




Parameter extraction in thin film transistors using artificial neural networks

Roberto C. Valdés^{1,*} , Farid García¹, Rodolfo Z. García¹, Asdrúbal López¹, and Norberto Hernández²

¹Autonomous University of the State of Mexico, Toluca, State of Mexico, Mexico

²Nanoscience, Micro and Nanotechnologies Centre of the IPN, Mexico City, Mexico

Received: 9 September 2022

Accepted: 20 January 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

ABSTRACT

This work presents a method based on supervised learning for the extraction of parameters in Indium Gallium Zinc Oxide Thin-Film Transistors with aluminium contacts, as an alternative regarding analytical and optimisation methods. The method consists of generating a set of I - V curves of the device of interest using Spice software. These curves are the input samples of the Artificial Neural Networks, from which it is intended to predict the different parameters such as threshold voltage, transconductance and contact resistance, from each sample curve. By generating the training set itself, it is possible to label each sample curve, which allows the type of learning to be supervised. The results show that ANNs provide parameters with which it is possible to model physical measurements with error rates of less than 5% when extracting the first two parameters, and errors of between 0.06% and 4.62%, when extracting the three parameters. In addition, a comparison was made between the results of the ANNs and the analytical extraction of parameters.

1 Introduction

Nowadays, electronic simulation allows the design and testing of electronic devices or circuits before their manufacture, without the need to commit resources [1, 2]. Although it also allows the analysis and evaluation of devices already manufactured, which allows understanding their operation and making future optimisations. This is of great support to the growing emergence of new technologies and increasingly efficient devices [3].

Simulation software works by using mathematical models of the different devices supported, these models represent the behaviour of the devices in the real world. The models are made up of variables known as parameters, which can change from one device to another. The value assigned to them will define whether the simulation will correspond to the real-world behaviour.

For this reason, it is of utmost importance to know the parameters of the devices, for which a process known as parameter extraction is performed. Analytical methods can be found in the literature to

Address correspondence to E-mail: roberto.cvg@hotmail.com

perform this task [4–15], and methods based on Genetic Algorithms (GA) [16–19] and Fuzzy Logic (FL) [20], have also been proposed as an alternative. These methods require a lot of experience and knowledge in the operation of the device models or defining the correct fitness function as in the case of GAs, to obtain good results. Not to mention emerging technologies, although more efficient devices, result in complex models with a large number of parameters, making parameter extraction a complex task. Recently, the application of machine learning in the process of parameter extraction has begun to be explored [21, 22].

Therefore, a method using ANNs is proposed to perform parameter extraction applied on TFT IGZO with Al contacts. The ANNs are trained to identify the parameters of a set of samples, consisting of I – V curves of the device. This work shows that using ANNs are an alternative to provide parameters that allow a good fit between physical measurements and those simulated with the extracted parameters. The contribution of this study is a method of parameter extraction with which it is not necessary to have extensive experience in this task, nor to have a deep knowledge of the mathematical models of the devices. Although basic knowledge of the operation of NNs is required, their programming and implementation is not complicated due to the specialised Machine Learning libraries available today in different programming languages. On the other hand, using this method it is not necessary to process each I – V curve individually as it is in the analytical methods. A trained NN can receive a single file with many curves and provide results almost instantaneously. In this paper not only a method for extraction is reported, but there are also two experiments, one in which the flexibility of NNs to compensate for the parameters to be extracted using their acquired experience is tested. The second one proposes a quick solution for when there are not enough samples to perform extraction in devices with different dimension.

Due to the early application of supervised learning for parameter extraction, the extraction of the most relevant transistor parameters such as threshold voltage, transconductance and contact resistance has been limited. The method was tested by extracting in different transistors manufactured by the Nanoscience, Micro and Nanotechnologies Centre of the National Polytechnic Institute, Mexico.

The rest of the paper is divided as follows: Sect. 2 presents a quick review of the state of the art; Sect. 3 introduces the TFT model and briefly presents the fundamentals of ANNs; Sect. 4 introduces the proposed method for parameter extraction; Sect. 5 presents the experiments carried out; Sect. 6 presents the results and discussion; and the paper closes with conclusions in Sect. 7.

2 Previous works

This section presents work related to the extraction of parameters in different types of transistors. Amongst the methodologies based on mathematical analysis, the following works stand out.

In [4], a method for parameter extraction in a short-channel amorphous InGaZnO TFT using experimental and simulated measurements was proposed. In [5], a comprehensive review of the most commonly used methods for calculating the threshold voltage in MOSFET devices was presented and includes work where the methods were applied to real devices. In [6] a parameter extraction for the AIM-Spice model of an amorphous TFT, avoiding non-linear optimisation, was proposed, the method was based on the integration of experimental measurements, and was tested in the linear and saturation regimes. A discussion on the use of technology based on OTFTs, OLETs, OLEDs and OPVs was presented in [7], concentrating on OLET devices, which have optical and electrical properties, the authors analysed the device in terms of drive current, threshold voltage, mobility and others. With the transfer curve analysis, they extracted the mentioned parameters. In [8], a method for parameter extraction of a polycrystalline silicon (poly-Si) TFT in weak conduction and triode region was proposed. They used two functions based on the integration of experimental measurements. In [9], an analysis of the behaviour of the OTFT, in the top (TC) and bottom (TB) contact, was made by modifying the shell drift model with respect to the series resistance. Using the mathematical model, the contact resistance, mobility and drain current, in linear and saturation regime, were extracted. In [10], a comparison of the performance of top and bottom contact OTFT structures was performed using two-dimensional numerical simulations. An estimation of the contact resistance using the conventional transmission line method and modified transmission line

method (M-TLM) was also performed. In [11], an accurate and broadband method for extracting parameters from a small-signal model in heterojunction bipolar transistors (HBTs) was presented. An equivalent HBT circuit was used for the extraction of access resistance and parasitic conductance. In [12], the limited behaviour of the contact resistance in an organic pentacene transistor is described by simulation and mathematical modelling. An analysis of a difference to the Shockley model, which is due to a non-linear behaviour of the contacts in organic devices, was performed. In [13], the specific contact resistance (p_c) was determined between an amorphous indium-gallium-zinc-oxide (IGZO) semiconductor and different contact electrodes was found using TFT transistors. Chemical states of the contacts/semiconductor interfaces were used and analysed with X-ray photoelectron spectroscopy (XPS) to explain the differences in resistance. It was found that the lowest p_c was obtained using Ti/Au. In [14] three methods were used to extract the contact resistance of CNTFETs. The transfer length method (TLM) and two variants of the Y-function method were applied. It was found that the standard Y-function method does not give correct resistance values. In [15], a methodology was proposed to determine the asymmetrical source and drain resistances R_S and R_D from MoS₂ field-effect transistors (EM-FETs). By combining capacitance–voltage (C – V) and current–voltage characteristics, these resistances were extracted separately. First, the authors used C – V frequency dispersion from 0.3 to 10 kHz, then R_S and R_D were characterised by removing parasitic capacitances from the pad. The proposal was compared with the channel resistance method.

Also, works for parameter extraction based on GAs can be found. In [16] the authors proposed a hybrid algorithm for parameter extraction in OTFT transistors, based on the bee colony evolutionary algorithm (EA) to which they added a GA operator. The proposed algorithm extracted parameters from two devices and the results were compared with a simple GA. In [17] a machine learning approach to nonlinear regression with six input variables was used to measure the impact of process variability on the threshold voltage of a silicon-on-insulator (SOI) junctionless transistor (JLT). The GA was implemented in MATLAB to test the stated hypothesis. In [18], a GA was used to optimise the parameters of a

Carbon Nanotube Field Effect Transistor (CNFET), the performance of these devices depends on parameters such as the CNT diameter, the number of nanotubes and the spacing between the inter-tubes. The aim of this work was to minimise the Power-Delay Product (PDP). A compact analytical model for organic field effect transistors (OFETs) is presented in [19]. The proposed model describes the behaviour of the device in the above-threshold and below-threshold regime. This was achieved by calculating the total OFET current as the sum of both components where was added a transitive function to smooth the junction. A GA-based approach was also used as a tool for parameter extraction.

In [21] there is one of the few works where machine learning is used to extract parameters from IGZO TFTs, where 618 samples of I–V curves were used. In this work the authors start with an unsupervised learning approach using K-means to group the samples into 4 performance categories, and then use conventional NNs and Deep Neural Networks (DNN) to perform the parameter identification. By using physical measurements, they first performed analytical extraction, which allowed them to use the supervised learning approach. Their results show error rates of up to 10.6% in mobility extraction, up to 131.25% in subthreshold swing, 26.28% in threshold voltage and up to 71.36% in on/off current ratios using NNs. In [22] a work of extraction of extraction in Insulated Gate Bipolar Transistor (IGBT) is presented, where the extracted parameters were breakdown voltage (BV), on-state voltage (V_{on}), static latch-up voltage (V_{lu}), static latch-up current density (J_{lu}) and threshold voltage (V_T). The authors propose a two-layer NN to predict the above parameters using as inputs device structural parameters (not I–V curves) such as N-drift doping, N-buffer doping, P-well doping, P + well doping, N-drift thickness, N-buffer thickness and channel length. Their results showed average error rates of up to 7.7%. In both works the extraction of R_C , which is a complicated parameter to extract due to the need for physical measurements of devices with different dimensions, was not performed.

3 Theoretical background

3.1 TFT model

TFTs (Thin Film Transistors) are MOSFET (Metal Oxide Semiconductor Field Effect Transistor) devices, and are used in today's displays, such as those in mobile phones and smart TVs, whose development is progressing rapidly. The mathematical model describing the current behaviour of a TFT is given by Eq. (1) (model implemented in AIM Spice [23]).

Current I_D above the threshold voltage occurs when $V_{GT} > 0$.

$$I_D = \begin{cases} \mu_{FET} C_{ox} \frac{W}{L} \left(V_{GT} V_{DS} - \frac{V_{DS}^2}{2\alpha_{sat}} \right), & V_{DS} < \alpha_{sat} V_{GT} \\ \mu_{FET} C_{ox} \frac{W}{L} \left(\frac{V_{GT}^2 \alpha_{sat}}{2} \right), & V_{DS} \geq \alpha_{sat} V_{GT} \end{cases}, \quad (1)$$

where μ_{FET} is the effective mobility of the device, C_{ox} is the capacitance of the oxide layer, given by the dielectric permittivity constant (ϵ_i) divided by the thickness of the oxide layer (T_{ox}), W and L are the width and length of the channel respectively; V_{GT} is the result of the applied gate and source voltage, minus the threshold voltage (V_T); V_{DS} is the applied voltage between drain and source terminals. Finally, α_{sat} is the proportionality constant of V_{sat} (saturation voltage).

