Non-destructive strength prediction of composite laminates utilizing deep learning and the stochastic finite element methods

Nastos, Christos; Komninos, Panagiotis; Zarouchas, Dimitrios

DOI
10.1016/j.compstruct.2023.116815

Publication date
2023

Document Version
Final published version

Published in
Composite Structures

Composite Structures 311 (2023) 116815

Contents lists available at ScienceDirect

Composite Structures

journal homepage: www.elsevier.com/locate/compstruct

Non-destructive strength prediction of composite laminates utilizing deep learning and the stochastic finite element methods

Christos Nastos, Panagiotis Komninos, Dimitrios Zarouchas *

Center of Excellence in Artificial Intelligence for Structures, Prognostics & Health Management, Aerospace Engineering Faculty, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, The Netherlands

ARTICLE INFO

Keywords:
Deep learning
Uncertainty
Composite structures
Probabilistic models
Strength prediction
Stochastic finite element method

ABSTRACT

A hybrid methodology based on numerical and non-destructive experimental schemes, which is able to predict the structural level strength of composite laminates is proposed on the current work. The main objective is to predict the strength by substituting the up to failure experiments with non-destructive experiments where the investigated specimen is loaded up to 20% of its maximum load. A significant gap exists between the 20% and the 100% load which is proposed to be treated by high fidelity physics-based numerical models, deep learning techniques, and non-catastrophic experiments. Thus, a deep learning algorithm is developed, based on the convolutional neural networks and trained by probabilistic failure analysis datasets which result from the utilization of the stochastic finite element method. Also, the Monte Carlo dropout technique is embedded into the developed convolutional neural network to estimate the uncertainty induced by the investigated variations between the simulated and experimental data. The current paper provides a thorough description of the proposed methodology and a practical example which demonstrates the validity of the method.

1. Introduction

Composite materials play a crucial role in the aerospace, wind energy, and automotive industries, as they offer a unique combination of low weight and high strength compared to traditional materials. This has led to a growing demand for the design and analysis of composite structures, as well as an increased need for experiments to verify and validate the design and analysis process with the final product and its desired specifications. However, experiments are costly in terms of energy, carbon emissions, human effort, and time, which can increase the total cost of the final product and the time it takes to bring a concept to market. Therefore, there is a need to explore alternative, more efficient methods to substitute the majority of the experimental effort in order to actively contribute to the verification and validation of the design and analysis process of composite structures.

The exploitation of digitalization and computer science can lead to trustworthy results by reducing significantly the experimental effort. More specifically, high fidelity simulations which resulted from the area of computational mechanics and robust machine learning (ML) techniques could be integrated in order to establish a sophisticated methodology of conducting virtual and hybrid (i.e. realistic and virtual) experimental procedures for the investigation of the mechanical response in composite structures.

The scope of current work is to develop and present a prognostic methodology which can predict the strength values in composite structures by avoiding the conduction of experimental procedures that lead to catastrophic failures. This practically means, that the strength of a composite structure is aimed to be predicted by conducting non-catastrophic tensile experiments up to 20% of maximum strain of the structure. The developed methodology encompasses the inherent stochastic nature of composite materials by utilizing the stochastic finite element method (SFEM). The dataset acquired by the probabilistic numerical analysis, is exploited as a training dataset to a two dimensional convolutional neural network (CNN) as it helps with automatic feature extraction and dimensionality reduction, since there is a spatial relationship between the data, making this type of artificial neural network (ANN) advantageous. Afterwards, the validity of the

Abbreviations: 2D, two-dimensional; ANN, artificial neural networks; BI, Bayesian inference; BNN, Bayesian neural networks; CDF, cumulative distribution function; CFRP, carbon fiber reinforced polymer; CNN, convolutional neural networks; DIC, digital image correlation; DL, deep learning; FC, fully connected; FE, finite element; FEA, finite element analysis; FEM, finite element method; FSDT, first-order deformation theory; GP, Gaussian process; K–L, Karhunen–Loève; KL, Kullback–Leibler; LHS, latin hypercube sampling; MC, Monte Carlo; MCS, Monte Carlo simulation; ML, machine learning; MLP, multi-layer perceptron; PDF, probability density function; RF, random field; SFEM, stochastic finite element method; VI, variational inference

* Corresponding author.
E-mail address: D.Zarouchas@tudelft.nl (D. Zarouchas).

https://doi.org/10.1016/j.compstruct.2023.116815
Received 12 October 2022; Received in revised form 16 January 2023; Accepted 11 February 2023
Available online 17 February 2023

proposed methodology is investigated in terms of strength prediction in quasi-isotropic carbon fiber reinforced polymer (CFRP) laminates by performing realistic tensile experiments. From the ML's point of view, it is challenging to treat data that comes from different sources for training and testing processes, because mostly the simulated data diverge from the experimental. Although the mismatch between data is a common issue in ML, the proposed method bounds successfully the acquired experimental data within the strain range calculated by the high fidelity SFEM. Regarding the induced uncertainty in the interval, a state-of-art technique is embedded into the CNNs for its estimation. In fact, it is possible to estimate this uncertainty even in black-box deep learning (DL) models by utilizing the Monte Carlo (MC) dropout technique which is completely integrated inside the ANN. The proposed methodology is expected to reduce substantially, the time required for a product to enter the service life, the material waste, the carbon emissions and consequently the final cost of the product.

Over the past decades, various numerical methods have been investigated and employed for the simulation of the mechanical response in composite structures. Fully analytical [1,2] and semi-analytical finite element methods [3,4] result in accurate solutions, but they are applicable to limited types of structural configurations (e.g. semi infinite strips and plates) and boundary conditions. On the other hand, the finite element method (FEM) is extensively used as a trustworthy numerical tool for the simulation of engineering problems including more complex geometry and boundary conditions [5]. The spectral element method seems more promising than the FEM especially, in the transient response analysis of composite structures. This is due to the exploitation of high order polynomial shape functions and Gauss–Lobatto–Legendre quadratures, which yield to fast and accurate solutions with consistent diagonal mass matrices [6,7]. However non-uniform remeshing is required when increasing the order of polynomial shape functions, which is a time-consuming process. The non-uniform remeshing and the consistent diagonal mass matrices are remedied simultaneously by the exploitation of modern wavelet-based elements which were recently reported in high demanding wave propagation simulations [8,9]. The main disadvantageous aspect that all the aforementioned numerical methods retain in common, is their deterministic nature. This means that they consider the mechanical properties of the composite materials as deterministic values, which may lead to significant deviations between the simulations and the experimental data.

To overcome this limitation, various stochastic finite element methods have been arisen which consider the inherent randomness of the composite materials and consequently the uncertainties on the investigated composite structures. The SFEM provides the ability to perform probabilistic analysis including material and/or loading stochasticity and predict the mechanical response in a probabilistic way, which is the most realistic approach. It is an extension of the deterministic finite element (FE) approach and is able to treat random effects by modeling uncertainties during the simulation of engineering problems [10]. The Monte Carlo simulation (MCS) and perturbation methods have been mainly utilized for the conduction of stochastic finite element analysis (FEA). However, the MCS is hindered by the demanding computational effort due to the large number of samples required [11] and the perturbation technique is limited to small perturbations and does not readily provide information on high-order statistics [12,13]. On the other hand, the combination of the Karhunen–Loève (K–L) expansion method with the Latin Hypercube Sampling (LHS) method have shown a great potential in terms of validity and computational efficiency. A recent work, presents that the aforementioned scheme is able to perform reliable probabilistic failure analysis for CFRP structures in a computationally efficient way [14].

Apart from the computational engineering methods, the recent rise of data-driven techniques has emerged new approaches for material strength estimation. ML has been proven to be a powerful data-driven technique for strength prediction using various algorithms [15–19]. From these methods, deep learning (DL), a subfield of ML, is reported to be the leading data-driven tool for failure analysis in composite structures that utilize large datasets. DL is a special kind of ML that uses an ANN inspired by the biological structure of the brain. In this regard, Hossain et al. [20] developed an ANN model to determine the compressive and tensile strength of engineered cementitious composite based on the mix design parameters. The model was trained and tested on experimental data and demonstrated more than 95% accuracy on the predicted experimental strengths. Deng et al. [21] reported a CNN model for compressive strength prediction of recycled concrete using 74 experimental sets of concrete block masonry with different mix ratios. Since this type of ANN captures spatial relationships inside each input sample, its properties are desirable for these varying mix ratios of the multi-dimensional recycled concrete experiments. The efficiency of the proposed CNN model was found to be higher than the traditional ANN models.

Previous studies for strength prediction through ML techniques exist in the literature, however intense experimental campaigns have been conducted in order to train the ML algorithms. Mangalathu et al. [22] have performed more than 500 experimental tests om reinforced concrete beam–column joints in order to train the ML algorithms to classify the failure modes and predict the shear strength values. Chopra et al. [15] have compared three ML techniques, the decision tree model, the random forest model and the neural network model in order to predict the compressive strength of concrete during three age levels, 28, 56 and 91 days. The ML algorithms have been trained by experimental data and the neural network method was found to provide the best predictions. Feng et al. [16] have developed an adaptive boosting approach based on ML methods to predict the compressive strength of concrete. Their method was compared with the ANN, the support vector machine, the linear regression and the classification and regression trees algorithms, obtained from the literature and found to outperform the latter. However, more than 1000 experimental tests were utilized to train the proposed boosting approach. On the same way, Abuodeh et al. [23] have reported a study based on the prediction of shear strength on externally strengthening reinforced concrete structures by the exploitation of ML techniques. A parametric study has been conducted to investigate the parameters effect on the shear capacity of strengthened reinforced concrete beams, whilst the ML algorithms were trained by an experimental campaign. Karina et al. [24] have reported a hybrid way for the prediction of tensile strength of corroded steel plates, which results from the combination of material characterization experiments with FE models used for the development of a machine learning predictor. All the aforementioned studies are focused on concrete structures with applications for the civil engineering sector.

All of the above ML techniques have one common limitation which is the lack of estimating the induced uncertainty that results from different sources, such as environmental conditions, sensor noise, human errors, model's approximations, along with others. Consequently, the uncertainty estimation is a key factor that should be considered during the development of a ML model. There are numerous works related to reliability analysis after predicting the mechanical properties of composite materials. Yan et al. [25] proposed a method for the prediction of the probability of failure for a bridge due to traffic overloading by combining an ANN and the MCS. The ANN was trained and tested on simulation data while the MCS technique managed to estimate the fatigue failure probability by randomly sampling the overloaded truck traffic and bridge parameters from a normal distribution. An estimation of the compressive strength of high-performance concrete via the Gaussian process (GP) has been established by Hoang et al. [26]. The GP has the privilege to automatically capture the uncertainty that comes from noisy data, nonetheless, it is computationally expensive when the training set consists of hundreds or thousands of samples with increasing dimensions.

Although the aforementioned approaches supported the proposed models with uncertainty, they considered only the one that results from noisy data, usually defined as aleatoric or data uncertainty. However,

in realistic applications, it is also very common to have a variation between the distributions of the training and testing sets leading to poor predictions even though the aleatoric uncertainty has been estimated. This variation could be detected by the model itself as another measure of the uncertainty that is known as epistemic or model uncertainty. The epistemic uncertainty could be the dominant uncertainty especially when the training set comes from simulations while the testing set comes from real experiments. In particular, for strength prediction models, a denoised version of training data could be generated from the finite element methods, whilst the trained model could be evaluated to the corresponding real structure. In such cases, the aleatoric uncertainty could be neglected, however the predictions during the testing phase may still be far from accurate, indicating that the epistemic uncertainty dominates. Consequently, it is also compulsory to estimate this source of uncertainty as its existence highlights a possibly large variation between training and testing sets or a suboptimal trained model.

Regarding the area of composite materials, a deep learning method which predicts the delamination in fatigue loaded composites is recently reported [27]. Strain patterns acquired by an experimental campaign, were employed as the training data to a CNN which is shown to have the potential for delamination prediction. In a different way, Chen et al. [28] have reported a ML technique to predict the failure of a unidirectional lamina subjected to triaxial loading conditions. The ML system has been previously trained by micromechanics-based FE models based on deterministic input. Gu et al. [29] have performed a parametric study on the prediction of joint strength in adhesive bonded joints with composite materials via FE models. The output of the FE models were exploited for the training of ML models in order to construct a rapid method to explore specific design variables and their effect on the joint strength. On a similar way, recent reports [30,31] utilize the combination between FEA and fracture models in order to train ML models and as a result to substitute the computationally demanding FE simulations with them.

According to the literature review, the exploitation of artificial intelligence and ML methods for the prediction of the mechanical response and the diagnosis of the damage state in both concrete and composite structures is very recently reported and seems to have a growing potential on engineering. At present, there are several reports which employ costly and time consuming experimental procedures to train the ML algorithms and some studies which employ deterministic FE models for the training process. However, to the authors best knowledge, there is no reported method which trains the ML algorithms with data obtained from FE analysis and test them with experimental data in order to validate the efficiency of the algorithm in a more realistic way. The novelty of current research is that a DL method utilizes numerical data obtained from the stochastic finite element method as a training data set. The SFEM encompasses the inherent uncertainties of the composite materials, thus leads to a more efficient training process. Afterwards, the validity of the DL method is investigated by performing realistic experiments and providing them as input to the ML algorithms which were previously trained by numerical stochastic data. As a consequence of this sim-to-real process, the potential shift of the data distribution due to the experimental data is estimated via epistemic uncertainty for strength prediction problems for the first time.

The current article is organized as follows. Section 2 presents a brief description of the stochastic finite element method, which is employed for the training process. A stochastic distribution algorithm is encapsulated on the FEM and stochastic stiffness matrices for probabilistic analysis are derived. Section 3 describes the theoretical background of the CNN and the uncertainty estimation process included on the current work. Section 4 describes the concept and implementation of the proposed methodology and mentions all the parameters used on the SFEM and CNN algorithms for the non-destructive strength prediction in composite laminates. Section 5 presents a demonstrative example of the proposed method, where the strength values of ten quasi-isotropic specimens are predicted by conducting non-destructive tensile tests up to 20% of the maximum strain and inserting the obtained full-field strain field as input to the DL models which were previously trained by the respective stochastic finite element model.

2. The stochastic finite element method

The integration of the K–L expansion and LHS method into FEA leads to a beneficial SFEM in terms of probabilistic distribution and computational efficiency among the others. Thus, the present work exploits the LHS method to generate an adequate sampling size of random values and the K–L method to distribute the uncertain material properties along the random field (RF) mesh. The RF is the mesh of the investigated structural domain where random mechanical properties are distributed. A mapping interpolation algorithm is used to ensure the linking of the RF nodes with the Gaussian integration points exist on the FE mesh. The first-order shear deformation laminated plate theory (FSDT) is employed for the calculation of stochastic stiffness matrices and for the development of the proposed SFEM. During the stochastic finite element analysis, the Puck's failure criterion is exploited to evaluate the existence and the type of failures on the matrix and on fibers as well.

