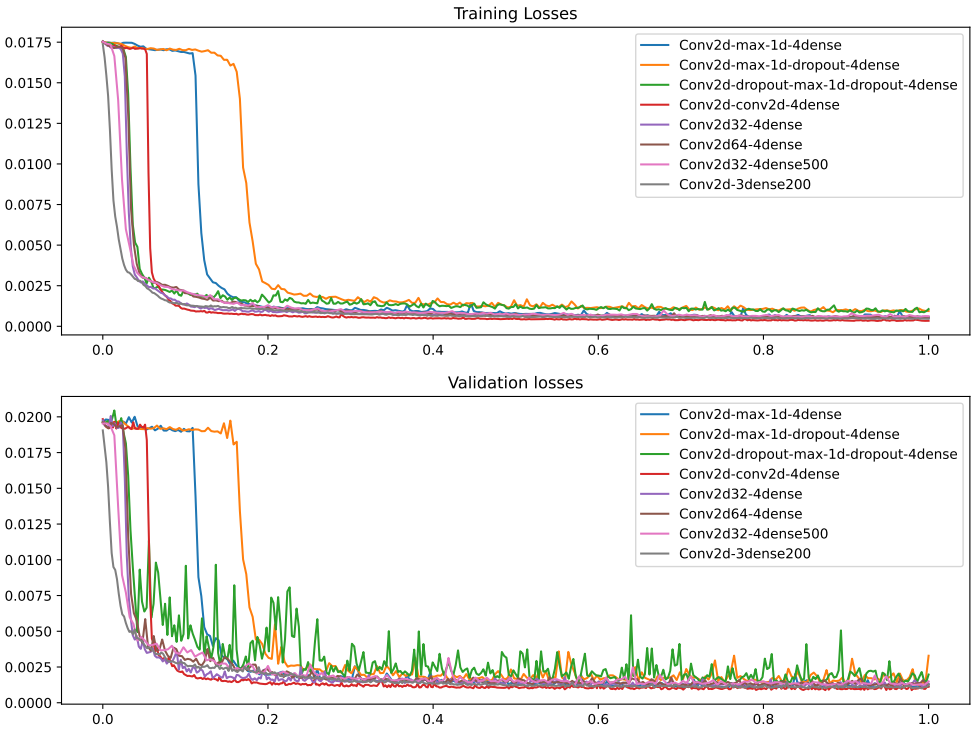


FIGURE 8.2 – Training and validation loss of 8 different CNN architectures.

TABLE 8.2 – Table of convergence rate and losses of selected architectures

Model Color	Epochs	Validation Loss
Blue	286	5e-4
Orange	260	9e-4
Green	359	1e-3
Red	489	**3e-4**
Purple	414	5e-4
Brown	293	5e-4
Pink	**216**	8e-4
Gray	467	4e-4

The training loss is a metric used to assess how better the CNN model extracted the features of the training data. In other words, it assesses the error of the model on the training set. Computationally, the training loss is calculated by taking the sum of errors for each example in the training set and measured after each number of training examples utilized in one iteration (batch). On the other hand, validation loss is a metric used to assess the performance of the CNN model on the validation set. The validation set is a portion of the dataset set aside to validate the performance of the model during training. The validation loss is calculated from the sum of the errors for each batch in the validation set, similar to the training loss.

All models use the mean squared error loss metric. Usage of other metrics did not have an effect on the performance of the model. Table 8.2 shows that most architectures have the same performance. When we examine the results in detail we observe that the pink architecture has a higher loss compared to the others. Similarly, we observe that the red architecture takes longer to converge to a solution and is prone to overfitting (The models that take a longer training phase to show the same performance as others are more prone to overfit). Therefore we chose to use the blue architecture that learns fast and has a low loss as our V_{th} extractor.

8.1.2 CHOOSING THE RIGHT LOSS METRIC

4 different loss metrics were considered for the model. These are

— Mean absolute percentage error (MAPE)

— Mean-squared logarithmic error (MSLE)

— Mean-squared error (MSE)

— Mean absolute error (MAE)

