Bulletin of Computational Intelligence
https://www.sciltp.com/journals/bci

Article

DeepONet for the Prediction of Failure Response of a Two-Dimensional Fibre-Reinforced Composite Plate

Georgios A. Drosopoulos [1,2,*] and Georgios E. Stavroulakis [3]

[1] Department of Civil Engineering, International Hellenic University, 62124 Serres, Greece
[2] Discipline of Civil Engineering, University of KwaZulu-Natal, 4041 Durban, South Africa
[3] School of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece
* Correspondence: gdrosopoulos@ihu.gr

How To Cite: Drosopoulos, G.A.; Stavroulakis, G.E. DeepONet for the Prediction of Failure Response of a Two-Dimensional Fibre-Reinforced Composite Plate. *Bulletin of Computational Intelligence* **2025**, *1*(1), 76–88. https://doi.org/10.53941/bci.2025.100005

Received: 24 July 2025
Revised: 30 August 2025
Accepted: 9 September 2025
Published: 12 September 2025

Abstract: Applications in the field of data-driven mechanics are widely studied in the last years exploiting latest development of artificial intelligence. In this context, several machine learning techniques have been adopted to offer a fast and accurate prediction of the structural response of materials and complex structural systems. A relatively new machine learning concept relies on the use of Deep Operator Networks (DeepONets) that can approximate operators accurately and efficiently, from a relatively small dataset. The article, therefore, provides the methodology framework of applying a deep operator network (DeepONet) in structural mechanics applications. A dataset is developed using parametric non-linear finite element simulations for a two-dimensional fibre-reinforced composite structure. Then, a DeepONet is developed, aiming to predict the failure response of this structure. Comparison with results obtained from traditional Artificial Neural Networks (ANNs) is also presented. Results obtained from testing the trained DeepONet model on data not included in training indicate a proper performance. Testing the DeepONet model on unseen trunk input or branch input functions leads to satisfactory accuracy, while testing it on unseen trunk and branch input leads to a decent accuracy, that is improved compared with the one received from ANNs. Thus, the capacity of DeepONet to predict the response in the context of non-linear structural mechanics is evaluated.

Keywords: structural engineering; neural operators; DeepONet; artificial neural networks; finite element analysis; data-driven mechanics

1. Introduction

Several studies have been conducted in the last years in the field of solid mechanics, aiming in evaluating the structural response of different types of structural systems and materials using machine learning techniques. The advantage of using machine learning in structural engineering and mechanics, is attributed to the fast and accurate prediction of the response, without the need to develop new, computationally expensive numerical models when random input data appears.

In this framework, traditional as well as more recent machine learning methods have been adopted in various structural engineering tasks. In [1] different machine learning approaches such as artificial neural networks (ANNs), ensemble intelligent predictive models, fuzzy inference systems (FIS), support vector regression models (SVR), combinatorial methods of group data handling (GMDH-Combi), and others are listed, predicting the compressive strength of masonry walls. Recently, modern regression tasks have been developed, implementing among others multi-scale analysis [2], or incorporating the physical description of the problem in traditional neural networks, for instance, physics-informed neural networks (PINNs) [3,4].

In several studies, ANNs which are among the simplest but still efficient machine learning methods, are used to predict the dynamic response of multi-storey steel buildings [5], to examine the natural period and strength ratio as well as serviceability requirements on seismic responses of structures [6], to predict the compressive strength of masonry walls [1,7], concrete [8,9], cement-based mortars [10], to evaluate the ultimate limit load and collapse mechanism of masonry arch bridges [11], and to determine the response of composite materials within homogenization techniques [12,13].

Despite their wide applicability, most existing surrogate models would need to be retrained when some of the input parameters are changed, such as the loading, boundary conditions, geometry of the structure, material properties and others. To address this issue, a new class of deep learning methods have been introduced the last years, aiming in operator learning. The goal of these efforts is to offer a more general, function-to-function prediction capacity, using neural operators. Two examples of neural operators are the Fourier neural operator (FNO), and the deep operator network (DeepONet). The FNO was introduced in [14] to solve partial differential equations considering parametric input functions. The input functions are processed using multiple Fourier layers each adopting the fast Fourier transform to its input, establishing a mapping to output function space.

DeepONet [15,16], relies on the concept of introducing an architecture of two sub-networks, targeting to provide the function-to-function mapping. One of these two networks is called branch network and is used to encode the input functions at a fixed number of sensors. The other is known as trunk network and is adopted to encode the locations (positions) for the output functions. Applications of DeepONet can be found in the field of multi-scale problems in mechanics [17], in problems with time-dependent loads [18], in a physics-informed framework to predict crack paths [19], in applications highlighting novel architectures, suggesting for instance U-net network for the trunk network instead of conventional ANN [20] as well as in other recent applications in structural engineering [21,22].

To further improve the capacity of DeepONets to predict the mechanical response, physical conditions in the form of governing equations can be included in the loss function of the model, introducing in this way Physics-Informed Neural Networks [23] in DeepONets. Doing that eliminates the need for using datasets in the forward problem. On the other hand, adding governing equations in the form of differential equations may increase the complexity of the model, for more advanced structural problems. Recent approaches aim to address this issue and simplify the process, using for instance structural stiffness matrices to enforce equilibrium and energy conservation principles [21].

Despite the mentioned applications of DeepONet in structural mechanics, more work is needed to highlight the capacity of DeepONet to predict the non-linear, failure response of structural systems including composite materials. Accurate failure prediction would be a significant step towards developing structural digital twins, where the real structure is represented by its digital counterpart, and data is exchanged between the real structure (measurements) and the digital one, aiming in predicting the response in real time mode. Therefore, the aim of this article is to cover this research gap and explore how DeepONet can be adopted to predict failure. In this context, the framework and main steps of using DeepONet to predict failure will be presented and the accuracy of the approach will be evaluated.

A two-dimensional fibre-reinforced composite plate model is developed in commercial finite element software. A non-linear plasticity law is assigned, and the model is solved parametrically, using as input loading functions and output equivalent plastic strains measured on chosen positions of the domain (input domain geometry). The proposed DeepONet is then used to predict the failure response as this is provided by the output plastic strains.

It is noted, that according to [15,24] a neural network even with a single hidden layer can approximate accurately any nonlinear continuous operator. In relation to this statement, the article provides also results for the same structural problem, from a conventional Artificial Neural Network, demonstrating its capacity to predict the structural response.

2. DeepONet

2.1. Introduction to DeepONet

The development of DeepONets emanates from the universal approximation theorem [15,25] which states that neural networks can be adopted to approximate any continuous function when there are no constraints in the number of hidden layers and neurons. Even a neural network with a single hidden layer can approximate any non-linear continuous function or operator [15]. It is noticed that a functional defines a mapping between a space of functions and real numbers while an operator indicates a mapping between a space of functions and another space of functions.