2.1. The spatially varying random field distribution method

The spatially varying random field distribution method employed on the current paper, encompasses the K–L expansion, the LHS and the mapping interpolation method. A continuous random function can be represented by a complete set of deterministic functions with corresponding random coefficients. Based on this idea, the K–L expansion was introduced by Spanos and Ghanem [32]. The K–L expansion can be seen as a special case of the orthogonal series expansion where the orthogonal deterministic functions, the eigenfunctions of the covariance function for the random field and the uncorrelated random variables are involved [33]. The current paper employs the K–L expansion to discretize spatially varying random fields in the two-dimensional (2D) domain. Thus, both the material properties (elastic modulus, shear modulus, Poisson ratios) and the strengths (tensile, compressive) on fiber and matrix direction are decomposed into a deterministic and a stochastic part as well. Consider a random field $w(\mathbf{x},\theta)$ with mean value $\mu_w(\mathbf{x})$, the K–L expansion is written as [34]

$$w(\mathbf{x},\theta) = \mu_w(\mathbf{x}) + \sum_{i=1}^{M} \sqrt{\lambda_i}\phi_i(\mathbf{x})\xi_i(\theta) \tag{1}$$

where $\mu_w(\mathbf{x})$ could be any of the randomly distributed values along the RF mesh, $\mu_w(\mathbf{x})$ is the mean value, M is the number of K–L terms, $\sqrt{\lambda_i}$ and $\phi_i(\mathbf{x})$ are the eigenvalues and the eigenvectors respectively resulted from the eigenvalue analysis of the Fredholm integral equation of the second kind, $\xi_i(\theta)$ are the uncorrelated zero mean random variables obtained from the LHS method and θ is the size of the random variables generated by the LHS. The LHS method is reported to outperform the MCS, since it spreads the sampling points more evenly across all possible values by decreasing substantially the size of samples and consequently the computational effort [35,36]. The calculation of the eigenevalues and eigenvectors results from the solution of Eq. (2).

$$\mathbf{CD} = \mathbf{\Lambda CD} \tag{2}$$

$$\mathbf{C} = \sum_{2e=1}^{N_{RF}} \sum_{1e=1}^{N_{RF}} \int_{\Omega_{1e}} \int_{\Omega_{2e}} C(\mathbf{x}_1;\mathbf{x}_2) < \mathbf{N}(\mathbf{x}_1) >^T < \mathbf{N}(\mathbf{x}_2) > |\mathbf{J}_e|^2 dA_{2e} dA_{1e} \tag{3}$$

$$\mathbf{B} = \sum_{e=1}^{N_{RF}} \int_{\Omega_e} < \mathbf{N}(\mathbf{x}) >^T < \mathbf{N}(\mathbf{x}) > |\mathbf{J}_e| dA_e \tag{4}$$

where N_{RF} is the number of RF elements used for the stochastic distribution, $\mathbf{N}(\mathbf{x})$ are the shape functions used as basis functions for the RF mesh, $\mathbf{J}_e$ is the Jacobian and $C(\mathbf{x}_1;\mathbf{x}_2)$ is the covariance function for spatially varying fields derived in Eq. (5).

$$C(\mathbf{x}_1;\mathbf{x}_2) = \sigma_w^2 exp\left(-\frac{|x_1 - x_2|}{b_{c1}L_{D1}} - \frac{|y_1 - y_2|}{b_{c2}L_{D2}}\right), \ \mathbf{x}_1, \mathbf{x}_2 \in \Omega \tag{5}$$

b_{c1}, b_{c2} are the correlation length parameters of the two different directions of the domain, L_{D1}, L_{D2} are the physical characteristic lengths and σ_w is the standard deviation of the random property w.

The combination of Eqs. (2)–(5) leads to the solution of Eq. (1) which practically calculates the random distribution of the desired property w on the nodes of the RF mesh. The mapping interpolation algorithm is a stepwise procedure and is employed in order to transfer the calculated stochastic properties from the RF nodes to the FE nodes and subsequently to the Gaussian integration points included in the FE mesh. The number and location of the Gaussian integration points depends on the selection of the polynomial order of the shape functions. The current work exploits the 9-node quadratic quadrilateral FEs for the RF and the FE mesh as well. The link between the RF nodes and the FE nodes is described in Eq. (6), where the eigenfunctions are transferred from the RF to the FE nodes.

$$\phi_i^{jFE} = \sum_{m=1}^{9} N_m(\xi_i^j, \eta_i^j)\, \phi_m^{jRF} \tag{6}$$

where ξ_i^j, η_i^j are the isoparametric coordinates attached to the jth RF element, which correspond to the ith FE node and are calculated within each RF element. A nonlinear system of equations that relates the FE nodal coordinates $(x_i^j,\ y_i^j)$ within the jth RF element with RF nodal coordinates $(X_m^j,\ Y_m^j)$ of the same jth RF element has to be solved and is shown in Eq. (7).

$$\begin{aligned} &\sum_{m=1}^{9} N_m(\xi_i^j, \eta_i^j) X_m^j - x_i^j = 0 \\ &\sum_{m=1}^{9} N_m(\xi_i^j, \eta_i^j) Y_m^j - y_i^j = 0 \end{aligned} \tag{7}$$

The next step is to transfer the random properties from the FE nodal points to the Gaussian integration points included on each FE element for the consistent calculation of the stochastic stiffness matrices. Eq. (8) describes the link between the FE nodes and the Gaussian integration points for each random property and each θ random case generated by the LHS method.

$$w^G(\xi, \theta) = \sum_{i=1}^{9} N_i(\xi) w_i^{FE}(\mathbf{x}, \theta) \tag{8}$$

where ξ are the local coordinates of the FE field (ξ^j, η^j) for the jth RF element. A thorough description of the spatially varying random field distribution method used in the current research can be found in [14,37].

2.2. Formulation of the stochastic laminated composite plate element

The integration of the aforementioned spatially varying random field distribution method is performed within the first-order shear deformation laminated plate theory, where the transverse normals do not remain perpendicular to the midsurface after deformation [38]. The kinematic assumptions used to describe the stochastic displacement field for each random case θ is of the form

$$\begin{aligned} u(x, y, z, \theta) &= u_0(x, y, \theta) + z \cdot \beta_x(x, y, \theta) \\ v(x, y, z, \theta) &= v_0(x, y, \theta) + z \cdot \beta_y(x, y, \theta) \\ w(x, y, z, \theta) &= w_0(x, y, \theta) \end{aligned} \tag{9}$$

where u_0, v_0, w_0 denote the stochastic displacements on the x-axis, y-axis, and z-axis at the mid-plane of the plate respectively; β_x, β_y denote the stochastic rotations of the cross-section and z is the local thickness coordinate.

After the derivation of the displacement field into the strain field, the principle of virtual work is employed and finally the calculation of the stochastic stiffness matrices $[\mathbf{K}_e(\theta)]$ for each element e and for each randomly generated sample θ is described in Eq. (10).

$$[\mathbf{K}_e(\theta)] = \iiint_V [\mathbf{R}^T][\mathbf{K}_L(\theta)][\mathbf{R}] dV \tag{10}$$

$\mathbf{K}_L(\theta)$ includes the $A(\theta)'s$, $B(\theta)'s$ and $D(\theta)'s$ terms which are the extensional, bending and bending-extensional coupling stiffnesses for each random θ case and $\mathbf{R}$ is the matrix resulted from the derivation of shape functions according to the FSDT. A detailed description of the stepwise formation of stochastic finite elements based on the FSDT for laminated composite plates is reported in [14].

3. Convolutional neural networks and uncertainty estimation

The current section describes briefly the theoretical background of DL starting from the ANN to the CNN and mentions the significant importance of the latter and the method to integrate the uncertainty estimation into neural networks.

3.1. Convolutional neural networks

Deep learning is a subfield of ML that uses ANN as its key component. These networks are based on a collection of connected units or nodes called artificial neurons where each neuron receives an information and processes it, then it passes the information to its neighbor neurons, and so on. Technically, typical neurons are composed by weights and biases that adjust as learning proceeds. The mathematical process inside a neuron is defined as

$$N_j = \sum_{i=1}^{K} w_{ji}^l x_i + b^l \tag{11}$$

where N_j is the initial output of neuron and w_{ji}^l is the weight relevant to the connection between the jth neuron at the lth layer to the $(l-1)^{th}$ layer's i_{th} neuron. The neuron also contains b^l to consider the bias. The weights and biases are the parameters of the ANN that are being adjusted during training via the backpropagation algorithm. This algorithm starts by calculating the gradients of the error between the ANN's outputs and the corresponding true values and it goes backwards towards the calculation of the gradients of all the parameters with respect to the error. Details about the backpropagation algorithm can be found in [39]. The vanilla ANN presented above is called multi-layer perceptron (MLP) [40] and uses fully connected (FC) layers, i.e. each neuron of a layer is connected with all the neurons of the next one. The CNN is a specific kind of ANN that performs better than the MLP when a spatial or temporal relationship between the data exists as it automatically extracts features through convolution, thus keeping important spatio-temporal information by also reducing processing requirements. CNN could be 1-dimensional (CNN1D), 2-dimensional (CNN2D), or 3-dimensional (CNN3D). CNN1D is used to extract features from spectral data, such as time-series [41], CNN2D is applied with spatial data (e.g. images, 2D structures) [42,43] and CNN3D for spatio-temporal data, such as gesture recognition [44]. A CNN architecture usually consists of some convolutional layers followed by pooling layers, and then FC layers. Features are automatically learned and extracted layer by layer through multiple convolution layers and pooling layers, and then usually passed into an FC layer to produce the final output. Similar to MLPs, the calculation process of a convolutional layer uses again weights and biases, but the array of weights is an array of matrices where each matrix represents a filter or kernel that is convolved with the input matrix. Assuming the input of a convolutional layer is $X \in R^{H \times B}$ where H and B are the two input dimensions representing the height and width, respectively, then the output of the convolutional layer is calculated as follows:

$$z_{i,j,k}^l = x_{i,j}^l w_k^l + b_k^l \tag{12}$$

$$a_{i,j,k}^l = F(z_{i,j,k}^l) \tag{13}$$

where $z_{i,j,k}^l$ is the kth feature map of the kth filter or kernel, centered at location (i, j) of the lth layer, $a_{i,j,k}^l$ denotes the output after applying a nonlinear activation function F to the feature map, w_k^l and b_k^l are the

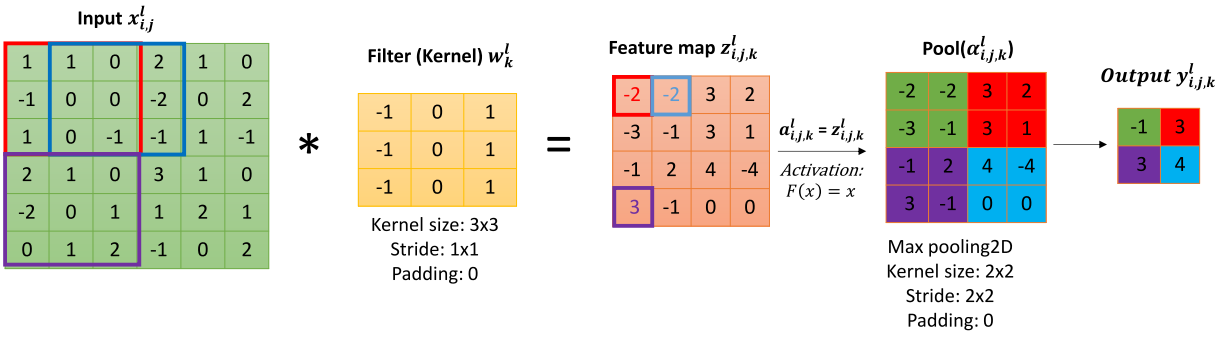


Fig. 1. An example of 2-dimensional convolution operation followed by a 2-dimensional pooling technique.

weight vectors and bias of the kth filter and lth layer, and $x^l_{i,j}$ is the corresponding input of that layer. It should be noted that the weights of the kernel w^l_k that produce the feature map $z^l_{i,j,k}$ are shared for different locations (i, j). This sharing mechanism reduces the model complexity, and makes the network easier to train, while it retains the nonlinearities from the activation function which are needed for the detection of nonlinear features. After the kernel is applied around the location (i, j) of the input matrix, it slides to the neighboring locations resembling to a 2-dimensional overlapping window, called a stride. Applying a stride (x_{stride}, y_{stride}) moves the kernel to be centered firstly, at the location $(i + x_{stride}, j)$ and then at $(i + x_{stride}, j + y_{stride})$. If the stride is large and exceeds one of the dimensions of the input, then the input matrix is zero-padded to increase the size of dimensions accordingly. Padding could be also applied to upsample the input dimensions if the dataset has relatively small H and B. Finally, backpropagation is applied in the same way as in the MLP.

Usually, between two convolutional layers, there is a pooling layer that aims to reduce (downsampling) the dimensionality of the activated feature maps $a^l_{i,j,k}$. For each $a^l_{i,j,k}$ the pooling process is as follows:

$$y^l_{i,j,k} = pool(a^l_{a,b,k}), \quad \forall a, b \in V^{ij} \subseteq \mathbb{R} \tag{14}$$

where $pool(\cdot)$ is the pooling function and V^{ij} is a sub-space around location (i, j) where pooling is applied. In CNNs two pooling operations are often applied; max pooling or average pooling [45]. Fig. 1 shows the convolution operation which is usually followed by a pooling technique to further reduce the input dimensions. In this example, a 6×6 input array and a 3×3 filter are convolved with the input to give the feature map. Here, the stride is equal to 1 for both dimensions, hence the filter slides by one position to the right or down giving a 4×4 output after doing element-wise multiplication. For simplicity, the activation function is assumed to be the identity function, so the feature map remains the same. Next, the feature map is passed through a max pooling operation, i.e. the maximum around a location is taken, with a 2×2 filter and 2×2 stride to give the final reduced 2×2 output.

Summarizing, the CNNs start from a convolutional layer, a nonlinear activation function is applied, and the new array is passed through a pooling layer. This is a group of layers that forms the first part of a typical CNN architecture. The second part is the MLP with some FC layers. Before applying FC layers, it is necessary to reshape the array to become 1-dimensional, thus a special layer termed as flattening layer is used that simply concatenates the dimensions of the output of the first part of the group in one dimension. Additionally, in the first part of the group, some other layers could be added, such as dropout [46] which is a simple technique to avoid overfitting by regularizing the neural network, and a batch normalization layer [47], which is a method to normalize the output of each convolutional layer for reducing the intrinsic covariance shift induced by the neural network itself. The accuracy of the CNN is improved by stacking many groups of layers together for confronting problems with spatial or temporal relationships.

