

Sensor Qualification In An Industrial Environment Using A Convolutional Neural Network

Daniel Henrique da Costa Sousa
daniel.h.c.sousa@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

December 2020

Abstract

Nowadays, sensors occupy a significant role in our everyday lives. Magnetoresistive sensors have been proven to be very useful in many applications, from sensing magnetic fields to precision encoding. As a consequence, there is a constant demand to optimize such devices and maximize their production yield.

In any nano/microfabrication process, time is one of the most valuable resources. It is crucial to have tools to evaluate rapidly if the produced structures are within the expected outcomes or not.

This work presents a tool for quantitative analysis of some critical parameters of Magnetoresistive Devices fabrication. Furthermore, the software includes an automatic curve classifier using a Convolutional Neural Network to qualify Magnetoresistive Devices, focusing on sensors. Alongside the software for quantitative analysis and the curve classifier for sensors, it is also briefly discussed the use of a machine learning-based algorithm to predict the yield of a given nano/microfabrication fabrication process.

Keywords: Magnetoresistive Sensors, Microfabrication, Sensor Qualification, Machine Learning, Convolutional Neural Network

1. Introduction

Besides being a mature technology, magnetoresistive sensors are an attractive topic currently pursued by companies toward front end technology. Refs. [1, 2]

In applications that need sensing magnetic fields such as biomedical Refs. [3, 4], smart cities in Refs. [5, 6]. Magnetoresistive (MR) sensors have attracted much interest in particular because of their high sensitivity, low power consumption, low cost and small size, Refs. [7, 8].

Most of these electronics are built using silicon-based semiconductor processes, where the device is built using a top-down approach. There are mainly three types of processes that can be used to build the majority of devices: (i) Deposition of materials (films); (ii) Lithography; (iii) Etching.

Due to the typical sizes of the elements involved, tens of microns down to a few nanometers, strict micro/nanometric control is essential in all of the fabrication processes. Deviations in the process conditions can compromise the behavior of the final device, either electrically (e.g., Resistance Area product, $R \times A$) or magnetically (e.g., saturation magnetization, M_S).

Metrology is then critical for process validation and control for process evolving multilevel steps.

One common technique to test produced MR sensors is by measuring their resistance (R) across a varying external magnetic field (H), $R(H)$ is one of the main figures of merit of these devices. In an eight-inch wafer there are tens of thousands of sensors, corresponding to tens of thousands of $R(H)$ sets of points for each sensor. To fully qualify the produced wafer, one would have to manually classify every sensor, spending a considerable amount of time evaluating $R(H)$ loops. Additionally the quality of the sensors cannot be tackled only by evaluating easily defined numerical parameters. The shape of the loop is also relevant, hence using machine learning algorithms to help in the classification is extremely profitable. If it takes 5s to classify each of the tens of thousands of the $R(H)$ loops on a eight-inch wafer, then one would take at least one day to qualify it completely. A faster solution is therefore needed.

Another constant challenge that exists at micro and nanofabrication facilities, namely at *Instituto de Engenharia de Sistemas e Computadores - Microsistemas e Nanotecnologias* (INESC MN), is to improve the yield of manufacturing processes. By having the highest yield possible, one maximizes the use of time, resources, and hence money.

With this work, three different challenges were tackled: (i) Build an easy-to-use software that can digest ".SMP" files and provide easy visualization of many important parameters for MR devices as e.g. $R(H)$ plots, the minimal resistance, the magnetoresistance percentage (MR), among others; (ii) Build a machine learning-based tool to facilitate the analysis of large volumes of data, regarding $R(H)$ measurements of sensors, while providing an automatic classification of the $R(H)$ hysteresis curve shape; (iii) Propose a method to predict the yield of a micro or nanofabrication process. The method should allow the identification of the critical steps of the manufacturing process to improve the overall yield.

2. Magnetoresistance

The MR is a physical phenomenon where the electrical resistance of a material or a set of materials depends on the external magnetic field applied to it. The materials response can be engineered to have a behavior that can be used to solve different technological challenges.

Although this phenomenon, where some material's resistance is dependent on the magnetic field applied to it, was discovered in the middle of the 19th century, it was only at the end of the 20th century after Albert Fert, Ref. [9] and Peter Grünberg, Ref. [10], described both the Giant Magnetoresistance (GMR) effect independently, that this effect started to be used by the industry.

To characterize such devices, one important quantity is named after the effect. The MR is defined as:

$$MR = \frac{R_{High} - R_{Low}}{R_{Low}} \quad (1)$$

In Eq. 1, R_{High} is the high state resistance, and R_{Low} the low state resistance.

Both groups noticed that the resistance of two adjacent ferromagnetic (FM) layers separated by a non-magnetic (NM) layer, typically referred to as spacer, is dependent on the angle between magnetizations and the external magnetic field applied.