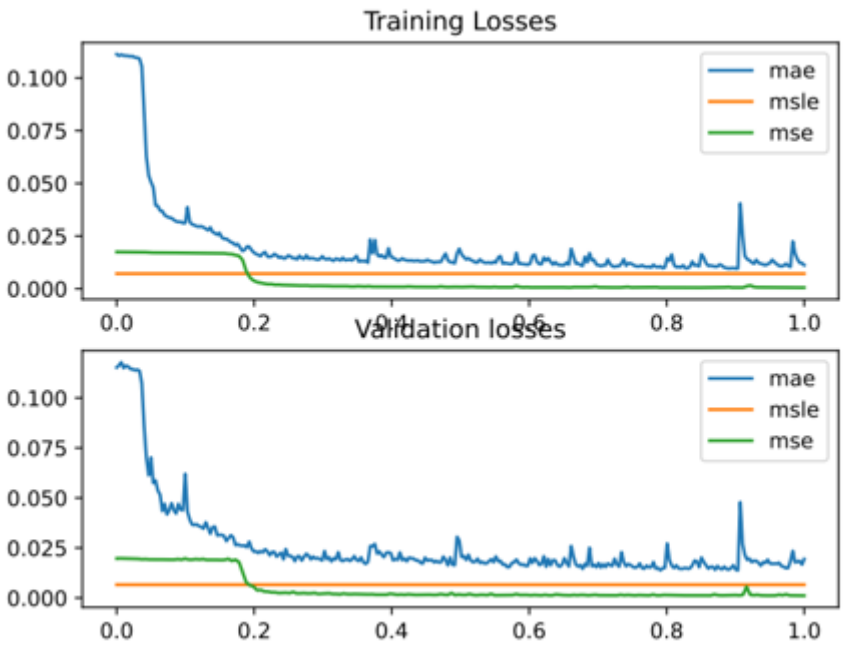


FIGURE 8.3 – Training and validation loss with respect to 3 different loss functions

the MAPE loss results produced higher values than the other metrics and were excluded from the graph. When the other metrics are considered the best performance was obtained using the MSE metric.

8.2 MODEL DETAILS OF FINALIZED EXTRACTOR MODEL

Our extractor model resembles our classifier model. However, we removed the dropout technique in our model to reduce the complexity and increase computational efficiency. The raw data composed of 4 terminals and 53 Vg step points is the first input to a convolution layer with 3x3 filters and max-pooling is applied. After this, our feature set becomes a 1D signal and 1x3 filters are applied in a second convolution layer. The output of these convolution layers are transformed into neurons and 3 fully connected layers, size of layers respectively 3200, 128, and 1 are applied to reduce the data and to give a single output which is treated as the V_{th} value. PRelu activation function is used between the layers and Mean-squared error is chosen as the loss function. The network is optimized using the ADAM optimizer.

In the initial layer, we have 32 3x3 kernels with different random weights that are optimized using backpropagation (Goodfellow et al., 2016). After convolving with these

kernels, our initial data of 4x53 dimensions becomes a feature set of size 2x51. To this feature set a max-pooling (Yamaguchi et al., 1990) layer of 2x2 is applied, this reduces the size of the data passing through the network by 1/4 and at the same time preserves the temporal and spatial nature of the data. This has an effect on the accuracy and the speed of the network. Later the same convolving and maxpooling operations reapplied to the data and we obtain 1x26 data. After this phase, we can apply 1D convolution and reduce the size of the data to 1x24. Finally, fully connected layers are applied to obtain the actual value output by the model. These steps can be observed in Fig. 8.4.

Our overall ML synoptic is presented in Fig. 6.1.

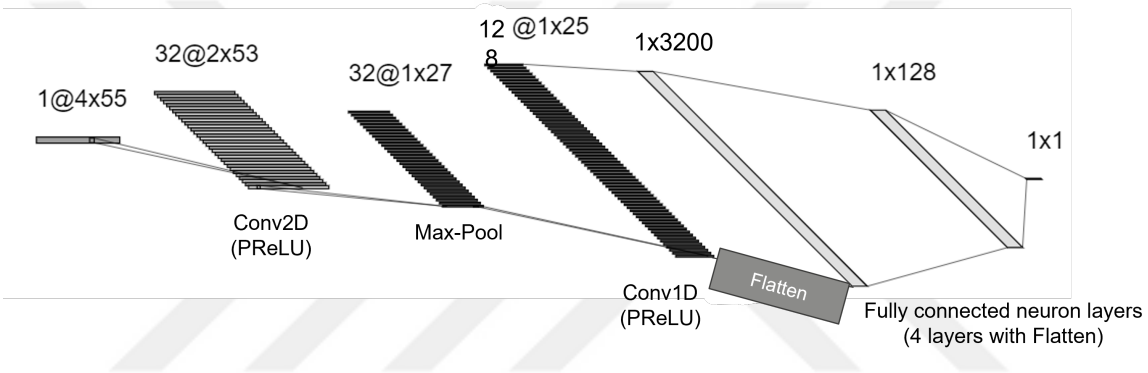


FIGURE 8.4 – ML Architecture illustration of CNN-based extractor model.