3.2. Uncertainty estimation in artificial neural networks

Uncertainty exists in every type of ML problems and sometimes it may be so large that the model could not provide any reliable prediction. In the literature, there are many different definitions of uncertainty, but the most widespread is its decomposition into epistemic and aleatoric uncertainty [48]. On the first hand, epistemic or model uncertainty confronts the fact that no precise ML model exists that completely understands the training and testing distributions as well. Having high epistemic uncertainty means that the model is unable to effectively capture the entire distribution of the dataset. The out-of-distribution data may arise at the testing phase and could be completely different from the training set. A mismatch between test and training distributions is called dataset shift [49]. This shift in data, could be detected via epistemic uncertainty and reduced by acquiring more data that could assist the model to better extract features from the inputs. It should be noted that the term "out-of-distribution" data used in the literature may confuse the readers as it does not necessarily mean that the input data are out of the boundaries of the training set, but they can lie inside those boundaries at locations where data does not exist. In the current case, the SFEM that provides the training data set has been already validated with experiments [14] and guarantees that the testing set is enclosed in the training one. On the other hand, the aleatoric uncertainty arises due to the complexity, noise, and random nature of the data. This type of uncertainty cannot be reduced by acquiring more data, hence it is an irreducible uncertainty. The total uncertainty is simply the sum of the aleatoric and epistemic.

Compared with ANNs, the epistemic uncertainty could be detected in the model's weights and be further reduced by tuning the hyperparameters of the ANN or by acquiring more data for training. The detection of the aleatoric uncertainty is demanding and should be expressed as an additional output neuron that predicts the variance in regression tasks [48] or to measure the model's entropy on the outputs in classification tasks [50]. In the current study, the aleatoric uncertainty is treated by the SFEM, which distributes successfully the uncertain mechanical properties resulted from the manufacturing process. However, the exploitation of simulation data for training while testing on experimental data can result in alternate distributions for those two phases leading to a potential dataset shift. In this regard, the estimation of epistemic uncertainty is crucial for ensuring the proposed model's strength predictions. Instead of outputting a single deterministic value which is usually an overconfident prediction of the model, the mean value and the variance of stochastic predictions is estimated, hence a distribution over each prediction is achieved which enhances the reliability of the proposed model.

3.2.1. Estimation of epistemic uncertainty

As mentioned before, the epistemic uncertainty emerges mainly when there is a large variation between the training and testing sets or when the model is poorly trained. This type of uncertainty is mistakenly ignored in many situations and, in this study, it is the dominant source

of uncertainty. This can be easily understood as the training data comes from the stochastic finite element method while the testing data comes from a real case study, thus the variation between those sets is expected to be large. To capture the epistemic uncertainty in an ANN, one approach is to use deep ensembles [51–53]. Instead of using one ANN, with ensembles, a discrete number of models can be used that would converge differently due to the random initialization of the parameters. This implies that each ANN model is expected to give different outputs for a specific input sample, hence a measure of the epistemic uncertainty over the parameters could be achieved. Despite the effectiveness of deep ensembles, they are computationally prohibitive for large or high-dimensional datasets. A more attractive approach is to convert the model's deterministic parameters into distributions, for instance, a Gaussian distribution: $\theta_{NN} \sim N(0, \mathbf{I})$, where $\theta_{NN} = \{w, b\}$ denotes the parameters of ANN (weights, biases), and $\mathbf{I}$ is the identity matrix. These models are known as Bayesian Neural Networks (BNN) [54–57] and are considered as a special case of the ensembles [58]. Instead of optimizing those parameters directly, they are updated via the Bayes' theorem at first, and then marginalization is performed by averaging over the sampled parameters. To achieve this, the Bayesian approach defines a model likelihood $p(y|x, \theta_{NN})$ with distributed parameters θ_{NN}, and it is possible to calculate the posterior distribution $p(\theta_{NN}|X, Y)$ according to Bayes' theorem.

$$p(\theta_{NN}|X, Y) = \frac{p(Y|X, \theta_{NN})p(\theta_{NN})}{p(Y|X)} \tag{15}$$

where the denominator is calculated by:

$$p(\theta_{NN}|Y, X) = \int_{\theta_{NN}} p(Y|X, \theta)p(\theta_{NN}) \tag{16}$$

The probability $p(Y|X)$ represents the uncertainty of a set of outputs Y given a set of inputs X. Given a sample x from inputs X the corresponding output is predicted y:

$$p(y|x, X, Y) = \int_{\theta_{NN}} p(y|x, \theta_{NN})p(\theta_{NN}|X, Y)d\theta_{NN} \tag{17}$$

Eq. (17) is the marginalization process mentioned above or inference, otherwise, and the output probability distribution is called predictive distribution. Unfortunately, the posterior $p(\theta_{NN}|X, Y)$ cannot be computed analytically because the denominator of Eq. (15) is intractable, however it can be approximated via the Bayesian inference (BI) or variational inference (VI). In case of BI, the posterior distribution is directly computed effectively via the MC techniques, but these techniques are extremely slow, especially when it comes to training neural networks. Since BI requires large computation power and resources, VI is preferred in Bayesian neural networks (BNNs). In VI the aim is to approximate the posterior distribution with another distribution $q_\phi(\theta_{NN})$, with θ_{NN} representing the weights of the BNN, and ϕ the additional variational parameters to approximate the posterior distribution of those parameters θ_{NN}. After training, the distribution of $q_\phi(\theta_{NN})$ should converge to the posterior. In this regard, the Kullback–Leibler divergence (KL) [59] is computed and minimized to update the variational parameters ϕ:

$$KL(q_\phi(\theta_{NN}) \parallel p(\theta_{NN}|X, Y)) = \int_{\theta_{NN}} q_\phi(\theta_{NN}) log \frac{q_\phi(\theta_{NN})}{p(\theta_{NN}|X, Y)} d\theta_{NN} \tag{18}$$

the KL estimates the level of similarity between the assumed distribution q modeled with parameters ϕ and the posterior distribution produced from Bayes' theorem which updates the parameters θ_{NN} of the BNN. The $q^*_\phi(\theta_{NN})$ term is denoted as the optimized distribution after minimizing the KL, then the predictive distribution of Eq. (17) can be approximated:

$$\begin{aligned} p(y|x, X, Y) &= \int_{\theta_{NN}} p(y|x, \theta_{NN})p(\theta_{NN}|X, Y)d\theta_{NN} \\ &\approx \int_{\theta_{NN}} p(y|x, \theta_{NN})q^*_\phi(\theta_{NN})d\theta_{NN} \end{aligned} \tag{19}$$

$$q^*_\phi(\theta_{NN}) = argmin[KL(q_\phi(\theta_{NN}) \parallel p(\theta_{NN}|X, Y))] \tag{20}$$

However, one problem still exists; KL minimization process is still intractable as it demands the posterior distribution, hence the denominator of Eq. (15) is needed. Fortunately, it is proven that after rearranging KL into the evidence lower bound (ELBO) maximization [60], the intractable denominator could be avoided and lead to the objective of VI:

$$\begin{aligned} L_{VI} = &\int_{\theta_{NN}} q_\phi(\theta_{NN}) log p(Y|X, \theta_{NN})d\theta_{NN} \\ &- KL(q_\phi(\theta_{NN}) \parallel p(\theta_{NN})) \end{aligned} \tag{21}$$

$$\begin{aligned} p(y|x, X, Y) \approx &\int_{\theta_{NN}} q^*_\phi(\theta_{NN}) log p(Y|X, \theta_{NN})d\theta_{NN} \\ &- KL(q^*_\phi(\theta_{NN}) \parallel p(\theta_{NN})) \end{aligned} \tag{22}$$

$$q^*_\phi(\theta_{NN}) = argmax[L_{VI}] \tag{23}$$

Interested readers may refer to [60–62] for the analytical approximation of the posterior distribution via the VI and for the proof of the KL rearrangement that leads to Eq. (21). Notice that the minimization objective now becomes a maximization of the L_{VI} as KL should be minimized and contributes negatively to Eq. (21). The prior distribution $p(\theta_{NN})$ over the parameters θ_{NN} and the likelihood $p(Y|X, \theta_{NN})$ are known, thus the predictive distribution is now tractable.

The main idea in VI is to choose the parameterized distributions q_ϕ, which for the BNNs are the distributions from which the parameters θ_{NN} are being sampled. For example, if one assumes an initial family of normal distributions $q_\phi(\theta_{NN}) \sim N(\mu, \sigma^2)$ for each parameter θ_{NN}, then the parameters to be optimized are $\phi^i = \{\mu_i, \sigma_i^2\}, i = \{0, 1, \ldots, N_{\theta_{NN}}\}$, where $N_{\theta_{NN}}$ is the total number of the BNN parameters. After finding the corresponding ϕ that optimizes q_ϕ, a good approximation of the intractable posterior distribution is obtained and the parameters θ_{NN} could be updated by sampling over those distributions.

3.2.2. Monte Carlo dropout

Despite VI being a great approximation of the posterior distribution, it is still practically challenging, especially when it comes to tuning the hyperparameters of the ANN where the same training process should be processed in an iterative way prior selecting the optimal hyperparameters. It should be noted that the term hyperparameter implies every constant variable which value is used to control the learning process, such as the number of neurons and layers, and the learning rate for updating the weights and biases during the backpropagation. Thereby, the Monte Carlo dropout [63] is a simple and computationally efficient technique for estimating the epistemic uncertainty by approximating the predictive distribution described in Eq. (22). In general, dropout [46] is used as a regularization term in ANNs. In every training step, binary variables for each layer's neuron are sampled. Each binary variable has a value of 1 with a probability p_i for the ith layer, and a value of 0, otherwise. If the value is equal to 0, then the neuron is dropped and is not used for calculating the layer's output, while it remains active, if it is equal to 1. The same binary variable values are used during backpropagation. The probability p_i is called dropout rate.

Dropout is an effective technique for solving the problem of overfitting, which is when the model perfectly fits the training data and cannot generalize to the unseen one. Usually, dropout is enabled during the training process and then is disabled during the testing, in order to provide a deterministic output. In case of retaining it active during the test phase, the ANN would provide a variety of output values for each specific input. This provides an estimation about the epistemic uncertainty. Therefore, during the testing phase N different outputs could be produced by inferring over each input x_i as in the MC techniques. The process that enables dropout during the testing phase to apply inference is termed as MC dropout.

The MC dropout is a special case of BNN and an easy-to-implement technique since it can be applied naturally inside the ANN. Fig. 2

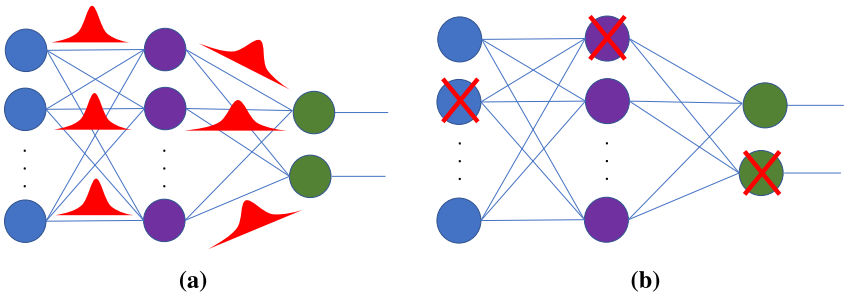


Fig. 2. A graphical representation of: (a) BNN; (b) dropout.

illustrates a BNN and a vanilla ANN including dropout. As already mentioned, in case of BNN, the weights are distributed by selecting a prior distribution and then the VI is applied to update the weights using another parameterized distribution $q_{\theta_{NN}}$ called variational distribution. Dropout randomly removes some neurons in layer i with probability p_i, hence it could be considered as a special case of a variational distribution [64].

Since dropout provides a binary condition for each neuron (enable, disable), the distribution that manipulates the neurons is mostly a Bernoulli distribution. To relate the VI, the BNN, and the MC dropout, the variational distribution $q_\phi(\theta^i_{NN})$ used in the current research is defined for every layer i as:

$$\theta^i_{NN} = \phi^i \cdot diag([u_{i,j}]_{j=1}^{K}) \tag{24}$$

$u_{i,j} \approx Bernoulli(p_i), i = \{1, \ldots, L\}, j = \{1, \ldots, K_{i-1}\}$, where $u_{i,j}$ denotes the random variable of the ith layer and jth neuron that follows a Bernoulli distribution with probability p_i, L is the number of layers, and K is the number of neurons for the ith layer. The $diag(\cdot)$ operator maps vectors to diagonal matrices whose diagonals are the elements of the vectors. The authors in [63] suggest that the dropout should be applied after every layer that has weights and biases with a dropout rate value of around 0.5, except the input layer that should have less neurons zeroed than the other, thus a lower dropout rate to avoid losing important information from the input data. Consequently, the posterior distribution could be approximated over the parameters θ^i_{NN} by updating the variational parameters ϕ^i via the loss function:

$$L_{dropout} = \frac{1}{N}\sum_{i=1}^{N} MSE(y_i, \hat{y}_i) + \lambda \sum_{i=1}^{L} \|w_i\|_2^2 \tag{25}$$

where N is the number of samples and the second term is equivalent to the L_2 weight regularization assuming a normal prior distribution over the parameters. Using the current loss function for training the model, the approximation of the integral of the predictive distribution during inference is achieved in Eq. (22)

$$\begin{aligned} p(y|x, X, Y) &\approx \int_{\theta_{NN}} p(y|x, \theta_{NN}) q^*_\phi(\theta_{NN}) d\theta_{NN} \\ &\approx \frac{1}{T}\sum_{i=1}^{T} p(y|x, \hat{\theta}^i_{NN}) \end{aligned} \tag{26}$$

where T denotes the MC samples and $\hat{\theta}^i_{NN}$ is the estimation of the updated parameters θ^i_{NN}. Usually, some tens or hundreds of T predictions are adequate [46,63] for calculating the uncertainty. The summation operation in Eq. (26) provides an array of predictions for each x_i from which the estimation of the epistemic uncertainty is achieved for the proposed model. The uncertainty can be expressed via the probability density function (PDF) in many ways. In this study, a non-parametric density function called kernel density estimator [65] has been chosen to approximate the PDF that expresses the epistemic uncertainty. Moreover, the 95% confidence intervals are calculated by using the variance of the predictions.

4. The concept and implementation of the method

The current section describes the concept of the proposed methodology in a stepwise procedure from the beginning where the initial lamina properties and the geometry of the investigated composite laminate are inserted, until the final prediction of the strength distribution of the specific laminate. In addition, all the parameters employed for the implementation of the SFEM and of the CNN algorithms are presented.

