

Prediction of Ductile Damage in Composite Material Used in Type IV Hydrogen Tanks by Artificial Neural Network and Machine Learning with Finite Element Modeling Approach

Kheireddin Kadri, Achraf Kallel, Guillaume Guerard, Abir Ben Abdallah, Sébastien Ballut, Joseph Fitoussi, and Mohammadali Shirinbayan*

This study investigates the degradation process of composite materials used in high-pressure hydrogen storage vessels by employing advanced computational techniques. A recurrent neural network, specifically a bidirectional long short-term memory (Bi-LSTM) network, is utilized to predict the temporal evolution of ductile damage. The key degradation features are extracted from finite element modeling (FEM) computations using group method of data handling algorithms and treated as time-series data. Results demonstrate that the Bi-LSTM network can accurately undergo both elastic and plastic behaviors of the composite under tensile strength. Additionally, traditional machine learning (ML) algorithms such as extreme gradient boosting and random forest are employed to forecast strain degradation, showing promising results. This hybrid approach combining FEM, ML, and deep learning provides a comprehensive method for predicting the degradation of composite materials, offering significant potential for optimizing the design and durability of hydrogen storage vessels.

alternative materials in various applications due to their promising properties. Composite fiber-reinforced polymers (CFRP), which consist of a polymer matrix and reinforcing fibers like carbon fiber, are particularly noteworthy for their use in hydrogen storage vessels owing to their lightweight, high-strength, and corrosion-resistant characteristics. However, the design and testing of these vessels are often costly and time-consuming, impeding rapid iteration and improvement of designs.

In recent decades, there has been a rapid convergence between the field of machine learning (ML) and industrial applications of CFRP. From a general landscape overview some recent literature's papers^[1] had shed lights on the applications of the ML to improve properties of CFRP. Wang et al. had examined the use of ML in predicting

1. Introduction

Composite materials, including high-performance fibres such as glass, carbon, aramide, and organic fibers, have emerged as


the long-term performance of fibre-reinforced polymer (FRP) structures. This work showed how ML models can address the nonlinear and complex nature of durability problems in FRP composites under different environmental conditions. The review emphasizes the importance of variables like exposure time, temperature, pH value, and fibre volume fraction in predicting the residual mechanical properties of FRP composites. For this reasons in other research works,^[2] authors concentrated in particular on various mechanical problems in fibre composite materials (FCMs), such as constitutive laws, defect detection, and fatigue failure. They highlighted effectiveness of ML models like artificial neural network (ANNs), support vector machines (SVMs), or convolutional neural networks (CNNs), in solving complex problems that traditional methods struggle to address. Liu et al.^[2] evinced the advancements in automated defect detection and impact dynamics analysis through ML like autoencoders (AEs) for noise reduction in image data and, convolutional neural networks (CNNs) for impact damage monitoring. The mechanical behavior of laminated FRP composites had become a central matter of research.^[3] The detailed exploration of various ML algorithms, including deep learning (DL) models like CNNs and LSTMs, proved the potential of these methods in solving complex problems related to FRP composites. A key stone is the successful combination of ML models with finite element modeling (FEM), allowing more accurate and efficient predictions of composite behavior. In this state-of-art review,^[4] authors insist on the

K. Kadri, A. Ben Abdallah, S. Ballut
R&D Departement
Aptiskills
28 Rue Edouard Vaillant, 92300 Levallois-Perret, France

A. Kallel
Direction de Programmes et Projets
IRT SystemX
91120 Palaiseau, France
E-mail: achraf.kallel@irt-systemx.fr

A. Kallel, J. Fitoussi, M. Shirinbayan
Laboratoire PIMM
Arts et Métiers
CNRS
Hesam Université
151 boulevard de l'Hôpital, 75013 Paris, France

G. Guerard
De Vinci Research Center
Léonard de Vinci Pôle Universitaire
92916 Paris La Defense, France

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/ente.202401045>.

DOI: 10.1002/ente.202401045

role of ML techniques for process simulation. The article offers a comprehensive overview of ML application in the process simulation of polymer composites. It focuses on highlighting the potential of ML in enhancing the efficiency and accuracy of simulations. This last point is traditionally computationally expensive and complex. Several case studies and applications where ML has been successfully integrated with conventional simulation techniques like FEM. For instance, optimizing fibre placement in automated processes was addressed.^[4] The versatility of ML across different aspects of composite manufacturing was reported. The authors^[4] pointed out promising ML techniques such as graph neural networks (GNNs) and physics-informed neural networks (PINNs) as potential future tools for improving the simulations process.

The major drawbacks that were commonly reported from those articles' reviews are the following points. First there is a limited scope of ML applications^[1] the current applications are primarily focused on tensile properties of FRP. There is a limited exploration of other mechanical properties such as shear and compressive strengths. The second point of weakness is the limited generalizability due to dataset quality.^[1,2] The concern of physical principles integration for interpretability was clearly highlighted.^[2] The current methods often overlook the physical meaning of parameters, leading to a loss of model interpretability. This lack of interpretability can be a significant drawback, especially in scientific applications where understanding the underlying physical phenomena is crucial. The last issue being pointed out^[3,4] is inherent to the field of ANNs. In fact there are difficulties in selecting and configuring the appropriate ML model for specific tasks. The complexity of determining the optimal network architecture for DL models or choosing between conventional and unconventional ML approaches is a recurrent interrogation.

More specifically several articles were dealing with applications of ML in mechanical property prediction. Two principal axes of research have emerged according to the level of the numerical simulation realized. The first axe regroups macroscopic mechanical properties.^[5-9] In this article,^[5] the developed model is a regression approach for predicting macroscopic transverse mechanical properties of FRP laminae. The dataset was generated by discrete element method (DEM) simulations of 2000 representative volume element (RVE) with 200 different sets of fibre volume fractions and fibre radii. The model was able to predict the macroscopic transverse mechanical properties.

While the second axe of research concerns the microscopical level on damage modeling that occurs in CFRP.^[10-12] Li et al. has reported a coupled approach to overcome the inherent difficulty to characterize the anisotropic plasticity. In this work,^[10] a mechanics-informed ML approach that enables to employ a small training database is proposed to predict the elastoplastic behaviors of unidirectional CFRP. Direct simulations based on a RVE were used for validating the ML model. It was shown that the proposed ML constitutive model offers excellent predictive accuracy even when using a small training dataset.

ML techniques were specifically used in studying composite aging and degradation. Numerous studies^[1,13] concerning aging and durability prediction have been catching an increasing interest. Very lately, advanced ML techniques for specialized composite simulations have met a tremendous interest. In a

close relation to industrial concern about a better characterization of damage triggering and cracks propagation's, real-time monitoring and prediction became a new paradigm.^[14] In this work,^[15] a process-induced distortions (PIDs) of asymmetrical laminates was predicted accurately and tailored at the early design stage. Practically, rapid prediction of composite materials behaviour using ANNs was realized. As stated by the authors of this article,^[16] because of the heterogeneity of composite materials, there is a dilemma. Either "computational bottlenecks", if modeled rigorously or there is a solution inaccuracies in the stress field and failure predictions if approximated. State-of-art models of neural networks have been successfully deployed in order to overcome that dilemma. A surrogate models that was calibrated with a single microstructure was implemented. The major outcome was obtaining a transformer network predicting the history-dependent, nonlinear, and homogenized stress-strain response at microstructure level.

RNNs were applied in the specific case of woven composites exhibit complex meso-scale behavior.^[17] As a basis for RNN training, a mean-field model generated a comprehensive dataset representing elastoplastic behavior. Time history of strain was used to generate multiaxial stress-strain histories under random walking a cyclic loading condition. In this principal contribution of research,^[13] a self-constructed strategy-based reinforcement LSTM approach was adopted to improve generalization ability. Two datasets, one consisting of finite element analysis (FEA) results, and one of real experimental (EXP) data were selected to verify the validity of the degradation performance predictions. It was shown that the reinforced LSTM is more suitable for the nonlinear degradation performance analysis of FRP, especially with higher prediction accuracy for experimental data.

2. Problem Position and Model

2.1. Problem Position

The problem explored in this study, is the temporal evolution of a ductile damage for a composite material under external tensile force. In fact, the main difficult issue is the non-linear aspect of degradation. Once values of physical properties are calculated from FEM during a limited time of loading, ML model will be able to make prediction for degradation. For this reason, we propose to combine a FEM and an ANN in order to obtain a hybrid model with optimal characteristics. In fact, FEM allows to have a considerable amount of results depending on the size of the simulated 3D volume and its degree of discretization. Then, this quantity of outcomes is advantageously used by the ANN. Besides, as it is difficult to have a complete physical model describing every degradation feature of a composite, we propose to employ an alternative approach. By confronting a ML model to a large number of composite material samples submitted to external solicitation, this model will predict the temporal evolution of strain degradation by only preserving the main features.

2.2. Finite Element Model and Materials

This model will use the data generated by the FEM to feed the ANN in order predict the composite ductile degradation. At first,

the hybrid model aims to generate a trained ANN to have an overview of the into degradation occurring during a short loading time. Second, it attempts to predict the degradation's evolution during a time laps, then to forecast future evolution. Last goal is to characterize more precisely the response of a composite material submitted to tensile strength solicitation.

The general methodology used in this study for numerical modeling is a classical schema^[18] commonly followed. Commercially available, finite element software, Abaqus was used to perform a numerical simulation of mechanical properties by using representative volume elements (RVEs).^[19] The shape of the RVE is chosen according to the microstructure.^[20] This space reduction allows to capture the entirety of the physical problem, and reduce the time machine while preserving a faithful determination of composite properties during modeling. A centred cubic body RVE is chosen since the microstructure is symmetrical.^[21–23] In this study, unidirectional fiber is distributed at the center and the corner of the RVE, the body-centered cubic symmetry is chosen as an ideal representation of a unidirectional composites material.^[24]

A $1 \mu\text{m}^3$ volume of the RVE is submitted to tensile traction in (Z) direction during a static loading. Then data extracted from this one type of external solicitation are employed to feed the ML/DL models, which will permit to depict the temporal evolution of degradation process.