Physically the GMR described by A. Fert and P. Grünberg is a consequence of the asymmetry in the diffusive scattering of the conduction electrons at NM/FM interfaces.

Depending on the type of the spacer, different phenomena occur, Ref. [11]. The most relevant types of spacers/phenomena are: (i) If the spacer is a conductive NM layer, this effect gains the name Giant Magnetoresistance (GMR); (ii) In the case of a thin insulating spacer such as an oxide, where the electrons can pass from one FM layer to the other by quantum tunneling effect, the phenomenon is named Tunneling Magnetoresistance (TMR), such structures are called Magnetic Tunnel

Junctions (MTJs); (iii) There are other types of devices/phenomena, such as the Anisotropic Magnetoresistance (AMR), which is a single layer type of structure, but with lower values of MR and smaller linear range.

This work is focused on GMR and TMR sensors since they are the most recent technologies and have the highest MR values.

3. Artificial Neural Networks

Artificial neural networks (ANN) have been made famous because of their broad spectra of applications. Refs. [12, 13]

One type of network proven to be useful in several problems is the Convolutional Neural Network (CNN); as the name states, this type of ANN includes one or more convolution layers.

Convolution, in the context of a neural network, is a layer that receives its input from more than one output of the preceding layer, which are close to each other, meaning that they form a neighborhood in the previous layer. After having the neighborhood, a linear operation is done by multiplying the values by a set of weights. The product is done by multiplying the input data, which, in the present case, will be a two-dimensional array, with a two-dimensional array of weights called filter or kernel. Since one uses the dot product, the outcome of such an operation is just a value. The kernel is then swept across the input; depending on the kernel's dimensions and the input, the convolution's output can be an array of values. Refs. [14, 15]

The convolution layer is of significant importance when one is trying to recognize a specific feature across one dataset. The convolution is invariant to translation; this becomes very handy if one is trying to detect specific features across multiple locations in a dataset. Refs. [14, 15]

4. Transfer Curve

A transfer curve describes how the magnitude of a physical quantity is translated into an electrical signal by a given sensor. In a MR device this curve translates the sensed magnetic field into resistance values, $R(H)$.

In Fig. 1, the transfer curve of an ideal sensor is presented. The perfect sensor should have a linear behavior between the two resistance plateaus, R_{Low} and R_{High} . For each point, ideally, there should be a unique correspondence between the magnetic field and resistance. In-between the two states, one should have a linear region that can be used to sense magnetic fields. Non-ideally, the curve could present a hysteric behavior, characterized by coercivity - H_c , defined as the curve's broadness. The magnetic field, after which the plateaus are attained, is called saturation field, H_{sat} .

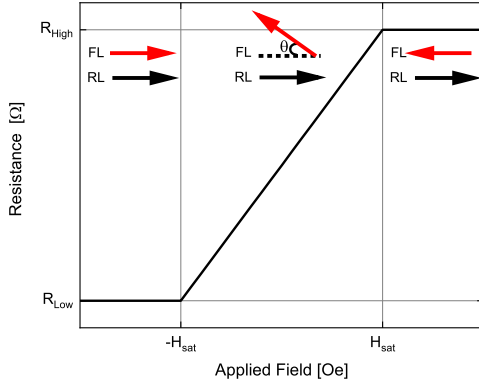


Figure 1: Transfer curve of an ideal sensor. The arrows represent the angle between the magnetization of the Free Layer (FL) and Reference Layer (RL).

The curves can be mirrored with respect to the y-axis; the only difference is the direction of the applied magnetic field relative to the reference of the sample. When fabricating a MR sensor, one wishes to have all the sensors with a $R(H)$ loop similar to Fig. 1.

5. Fabricating a magnetoresistive sensor

There are mainly three families of fabrication steps that can be used to manufacture MR devices: (i) Deposition of materials; (ii) Lithography; (iii) Etching.

In this section, fabrication particularities are addressed alongside possible sources of problems towards the final device performance.

5.1. Thin Films Deposition

This process aims to deposit the required stack of materials to build the desired devices; such devices require different materials and deposition steps.

At INESC MN, three techniques are available: Ion Beam Deposition, Direct Current/Radio Frequency Sputtering, and Magnetron Sputtering. All of them have their target application, and all of them have their limitations.

All the latter have a radial symmetry in the thickness deposited. Such symmetry is a result of how the beam of depositing species is created. The thickness (t) as a function of the radius (r) to the center of the wafer can be modulated by the Eq. 2.

$$t(r) = C_0 + C_1 e^{-\frac{(r-C_2)^2}{2C_3^2}}, \quad C_0, C_1, C_2, C_3 \in \mathbb{R} \quad (2)$$

Fig. 2 shows the deposited thickness as a function of radius for the deposition of MgO using the Nordiko 3600, an IBD-based machine available @ INESC MN. A Gaussian-like behavior is clearly present.