8.3 EXPERIMENT RESULTS

In this work, we have tested our 2-step CNN models together with IV curve samples that are described in the section 4.2, and the training methodology is described in the section 8.1. Table 8.2 shows the final validation loss which is a 5mV error at the total when training is finished.

When using the proposed network, we observe that after our classifier filters the data, 2253 samples are correctly labeled. Rest 549 samples that are labeled as not-healthy tags are stored for further expert review. The classified data was fed to the V_{th} extractor network.

Our extractor model was able to predict the V_{th} values of 2253 transistors in 2 seconds. The comparison of the predicted values and the values obtained via the gm-peak may

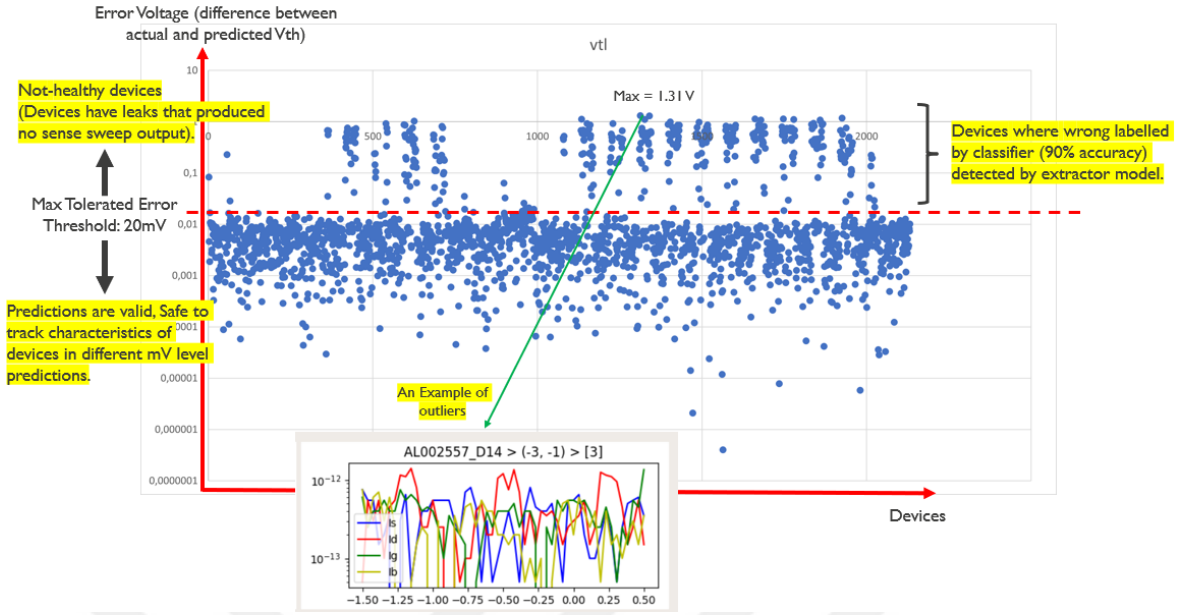


FIGURE 8.5 – The error of each prediction. Max tolerated error voltage of 20mV is specified and non-complied devices are classified as not-healthy

be observed in Fig. 8.5. After consultation with experts, a threshold value of 20mV was chosen. With the filtered outputs of the classification network, the V_{th} predictor network is capable of identifying outliers and further improving the accuracy of the system. Our system provides fast results and will facilitate device development in this field. Another use for the system is as a secondary classifier if the V_{th} values are calculated by the neural network and one of the traditional algorithms. When our V_{th} estimator neural network is used with all data including the outliers, the error rate is about 50mV. However if the data is pre-classified, and only the transistors passing the first neural network are fed into the system, the error rate drops to 8mV.

We recorded the time-cost of the peak-gm method and our approach for making a comparative analysis. Without any pre-filtering process, the peak-gm V_{th} extraction method lasted about 14 minutes for 2253 devices in our dataset. In the same dataset, the total time-cost of our ML approach for training 2 models lasted for about 50 minutes while using the models after training for classification and extraction processes combined lasted for only 1 minute. However, the time cost of the proposed V_{th} extractor model alone with no filtering operation is 5 seconds.