4.1. The concept

The objective of the current method is to perform non-destructive strength predictions for composite laminates by utilizing DL techniques, the SFEM and experimental data acquired from tensile/compressive experiments. The overall concept is illustrated in Fig. 3 in a stepwise procedure:

- Step 1. Define the lamina (ply) mechanical properties, the orientation and geometry (length, width, thickness) of the investigated laminated composite.
- Step 2. Exploitation of the K–L expansion and the LHS method to distribute the stochastic ply-level mechanical properties along the structural domain and to generate n random structures respectively.
- Step 3. Perform probabilistic failure analysis by employing the SFEM for the calculation of: (i) the full-field axial strains on the load steps of 10% and 20% of the maximum load (ii) the probability density function (PDF) regarding the strength of the entire laminate.
- Step 4. Insertion of the strains and strengths acquired from the SFEM as the input required by the DL algorithms for the training process.
- Step 5. Perform the non-destructive tensile/compressive experimental test for the investigated laminate coupon until 20% of the maximum load and acquire the full-field axial strains for both 10% and 20% loading steps by the digital image correlation (DIC) method.
- Step 6. Insert the two full-field axial strains of the investigated composite laminated specimen into the DL algorithms and predict a distribution of strength values including the epistemic uncertainty by utilizing the MC dropout technique.

Steps 1 to 4 are performed offline before the actual non-destructive testing process of the real specimen begins. This improves substantially the execution time of the proposed method.

The proposed method avoids the tensile/compressive tests until failure initiation or breakage in order to extract the overall strength of the laminated structure, since it is predicted by a high-fidelity stochastic numerical method and a trained DL algorithm.

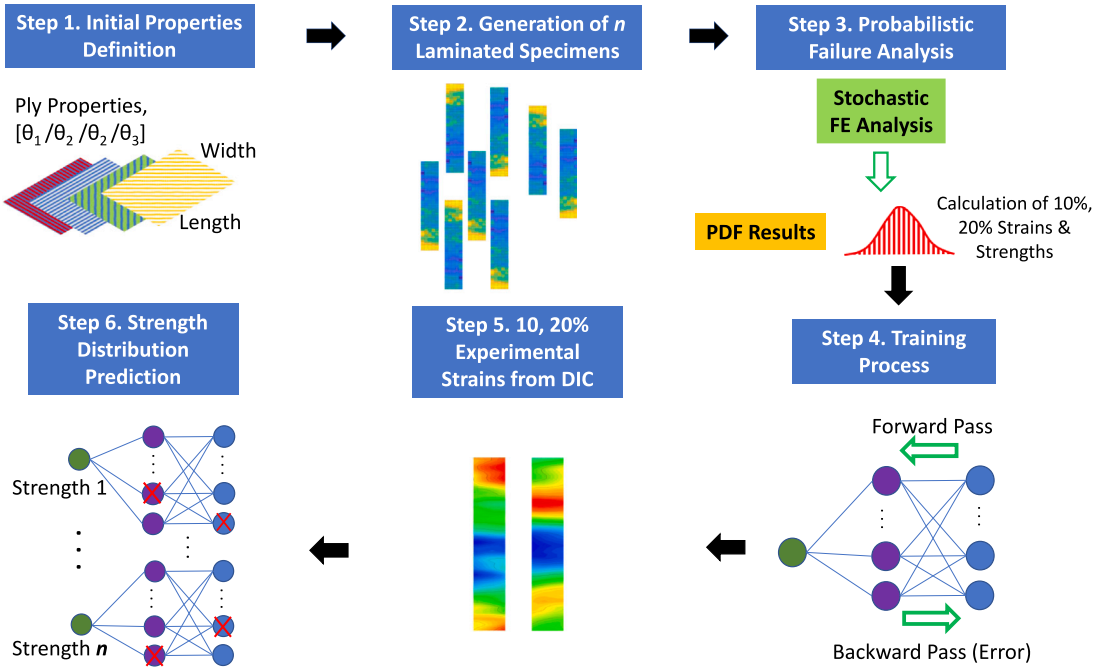


Fig. 3. The concept of the non-destructive methodology for prediction of strength in composite laminates.

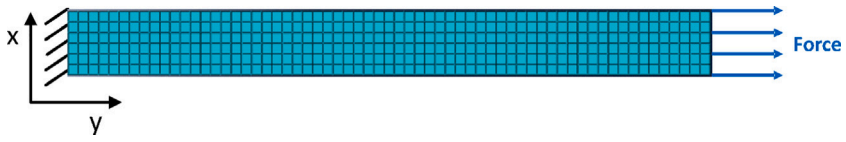


Fig. 4. A sketch of the quasi-isotropic laminate model clamped on its left edge and subjected to tensile loading on its right.

4.2. Implementation

The current subsection describes the implementation of the training process followed for the proposed DL method in order to achieve a non-destructive strength prediction for composite laminates. All the parameters employed in both the stochastic failure analysis models and the deep learning algorithms are described.

4.2.1. Description of the stochastic finite element model

A stochastic numerical model for the probabilistic failure analysis of a quasi-isotropic CFRP is developed by exploiting the aforementioned SFEM. The dimensions of the simulated composite laminate are $250 \times 25 \times 2$ mm^3 with lamination of $[(0/90/\pm 45)_s]_2$. The boundary conditions of the simulated component are shown in Fig. 4 where a clamp on the left edge and a distributed incremental load on the right edge are enforced.

The 9-node quadratic type of finite element is exploited for both the RF and the FE mesh. The first type of mesh ensures the adequate distribution of the randomness in terms of mechanical properties and the latter ensures the convergence of the obtained solution in terms of displacements, stresses and strains. For that reason, a RF mesh size of 30 × 3 elements and a FE mesh size of 60 × 6 are used. The K–L expansion parameters employed by the stochastic FE model are shown in Table 1. The selection of the parameters is explained in the authors recent work [14], where convergence studies are reported. The only difference between the current and the previous work in terms of stochastic parameters is the number of randomly generated samples. It has been previously shown that 100 samples obtained by the LHS method are adequate in order to describe the uncertainties along the structural domain, since it is a more efficient method than the MCS. However, on the current work 1000 samples are used, in order to achieve improved results regarding the training process of the DL algorithms. During the current research, the Puck's failure criterion [66] is exploited at each incremental solution for the probabilistic assessment of different failure modes. Obviously, the proposed method is able to accommodate any type of failure criterion according to the user's selection.

Table 1
K–L expansion parameters used for the development of the stochastic finite element model.

Property type (Symbol)	Values
K–L terms (M)	12
Correlation length x-axis (b_{c1})	0.1
Correlation length y-axis (b_{c2})	0.1
Number of samples (θ)	1000

4.2.2. Implementation of the deep learning techniques

Each sample in the input data consists of a 3D array with shape (6, 60, 2), where the first two dimensions represent the specimen's strain values and the last one the 10% and 20% strain conditions, termed as features in the area of ANN. The values 6, 60 result from the number of FEs employed on the y- and x-axis (Fig. 4) respectively. The output shape of each sample is a single value representing the specimen's strength. The training set consists of 1000 samples that are randomly split into 90%/10% representing training/validation sets. The validation technique is employed during the training process to evaluate the current model on a small unseen dataset and to check for any potential overfit to the training data. This is also useful if in the future the model needs retraining to predict the strength of varying structures by securing an appropriate learning process on the simulation data

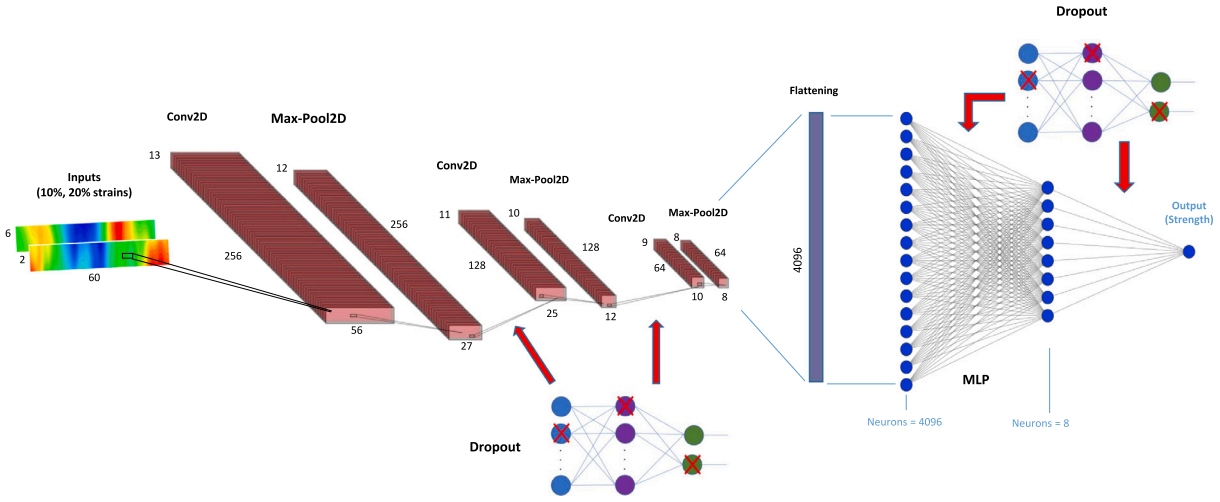


Fig. 5. The architecture of the proposed CNN model.

before applying to the real cases. Before training the model, the training set was randomly shuffled to avoid learning any sequential pattern of the data. After the training process, the developed model is evaluated on experimental data resulted from tensile experiments during the testing phase. The experimental dataset consists of full-field strain measurements and was acquired by the DIC method for 10 specimens at 10% and 20% of the maximum strain level, without inducing any failures on the specimens. The testing data has the same dimensions as the training one. 200 MC samples are executed by following the MC dropout technique instead of just outputting a single value for each of the 10 specimens representing their predicted strength. In this regard, the dropout layers should be manually activated during the testing phase to lead in different predictions given an input.

To apply any ANN model, the dataset should be normalized or standardized. In this work, the input data is normalized to the range of $[-1, 1]$ using min–max normalization scaling across each feature:

$$norm_i = \frac{x_i - min(x_i)}{max(x_i) - min(x_i)}(b - a) + a \tag{27}$$

where $[a, b]$ is the desired rescaled range. In this case a, b are equal to unit ($a = -1, b = 1$). The same normalization technique was applied to the outputs. The minimum and maximum values of the training set were used to normalize also the validation and testing set to avoid data leakage. Fig. 5 illustrates the architecture of the proposed CNN model. A CNN2D is applied with 3 groups of layers and the MLP. Each group is composed of a convolution layer, a max pooling layer, and a dropout layer, with the latter one to be used for both regularization and estimation of the epistemic uncertainty at the testing phase. The output of the last group of layers is flattened into a 1-dimensional array to be fed into the MLP which consists of 3 FC layers with 4096, 8, and 1 neuron, respectively. The last neuron represents the model's output, i.e. the strength prediction given the 3D input. Table 2 summarizes the input dimensions of each layer and the hyperparameters that are needed for each layer. An important remark is that the dropout rate after the input layer is equal to 0.2 and lower than the dropout rates for the hidden layers as dropping the input neurons, and thus, the input data massively may adversely affect training. Lastly, the weights for each layer's neuron were initialized by the standard normal distribution.

After the investigation of different nonlinear activation functions that could fit the current problem, such as Softplus, ReLU, LeakyReLU, Tanh, Sigmoid, and ELU (Fig. 6), the one with the optimal performance which was applied after each convolutional and FC hidden layer, is the Softplus. For the output layer, the Tanh activation function is employed as the SFEM guarantees that the testing set is a sub-space encapsulated in the training set. Therefore, none of the experimental values are expected to lie outside the training space, i.e. out of the range $[-1, 1]$, created by the Tanh. Because tuning the hyperparameters via trial and error is a time-consuming and error-prone process, every hyperparameter was tuned in an automatic way via the Bayesian optimization process [67]. Table 3 depicts the chosen range of the hyperparameters to be included for the Bayesian optimization and their optimized values to be used for the training process. After the optimization, the values were rounded to the closest integers as the optimized values could be float numbers. For memory optimization [68], mini-batches, neurons, and filters were rounded to the closest power-of-two value. It should be noted that because the first FC layer that comes after the flattening layer accepts the flattened output of the convolutional and pooling layers, it must have a number of neurons equal to that output, therefore it should be excluded from Bayesian optimization. Furthermore, increasing the ranges of hyperparameters, requires a larger computation time for the algorithm to find the optimal hyperparameters, thus an appropriate rate has been decided.

Table 4 presents the values of the additional hyperparameters that do not belong to the model's architecture but are mandatory to train the CNN model. Some of them were optimized via the Bayesian optimization (learning rate, mini-batches, epochs) and the others were tuned manually. In deep learning, using only one sample at a time to update the parameters is not efficient for applying a gradient descent algorithm, and it is shown that the performance is increased remarkably by using a small set of samples at a time called mini-batch [69] which has a value of 32 in the current case. Moreover, the model updates its parameters with a learning rate equal to $5 \cdot 10^{-4}$ and the entire dataset is passed 200 times which represents the epochs hyperparameter. It should be noted that after 50 subsequent epochs where the validation loss remains constant, the training is terminated to avoid a potential overfitting. This process is called early stopping [70] and is exploited when the model needs to be trained for many epochs without knowing when the loss will stop decreasing. Thus, each number of epochs that is tried during the Bayesian optimization represents the maximum possible times that the model should observe the whole dataset. Finally, the model is trained using the Adam optimizer [71] by minimizing the mean square error (MSE).

5. Results and discussion

The current section is divided in six subsections which are in line with the six steps that are followed during the proposed methodology and presents the functionality of the method for the strength prediction of the aforementioned quasi-isotropic laminate. Hence, indicative results are presented in a stepwise way for each described step.

Table 2
Input dimensions of each layer of the current CNN model architecture.