The studied composite material is composed from poly-amid (PA12) matrix and reinforced with aramid glass fibers (AF). The explored volume fraction of these long fiber is 35%.^[25]

Table 1 presents the Young modulus E_{Young} and Poisson's ratio ν of both fiber and matrix.

The used plastics values for fibers are taken from the following references (Campus datasheet DUPONT engineering Plastic Zytel 70K20HSL BK284-PA66AF20). **Figure 1**, describes the RVE submitted to tensile solicitation.

Both matrix and fiber are defined within Abaqus as an elasto-plastic material.^[26] Values of these mechanical properties are resumed in Table Tabular values 2 (**Table 2**).

The generalization of Hooke's law to a 3D continuum is a relationship that can be interpreted as the elasticity tensor of a fourth-rank. It describes the stress–strain relation in a linear elastic material. Termed tensor of stiffness \bar{C} , it is the elasticity response of the heterogeneous material under solicitation, obtained numerically. This tensor depends on different input variables that are precised in the next Equation (1)

$$\bar{C}(F_v, E_{\text{matrix}}, \nu_{\text{matrix}}, E_{\text{fiber}}, \nu_{\text{fiber}}) \quad (1)$$

where F_v is the volume fraction of the fibre inside the matrix.

Table 1. Detailed range values of the input parameters used in the python script: matrix and fiber properties, volume fraction percentage, and solicitation directions. The data are taken from the following website ref. [61].

Matrix name	Fiber name	Volume fraction	Solicitation
$(E_{\text{Young}} [\text{MPa}], \nu)$	$(E_{\text{Young}} [\text{MPa}], \nu)$	[%]	External force
PA12 (1250, 0.39)	Glass (8000, 0.34)	35%	Tensile Z
		–	–

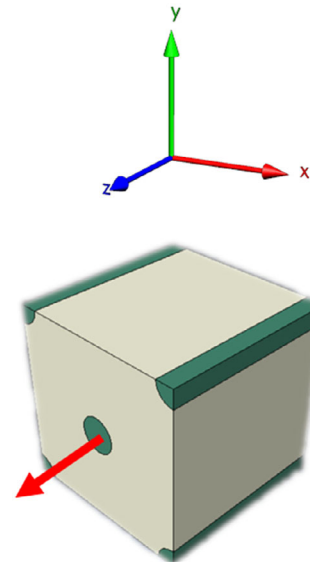


Figure 1. Illustration of a periodical RVEs of idealized unidirectional oriented long fibre composites. The RVE made of the (PA12, glass) composite designed with Abaqus software. The tensile strength solicitation realized during the homogenization method is schematized by a red arrow.

Table 2. Explored range of the input parameters of the plastic behaviour P_s and the yield value Y_s of both matrix and fiber used in FEM simulation. Extracted from the open access database.^[61]

PA12		Aramid fiber	
Plastic strain [MPa]	Yield stress [%]	Plastic strain [MPa]	Yield stress [%]
0	0	0	0
0.16	0.01	26.02	0.54
7.71	0.5	49.28	1.08
14.8	1.0	68.01	1.62
21.3	1.5	81.52	2.17
26.9	1.99	89.97	2.71
32	2.5	95.14	3.25
36	3.0	98.62	3.79
38.3	3.5	101.39	4.32
39.7	4.0	103.83	4.86
40.3	4.5	105.34	5.4
40.5	5.0	–	–
191.22	2.31	–	–
200.01	2.6	–	–
204.36	3.01	–	–

E_{matrix} and E_{fibre} are respectively the Young's modulus of matrix and fibre. ν_{matrix} and ν_{fibre} are respectively the Poisson's index of matrix and fibre. Here, \bar{C} is the relationship between the stress $\bar{\sigma}$ and strain $\bar{\epsilon}$ of the homogenized heterogeneous material^[27] obtained according to the following Equation (2)

$$\{\bar{\sigma}\} = \bar{C}\{\bar{\varepsilon}\} \quad (2)$$

This form of tensor stiffness \bar{C} with inner arguments resulting from input parameters are related to mechanical properties of the RVE. The diversity of RVE that can be obtained when each argument in Equation (1) is changed recursively, a whole new \bar{C} is obtained. For this reason deep learning approach was applied in this work.

According to the general theory of homogenization,^[28–30] the average stress and strain in RVE are calculated by the following formula 3

$$\begin{cases} \bar{\sigma}_{ij} = \frac{1}{V} \int_V dV \sigma_{ij} \\ \bar{\varepsilon}_{ij} = \frac{1}{V} \int_V dV \varepsilon_{ij} \end{cases} \quad (3)$$

where V is the total volume of the RVE. $\bar{\sigma}_{ij}$ and $\bar{\varepsilon}_{ij}$ are the micro stress and strain tensor in terms of nodal displacements resulting from mesh discretization.

2.3. Finite Element Mesh Generation

Figure 2 illustrates the evolution of the material strain in (Z) direction S_{33} for different mesh values that has been run over in ABAQUS.

As the mesh density increases, the FEM solution will continue to change until an optimal threshold is reached (red dashed line) where no change in the calculated value occurs. Here the S_{33} that reaches a threshold value and remain constant, is a proof that the convergence is obtained.

The choice of hexahedral mesh is made based on two reasons. Firstly, this type of mesh allows to attain more precision on the deformation homogenization than with tetrahedral mesh. Secondly, the longer time FEM's calculation takes, the higher precision in learning will be reached. Therefore, the fixed mesh value is finely meshed corresponding to: 740 nodes, and 1480 hexahedral elements. Fibre part is finely meshed as well.

In FEM study, in order to reach convergence, the solution must be mesh size independent. For these reasons, a balance between fine mesh and time of calculus have to be found.

2.4. The Loading Conditions and Homogenized Properties

The structure has to be constrained while applying a unidirectional load. The next Figure 3 illustrates the six faces that will be constrained according to the applied load. For the case of tensile strength in Z direction, the back and the bottom faces, will only be constrained. Only the front face will be pulled. A displacement in form of constraint, fixed at 20% of the total size of the RVE (see Table 3).

The applied loading represents a strain of 7 MPa which is 10% of the nominal pressure of 70 MPa inside a type 4 hydrogen storage vessel.^[31] The constraint of enforcing edges will remain planar after deformation. The displacement of each pair of nodes on the parallel boundary surface of the RVE (see 3) are written according to the following formula

$$\begin{cases} u_i^A = \bar{\varepsilon}_{ik} x_k^A + u_i^* \\ u_i^B = \bar{\varepsilon}_{ik} x_k^B + u_i^* \end{cases} \quad (4)$$

A and B designate of parallel surface contacts where i is the i^{th} number of the pair nodes. Those paired nodes of surfaces in ABAQUS are referred when the geometry of the studied RVE is initially undeformed.

To ensure the symmetry of mesh nodes on the periodic plane, three partitions according to three symmetry plans (XY),(YZ),(ZX) are made. In this Equation (5) $\bar{\varepsilon}_{ik}$ corresponds to the average strain of the RVE and u_i^* is the displacement components of the periodic part. Therefore, it is possible to write the relative displacement between two nodes as follows^[18]

$$u_i^A - u_i^B = \bar{\varepsilon}_{ik} (x_k^A - x_k^B) = \bar{\varepsilon}_{ik} \Delta x_k \quad (5)$$

where Δx_k is the constant for each pair of boundary parallel surfaces (see 3). The previous equation is implemented by setting

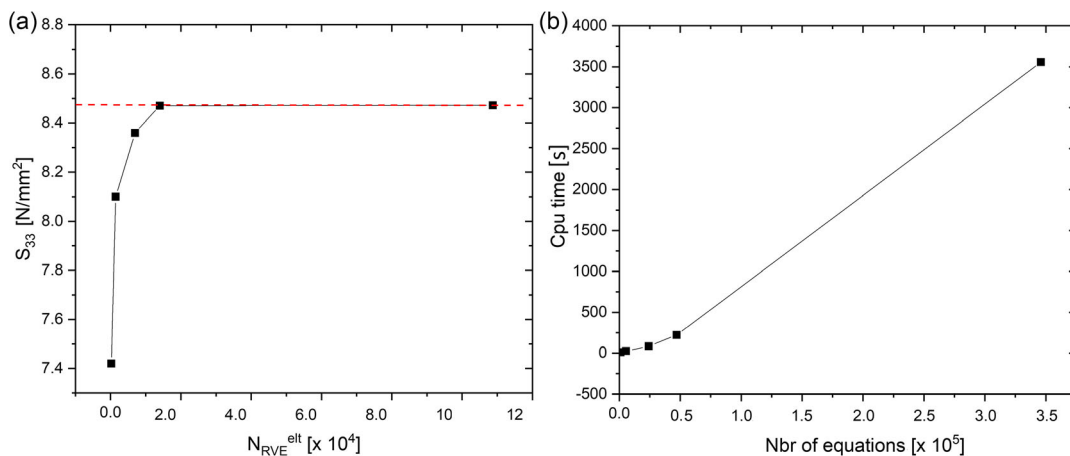


Figure 2. a) The dependence of FEM solutions on mesh density for strain values S_{33} for an RVE. b) Evolution of CPU time according to number of equations.

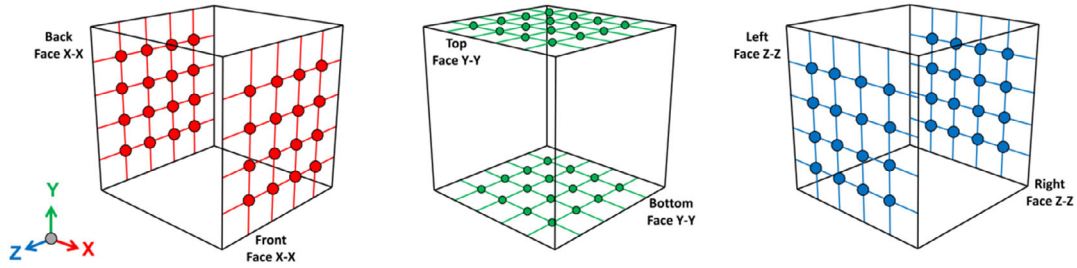


Figure 3. In red are represented the back and front faces. In green the top and bottom face. Then in blue color the left and right faces. The nodes are schematized as square grids.