Based on this description, DeepONets aim to improve the capacity of neural networks to learn operators accurately and efficiently, using relatively small datasets. If G is an operator that uses as input the function u, then the output function is $G(u)$. This output is evaluated at the positions y of the domain, and thus, $G(u)(y)$ is the DeepOnet real number output.

According to this layout, the input consists of (a) the input function u that is encoded in the branch neural network and (b) the input positions y, encoded in the trunk neural network. The general architecture of DeepONet is schematically shown in Figure 1.

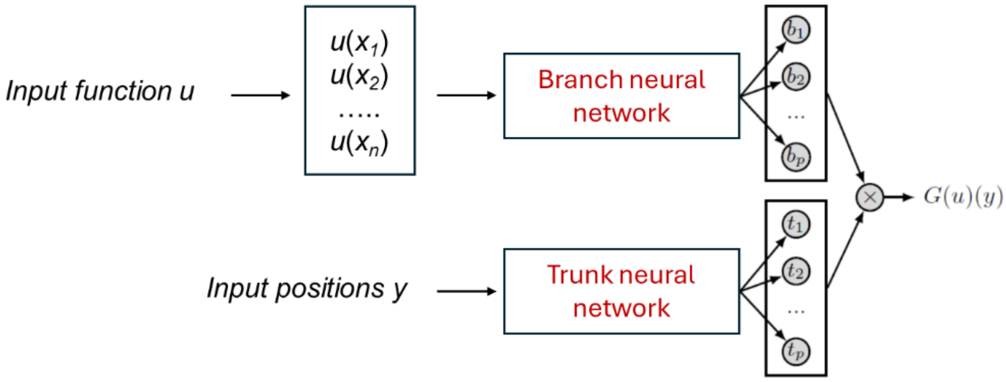


Figure 1. Layout of unstacked DeepOnet.

According to the architecture shown in Figure 1, the input branch neural network takes as input the vector $[u(x_1)\ u(x_2)\ \ldots\ u(x_n)]^T$ and outputs $[b_1\ b_2\ \ldots\ b_p]^T$. In addition, the trunk neural network takes as input the positions y and outputs the vector $[t_1\ t_2\ \ldots\ t_p]^T$. The output of the overall DeepONet is derived from the dot product of the outputs of the branch and trunk networks, potentially followed by the addition of a bias term, as shown in Equation (1):

$$G(u)(y) \approx \sum_{k=1}^{p} b_k t_k + b_0 \tag{1}$$

The mechanism of DeepONet that explains its capacity to offer improved function-to-function prediction, arises from the operator-learning architecture which is described in this section. In particular, the branch network encodes the dependency of the model on input functions, while the trunk network encodes the positions where the output functions are measured. Outputs of the branch and trunk networks are combined through a dot product. The overall scheme offers a factorization, representing the functional structure of the problem, enabling the function-to-function prediction capacity of DeepONets.

It is noted that for the branch and trunk networks, any type of known neural networks can be adopted, such as Fully-connected Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks and others. To implement DeepONets in computing code for real-world applications, the architecture described in this section and shown in Figure 1 is relatively simple. More important for the interested user is to properly select the input functions and positions to measure the output functions, as well as to choose the output functions.

2.2. Application of DeepONet on the Chosen Structural Problem

Motivation for this research is to highlight the steps and evaluate the accuracy of using DeepONets in structural problems. The main research question refers to the capacity of DeepONets to predict the failure response of composite structures. To address the research question, the article proposes a methodology for using DeepONets within structural mechanics, highlighting in this context the main steps of the method, that include the definition of input (loading) functions, definition of positions to measure the output in the structure, as well as definition of output functions predicting failure. As it is shown in subsequent sections, to check the accuracy and capacity of DeepONets to predict failure using random input, configurations with unknown input data for the branch network, for the trunk network as well as for both the branch and the trunk networks are tested.

Therefore, DeepONet is used to predict the failure response of a two-dimensional fibre-reinforced composite plate, within plane stress conditions. This problem arises during the calculation of the response of a representative volume element (RVE) within multi-scale computational homogenization [13]. However, a simpler scheme is discussed here to better highlight the applicability of DeepONet, avoiding elements of computational homogenization such as the calculation of effective stress and stiffness or the application of periodic boundary conditions.

The input function u is defined by enforced displacement loading at one edge of a two-dimensional non-linear finite element model, representing the composite material. The level and direction of loading are defined by parameters L and s as explained in Section 3.1.

The input positions y are defined by chosen coordinates in the domain of the structure, where the output is measured. Output is the equivalent plastic strain, evaluated at the input positions y. The proposed concept is schematically presented in Figure 2.

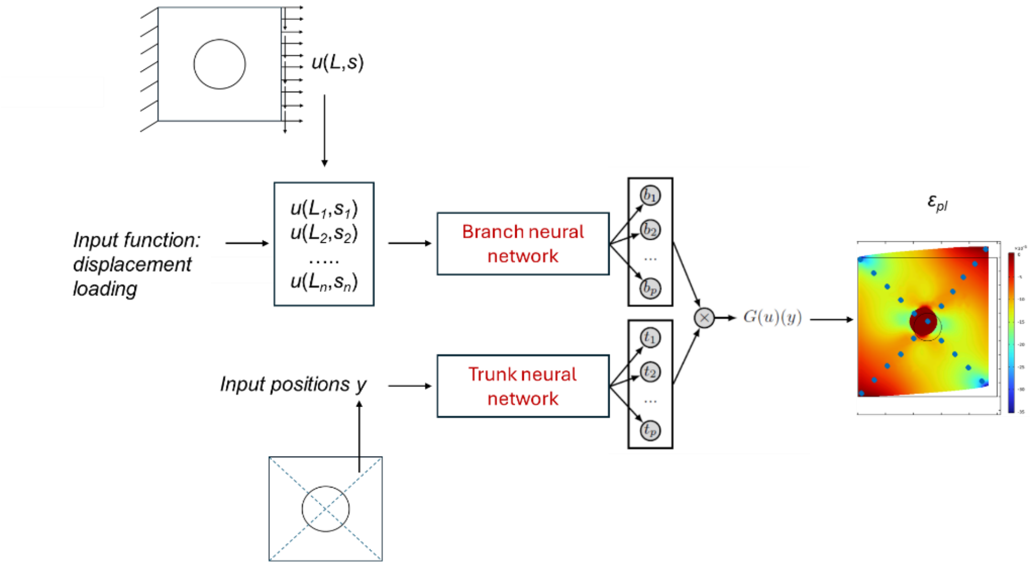


Figure 2. Application of DeepONet on a fibre-reinforced composite plate to predict failure.