To achieve the optimum design in architecture, experts test a great number of transistors with different designs. Our approach may seem as slower than traditional methods when the time-cost of training and testing is combined but training the models are

one-time operation and ML models can repeatedly be used on the same architecture family over and over after successful training. Therefore it provides a significant time-cost advantage by reducing the V_{th} extraction process to seconds in the long run. In addition, traditional V_{th} extraction methods that cover the full range of drain voltage are rare, mainly due to the regional approaches underlying many conventional methods (Ortiz-Conde et al., 2013) as the Vds input increases, the gm point closes to Vg_{max} that results in erroneous V_{th} output and V_{th} values are ignored often at that level. Our method does not rely on any Vds input or constraints of the device. Therefore, we produce the V_{th} predictions by learning the relationship on Id-Vg and filling the blanks where the traditional method can fall short.

9 VISUALIZATION OF RESULTS

Researchers who are experts in the semiconductors domain may want to check the produced devices one last time before presenting them to the customer or academic world after all quality controls are completed, and passed through these controls. Presenting the results in a visual format that increases comprehensibility will be an appropriate choice as it will reduce the complexity of the research. Therefore, we have prepared a graphical user interface that can visually display the results from our classifier and extractor. The GUI contained 2 different display modes and the expert can switch between them.

9.1 WAFER VIEW

The visualizations are prepared by adhering to the positions of the devices receiving the wafer and the coordinates of the transistors they contain, taking appropriate colors according to the result of the classifier and helping the expert to distinguish them. The visual example is indicated in section 9.1. A color pallet complying with human-machine interaction dynamics is formed. And the location of the chip on a wafer is displayed.

