Choice of Parameters of an LSTM Network, based on a Small Experimental Dataset

Océane-Sophie Bérot, Hélène Canot, Philippe Durand, Bouchra Hassoune-Rhabbour, Hervé Acheritobehere, Caroline Laforet, and Valérie Nassiet

***Abstract*—This paper highlights the impact of specific structural parameters of an LSTM network on the relevance of the prediction results obtained based on the temporal physical data of a polymer material subjected to aging. Several parameters are compared in this study, such as the type of interpolation functions, the number of hidden layers, the number of units per hidden layer, and the activation function used within the LSTM network. The dataset is based on experimental data derived from gravimetric measurements of an epoxy adhesive subjected to hygrothermal aging for three different temperatures (50°C, 70°C, and 90°C) and relative humidity by 95%. It is possible to use interpolation functions to fill the missing data specific to this type of experimental data. Using a single hidden layer of 150 units with a hyperbolic tangent type activation function leads to the best results. The results are less sensitive to the change of interpolation functions. This work first showed the relevance of using LSTM to predict the evolution of an irregular physical parameter over time, such as gravimetry. This article also highlights that the choice-specific parameters will significantly influence the results and the stability of the model.**

***Index Terms*—Artificial Neural Network, LSTM parameters, small dataset, prediction, adhesive.**

I. INTRODUCTION

ARTIFICIAL neural networks (ANN) are increasingly used in materials to meet the needs of extensive calculations and deepen knowledge of engineering materials, processes, and structures [1], [2]. Integrating many parameters without determining the exact physical law or the mathematical model associated with a physicochemical mechanism is possible. Indeed, the prediction method based on neural networks does not need to rely on models (that for some processes are quite complex) and has the advantage of solving nonlinear problems [3], which facilitates modeling or prediction issues, for example. Furthermore, a network can be used for different applications because it demonstrates a certain "universality." The structure of the neural networks and the notation used are the same regardless of the field of use of this tool.

Different approaches of ANN have proven to be suitable for predicting the behavior of different materials. For instance, authors in [4] have used active learning to predict the strength of epoxy from a small dataset. Moreover, neural networks have been used to predict the fatigue life of natural rubber composites [5]. At the same time, the prediction of Young's modulus for silicate glasses has been made with sparse datasets using Gaussian process regression for ANN [6]. Recently, the determination of aging conditions by analysis of infrared spectra using ANN has been done in [7].

Different types of ANNs exist and are used in materials science, such as MLP (for MultiLayer Perceptron) networks. A multilayer perceptron is a type of neural network most used in supervised and unsupervised learning, consisting of several layers of interconnected neurons. It is used for classification and regression problems [8]. For example, Zhu *et al.* [9] rely on a small dataset to predict the lifetime of an epoxy resin. Ding *et al.* [10] rely on neural networks to predict the macroscopic mechanical properties (such as Young's modulus) and the appearance of microscopic cracks in polymer composites reinforced with unidirectional fibers.

Other authors rely on convolutional neural networks (CNN) for image recognition. Indeed, a convolutional neural network (CNN) is a neural network designed to process images or signals. It is, therefore, used for problems of image classification and object detection. It relies on convolution filters to extract features from input data and pooling layers to reduce its dimensionality [8]. For example, Hsu *et al.* [11] use modeling of crack propagation within a material that a CNN will analyze and then transmit to a recurrent network (RNN) to predict the evolution of the crack. MLPs and CNNs are potent tools to make configurational predictions by the association of parameters or classifications of data.

Previous studies show consistency between experimental values and values calculated by neural networks [7], [9]. Predictive modeling by neural network seems suitable for predicting a parameter related to the aging of a material as faithfully as possible.

RNNs predict the evolution of parameters over time in a wide variety of fields, such as soil geotechnics [12], voice recognition [13], spread of COVID-19 disease worldwide [14], or prediction of the autonomy of lithium-ion batteries in electric cars [15]. In the case of materials, they are increasingly used to predict, for example, the glass transition temperature of polymers [16], the laser cutting

Manuscript received June 06, 2023; revised October 25, 2023.

OS. Bérot is PhD student in Laboratoire Génie de Production at ENIT, 47 avenue d'Azereix 65000 Tarbes, France, (e-mail:oceanesophie.berot@enit.fr).

H. Canot is associate researcher in Department of Mathematics of Bretagne Sud, rue André Lwoff 56017 Vannes, France, (e-mail:helene.canot@univ-ubs.fr).

Ph. Durand is assistant professor in Department of Mathematics (modélisation mathématique et numérique), Conservatoire National des Arts et Métiers, 292 rue Saint Martin, 75141 Paris, France, (e-mail:philippe.durand@lecnam.net).

B. Hassoune-Rhabbour is assistant professor in Laboratoire Génie de Production at ENIT, 47 avenue d'Azereix, 65000 Tarbes, France, (e-mail: bouchra.hassoune@enit.fr).

H Acheritobehere is engineer in Roxel, rue Gay Lussac, 33160 Saint-Médard-en-Jalles, France (e-mail: H.ACHERITOBEHERE@roxelgroup.com).

C. Laforet is engineer in Roxel, 1 route de l'Aérospatiale, 18570 Le Subdray, France, (e-mail: C.LAFORET@roxelgroup.com).

V. Nassiet is full professor in Laboratoire Génie de Production at ENIT, 47 avenue d'Azereix, 65000 Tarbes, France, (e-mail: valerie.nassiet@enit.fr).

parameters of basalt fibers [17], or even tear propagation on textiles, intended for spatial purposes [18].

Based on these state-of-the-art results, RNNs are the most suitable for predicting the evolution of a parameter over time, in other words, the evolution of a time series. Among the category of RNNs, LSTMs (for Long Short Term Memory) are suitable tools for time series. LSTMs were proposed in 1997 by Hochreiter and Schmidhuber [19] to learn dependencies and retain information over the long term. This network, therefore, has a larger memory capacity and additional stability, avoiding the problems of gradient explosion [20].

A major constraint in materials science is that, in general, experimental datasets are limited and expensive to build. Datasets present in the literature can be either scattered or noisy and inconsistent. For these reasons, it is challenging to obtain a consistent and complete source of information on the properties of polymeric materials. It can, therefore, be challenging to use AI tools on materials science applications as the lack of data can lead to over-fitting issues and, consequently, to a loss of reliability regarding the predictions made. Over-fitting appears when the network learns the training dataset "too" well and cannot adapt to new data. In addition, other essential issues are applicability and feasibility in LSTMs with small datasets. Indeed, the LSTM, by its structure and operation, requires that the data respect the periodicity principle, meaning that the time step between two samples does not change. Therefore, this constraint will inevitably lead to missing data problems since it is impossible to carry out tests with the same time difference of the order of an hour. Thus, assigning missing data is the priority step of data preprocessing.

However, there are different ways to circumvent this experimental problem to obtain sufficiently reliable results. Authors in [4] have used, for example, an active learning "pipeline" (otherwise called "optimal experimental design"). Zhu *et al.* rely on physical material degradation models to increase the data [9]. To analyze crack propagation within a material, Hsu *et al.* use numerical crack simulations to feed their LSTM network [11]. Thus, to circumvent this missing data problem, relying on physical laws [9], numerical simulation tools [11], or modules directly integrated within the network is possible [4]. Our study is based on another method to limit the problems related to a restricted dataset: interpolation functions, which are mathematical tools that recreate all the data between two experimental points.

Another problem related to the neural network, in general, appears in the choice of network parameters. Indeed, in many articles, the choice of parameters is not discussed in depth because there is no absolute rule to determine the network parameters. Often, the programming experience makes it possible to choose one parameter or another. This study compares the impact of different parameters to justify the choice of specific parameters (like time step, number of layers, or number of nodes) and to determine, to a certain extent, the best combination for the dataset transmitted to the LSTM.

To test the prediction abilities of the LSTM on exclusively experimental data, we used a dataset based on mass intakes of epoxy-type adhesive samples subjected to hygrothermal aging at 50°C, 70°C, and 90°C with a high relative humidity of 95% for each temperature.

In this article, the first part will explain the dataset used and the state of the art about the aging phenomena of the adhesive. A second part will present the interpolation functions compared, and a third part will focus on the parameters of the LSTM at different structural levels. Finally, the last part will present the results and the associated discussion.