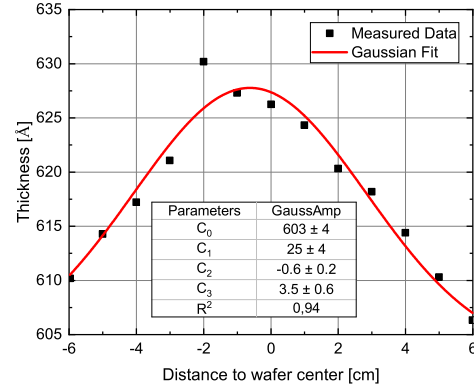


Figure 2: Deposition and Gaussian fit of measured thickness of a deposition of MgO in a 6'' wafer at INESC MN using Nordiko 3600.

5.2. Lithography

Lithography is the step where the wanted pattern is transferred onto a sample. The lithography needs three elements, a radiation source, a photoresist (PR) coated sample, and a system to control how the sample is exposed to the radiation.

At INESC MN, the techniques available are Direct Write Laser (DWL) Lithography, Electron Beam Lithography (EBL), and UV lithography. The latter uses hard masks. Both the EBL and DWL use virtual masks, allowing to design features with minimal size up to 50 nm and 1 μm respectively.

The errors associated with this process are substantially different from what happens in the deposition and etching. Consider the DWL system; the laser is scanned across the sample using a high precision moving stage, which moves according to the virtual mask. The limitation in terms of feature size is due to the diffraction limits and the optical/mechanical system used.

Some of the challenges of this step might result from a non-uniform profile of the laser. That would cause the PR not to be fully affected by the radiation and hence not be correctly developed in the developing step. Another limiting factor of this step is the profile of the PR. While coating, different profiles of the PR might arise due to the amount of the PR dispensed, erroneous spinner rotational speed, non-optimized baking, etc.

5.3. Etching

The etching is a process through which material is removed from the sample.

At INESC MN, there are mainly two different types of processes available, Ion Beam Milling (IBM) and Reactive Ion Etching (RIE).

In an IBM system, a beam of ions, typically Ar^+ , is used to bombard a target material, thereby transferring their momentum to the atoms on the sam-

ple, causing them to be removed from it.

Some of the limitations of this technique are due to the Gaussian nature of the beam of etchant species, the redeposition of material during the etching, and the lateral profile of etched volumes and trenching.

The RIE system is similar to IBM, but instead of just transferring momentum from the ions to the sample to release atoms/molecules, one also uses chemical reactions to facilitate their release. There are mainly two challenges in RIE that can compromise this step: microloading and the Aspect Ratio Dependent Etch Rate (ARDE).

If the sample is unpatterned, then one expects the uniformity across the etched surface to have radial symmetry, following a Gaussian profile.

Eq. 2 might be used to express the profile of an unpatterned sample after etching in both RIE and IBM if all difficulties except for the Gaussian nature are discarded. In this case, the higher etching occurs at the center of the beam of ions, which one aims at the center of the wafer.

5.4. Summary of Possible Challenges and Metrology tools

There are many features to control in a micro/nanofabrication process, Tab. 1. This, alongside the several steps it takes to produce a given MR sensor, makes predicting the yield of a micro/nanofabrication a challenging task.

6. Treating and Classifying Autoprober Data

A novel and faster way of interpreting the data retrieved from measurements done in the Autoprober @ INESC MN, is discussed next. It is also presented a transfer curve classifier for sensors that can distinguish between four different classes of sensor quality.

With this in mind, the work is divided into three stages: (i) Treating the data from the Autoprober to obtain relevant quantities towards the analysis of MR devices; (ii) Build an automatic classifier for the fast analysis of $R(H)$ loops of MR sensors; (iii) Develop an easy-to-analyze tool to allow the visualization of the relevant quantities.

I used Anaconda to tackle all the challenges above (Python 3.8), Ref. [16]. To build the automatic $R(H)$ classifier I used Keras/Tensorflow Refs. [17, 18], and Plotly, Ref. [19], for the easy-to-analyze tool.

6.1. Data Processor

In Fig. 3, one may see the setup schematic used at INESC MN for automatically taking measurements of the $R(H)$ loop. This system requires the user to input a grid of locations where the probes should be placed (map), and hence it allows to take automatic measurements across a

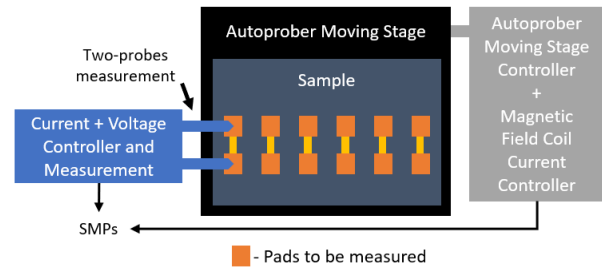


Figure 3: Autoprober setup schematic at INESC MN.

complete wafer; the system returns a file of extension ".SMP" which is similar to a *comma-separated values* (".csv") file, but uses semicolons as separator of data.