The simplest case of a Fully-connected Feed Forward Neural Network is adopted for both the branch and the trunk networks in this article. Each of these networks has 4 hidden layers, with 128 neurons per hidden layer. The activation function used is the ReLu function. Normalization of the input-output is considered, and the overall dataset is split to 80% for training and 20% for testing. The proposed framework is applied in Python 3.12.11, environment, using the Tensorflow package, version 2.19.0.

3. Dataset

3.1. Training of DeepONet

Aim of this article is to highlight the main steps, as well as the capacity of DeepONet to predict the function-to-function response of a structural system. A dataset is generated for this purpose, using finite element analysis. In this context, a non-linear finite element model has been developed using Comsol Multiphysics. It consists of a two-dimensional fibre-reinforced composite structure, as shown in Figure 3. The structure is fixed in the left vertical edge, while a displacement loading is applied in the right vertical edge, along the horizontal and vertical axis (Figure 3).

The model is assigned a von Mises plasticity law for the matrix and the fibre reinforcement, within plane stress conditions. The material properties of the matrix and fibre reinforcement are provided in Table 1.

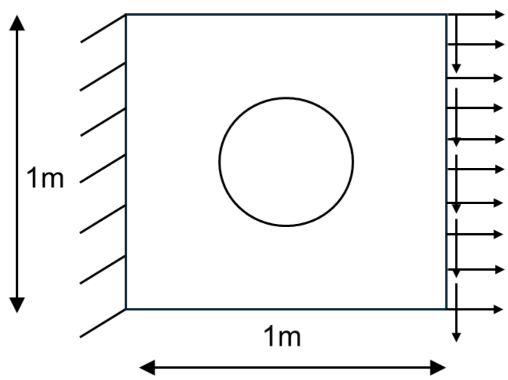


Figure 3. Geometry, boundary conditions and loading of the fibre-reinforced composite structure.

Table 1. Material properties of the fibre and the matrix.

Material	Young's Modulus (GPa)	Poisson's Ratio	Yield Stress (MPa)	Isotropic Tangent Modulus (GPa)
Matrix	70	0.3	20	7
Fibre	400	0.2	200	40

To generate the dataset, that will be used to describe the failure response of the structure shown in Figure 3, parametric non-linear finite element simulations are implemented. Parameters are the radius R of the fibre reinforcement and L, s values which appear in the displacement loading functions shown in Equation (2). The L and s parameters are introduced to control the magnitude of the displacement loading, which is applied at the right vertical edge of the structure (Figure 3).

$$u_x = Lsin(2\pi s),\ \ u_y = Lsin\left(4\pi s + \frac{\pi}{3}\right) \tag{2}$$

These loading functions as well as parameters R, L, and s will be used to train the branch neural network in the DeepONet architecture. The range of these parameters are shown below:

$$R \in range(0.05, 0.05, 0.4), s \in range(-0.25,0.025,0.25), L \in range(-0.1,0.02,0.1) \tag{3}$$

Outputs from each parametric simulation are the equivalent plastic strains $\varepsilon_{plxx}, \varepsilon_{plyy}, \varepsilon_{plxy}$ obtained in chosen points of the structural domain, located in the two dashed diagonal lines of the structure, as shown in Figure 4. In total, 21 equally spaced points have been considered in these two lines, representing the input positions y. The coordinates of these points will be used by the trunk network in the training of DeepONet.

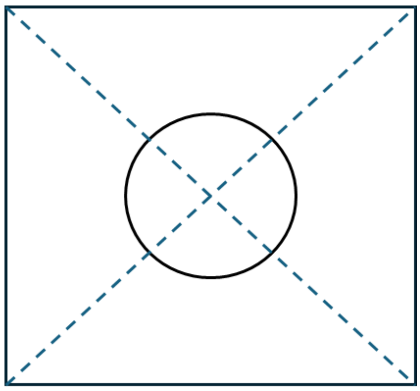
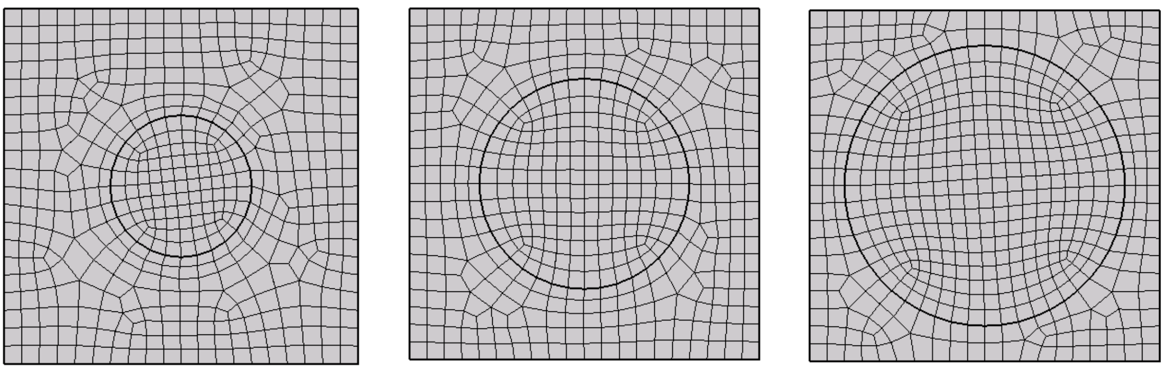


Figure 4. Diagonal lines in the fibre-reinforced composite plate representing the positions for the trunk network coordinates where the equivalent plastic strain output is captured.

Following the provided concept, 1833 dataset lines have been generated from parametric simulations and used to train the DeepONet. The trained machine learning model will be a computational tool that predicts failure in the form of equivalent plastic strain, at random positions of the structure, for a random loading and radius of the fibre reinforcement. The mesh for some of the parametric models with varying radius of the fibre reinforcement is shown in Figure 5.



(**a**) (**b**) (**c**)

Figure 5. Mesh of the parametric models with radius R of the reinforcement equal to (**a**) 0.2 m, (**b**) 0.3 m, (**c**) 0.4 m.

3.2. Testing of DeepONet

To test the capacity and accuracy of the trained DeepONet to predict failure of the structure, three testing configurations shown in Figure 6 are used. According to the first configuration shown in Figure 6a, the same branch input as the one used to train the DeepONet is adopted, but in different trunk coordinates as compared to training. In this case the coordinates of six points in the line shown in Figure 6a are used for the trunk network. Thus, the capacity of DeepONet to predict the equivalent plastic strain in new positions of the structure is tested.