Table 3. The multi-freedom constraint equations for nodal displacements of a 3-D RVE.

Case	Coefficients	Nodal set	DOF	load
Uni-axial Z	1	Z-front	3	20%
Uni-axial Z	-1	RP-set	3	20%

the linear constraint of the displacement between each pair of nodes on the parallel boundary surface. These boundary conditions guarantees continuous displacement but also tensile continuity on the parallel surfaces of the RVE. Each node of a given surface, must correspond to a node on the opposite surface. The six independent boundaries are calculated independently for each RVE.

2.5. Ductile Damage Model for Composites Materials under Solicitations

As the defined composite materials is displayed as elastoplastic behaviour, studies using similar type of materials^[32,33] have proposed an isotropic damage models to predict the sample damage evolution. Failure initiation, commonly occurs in the matrix inside composites, making them to be considered as subject to ductile-dominating failures. The void nucleation, growth, and coalescence^[34] are the foundation for developing and incorporating the damage model. The developed damage model^[35,36] assumes that the onset of damage $\bar{\epsilon}_D^{pl}$ is a function of stress triaxiality $\eta = -\frac{\sigma_m}{\sigma_{eq}}$ and the equivalent plastic strain rate $\bar{\epsilon}_0^{pl}$, as shown in Equation (6). Where σ_m is stress state hydrostatic component and, σ_{eq} is Huber-von Mises equivalent stress.

$$\bar{\epsilon}_D^{pl}(\eta, \bar{\epsilon}_0^{pl}) \quad (6)$$

The stress triaxiality η is further defined in the following Equation (7).^[36] Where $T = \sigma_{xx} + \sigma_{yy} + \sigma_{zz}$ is a trace of the stress tensor in each direction, which equals the sum of principal stresses. The stress triaxiality for uniaxial is fixed to $\eta = -0.333$ and the equivalent plastic strain rate $\bar{\epsilon}_0^{pl} = 0$, as the materials are strain-rate-independent.

$$\eta = -\frac{\frac{1}{3}\text{trace}(T)}{\sigma_{mises}} \quad (7)$$

To initiate damage, the equivalent plastic strain ($\bar{\epsilon}_{eq}^{pl}$) reaches a threshold value. This value is the fracture strain ($\bar{\epsilon}_0^{pl}$), where the variable that characterize damage D is initially $D = 0$. Here, the plastic deformation increases monotonically as the state variable named ω_D increases. For the damage to develop and increases ($\bar{\epsilon}_{eq}^{pl} > \bar{\epsilon}_0^{pl}$), the variable D increases from 0 to 1. At this maximum value, the material loses its load-carrying capacity for the equivalent plastic strain ($\bar{\epsilon}_{eq}^{pl} = \bar{\epsilon}_{finale}^{pl}$).

In the present study the damage evaluation law^[35,36] can also be defined using displacement at failure. The governing equation for equivalent plastic displacement is reported in the following equation

$$D = D(u^{pl}) \quad (8)$$

where u^{pl} is the effective plastic displacement. The linear evolution of the damage variable with effective plastic displacement is assumed, as it is expressed in Equation (8).^[36,37] The calculus of the degradation of the composite material used in the present study stopped automatically after reaching the fracture point. The displacement at failure (u^{pl}) is in fact fixed at 0.10. The value of 0.10. This value is significant because it represents the threshold beyond which the material undergoes substantial plastic deformation leading to failure. In ductile materials, a displacement at failure of 0.10 correspond to a threshold point where the material has accumulated sufficient strain to trigger extensive micro void coalescence, leading to macroscopic failure. The damage parameters are gathered in the following

Table 4

Using strain degradation as a variable in ABAQUS for time-series analysis of ductile damage is justified due to its physical significance in capturing the material's failure mechanisms, the advanced modeling capabilities of ABAQUS, and the predictive power of time-series analysis. This approach is expected to

Table 4. Damage parameters used to implement numerically the ductile damage of the composite RVE.^[62]

Fracture strain	Stress triaxiality	Strain rate
0.05	-0.333	0.1

enhance the understanding of material degradation but also provides practical tools for improving the safety and durability of critical structures like hydrogen storage vessels.

2.6. Results of FEM: as Input Dataset for the Neural Network

The results obtained from FEM will be used in DL neural network. **Figure 4**, shows the composite response under tensile solicitation obtained from FEM simulation. According to these results we notice that, the tensile strength solicitation causes localized damage in the matrix. At contrary, the fibre in center or corners are less affected (in green and blue). The fibre is the component that support all the stress with minimum deformation.

Figure 5 resumes the cumulative temporal evolution of the strain degradation variable SDEG from $T=0$ to $T=1$. The non-dimensional time for one RVE is accumulated over all nodes. The total simulation time is subdivided into 20 frames which correspond to a time step of $dT=0.05$. One can notice that initially when the displacement load is applied in Abaqus, most values of SDEG are null, then there are increasing levels of degradation with sharp transitions.

When a tensile strength in the direction of the fibre is applied, the strain is distributed on average equally on all nodes. With time, dynamical evolution of the strain degradation present the shape of a stair-step. At each step of the temporal discretization, when the degradation value is low at around 0.04 SI, the RVE is minimizing the strain over all the nodes. Then, with increasing deformation, the degradation is no more possible to redistribute over all nodes. As a consequence, more sharps steps occur, leading to higher degradation's values. Another way to visualize the strain degradation is, to draw the temporal evolution.

According to **Figure 6**, a monotonic increasing of strain degradation with time is reported. Initially, no degradation of the material is observed. Then, after a transient initially flat, a rising value is occurring until it reached the degradation limit. This curve can be seen as an addition of a linear part related to elasticity and several short peaks that reflect the nonlinear part related to plastic response to strain. On the left of the strain

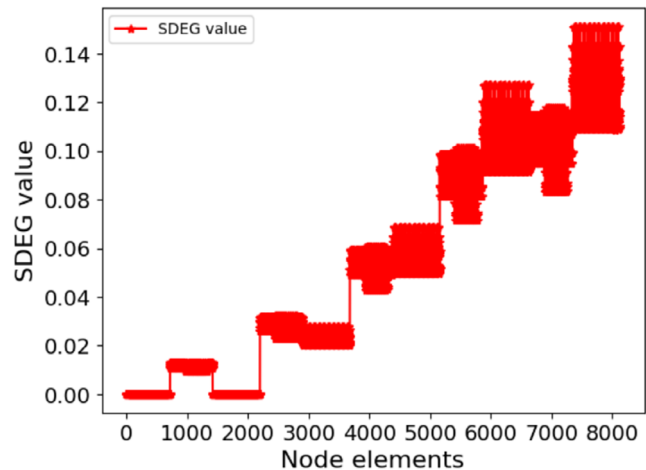


Figure 5. Evolution of the strain degradation during static load displacement of 20% for Z -tensile strength at each node of one RVE. The record is running over all the dimensionless time of loading $T_{\text{initial}}=0$ to $T_{\text{final}}=1$.

degradation curve is drawn the histogram of the occurrences of each value of strain degradation. The lowest values between 0 and 0.03 are the most present over all the nodes of the RVE. Later with advancing simulation's time, there is a slip from low values of SDEG to higher ones. This effect is underlined by the kernel density estimation (kde) curve.^[38] Here, we can see that the initial narrow peak at 0.025 is moved to higher values of $SDEG=0.10$ with a larger spread.

This visualization allows to identify key phases in the degradation process, such as the onset of significant damage, the progression of degradation. It provides insights into both the elastic and plastic behavior of materials and helps in predicting the material's lifespan under loading conditions. For critical components as hydrogen storage vessels, understanding the time-dependent degradation of materials is crucial. The SDEG versus time curve provides a direct way to assess how quickly a material might fail, the previous curves 6 reveal critical points where degradation accelerates, indicating the transition from elastic to plastic behavior.

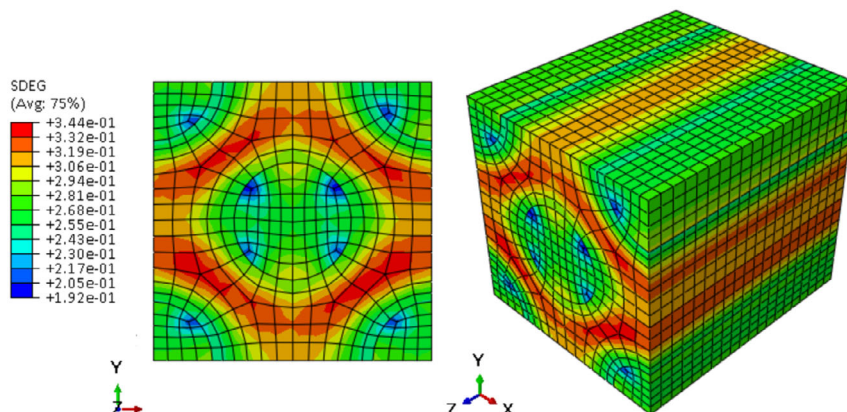


Figure 4. The Z -tensile strength solicitation's effect on degradation realized by the homogenization method on one REV ($F_v=35\%$, Fiber = aramid, Matrix = PA12) at last temporal frame $T=1.0$. The overall scalar stiffness degradation is represented in color map.

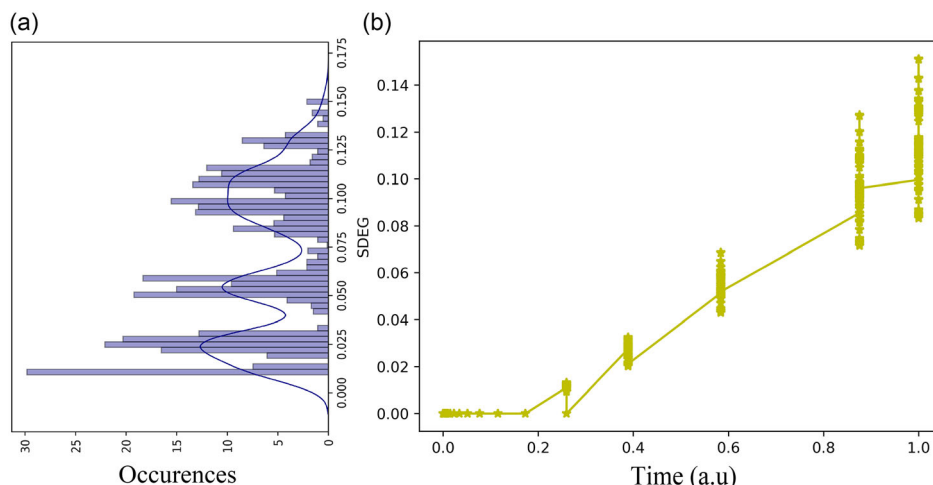


Figure 6. In a) Histogram of strain degradation occurrences. In blue is drawn the kde curve fitting. In b) is drawn the graphical representation of the temporal evolution of the strain degradation during static load displacement of 20% for *Z*-tensile strength from the adimensional time $T = 0$ to $T = 1$.