The file contains many columns, and each row corresponds to a given sensor, a specific magnetic field, and a measured resistance.

In order to use the code, the users should provide the necessary inputs like the SMPs, the column which are relevant, etc. Fig. 4 shows the interface and all entry fields.

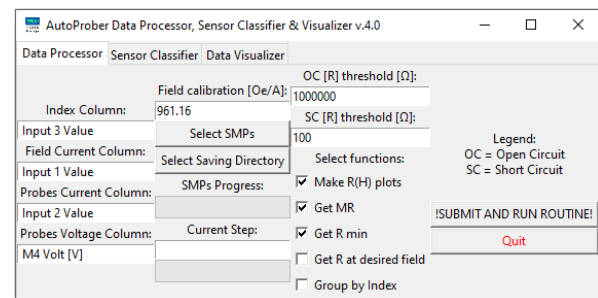


Figure 4: Autoprober Data Processor version 4.0 interface with the default values.

The current solution includes five different functions: (i) Make $R(H)$ plots; (ii) Get MR ; (iii) Get the R min; (iv) Get R at desired field; (v) Group by Index.

Make $R(H)$ plots - Function that draws and saves $R(H)$ plots under ".png" format alongside with ".csv" file for each curve, containing current used to generate magnetic field, magnetic field, resistance measured and bias current.

Get MR - Function that returns a ".xlsx" file with the computed MR for each curve, including OC/SC filtering.

Get R min - Function that returns a ".xlsx" file with the value of resistance of the low resistance plateau for each curve, including OC/SC filtering.

Get R at desired field - Function that requires the user to input a target magnetic field and returns two different ".xlsx" files; one file contains the two values closest to target field and the correspondent values of resistance, for each curve; the other file contains the average of both resistance and magnetic field for each of the curves. The file with the averages is OC/SC filtered.

Process	Possible Challenges	Metrology tools @ INESC MN
Deposition	Incorrect deposition parameters	Profilometer Scanning electron microscopy
	Incorrect profile of thickness or roughness	Ellipsometer Optical microscope
	Unsuited composition of the deposited layer	X-ray diffractometer Resistivity measurement
Lithography	Miss alignment between lithography levels	Profilometer
	Incorrect exposure to radiation	Optical microscope
Etching	Incorrect thickness etched	Profilometer Scanning electron microscopy
	Incorrect profile of the etched layer	Ellipsometer Optical microscope
		Resistivity measurement

Table 1: Summary of most common challenges that can happen in each of the processes.

Group by Index - Specific applications may require more than one measurement per die/chip. When using the Autoprober one way to optimize time consumption is to measure the same set of pads across all dies/chips, which have the same interdistance, in all equal dies/chips. Considering N different dies/chips that need n measurements taken per dies/chip, this function allows to reorganize the N transfer curve plots in n different folders.

6.2. Sensor Curve Classifier

This work second goal was to build a tool to facilitate the $R(H)$ characterization of large volumes of data for sensing applications. I trained a CNN to do an automatic classification of the curve shape which is faster and cheaper then performing it manually. The goal of this classifier is to sort $R(H)$ curves of both GMR and TMR based devices into four different categories.

It is crucial to understand that this classification only takes into account the shape of the curves and not the values of resistance/magnetic field. Therefore this should only be interpreted as complementary information.

6.2.1. Data Preparation

Data was treated before trying to train such network. To be independent of the stack deposited, there is the need to normalize both the resistance and the applied field in all curves between the same values. The following mathematical operation to a list of n values $\{x\} = \{x_0, x_1, \dots, x_n\}$ is defined:

$$\{X\} = 2 \cdot \frac{(\{x\} - \max(\{x\}))}{\max(\{x\}) - \min(\{x\})} + 1$$

$\{X\}$ is the normalized list of $\{x\}$ between between -1 and 1. The function is applied to both the x and y-axis, and thus all curves will be majored between the same values, increasing the amount of data for each class without losing the curve tendency. Without this operation, a much vast database is required to train the network, since examples for

each stack/each value of $R \times A$, etc., would be needed.

Other critical functions defined were upsampling and downsampling functions. This is useful when acquiring data in the Autoprober. Different devices and different users may choose different amounts of sampling points; the number of features of an input of a CNN cannot change after its defined; therefore, it is crucial to have upsampling and downsampling functions to downsample or upsample data to meet the requirements of the input layer.