The current I_D below the threshold voltage is given by Eq. (2).

$$I_{sub} = MUS \times C_{ox} \frac{W}{L} V_{sth}^2 \exp\left(\frac{V_{GT}}{V_{sth}}\right) \left[1 - \exp\left(-\frac{V_{DS}}{V_{sth}}\right) \right], \quad (2)$$

where MUS stands for electric mobility; V_{sth} is the product of ETA (subthreshold ideality factor) and V_{th} (Eq. 3).

$$V_{th} = k_B \times TEMP/q, \quad (3)$$

where k_B is the Boltzmann's constant, TEMP is the temperature and q is the electron charge magnitude. A part of the model can be summarised by the parameter of transconductance (KP), this parameter is widely used by circuit and device designers, which is given by Eq. 4, where ϵ_0 is the vacuum permittivity.

$$KP = \mu_{FET} C_{ox} = \mu_{FET} \frac{\epsilon_i \epsilon_0}{t_{ox}}. \quad (4)$$

KP is one of the parameters to be extracted in this research, and with which it is possible to calculate the mobility which is a parameter used by people who analyse the behaviour of devices [24]. When the induced channel extends from the source to the drain, the transconductance can be rewritten as:

$$\beta = KP \frac{W}{L}. \quad (5)$$

Considering that the material of the device presents resistance, the mobility and electric current will be inversely proportional to the resistance. And the total resistance R_T of a device is given by Eq. (6).

$$R_T = R_C + R_{CH} = V_{DS}/I_D, \quad (6)$$

$$R_C = R_S + R_D, \quad (7)$$

where R_C is the contact resistance, R_{CH} is the channel resistance, R_S and R_D are the resistances at the source and drain terminals respectively [12, 14].

3.2 Artificial neural networks

ANNs or just NNs are a simplified approximation of the brain, represented by an ensemble of artificial neurons. The artificial neuron concept was proposed by Warren S. McCulloch and Walter Pitts in 1943 [25].

NNs have been widely used for pattern recognition, big data analysis, feature extraction, classification, regression, system identification, amongst other applications [26]. NNs are inspired on the biological functioning of how living things learn from experience. NNs acquire knowledge from a set of representative examples or samples in a process known as training. There are three basic ways in which the network learns: supervised, unsupervised and reinforcement learning.

In supervised learning, the set of samples is pre-labelled, i.e. for each of the examples, the correct answer is known, usually used for classification and regression problems. In unsupervised learning, the answer corresponding to each training sample is not known, the NN adopts the underlying structure of the training set; usually, the unsupervised NNs are employed for clustering problems. In reinforcement learning, the goal is to obtain intelligent agents capable of taking the best action in given situations or environments, with the aim of obtaining the

maximum reward or the least punishment. Reinforcement learning is used in control, autonomous systems and game theory [26].

Formally, the output of a neuron is given by Eq. (8), where the product of an input p and a synaptic weight w , plus a bias b , is evaluated in an activation function f .

$$a = f(wp + b). \tag{8}$$

The activation function f is responsible for setting a range of values over which the network will be working, and the function is selected by the nature of the problem.

A neuron can have multiple inputs (p_R), and the output of the neuron is given by Eq. (9), where each input is multiplied by a corresponding synaptic weight. The output of the multi-input neuron can be written in vector form (Eq. 10), where W is the vector of weights and p is the vector of inputs.

$$a = f(w_{1,1}p_1 + w_{1,2}p_2 + w_{1,3}p_3 + \dots + w_{1,R}p_R + b), \tag{9}$$

$$a = f(Wp + b). \tag{10}$$

In a layer of neurons s , each neuron has an output a_i . Each individual input $p_1 \dots p_R$ of the input vector p is connected to each neuron and is multiplied by the corresponding weight of the matrix W . Each neuron has its bias b_i and its activation function f . So, the outputs of the neurons are given by Eqs. (11) and (13).

$$a_1 = f_1(w_{1,1}p_1 + w_{1,2}p_2 + w_{1,3}p_3 + \dots + w_{S,R}p_R + b_1), \tag{11}$$

$$\begin{aligned} a_2 &= f_2(w_{2,1}p_1 + w_{2,2}p_2 + w_{2,3}p_3 + \dots + w_{S,R}p_R + b_2) \\ &\vdots \end{aligned} \tag{12}$$

$$a_S = f_S(w_{R,1}p_1 + w_{R,2}p_2 + w_{R,3}p_3 + \dots + w_{S,R}p_R + b_S). \tag{13}$$

In vector form the output of a layer of neurons is given by Eq. (14). Where, b is the bias vector.

$$a = f(Wp + b). \tag{14}$$

In a network with a number n of layers, the output generated by each neuron in the first layer is the input to the second layer and so on. So, each layer has its vector of inputs p , its matrix of weights w , a bias

vector and its activation function. The output of 3 layers and the total network output is given by the following equations. The output of network layer n is given by Eq. (19).

$$a^1 = f^1(W^1p + b^1), \tag{15}$$

$$a^2 = f^2(W^2a^1 + b^2), \tag{16}$$

$$a^3 = f^3(W^3a^2 + b^3), \tag{17}$$

$$a^3 = f^3(W^3f^2(W^2f^1(W^1p + b^1) + b^2) + b^3), \tag{18}$$

$$a^n = f^n(W^n a^{n-1} + b^n). \tag{19}$$

As mentioned, learning is obtained in the training stage, where the error between the expected output and the obtained output is obtained in an iterative way, known as epochs. In supervised learning, the backpropagation algorithm is the most widely used. This gradient-based algorithm aims to reduce the mean square error MSE (Eq. 20). In training, at each epoch, the synaptic weights w and the bias are modified, obtaining the minimum error or up to the defined number of epochs.

$$F(x) = E[e^2] = E[(t - a^2)], \tag{20}$$

$$w_{ij}^m(k + 1) = w_{ij}^m(k) - \alpha \frac{\partial F}{\partial w_{ij}^m}, \tag{21}$$

$$b_i^m(k + 1) = b_i^m(k) - \alpha \frac{\partial F}{\partial b_i^m}, \tag{22}$$

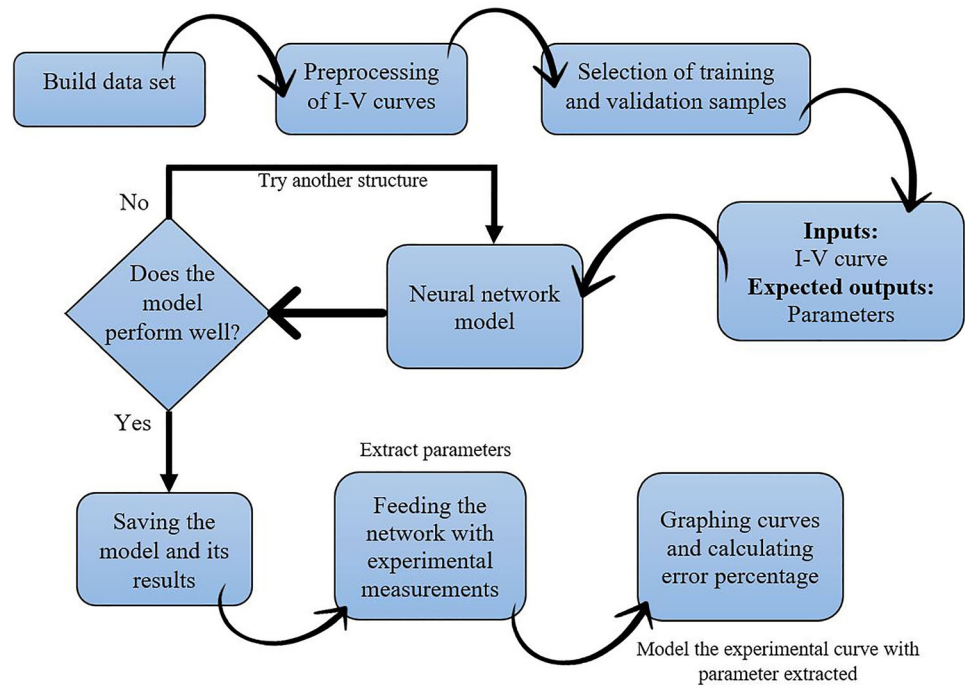
where t is the target, or desired output, and a , is the output obtained, k is the epoch number, m is the layer number, i and j are the identifier of the weights and bias, α is the learning factor, which determines the change in w and b [27, 28].

4 Parameter extraction method

Figure 1 summarises the methodology for parameter extraction using NNs. The extraction steps are summarised as follows:

- (1) A training dataset is constructed using I-V curves obtained by LTspice [29] simulation.
- (2) Curve data such as V_{GS} and I_D is normalised or rescaled with Eq. (24).
- (3) The total samples ($I-V$ curves) are divided into two sets, one for training and one for validation.

Fig. 1 General diagram of the methodology for the extraction of parameters



- (4) For each simulated curve (input) its parameters (outputs or targets) are known.
- (5) With the data sets ready, the training of NNs with different numbers of layers and neurons is started.
- (6) If the determination coefficient of the trained NN model (R^2) is greater than 90%, then, the model and its results are saved.
- (7) Once the NN has been trained, it is used to extract the parameters of physical measurements.

Finally, the extracted parameters are employed to simulate the device, and compare with the physical measurements and calculate the percentage error between the curves.

4.1 Training dataset and preprocessing

The dataset of samples for training the NNs consists of a set of I - V curves of the transistor. These curves are obtained by simulation, using a given model (defined in the simulator). In this case, level 1 of the SPICE Model for MOSFETs is used, although it is a basic model, it was used because it allowed modelling the behaviour of the TFTs. To build the training dataset, the first step is to define the parameters of interest, i.e. the parameters that will be extracted. In this case the parameters of interest are KP , V_T and R_C .