II. EXPERIMENTAL DATASET

The dataset relates to a two-component adhesive from the mixing, in stoichiometric proportions, of an epoxy resin DGEBA (Diglycidyl ether of bisphenol A) and a polyamidoamine hardener. The formulation contains 40% w/w fillers (mainly kaolin) [21], [22]. Epoxy resins are widely used as an adhesive for bonded assemblies thanks to their excellent ability to adhere to substrates, low shrinkage after polymerization, and good mechanical and fatigue resistance. The adhesive studied here is flexible at room temperature because its glassy transition (Tg) is 31,8 $\pm$ 0,4°C, allowing good impact resistance. If the adhesive is subjected to accelerated and relatively violent aging, it can see a visible and measurable change in its water uptake. Therefore, there is an apparent evolution of the physical parameter, resulting in a more conducive dataset for use in a neural network. Samples were weighed using a 0.1 mg resolution Mettler AE 200 electronic balance. Mass uptakes are carried out on four rectangular samples of 10 mm x 10 mm x 2 mm for each wet aging temperature to ensure the reproducibility of the results.

When the epoxy adhesive is exposed to an aggressive environment, such as high humidity and aging temperatures, changes in its mechanical, physical, and chemical properties may occur [23], [24], [25]. The water will first act at the adhesive/external environment interface level. It will then be able to diffuse towards the core of the material that will be more or less hydrophobic [26], depending on the aging conditions (due to chemical and physical evolutions of the polymer network).

This water diffusion will lead to short-term changes, such as lattice plasticizing [27], [28], irreversible long-term changes, such as swelling (which can lead to the appearance of cracks) [27], [29] or hydrolysis (which can trigger leaching phenomena) [30].

Figure 1 shows the sorption curves obtained by gravimetry in immersion at 50°C, 70°C and 90°C in 95° HR. Kaolin particles being impermeable, the adhesive mass variations in percentage are reduced to the matrix mass fraction. The square root of time in seconds is divided by the thickness in millimeters to overcome thickness shifts.

As shown in Figure 1, from the beginning of hygrothermal aging, the mass of the adhesive samples increases significantly until reaching a stabilization phase that depends on the aging temperature. Indeed, for aging at 50°C, the water uptake of the material is relatively stable, around 4%, and

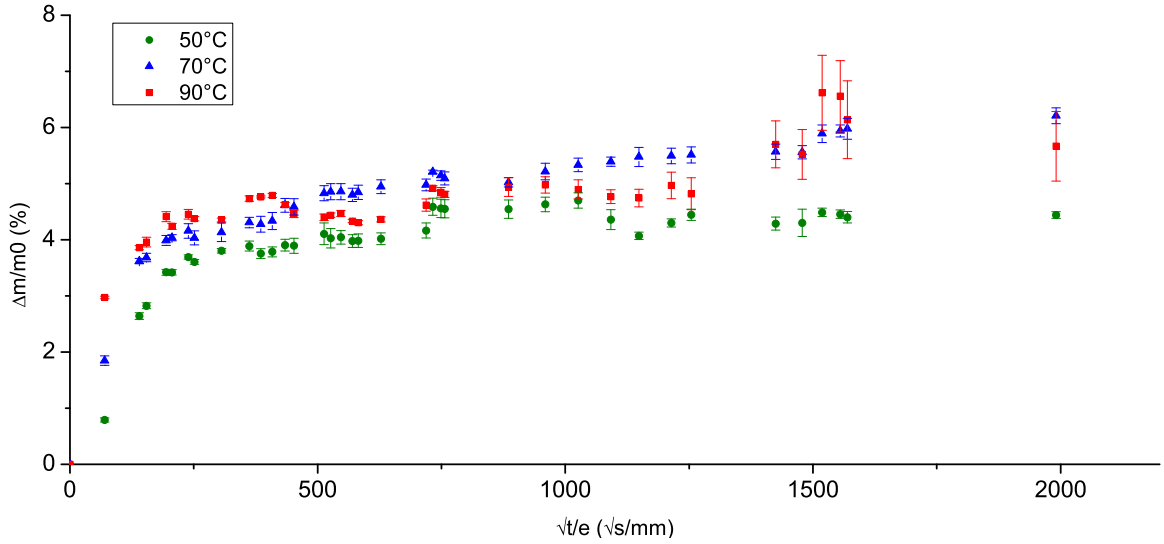


Fig. 1: Water sorption curves in 95° HR at 50°C, 70°C and 90°C reduced to the matrix mass fraction.

seems to reach saturation. For 70°C, after a first fast water intake, a slow increase appears from 250 $\sqrt{s}/mm$ without stabilization. For 90°C, the water sorption curve is more fragmented, and a sudden increase in the mass of the samples appears towards the end of aging. Uncertainties are also more significant for this curve area than 50°C and 70°C aging. The three curves show different changes in water gain, which correspond to the following phenomena [31], [32]:

- At 50°C, the diffusion kinetics of the adhesive is close to a Fickian behavior despite the noisy mass variation on the plateau. We can assume that, preferably, water diffusion occurs at the level of the "free volume" of the material without interacting directly with the polymer [31], [33].
- At 70°C, the progress in water absorption in the second phase may be due to a change in hydrophily and possible chemical interaction between polymer and water.
- At 90°C, the loss of mass observed from 1520 $\sqrt{s}/mm$ of aging is characteristic of a chemical interaction between water and the polymer, i.e., hydrolysis, which will lead to oligomer leaching and filler exudation [30]. This last phenomenon can also be seen through the significant increase in measurement uncertainties. We can explain it because each sample's exudation amount of filler or chain leakage may differ.

Experimental data was collected to train the LSTM network and build the model. Thirty-eight measurements were taken for each temperature during a hygrothermal aging period of 203 days. The data was collected regularly, except for a gap between the penultimate and last data, to provide an ideal dataset for use in an LSTM. However, the discontinuity at the end of the dataset can be seen as a disadvantage that could affect the network's results. Nevertheless, it was essential to use a realistic experimental setup to demonstrate the network's adaptability in the face of an imperfect dataset. Experimental tests are prone to many unforeseeable factors, especially during an aging campaign.

III. MISSING DATA PROBLEM

The treatment of missing data during the time series analysis is crucial because of the temporal periodicity constraint of the LSTM. In order to overcome this problem, we use the interpolation functions in this study. It allows us to fit a function to our experimental points and uses this function to interpolate the missing data. It makes it possible to go from a dataset composed of 38 experimental samples to a complete dataset of 814 samples with a constant 6-hour time step. The choice of experimental data is essential because it will partly condition the results of the interpolation functions and the prediction results of the LSTM network. Using the interpolation function makes it possible to reduce the time step between samples when it is impossible to do it experimentally. In this way, the total number of samples can be increased, as well as reconstructing some possible missing data.

After testing different interpolation functions (such as the polynomial of order 2, the cubic spline [34] or the akima [35] functions), two of them were chosen to be compared in this study: the "pchip" function (for Piecewise Cubic Hermite Interpolating Polynomial) and the piecewise polynomial function (see Figure 2). Indeed, these two functions make it possible to avoid the Runge phenomenon, which consists of a divergence in the form of undulation between the experimental points and the interpolation function.

Hermite's interpolation consists of passing through the experimental points and ensuring that the slope of the curve at each point, in other words, the derivative, is an imposed value. In this study, the pchip function is used. It is built on the linear combination of four unit polynomials of degree three. In the case of an experimental evolution that can be noisy or fragmented, the pchip function seems well-suited [36]. It makes it possible not to exceed the given experimental points because the interpolation is monotonous and imposes the continuity of the first derivatives.

The piecewise polynomial method divides the dataset points into several intervals, determining each segment's

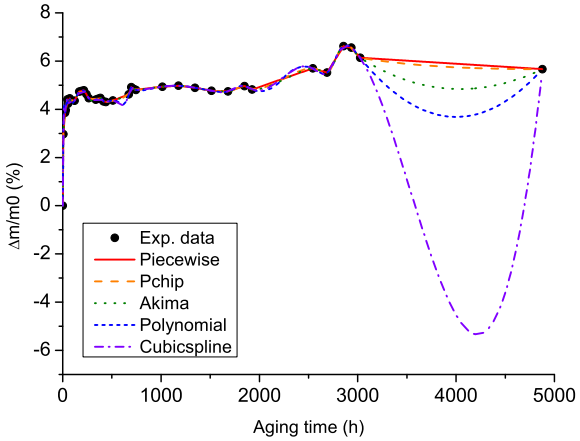


Fig. 2: Comparison of interpolation functions for water absorption at 90°C.

most appropriate polynomial function. This function, therefore, makes it possible to model more complex datasets with sudden or discontinuous evolution. However, if the size of the intervals is not regular, it can eventually lead to over-fitting.