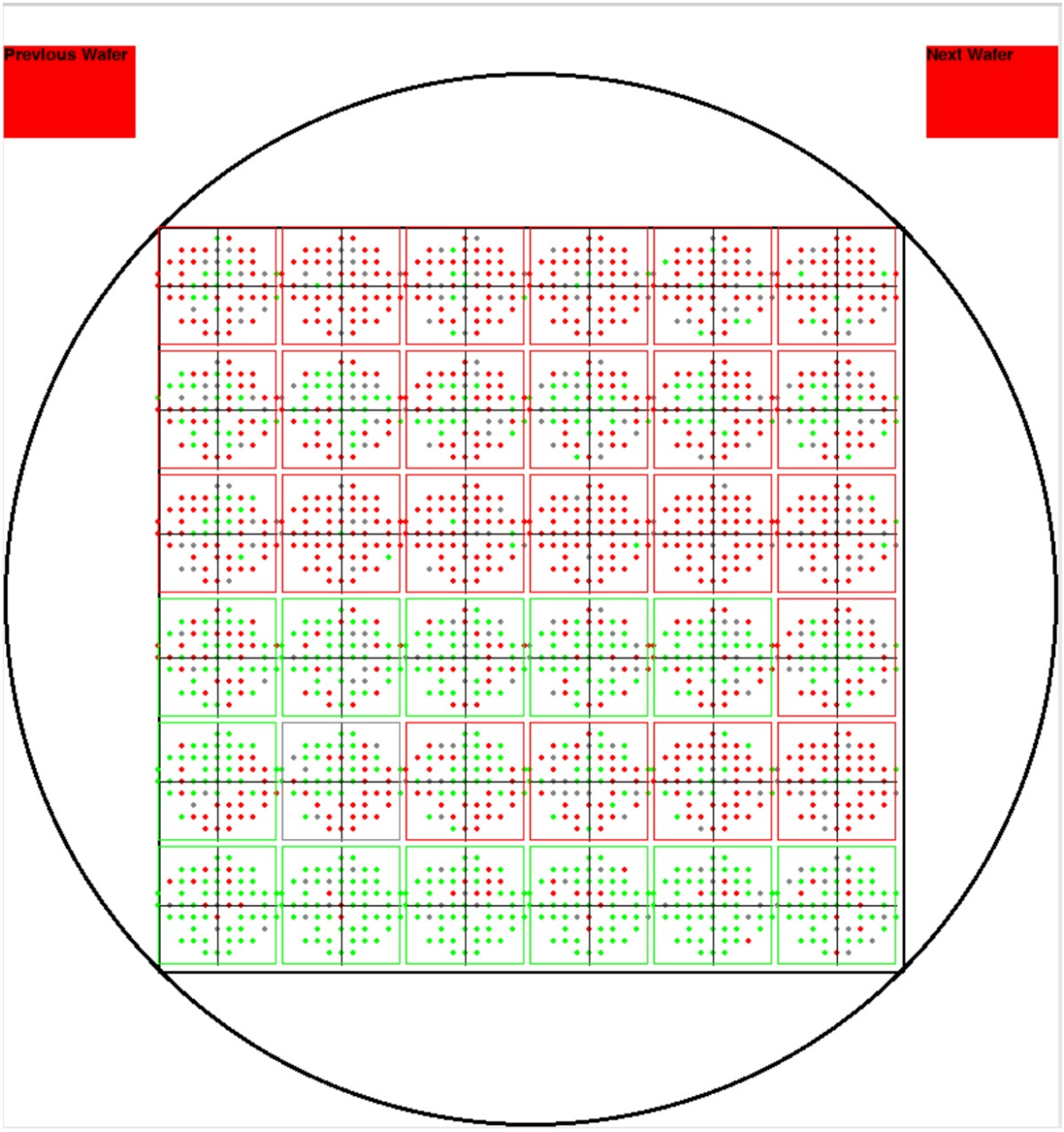


FIGURE 9.1 – An example of wafer view of modules. The green border represents the modules that have no leaks, while the red border represents the modules have leaks. The squares inside modules represent devices and their leaks statues with their color.

9.2 DEVICE VIEW

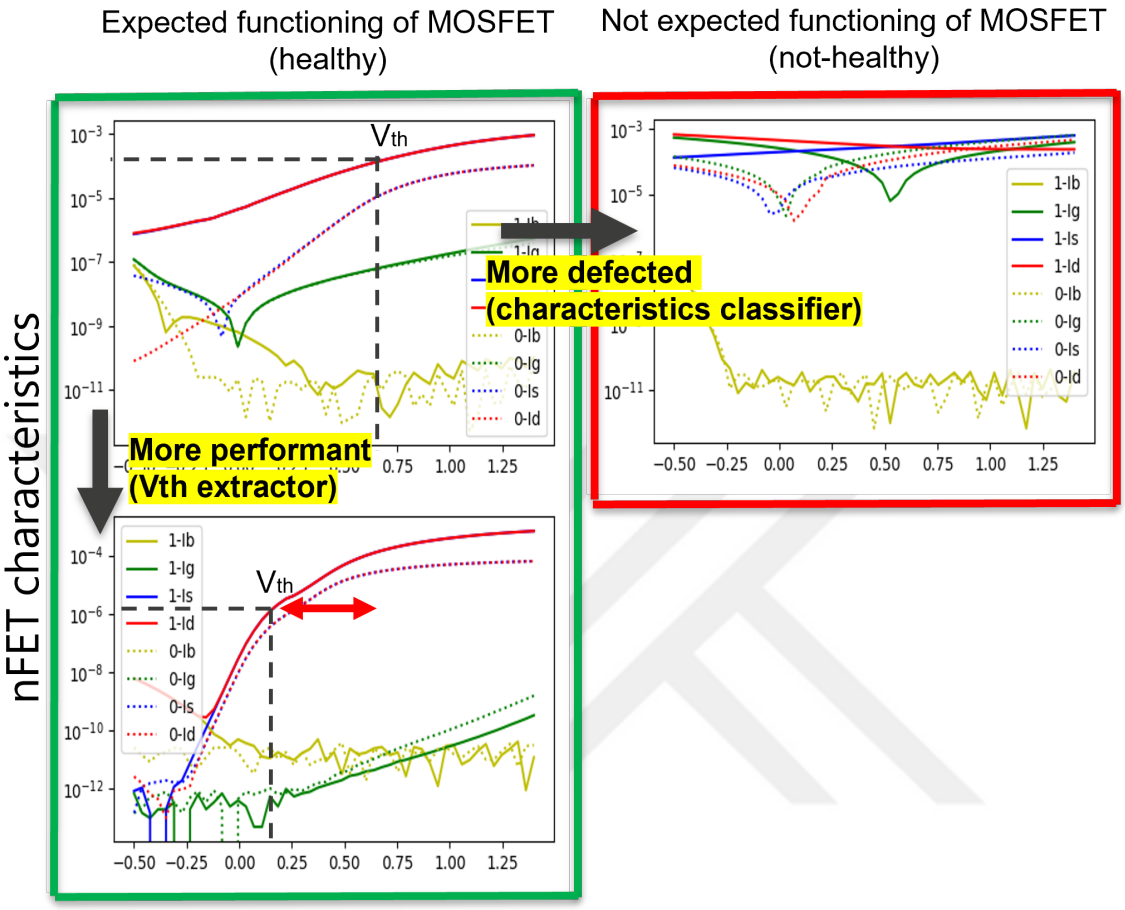


FIGURE 9.2 – Visualization of classifier outputs and extractor models on Id-Vg curve of each device.

An alternative view option is a list of the Id-Vg curves of the entire dataset or a subset of the devices selected by the evaluator. The 2 labels (healthy and non-healthy) are displayed on two separate columns. On the functional device column, the outputs of the V_{th} extractor are inferred and sorted according to their performance Fig. 9.2. This enables the easy viewing and interpretation of the combined results of our 2 step ML approach.