The reason of this selection is that these electrical parameters are frequently used to characterise and model the TFTs electrical behaviour. The simulations will be performed by varying the values of the parameters of interest. The range of variation and the number of simulations that make up the training set will define how knowledgeable or robust the NN model will be. In the present work, an initial analytical parameter extraction was performed on one of the physical transistors, and these results were used as the basis for defining the range of values with which the simulation would be performed. It is important to note that this range of values must be sufficiently wide for the NN to provide adequate results (with low error percentages). Depending on the results obtained from the extraction using the trained NNs, the training sets can be complemented with more samples to improve the performance of the NNs.

The process for carrying out the simulations is as follows. To extract KP , V_T and R_C , starting to set in the simulator (optional) in this case LTspice, KP and V_T are set to their initial value $KP = 1 \times 10^{-6} \text{ A/V}^2$ and $V_T = 1.5 \text{ V}$, keeping KP and V_T as constants, R_C is swept from 0 to $6 \times 10^3 \Omega$ with increments of $1 \times 10^3 \Omega$. Then the value of V_T increases, KP remains constant, $V_T = 1.8 \text{ V}$ and again R_C is swept. When R_C was swept for each value of V_T (up to 3.6 V), now KP

is increased and V_T starts with the initial value, $KP = 1.4 \times 10^{-6} \text{ A/V}^2$, $V_T = 1.5 \text{ V}$ and so on until KP reaches its maximum value. This is the most time-consuming step, if done by one person. In each run, the I_D data are stored. The approximate time is 120 samples of $I-V$ curves in 1 h. The process in which the parameters are swept could be done by programming and using the mathematical models of the device Eqs. (1) to (7), although the possibility of human error in programming is not ruled out, so it was decided to use simulation software which uses the same mathematical models and ensure that the $I-V$ samples are accurate. For the experimentation the following data sets were created. The first set (289 samples) to extract the parameter KP and V_T is given by Eqs. (23) and (24). A second set was made to extract KP , V_T and R_C (448 samples), which is given by Eqs. (25), (26) and (27). The third set (392 samples) is given by Eqs. (28), (29) and (30). And finally, the fourth set is made up of the union of the second and third set with a total of 840 samples.

$$K = \left\{ 1 \times 10^{-6} / V^2 + k\Delta \mid k = 0, 1, \dots, 16 \right\}, \tag{23}$$

where $\Delta = 0.2 \text{ A/V}^2$.

$$V = \{ 1.0V + k\Delta \mid k = 0, 1, \dots, 16 \}, \tag{24}$$

where $\Delta = 0.2 \text{ V}$.

$$K = \left\{ 1 \times 10^{-6} \text{ A/V}^2 + k\Delta \mid k = 0, 1, \dots, 7 \right\}, \tag{25}$$

where $\Delta = 0.4 \text{ A/V}^2$.

$$V = \{ 1.0V + k\Delta \mid k = 0, 1, \dots, 7 \}, \tag{26}$$

where $\Delta = 0.3 \text{ V}$.

$$R = \{ 0\Omega + k\Delta \mid k = 0, 1, \dots, 6 \}, \tag{27}$$

where $\Delta = 1 \times 10^3 \Omega$.

$$K = \left\{ 1.2 \times 10^{-6} \text{ A/V}^2 + k\Delta \mid k = 0, 1, \dots, 6 \right\}, \tag{28}$$

where $\Delta = 0.4 \text{ A/V}^2$.

$$V = \{ 1.0V + k\Delta \mid k = 0, 1, \dots, 7 \}, \tag{29}$$

where $\Delta = 0.3 \text{ V}$.

$$R = \{ 0\Omega + k\Delta \mid k = 0, 1, \dots, 6 \}, \tag{30}$$

where $\Delta = 1 \times 10^3 \Omega$.

The voltages used were V_{GS} from -6 to 6 V and $V_{DS} = 6 \text{ V}$, because the physical measurements were taken at these voltages.

The pre-processing step consists of rescaling the data, usually in the ranges $[-1, 1]$ or $[0, 1]$, depending on the nature of the data. This step allows each sample to be transformed in a way that gives more information to the NN to facilitate pattern detection. In this case, each input was divided between the highest value it could have, thus obtaining the range $[0, 1]$.

4.2 NNs training

When solving a classification or regression problem using NNs it is necessary to train more than one network model (number of layers and neurons), as there is no way of knowing which NN will be the most suitable for each problem. Although it has been observed that for regression problems, the higher the number of layers and neurons, the higher the learning rate, this is not always true [27]. So, networks with different numbers of layers, neurons, activation functions and learning factors must be tested until the models with the highest learning from the data are found.

The best way to find the correct values of the hyperparameters (characteristics) of the network is to use the grid search [30]. This technique consists of sweeping through the different hyperparameters from a pre-determined initial value to a final value, combining all of them to identify which one provides the best performance. As mentioned above, in order to find the best learning NN model, approximately 24 NNs were trained, of which 8 of them performed the best. Table 1 shows the top 8 models that were trained to perform TFT parameter extraction. The function in the hidden layers is *Relu* and in the output layer *Linear*. Each model was trained with 2000 epochs. NN models with a smaller number of layers

Table 1 Top 8 trained NNs models

NN model	No. layers	No. of neurons in each layer
1	2	128/64
2	2	256/32
3	2	256/64
4	2	256/128
5	2	256/256
6	3	128/64/32
7	4	256/128/64/32
8	5	256/128/64/32/16

and neurons presented underfitting (lack of learning) and with larger NNs, learning does not increase (a complex NN will not always be the best), and in computing when two NN models have the same performance, it is better to select the smaller one.

The training of the different NNs models was implemented using Google Colab [31], which is a Jupyter-based environment for programming in the Python language, and it is possible to make use of Google computing resources. The library used was Keras from Tensorflow [32] which is intended for training different types of NNs.

4.3 NN evaluation

Metrics for evaluating the results of parameter prediction (extraction) are presented in this subsection. In addition to the MSE and R^2 in the parameters in each of the NN models. In the extraction of parameters from the physical measurements, the percentage error was calculated using the area under the curve (trapezoid method) in Eq. (31) between the experimental curve and those modelled with the extracted parameters.

$$\% \text{ error} = \left| \frac{\text{area}_{\text{ext}} - \text{area}_{\text{real}}}{\text{area}_{\text{real}}} \right| \times 100. \quad (31)$$

In computing, the selection of the best NN model is done by simply taking the model with the highest accuracy in case of classification or R^2 in regression, considering only the results on the validation samples. In this work, the best model would be the one that obtained the highest R^2 in the validation samples, for example, model 8 in Table 3. But it is possible to use physical measurements as a test set to perform the parameter extraction, using not only the model 8 that was the best, in this work we tested the best 8 out of 24 models. In this way, and for this problem, it is recommended to have more than one NN model trained, and the best one will be the one that allows to obtain the lowest error percentage.

5 Experiments

This section describes the different experiments and analyses that were performed during the training of the NNs. This research was developed incrementally, starting with the extraction of two parameters (KP and V_T), once it was verified that the NNs extracted

parameters from physical measurements with good accuracy, extraction of KP , V_T and R_C was carried out. For the extraction tests once the training of the NNs models was completed, physical measurements of IGZO TFTs were used, which were made with corning glass and at room temperature (25 °C), it should be mentioned that the proportionality saturation voltage is not required by the Spice model used. The geometrical parameters are presented in Table 2. Because there is more than one TFT transistor, and also with different dimensions (Table 2), from now on, each physical measurement will be identified with a number, e.g. T1M1 refers to transistor 1 with dimensions $W = 80 \text{ um}$ and $L = 80 \text{ um}$, M1 refers to the first of all measurements with the same dimensions. Also, can be found “lin” to refer to measurement in the linear regime, if not present the physical measurement is in the saturation regime.

5.1 Experiment 1

In the first experiment, the first data set was used to train NNs to learn to extract the KP and V_T parameters in both the saturation and linear regimes. The training data are intended for a TFT with a $W = 80 \text{ um}$, $L = 80 \text{ um}$ and a $T_{\text{ox}} = 15 \text{ nm}$ (T1). After the NN models are trained and stored, they are tested to perform the extraction of physical measurements. The values provided by the NN must be treated in the inverse way in which they were pre-processed to obtain the real value of the parameter to be used in the electronic simulator.

The physical measurements used for the tests in this experiment correspond to T1 with two measurement conditions (V_{GS} application). It was observed that the behaviour changed slightly if V_{GS} was applied from negative to positive (NP) and vice

Table 2 Dimensions of the transistors from experimental measurements

Transistor	W (um)	L (um)	T_{ox} (nm)	No. of measurements
T1	80	10	15	3
T2	40	10	15	2
T3	80	5	15	2
T4	40	40	15	1
T5	80	20	15	1
T6	80	80	15	1
T7	40	5	15	1

versa (PN). If no NP or PN is specified, the measurement was taken with V_{GS} sweep from negative to positive.

Figure 2 shows how the device (T1) changes its transfer curve when measured from negative to positive and from positive to negative. Not all transistors were measured in both ways, though. If not specified with a PN, it means that the physical measurement was obtained with V_{GS} from negative to positive. The analysis of the hysteresis that occurred in the fabricated transistors will be reported in a separate article.

5.2 Experiment 2

In this experiment, no training of NNs is performed, here was used one of the NN models that was trained in experiment 1. During experiment 1, the NNs learned from samples that did not have contact resistance. The aim of this second experiment is to analyse the results of the NN already trained when it is fed with a sample to which R_C was added.