IV. LSTM PARAMETERS

An LSTM layer consists of a set of recursively connected memory blocks. Each block comprises one or more memory cells and three multiplication units [13]. An LSTM cell is a succession of activation, product, and addition functions that filter the data transmitted to the network. The goal is to keep only the valuable data for prediction. An LSTM cell uses three distinct gates, similar to valves transmitting more or less information, and can write, read information, or reinitialize the cells (Figure 3).

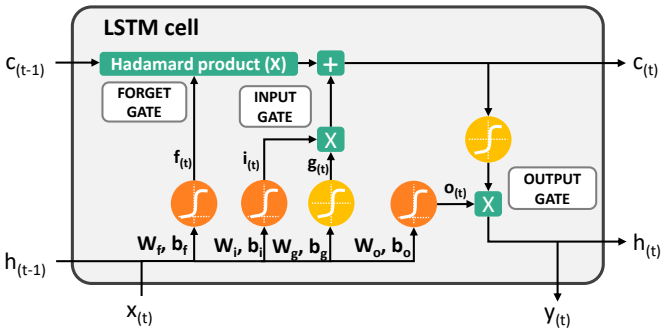


Fig. 3: Structure of an LSTM network (with x(t) as the input vector, h(t) the hidden state vector (short memory), c(t) the cell state vector (extended memory), y(t) the output vector, W(k) the weight matrix and b(k) the biases).

Weight matrices and biases are defined during the learning process. A bias is an additional parameter associated with each neuron, making it possible to determine the sensitivity of a neuron to its input. This parameter is taken into account in the same way as the interconnection weights for each input value and will, therefore, influence the behavior of the activation function [37].

LSTM cell is defined by nonlinear equations that integrate current and previous time inputs, cell outputs, and model parameters :

- Input gate: it allows information to be filtered and decides whether the latter should modify the content of the memory cell.

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \tag{1}$$

- Forget gate: based on the Hadamard product, it makes it possible to recover the long-term memory c(t-1) and to be able to forget or not certain information from the cell state c(t -1).

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \tag{2}$$

- Output gate: it allows us to define if the contents of the memory cell must influence the output of the cell and the hidden state h(t).

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \tag{3}$$

Once these gates are defined, the cell state (memory) and the candidate state, which represents candidate values for cell state at time (t), are computed :

- Cell state

$$c_t = i_t * g_t + f_t * c_{t-1} \tag{4}$$

- Candidate state

$$g_t = tanh(W_g.[h_{t-1}, x_t] + b_g) \tag{5}$$

- Output

$$y_t = h_t = o_t * tanh(c_t) \tag{6}$$

With σ the sigmoid function, W_i, W_f, W_o, W_g the weights of the corresponding matrices for the input, forget, and output gates, and the candidate state, while b_i, b_f, b_o, b_g are the corresponding biases for the input, forget and output gates, and the candidate state.

Before presenting the parameters of the LSTM network, it is essential to explain the preparation of the dataset so that they can feed the LSTM.

A. Preparation of the dataset

The first step is to define the time step between each data in the dataset. Once we have figured out the best time interval for our research based on our experimental data, preparing the data before sending it to the network becomes crucial. Standardizing and dividing the data into smaller groups are essential steps in this preparation.

The dataset consists of a gravimetric follow-up over time for each hygrothermal aging temperature of the adhesive. Various tests were carried out to determine which was the optimal time step for this study, i.e., the one that makes it possible to achieve reliable prediction results without weighing down the calculations. Generally, to evaluate the performance of the model, several performance indicators can represent the difference between reality and prediction [7], [9], [11], [38]. In our study, we chose to use MSE because it makes it possible to consider the differences between reality and prediction more critically and thus to "sanction" the most significant errors in predictions. The following equation gives the MSE expression:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{7}$$

With y_i the predicted value and $\hat{y}_i$ the observed value.
During the learning process, we will follow the evolution of several MSEs, such as the Train MSE (for learning) and the Valid MSE (for validation). For these two parameters, we will rely not only on the final value at the end of the epochs but on an average of the last ten values. Indeed, this makes it possible to consider variations that may appear during learning and penalize models with instability during learning. The Test MSE parameter is used to evaluate the prediction reliability of the model by giving an average value of the MSEs calculated at each point between the value predicted by the network and that of the test dataset.

Time steps of 48h, 24h, 12h, 6h, and 3h were tested with the dataset at 70°c (Figure 4). The neural network parameters were fixed following preliminary results to test these different time step values and see their influence on the prediction results. The values of the performance indicators (Train MSE, Valid MSE, and Test MSE) are low for the different configurations. Qu *et al.*'s work [39] also showed that the lowest time step did not necessarily deliver the best results. Thanks to the results, the 6h time step seemed the most suitable and was therefore chosen for the rest of this study.

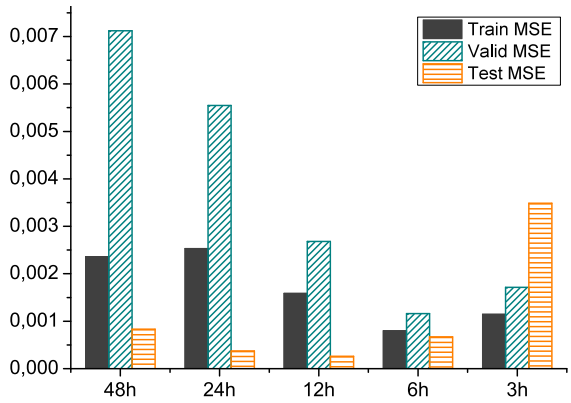


Fig. 4: Comparison of Train MSE, Valid MSE, and Test MSE performance indicators for 48h, 24h, 12h, 6h and 3h time steps.

After adjusting the initial dataset to obtain periodic data, it is necessary to scale the data to lighten the matrix computations carried out by the neural network. Scaling helps avoid premature saturation of hidden layer nodes by avoiding overlapping different number scales [40]. Normalization and standardization are the two most common methods for scaling data. For normalization and standardization, we rely on, respectively, the min and max values of the dataset and the mean and the standard deviation of the data. However, normalization disturbed the convergence of the results, which is visible thanks to the Test MSE of the MinMax function (Figure 5). It could appear because the predicted data values are higher than the initial set. Therefore, they would be outside the scaling range, which is based on our dataset's min and max values. The standardization (noted "Standard" on Figure 5) shows better results than normalization, mainly for Test MSE.

Another data scaling module exists, the RobustScaler. Instead of relying on the mean and scaling the unit variance, it relies on the median and interquartile range (between the first and third quartile of the data). This model makes it possible to take less into account the aberrant values which could disturb the results. As shown in Figure 5 (for the 70°C dataset), RobustScaler presents the most relevant results. For this reason, it was chosen as the method to scale the whole experimental dataset.

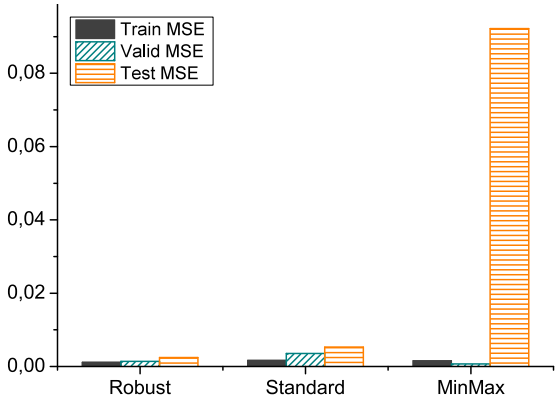


Fig. 5: Comparison of Train MSE, Valid MSE, and Test MSE for Robust, Standard, and MinMax scaling methods for the 70°C dataset.

The overall dataset is separated into a training set (representing 80% of the data), a validation set (10%), and a test set (10%). The learning dataset is used to build the model, the validation dataset is used to adjust the parameters, and the test dataset is used to compare network predictions with experimental data.

In general, samples should be randomly mixed to give the network a better learning phase. In the case of LSTM, this is not directly possible as the network would lose the temporal logic. Therefore, the dataset was divided into several subgroups, where samples in each group respect the chronological order. That makes it possible to mix the groups of data and not the data directly.

A time-series generator module from Tensorflow is used for training and testing datasets to create the necessary batches and sequences for training the network. Thanks to the time-series generator module, it is possible to dissociate the values of an input matrix X and an output matrix Y step by step. The matrix X corresponds to a batch of several temporal sequences with several temporal data. On the other hand, the Y matrix corresponds to the following time values for each defined time sequence so that the model can subsequently predict the values of Y. Figure 6 schematically presents the X and Y matrices. The batch size is an important parameter impacting the network prediction results.

The experimental data can now be integrated into the neural network characterized and adapted according to the hyperparameters. Hyperparameters will impact the performance of the LSTM network. They are divided into two categories according to their nature.