In the second configuration shown in Figure 6b, a new input for the branch network is used. Thus, new input parameters and displacement loading functions are adopted, as given by Equation (4). This layout is about to test the capacity of DeepONet to predict the response when new input functions not included in the training process are used. For this case, the output positions used for the trunk network are kept the same as the ones in the training (points located on two diagonal lines of the structure, Figures 4 and 6b. Two sample points and the same 21 coordinates for the trunk network as in the training are considered here.

$$u_x = (2L/5)sin(\pi/s + \pi/4),\ \ u_y = 3Lsin(\pi s) \tag{4}$$

In the last configuration used to test the trained DeepONet, the new input branch network functions provided by Equation (4) are used. In addition, new trunk coordinates located on the horizontal-vertical dashed lines in Figure 6c are adopted, as compared to the training process. Two sample points are considered in this case as in the second configuration.

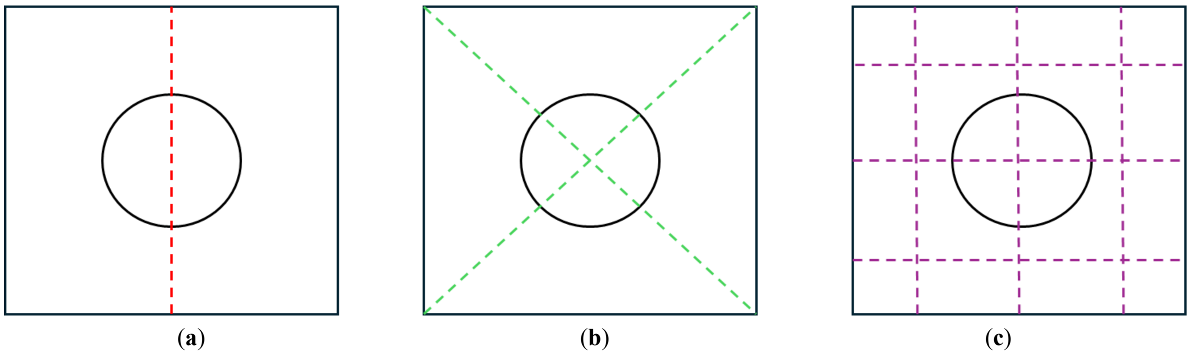


Figure 6. Lines in the fibre-reinforced composite plate representing the position for the trunk network coordinates where the equivalent plastic strain output is captured, for the case (**a**) of same branch network input as the one in training but different trunk network coordinates, (**b**) of different branch input (different loading function) but same trunk network coordinates as in the training, (**c**) of different branch input and trunk network coordinates than the ones in training.

3.3. Application of ANNs on the Same Structural Problem

Feed-forward neural networks are also developed and used in this article, to highlight the performance of this efficient and widely used machine learning approach on the same structural problem. Inputs of the developed ANN models are the parameters *R*, *L*, *s* and displacement loading u_x, u_y, as they are introduced in Section 3.1. Outputs are the equivalent plastic strains ε_{plxx}, ε_{plyy}, ε_{plxy} obtained in the positions indicated in Sections 3.1 and 3.2. A brief investigation of different ANN architectures is also conducted, as discussed in the results.

4. Results and Discussions

4.1. Details of the Dataset

First, the results of the DeepONet training are presented. Then, the performance of the DeepONet on unseen input data is evaluated. For all cases, results from the traditional Artificial Neural Network are also provided, to offer an additional measure of the response of DeepONet.

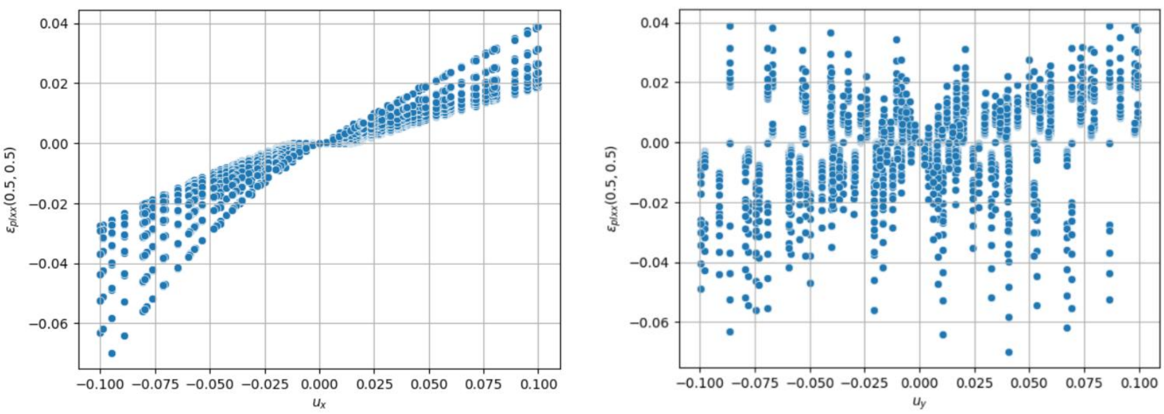
The quality of the dataset that is used for the training process significantly influences the performance of the neural network [1,7]. In Table 2 statistics of the dataset parameters adopted for training are provided. It is noted that the output equivalent plastic strain ε_{plxx} is provided, noticing that the same process is followed for ε_{plyy}, ε_{plxy}. Also, the 21 trunk network coordinates are given for each output plastic strain, in the first column of Table 2.

Table 2. Statistics of dataset parameters used for the training of DeepONet.