10 FUTURE WORKS

This work is done in the scope of a major project which aims to achieve fully autonomous quality control checks during the manufacturing process of transistors in chips using machine learning techniques. This thesis explores potential implementations of deep learning algorithms for quality checks and threshold voltage extraction.

The 2-step multi model ensemble of CNN's and CNN-based V_{th} extractor model developed within the scope of this thesis is accurate enough and can be evaluated fast enough to be useful in real manufacturing settings. The threshold voltage extractor that we developed provides valuable information regarding the chip.

The extractor model proposed in this work is already providing valuable information about the performance of chip architecture. It is the authors' expectation that if the model is further developed to be able to extract the subthreshold swing (SS) and gm peak parameters, it will become a more versatile and indispensable tool for research centers. Our model is trained with FinFET transistor data and therefore the parameters and settings are optimized for them. To further increase the capabilities of the model and to provide further generalization, other transistor technologies such as Two dimensional transistors (Novoselov et al., 2004) and IGZO planar transistors (Nomura et al., 2003) will be included in the dataset. As our models do not have hard coded transistor information, our ML approach will become a more universal model once this future work is completed.

Our work focused on error detection and performance evaluation. The interpretation of the results, defect reasoning, and defect profiling play an important role in the formation of the transistor development roadmap employed by research centers (Krause-Rehberg and Leipner, 1999). Our current work only detects defects and interpretation is left to the experts. Following the data collection and ML development phases of this project hope to relieve experts of this interpretation phase as well.

11 CONCLUSION

In this thesis, we present a new approach to the semiconductor industry. A multi-model ensemble of CNN models that can autonomously categorize the functionality of Id-Vg curves of transistors in minutes with high accuracy. A CNN model for V_{th} extraction of functional devices in seconds with less than 8mV error rate is also added as a secondary ML. Using a multi-model approach, even when one of the models in the ensemble deems a transistor to be not functional, it is classified as not-healthy and can be viewed by experts in later stages of the process. Without any changes to the classifier, it can successfully learn to categorize different types of transistors based on their I-V curves with no constraints. Later, the pre-filtered data set can be used with our second network where the performance of the transistor can be obtained and the application areas of the transistor can be determined. This network operates in the order of seconds and achieves an error of 8mV.

The lack of common perception on the impact of defects on the characteristics of humans manifests itself as increased false positives in the classification of the data and this may cause misdirected results during the decision-making phase. Along with our ML approach, the system was also designed to provide double-check control to the experts as an auxiliary tool that informs the expert when they decide against the results generated by our model. In essence, we propose a 2-step approach where 2 ML models run in series. However, it is also possible to run our models independently. Our models may be used for the classification of defects of a particular nature or for matching certain characteristic patterns. The V_{th} extractor network can be used without the filter network to obtain the V_{th} points from the Id-Vg routines very quickly.

An ML solution is useful not only for the decrease in development times but also can be used as a more centralized part of the decision making process to help clear up confusion among experts. At the same time, due to its fast response, it can be very useful in terms of verification if there is any counter-labeling situation between the expert and CNN. With these benefits at hand, ML is promising an increment in customer acceptance due to easier access and accelerated technology development. Also, the experiences of experts can be distilled and a model formed from them.

Our approach differs from the rigid formula-based approaches where some additional parameters have to be input into the system, and some parameters have to be assumed. We propose an alternative solution that does not necessitate any domain knowledge or physics constraints, is more adaptable to different architectures and is easily expandable.

In the light of the presented results, it is foreseen that ML techniques will further be developed for use in terms of both reductions in time cost and an increase in the detection accuracy of defects.

REFERENCES

Blaesi, T. (2018). Spice model generation by machine learning.
URL: *https ://silvaco.com/blog/spice-model-generation-by-machine-learning/*

Bousquet, O., Boucheron, S. and Lugosi, G. (2003). Introduction to statistical learning theory, *Summer school on machine learning*, Springer, pp. 169–207.

Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V.-K. and Strintzis, M. G. (2005). Knowledge-assisted semantic video object detection, *IEEE Transactions on Circuits and Systems for Video Technology* **15**(10) : 1210–1224.

Ding, D. et al. (2009). Machine learning based lithographic hotspot detection with critical-feature extraction and classification, *2009 IEEE International Conference on IC Design and Technology*, IEEE, pp. 219–222.

Fleury, D. et al. (2008). New y-function-based methodology for accurate extraction of electrical parameters on nano-scaled mosfets, *2008 IEEE International Conference on Microelectronic Test Structures*, IEEE, pp. 160–165.