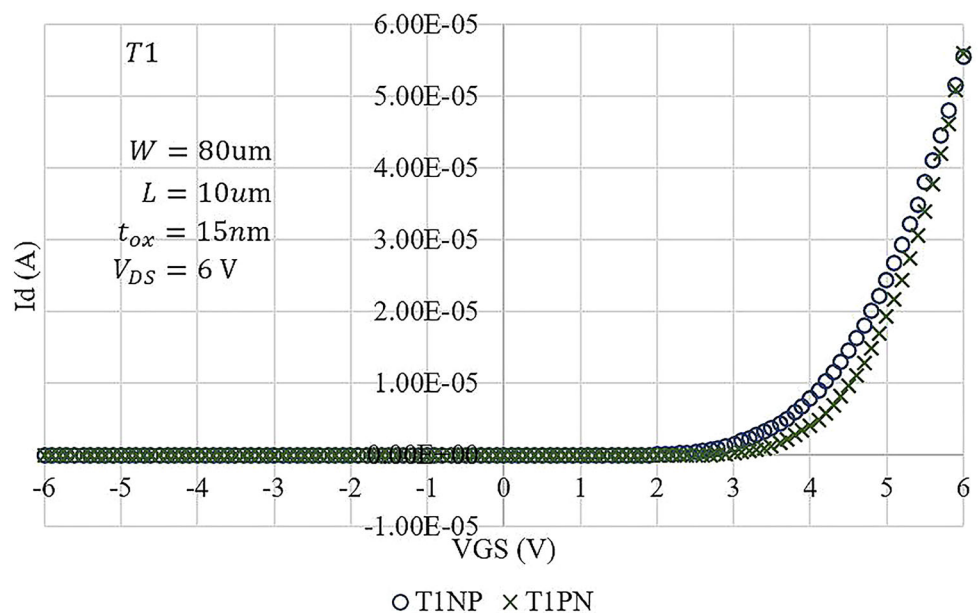
To achieve this, the parameters extracted from KP and V_T in experiment 1 were used, which allowed the behaviour of T1 to be modelled accurately. Using again the LTspice simulator, the parameters KP and V_T were set as constants, and 6 runs were performed, where in each of them the contact resistance had the following values: 0Ω , $1 \times 10^3 \Omega$, $4 \times 10^3 \Omega$, $16 \times 10^3 \Omega$, $32 \times 10^3 \Omega$ and $64 \times 10^3 \Omega$.

The 6 simulated samples with R_C increasing from 0 to $64 \times 10^3 \Omega$ were stored and pre-processed to feed the NN model 1 and extract its KP and V_T parameters. This experiment allows us to demonstrate how NNs can make use of the knowledge acquired during training and approximate results for input samples that have information that was not necessarily learned.

5.3 Experiment 3

The main disadvantage of NNs and other machine learning methods is the need for sufficient samples to learn, plus the fact that each dataset is problem-specific. So, if an NN was trained to extract parameters with certain dimensions such as $W = 80 \mu\text{m}$ and $L = 10 \mu\text{m}$, as in experiment 1, the NN would not be able to extract parameters from a transistor with different dimensions and it would be necessary to generate the training set for that other device. The process of making a data set for each of the transistors available in this research (Table 2) would become a tedious task. Furthermore, for some combination of W and L the I_D current would be the same as in the case of T4 and T6. In a future extractor, or even in a commercial extractor, generalised and specialised system for extraction for an established technology, where the geometrical factors are always the same, including the parameters W , L and T_{ox} during NN's training would be worthwhile and absolutely necessary.

Fig. 2 T1 transferential curve with different measurement condition



In the third experiment, the geometrical parameters of the samples that make up the first dataset were set to be removed. This is achieved by normalising the I_D current of each of the samples. This normalisation was performed using Eq. (32), where $W = 80$ μm and $L = 10$ μm . The NN models (first three in Table 1) were then re-trained with the normalised sample set. After training, the NNs are ready to perform the extraction tests on the physical measurements, in order to feed the NNs, the measurements must also be normalised using their own channel widths and lengths.

In this experiment, a way was proposed to avoid the need for examples for each problem to be solved by the NNs.

$$I_{\text{norm}} = \frac{L}{W}(I_D). \quad (32)$$

5.4 Experiment 4

In the fourth experiment, training of NN models was carried out for dataset number 2 (Eq. 25, 26, 27), training was carried out for dataset number 3 (Eqs. 28, 29, 30) and training was also carried out by joining these datasets. This fourth experiment aimed to obtain NNs capable of extracting KP , V_T and R_C . The data were not normalised to remove the W and L parameters (Eq. 32). For this experiment the datasets are designed to extract from a TFT with $W = 80$ μm and $L = 10$ μm .

The training performed on the first dataset was done as would commonly be done in computing, focussing on obtaining the most learning possible with the highest score on the validation samples. As mentioned in 4.2, this is achieved by treating the data in a way that provides the most knowledge for better pattern detection. For set 3 and the fourth set formed by the union of the second and third sets, the I - V curve samples did not undergo a special pre-processing to increase learning and have an excellent score in the validation samples, only the I_D current was increased one hundred times, with the purpose of having larger values and facilitating the identification of R_C , since as shown in Fig. 5a, the effect of $R_C < 4 \times 10^3 \Omega$ is not noticeable to the naked eye, and only the I_D values were used when $V_{GS} > 1.9$ V, which is the most significant section of the curve.

After completion of the NN models training for the different datasets, the extraction tests were performed using the physical measurements of the T1.

6 Results and discussion

This section presents the results obtained in the different experiments described in the previous section. To validate the results, a comparison of the modelling obtained from the physical measurements was made, using the parameters extracted by the NN models and an extraction performed by the analytical method (using Eq. 1). Thereby, the accuracy achieved by the proposed method is demonstrated. The analytical extraction was only performed to extract KP and V_T in saturation regime, analytical extraction of R_C is not available.

6.1 Results of experiment 1

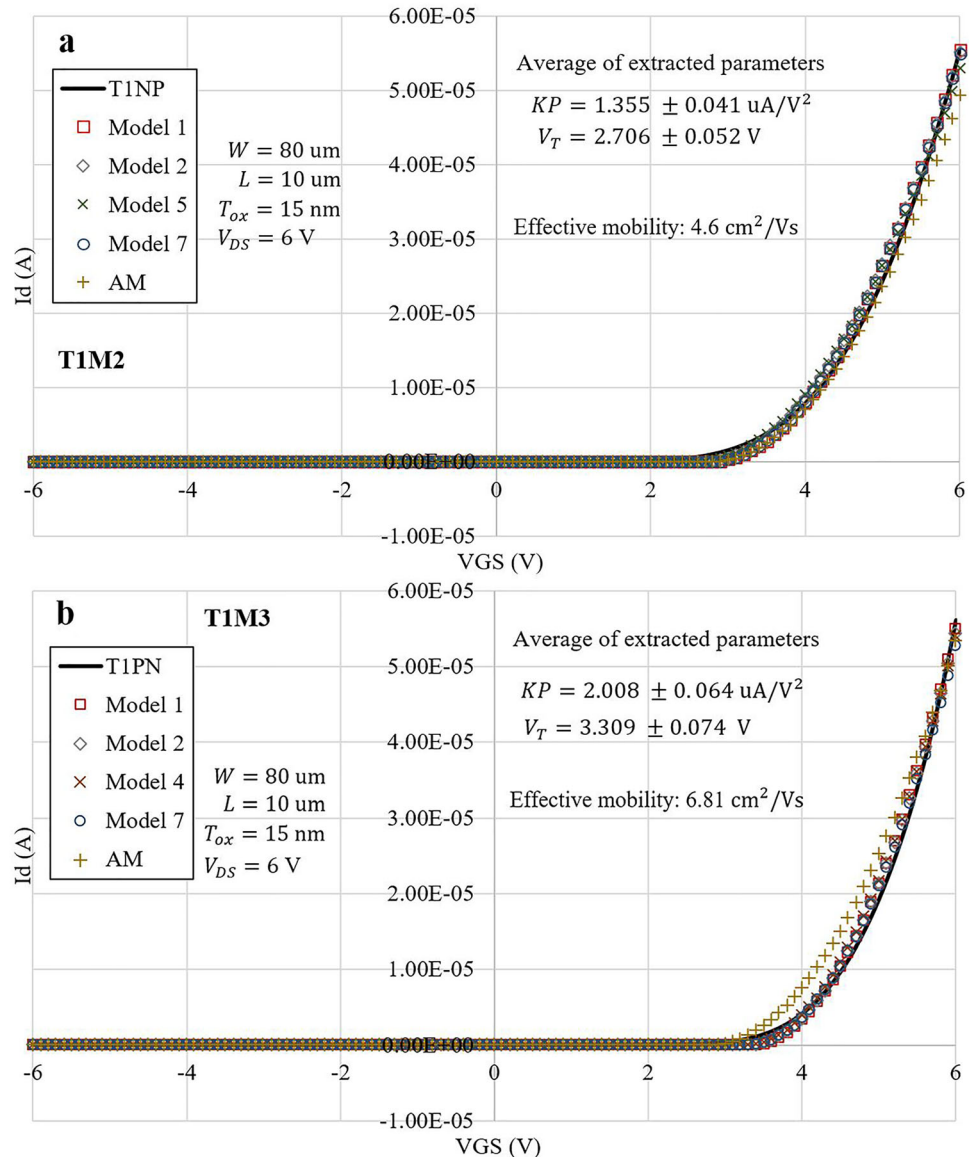
The training of the NN models (Table 1) showed good performance (no overfitting or underfitting). Table 3 presents the MSE and R^2 obtained by each model in the validation samples, in both saturation and linear regimes. The models can predict the parameters KP and V_T up to 99% average precision rate.

Figure 3a shows the modelling with the extracted parameters at T1, measured from negative to positive (measurement 2), where NN model 1 obtained an error rate of 1.67%, model 2 obtained 3.01%, model 5 obtained 2.07%, model 7 obtained 1.42% and AM obtained 9.67%. Figure 3b shows the modelling of the T1 measurement, measured from PN (measurement

Table 3 Dimensions of the transistors from experimental measurements

NN model	Saturation		Linear	
	MSE	R^2	MSE	R^2
1	1.150×10^{-4}	0.997	3.964×10^{-5}	0.999
2	4.747×10^{-4}	0.991	5.598×10^{-5}	0.998
3	3.119×10^{-4}	0.994	4.621×10^{-5}	0.999
4	8.763×10^{-4}	0.998	2.714×10^{-5}	0.999
5	1.802×10^{-4}	0.996	2.432×10^{-5}	0.999
6	1.476×10^{-4}	0.997	3.141×10^{-5}	0.999
7	1.643×10^{-4}	0.996	7.609×10^{-5}	0.998
8	1.139×10^{-4}	0.998	2.348×10^{-5}	0.999

Fig. 3 TFT modelling with extracted parameters in negative-to-positive and positive-to-negative measurement (saturation regime), **a** corresponds to the measurement supplied from negative to positive, and **b** from positive to negative



3), where NN model 1 obtained an error rate of 2.61%, model 2 obtained 2.03%, model 4 obtained 3.43%, model 7 obtained 0.21% and AM obtained 19.81%.