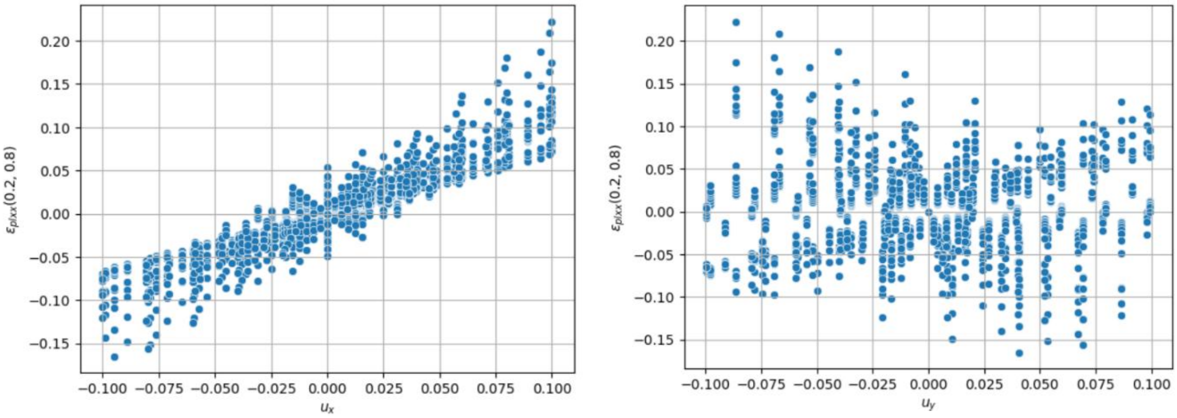
	Min	Max	Mean	Standard Deviation
R (m)	0.05	0.4	0.224195	0.11436
s	−0.25	0.25	0.000641	0.150744
L	−0.1	0.1	−0.00038	0.062872
u_x (m)	−0.1	0.1	0.000735	0.045169
u_y (m)	−0.09945	0.099452	0.000278	0.044695
ε_{plxx} (0,0)	−0.12477	0.33666	0.012136	0.092045
ε_{plxx} (0.1,0.1)	−0.10426	0.20028	0.005285	0.053579
ε_{plxx} (0.2,0.2)	−0.15626	0.22344	0.003143	0.056888
ε_{plxx} (0.3,0.3)	−0.14725	0.16106	0.00026	0.041256
ε_{plxx} (0.4,0.4)	−0.10158	0.10408	−0.0011	0.026136
ε_{plxx} (0.5,0.5)	−0.0698	0.039043	−0.00169	0.014627
ε_{plxx} (0.6,0.6)	−0.09837	0.10334	−0.00108	0.025966
ε_{plxx} (0.7,0.7)	−0.15274	0.16621	0.000236	0.042243
ε_{plxx} (0.8,0.8)	−0.15382	0.21859	0.003281	0.056475
ε_{plxx} (0.9,0.9)	−0.09727	0.20174	0.006145	0.053044
ε_{plxx} (1,1)	−0.12495	0.33747	0.012128	0.092084
ε_{plxx} (1,0)	−0.12447	0.31925	0.003868	0.063527
ε_{plxx} (0.9,0.1)	−0.09459	0.19393	0.002675	0.039042
ε_{plxx} (0.8,0.2)	−0.15991	0.21298	0.002174	0.045741
ε_{plxx} (0.7,0.3)	−0.1431	0.1513	0.000126	0.036583
ε_{plxx} (0.6,0.4)	−0.09764	0.09956	−0.00103	0.024632
ε_{plxx} (0.4,0.6)	−0.10697	0.10947	−0.00097	0.024991
ε_{plxx} (0.3,0.7)	−0.15275	0.16683	0.00023	0.037837
ε_{plxx} (0.2,0.8)	−0.16481	0.22258	0.002221	0.047116
ε_{plxx} (0.1,0.9)	−0.11202	0.18504	0.002527	0.039106
ε_{plxx} (0,1)	−0.12449	0.31938	0.003872	0.063511

The dispersion of the dataset points can also influence the performance of the machine learning model. In diagrams shown in Figure 7 the relation between some of the input parameters (displacement loading) and output parameters (equivalent plastic strain) is provided.

Some results from the parametric investigation representing the equivalent plastic strain in the domain, for different R, L, s values are shown in Figure 8. According to these results, it seems that failure is mainly developed in the matrix, which is expected due to its lower strengths as compared to the fibre reinforcement (Table 1).

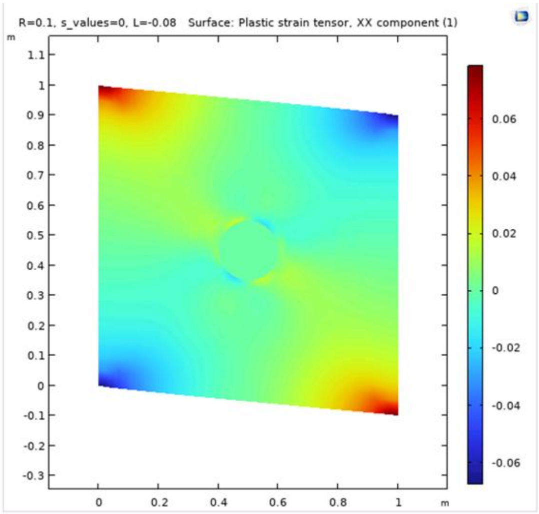


(**a**)

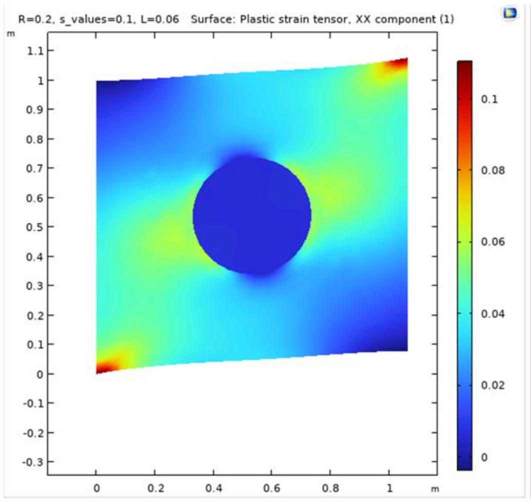


(**b**)

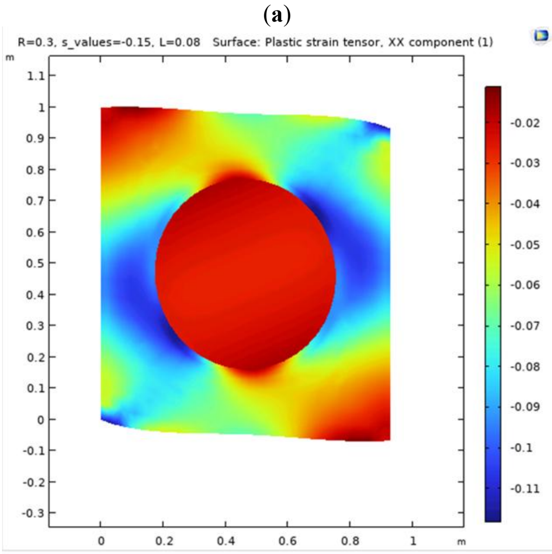
Figure 7. Plastic strain output ε_{plxx} versus input loading along directions *x* and *y* for (**a**) output captured on point (0.5, 0.5) (middle of the fibre reinforcement), (**b**) output captured on point (0.2, 0.8) (matrix).



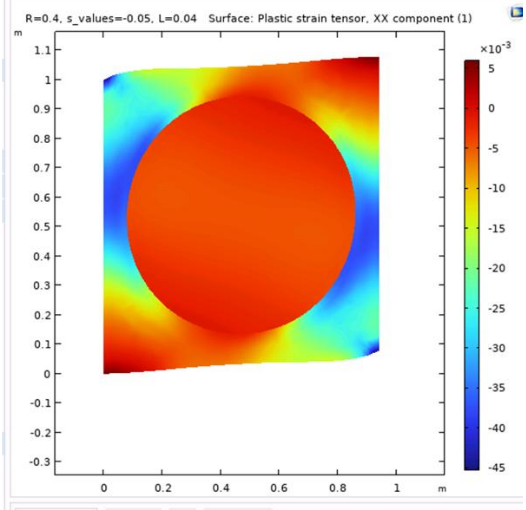
(**a**)



(**b**)



(**c**)



(**d**)

Figure 8. Distribution of equivalent plastic strain ε_{plxx} for (**a**) *R* = 0.1, *s* = 0, *L* = −0.08, (**b**) *R* = 0.2, *s* = 0.1, *L* = 0.06, (**c**) *R* = 0.3, *s* = −0.15, *L* = 0.08, (**d**) *R* = 0.4, *s* = −0.05, *L* = 0.04.