Fu, W. et al. (2018). A hybrid forecasting framework with neural network and time-series method for intermittent demand in semiconductor supply chain, *IFIP International Conference on Advances in Production Management Systems*, Springer, pp. 65–72.

Glorot, X. et al. (2011). Deep sparse rectifier neural networks, *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, pp. 315–323.

Goodfellow, I. et al. (2016). *Deep learning*, MIT press.

Graves, A. et al. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures, *Neural networks* **18**(5-6) : 602–610.

Graves, A. et al. (2013). Speech recognition with deep recurrent neural networks, *2013 IEEE international conference on acoustics, speech and signal processing*, Ieee, pp. 6645–6649.

Göke, S. et al. (2021). Scaling ai in the sector that enables it : Lessons for semiconductor-device makers.
URL: *https ://www.mckinsey.com/industries/semiconductors/our-insights/scaling-ai-in-the-sector-that-enables-it-lessons-for-semiconductor-device-makers*

Hawkins, D. M. (2004). The problem of overfitting, *Journal of chemical information and computer sciences* **44**(1) : 1–12.

He, K. et al. (2015). Delving deep into rectifiers : Surpassing human-level performance on imagenet classification, *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034.

He, K. et al. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural computation* **9**(8) : 1735–1780.

Hochreiter, S. et al. (2001). A field guide to dynamical recurrent neural networks, *chapter Gradient Flow in Recurrent Nets : The Difficulty of Learning Long-Term Dependencies* pp. 237–243.

Horiguchi, N. et al. (2020). A view on the logic technology roadmap.
URL: *https ://online.publicationprinters.com/html5/reader/production/default.aspx ?pubname=3cca-46dd-930c-ed28ab9ac1ff*

Hsu, C.-Y. et al. (2021). Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing, *Journal of Intelligent Manufacturing* **32**(3) : 823–836.

Huang, H. et al. (2020). Unet 3+ : A full-scale connected unet for medical image segmentation, *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp. 1055–1059.

Irani, K. B. et al. (1990). Application of machine learning techniques to semiconductor manufacturing, *Applications of Artificial Intelligence VIII*, Vol. 1293, SPIE, pp. 956–965.

Kampouraki, A. et al. (2008). Heartbeat time series classification with support vector machines, *IEEE transactions on information technology in biomedicine* **13**(4) : 512–518.

Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. **URL:** *https ://karpathy.github.io/2015/05/21/rnn-effectiveness/*

Kingma, D. P. and Ba, J. (2014). Adam : A method for stochastic optimization, *arXiv preprint arXiv :1412.6980* .

Klemme, F. et al. (2020). Modeling emerging technologies using machine learning : challenges and opportunities, *Proceedings of the 39th International Conference on Computer-Aided Design*, pp. 1–9.

Krause-Rehberg, R. and Leipner, H. S. (1999). Positron annihilation in semiconductors : defect studies.

Lamamra, K. and Berrah, S. (2016). Modeling of mosfet transistor by mlp neural networks, *International Conference on Electrical Engineering and Control Applications*, Springer, pp. 407–415.

LeCun, Y. et al. (1995). Convolutional networks for images, speech, and time series, *The handbook of brain theory and neural networks* **3361**(10) : 1995.

LeNail, A. (2019). Nn-svg : Publication-ready neural network architecture schematics., *J. Open Source Softw.* **4**(33) : 747.

Li, M. et al. (2016). Physics-inspired neural networks for efficient device compact modeling, *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits* **2** : 44–49.

Li, S. et al. (2018). Independently recurrent neural network (indrnn) : Building a longer and deeper rnn, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5457–5466.

Lu, L. et al. (2019). Dying relu and initialization : Theory and numerical examples, *arXiv preprint arXiv :1903.06733* .

Luo, R. (2013). Optical proximity correction using a multilayer perceptron neural network, *Journal of Optics* **15**(7) : 075708.

Meijer, P. B. L. (1996). *Neural network applications in device and subcircuit modelling for circuit simulation*, Philips Electronics.

Nanopoulos, A. et al. (2001). Feature-based classification of time-series data, *International Journal of Computer Research* **10**(3) : 49–61.

Nawaz, J. M. et al. (2013). Time series fault prediction in semiconductor equipment using recurrent neural network, *International Symposium on Neural Networks*, Springer, pp. 463–472.

Nomura, K. et al. (2003). Thin-film transistor fabricated in single-crystalline transparent oxide semiconductor, *Science* **300**(5623) : 1269–1272.