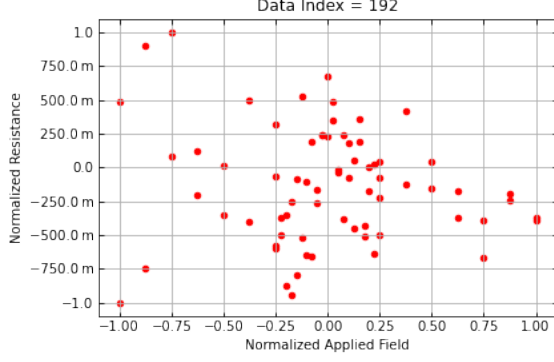
Only the downsampling function was used in the studied dataset, being 70 the number of points required by the input layer of the defined network.

6.2.2. Classes and Datasets

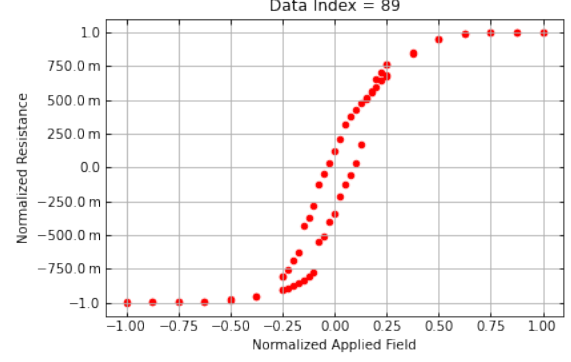
After treating the data I sorted all $R(H)$ loops into four distinct categories: (i) NOK - Curves with no visible sign of a MR behavior of the material; (ii) A - Curves with some MR behavior of the material, but not as expected. As the classifier was implemented for sensor applications, the essential criteria for this category were: (a) large and irregular coercivity along the curve ($\gtrsim 10 Oe$) - non-normalized plot; (b) large Barkhausen jumps in the transfer curve ($\gtrsim 0.25 AU$) - normalized plot. (iii) M - Curves where there is a MR behavior of the material, but not of an ideal sensor. Criteria to include curves in this category were: (a) small coercivity along the curve ($\lesssim 10 Oe$); (b) no large Barkhausen jumps in the hysteresis curve ($\lesssim 0.25 AU$). (iv) OK - Curves where there is a MR behavior typical of an ideal sensor. Criteria to include curves in this category were: (a) no visible coercivity ($\lesssim 1 Oe$); (b) no Barkhausen jumps ($\lesssim 0.04 AU$).

Class	Number of Occurrences
NOK	618 (55.48%)
A	325 (29.17%)
M	73 (6.55%)
OK	98 (8.80%)
Sum	1 114 (100%)

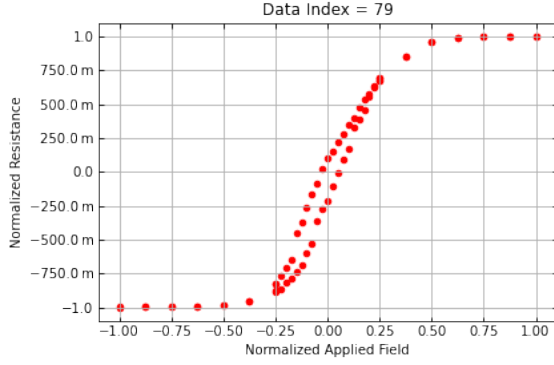
Table 2: Table of occurrences of each class in the dataset.



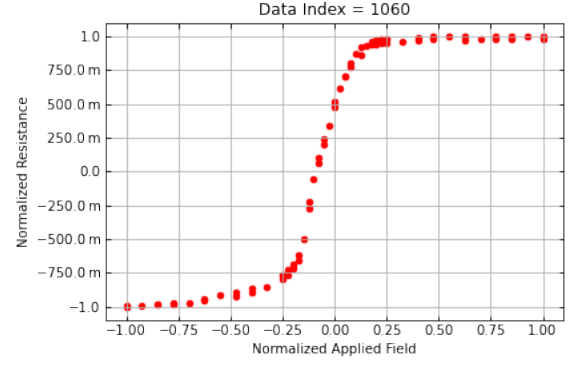
(a) "NOK" class example.



(b) "A" class example.



(c) "M" class example.



(d) "OK" class example.

Figure 5: Normalized examples for all classes.

In order for all evaluated models to be tested under the same conditions, the datasets for training, validation, and test were kept constant. Chosen dataset ratio distribution was 60% of all available data for the training set, 20% for validation, and the remaining 20% for testing. For each dataset, the data was picked randomly, with each class having identical distributions on all datasets. This is crucial because the full dataset used is imbalanced, as shown on Tab. 2.

One-hot encoding of the categories was performed since there is no ordinal relationship exist between the different classes.

6.2.3. Model and Training

Seven different CNNs and one ANN were tested, using two different metrics to evaluate the models: the regular accuracy and the F_1 score.