3. Classical Machine Learning Prediction of Degradation as a Time Series

In this study, the degradation of strain under applied tensile strength is treated as a time series. The temporal evolution of SDEG is forecasted using three classical algorithms^[39]: extreme gradient boosting (XGBoost),^[40] random forest (RF),^[41] and support vector regressor (SVR) with a radial basis function as a kernel.^[42]

Time series data is transformed into a supervised learning problem, allowing the model to be trained on past values to predict future ones. This transformation involves using a sliding window, shifting the window of inputs and expected outputs forward through time to create new samples for the supervised learning model. The precision of forecasting depends on the number of observations recorded for a given time in a time-series dataset, and traditionally, two types of datasets are distinguished:^[43] 1) Univariate time series: These datasets involve observing only a single variable at each time, such as hourly temperature. In our case, the univariate time-series dataset is the SDEG; 2) Multivariate time series: In these datasets, two or more variables are observed at each time. Here, two features are selected using the GMDH feature selection algorithm, discussed in the next section. The chosen observables are the time step and the component of stress in the X direction, referred to as S.S11.

The model must be trained on past values to predict future ones, leading to the utilization of walk-forward validation.^[44,45] This validation technique involves splitting the dataset into training and test sets by selecting a cut point. All data, except the last

100 time steps, are used for training, and the final 100 time steps are reserved for testing.

In making a one-step forecast, predicting one-time step into the future, the model is evaluated by training on the training dataset and predicting the first step in the test dataset. Subsequently, the actual observation from the test set is added to the training dataset, and the model is refitted. This process is repeated for the entire test dataset, yielding a one-step prediction for the entire test dataset, from which an error measure can be calculated to evaluate model performance (see **Figure 7**).

In this implementation, only the preceding 200 time steps are utilized as input to the model, with default values for hyperparameters. The loss function is modified to the squared error to prevent warning messages, and a total of 1000 trees are included in the ensemble to avoid under-learning. **Table 5** provides a summary of the hyperparameter values for the XGBoost, RF, and SVR models.

The next step involves using the model to make predictions on new SDEG data. This process, known as an out-of-sample forecast, extends the predictions beyond the training dataset. Model performance is evaluated using the mean absolute error (MAE) metric. The strain degradation's temporal evolution is illustrated in **Figure 8**, depicting the FEM simulation time in red and the 200 forecasted values from the iterative XGBoost prediction in blue crosses.

This visual representation indicates a promising forecasting trend. Beyond the dimensionless time $T = 1$, the XGBoost recursive forecasting captures the last high values of strain degradation. Shortly after $T = 1.0$, the forecasted SDEG values exhibit a



Figure 7. Illustration of the idea of walk forward validation on time-series data with sliding window.

Table 5. Sum up of the hyper parameters values used for the recursive XGBoost, RF, SVR forecasting algorithms for the SDEG time's series.

Model	Hyperparameters			
XGBoost	Metric	$N_{estimators}$		
	Squared error	1000		
RF	Metric	$N_{estimators}$		
	Squared error	1000		
SVR	Kernel	γ	C	ϵ
	rbf	0.5	10	0.05

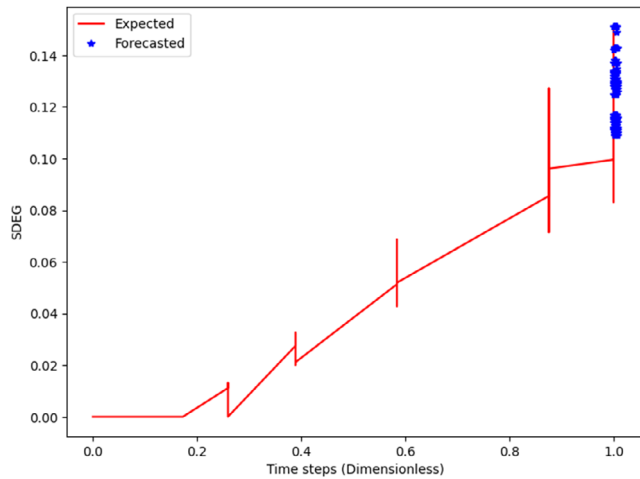


Figure 8. Superposition of temporal evolution of expected strain degradation (red line) and forecasted values.

peak (blue star dots), reproducing the nonlinearity of the last peak. Subsequently, more regular values are obtained over time.

To compare the forecasting results across multiple ML algorithms, various metrics are employed (see Table 6). Notably, the R^2 score metric identifies the RF model as the best performer in forecasting SDEG values, a result consistent with the known tendency of RF to overfit. However, XGBoost and SVR (RBF) exhibit comparable performance, particularly in terms of the MAE metric over 20 time-series split cross-validation. The SVR model with a radial basis function kernel demonstrates slightly lower performance.

Overall, except for the RF model, XGBoost emerges as the best-performing model based on the MSE, RMSE, and MAE metrics. In addition to those supervised method, let's present

Table 6. Metrics values obtained from the classical ML algorithm for forecasting over strain degradation values SDEG with time-series split cross validation.

Model name	R^2	MAE	MAPE	RMSE
XGBoost	0.965	-0.0007	-0.0269	-0.0055
RF	0.989	0.001	0.005	0.001
SVR(rbf)	-3.891	0.023	0.173	0.027

a statistical method as auto-regressive integrated moving average (ARIMA) modeling method.^[46]

The ARIMA algorithm operates on the premise that information gleaned from past values of a time series can singularly suffice to predict future values. The model elucidates a given time series based on its historical values, encompassing its own lags and lagged forecast errors. The ARIMA equation used for forecasting future values is articulated as follows

$$Y_t^{ARIMA} = \underbrace{\alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \dots + \beta_p Y_{t-p}}_{Y_t^{AR}} \epsilon_t + \underbrace{\phi_1 \epsilon_{t-1} + \dots + \phi_q \epsilon_{t-q}}_{Y_t^{MA}} \quad (9)$$

where: Y_t is the value of the time series at time t ; α is the constant term; β_p are the coefficients of the autoregressive terms; ϕ_q are the coefficients of the moving average terms and ϵ is the white noise error term.

Here, the predicted variable Y_t^{ARIMA} comprises auto-regressive (AR) terms denoted as Y_t^{AR} and, moving average (MA) terms denoted as Y_t^{MA} . Where α is a constant and can be interpreted as the mean value of the series. It represents the level around which the time series fluctuates. The order of the AR term denoted as p and the order of the MA term denoted as q . The AR term Y_t^{AR} relies solely on its own lags, that is, $Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}$. Conversely, the MA term Y_t^{MA} hinges on lagged forecast errors, that is, $\epsilon_{t-1}, \dots, \epsilon_{t-q}$. The initial step involves differentiation to render the time-series stationary.

In time-series cross-validation, several steps back in time are taken, forecasting into the future for each step, followed by a comparison against the actual values. The training and testing dataset of the time series is partitioned into a 75:25 ratio, preserving the time series' order for forecasting. A stepwise approach explores multiple combinations of p, d, q parameters to determine the optimal model with the least Akaike Information Criteria (AIC). A lower AIC indicates a better model fit.

The optimized values of p, d, q for the best ARIMA model, obtained by minimizing the AIC criteria, are summarized in Table 7.

Figure 9 illustrates the superposition of the temporal evolution of strain degradation's actual values (in blue) and forecasted values (in green) obtained by the ARIMA method. The gray shaded area represents the minimum and maximum predicted values. Several observations arise from Figure 9. Firstly, the model exhibits a commendable prediction of the general trend of the degradation curve, accurately predicting high degradation values beyond the dimensionless time $T=1.0$ over 1000 time steps. Secondly, there is an absence of sharp transitions in values. The inset showcases the initial hundred values, depicting a gradual decrease in strain degradation over time before reaching a plateau. This behavior corresponds to a redistributed strain

Table 7. Optimized value of the best ARIMA model used for forecasting. The values p, d, q are obtained by minimizing the AIC criteria.

ARIMA parameters	p	q	d
ARIMA	2	1	3

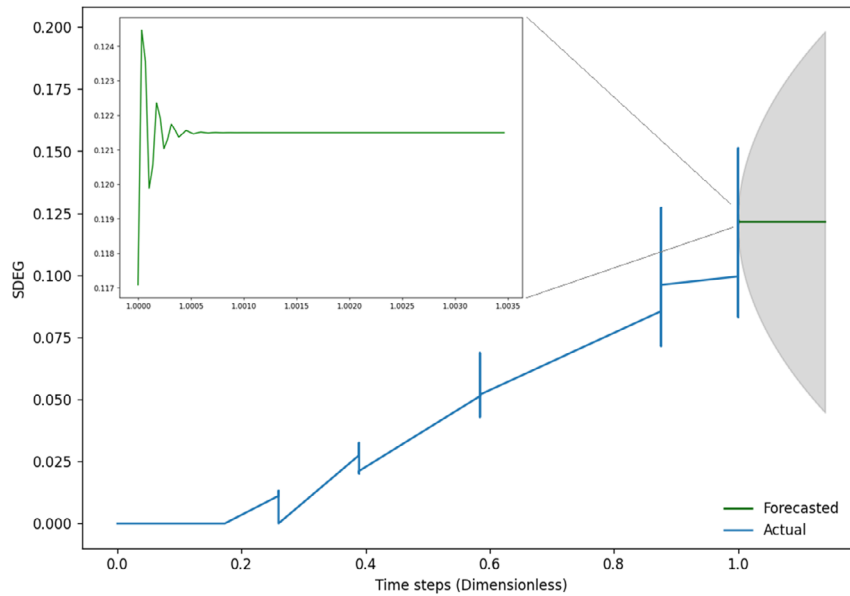


Figure 9. Graph superposition of temporal evolution of strain degradation SDEG of actual values (in blue) and forecasted values (in green) obtained by ARIMA method. The gray shaded area represent minimum and maximal values.

across each node of the RVE. Lastly, the large spread (in gray shade) of the predicted minimum and maximum values of strain degradation curtails the scope of long-term predictions.