4.2. DeepONet and ANN Training Results

Results of the DeepONet and ANN training are presented in this section. A short parametric investigation is conducted first, aiming to identify the best ANN architecture. Thus, the number of hidden layers and the number of neurons per hidden layer vary, as shown in Table 3. According to this investigation, the ANN with 4 hidden layers and 128 neurons per hidden layer, seems to perform better than the other ANN models. Results obtained using this architecture will be presented next.

Table 3. Prediction metrics of ε_{plxx} output using the testing data for different ANN architectures.

ANN Model		R^2	RMSE	MAE
Hidden Layers	**Neurons per Hidden Layer**			
1	13	0.8935	0.016885	0.012326
1	30	0.9583	0.010563	0.007017
1	128	0.9844	0.006453	0.004403
2	30	0.9930	0.004315	0.003060
2	128	0.9989	0.001718	0.001185
3	128	0.9990	0.001671	0.001160
4	128	0.9996	0.001024	0.000721

A similar architecture to the one selected for the ANN is also chosen for the DeepONet model. Thus, 4 hidden layers are used for the branch and 4 for the trunk networks, with 128 neurons per layer. In Table 4 the hyperparameters of the adopted DeepONet and ANN models are presented.

Table 4. Hyperparameters of the used DeepONet and ANN models.

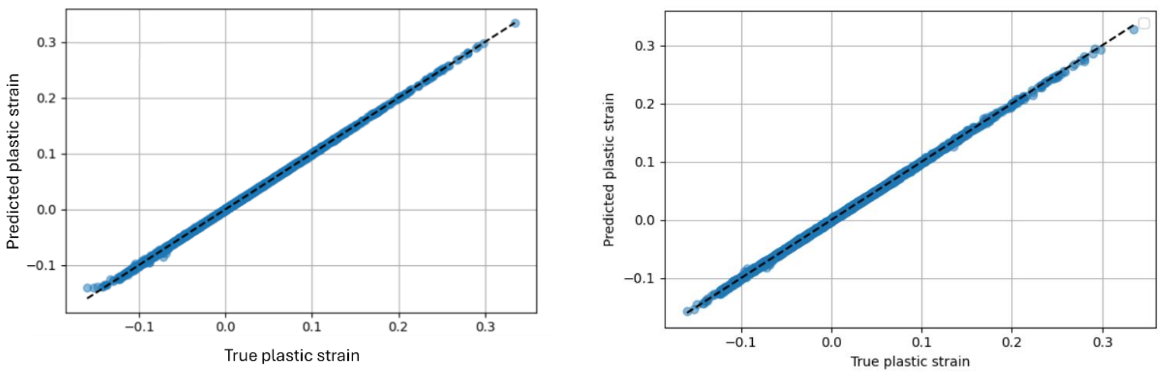
Model	Hidden Layer Number	Neurons per Hidden Layer	Activation Function	Optimizer	Learning Rate	Batch Size	Epochs Number	Loss Function
DeepONet	4 (branch) 4 (trunk)	128	ReLu	Adam	0.001	64	200	MSE
ANN	4	128	ReLu	Adam	0.001	64	200	MSE

In Table 5 metrics of output prediction are shown as derived from the DeepONet and the chosen ANN model. According to this table, both models perform well in terms of the regression factor R^2, which is close to 1. The root mean squared error (RMSE) and mean absolute error (MAE), are however, lower for the DeepONet.

Table 5. Prediction metrics of ε_{plxx} output from DeepONet and ANN, using the testing data.

Model	R^2	RMSE	MAE
DeepONet	0.9998	0.000683	0.000398
ANN	0.9996	0.001024	0.000721

It is noted that a similar computational cost arises from the training process using both methods. DeepONet training time is 8.3 min while training of the chosen ANN model takes 8.8 min. For the simulations Google Colab online platform was used and thus, no local pc resources were adopted. Next, regression lines for the testing dataset are provided in Figure 9, depicting the capacity of both models to predict the equivalent plastic strain at the chosen positions of the structural domain.

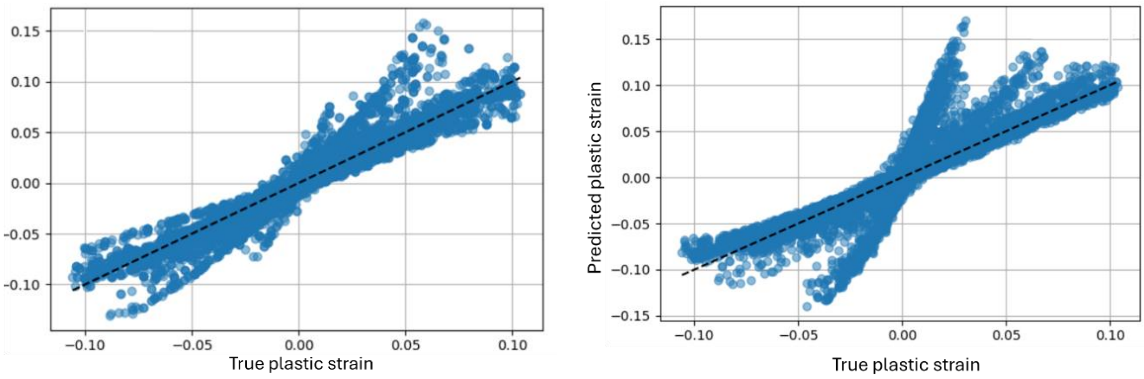


(**a**) (**b**)

Figure 9. Regression plots from the testing data of (**a**) the DeepONet, (**b**) the ANN.

4.3. DeepONet and ANN Testing Results in New Data

The trained DeepONet and ANN machine learning models are now tested on new data not included in the training dataset, evaluating their capacity to predict the failure response of the structure. Three testing configurations are considered, as described in Figure 6. Table 6 and Figure 10 provide output metrics and regression plots for the case of same branch network input as in training but different trunk network coordinates (configuration shown in Figure 6a). This configuration evaluates the capacity of the tested machine learning techniques to predict the failure response (equivalent plastic strain), when the same loading is applied in the structural system, but the output positions change as compared to training.