Layer name	Output shape (MB, H, W, F)	Hyperparameters
Input layer	(32, 6, 60, 2)	–
Convolutional layer 1	(32, 13, 56, 256)	Filters=256, kernel size=(2,5), strides=1, pad=4
Max pooling layer 1	(32,12,27,256)	Kernel size=(2, 4), strides=(1,2), pad=0
Batch normalization, Dropout rate	(32, 12, 27, 256)	–, 0.2
Convolutional layer 2	(32, 11, 25, 128)	Filters=128, kernel size=(2,3), strides=1, pad=0
Max pooling layer 2	(32, 10, 12, 128)	Kernel size=(2,3), strides=1, pad=0
Batch normalization, Dropout rate	(32, 10, 12, 128)	–, 0.5
Convolutional layer 3	(32, 9, 10, 64)	Filters=64, kernel size=(2,3), strides=1, pad=0
Max pooling layer 3	(32, 8, 8, 64)	Kernel size=(2,3), strides=1, pad=0
Flattening layer	(32, 4096)	–
FC layer 1	(32, 4096)	Neurons=4096
Dropout rate	(32, 4096)	0.5
FC layer 2	(32, 8)	Neurons=8
Dropout rate	(32, 8)	0.5
Output Layer	(32, 1)	Neurons=1

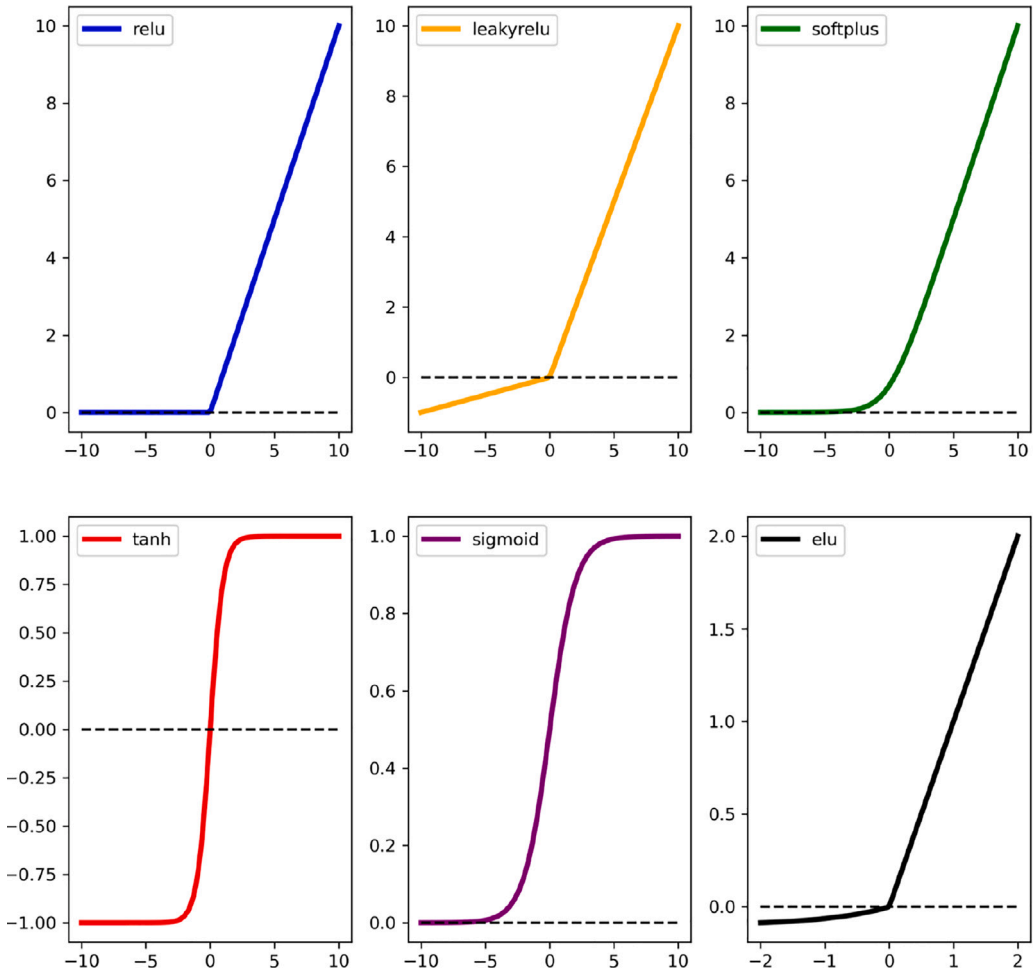


Fig. 6. Different activation functions that were tested during the development of the DL model.

Table 3
Input dimensions of each layer of the current CNN model architecture.

	Learning rate	Mini-batches	Epochs	Neurons for 2nd FC layer	Filters per convolutional layer	Kernel size per layer
Range	$10^{-5} - 10^{-2}$	8–128	100–700	4–128	512–1024, 64–256, 16–128	(1, 1)–(10, 10)
Optimized value	$5 \cdot 10^{-4}$	32 (28)	200	8 (10)	256 (247), 128 (151), 64 (53)	(2, 5), (2, 4), (2, 3), (2, 3), (2, 3), (2, 3)

5.1. Step 1. Material characterization

First of all, the initial mechanical properties regarding the lamina level of the investigated structure should be defined in terms of mean value and deviation as well. These values consist the input for the stochastic finite element analysis model. Material characterization experiments were conducted in order to extract the ply properties of the Hexply® F6376C-HTS(12 K)-5-35 prepreg used for the manufacturing

Table 4
Values of additional hyperparameters used for training and evaluating the proposed CNN model.

Learning rate	0.0005
Mini-batches	32
Epochs	200
Early stopping (after No. epochs)	50
Train/Validation (%)	90/10
Optimizer	Adam
Loss	MSE

Table 5
Elastic and strength properties obtained from material characterization experiments for the Hexply® F6376C-HTS(12 K)-5-35.

Property type (Units)	No. of Specimens	Mean value	Deviation
E_{11} (GPa)	11	143.7	18.4
E_{22} (GPa)	7	9.2	2.0
G_{12} (GPa)	5	5.1	0.7
υ_{12} (−)	11	0.37	0.14
X_T (MPa)	11	1924	146.9
Y_T (MPa)	7	107.6	9.1
S (MPa)	5	96.3	0.8

of the quasi-isotropic laminated specimens with lamination of $[(0/90/\pm 45)_s]_2$. The number of specimens, the mean values and deviations of each property extracted by the material characterization process are enlisted in Table 5.

In order to emphasize the inherent uncertainty of the composite materials, the current work provides the cumulative distribution functions (CDFs) of the mechanical properties, acquired by the material characterization procedure. Fig. 7 shows the CDF curves of the elastic modulus on the fiber (E_{11}), matrix (E_{22}) and shear (S) direction respectively. On the same way, Fig. 8 depicts the CDF curves of the tensile strengths on the fiber (X_T), matrix (Y_T) and shear direction (S). The confidence bounds are larger on the cases of E_2, Y_T, G_{12} and S in comparison with the case of E_1, X_T, due to the smaller amount of tested specimens.

5.2. Step 2. Stochastic distribution of the material properties

The inherent uncertainty of the composite materials, which was shown in Section 5.1 during the material characterization procedure, motivates the authors to implement the SFEM in order to distribute this uncertainty along the structural domain. The exploitation of the K–L expansion ensures that each mechanical property is distributed in a random way by defining the mean values, the deviation, the size of the RF mesh and the rest K–L parameters already mentioned in Table 1, Table 5. Figs. 9, 10 demonstrate the random distribution of the mechanical properties on a ply level for a θ sample, which is arbitrarily selected. In practice, each ply and each sample θ are described by a different stochastic distribution.

5.3. Step 3. Probabilistic finite element analysis

After distributing the uncertainty of the mechanical properties into the Gauss integration points for each ply and for each sample θ, the stochastic stiffness matrices are calculated for each sample by the Eq. (10). The axial strains ϵ_x at levels 10%, 20% of the maximum strain are calculated by the stochastic FEA and are exploited as a part of the input dataset for the training process. In addition, the strength values of the investigated structure are calculated for each virtual specimen θ by utilizing the Puck's failure criterion and by considering the strength as the load in which the last-ply-failure exists on each case. Fig. 11 shows the calculated strain values ϵ_x along the domain of an arbitrarily selected θ specimen at 10% and at 20%. Fig. 12 depicts the CDF curve of the strength values, which is exploited as a training data set on the DL algorithm as well.

5.4. Step 4. The learning process of the deep learning model

During the current step, the axial strain and the strength values obtained by the SFEM are employed as the training dataset for the learning process. The learning process is depicted in Fig. 13. In particular, the training and validation of the MSE loss (Fig. 13(a)), and the average strength prediction error in MPa for the train and validation set of the simulation data (Fig. 13(b)) are presented. To compute the average strength prediction error, the absolute value after the substraction between the real and predicted strengths was considered and then the mean of this value was computed. One may notice that the validation loss is constantly lower than the training loss before they are converged. This is rational since dropout, which is a regularization technique, has been used to avoid overfitting by increasing the stochasticity of training process. Since training and validation sets converge to similar errors in the learning curves, one could agree that the learning process is successful. Indeed, after 200 epochs none of the model's average predictions over a mini-batch of unseen simulation data is more than ±4 MPa (Fig. 13(b)) of the corresponding true simulated strengths. However, an increase in the prediction error is expected for the experimental data at the testing phase due to the sim-to-real deviations which lead to the aforementioned dataset shift.

5.5. Step 5. Measuring the experimental strains

After the completion of the training process on Step. 4, the DL model is functional and capable of predicting the strength value of the quasi-isotropic laminate, once the 10% and 20% strain measurements acquired by the non destructive tensile experiment. To demonstrate the validity and the repeatability of the proposed methodology, 10 tensile experiments with 1 mm/min loading rate were conducted and the full-field strain measurements were acquired by the DIC. Fig. 14 shows the speckle pattern of a specimen clamped into the tensile machine. A pair of 5 Megapixel cameras with 23 mm lens and 75 frames-per-second was placed in the front side of the specimen. The post-processing was performed using the commercial software VIC-2D® by Correlated Solutions. A subset size of 29 pixels and step size of 7 pixels were selected for correlation analysis and the obtained strains were stored in a 6×60 matrix.

Fig. 15 depicts the axial strains ϵ_x measured by the DIC on 10%, 20% strain levels respectively. The strains acquired near the boundaries are excluded, as their values diverge highly and relegate the performance of the DL algorithm.

5.6. Step 6. Prediction of the strength distribution

Step. 6 is the last step of the proposed methodology where the full-field axial strains, obtained by the tensile experiments are inserted in the DL algorithm for the prediction of the strength distribution. Fig. 16 presents the model's predictions for each of the 10 specimens which were previously subjected to tensile experiments. The presented predictions are satisfying inside an acceptance ratio (tolerance), which is ±2% from the true experimental strength of each specimen. Undoubtedly, an increase in the error for some of the experiments is detected compared to the training/validating set. This increase is due to the variations between the training and testing distributions (dataset shift). The proposed model captures this variation as an epistemic uncertainty using the MC dropout. In this regard, 200 MC samples produced 200 predictions for each test sample to form the corresponding probability density distributions functions via the kernel density estimator technique using a bandwidth according to the Scott's rule [72]. Although in some experimental cases (specimen 1, 3, 7, 8) some of the model's predictions may be out of the predefined tolerance, the model still predicts a satisfying average strength and simultaneously, provides a wider distribution around the mean values. This indicates that the distributed

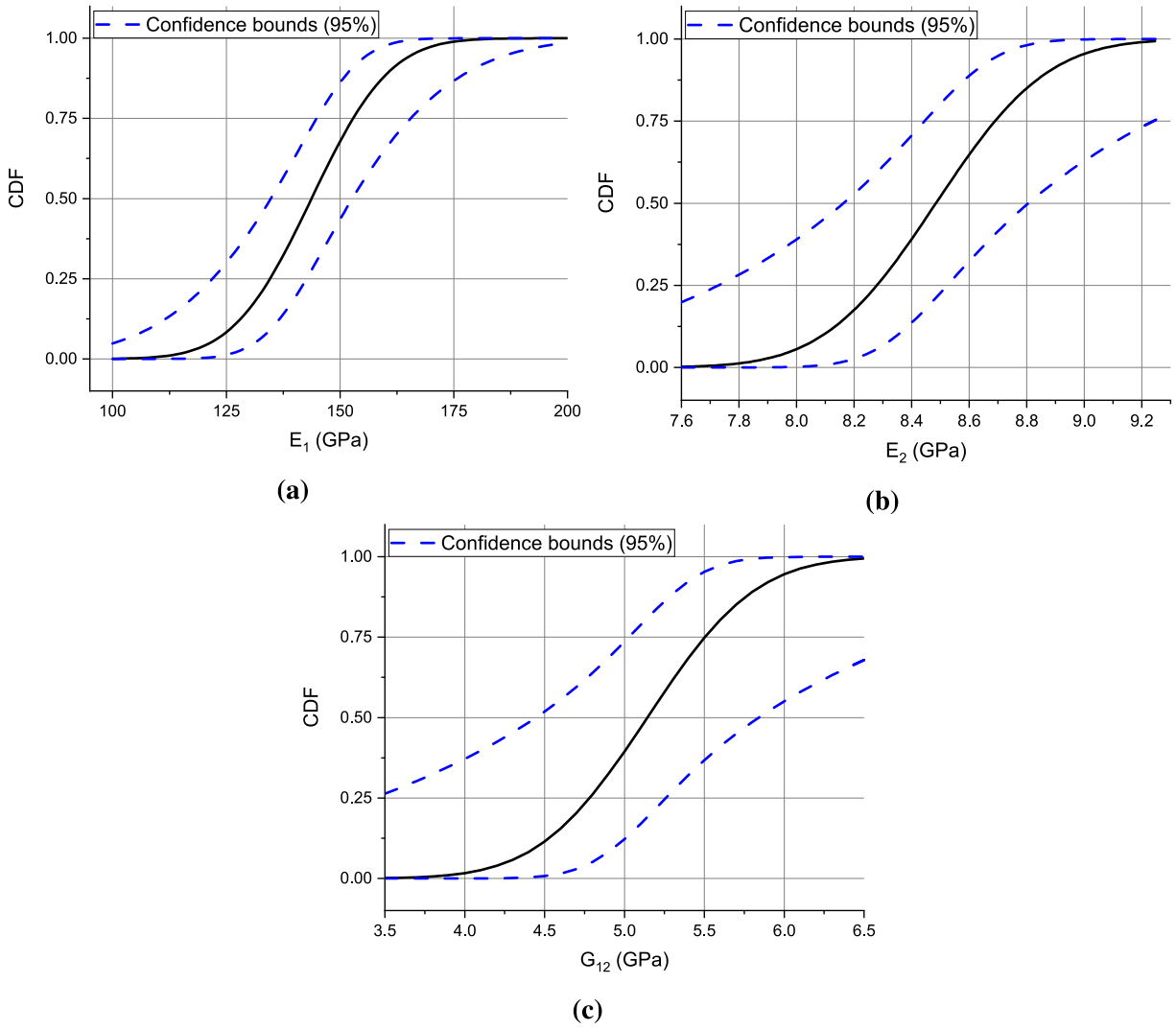


Fig. 7. Cumulative distribution function curves including confidence bounds (95%) of the lamina elastic properties: (a) Fiber modulus, E_{11}; (b) Matrix modulus E_{22}; (c) Shear modulus G_{12}.

strains of the corresponding experiments shifts additionally from the distributed simulated samples, which have been previously used for training. Although the model's predictions even in these extreme cases are able to approximate the experimental strengths, if the deviation between the simulation and the experimental sample is further increased, then the epistemic uncertainty would be also increased which is an indicator of not trusting the corresponding model's average strength prediction. Consequently, the uncertainty estimation further improves the reliability to the proposed ANN model. Finally, the 95% confidence intervals for each probability distribution were calculated using the statistical measure of variance and were interpreted as strengths; 95% of the predictions fall between the corresponding strength range (in MPa) around the mean values of strength.