In the context of modeling temporal damage evolution in RVEs, a comprehensive evaluation reveals the efficiency of classical ML methods, namely XGBoost, RF, and SVR(rbf), when applied recursively. These classical models demonstrate a commendable ability to generate accurate predictions for strain degradation. In stark contrast, the long short-term memory (LSTM) network exhibits a notably lower accuracy, especially when no hypertuning is operated, hovering around 25%. This outcome aligns with a broader observation,^[47] emphasizing the superior performance of ML methods over DL methods, particularly in short-term forecasted periods.

It is crucial to highlight that both DL and ML predictions exhibit diminished precision beyond peak values. This diminished accuracy in forecasting post-peaks may be attributed to the inherent nonlinearity affecting certain nodes in the material during specific loading conditions. This complexity poses a significant challenge in accurately simulating damage in composite materials.

Furthermore, it is noteworthy that ML algorithms, exemplified by XGBoost, outperform DL methods, such as LSTM, in reproducing elastoplastic behavior. This observation underscores the advantage of classical ML approaches in capturing the intricacies of material response, particularly in scenarios involving complex elastic and plastic deformations.

4. Deep Learning Architecture

DL architecture, a subset of ML, emulates the structure of a biologically inspired ANN.^[48] This intricate system is defined by a layered arrangement of interconnected neurons, combining various algorithms to model and process highly complex nonlinear relationships (refer to **Figure 10** for a visual representation).

The foundation of neural networks lies in the multitude of artificial neurons organized into layers. These layers, including the input layer, hidden layers, and the output layer, facilitate communication only with the immediately preceding and following layers. Information traverses through the ANN from the input layer, where external data is received, to the output layer, which produces the final result. The number of hidden layers is contingent upon the complexity of the problem at hand. Within each hidden layer, neurons receive input signals, process them by combining with their internal state, and generate output signals.

Each neuron boasts multiple input and output connections, assigned weights that collectively form the layer of the ANN. The learning process involves network adaptation, specifically the adjustment of connection weights, aiming to minimize observed errors in the network's output.

Given its versatile and self-adapting architecture, RNN emerges as a specialized class of neural networks designed to handle temporal data. RNN neurons are equipped with a cell state/memory, and input processing aligns with this internal state, facilitated by loops within the network. To address the limitation of short-term memory in RNNs, LSTM models are introduced.

Prior to applying RNN to data, a crucial preprocessing step involves refining the data through feature selection. This meticulous process enhances the effectiveness of the neural network in extracting meaningful patterns and insights.

4.1. Data PreProcessing: Use of the Group Method of Data Handling (GMDH) for Feature Selection

In the late 1960s, Ivakhnenko introduced a groundbreaking mathematical algorithm designed to tackle intricate and nonlinear problems. Termed the group method for data handling (GMDH) algorithm,^[49] this solution provided a self-contained

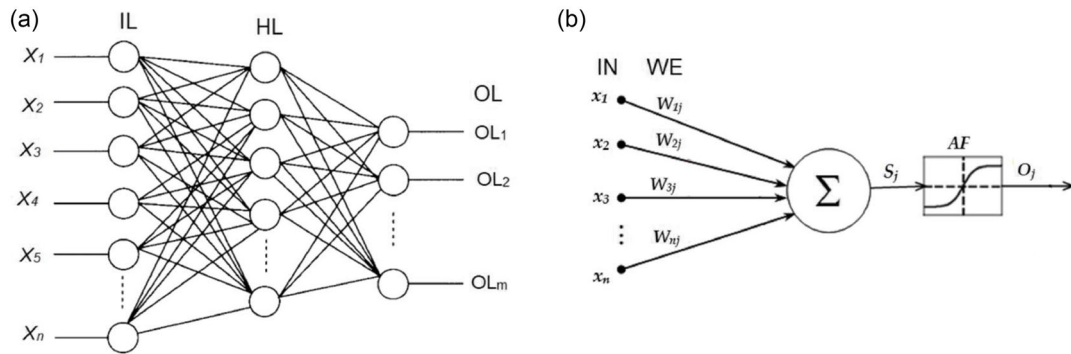


Figure 10. Schematic representation of artificial neural network featuring several hidden layers of neurons between the input and output layers. a) represent input layer (IL), hidden layer (HL) and each node fully connected to the output layer (OL). b) Individual node schematize set of function to estimate the weight (WE) and bias for each node. The final value is obtained through the activation function (AF) adapted from.^[59]

model capable of addressing both classification and regression problems. GMDH operates in diverse fields such as data mining, knowledge discovery, optimization, and pattern recognition.^[50] Its predictive capabilities have found applications in chemistry,^[51] environmental studies,^[52] and civil engineering.^[53] Notably, GMDH algorithms offer the automation of feature selection for datasets.

The GMDH algorithm excels in automatically identifying interrelations in data, selecting optimal model structures or networks, and enhancing the accuracy of existing algorithms. By minimizing external operator influence, the computer autonomously determines the optimal model structure or governing laws within a system.

The mapping of input to output variables is orchestrated by the GMDH neural network, functioning as a nonlinear function known as the Volterra series. The Equation (10) captures this relationship, illustrating the complexity of the mapping

$$\gamma = a_0 + \sum_{i=1}^m a_i x_i + \sum_{i=1}^m \sum_{j=1}^m a_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m a_{ijk} x_i x_j x_k \quad (10)$$

where $x(x_1, x_2, \dots, x_m)$ represents the input vector with extracted features, $a(a_1, a_2, \dots, a_m)$ symbolizes the coefficients or weights vector, and γ denotes the network output. The GMDH algorithm aims to unravel the unknown coefficients a_i in this complex equation.

The Volterra series, expressed as a two-variable second-degree polynomial in Equation (11), provides a more comprehensible form for understanding

$$G_{ij}(x_i, x_j) = a_0 + a_1 x_i + a_2 x_j + a_3 x_i^2 + a_4 x_j^2 + a_5 x_i x_j \quad (11)$$

To determine the a_i coefficients, regression methods are employed for each pair of x_i and x_j input variables. The optimization of coefficients involves minimizing the squared error E_L in Equation (12), ensuring the model's fidelity to the data.^[54,55]

$$E_L = \sum_{i=1}^m (y_i - G_L)^2 \quad (12)$$

The GmdhPy package,^[56] a freely available tool, facilitates feature selection. The self-organized DL polynomial neural

network, as depicted in **Figure 11**, showcases the parameters autonomously explored by GMDH during calculations.

From this representation, it's evident that only one feature has been retained. For external traction solicitation, two hidden layers were employed to regress the output values of the scalar stiffness degradation variable SDEG.

To validate the GMDH results, MSE, RMSE, and R-squared (R^2) metrics are employed, as summarized in **Table 8**.

4.2. The Structure of the LSTM Model for Deep Learning

When dissecting a symphony, isolating each note may initially result in a cacophony of sounds, rendering individual notes unclear. The appreciation of each note relies on the context provided by previous ones, demonstrating the persistence of auditory information. RNN address this temporal dependency

Self-organizing deep learning polynomial neural network

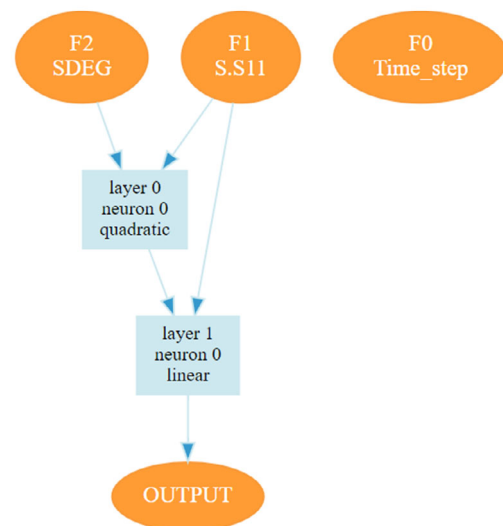


Figure 11. Representation of the self-organized DL polynomial neural network. Only one main features inside orange ellipse have been selected to predict the strain degradation SDEG values for traction in Z direction.

Table 8. Metrics values obtained from the GMDH feature selection for regression over strain degradation values SDEG for tensile strain.

Metrics	RMSE	MSE	MAE	R ²
values	7.2×10^{-7}	5.2×10^{-13}	5.9×10^{-13}	0.999

challenge by incorporating the temporal variable underlying degradation data. These networks, featuring internal loops, enable information to persist over time.

LSTMs belong to the RNN family and excel in learning long-term dependencies. **Figure 12** illustrates the LSTM architecture, where pink circles represent pointwise operations, yellow boxes denote learned neural network layers, and lines symbolize concatenation or copying of content. Each rectangle signifies a cell state C , and gates, comprising sigmoid neural net layers and point wise multiplication operations, regulate the addition or removal of information from the cell state.

The dataset comprises 36 observable variables from the FEM output database, with 28 120 rows representing 20 frames for all discretized elements. For forecasting degradation values, two approaches—multivariate and univariate time series—are employed. To prepare the data for LSTM architecture, two key concepts are introduced: 1) Horizon: The number of time steps to predict in the future; and 2) Window: The number of time steps from the past used to predict horizon values.

The LSTM model parameters for both univariate and multivariate time series are summarized in the **Table 9** below

The predictive performance of the LSTM model is enhanced using the rectified linear unit (ReLU) activation function, introducing nonlinearity. The ReLU function is described in Equation (13). The performance of the LSTM neural network is evaluated using four metrics: RMSE, MAE, MAPE, and accuracy.

$$\text{ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (13)$$

Table 9. Table reporting horizon and windows size values for both univariate and multivariate approaches for LSTM.