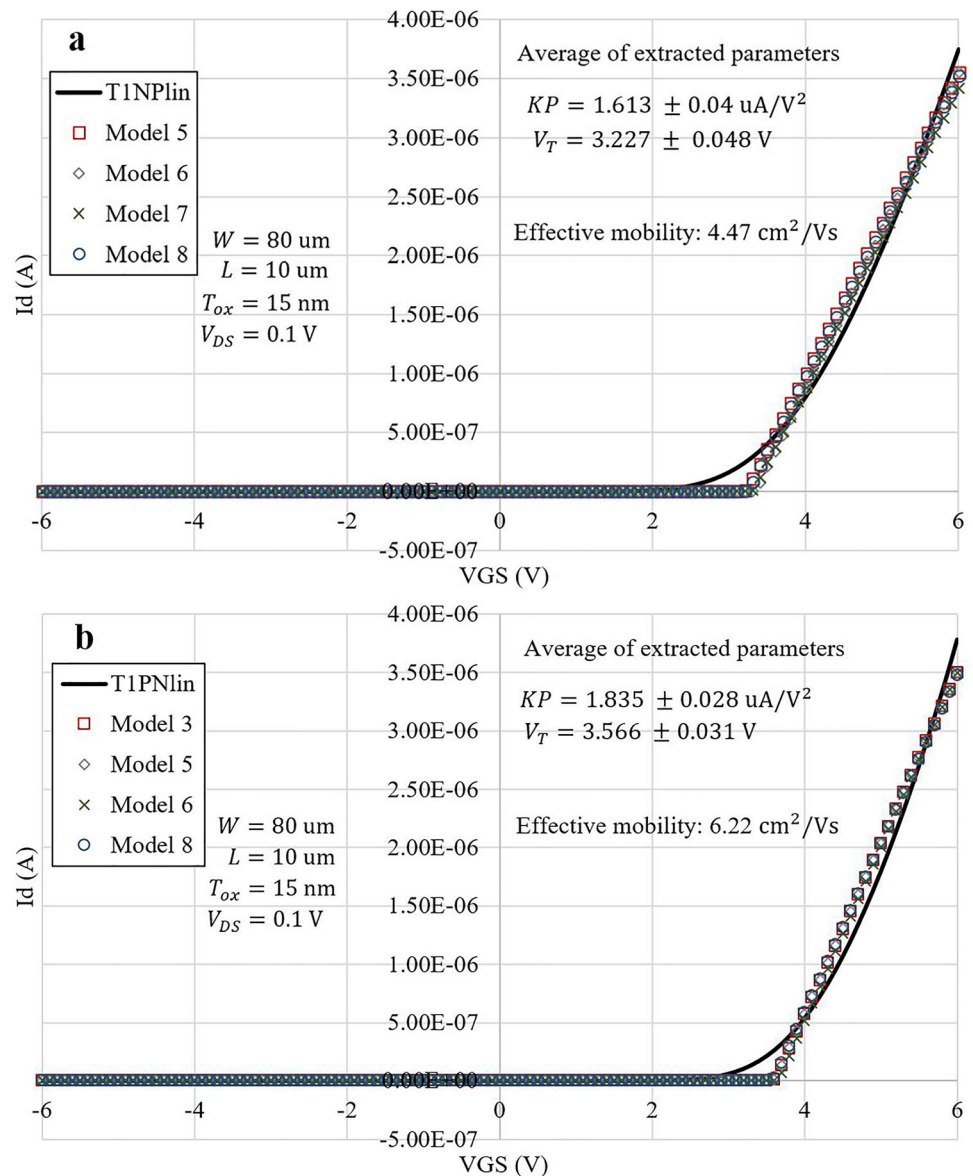
Figure 4a shows the modelling with the parameters extracted from T1 in linear regime. In Fig. 4a (negative to positive measurement), NN model 5 obtained an error rate of 3.61%, model 6 obtained 1.25%, model 7 obtained 3.47% and model 8 obtained 2.03%. In Fig. 4b (positive to negative measurement), NN model 3 had an error rate of 2.09%, model 5 had 2.26%, model 6 had 0.17% and model 8 had 2.20%. In the linear regime measurements, extraction with an analytical method is not available.

6.2 Results of experiment 2

In order to analyse the effect of R_C , the TFT was simulated with the parameters extracted in physical measurement 1, of T1 by model 1. Where $KP = 1.6107 \times 10^{-6} \text{ A/V}^2$ and $V_T = 2.667 \text{ V}$ are set as constants. Figure 5a shows that there is not significant change from 0 to $4 \times 10^3 \text{ }\Omega$, but from $16 \times 10^3 \text{ }\Omega$ the electric current starts to decrease, having a large loss at $64 \times 10^3 \text{ }\Omega$.

By feeding the NN model 1 with the transfer curves with different resistances (Fig. 5a) it was found that the NN compensates the value of KP and V_T to obtain parameters that fit the input curves. Figure 5b shows the fittings obtained by using the

Fig. 4 TFT modelling with extracted parameters in negative-to-positive and positive-to-negative measurement (linear regime), **a** corresponds to the measurement supplied from negative to positive, and **b** from positive to negative



parameters extracted by the NN model 1, I - V curve with a R_C of $16 \times 10^3 \Omega$, obtained an error rate of 1.64%, I - V curve with $32 \times 10^3 \Omega$, obtained 2.38% and I - V curve with $64 \times 10^3 \Omega$ obtained 11.48%.

Figure 6a shows the percentage of error obtained between the curves with increased resistance and those modelled with the extracted parameters, where it can be seen that the error increases when R_C is higher. Which means that the NN is able to compensate the parameters at KP and V_T on curves with resistance of about $32 \times 10^3 \Omega$, for larger values the error starts to become significant.

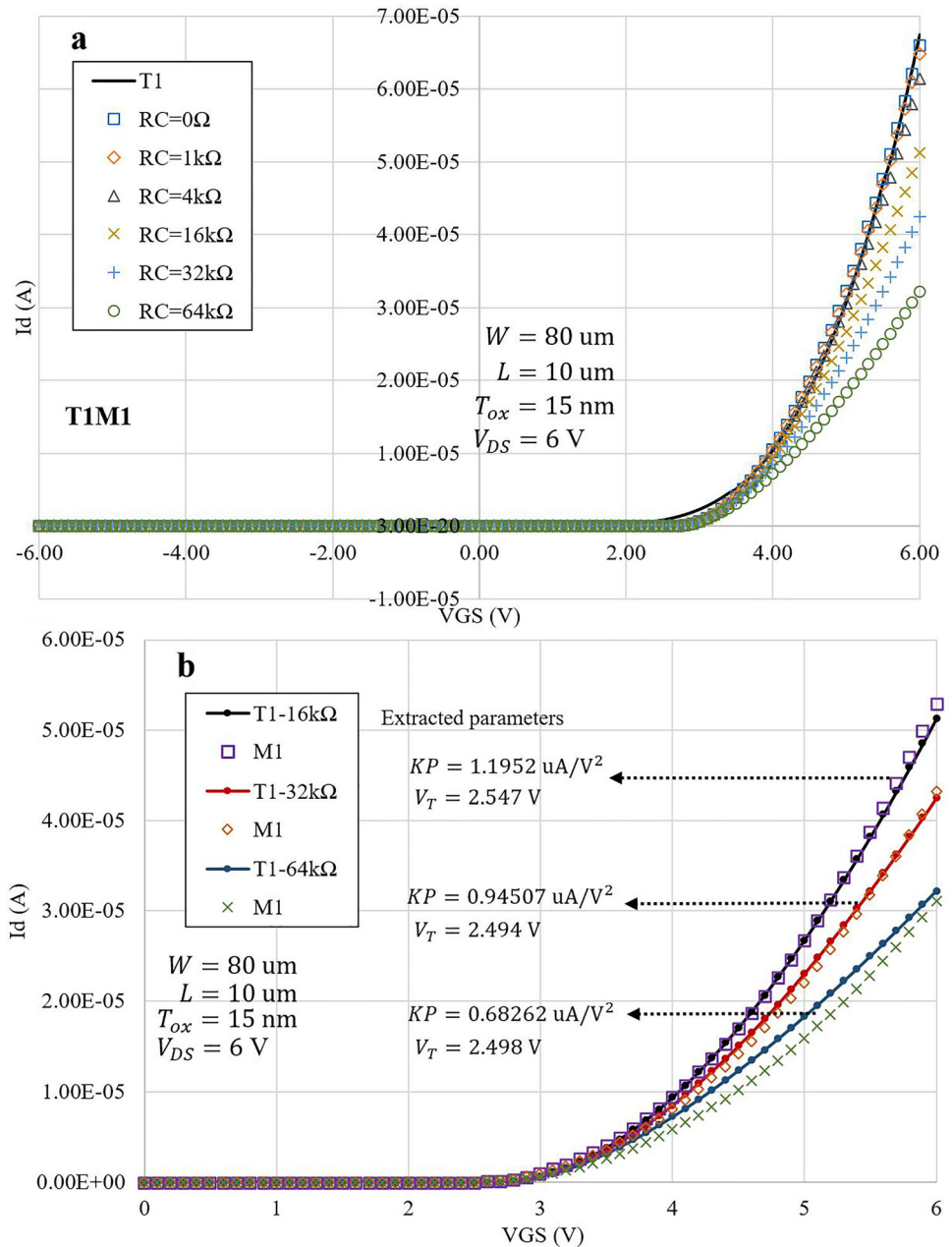
Figure 6b and c show the trend of the parameters V_T and KP , where it is observed that V_T decreases for

resistance values from $10 \times 10^3 \Omega$, remaining constant at approximately 2.5 V for resistance values higher than $32 \times 10^3 \Omega$. KP also appears to be inversely proportional to R_C in an almost linear trend, this behaviour is the same for mobility (Fig. 6d), these two parameters are directly affected by increasing resistance.