The Adam Optimization Algorithm was chosen to train the model, a commonly used optimizer for such applications. All algorithm parameters had their default values. Ref. [20]

The Categorical Cross-Entropy (CCE) was employed as the loss function, defined as:

$$CCE = - \sum_{i=1}^{output} y_i \cdot \log(\hat{y}_i) \quad (3)$$

where y_i is the real label of the i -th index of the target array, and \hat{y}_i is the model prediction (probability).

To prevent overfitting, early stopping was used. Training the model too many epochs will result in an overfitted while training few epochs will result in an underfitted model, having an underfitted or overfitted model will fail to predict the desired patterns. Early stopping checks the performance of the model in the validation dataset, once the performance of the model on the validation dataset stops improving, the training is stopped, and the best model is saved.

6.2.4. Evaluation and Final Model

Accuracy should be used when all the classes are equally important and when one has a balanced dataset. However, the accuracy can be misleading when one has an imbalanced dataset. In imbalanced datasets, and when one cares about the incorrectly classified examples, the F_1 score is typically used as the evaluation metric. Ref. [15]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Where TP are the true positives, TN are the true negative, FP are the false positives, and FN the false negatives.

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (7)$$

When dealing with multiclass, some modifications to the F_1 formalism is required. The F_1 was defined as the average of the computed F_1 score for each class without any weights to account for the imbalanced dataset. This might lead to a larger penalization if the classifier does not perform well with the minority classes, which could be profitable while evaluating an imbalanced dataset.

The best model in the test dataset had an accuracy of 0.9865, and an F_1 score of 0.9623, and the optimal model was trained in epoch number 229. It was composed of two convolutions, two max pool, one flatten, and two dense layers. Check Refs. [14, 15] for insights on the layers.

After selecting the model's architecture which lead to the best performance, the same architecture was trained again but with a vaster test dataset. Even though the optimal number of epochs may change from run to run due to the process's stochastic nature, it was assumed to be the same for a not so different training dataset.

Now the training dataset can have 70% of all data, and the rest 30% can be saved for testing. This will lead to more reliable testing; however, the training might not be stopped in the optimal epoch since no validation dataset for the early stooping exists.

Fig. 6 shows that the model over classifies curves as a class "M". This might result from the fact that there is a minimal difference between this class and some curves of the "A" and "OK" class.

In this final model, the accuracy over the test dataset was 0.9672, and the F_1 score was 0.9257, giving thus good results. The model is expected to give similar results in upcoming wafers, as long as some requirements are fulfilled: (i) Low resistance plateau must be on the curve's left-hand side and high resistance plateau should be on the right-hand one. This can be achieved by merely mirroring the curves; (ii) If curves have a number of sampling points different from 70, either upsampling or downsampling needs to and will occur.

6.2.5. Implementation

An easy-to-use interface for the automatic classification of sensors was implemented. Please note that no quantitative parameter, such as MR or R_{min} , is taken into account to make such classification.

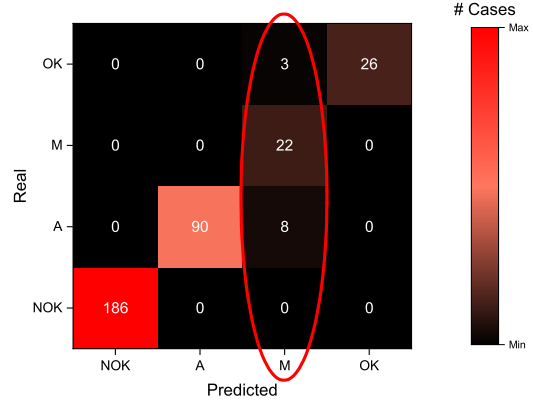


Figure 6: Confusion matrix heatmap for the final model evaluation using a test dataset with 30% of all data.

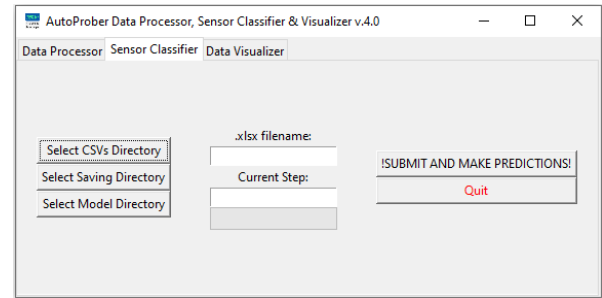


Figure 7: AutoProber Sensor Classifier version 4.0 interface with the default values.

In order for the script to successfully run, the user has to input the following data: (i) Folder with the $R(H)$ loops data (".csv" from Sec. 6.1); (ii) Directory where the predictions should be saved; (iii) Name for the ".xlsx" file of predictions; (iv) Folder with the model inside, and it should be saved under the TensorFlow SavedModel format. The model trained above is supplied alongside the code;

The code returns a ".xlsx" file with classifications for all the curves and probabilities for each class.