Model Nbr	Model type	Horizon size	Window size
Univariate	LSTM	100	100
Multivariate	LSTM	100	100

Table 10. Table Summing up the different ANN: LSTM, Bi-LSTM, Bi-LSTM-Attention, tested to predict the degradation value.

ANN	Hidden layer	Nbr of neurons Hidden(l)	Nbr of neurons Dense(l)	Dropout rate	Tot parameters
LSTM	3	300	100	0.2	1.539.007
Bi-LSTM	3	300	100	0.2	4.317.957
Bi-LSTM Attention	1	300	100	0.2	850.957

The LSTM neural network used in this study is made of one input layer of 300 neurons of the range of the size of the window of time's series that is trained on. Then there are three hidden layers of 300 cells. The last output layers are of type dense, which means all nodes of the network are connected. The last layer is made of the same number as the number of future points it is supposed to predict. Here, future points are called Horizon.

In the following **Table 10** is summed up the detailed architecture of the ANNs that was implemented for the forecasting.

4.3. Results

This section presents the results for both multivariate and univariate models (see **Figure 13** and **14**) aimed at predicting the temporal evolution of strain degradation under tensile strength in the Z direction.

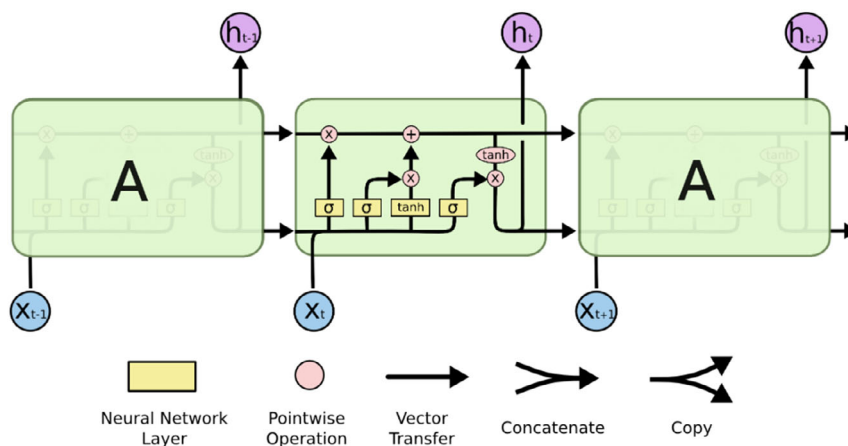


Figure 12. Schematic representation of a DL network. LSTM type of ANN featuring several hidden layers of neurons between the input and output layers. Adapted from.^[60]

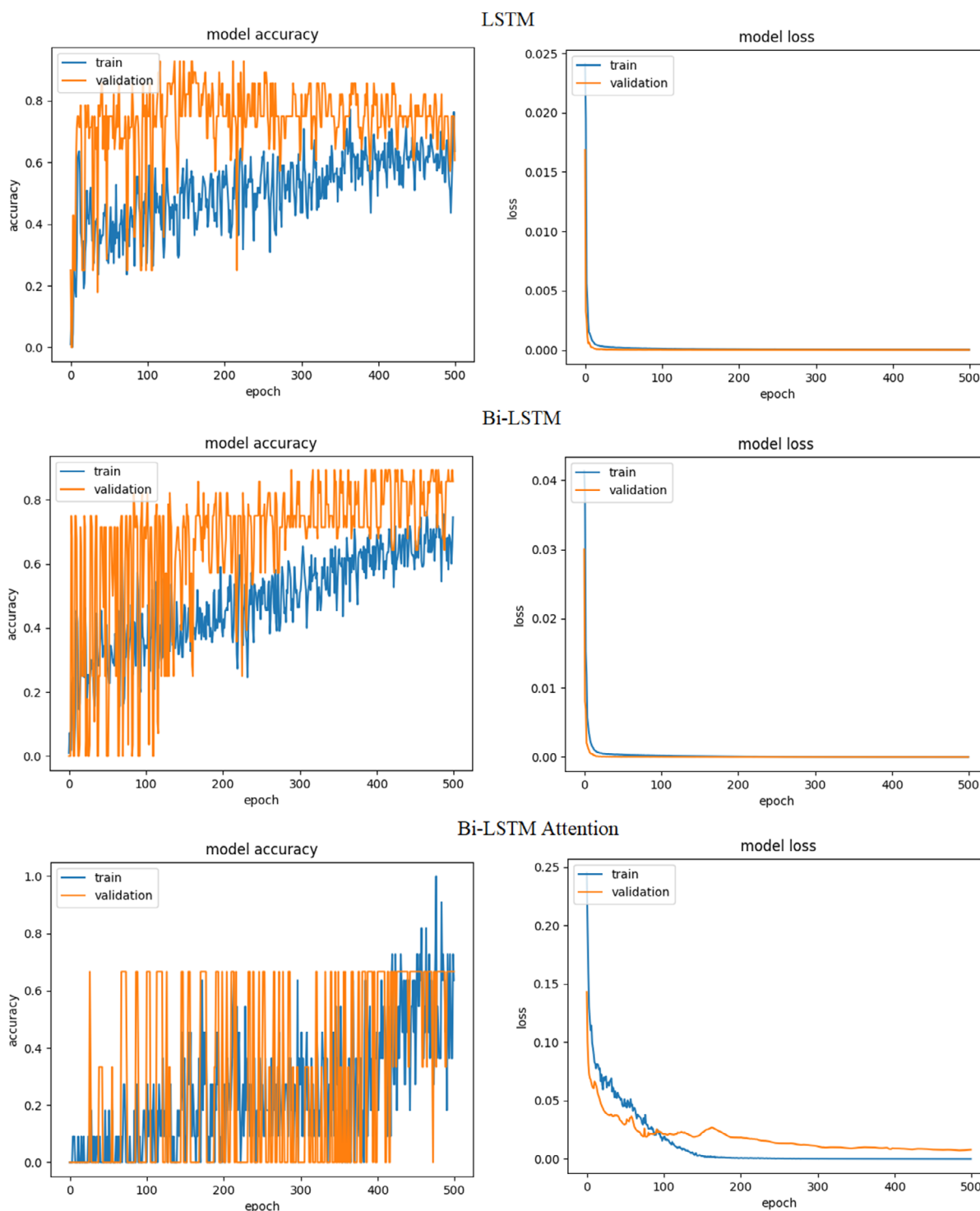


Figure 13. Graphs representing the evolution of accuracy metric and loss function according to the number of epochs for LSTM, Bi-LSTM, Bi-LSTM-attention ANNs for multivariate approach.

4.3.1. Multivariate Model

The multivariate model retains all features, with strain degradation as the dependent variable for prediction. Three

models—LSTM, Bi-LSTM, and Bi-LSTM-attention—are compared 13. For both LSTM and Bi-LSTM models, the accuracy metric reached 0.8 for the validation datasets, showing a clear increasing trend for both learning curves of training and

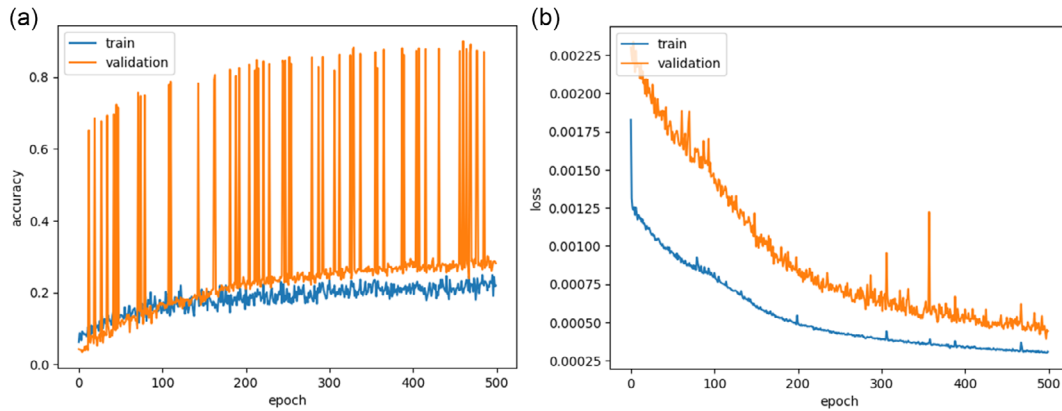


Figure 14. Learning curves for the univariate model over 500 epochs. a) Evolution of accuracy for the training and validation sets and b) Variation of the loss function for the training and validation sets.

validation datasets. However, the Bi-LSTM model with an attention layer struggled to reach a metric accuracy value of 0.7, with a less distinct monotonous increasing trend.

The comparison of accuracy curves across epochs reveals a common noisy behavior, attributed to the global nature of the strain degradation curve (Figure 5), which the models attempt to learn across multiple epochs. The curve exhibits linear segments due to elasticity features and piece-wise scaled functions representing nonlinear local responses of the composite material to external tensile strength.

4.3.2. Univariate Model

The univariate model retains only two columns: the first representing discretized time ($T = 0$ to $T = 1$), and the second containing strain degradation (SDEG) values. Figure 14 highlights the importance of the accuracy metric. The learning curves demonstrate LSTM's improvement until stabilization after the 200th epoch at a value of 0.25. Notably, a leap in accuracy occurs from 0.25 to around 0.80. This leap may be attributed to linear evolution in the validation set for specific nodes, while others exhibit nonlinear behavior, potentially causing accuracy fluctuations.

The loss function's classical evolution is depicted in Figure 14b, where both train and validation curves converge and reach a minimum value. **Table 11** summarizes the metric values for the three models.

Table 11. Table summing up the metrics values for the three DL models LSTM, Bi-LSTM, and Bi-LSTM with attention layer. The Bi-LSTM model is showing the best metric's value.