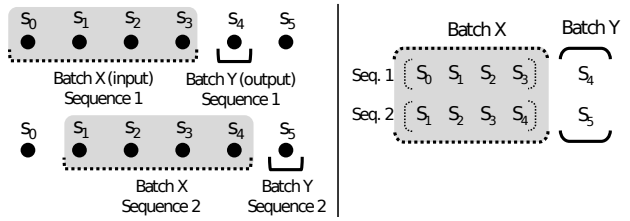


Fig. 6: Diagram of the division of data into sequence and batch.

B. Layer hyperparameters

In an LSTM network, it is possible to add *several hidden layers* to increase the network's complexity to cope with extensive data. It is also possible to define the *number of associated nodes* in each of these hidden layers. However, the more nodes per layer, the larger the number of parameters that will have to be computed by the network in order to operate and provide a reliable result.

The *dropout* is a parameter that makes it possible to deactivate, during the learning phase, specific neurons in order to improve the performance of the model by soliciting exclusively some neurons. It is a parameter that partially solves over-fitting. Generally, the dropout is between 0.2 and 0.5 (i.e., between 20% and 50% of the neurons are randomly deactivated).

Another hyperparameter is the *activation function*. It allows most of the time to mathematically define a nonlinear relationship between the inputs and outputs of a neuron in order to transfer or not input data to the next layer. Several activation functions exist, such as sigmoid, hyperbolic tangent, ReLU, ELU, or SELU. In our study, we chose the ELU and hyperbolic tangent ("tanh") functions as they provided the best preliminary results.

The ReLU (for Rectified Linear Unit) function only allows positive values to be transmitted, which can give inaccurate results. The ELU function (for Exponential Linear Unit) is an improvement of ReLU. Indeed, it smooths the function's output values when the input values are negative by relying on an exponential function that returns negative values. The following equation defines it.

$$f(x) = \begin{cases} \alpha(e^x - 1) & \text{if x strictly less than 0} \\ x & \text{otherwise} \end{cases} \quad (8)$$

The parameter α is a variable that makes it possible to control the slope of ELU when x is negative (the larger α, the steeper the curve). The mean of the function will, therefore, be closer to 0, and learning will be faster (because it is closer to the natural gradient of the function). The more the input value decreases, the more ELU saturates at a negative value, which induces a weak derivative and, therefore, more robust result stability.

C. Compilation hyperparameters

The compilation parameters are specific to the learning phase and this step's optimization. In particular, the following parameters can be defined:

- The *Adam* (for Adaptive Moment Estimation) optimization function is used for this model because it is currently the most efficient. It is an extension of stochastic gradient descent and was first proposed in 2015 by Kingma [41], based on adaptive learning rates [20].
- The *learning rate* indicates the speed at which the coefficients are updated. It can be fixed or variable; by default, it is set to 0.01.
- During the learning phase, the *number of epochs* makes it possible to define the number of iterations for which the training dataset will be transmitted to the network so that the latter gradually adjusts its weight and biases. Having many epochs to achieve a stable output result is not helpful, as Sen [42] had shown in his work.

V. DATABASE PRESENTATION

To train the LSTM and estimate which combinations of parameters give the best results, we have chosen to use the results of the three aging temperatures: 50°C, 70°C, and 90°C. As was shown previously, the evolutions of gravimetry for the three temperatures are different; therefore, the objective is, among others, to highlight a combination of parameters that makes it possible to obtain the best results, whatever the evolution of the data.

As it is impossible to test all the different combinations of parameters, some have been fixed in order to reduce the number of combinations tested, such as:

- Data scaling: Robust
- Batch size: 5% of the training dataset
- Loss: MSE
- Dropout: 0.2
- Optimization function: Adam
- Learning rate: 0.01
- Number of epoch: 30

However, we tested 16 different combinations for each aging temperature to show the impact of the different criteria chosen, such as:

- Interpolation function of experimental data: pchip or piecewise
- Number of layers: 1 or 2
- Number of neurons per layer: 50 or 150
- Activation function: ELU or hyperbolic tangent (tanh)

The impact of interpolation functions has been explained previously. Depending on the number of associated nodes, hidden layers can be more or less critical. The choice to test 1 or 2 layers was made based on examples in the literature [11], [13], [43]. So, testing the impact of the number of layers and associated nodes is essential.

VI. RESULT AND DISCUSSION

A. Comparison of different configurations

The network implementation was done using the Keras library, which includes a set of already coded functions that we can use directly. The architecture and construction of our network were also based on the work of Parouty [44] and Chollet [8].

For each combination, a repetition of 5 learning sessions was carried out in order to ensure the repeatability of the results. The results of the 16 combinations by temperature are shown in Figure 7, Figure 8, and Figure 9 for 50°C, 70°C and 90°C respectively. For each temperature, the results of the 3 MSEs are presented for the training phase, validation, and testing. To simplify the understanding of the different configurations, those carried out with the interpolation function pchip (denoted pch) appear in black, while those with piecewise (designated pi) appear in green. The configurations shown with hatching use the ELU activation function. Each combination is numbered depending on the number of layers (1 layer "1l" or 2 layers "2l") and knots (50 or 150).

A ranking of the best results for each of the datasets and the three best configurations by temperature is shown in Table I, Table II, and Table III:

TABLE I: MSE values for 50°C.

	Configuration	Train MSE	Valid MSE	Test MSE
1st	pi tanh 1l 150	0,015918	0,000157	0,000025
2nd	pi tanh 1l 50	0,024822	0,000069	0,000084
3rd	pi tanh 2l 50	0,034872	0,000465	0,000403

TABLE II: MSE values for 70°C.

	Configuration	Train MSE	Valid MSE	Test MSE
1st	pi tanh 1l 150	0,000988	0,001457	0,000995
2nd	pch tanh 1l 150	0,001167	0,001408	0,001447
3rd	pi tanh 2l 150	0,001388	0,002026	0,000127

TABLE III: MSE values for 90°C.

	Configuration	Train MSE	Valid MSE	Test MSE
1st	pch tanh 1l 150	0,002614	0,000090	0,000151
2nd	pi tanh 1l 150	0,00228	0,000141	0,000643
3rd	pch elu 1l 150	0,002876	0,000166	0,000232

Given this ranking, the best combinations could be highlighted for the three temperatures:

- 50°C: the 1l/150/tanh/piecewise configuration
- 70°C: the 1l/150/tanh/piecewise configuration
- 90°C: the 1l/150/tanh/pchip configuration

B. Prediction analyze

We obtain two configurations that give the best results for the three temperatures: 1l/150/tanh/piecewise and 1l/150/tanh/pchip. The model parameters appear more discriminating than the interpolation function to prepare experimental input data. Indeed, given the graphs presented previously for the three datasets, it is difficult to conclude the most convincing interpolation function to obtain the best results. As long as the interpolation function is consistent with the evolution of the experimental data and avoids the appearance of Runge's phenomenon, then it has little impact on the prediction results.

However, these results make it possible to highlight other conclusions on the impact of the choice of other parameters:

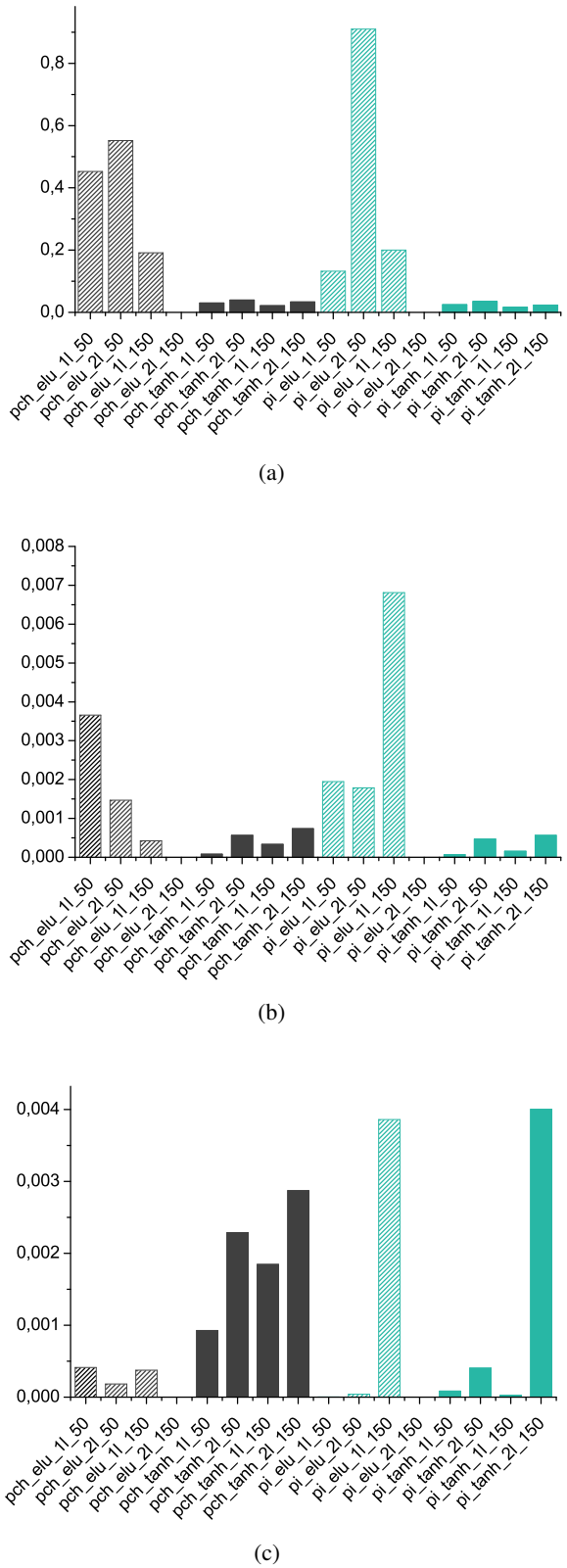
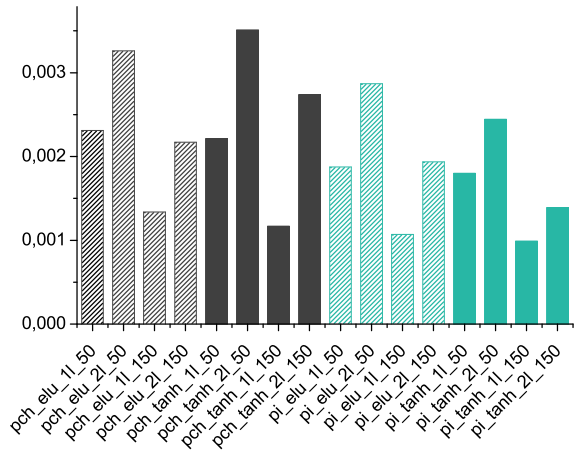
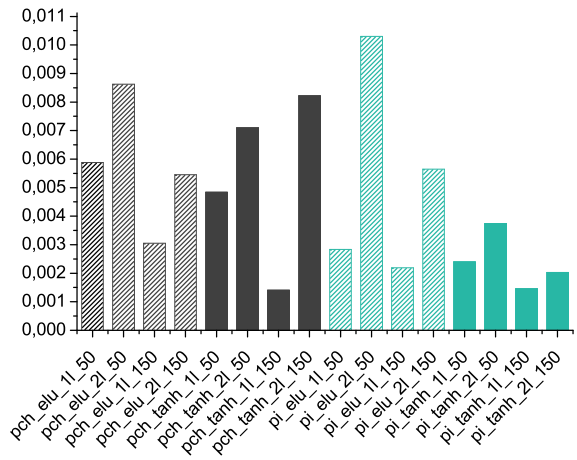


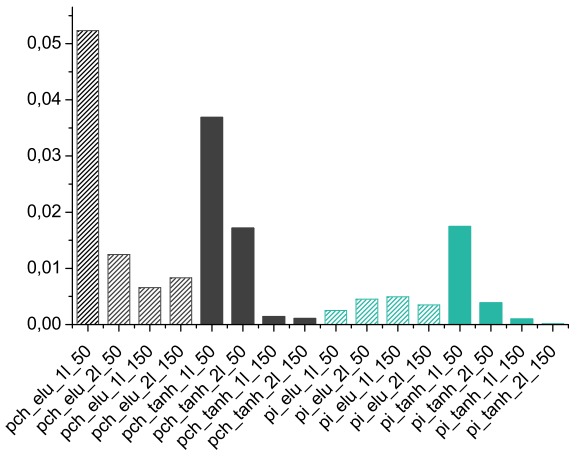
Fig. 7: 50°C dataset: with a) train MSE, b) validation MSE, and c) test MSE.



(a)

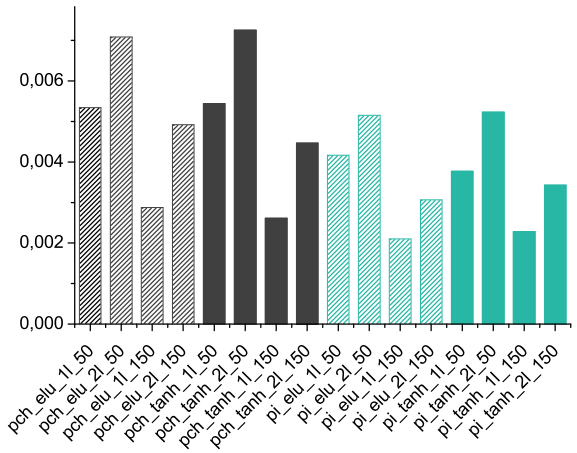


(b)

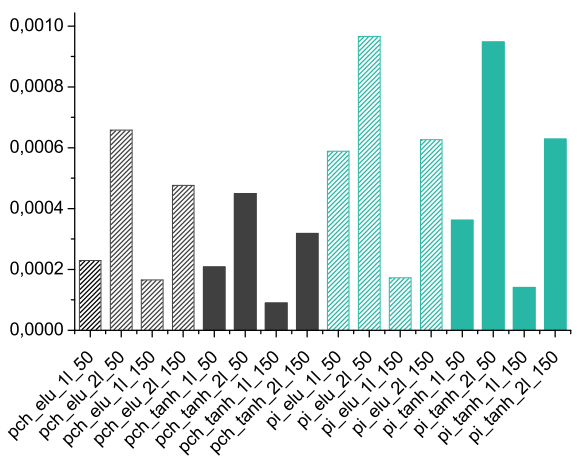


(c)

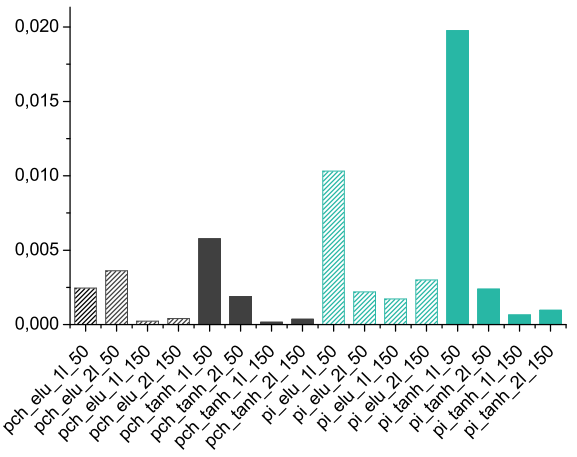
Fig. 8: 70°C dataset: with a) train MSE, b) validation MSE, and c) test MSE.



(a)



(b)



(c)

Fig. 9: 90°C dataset: with a) train MSE, b) validation MSE, and c) test MSE.

Layers number influence

- A too-complex network with too many layers does not provide better results.

Indeed, almost no 2-layer configuration appears among the top three configurations. Compared to the input data, a too-complex network would result in more difficult network convergence. Indeed, a relatively light network can process the information and provide relevant results for input data with a single parameter. For all the analyzed cases, for identical parameters, the Train MSE is always more important for a 2-layer network. For Valid MSE and Test MSE, overall, the same trends appear. This result highlights that an excess of comparable data does not provide more information. In the literature, some authors have also tested the impact of the number of layers on their network. Salman's article [45] shows that a multilayer network provides better results than a single-layer network. However, their dataset consists of 4 different parameters for more than 40,000 meteorological data, so it is a very different configuration from ours. That can explain the differences in results. For its part, Halpern-Wight's paper [46] presents an LSTM network working on a more restricted number of data (less than 300). It also highlighted that a 5-layer network did not make it possible to achieve better performance than a single-layer network. Another authors [47] also showed that a 2-layer LSTM achieves better results than a 3- and 4-layer network. According to our observations, this excess of data might prevent the network from converging to the expected results. Another issue is that the more layers there are, the more the number of parameters increases, raising the computation times without ultimately leading to better results.

Nodes influence

- 150 nodes per layer provide more relevant results than 50 nodes.