6.3 Results of experiment 3

With the dataset where the I_D current was normalised so as not to depend on the geometrical parameters (W and L), the first 3 NN models were trained (Table 1). This experiment was performed only in the saturation regime. Table 4 shows the evaluation

Fig. 5 Behaviour of T1 with R_C added and modelled with parameters extracted by NN model 1, **a** corresponds to the I-V curve and its variation with resistance, and **b** presents the fit of three curves with extracted parameters



achieved by the models with the validation measurements samples, where the 3 models reached $R^2 = 0.99$.

Figure 7a presents the T3 measurement and the curves modelled with the parameters extracted by three NN models trained with the normalised current (Eq. 24) and by the analytical method. Where model 1 obtained an error rate of 12.4%, model 2 obtained 16.49%, model 3 obtained 22.98% and AM obtained 9.86%. Figure 7b presents the modelling with the extracted parameters for T7, where model 1 obtained

an error rate of 11.79%, model 2 obtained 14.22%, model 3 obtained 17.62% and AM obtained 15.52%. The results show an increase in modelling error with the parameters extracted by the NN models that were trained with the normalised $I-V$ curves to remove the channel width and length. In this comparison, the AM has a smaller error rate compared to the NNs.

Figure 8 shows the average error rate (the 3 NN models) obtained for the 7 transistor sizes (Table 2), where an error between 12 and 19% is observed, with a peak of up to 25% for the $W = 80 \mu\text{m}$ and $L = 10 \mu\text{m}$

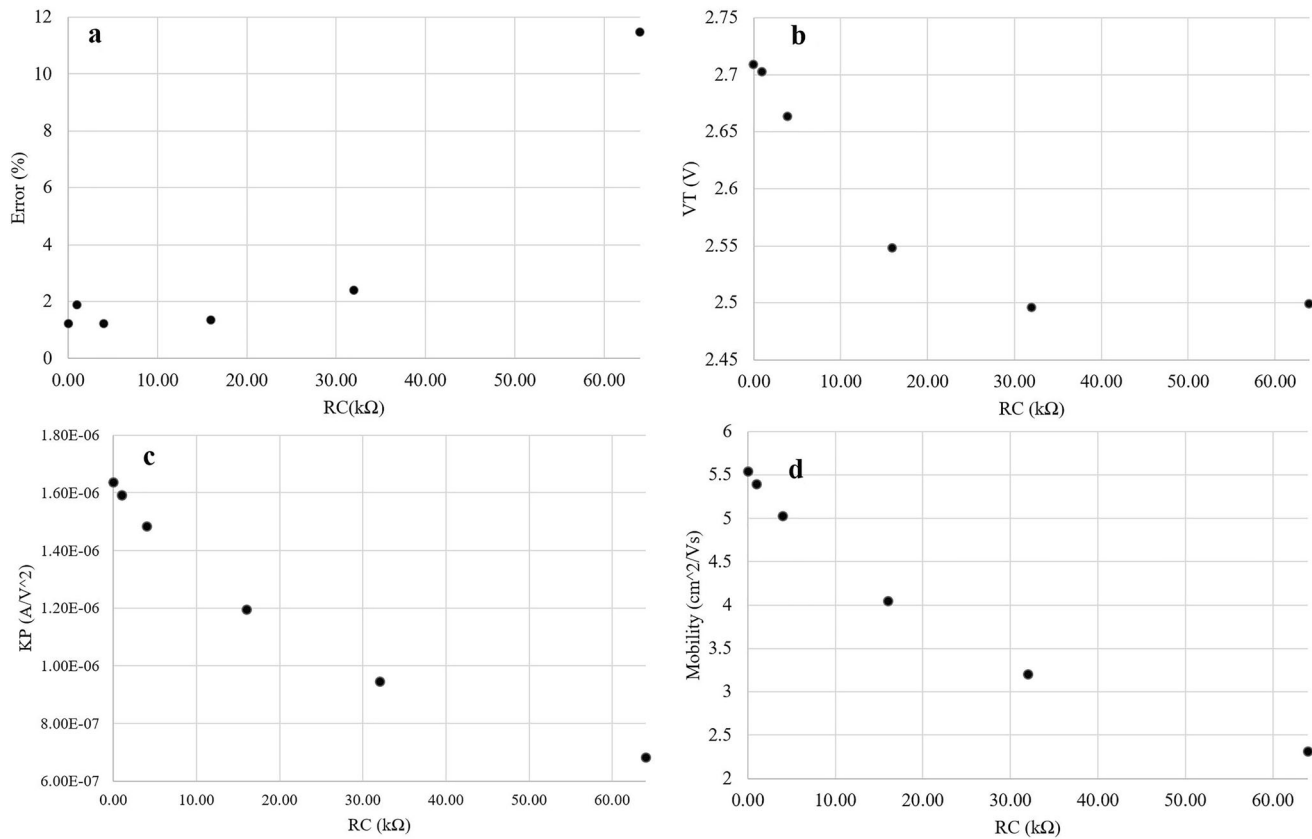


Fig. 6 Percentage error in parameter extraction (a), threshold voltage (b), transconductance (c) and mobility (d) as a function of R_C

Table 4 Evaluation of the NN models using normalised current

NN model	MSE	R^2
1	2.209×10^{-5}	0.996
2	2.258×10^{-5}	0.999
3	2.244×10^{-5}	0.998

transistor. The total average error of 16.89% could be acceptable considering that it was not necessary to have samples for each transistor and the extracted parameters are close to the desired one, these values could be adjusted by the expert personnel in charge of the extraction task, this manual adjustment is also often used in analytical methods when the result presents a notable error between the physical measurement and the one modelled with extracted parameters.

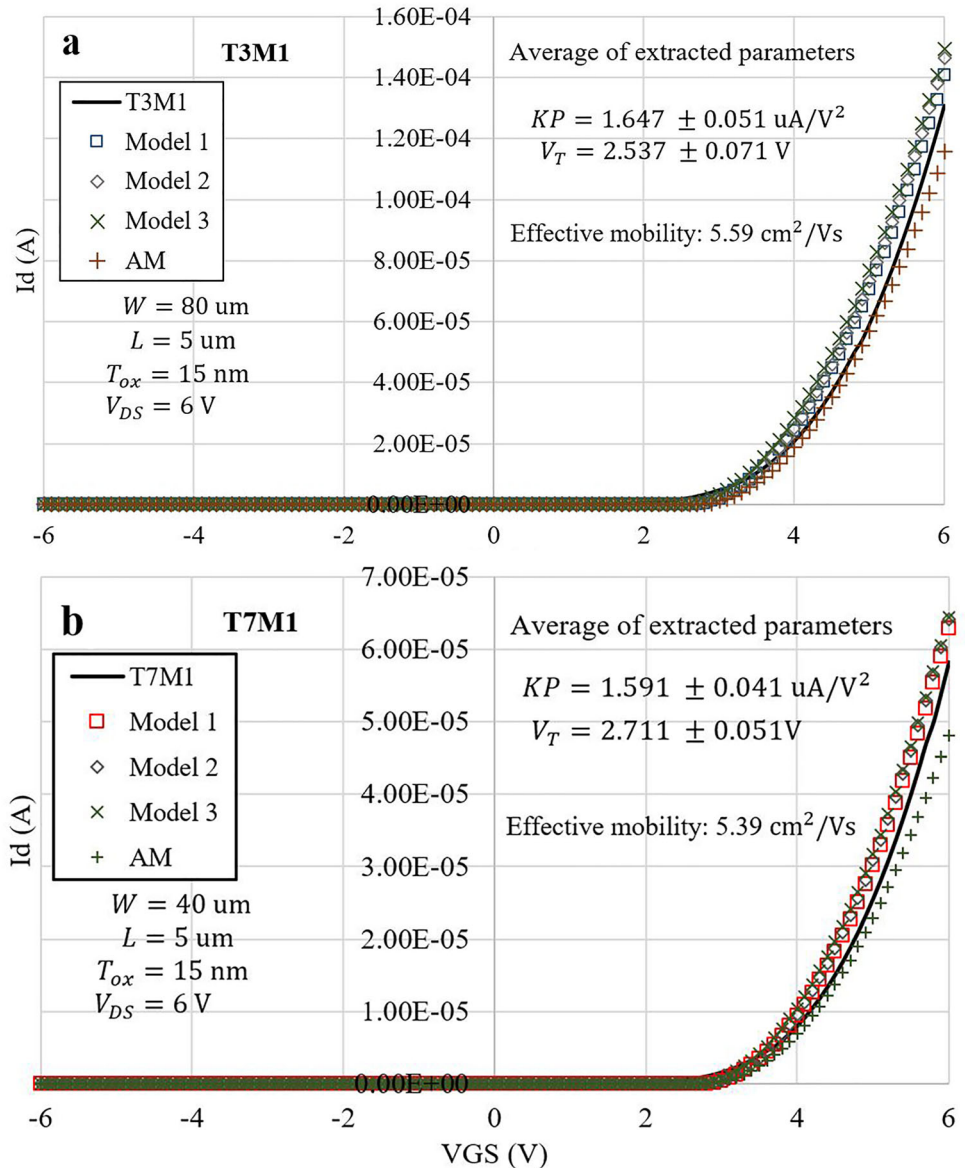
6.4 Results of experiment 4

This section presents the results of the extraction of KP , V_T and R_C , for which the training of NN models

was carried out using three different datasets. Figure 9a presents the average performance obtained by the eight trained NN models (Table 1) using the second dataset. This was achieved by increasing the current of each sample, dividing them by the maximum current of the set, which was $I_D = 2.856384 \times 10^{-4}$ A. For example, if the current value was $I_D = 6.8323 \times 10^{-5}$ A, when $V_{GS} = 6$ V, after division the value would be $I_D = 2.392 \times 10^{-1}$ A. This increase allows to have a greater distance between each sample and thus facilitate the identification between them and as a result, to have a better training of the NNs. After training, a performance of 99.4% for KP , 99.7% for V_T and 87.5% for R_C was obtained. Having the lowest learning in R_C in 87.5% due to the slight effect that this parameter has in comparison with KP and V_T .

Figure 9 presents the physical measurement of T1NP, together with the curves obtained using the extracted parameters. Where the four models with the lowest error percentages were selected, model 4 had 14.25% error, model 5 had 45.16% error, model 6 had 64.39% error and model 7 had 19.53% error.