Models	Accuracy	MAE	MSE	RMSE
LSTM	0.61213	0.0016	2.635×10^{-6}	0.041
Bi-LSTM	0.8199	0.0063	0.0006	0.079
Bi-LSTM-Attention	0.6097	0.0025	3.481×10^{-6}	0.050

4.3.3. Results Analysis

Interestingly, the LSTM and Bi-LSTM-attention models exhibit lower error metrics, with LSTM showing an MAE of 0.0016, MSE of 2.635×10^{-6} , and RMSE of 0.041, and Bi-LSTM-attention showing an MAE of 0.0025, MSE of 3.481×10^{-6} , and RMSE of 0.050. These lower error values suggest that these models are effective in minimizing general prediction errors. However, despite their strong performance in these metrics, both LSTM and Bi-LSTM-attention struggle to accurately predict spikes or sudden changes in the strain degradation curve.

This limitation is particularly crucial because accurately capturing these spikes is essential for modeling the non-linear, abrupt changes often observed in material degradation. The Bi-LSTM model, although having slightly higher error metrics, manages to capture these temporal spikes more effectively, resulting in a higher overall accuracy (0.8199) see Table 11. This outcome underscores the importance of not only focusing on error metrics like MAE, MSE, and RMSE but also ensuring the model's capability to predict critical events in the data, which in this case, are the spikes in strain degradation (**Figure 15**).

4.3.4. Discussions

The multivariate model's performance indicates that LSTM and Bi-LSTM models outperform the Bi-LSTM-attention model in predicting strain degradation. Similarly to ref. [13], our findings reinforce that recurrent networks, particularly LSTM-based architectures, are highly effective for solving time-series problems. These models excel in capturing both linear and nonlinear temporal dependencies, which are critical in accurately modeling the complex behavior of materials under stress. The noisy behavior observed in the accuracy curves (see Figure 14) can be attributed to the inherent complexity of the strain degradation curve, which involves both linear and nonlinear contributions. Despite the noise, the Bi-LSTM model showed superior performance, particularly in scenarios where understanding the temporal sequence of material degradation is crucial.

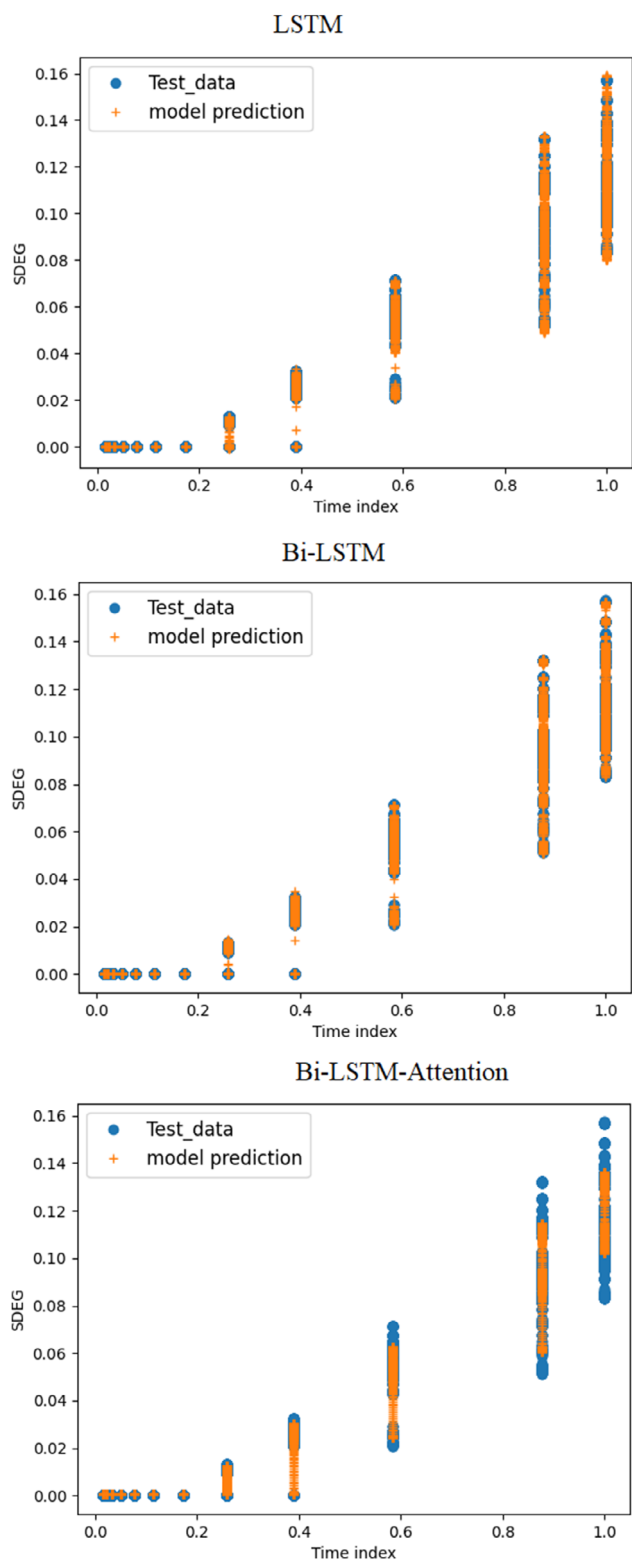


Figure 15. Graphs representing the prediction results for three different models: LSTM, Bi-LSTM, Bi-LSTM-attention.

In the univariate model, the observed leaps in accuracy suggest that the LSTM model effectively learns from the training dataset. However, challenges arise when predicting the nonlinear behavior of specific nodes in the validation set. This issue highlights a common limitation of LSTM models when applied to highly nonlinear time-series data, where the model might overfit to the training data and struggle with generalization. Despite this, the convergence of the loss function indicates effective model training, achieving a minimum value that signifies the model's capacity to learn the overall trends within the data. These findings underscore the potential of LSTM models in capturing temporal dependencies and predicting strain degradation, thereby paving the way for improved simulations in composite material engineering.

Our results also demonstrate that the Bi-LSTM model significantly outperforms traditional ML methods such as RF and EGB in predicting the temporal evolution of ductile damage. This advantage is primarily due to the sequential nature of the Bi-LSTM network, which is inherently well-suited for time-series data. Understanding the sequence of degradation during tensile strength is crucial for accurate predictions, and the Bi-LSTM model's ability to process information in both directions (past and future) enhances its predictive accuracy. The model's performance indicates that it can be a reliable tool for forecasting material behavior, especially in scenarios where the history of strain impacts future material states.

The addition of an attention layer to the Bi-LSTM model, while theoretically advantageous, did not result in superior performance in our study. The Bi-LSTM-Attention model exhibited slightly lower accuracy compared to the standard Bi-LSTM, which raises important considerations regarding the integration of attention mechanisms in sequence modeling for material behavior prediction. Although the attention mechanism is designed to enhance the model's focus on the most critical parts of the input sequence, it may have been less effective in this particular context due to the relatively homogeneous nature of the strain data or the specific tuning parameters used. This result suggests that while attention layers can provide benefits, their effectiveness may depend heavily on the complexity and heterogeneity of the dataset.

These conclusions are supported by findings in recent literature. For instance, Mozaffar et al.^[57] demonstrated that DL models, specifically RNNs, could effectively predict path-dependent plasticity in materials by capturing complex temporal sequences without relying on traditional plasticity theories, which often involve iterative solution schemes. Their work highlights the strength of sequence learning in materials modeling, providing a strong foundation for the use of Bi-LSTM networks in similar applications. Mozaffar et al. explored the application of ML to predict mechanical behavior in composite materials, emphasizing the need for models that can account for nonlinear, history-dependent behavior—a challenge that Bi-LSTM models are well-suited to address.

Integrating physical principles into machine learning models through approaches like physics-informed neural networks

(PINNs), as proposed by Cuomo et al.^[58] could enhance both model interpretability and accuracy. PINNs incorporate known physical laws directly into the training process of the model, which can help to guide the learning process and ensure that the model's predictions are physically plausible. This approach is particularly beneficial in fields like material science, where understanding the underlying physical phenomena is crucial for accurate modeling and prediction. By incorporating such physical constraints, future models could not only improve prediction accuracy but also provide more interpretable results that align with established scientific knowledge.

Overall, these results emphasize the considerable potential of LSTM-based models, particularly Bi-LSTM architectures, in capturing the temporal dependencies inherent in material behavior. However, the study also highlights the importance of continued exploration and refinement of advanced models and techniques, such as attention mechanisms, transformer models, and physics-informed approaches, to address the challenges posed by complex, nonlinear time-series data. These advancements could pave the way for more accurate and reliable simulations in composite material engineering and beyond, contributing to better predictive models for a wide range of applications in engineering and science.

5. Conclusion

The novelty of this research lies in the innovative coupling of FEM with advanced ML and DL techniques to predict ductile damage in composite materials used in type IV hydrogen storage tanks. Unlike traditional approaches that rely solely on experimental or purely computational methods, this study integrates FEM results with ANNs, specifically utilizing Bi-LSTM networks. This hybrid methodology not only enhances the accuracy of damage predictions but also significantly reduces the computational cost and time associated with material degradation studies. By employing the GMDH for feature selection and leveraging both classical ML algorithms and sophisticated DL models, the research provides a robust framework for forecasting the temporal evolution of strain degradation. This original approach offers a scalable and efficient tool for optimizing the design and durability of composite materials in hydrogen storage applications. Key findings from this research are:

Hybrid modeling approach by coupling FEM with ML/DL models, particularly the Bi-LSTM network, we achieved a comprehensive method for forecasting the degradation of composite materials. This hybrid approach significantly reduces computational costs and time compared to traditional experimental methods.

Feature Selection Using GMDH: the GMDH algorithm proved to be an effective tool for selecting the most relevant features from the FEM output, enhancing the predictive capabilities of the ML/DL models.

Comparison of ML Algorithms: the study compared the performance of various ML algorithms, including XGBoost, RF, and Support vector regression (SVR), in predicting strain degradation. XGBoost emerged as the most promising model, demonstrating the potential of classical ML methods in forecasting material behavior.

The Bi-LSTM network outperformed other DL models, including the standard LSTM and Bi-LSTM with attention layers, in predicting strain degradation. The Bi-LSTM model achieved an accuracy of 0.82, highlighting its effectiveness in capturing the complex temporal dependencies in the degradation data.

The ML/DL models were successful in reproducing both the elastic and plastic behaviors of the composite materials under tensile strength, providing valuable insights into the material's performance and degradation over time.