(**a**) (**b**)

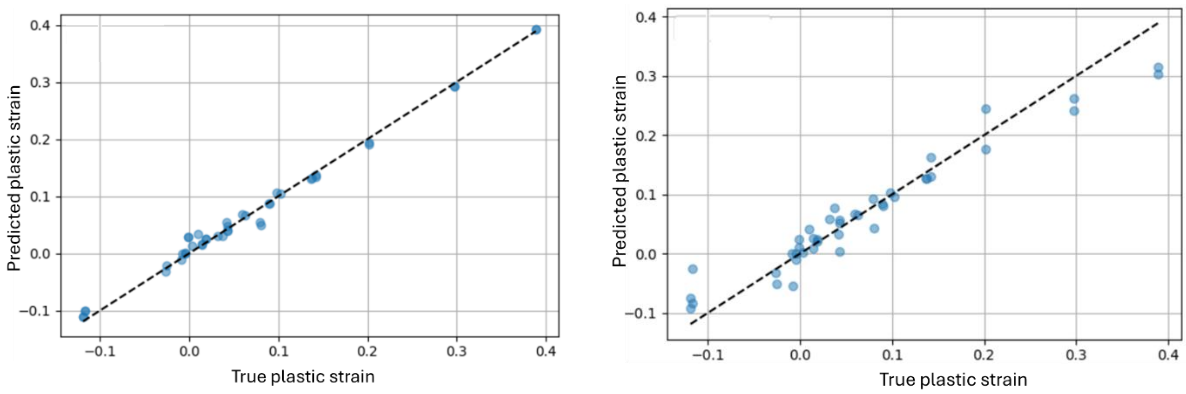
Figure 10. Regression plots for the case of same branch network input as in training but different trunk network coordinates, for (**a**) the DeepONet, (**b**) the ANN.

Table 6. Prediction metrics of ε_{plxx} output for the case of same branch network input (same loading) as in training but different trunk network coordinates (different output positions).

Model	R^2	RMSE	MAE
DeepONet	0.8836	0.011406	0.006419
ANN	0.6673	0.019287	0.009316

Both Table 6 and Figure 10 indicate that DeepONet leads to an improved regression coefficient as compared to ANN, when tested on unseen output positions. However, the error metrics are of the same magnitude in both cases, but lower for DeepONet.

Next, the capacity of both models to predict the failure response when different branch input (loading functions given by equation (4) instead of (2) in training) but same trunk network coordinates as in training (same output positions, Figure 6b), is evaluated. Results of this investigation presented in Table 7 and Figure 11 indicate that both models perform well, with DeepONet providing less MAE and higher regression coefficient.



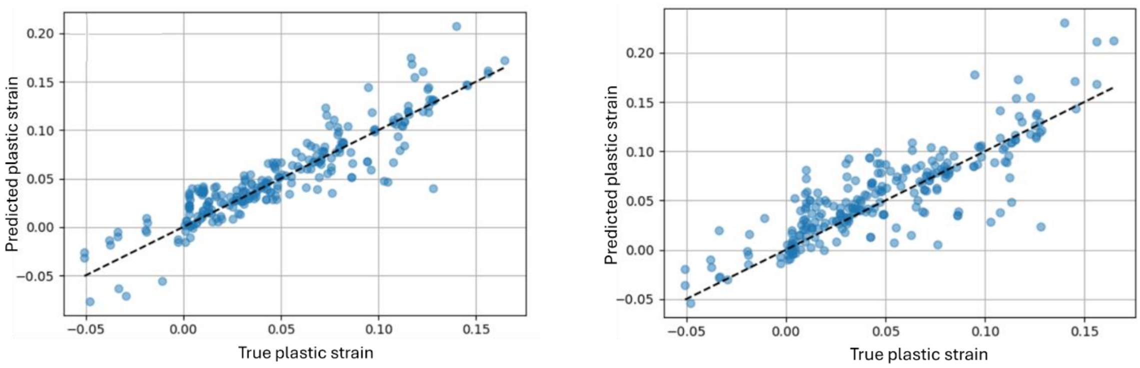
(**a**) (**b**)

Figure 11. Regression plots for the case of different branch network input but same trunk network coordinates as in training, for (**a**) the DeepONet, (**b**) the ANN.

Table 7. Prediction metrics of ε_{plxx} output for the case of different branch network input (different loading) but same trunk network coordinates (same output positions) as in training.

Model	R^2	RMSE	MAE
DeepONet	0.9901	0.011727	0.008581
ANN	0.9246	0.032313	0.023593

A last investigation is provided, evaluating the capacity of DeepONet and ANN to predict the plastic strain, when new input branch functions (loading) and new output trunk coordinates (positions) as compared to training are considered (Figure 6c). According to Table 8 and Figure 12 there is a significant reduction of the correlation coefficient for both models, with DeepONet presenting a better capacity to predict the output.



(**a**) (**b**)

Figure 12. Regression plots for the case of different branch network input as well as different trunk network coordinates as compared to training, for (**a**) the DeepONet, (**b**) the ANN.

Table 8. Prediction metrics of ε_{plxx} output for the case of different branch network input (different loading) as well as different trunk network coordinates (output positions) as compared to training.

Model	R^2	RMSE	MAE
DeepONet	0.8023	0.019456	0.013802
ANN	0.6495	0.025909	0.018179

5. Conclusions

The steps and key aspects of applying neural operators in the context of structural mechanics are presented in this article. A DeepONet machine learning model is proposed, aiming in evaluating the non-linear failure response of a fibre-reinforced composite structure, due to mechanical actions under varying loading paths. The DeepONet uses as input the loading functions and the positions in the domain where the output is reported, and as output the equivalent plastic strains on those positions.

Results from the DeepONet, as well as results obtained from a conventional, Feed-forward Neural Network are provided. Different testing scenarios of the trained computational models are evaluated, considering the case of new input functions and new positions where the output is measured. It seems that while DeepONet performs generally better in terms of accuracy and regression capacity on unseen data, both models lead to a satisfactory prediction capacity.

The above investigation considers a limited number of parameters which determine the response of the machine learning models. For instance, the influence of the quantity of input (functional) data on the response of neural operators and their capacity to predict the output, would further highlight the topic. Therefore, further investigation, using additional parameters highlighting the capacity of DeepONet to establish the function-to-function correlation in structural mechanics applications, is left for future investigation.

While the current study focuses on FEM-generated data, which is noise-free, real experimental or sensor-based data may include measurement uncertainties. DeepONets, due to their operator-learning structure, are generally robust to moderate noise [26], however, the matter deserves more investigation, in particular when DeepONets are integrated in Digital Twins where noisy measurement data is expected. In future work, robustness under noise conditions will be investigated, by augmenting the training data with synthetic noise injection, employing regularization strategies and uncertainty quantification approaches or testing alternative neural operators such as Fourier-enhanced deep operator networks [26].

Except ANNs, more traditional machine learning methods can also be used in future efforts to provide a more holistic comparison between those methods and DeepONets.