Fig. 7 Modelling of two devices, T3 (a) and T7 (b) using parameters extracted with NN trained with normalised current



The lack of fit of the physical measurements with the parameters extracted by the NNs was due to low KP values. This happened because the focus was on the NNs predicting the parameters as accurately as possible during their training, falling into what is known as overtraining, which means that the NNs learn the input and output data “by heart”, but with new samples the results are erroneous.

To improve the results, training was performed with the third dataset, but for this training, the samples were not pre-processed using the maximum current value of the set, but only increased using $I_D = I_D \cdot 100$. For example, if in one sample the current when $V_{GS} = 6\text{ V}$ was $I_D = 6.5256 \times 10^{-5}\text{ A}$, after

multiplication, it would be $I_D = 6.5256 \times 10^{-3}\text{ A}$, this slight increase allows to improve the identification of KP and V_T , but not to the point of overtraining again. After training the eight NN models, an average performance of 94.6% was obtained for predicting the KP parameter, 99.1% for predicting V_T and only 21.3% for predicting R_C . The decrease in the identification of the R_C parameter is noticeable when the I_D of the samples is not increased, and the NNs are allowed to gain as much knowledge as they can from the data themselves.

Figure 10a presents the physical measurement of T1NP, and the curves obtained using the extracted parameters, again, with the intention of not

Fig. 8 Average extraction error rate for different transistors

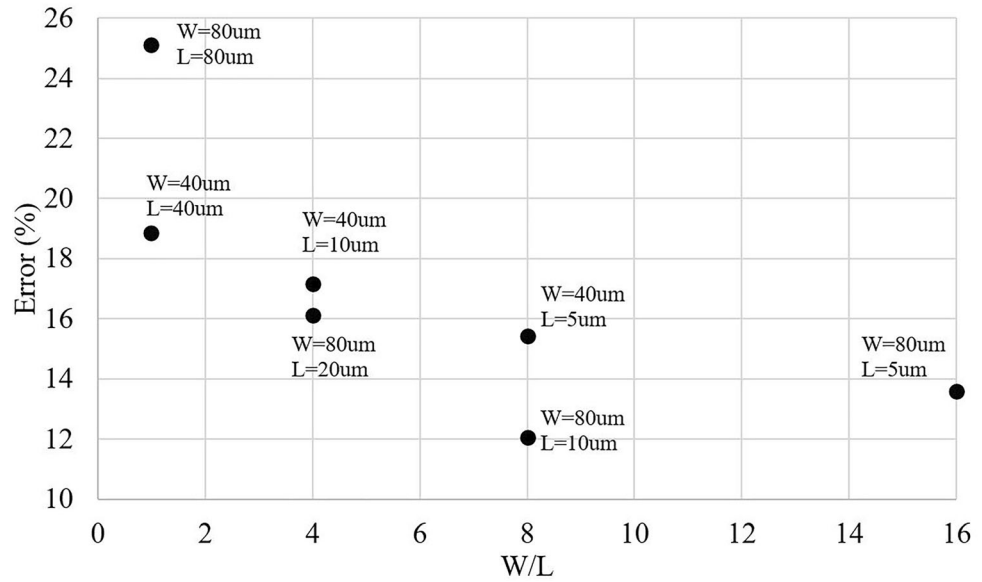
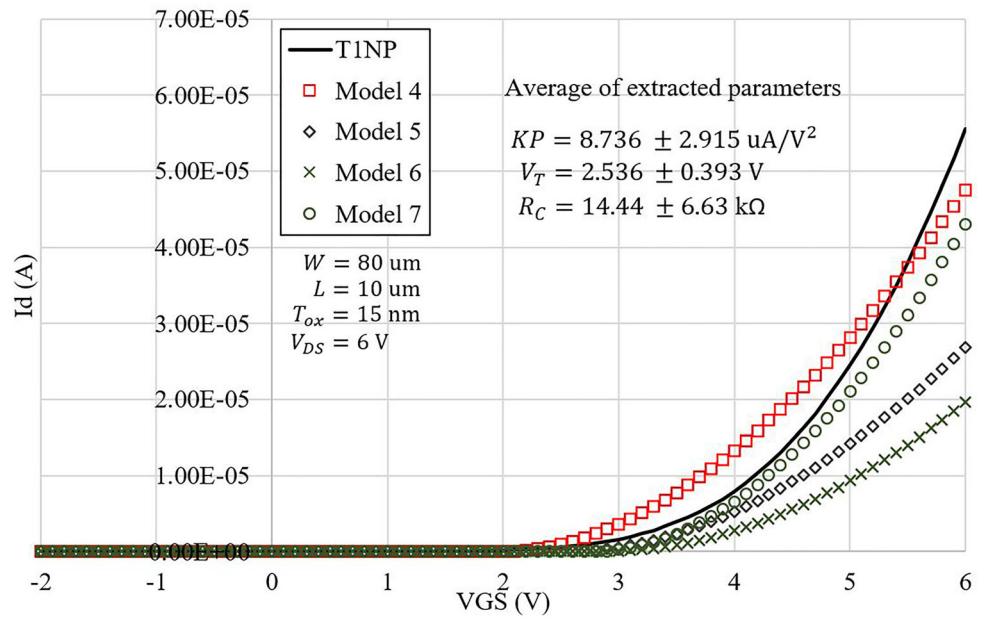


Fig. 9 Extraction and modelling of TINP with extracted parameters. The training of the NNs was done using the second dataset

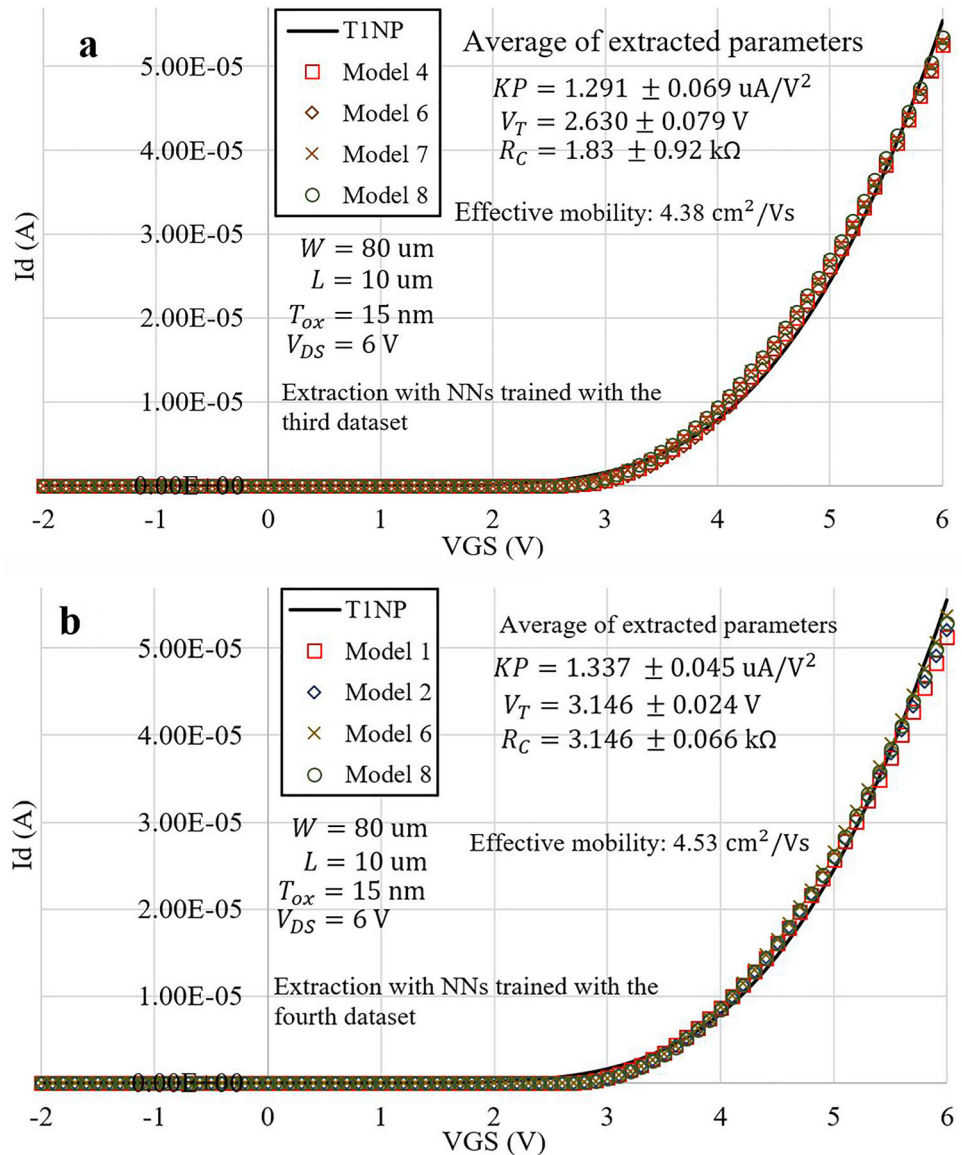


saturation of the graph, the 4 models with the lowest percentage error are presented. Where model 4 obtained an error percentage of 0.93%, model 6 obtained 1.69%, model 7 obtained 3.62% and model 8 obtained 4.62%. The decrease in error is noticeable when overtrained models are not obtained. Although, the trained NN models for the third set only learned 21.3% in the R_C parameter, they identified with high accuracy the parameters of the physical measurements.

The training of NN models using the union of the second and third sets aimed to test whether R_C

learning improves with more examples (840 samples). Using the same form of preprocessing as for the third set ($I_D = I_D \cdot 100$), a performance of 94.7% for KP , 99.2% for V_T and 12.03% in R_C was obtained. The learning of R_C did not improve by having more examples, on the contrary it decreased, this is because the union of the second and third set shortened the difference in the KP parameter, so that two samples, one with a $KP = 1.2 \times 10^{-4} \text{ A/V}^2$ and another with a $KP = 1.4 \times 10^{-4} \text{ A/V}^2$, both with the same value of R_C , have no great difference, considering that R_C affects the last region of V_{GS} . Figure 10b shows the

Fig. 10 Extraction and modelling of T1NP-PN with extracted parameters. Using the third and fourth dataset during training, where **a** corresponds to the supplied measurement from negative to positive and **b** from positive to negative



physical measurement of T1NP and the curves obtained using the extracted parameters, where again the four with the lowest error are shown. Where model 1 obtained a percentage error of 0.92%, model 2 obtained 0.06%, model 6 obtained 2.42% and model 8 obtained 0.51%. It is possible to observe that having a greater number of examples did improve the extraction of KP and V_T , which was reflected in the fact that more models obtained an error of less than 1%.