Since the ANN's predictions are based on its ability to detect the relationship between the inputs (strains) and the outputs (strengths), a regression problem exists. In other words, the ANN is seeking a mapping function between the strains and strengths. To evaluate how accurate this mapping function is, the regression curves of the results are presented. Fig. 17 shows the predicted versus the simulated curves for the training (Fig. 17(a)), validation sets (Fig. 17(b)), and the predicted versus the experimental curve (Fig. 17(c)). For evaluating the training, validating, and the testing processes, all of the available samples are used, hence 900, 100, and 10 samples, respectively. It is observed that the correlation coefficient (R^2) values for the training, validating, and testing sets between predicted and true strengths were recorded as 0.972, 0.9739 and 0.7992 respectively. This indicates that the goodness of fit of the proposed model is close to the real values. Indeed, the regression line between the experimental and the predicted average strengths approximates satisfactorily the diagonal line (perfect fit) even in the testing set. Although a relatively large drop is shown in R^2 in the testing phase, this is not always a caveat as it indicates that there is a barrier that prevents the predicted and simulated strengths to be perfectly correlated. Interestingly, this barrier is produced by the large variations between the training and testing sets, i.e. epistemic uncertainty is present. As a result, during the testing process, the regression curve appears with predictions under epistemic uncertainty. Therefore, error bars were added that represent the uncertainty by using the corresponding confidence intervals mentioned before.

Table 6
Statistical measures for train, validation, and test phases.

Statistical Measures	MAE (MPa)	RMSE (MPa)	R^2
Equation	Eq. (28)	Eq. (29)	Eq. (30)
Train	5.0867	5.181	0.972
Validation	5.3417	5.5231	0.9739
Test	8.102	8.3066	0.7992

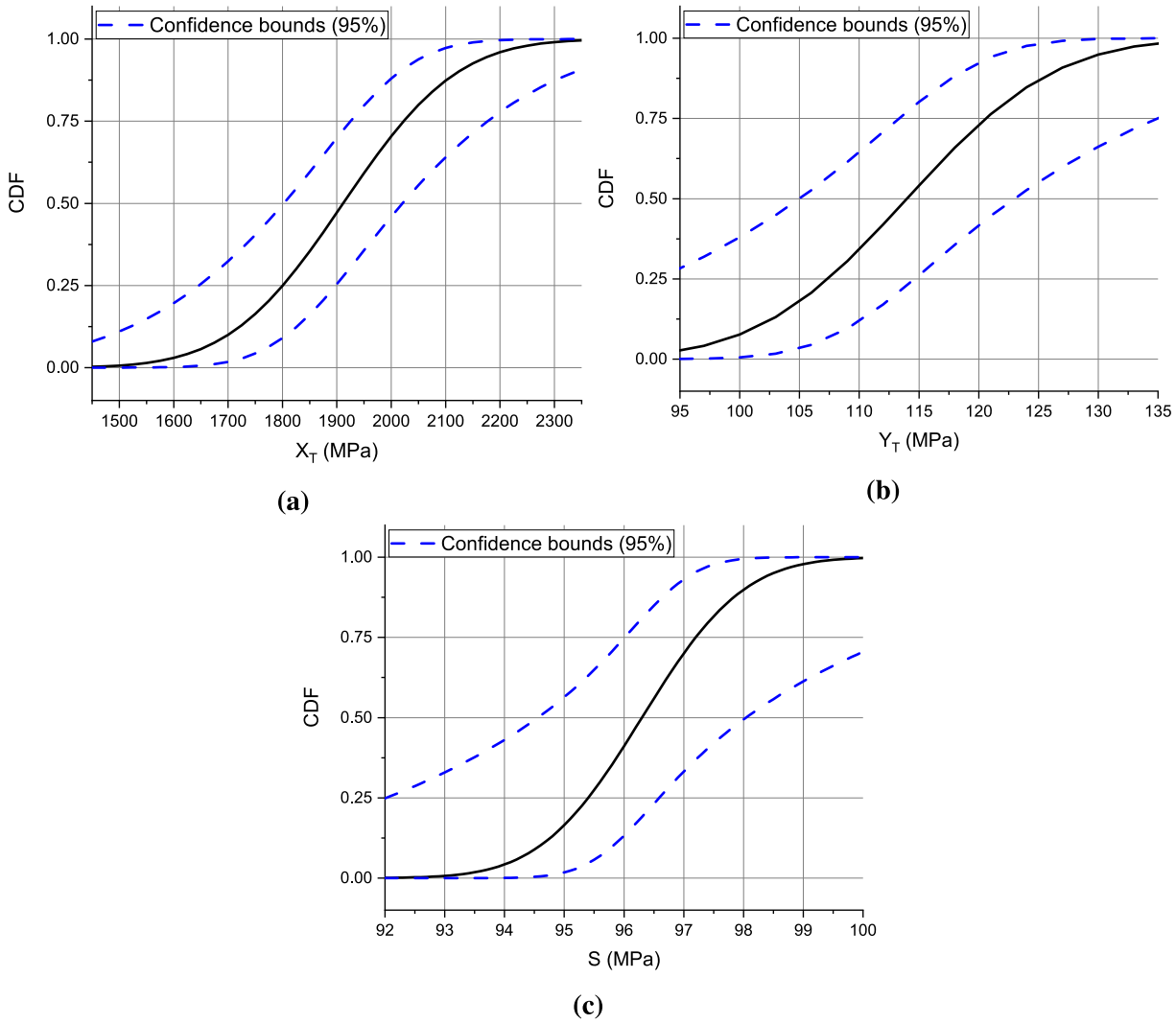


Fig. 8. Cumulative distribution function curves including confidence bounds (95%) of the lamina strength properties: (a) Tensile fiber strength X_T; (b) Tensile matrix strength Y_T; (c) Tensile shear strength S.

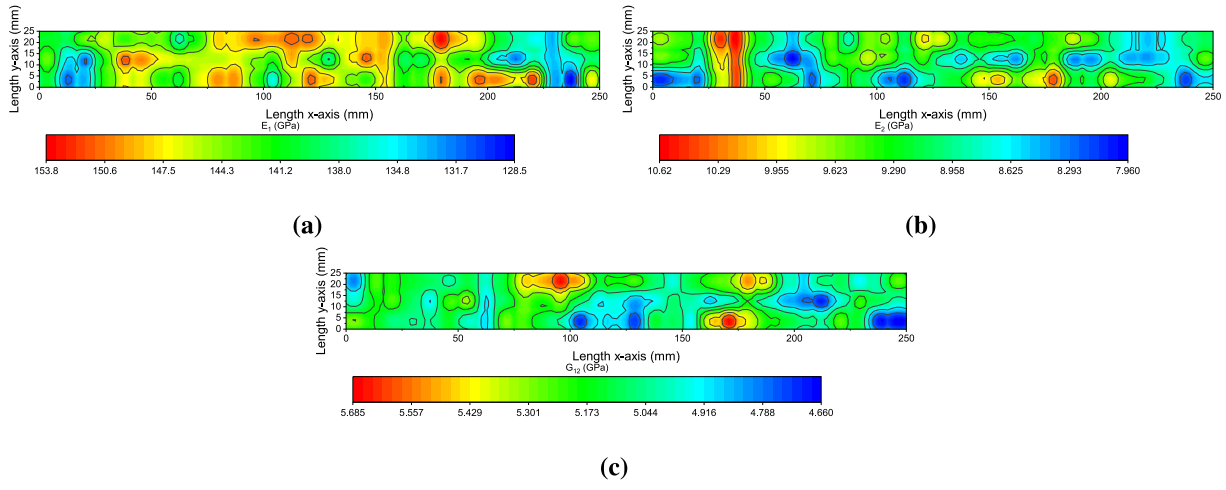


Fig. 9. The random distribution resulted by the K–L expansion of: (a) E_1; (b) E_2; (c) G_{12} moduli values on the Gauss integration points of the simulated structure.

Despite the useful information that is given by R^2, it cannot be used as a standalone metric for the model evaluation as it indicates the correlation between the examined variables. It is very common, as it is also observed in this study, that a smaller value of R^2 may come up with low prediction errors in the model. In this regard Table 6 summarizes two additional important statistical measures alongside R^2;

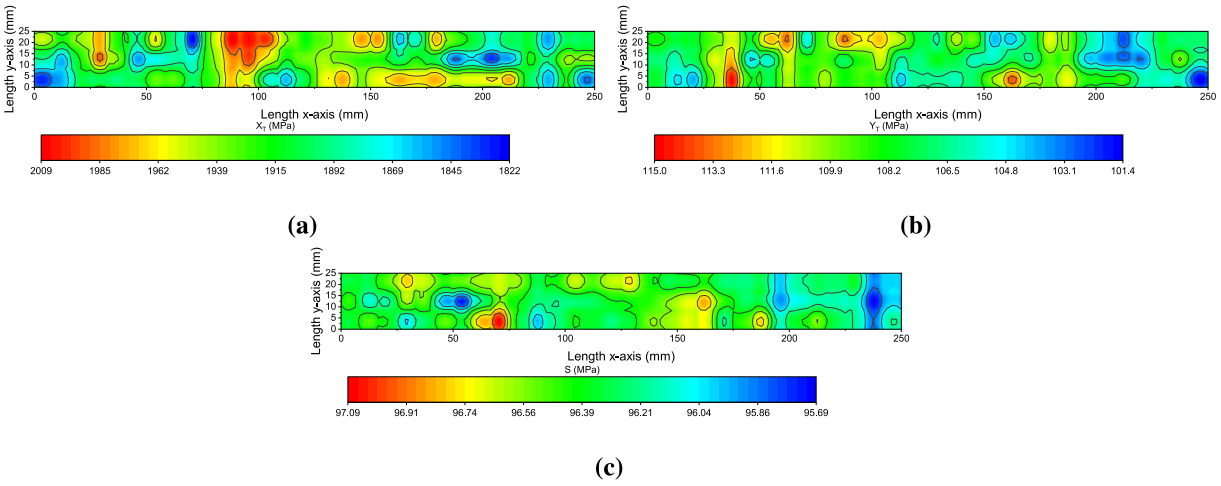


Fig. 10. The random distribution resulted by the K–L expansion of: (a) X_T; (b) Y_T; (c) S strength values on the Gauss integration points of the simulated structure.

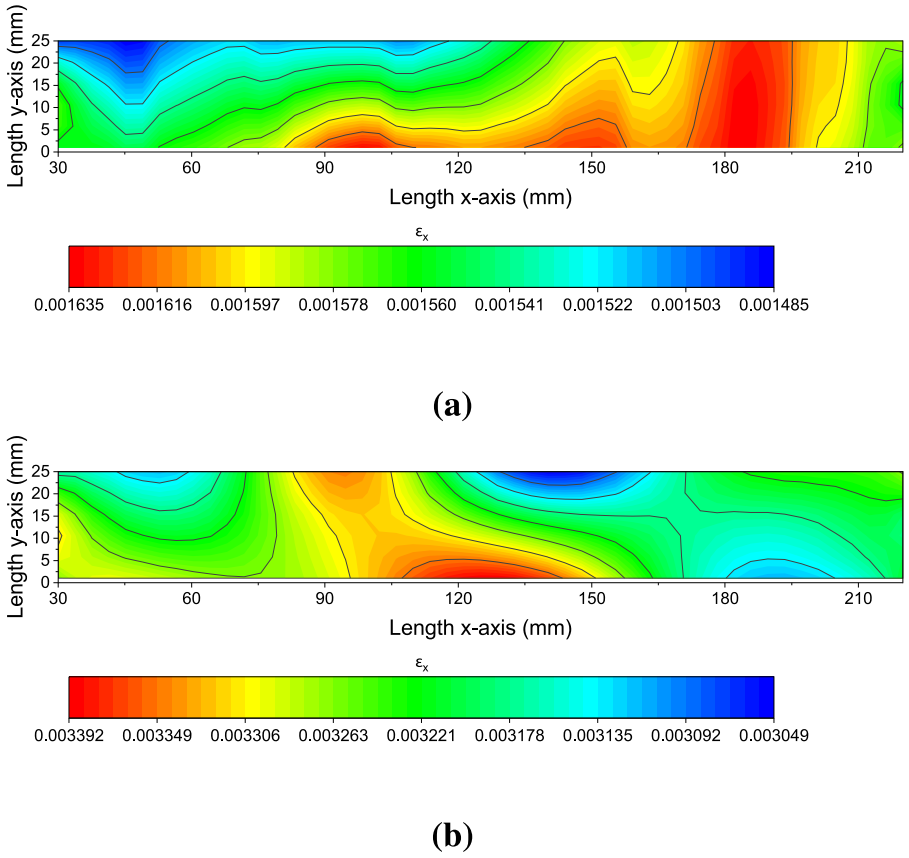


Fig. 11. Stochastic distribution of strain values of the investigated quasi-isotropic laminate calculated by the stochastic finite element model at: (a) 10%; (b) 20% strain level under tensile load.

the mean absolute error (MAE), and root MSE (RMSE). According to these metric measures, the RMSE indicates that the standard deviation of the predicted strength errors is fairly low, while the MAE provides an easy-to-understand measurement of the model's strength predictions. One could notice a difference in MAE of train/validation sets from the ones in Fig. 13(b) after 200 epochs. This is reasonable since a smaller set (mini-batches) for visualizing the learning process is utilized than the set for evaluating the model's training and validating performances, where the whole dataset has been considered for extracting those statistical measures. Consequently, the final MAE measurements provide a better insight of the learning process and they are approximately 5.09 MPa and 5.34 MPa for the training and validating sets, respectively. As expected in testing phase, the MAE has increased by approximately 4 MPa on average compared with the predictions during the train/validation phase, however, the average predictions still lie inside the ±2% acceptance ratio. This further ensures the efficiency of the proposed model.

$$\frac{1}{n}\sum_{j=1}^{n}\left|y_j-\hat{y}_j\right| \tag{28}$$

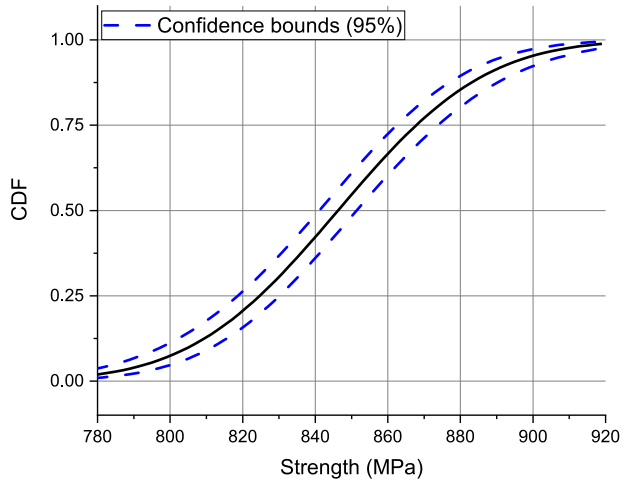


Fig. 12. Cumulative distribution function curve for the strength values of the investigated quasi-isotropic laminate calculated by the stochastic finite element method.