In the previous graphs, increasing the number of nodes makes it possible to decrease the values of the different MSEs compared to the 50-node configurations, independently of the number of layers used (mainly for 70°C and 90°C). Indeed, going from 50 to 150 nodes multiplies by more than eight times the number of parameters. For one layer, 50 nodes give 10,451 parameters, and 150 nodes correspond to 91,351 parameters, and for two layers, there is the same ratio (50 nodes: 30,651 parameters and 150 nodes 271,951 parameters). On average, 150 nodes instead of 50 allow us to reduce the MSE values by 0.011942 for the learning step, 0.000645 for the validation step, and 0.006292 for the test step.

Activation function influence

- The hyperbolic tangent activation function provides better results than the ELU function.

Eight of the nine best configurations presented for the three temperatures are realized with the hyperbolic tangent activation function, allowing us to obtain good results. In addition, for the dataset at 50°C, the impact of the activation function is very marked because much larger Train and Valid MSEs are to be noted when using the ELU function. For the configuration with two layers and 150 nodes, converging and obtaining a result during learning is impossible. In contrast, for the hyperbolic tangent function, the results obtained are of the same order of magnitude as the other configurations with this function. Dubey [48] and Kim [49] also compared different activation functions using an LSTM network. They report a better result for the hyperbolic tangent. For its part, the ELU function would be more appropriate for classification issues, according to Dubey.

The objective of using a neural network is to predict unknown future data, which makes it possible to anticipate the aging of materials in the case of our study. Contrary to the majority of the articles in the literature, we judged it essential to present the results of this prediction to show the relevance of using such a network. After determining the best combination of parameters, the model "piecewise_tanh_1l_150" was chosen to apply to our experimental dataset.

To apply the LSTM model to our datasets, we created a matrix to save each data resulting from the prediction from a data sequence. Our data sequence matches the size of our test dataset. In the case of this study, it is composed of the last 80 data, and each time, we will ask the network to predict the next unknown future value noted x_{new_i}. This value x_{new_i} will be added to the previously created matrix and the data sequence. The prediction process is restarted several times. The prediction results are therefore presented in Figure 10, Figure 11, and Figure 12. In each figure, we find the raw experimental data in black points and the interpolated data in solid orange lines, which are used to train the network. In the "test" area, we find the comparison between the test dataset's known values and the LSTM network's prediction results (in the dash on the graph). A zoom of this area is also shown on the graphs. The prediction made beyond the known values is represented with dots and is compared with an additional known experimental value (a cross). Therefore, it is possible to see the gap between the prediction and experimental reality, accompanied by measurement uncertainties. In the zoomed comparison area, the difference between the last value predicted by the network and the last value of the test dataset, which is also a known experimental value, has been compared in Table IV (for an aging time of 4878h).

TABLE IV: Comparison of MSE values for the last experimental point at 50°C, 70°C, and 90°C.

Temperature	MSE
50°C	5,05592E-06
70°C	0,002467271
90°C	0,0012436

These results are relevant and promising. Indeed, despite changes in the kinetic of water intake, therefore, changes in the gravimetric curve shape, the prediction results provided by the neural network are reliable. It is highlighted, on the one hand, by the results of the MSE test and, on the other hand, by the difference in the prediction between the future data and the last experimental data (the cross). The model, therefore, makes it possible to adapt to these various changes and evolutions. The prediction deviations measured for the last experimental point of the dataset supplied to the LSTM show a slight deviation for 70°C and 90°C. This deviation is

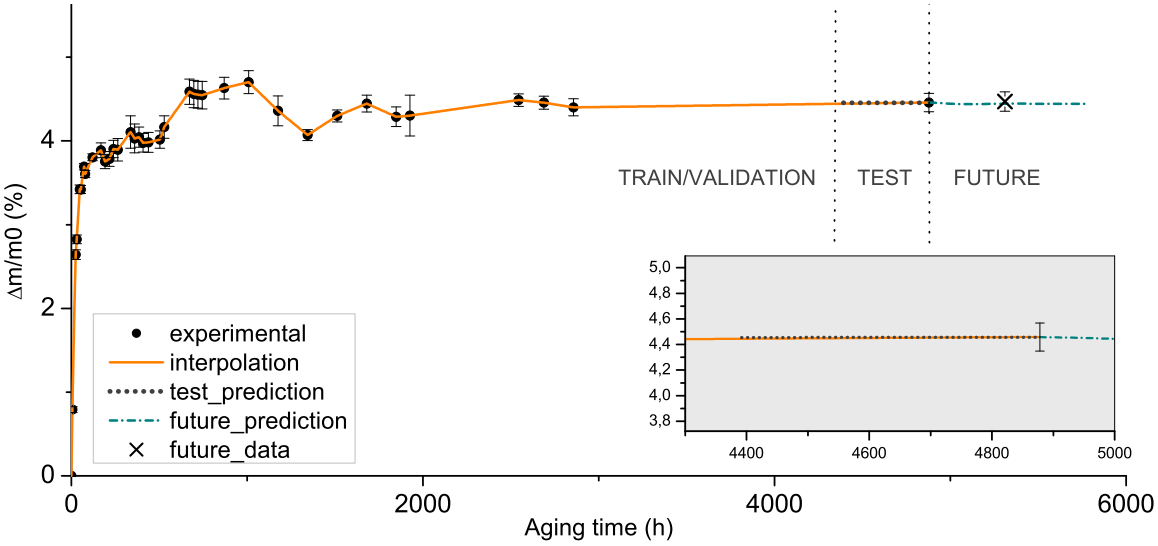


Fig. 10: Future prediction for gravimetry at 50°C with comparison of test prediction.

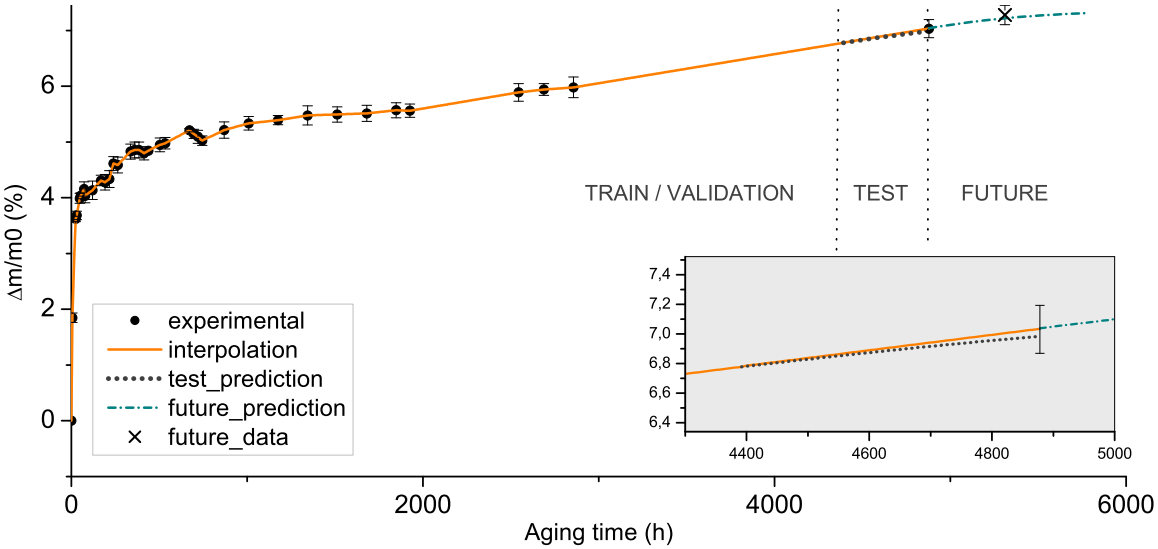


Fig. 11: Future prediction for gravimetry at 70°C with comparison of test prediction.

likely to be reflected in future predictions. The prediction results for 90°C seem to be further from experimental reality than 50°C and 70°C, which are very close. However, looking at the last aging times, the experimental point and the measurement uncertainty associated with this value are considerable. At 90°C, the aging conditions are more violent, leading to a more critical diffusion of water and new physical and chemical phenomena, such as hydrolysis and exudation. In the long term, this will damage the material and impact its properties. Under these conditions, the uncertainties are more significant because the samples are more damaged and do not all respond similarly to this aging at a time t. Consequently, it is consistent that the neural network, which relies on much noisier data for 90°C, gives a result that corresponds less to experimental reality. Despite this discrepancy, the prediction remains within the uncertainty range of the experimental point. Beyond the network parameters, it is, therefore, essential to choose the correct experimental data representative of the evolution of the chosen parameter. Moreover, the stable evolution of the parameters of Train MSE and Valid MSE (shown in Figure 13 for each temperature) ensures that no over-fitting phenomenon appears, especially when the datasets are small.