The findings demonstrate the scalability of the proposed hybrid approach for real-world applications, offering a robust framework for optimizing the design and durability of composite materials in hydrogen storage vessels.

As a perspective, by using transfer learning of weights on the last frozen layer it will be possible to use this layer onto real-world 3D composite material to forecast degradation's temporal evolution. Fine tuning of the Bi-LSTM model to optimize the parameters of learning throughout a pipeline approach seems to be the next step of improvement. In fact, once the best optimized model obtained, it will be possible to imagine an *ersatz* of a combination of a linear model to mimic the elastic behaviour and a stochastic continuous model^[11,12] like a Poisson law to mimic the plastic behaviour particularly the pics of strain degradation at nodes.

Acknowledgements

The authors would like to thank the DeVinci Research center (DVRC) for their implication and numerical resources provided.

Data Availability Statement

Research data are not shared.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

bidirectional (Bi-LSTM), degradation, ductile damage, general method of data handling, hydrogen storage vessel, long short term memory, recurrent neural network

Received: May 23, 2024
Revised: September 2, 2024
Published online:

- [1] J. Wang, S. Karimi, P. Zeinalzad, J. Zhang, Z. Gong, *Constr. Build. Mater.* **2024**, 438, 137264.
- [2] M. Liu, H. Li, H. Zhou, H. Zhang, G. Huang, *Compos. Commun.* **2024**, 49, 101988.
- [3] S. Samy, C. Abdelrahman, M. Shazly, *Heliyon* **2024**, 10, e33681.
- [4] S. Cassola, M. Duhovic, T. Schmidt, D. May, *Composites, Part B* **2022**, 246, 110208.
- [5] X. Ding, X. Hou, M. Xia, Y. Ismail, J. Ye, *Compos. Struct.* **2022**, 302, 116248.
- [6] K. Zhang, L.-H. Ma, Z.-Z. Song, H. Gao, W. Zhou, J. Liu, R. Tao, *Compos. Struct.* **2022**, 296, 115835.

- [7] H. Garmestani, *Int. J. Plast.* **2006**, 22, 1367.
- [8] A. A. Stepashkin, S. Chavhan, S. V. Gromov, A. Khanna, V. V. Tcherdyntsev, D. Gupta, H. Mohammad, E. V. Medvedeva, N. Gupta, S. S. Alexandrova, *Alexandria Eng. J.* **2023**, 82, 218.
- [9] J. Bonnar anak Jores, E. Jayamani, T. Sie Ming Lai, J. Subramanian, C. Rejeesh, *Mater. Today: Proc.* **2023**, <https://doi.org/10.1016/j.matpr.2023.08.351>.
- [10] Z. Li, X. Li, Y. Chen, C. Zhang, *Compos. Struct.* **2023**, 323, 117473.
- [11] D. Shi, X. Teng, X. Jing, S. Lyu, X. Yang, *Compos. Struct.* **2020**, 248, 112549.
- [12] H. Choi, S. Jung, C. Zhang, G. J. Yun, *Mech. Adv. Mater. Struct.* **2022**, 29, 633.
- [13] Z. Song, Y. Feng, C. Lu, J. Liu, W. Pan, *Compos. Sci. Technol.* **2024**, 248, 110414.
- [14] L. Zhang, Z. Wang, L. Wang, Z. Zhang, X. Chen, L. Meng, *Digital Commun. Networks* **2021**, 7, 551.
- [15] L. Luo, B. Zhang, G. Zhang, X. Li, X. Fang, W. Li, Z. Zhang, *Polym. Adv. Technol.* **2021**, 32, 1049.
- [16] E. Pitz, K. Pochiraju, *Eng. Appl. Artif. Intell.* **2024**, 134, 108622.
- [17] E. Ghane, M. Fagerström, M. Mirkhalaf, *Eur. J. Mech. A* **2024**, 107, 105378.
- [18] M. Okereke, S. Keates, *Finite Element Applications: A Practical Guide to the FEM Process*, Springer International Publishing, Switzerland **2018**.
- [19] W. Song, V. Krishnaswamy, R. V. Pucha, *Compos. Struct.* **2016**, 137, 9.
- [20] B. K. S. Bargmann, *Prog. Mater. Sci.* **2018**, 96, 322.
- [21] K. Terada, M. Hori, T. Kyoya, N. Kikuchi, *Int. J. Solids Struct.* **2000**, 37, 2285.
- [22] S. Firooz, S. Saeb, G. Chatzigeorgiou, *Math. Mech. Solids* **2019**, 24, 2961.
- [23] C. Heinrich, M. Aldridge, A. S. Wineman, J. Kieffer, A. M. Waas, K. Shahwan, *Modell. Simul. Mater. Sci. Eng.* **2012**, 20, 075007.
- [24] C. Sun, R. Vaidya, *Compos. Sci. Technol.* **1996**, 56, 171.
- [25] Y. Uetsuji, S. Yasuda, Y. Teramoto, *Compos. Struct.* **2022**, 301, 116201.
- [26] H. Mehdipour, P. P. Camanho, G. Belingardi, *Composites, Part B* **2019**, 165, 199.
- [27] Y. Liu, F. P. V. D. Meer, L. J. Sluys, J. T. Fan, *Compos. Struct.* **2020**, 252, 112690.
- [28] T. A. Dutra, R. T. L. Ferreira, H. B. Resende, A. Guimarães, J. M. Guedes, *Compos. Struct.* **2020**, 245, 112305.
- [29] Q. Chen, G. Wang, X. Chen, J. Geng, *Compos. Struct.* **2017**, 182, 457.
- [30] A. El Moumen, M. Tarfaoui, K. Lafdi, *Appl. Compos. Mater.* **2017**, 25, 569.
- [31] Q. Zhang, H. Xu, X. Jia, L. Zu, S. Cheng, H. Wang, *Compos. Struct.* **2020**, 236, 111915.
- [32] A. Melro, P. Camanho, F. Andrade Pires, S. Pinho, *Int. J. Solids Struct.* **2013**, 50, 1897.
- [33] A. Melro, P. Camanho, F. Andrade Pires, S. Pinho, *Int. J. Solids Struct.* **2013**, 50, 1906.
- [34] S. K. Gopalraj, T. Karki, *Polymers* **2021**, 13, 3192.
- [35] H. Hooputra, H. Gese, H. Dell, H. Werner, *Int. J. Crashworthiness* **2004**, 9, 449.
- [36] M. Smith, *ABAQUS/Standard User's Manual, Version 6.9*, Dassault Systemes Simulia Corp., Rhode Island, USA **2009**.
- [37] Abaqus Analysis User's Guide, Abaqus 6.13, <http://130.149.89.49:2080/v6.13/books/usb/default.htm?startat=pt05ch24s02abm43.html> (accessed: June 2014).
- [38] Scikit Learn Density Estimation, <https://scikit-learn.org/stable/modules/density.html> (accessed: 2014).
- [39] Time serie split, https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html (accessed: 2016).
- [40] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, *Xgboost: Extreme Gradient Boosting, R Package Version 0.4-2*, Vol. 1, Scientific Research, China **2015**, pp. 1–4.
- [41] P. Probst, M. N. Wright, A.-L. Boulesteix, *Wiley Interdiscip. Rev.: Data Min. Knowl. Discovery* **2019**, 9, e1301.
- [42] B. Scholkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, *IEEE Trans. Signal Process.* **1997**, 45, 2758.
- [43] G. Bontempi, S. Ben Taieb, Y.-A. Le Borgne, *Machine Learning Strategies for Time Series Forecasting*, Springer Berlin Heidelberg, Berlin, Heidelberg **2013**, pp. 62–77.
- [44] J. Brownlee, How to Convert a Time Series to a Supervised Learning Problem in Python, <https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/> (accessed: August 2019).
- [45] C. Chatfield, *Time-Series Forecasting*, CRC Press, London, UK **2000**, <https://books.google.fr/books?id=PFHMBQAAQBAJ>.
- [46] G. E. Box, G. M. Jenkins, G. C. Reinsel, G. M. Ljung, *Time Series Analysis: Forecasting and Control*, John Wiley & Sons, New Jersey **2015**.
- [47] S. Makridakis, E. Spiliotis, V. Assimakopoulos, A.-A. Semenovglou, *J. Oper. Res. Soc.* **2022**, 74, 840.
- [48] J. J. Hopfield, *IEEE Circuits Devices Mag.* **1988**, 4, 3.
- [49] A. G. Ivakhnenko, *IEEE Trans. Syst. Man Cybern.* **1971**, 1, 364.
- [50] H. Madala, *Inductive Learning Algorithms for Complex Systems Modeling*, 1st ed., CRC Press, Boca Raton, USA **1993**.
- [51] I. Ebtehaj, H. Bonakdari, *J. Irrig. Drain. Eng.* **2020**, 147, 1.
- [52] H. Zhang, X. Liu, E. Cai, G. Huang, C. Ding, *Comput. Geosci.* **2013**, 56, 23.
- [53] M. Najafzadeh, *Eng. Sci. Technol. Int. J.* **2015**, 18, 42.
- [54] L. Yang, H. Yang, H. Liu, *Sustainability* **2018**, 10, 217.
- [55] O. Dag, C. Yozgatligil, *Algorithms* **2016**, 8, 379.
- [56] Multilayered Group Method of Data Handling of Machine Learning for Python, <https://github.com/kvoyager/GmdhPy> (accessed: January 2016).
- [57] M. Mozaffar, R. Bostanabad, W. Chen, K. Ehmann, J. Cao, M. Bessa, *Proc. Natl. Acad. Sci.* **2019**, 116, 26414.
- [58] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, *J. Sci. Comput.* **2022**, 92, 88.
- [59] S. Bekesiene, R. Smaliukiene, R. Vaicaitiene, *Mathematics* **2021**, 9, 626.
- [60] C. Olah, LSTM, <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed: August 2022).
- [61] Campusplastic, Computer Aided Material Preselection by Uniform Standards, <https://www.campusplastics.com/> (accessed: 1988).
- [62] D. Pulungan, G. Lubineau, A. Yudhanto, R. Yaldiz, W. Schijve, *Int. J. Solids Struct.* **2017**, 117, 177.