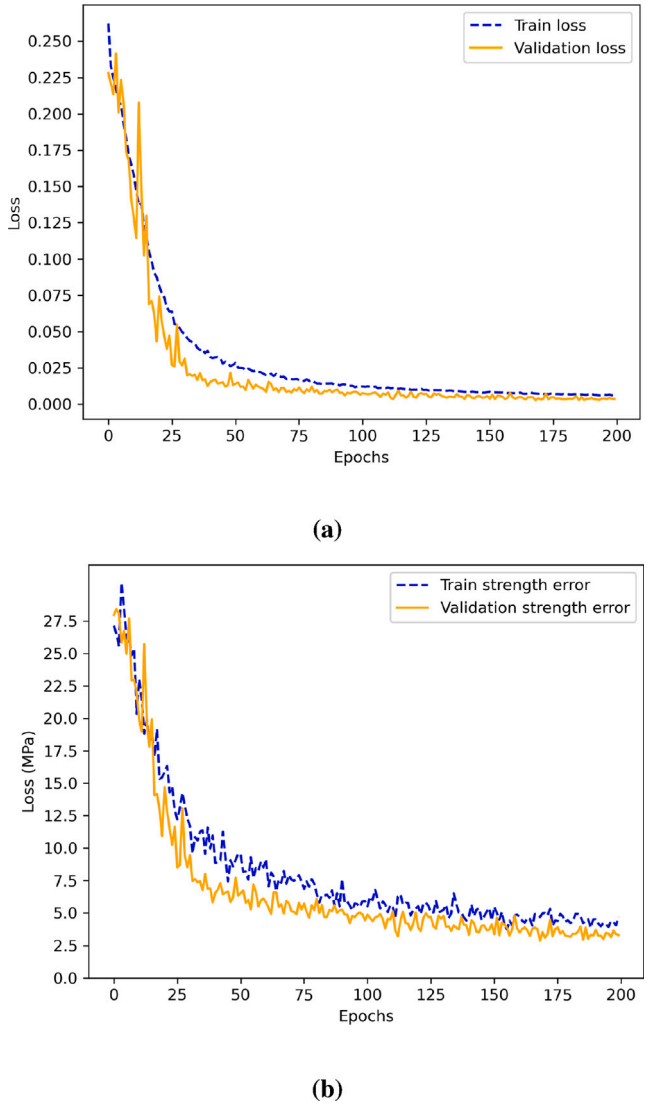


Fig. 13. Learning curves using simulation data. (a) Training and validation MSE loss; (b) Average strength prediction error for the two sets given the true strengths of each component.

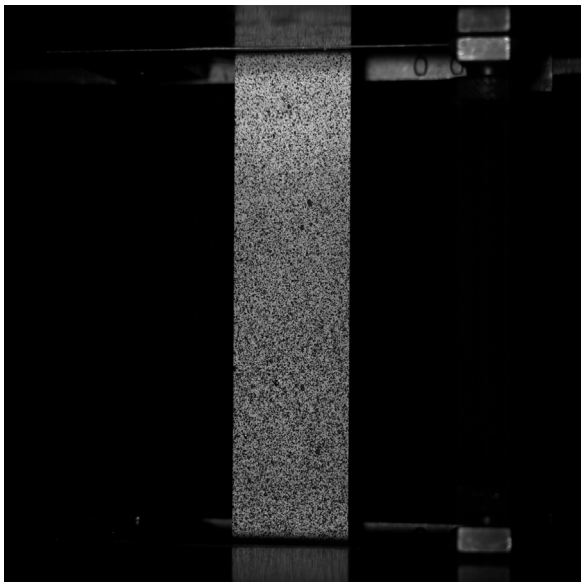


Fig. 14. The speckle pattern of a quasi-isotropic laminate clamped into the tensile machine.

$$\sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j-\hat{y}_j)^2} \tag{29}$$

$$1-\frac{\sum_{j=1}^{n}(y_j-\hat{y}_j)^2}{\sum_{j=1}^{n}(y_j-\bar{y}_j)^2} \tag{30}$$

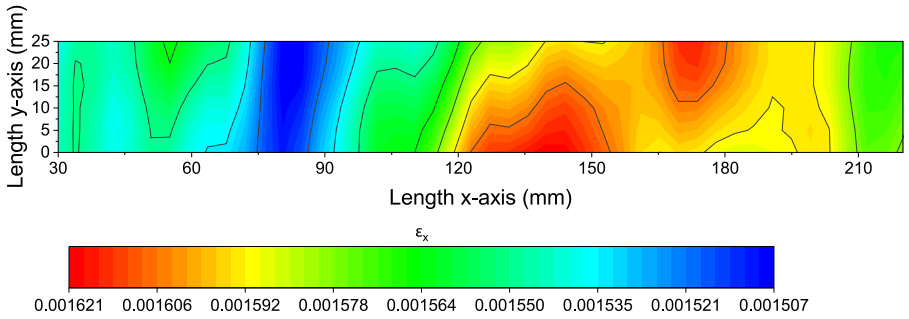
5.7. Limitations

It is important to note the limitations that exist in the current work for both the SFEM and the DL model. In regards to the SFEM, which is used to generate data for the training of the DL model, the considered stochasticity comes only from the material characterization process at the mesoscale level. This means that the loading and boundary conditions are considered equal and deterministic for all samples while during testing deviations may occur. Additionally, although SFEM allows to integrate several mesoscale failure criteria, it was selected to employ only Puck and thus the effect of other failure criteria on the strength predictions was not examined/investigated. However, the investigation of the efficiency of each criterion is outside the scope of the current work.

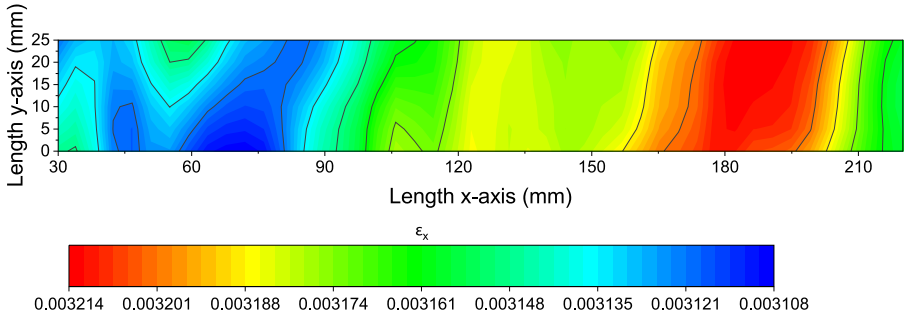
In addition to the limitations of the SFEM, the proposed DL model also has its own limitations. Firstly, while the proposed integration of uncertainty quantification into the ANN provides a detailed strength prediction by giving the entire distribution around each estimation, this uncertainty cannot be automatically reduced after the training process. Thus, it will also affect future predictions. Secondly, the training has been applied in an offline manner rather than online, which could include an adaptive learning procedure. Continuously updating the model with new data is crucial for improved strength predictions and is considered as future work.

6. Conclusions

The current work achieved to predict the probabilistic strength distribution of laminated CFRP specimens via non-destructive tensile tests up to 20% of maximum strain and via the advantageous contribution of high-fidelity stochastic finite element models and DL techniques. More specifically a DL algorithm consisted of CNNs was developed to predict the strengths given the corresponding strains as inputs. On the



(a)



(b)

Fig. 15. Experimental axial strains (ϵ_x) acquired by the DIC at 10%, 20% of the quasi-isotropic laminate under tensile test.

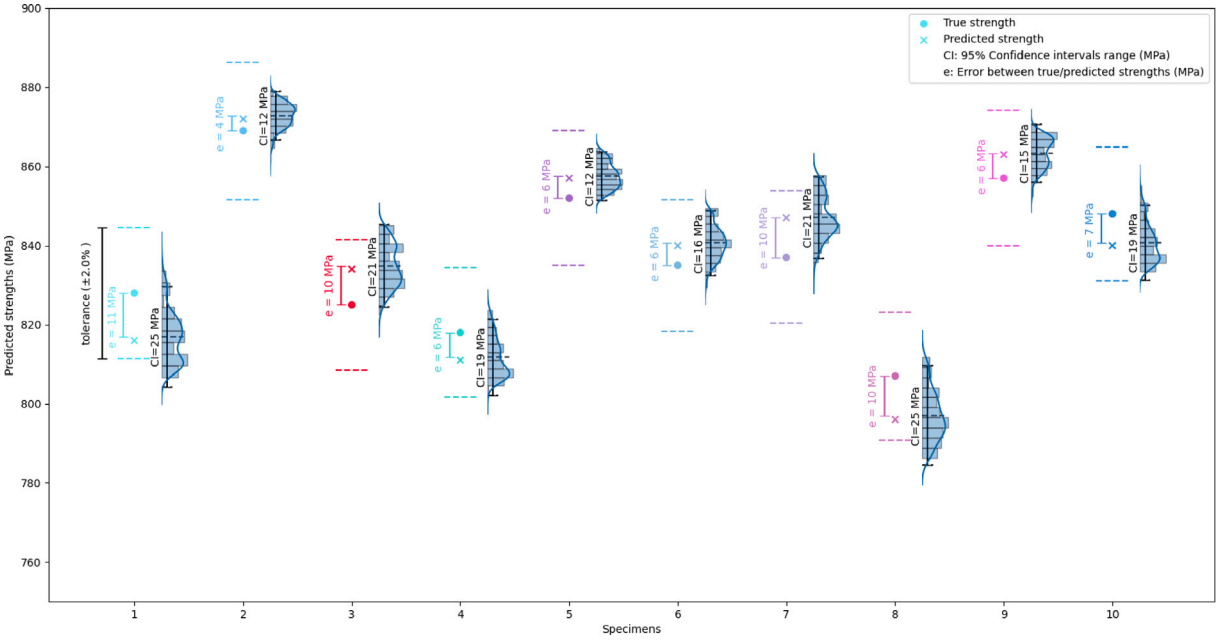


Fig. 16. Model's predictions on experimental data (testing phase). Using the MC dropout during inference 200 predictions for each test sample were made to produce the corresponding distributions that capture the epistemic uncertainty. The 95% confidence intervals were calculated using the statistical variance and are interpreted as strengths in MPa. Every average prediction is inside the range of $[-2\%, 2\%]$ of the true strength.

other hand, stochastic finite element models were constructed for the probabilistic failure analysis of quasi-isotropic CFRPs, employed as the training dataset.

The exploitation of SFEM leads to the prediction of probabilistic strain distributions and strength values, in an efficient way that encompasses the uncertainties induced by the manufacturing process of the

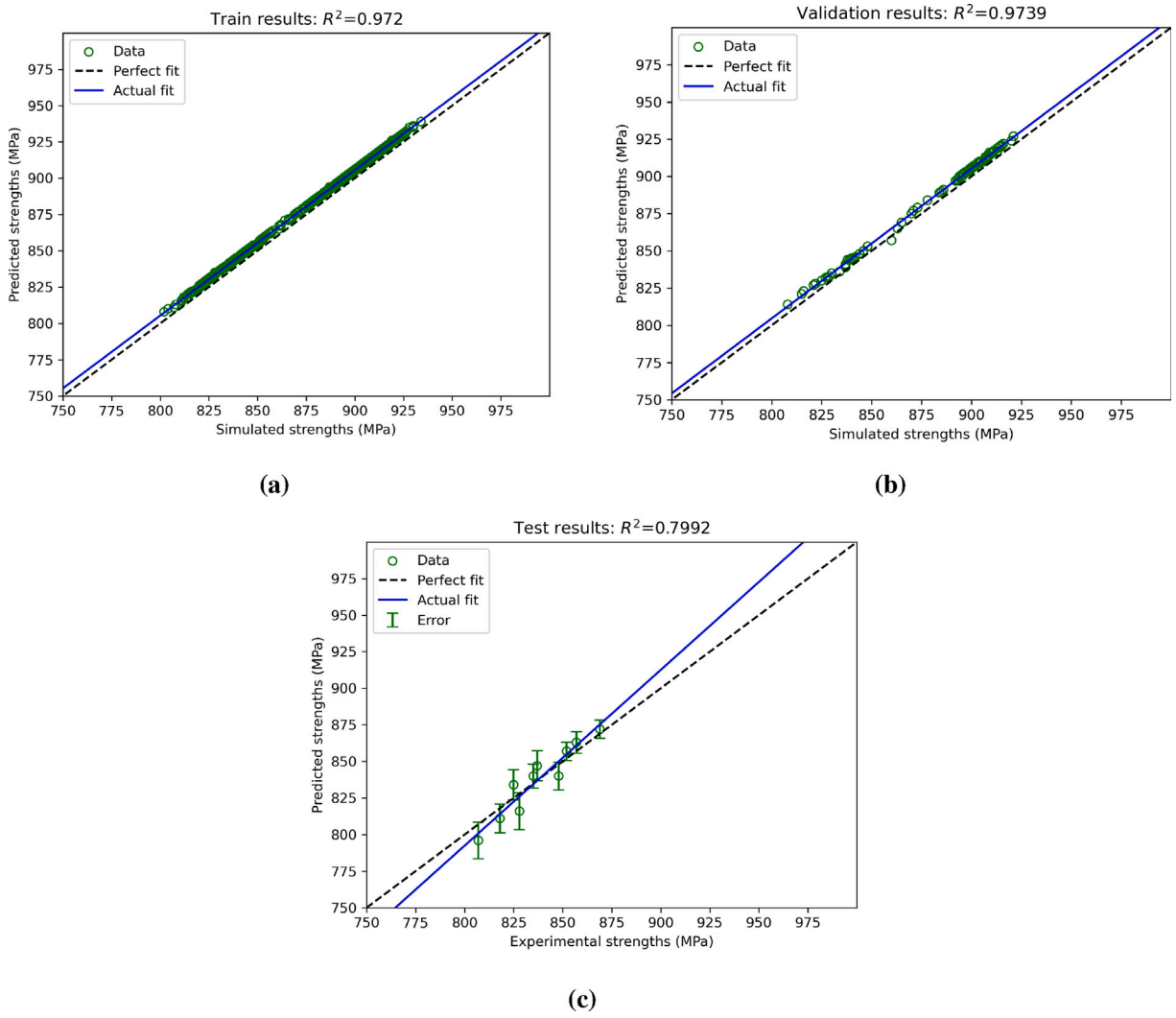


Fig. 17. Regression curves between real strengths and predicted average strengths for: (a) train; (b) validation; (c) test phases.

composites. Its utility was presented in terms of axial strains induced by tensile loading for the stochastic FEA and the experimental cases as well. The validity of the SFEM in comparison with the experiments, enhances the training process of the DL algorithm and consequently its robustness. Five key advantages were demonstrated: (1) a robust DL algorithm was developed by the exploitation of CNNs and a state-of-art uncertainty estimation technique concerning the MC dropout method; (2) the appealing efficiency of the SFEM including the K–L expansion and the LHS was utilized in order to encompass the stochastic nature of composite materials and to decrease the computational demand; (3) a material characterization process for the Hexply® F6376C-HTS(12 K)-5-35 prepreg material was conducted and mean values, deviation and CDFs curves of the material properties were inserted to the SFEM for probabilistic failure analysis; (4) the developed CNN was trained with numerical data and tested with experimental, which is observed in limited works on the literature due to the epistemic uncertainty induced by the dataset shift. The proposed methodology captures successfully this uncertainty due to the capability of SFEM to bound the calculated strains within the experimental range, thus limiting the dataset shift and subsequently the integration of MC dropout technique into the CNN; (5) the proposed methodology is able to predict the strength of a composite structure by avoiding the catastrophic experiments which avoids the sacrifice of manufactured specimens.