For this dataset and network configuration, predicting up to 40 days of aging into the future with relatively low error rates, compared to six months of hygrothermal aging, is possible. Indeed, this possibility of prediction, taking into account the architecture we used in the code, is based on the size of the dataset of the test phase. It corresponds to 10% of the 4878h of aging, i.e., approximately 480h. That is one of the limits of this tool because it cannot, from a small dataset, predict data over a very long time.

VII. CONCLUSION

In this work, we studied an LSTM-type neural network to predict the evolution of the mass setting of an epoxy adhesive during hygrothermal aging. The data collected for three aging temperatures were used as training, validation, and testing data for the network. The steps were presented to collect the data, interpolate the samples to fill in missing

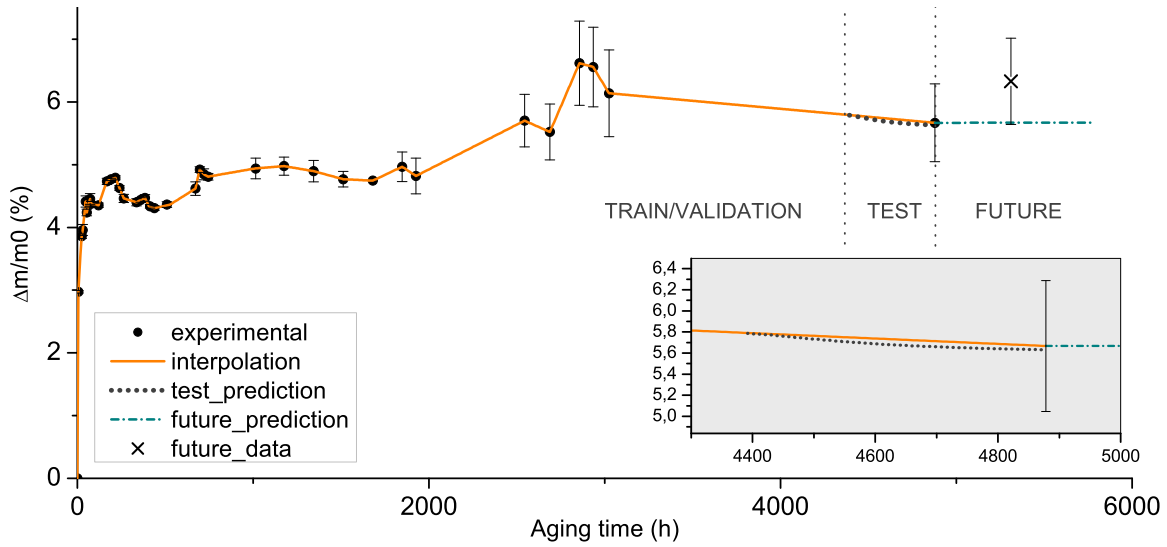


Fig. 12: Future prediction for gravimetry at 90°C with comparison of test prediction.

data and train an LSTM network. A large combination of parameters was tested for the LSTM network to highlight their influence on the training and predictions. This study showed the importance of specific parameters in improving the prediction results, such as the number of hidden layers, the number of nodes, and the activation function.

This study showed that using AI tools to predict the evolution of a physical parameter could be relevant. It was shown that a too-complex network may weigh down the calculations and will not allow better results. It was also pointed out that a complex multilayered network is not helpful or even counterproductive when faced with a simple, single-parameter dataset.

In this study, the optimal combination that achieved excellent results considering the performance indicators Train MSE, Valid MSE, and Test MSE against the experimental data of the three aging temperatures is as follows.

Fixed parameters:

- Learning/validation/test split: 80% / 10% / 10%
- Time step: 6h
- Data scaling: RobustScaler
- Batch size: 5% dataset of the training dataset
- Number of epochs: 30
- Loss: MSE
- Dropout: 0.2
- Optimization function: Adam
- Learning rate: 0.01

Best parameters compared:

- Interpolation function: pchip or piecewise
- Number of layers: 1
- Number of nodes/layer: 150
- Activation function: hyperbolic tangent

However, one drawback of this method is that a neural network cannot predict something it has not learned during its training. The network does not know the physical phenomena that may occur during hygrothermal aging and could generate new degradation kinetics. Therefore, it will deviate from experimental reality, such as during aging at 90°C. It allows us to have a vision of the evolution of the parameters only if it has already been confronted with this trend. It is, therefore, essential to have a relatively complete dataset that provides enough information to the network. Our next objective is to use this neural network on more chaotic experimental data from distinct characterization tests, such as infrared spectroscopy and rheology, to highlight the importance of data selection and the limitations of this network. For instance, seeing the influence of other neural network parameters, such as learning rate or batch size, could also be interesting for an application with noisier data. The prediction gap in the test phase could be reduced and thus allow us to refine the network's prediction results.

References

[1] D. M. Dimiduk, E. A. Holm, and S. R. Niezgoda, "Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering," *Integrating Materials and Manufacturing Innovation*, vol. 7, no. 3, pp. 157–172, Sep. 2018.

[2] W. Nash, T. Drummond, and N. Birbilis, "A review of deep learning in the study of materials degradation," *npj Materials Degradation*, vol. 2, no. 1, pp. 1–12, Nov. 2018.

[3] W. Zhang, T. Jiang, and X. Li, "Life-prediction of the CSADT based on BP algorithm of ANN," in *2009 Annual Reliability and Maintainability Symposium*, Jan. 2009, pp. 59–63.

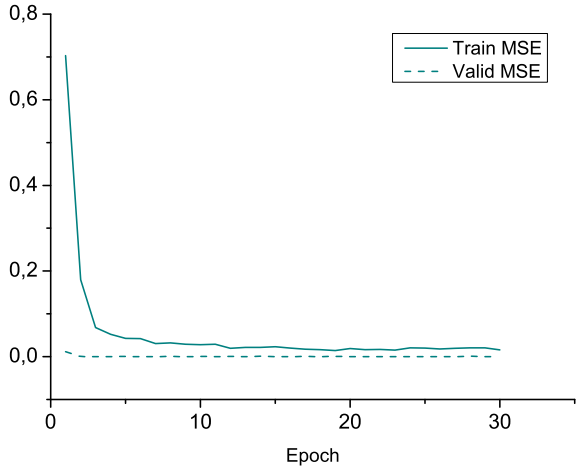
[4] S. Pruksawan, G. Lambard, S. Samitsu, K. Sodeyama, and M. Naito, "Prediction and optimization of epoxy adhesive strength from a small dataset through active learning," *Science and Technology of Advanced Materials*, vol. 20, no. 1, pp. 1010–1021, Dec. 2019.

[5] K.-L. Xiang, P.-Y. Xiang, and Y.-P. Wu, "Prediction of the fatigue life of natural rubber composites by artificial neural network approaches," *Materials & Design*, vol. 57, pp. 180–185, May 2014.

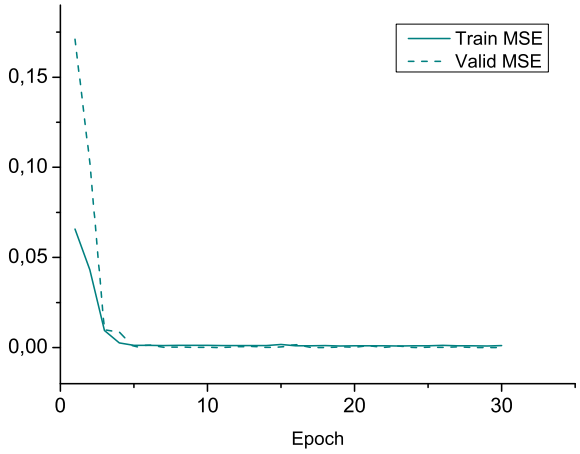
[6] S. Bishnoi, S. Singh, R. Ravinder, M. Bauchy, N. N. Gosvami, H. Kodamana, and N. M. A. Krishnan, "Predicting Young's modulus of oxide glasses with sparse datasets using machine learning," *Journal of Non-Crystalline Solids*, vol. 524, p. 119643, Nov. 2019.

[7] A. Doblies, B. Boll, and B. Fiedler, "Prediction of Thermal Exposure and Mechanical Behavior of Epoxy Resin Using Artificial Neural Networks and Fourier Transform Infrared Spectroscopy," *Polymers*, vol. 11, no. 2, p. 363, Feb. 2019.