6.3. Data Visualizer

The Data Visualizer uses files computed by the Data Processor/Sensor Classifier, and the map used in the AutoProber; with this information, it does an interactive plot which is saved in ".HTML" format.

In order for the script to successfully run, the user has to input the following data: (i) Titles for the points, they must be separated by commas; (ii) Static translation that the AutoProber map coordinates should suffer with respect to the final plot's coordinate system; (iii) Title to be shown in the plot; (iv) Filename of the final .HTML file; (v) Directory with the measurements that one wishes to display; (vi) Directory which has the maps inside, "X" and "Y" column should be identified in the first row; (vii) Directory to save the ".HTML" file; (viii) Select

from the list the variable to visualize; (ix) Check the "Drawn Dies Frames" if one wishes to draw die/chip limits. Some windows will pop-up asking for relevant information like dimensions and how many are in the x and y-direction.

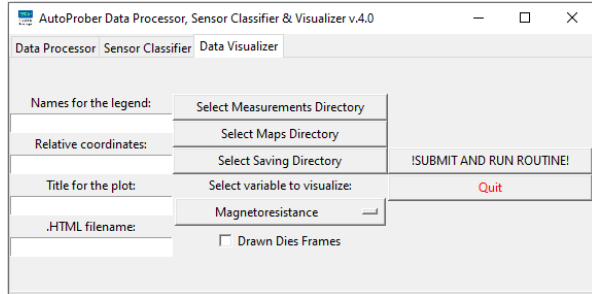


Figure 8: Autoprober Data Visualizer version 4.0 with the default values.

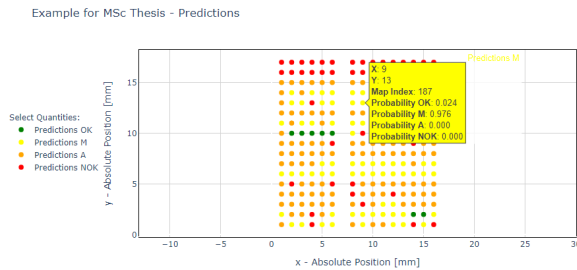


Figure 9: Autoprober Data Visualizer version 4.0 example using predictions from the Sensor Classifier.

In all examples, by hovering the mouse over the points, the user may see relevant information about the plot. For instance, in the prediction example, each class's probability is present, Fig. 9.

7. Proposal For Predicting Sensor Yield

There are many challenges to the manufacturing of a MR sensor, Sec. 5. The approach now presented is Machine Learning based.

One of the most challenging parts of Machine Learning is understanding what data is relevant or not. Since this is a novel problem, one should first define what data should be collected.

When asking someone for that, one should try to facilitate the person that will gather the data job. For instance, if one develops a Process Characterization Runsheet (PCR) to be filled alongside the regular runsheet, it is evident that such a document should be fast and easy-to-fill. If the document is dull, it may fail to engage users in filling it. With this in mind, a PCR was developed and is ready to be implemented. For the initial approach, it is proposed to start only with the analysis of lithography and etching steps.

The etching steps may sometimes be difficult to evaluate; this arises from the fact that some layers are so thin that it is difficult to know if one has

etched the sample as supposed or not. Sometimes, one can only rely on optical inspection and check the contrast of colors for different materials. When the layers are thick, one can rely on the profilometry to evaluate the etch depth.

After gathering the data, it should be organized to train the model successfully. Each process step's evaluation should be a feature, and the target should be the device's classification. One should use all the available outputs of Autoprober Data Processor, Sensor Classifier & Visualizer before deciding on the sensor classification. The Sensor Classifier could be used to check the shape of the curve and another complementary information to check if the sensor is within expected values, for instance, the MR or R_{min} .

Regarding the model selection, the same approach as in Sec. 6.2.3 should be used. The suggestion is to search for a similar problems and then try different solutions to check which one yields the highest performance over a validation dataset before training the final model.

If the model is appropriately trained, it will be advantageous to evaluate, for instance, exclusion zones of productions, which allows saving time. With this approach, the users can further understand which steps are the most critical of a micro or nanofabrication process and then try to improve them.

8. Conclusions and Future Work

In this work, easy-to-use tools that can provide a fast quantitative analysis of fabricated samples/wafers are presented. These tools include an automatic curve classifier that can successfully distinguish between what is an appropriate MR sensor response and what is not. The conjugation of the output of the quantitative analysis and the classifier is enough to make an automatic qualification of the sensors which are within working parameters. It is expected that the curve classifier will classify $>90\%$ of all curves correctly and hence as prove its value.

Due to the modular nature in which the Data Processor was built, more analysis functions could be added to calculate other relevant parameters, such as, for instance, the sensitivity. Future users should tailor the code to their needs if needed.