Novoselov, K. S., Geim, A. K., Morozov, S. V., Jiang, D.-e., Zhang, Y., Dubonos, S. V., Grigorieva, I. V. and Firsov, A. A. (2004). Electric field effect in atomically thin carbon films, *science* **306**(5696) : 666–669.

Ortiz-Conde, A. et al. (2002). A review of recent mosfet threshold voltage extraction methods, *Microelectronics reliability* **42**(4-5) : 583–596.

Ortiz-Conde, A. et al. (2013). Revisiting mosfet threshold voltage extraction methods, *Microelectronics Reliability* **53**(1) : 90–104.

Press, W. H. et al. (2007). *Numerical recipes 3rd edition : The art of scientific computing*, Cambridge university press.

Radamson, H. H. et al. (2020). State of the art and future perspectives in advanced cmos technology, *Nanomaterials* **10**(8) : 1555.

Ramachandran, P. et al. (2017). Searching for activation functions, *arXiv preprint arXiv :1710.05941* .

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *nature* **323**(6088) : 533–536.

Sak, H., Senior, A. and Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, *arXiv preprint arXiv :1402.1128* .

Schroder, D. K. (2015). *Semiconductor material and device characterization*, John Wiley & Sons.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning, *Journal of big data* **6**(1) : 1–48.

Singer, P. (2020). Semiconductors and global domination.
URL: *https ://online.publicationprinters.com/html5/reader/production/default.aspx ?pubname=*
3cca-46dd-930c-ed28ab9ac1ff

Smagulova, K. and James, A. P. (2019). A survey on lstm memristive neural network architectures and applications, *The European Physical Journal Special Topics* **228**(10) : 2313–2324.

Srivastava, N. et al. (2014). Dropout : a simple way to prevent neural networks from overfitting, *The journal of machine learning research* **15**(1) : 1929–1958.

Teo, C.-W. et al. (2019). Tcad-enabled machine learning defect prediction to accelerate advanced semiconductor device failure analysis, *2019 International Conference on Simulation of Semiconductor Processes and Devices (SISPAD)*, IEEE, pp. 1–4.

Tuv, E. et al. (2018). Faster more accurate defect classification using machine vision. **URL:** *https ://www.intel.com/content/dam/www/public/us/en/documents/best-practices/faster-more-accurate-defect-classification-using-machine-vision-paper.pdf*

Vaswani, A. et al. (2017). Attention is all you need, *Advances in neural information processing systems* **30**.

Wang, C.-H. et al. (2006). Detection and classification of defect patterns on semiconductor wafers, *IIE transactions* **38**(12) : 1059–1068.

Yamaguchi, K. et al. (1990). A neural network for speaker-independent isolated word recognition., *ICSLP*.

Ying, X. (2019). An overview of overfitting and its solutions, *Journal of Physics : Conference Series*, Vol. 1168, IOP Publishing, p. 022022.

Yuan, T. et al. (2011). Detection of spatial defect patterns generated in semiconductor fabrication processes, *IEEE Transactions on Semiconductor Manufacturing* **24**(3) : 392–403.

Zell, A. (1994). *Simulation neuronaler netze*, Vol. 1, Addison-Wesley Bonn.

Zhang, L. et al. (2017). Artificial neural network design for compact modeling of generic transistors, *Journal of Computational Electronics* **16**(3) : 825–832.

BIOGRAPHICAL SKETCH

Hüsnü Murat Koçak earned his B.Sc. degree in Computer Engineering from Konya Technical University in 2020 and enrolled in the M.Sc. program in the Computer Engineering department of Galatasaray University in the same year. He attended respectively INRIA and KU Leuven as a visiting scholar. Currently, he is conducting research for his master's thesis, namely "Machine Learning Based Autonomous Quality Check and Characteristics Extraction for Chip Research" at IMEC / KU Leuven.

His research interests mainly include optimization and excellence through machine learning and its applications to physics, computer vision, signal processing, pattern recognition, and software development.

PUBLICATIONS

— Koçak, Hüsnü Murat, Jerome Mitard, and Ahmet Teoman Naskali. "Combined Machine Learning Techniques For Characteristics Classification and Threshold Voltage Extraction of Transistors." 2022 IEEE International Conference on Microelectronic Test Structures. IEEE, 2022.

— Koçak, Hüsnü Murat, Ahmet Teoman Naskali, and Jerome Mitard. "Detecting Transistor Defects in Medical Systems Using a Multi Model Ensemble of Convolutional Neural Networks." 2021 IEEE International Conference on Big Data (Big Data). IEEE, 2021.