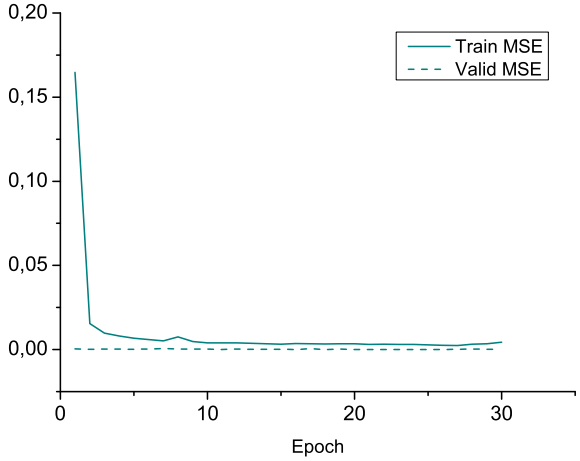
[8] F. Chollet, *Deep Learning with Python*. Simon and Schuster, Nov. 2017.



(a)



(b)



(c)

Fig. 13: Train MSE, and Valid MSE evolution during learning a) at 50°C b) at 70°C, and c) at 90°C.

[9] Y. Zhu, T. Zhao, J. Jiao, and Z. Chen, "The lifetime prediction of epoxy resin adhesive based on small-sample data," *Engineering Failure Analysis*, vol. 102, pp. 111–122, Aug. 2019.

[10] X. Ding, X. Hou, M. Xia, Y. Ismail, and J. Ye, "Predictions of macroscopic mechanical properties and microscopic cracks of unidirectional fibre-reinforced polymer composites using deep neural network (DNN)," *Composite Structures*, vol. 302, p. 116248, Dec. 2022.

[11] Y.-C. Hsu, C.-H. Yu, and M. J. Buehler, "Using Deep Learning to Predict Fracture Patterns in Crystalline Solids," *Matter*, vol. 3, no. 1, pp. 197–211, Jul. 2020.

[12] N. Zhang, S.-L. Shen, A. Zhou, and Y.-F. Jin, "Application of LSTM approach for modelling stress–strain behaviour of soil," *Applied Soft Computing*, vol. 100, p. 11, Mar. 2021.

[13] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, Jul. 2005.

[14] N. Yudistira, "COVID-19 Growth Prediction using Multivariate Long Short Term Memory," *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp. 829–837, 2020.

[15] X. Song, F. Yang, D. Wang, and K. Tsui, "Combined CNN-LSTM Network for State-of-Charge Estimation of Lithium-Ion Batteries," *IEEE Access*, vol. 7, pp. 88 894–88 902, 2019.

[16] S. Goswami, R. Ghosh, A. Neog, and B. Das, "Deep learning based approach for prediction of glass transition temperature in polymers," *Materials Today: Proceedings*, vol. 46, pp. 5838–5843, Jan. 2021.

[17] I. M. R. Najjar, A. M. Sadoun, M. Abd Elaziz, A. W. Abdallah, A. Fathy, and A. H. Elsheikh, "Predicting kerf quality characteristics in laser cutting of basalt fibers reinforced polymer composites using neural network and chimp optimization," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 11 005–11 018, Dec. 2022.

[18] J. Meng, N. Ma, Z. Jin, Q. Liu, and Z. Yue, "Prediction of tear propagation path of stratospheric airship envelope material based on deep learning," *Engineering Fracture Mechanics*, vol. 282, p. 109183, Apr. 2023.

[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[20] J. Brownlee, "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning," Jul. 2017.

[21] S. Popineau, "Durability in humid environments of aluminum/composite type bonded structural assemblies - durabilité en milieu humide d'assemblages structuraux collés type aluminium/composite," Doctoral thesis, École Nationale Supérieure des Mines de Paris, Mar. 2005.

[22] G. Bresson, "Reliable bonding for space: influence of process quality and assembly sizing - collage fiable pour l'espace : influence de la qualité des procédés et dimensionnement des assemblages," Doctoral thesis, Bordeaux 1, Nov. 2011.

[23] B. De'Neve and M. E. R. Shanahan, "Water absorption by an epoxy resin and its effect on the mechanical properties and infra-red spectra," *Polymer*, vol. 34, no. 24, pp. 5099–5105, Dec. 1993.

[24] S. Popineau, C. Rondeau-Mouro, C. Sulpice-Gaillet, and M. E. R. Shanahan, "Free/bound water absorption in an epoxy adhesive," *Polymer*, vol. 46, no. 24, pp. 10 733–10 740, Nov. 2005.

[25] B. Fayolle and J. Verdu, "Physical aging of polymer materials - vieillissement physique des matériaux polymères," *Techniques de l'Ingénieur*, 2005.

[26] S. Bistac, M. F. Vallat, and J. Schultz, "Durability of steel/polymer adhesion in an aqueous environment," *International Journal of Adhesion and Adhesives*, vol. 18, no. 5, pp. 365–369, Oct. 1998.

[27] J. Zhou and J. P. Lucas, "Hygrothermal effects of epoxy resin. Part I: the nature of water in epoxy," *Polymer*, vol. 40, no. 20, pp. 5505–5512, Sep. 1999.

[28] C. E. Browning, "The mechanisms of elevated temperature property losses in high performance structural epoxy resin matrix materials after exposures to high humidity environments," *Polymer Engineering & Science*, vol. 18, no. 1, pp. 16–24, 1978.

[29] R. J. Morgan, J. E. O'neal, and D. L. Fanter, "The effect of moisture on the physical and mechanical integrity of epoxies," *Journal of Materials Science*, vol. 15, no. 3, pp. 751–764, Mar. 1980.

[30] B. Dao, J. Hodgkin, J. Krstina, J. Mardel, and W. Tian, "Accelerated aging versus realistic aging in aerospace composite materials. V. The effects of hot/wet aging in a structural epoxy composite," *Journal of Applied Polymer Science*, vol. 115, no. 2, pp. 901–910, 2010.

[31] J. Verdu, "Action of water on plastics - action de l'eau sur les plastiques," *Plastiques et composites*, Jan. 2000.

[32] D. Colombini, J. J. Martinez-Vega, and G. Merle, "Dynamic mechanical investigations of the effects of water sorption and physical ageing on an epoxy resin system," *Polymer*, vol. 43, no. 16, pp. 4479–4485, Jul. 2002.

[33] J. Delozanne, "Durability of epoxies; application to aeronautical structural bonding - durabilité des époxys ; application au collage structural aéronautique," Doctoral thesis, Paris, ENSAM, Dec. 2018.
[34] H. Ozdemir, "Comparison of linear, cubic spline and akima interpolation methods," Tech. Rep., 2007.
[35] H. Akima, "A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures," *Journal of the ACM*, vol. 17, no. 4, pp. 589–602, Oct. 1970.
[36] "Top (GNU Octave (version 8.1.0))."
[37] A. Salvail-Berard, "Neural networks - réseaux de neurones," vol. 3, pp. 54–69, 2012.
[38] H. E. Balcioglu and A. Seçkin, "Comparison of machine learning methods and finite element analysis on the fracture behavior of polymer composites," *Archive of Applied Mechanics*, Sep. 2020.
[39] Y. Qu and X. Zhao, "Application of LSTM Neural Network in Forecasting Foreign Exchange Price," *Journal of Physics: Conference Series*, vol. 1237, no. 4, p. 042036, Jun. 2019.
[40] S. S. Haykin, *Neural networks and learning machines*, iii ed. New York: Prentice Hall, 2009.
[41] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *ArXiv*, 2015.
[42] H. Sen, "Time Series Prediction based on Improved Deep Learning," *IAENG International Journal of Computer Science*, vol. 49, no. 4, pp. 1133–1138, 2022.
[43] A. Yadav, C. Jha, and A. Sharan, "Optimizing LSTM for time series prediction in Indian stock market," *Procedia Computer Science*, vol. 167, pp. 2091–2100, Jan. 2020.
[44] J.-L. Parouty, S. Arias, and E. Maldonado, "Formation Introduction au Deep Learning," Feb. 2021.
[45] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting," *Procedia Computer Science*, vol. 135, pp. 89–98, Jan. 2018.
[46] N. Halpern-Wight, M. Konstantinou, A. G. Charalambides, and A. Reinders, "Training and Testing of a Single-Layer LSTM Network for Near-Future Solar Forecasting," *Applied Sciences*, vol. 10, no. 17, p. 5873, Jan. 2020.
[47] T. Zhang, S. Song, S. Li, L. Ma, S. Pan, and L. Han, "Research on Gas Concentration Prediction Models Based on LSTM Multidimensional Time Series," *Energies*, vol. 12, no. 1, p. 161, Jan. 2019.
[48] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, pp. 92–108, Sep. 2022.
[49] S. Kim, S. Choo, G. Lee, and S. Kim, "Predicting Demand for Shared E-Scooter Using Community Structure and Deep Learning Method," *Sustainability*, vol. 14, no. 5, p. 2564, Jan. 2022.