The last challenge of this work was to propose a technique to predict the yield of a micro or nanofabrication process while identifying the process's critical steps; for that goal, a Process Characterization Runsheet was created. The PCR allows gathering the necessary data to accomplish this task. The Runsheet created is for an MTJ based process, but it can be adapted for SV. If such approximation is implemented and the users could use it to predict the yield.

For the future, the code could be placed on a central service available through the internet, this way, users could process their files without running the software on their machines. This is already being pursued through the use of Google Colab, which supports Jupyter Notebooks. Please refer to "Cloud-based" folder, Ref. [21].

Regarding the third objective of this work, it is necessary to engage INESC MN users to fill in the PCR to gather the required data for successfully implementing such model.

In the Ref. [21] a stable version of the code is available alongside the Process Characterization Runsheet and instructions towards the installation of required libraries.

References

- [1] Magnetic Measurement - Bogen. Bogen Electronic GmbH, 2020. URL <http://www.bogen-electronic.com/en/magnetic-measurement-solutions/technology/magnetic-measurement.html>. Accessed on 27 August 2020.
- [2] Bosch Sensortec: Bosch MEMS Technology. Bosch Sensortec GmbH, 2020. URL <https://www.bosch-sensortec.com/>. Accessed on 27 August 2020.
- [3] T. Barroso et al. Detection of BCG bacteria using a magnetoresistive biosensor: A step towards a fully electronic platform for tuberculosis point-of-care detection. *Biosensors and Bioelectronics*, 100, 09 2017. doi: 10.1016/j.bios.2017.09.004.
- [4] X.-H. Mu et al. A new rapid detection method for ricin based on tunneling magnetoresistance biosensor. *Sensors and Actuators B: Chemical*, 284:638 – 649, 2019. ISSN 0925-4005. doi: <https://doi.org/10.1016/j.snb.2018.12.127>.
- [5] B. Yang and Y. Lei. Vehicle Detection and Classification for Low-Speed Congested Traffic With Anisotropic Magnetoresistive Sensor. *IEEE Sensors Journal*, 15(2):1132–1138, Feb 2015. ISSN 2379-9153. doi: 10.1109/JSEN.2014.2359014.
- [6] P. Pai et al. Non-intrusive electric power sensors for smart grid. In *SENSORS, 2012 IEEE*, pages 1–4, 2012.
- [7] C. Zheng et al. Magnetoresistive Sensor Development Roadmap (Non-Recording Applications). *IEEE Transactions on Magnetics*, 55(4):1–30, April 2019. ISSN 1941-0069. doi: 10.1109/TMAG.2019.2896036.
- [8] P. P. Freitas et al. Magnetoresistive sensors. *Journal of Physics: Condensed Matter*, 19(16):165221, April 2007. doi: 10.1088/0953-8984/19/16/165221.
- [9] A. Fert et al. Giant Magnetoresistance of (001)Fe/(001)Cr Magnetic Superlattices. *Phys. Rev. Lett.*, 61:2472–2475, Nov 1988. doi: 10.1103/PhysRevLett.61.2472.
- [10] P. Grünberg et al. Enhanced magnetoresistance in layered magnetic structures with antiferromagnetic interlayer exchange. *Phys. Rev. B*, 39:4828–4830, Mar 1989. doi: 10.1103/PhysRevB.39.4828.
- [11] K. M. Krishnan. *Fundamentals and Applications of Magnetic Materials*. Oxford University Press, Oxford, 2016. ISBN 9780199570447. doi: 10.1093/acprof:oso/9780199570447.001.0001.
- [12] D. Ciregan et al. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012.
- [13] C. Barata and J. S. Marques. Deep Learning For Skin Cancer Diagnosis With Hierarchical Architectures. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 841–845, 2019.
- [14] I. Goodfellow et al. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [15] S. Raschka and V. Mirjalili. *Python Machine Learning, 3rd Ed*. Packt Publishing, Birmingham, UK, 2019. ISBN 978-1789955750.
- [16] Anaconda software distribution. *Anaconda Documentation*, 2020. URL <https://docs.anaconda.com/>. Accessed on 18 September 2020.
- [17] F. Chollet et al. Keras. *GitHub repository*, 2015. Accessed on 20 September 2020.
- [18] M. Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- [19] Plotly Technologies Inc. *Collaborative data science*, 2015. URL <https://plot.ly>.
- [20] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. 2017. URL <http://arxiv.org/abs/1412.6980>.
- [21] D. Sousa. Autoprober Data Processor, Sensor Classifier & Visualizer - Google Drive. 2020. URL <https://drive.google.com/drive/folders/1RgHbQ2UwFniroyRiAt67cKYSTcLIhd-u?usp=sharing>. Accessed on 30 October 2020.