7 Conclusion

In this research, Artificial Neural Networks are proposed for the parameter extraction process in Thin Film Transistors IGZO with Al contacts. The parameters extracted were KP , V_T and R_C , as they are parameters that allow modelling the behaviour of the TFTs. Four different experiments were carried out to prove that given the correct dataset of $I-V$ sample curves, the Neural Networks are capable to identify and extract their parameters. In this work, errors were comparable to and in some cases smaller than in [21, 22].

In the first experiment, different models of NNs were trained to extract the parameters KP and V_T ,

where the average performance in the validation measurements reached an $R^2 = 0.998$, and in physical measurements with minimum error rates of 1.68–2.04% (average of the best 4 models) because the data sets designed for this experiment were in the perfect range in the KP and V_T sweeps. The result of the NNs extraction showed higher accuracy compared to the analytical method applied. In this experiment, model 7 (4 layers of 256, 128, 64 and 32 neurons) obtained the lowest error rates in 3 of the 4 extractions presented.

In the second experiment, a trained NN model was fed with a measurement of T_{IM1} to which, by means of the simulator, different values of R_C were added. It was found that the trained NNs are able to compensate the KP and V_T values to fit measurements to which a contact resistance was added from 0 to $64 \times 10^3 \Omega$, with a percentage of error of about 2% when $R_C < 64 \times 10^3 \Omega$. This allows the extraction of parameters from I – V curves even with medium–high resistances.

In the third experiment, the geometrical factor was removed from the first dataset, multiplying the I_D current by L/W , in order to use the first dataset to extract the parameters of physical measurements of different sized transistors, which were also multiplied by their own L/W . In this experiment only 3 different models were trained, which obtained an average R^2 of 0.997 in the validation samples, and from the 7 available dimensions an average error rate of 16.89% was obtained. With the increased error in this experiment, the extraction result with the analytical method was more accurate with an error of about 9%. This experiment was proposed as a simple way to overcome the need for samples to extract parameters in transistors with different channel widths and lengths. In this experiment, model 1 (2 layers of 128 and 64 neurons) was the one with the lowest extraction error rate.

In the fourth experiment, the extraction of the parameters KP , V_T and R_C was performed, for which training was done using the second, third and a union of these datasets. In the extraction using the second dataset, an example was presented of how NNs can learn to identify the parameters of validation samples with average R^2 of 95.53% but fail in extraction tests against physical measurements due to overtraining. On the other hand, by using the third

dataset, overtraining was solved, avoiding excessive preprocessing. In this way, an average R^2 of 71.6% was obtained, although it could be considered as low, these trained NN models obtained minimum error percentages of up to 0.93% in the extraction of physical measurement. Finally, training was performed with a dataset consisting of the second and third data sets, from which an average R^2 of 68.6% was obtained, again, a low overall performance, but with these trained NN models, error rates of up to 0.06% were obtained in extraction tests on physical measurements. In the extraction of KP , V_T and R_C in this experiment, model 8 (5 layers of 256, 128, 64, 32 and 16 neurons) had one of the lowest errors in the extractions presented, using both the third and fourth training set.

Acknowledgements

Nanoscience, Micro and Nanotechnologies Centre of the National Polytechnic Institute is thanked for the fabrication of devices whose transferential curves were used to test the methodology proposed in this research. Thanks, are also due to the National Council of Science and Technology for the scholarship for advanced studies.

Author contributions

RCV performed the parameter extraction using NNs and drafted the manuscript, NHC manufactured the transistors with which the method proposed in this work was tested, RZG performed the parameter extraction using the analytical method, FGL and ALC analysed the proposed method and improved the form and wording of the manuscript.

Funding

The authors have not disclosed any funding.

Data availability

The datasets generated during the current study are available from corresponding author.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. S. Raczynski, *Modeling and simulation*, 1st edn. (Wiley, Hoboken, 2014), pp.1–14
2. H. Khan, M. A. Bazaz, S. A. Nahvi, Simulation Acceleration of High-Fidelity Nonlinear Power Electronic Circuits Using Model Order Reduction, in: 5th IFAC Conference on Advances in Control and Optimization of Dynamical Systems (2018) <https://doi.org/10.1016/j.ifacol.2018.05.069>
3. P. Moreno, R. Picos, M. Roca, E. Garcia, B. Iniguez, M. Estrada, Parameter extraction method using genetic algorithms for an improved OTFT compact model. *Spanish Conf. Electron Devices* (2007). <https://doi.org/10.1109/SCED.2007.383996>
4. C. Tanaka, K. Ikeda, Comprehensive investigation on parameter extraction methodology for short channel amorphous-InGaZnO thin-film transistors, in: 2018 IEEE International Conference on Microelectronic Test Structures (IEEE, 2018) doi: <https://doi.org/10.1109/ICMTS.2018.8383756>
5. A. Ortiz-Conde, F.J. García-Sánchez, J. Muci, A. Terán Barrios, J.J. Liou, C.-S. Ho, *Microelectron. Reliab.* (2013). <https://doi.org/10.1016/j.microrel.2012.09.015>
6. A. Cerdeira, M. Estrada, R. García, A. Ortiz, F.J. García, *Solid State Electron* (2001). [https://doi.org/10.1016/S0038-1101\(01\)00143-5](https://doi.org/10.1016/S0038-1101(01)00143-5)
7. S.K. Ojha, B. Kumar, *SILICON* (2022). <https://doi.org/10.1007/s12633-021-01149-6>
8. C. Avila, A. Ortiz, J.A. Caraveo, M.A. Quevedo, *Trans. Electr. Electron. Mater.* **22**(4), 550–556 (2021). <https://doi.org/10.1007/s42341-020-00268-y>
9. P. Mittal, Y.S. Negi, R.K. Singh, *J. Comput. Electron* (2015). <https://doi.org/10.1007/s10825-015-0719-8>
10. K. Bhargava, V. Singh, *J. Comput. Electron* (2014). <https://doi.org/10.1007/s10825-014-0574-z>
11. S. Xin-zhi, L. Hai-wen, S. Xiao-wei, C. Yan-feng, C. Zhi-qun, L. Zheng-fan, *Wuhan Univ. J. Nat. Sci.* (2005). <https://doi.org/10.1007/BF02830676>
12. S. Nautiyal, P. Mittal, Contact resistance in organic transistors: Extraction using variable length method, in: 2017 International Conference on Computing, Communication and Automation, (IEEE, 2017) doi: <https://doi.org/10.1109/CCA.A.2017.8230051>
13. M.E. Rivas-Aguilar et al., *Curr. Appl. Phys.* (2018). <https://doi.org/10.1016/j.cap.2018.04.002>
14. A. Pacheco-Sanchez, M. Claus, S. Mothes, M. Schröter, *Solid State Electron.* (2016). <https://doi.org/10.1016/j.sse.2016.07.011>
15. H. Bae et al., *IEEE Electron Device Lett.* (2016). <https://doi.org/10.1109/LED.2015.2509473>
16. N. Akkan, M. Altun, H. Sedef, Parameter Extraction Method Using Hybrid Artificial Bee Colony Algorithm for an OFET Compact Model, in: 2018 15th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (IEEE, 2018) doi: <https://doi.org/10.1109/SMACD.2018.8434861>
17. S. Moparathi, P. K. Tiwari, G. K. Saramakala, Genetic algorithm-based threshold voltage prediction of SOI JLT using multi-variable nonlinear regression, in: 2021 International Symposium on Devices, Circuits and Systems (IEEE, 2021) doi: <https://doi.org/10.1109/ISDCSS2006.2021.9397911>
18. S. I. Sayed, M. M. Abutaleb, Z. B. Nossair, Improved CNFET performance based on genetic algorithm parameters optimization, in: 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEEE, 2017) doi: <https://doi.org/10.1109/IEMCON.2017.8117185>
19. I. Benacer, Z. Dibi, *Int. J. Autom. Comput.* (2016). <https://doi.org/10.1007/s11633-015-0918-6>
20. R. Picos et al., *Solid. State. Electron.* (2007). <https://doi.org/10.1016/j.sse.2007.02.031>
21. J. Oh et al., *ECS J. Solid State Sci.* (2022). <https://doi.org/10.1149/2162-8777/ac6894>
22. Q. Yao et al., *Micromachines* (2021). <https://doi.org/10.3390/mi13010004>
23. Berkeley, Aim-Spice, (Software, 2022), <http://www.aimspsice.com/>. Accessed 08 August 2022
24. R.J. Baker, *CMOS*, 3rd edn. (Wiley-IEEE Press, Hoboken, 2010), pp.131–145
25. M. Flasiński, *Introduction to artificial intelligence*, 1st edn. (Springer, Cham, 2016), pp.156–157. <https://doi.org/10.1007/978-3-319-40022-8>
26. P. Isasi, I. Galván, *Redes neuronales artificiales* (Pearson Prentice Hall, Madrid, 2004), pp.1–60
27. M. Hagan, H. Demuth, M. Beale, O. De Jesus, *Neural network design*, 2nd edn. (Ebook, Nedlands, 2014), p.2.2-2.10
28. W. Ertel, *Introduction to artificial intelligence*, 2nd edn. (Springer, Cham, 2017), pp.245–260. <https://doi.org/10.1007/978-3-319-58487-4>
29. Analog Devices Inc., LTspice, (Software, 2022), <https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html>. Accessed 09 August 2022
30. ScikitLearn, sklearn.model_selection.GridSearchCV, (Website, 2022), https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV. Accessed 09 August 2022

31. Google, Colaboratory, (Website, 2022), <https://research.google.com/colaboratory/intl/es/faq.html>. Accessed 09 August 2022
32. M. Abadabi et al., Tensorflow, (Website, 2022), <https://www.tensorflow.org/?hl=es-419> Accessed 09 August 2022

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.