In closing, the proposed methodology has the potential to reduce substantially the intense experimental campaigns conducted for strength characterization in composite structures, since it is based on well-established physics-based numerical models and robust deep learning techniques. This fact, reveals the outstanding potential of the method and its significant impact on the reduction of carbon emissions, production time and costs which consist a cornerstone on the sustainability of composite materials and structures.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CRediT authorship contribution statement

Christos Nastos: Conceptualization, Methodology, Software, Validation, Data curation, Writing – original draft. **Panagiotis Komninos:** Conceptualization, Methodology, Software, Validation, Data curation, Writing – original draft. **Dimitrios Zarouchas:** Conceptualization, Methodology, Writing – review & editing, Resources, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors greatly appreciate George Peppas and Tatiana Avloniti for their contribution on the tensile experiments of the quasi-isotropic specimens during their Erasmus placement in the Aerospace Engineering Faculty, Delft University of Technology.

References

[1] Raghavan Ajay, Cesnik Carlos ES. 3-D elasticity-based modeling of anisotropic piezocomposite transducers for guided wave structural health monitoring. J Vib Acoust 2007;129:739–51.
[2] Yu Lingyu, Bottai-Santoni Giola, Giurgiutiu Victor. Shear lag solution for tuning ultrasonic piezoelectric wafer active sensors with applications to lamb wave array imaging. Internat J Engrg Sci 2010;48(10):848–61.
[3] Ahmad ZAB, Vivar-Perez Juan Miguel, Gabbert Ulrich. Semi-analytical finite element method for modeling of lamb wave propagation. CEAS Aeronaut J 2013;4(1):21–33.
[4] Marzani Alessandro, Viola Erasmo, Bartoli Ivan, Di Scalea Francesco Lanza, Rizzo Piervincenzo. A semi-analytical finite element formulation for modeling stress wave propagation in axisymmetric damped waveguides. J Sound Vib 2008;318(3):488–505.
[5] Bathe Klaus-Jürgen. Finite element procedures. Klaus-Jurgen Bathe; 2006.
[6] Rekatsinas CS, Nastos CV, Theodosiou TC, Saravanos DA. A time-domain high-order spectral finite element for the simulation of symmetric and anti-symmetric guided waves in laminated composite strips. Wave Motion 2015;53:1–19.
[7] Kudela Paweł, Krawczuk Marek, Ostachowicz Wiesław. Wave propagation modelling in 1D structures using spectral finite elements. J Sound Vib 2007;300(1–2):88–100.
[8] Nastos CV, Theodosiou TC, Rekatsinas CS, Saravanos DA. A 2D daubechies finite wavelet domain method for transient wave response analysis in shear deformable laminated composite plates. Comput Mech 2018;62(5):1187–98.
[9] Nastos Christos V, Saravanos Dimitris A. Multiresolution Daubechies finite wavelet domain method for transient dynamic wave analysis in elastic solids. Internat J Numer Methods Engrg 2021;122(23):7078–100.
[10] Stefanou George. The stochastic finite element method: Past, present and future. Comput Methods Appl Mech Engrg 2009;198(9–12):1031–51.
[11] Argyris John, Papadrakakis Manolis, Stefanou George. Stochastic finite element analysis of shells. Comput Methods Appl Mech Engrg 2002;191(41–42):4781–804.
[12] Liu Wing Kam, Belytschko Ted, Mani A. Probabilistic finite elements for nonlinear structural dynamics. Comput Methods Appl Mech Engrg 1986;56(1):61–81.
[13] Kamiński Marcin, Kleiber Michal. Perturbation based stochastic finite element method for homogenization of two-phase elastic composites. Comput Struct 2000;78(6):811–26.
[14] Nastos Christos, Zarouchas Dimitrios. Probabilistic failure analysis of quasi-isotropic CFRP structures utilizing the stochastic finite element and the Karhunen–Loève expansion methods. Composites B 2022;235:109742.
[15] Chopra Palika, Sharma Rajendra Kumar, Kumar Maneek, Chopra Tanuj. Comparison of machine learning techniques for the prediction of compressive strength of concrete. Adv Civ Eng 2018;2018.
[16] Feng De-Cheng, Liu Zhen-Tao, Wang Xiao-Dan, Chen Yin, Chang Jia-Qi, Wei Dong-Fang, et al. Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. Constr Build Mater 2020;230:117000.
[17] Chou Jui-Sheng, Tsai Chih-Fong. Concrete compressive strength analysis using a combined classification and regression technique. Autom Constr 2012;24:52–60.
[18] Chou Jui-Sheng, Tsai Chih-Fong, Pham Anh-Duc, Lu Yu-Hsin. Machine learning in concrete strength simulations: Multi-nation data analytics. Constr Build Mater 2014;73:771–80.
[19] Yaseen Zaher Mundher, Deo Ravinesh C, Hilal Ameer, Abd Abbas M, Bueno Laura Cornejo, Salcedo-Sanz Sancho, et al. Predicting compressive strength of lightweight foamed concrete using extreme learning machine model. Adv Eng Softw 2018;115:112–25.
[20] Hossain Khandaker, Anwar Muhammed S, Samani Shirin G. Regression and artificial neural network models for strength properties of engineered cementitious composites. Neural Comput Appl 2018;29(9):631–45.
[21] Deng Fangming, He Yigang, Zhou Shuangxi, Yu Yun, Cheng Haigen, Wu Xiang. Compressive strength prediction of recycled concrete based on deep learning. Constr Build Mater 2018;175:562–9.
[22] Mangalathu Sujith, Jeon Jong-Su. Classification of failure mode and prediction of shear strength for reinforced concrete beam-column joints using machine learning techniques. Eng Struct 2018;160:85–94.
[23] Abuodeh Omar R, Abdalla Jamal A, Hawileh Rami A. Prediction of shear strength and behavior of RC beams strengthened with externally bonded FRP sheets using machine learning techniques. Compos Struct 2020;234:111698.
[24] Karina CN, Chun Pang-jo, Okubo Kazuaki. Tensile strength prediction of corroded steel plates by using machine learning approach. Steel Compos Struct 2017;24(5):635–41.
[25] Yan Wangchen, Deng Lu, Zhang Feng, Li Tiange, Li Shaofan. Probabilistic machine learning approach to bridge fatigue failure analysis due to vehicular overloading. Eng Struct 2019;193:91–9.
[26] Hoang Nhat-Duc, Pham Anh-Duc, Nguyen Quoc-Lam, Pham Quang-Nhat. Estimating compressive strength of high performance concrete with Gaussian process regression model. Adv Civ Eng 2016;2016.
[27] Cristiani Demetrio, Falcetelli Francesco, Yue Nan, Sbarufatti Claudio, Di Sante Raffaella, Zarouchas Dimitrios, et al. Strain-based delamination prediction in fatigue loaded CFRP coupon specimens by deep learning and static loading data. Composites B 2022;241:110020.
[28] Chen Jiayun, Wan Lei, Ismail Yaser, Ye Jianqiao, Yang Dongmin. A micromechanics and machine learning coupled approach for failure prediction of unidirectional CFRP composites under triaxial loading: A preliminary study. Compos Struct 2021;267:113876.
[29] Gu Zewen, Liu Yiding, Hughes Darren J, Ye Jianqiao, Hou Xiaonan. A parametric study of adhesive bonded joints with composite material using black-box and grey-box machine learning methods: Deep neuron networks and genetic programming. Composites B 2021;217:108894.
[30] Reiner Johannes, Vaziri Reza, Zobeiry Navid. Machine learning assisted characterisation and simulation of compressive damage in composite laminates. Compos Struct 2021;273:114290.
[31] Perera Roberto, Guzzetti Davide, Agrawal Vinamra. Graph neural networks for simulating crack coalescence and propagation in brittle materials. Comput Methods Appl Mech Engrg 2022;395:115021.
[32] Spanos Pol D, Ghanem Roger. Stochastic finite element expansion for random media. J Eng Mech 1989;115(5):1035–53.
[33] Zhang Jun, Ellingwood Bruce. Orthogonal series expansions of random fields in reliability analysis. J Eng Mech 1994;120(12):2660–77.
[34] Allaix Diego Lorenzo, Carbone Vincenzo Ilario. Discretization of 2D random fields: A genetic algorithm approach. Eng Struct 2009;31(5):1111–9.
[35] Huntington DE, Lyrintzis CS. Improvements to and limitations of Latin hypercube sampling. Probab Eng Mech 1998;13(4):245–53.
[36] Olsson Anders, Sandberg Göran, Dahlblom Ola. On Latin hypercube sampling for structural reliability analysis. Struct Saf 2003;25(1):47–68.
[37] Shang Shen, Yun Gun Jin. Stochastic finite element with material uncertainties: Implementation in a general purpose simulation program. Finite Elem Anal Des 2013;64:65–78.
[38] Reddy Junuthula Narasimha. Mechanics of laminated composite plates and shells: Theory and analysis. CRC Press; 2003.
[39] Buscema Massimo. Back propagation neural networks. Subst Use Misuse 1998;33(2):233–70.
[40] Gardner Matt W, Dorling SR. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. Atmos Environ 1998;32(14–15):2627–36.
[41] Zhu Jun, Chen Nan, Peng Weiwen. Estimation of bearing remaining useful life based on multiscale convolutional neural network. IEEE Trans Ind Electron 2018;66(4):3208–16.
[42] Wang Qibin, Zhao Bo, Ma Hongbo, Chang Jiantao, Mao Gang. A method for rapidly evaluating reliability and predicting remaining useful life using two-dimensional convolutional neural network with signal conversion. J Mech Sci Technol 2019;33(6):2561–71.
[43] Ding Pan, Liu Xiaojuan, Li Huiqin, Huang Zequan, Zhang Ke, Shao Long, et al. Useful life prediction based on wavelet packet decomposition and two-dimensional convolutional neural network for lithium-ion batteries. Renew Sustain Energy Rev 2021;148:111287.
[44] Chmurski Mateusz, Mauro Gianfranco, Santra Avik, Zubert Mariusz, Dagasan Gökberk. Highly-optimized radar-based gesture recognition system with depthwise expansion module. Sensors 2021;21(21):7298.
[45] Gholamalinezhad Hossein, Khosravi Hossein. Pooling methods in deep neural networks, a review. 2020, arXiv preprint arXiv:2009.07485.
[46] Srivastava Nitish, Hinton Geoffrey, Krizhevsky Alex, Sutskever Ilya, Salakhutdinov Ruslan. Dropout: A simple way to prevent neural networks from overfitting. J Mach Learn Res 2014;15(1):1929–58.
[47] Ioffe Sergey, Szegedy Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR; 2015, p. 448–56.
[48] Der Kiureghian Armen, Ditlevsen Ove. Aleatory or epistemic? Does it matter? Struct Saf 2009;31(2):105–12.
[49] Quinonero-Candela Joaquin, Sugiyama Masashi, Schwaighofer Anton, Lawrence Neil D. Dataset shift in machine learning. Mit Press; 2008.
[50] Malinin Andrey. Uncertainty estimation in deep learning with application to spoken language assessment [Ph.D. thesis], University of Cambridge; 2019.
[51] Lakshminarayanan Balaji, Pritzel Alexander, Blundell Charles. Simple and scalable predictive uncertainty estimation using deep ensembles. Adv Neural Inf Process Syst 2017;30.

[52] Jain Siddhartha, Liu Ge, Mueller Jonas, Gifford David. Maximizing overall diversity for improved uncertainty estimates in deep ensembles. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34, (no. 04):2020, p. 4264–71.
[53] Gustafsson Fredrik K, Danelljan Martin, Schon Thomas B. Evaluating scalable Bayesian deep learning methods for robust computer vision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020, p. 318–9.
[54] Denker John, LeCun Yann. Transforming neural-net output levels to probability distributions. Adv Neural Inf Process Syst 1990;3.
[55] MacKay David JC. A practical Bayesian framework for backpropagation networks. Neural Comput 1992;4(3):448–72.
[56] MacKay David JC. Bayesian neural networks and density networks. Nucl Instrum Methods Phys Res A 1995;354(1):73–80.
[57] Jospin Laurent Valentin, Laga Hamid, Boussaid Farid, Buntine Wray, Bennamoun Mohammed. Hands-on Bayesian neural networks—A tutorial for deep learning users. IEEE Comput Intell Mag 2022;17(2):29–48.
[58] Zhou Zhi-Hua. Ensemble methods: Foundations and algorithms. CRC Press; 2012.
[59] Kullback Solomon, Leibler Richard A. On information and sufficiency. Ann Math Stat 1951;22(1):79–86.
[60] Blei David M, Kucukelbir Alp, McAuliffe Jon D. Variational inference: A review for statisticians. J Amer Statist Assoc 2017;112(518):859–77.
[61] Tran Viet Hung. Copula variational Bayes inference via information geometry. 2018, arXiv preprint arXiv:1803.10998.
[62] Adamčík Martin. The information geometry of bregman divergences and some applications in multi-expert reasoning. Entropy 2014;16(12):6338–81.
[63] Gal Yarin, Ghahramani Zoubin. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: International conference on machine learning. PMLR; 2016, p. 1050–9.
[64] Gal Yarin, Ghahramani Zoubin. Bayesian convolutional neural networks with Bernoulli approximate variational inference. 2015, arXiv preprint arXiv:1506.02158.
[65] Rosenblatt Murray. Remarks on some nonparametric estimates of a density function. Ann Math Stat 1956;832–7.
[66] Puck A, Schürmann H. Failure analysis of FRP laminates by means of physically based phenomenological models. In: Failure criteria in fibre-reinforced-polymer composites. Elsevier; 2004, p. 832–76.
[67] Victoria A Helen, Maragatham G. Automatic tuning of hyperparameters using Bayesian optimization. Evol Syst 2021;12(1):217–23.
[68] Vanhoucke Vincent, Senior Andrew, Mao Mark Z. Improving the speed of neural networks on CPUs. 2011.
[69] Li Mu, Zhang Tong, Chen Yuqiang, Smola Alexander J. Efficient mini-batch training for stochastic optimization. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. 2014, p. 661–70.
[70] Zhang Tong, Yu Bin. Boosting with early stopping: Convergence and consistency. Ann Statist 2005;33(4):1538–79.
[71] Kingma D P, Ba L. A method for stochastic optimization. In: 3rd International conference on learning representations. 2015, p. 1–15.
[72] Scott David W. On optimal and data-based histograms. Biometrika 1979;66(3):605–10.