The presented methodology can be used for the creation of Artificial Intelligence based Reduced Order Models (AI-ROM) covering the response to different loading cases, which are important elements for the creation of structural Digital Twins [27]. Extension of the proposed methodology for the comparison of results with selected measurements and the estimation of possibly missing loading or material data could be investigated and will allow for application of the method on multi-scale analysis with representative volume element solutions and full field measurements like the ones obtained from digital image correlation.

Author Contributions

G.A.D.: conceptualization, methodology, software, writing—original draft preparation; G.E.S.: conceptualization, methodology, validation, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable

Informed Consent Statement

Not applicable.

Data Availability Statement

The details of research data are provided in the manuscript. The models and codes developed for this research are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Asteris, P.G.; Drosopoulos, G.A.; Cavaleri, L.; et al. Mapping and Revealing the Nature of Masonry Compressive Strength Using Computational Intelligence. *Structures* **2025**, *78*, 109189. https://doi.org/10.1016/j.istruc.2025.109189.
2. Zhongbo, Y.; Hien, P.L. Pre-Trained Transformer Model as a Surrogate in Multiscale Computational Homogenization Framework for Elastoplastic Composite Materials Subjected to Generic Loading Paths. *Comput. Methods Appl. Mech. Eng.* **2024**, *421*, 116745. https://doi.org/10.1016/j.cma.2024.116745.
3. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *J. Comput. Phys.* **2019**, *378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045.
4. Milićević, B.; Ivanović, M.; Stojanović, B.; et al. Optimization of Physics-Informed Neural Networks for Efficient Surrogate Modeling of Huxley's Muscle Model in Multi-Scale Finite Element Simulations. In Proceedings of the 2023 IEEE 23rd International Conference on Bioinformatics and Bioengineering (BIBE), Dayton, OH, USA, 4 December 2023; pp. 457–461.
5. Sukhnandan, J.; Drosopoulos, G.A. A Machine Learning Approach Used to Predict the Peak Displacement, Base Shear and Fundamental Frequency of Multi-Storey Steel Structures under Seismic Excitation. *Structures* **2025**, *73*, 108367. https://doi.org/10.1016/j.istruc.2025.108367.
6. Noureldin, M.; Ali, A.; Sim, S.; et al. A Machine Learning Procedure for Seismic Qualitative Assessment and Design of Structures Considering Safety and Serviceability. *J. Build. Eng.* **2022**, *50*, 104190. https://doi.org/10.1016/j.jobe.2022.104190.
7. Asteris, P.G.; Lourenço, P.B.; Hajihassani, M.; et al. Soft Computing-Based Models for the Prediction of Masonry Compressive Strength. *Eng. Struct.* **2021**, *248*, 113276. https://doi.org/10.1016/j.engstruct.2021.113276.
8. Asteris, P.G.; Skentou, A.D.; Bardhan, A.; et al. Soft Computing Techniques for the Prediction of Concrete Compressive Strength Using Non-Destructive Tests. *Constr. Build. Mater.* **2021**, *303*, 124450. https://doi.org/10.1016/j.conbuildmat.2021.124450.

9. Asteris, P.G.; Skentou, A.D.; Bardhan, A.; et al. Predicting Concrete Compressive Strength Using Hybrid Ensembling of Surrogate Machine Learning Models. *Cem. Concr. Res.* **2021**, *145*, 106449. https://doi.org/10.1016/j.cemconres.2021.106449.
10. Asteris, P.G.; Apostolopoulou, M.; Skentou, A.D.; et al. Application of Artificial Neural Networks for the Prediction of the Compressive Strength of Cement-Based Mortars. *Comput. Concr.* **2019**, *24*, 329–345. https://doi.org/10.12989/CAC.2019.24.4.329.
11. Motsa, S.M.; Stavroulakis, G.E.; Drosopoulos, G.A. A Data-Driven, Machine Learning Scheme Used to Predict the Structural Response of Masonry Arches. *Eng. Struct.* **2023**, *296*, 116912. https://doi.org/10.1016/j.engstruct.2023.116912.
12. Fuhg, J.N.; Marino, M.; Bouklas, N. Local Approximate Gaussian Process Regression for Data-Driven Constitutive Models: Development and Comparison with Neural Networks. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114217. https://doi.org/10.1016/j.cma.2021.114217.
13. Drosopoulos, G.A.; Stavroulakis, G.E. *Nonlinear Mechanics for Composite Heterogeneous Structures*; CRC Press: Boca Raton, FL, USA, 2022.
14. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; et al. Fourier Neural Operator for Parametric Partial Differential Equations. *arXiv* **2021**, arXiv:2010.08895.
15. Lu, L.; Jin, P.; Pang, G.; et al. Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229. https://doi.org/10.1038/s42256-021-00302-5.
16. Lu, L.; Meng, X.; Mao, Z.; et al. DeepXDE: A Deep Learning Library for Solving Differential Equations. *SIAM Rev.* **2021**, *63*, 208–228. https://doi.org/10.1137/19m1274067.
17. Yin, M.; Zhang, E.; Yu, Y.; et al. Interfacing Finite Elements with Deep Neural Operators for Fast Multiscale Modeling of Mechanics Problems. *Comput. Methods Appl. Mech. Eng.* **2022**, *402*, 115027. https://doi.org/10.1016/j.cma.2022.115027.
18. He, J.; Kushwaha, S.; Park, J.; et al. Sequential Deep Operator Networks (S-DeepONet) for Predicting Full-Field Solutions under Time-Dependent Loads. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107258. https://doi.org/10.1016/j.engappai.2023.107258.
19. Goswami, S.; Yin, M.; Yu, Y.; et al. A Physics-Informed Variational DeepONet for Predicting Crack Path in Quasi-Brittle Materials. *Comput. Methods Appl. Mech. Eng.* **2022**, *391*, 114587. https://doi.org/10.1016/j.cma.2022.114587.
20. He, J.; Koric, S.; Kushwaha, S.; et al. Novel DeepONet Architecture to Predict Stresses in Elastoplastic Structures with Variable Complex Geometries and Loads. *Comput. Methods Appl. Mech. Eng.* **2023**, *415*, 116277. https://doi.org/10.1016/j.cma.2023.116277.
21. Ahmed, B.; Qiu, Y.; Abueidda, D.W.; et al. Physics-Informed Deep Operator Networks with Stiffness-Based Loss Functions for Structural Response Prediction. *Eng. Appl. Artif. Intell.* **2025**, *144*, 110097. https://doi.org/10.1016/j.engappai.2025.110097.
22. Abueidda, D.W.; Pantidis, P.; Mobasher, M.E. DeepOKAN: Deep Operator Network Based on Kolmogorov Arnold Networks for Mechanics Problems. *Comput. Methods Appl. Mech. Eng.* **2025**, *436*, 117699. https://doi.org/10.1016/j.cma.2024.117699.
23. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; et al. Physics-Informed Machine Learning. *Nat Rev Phys* **2021**, *3*, 422–440. https://doi.org/10.1038/s42254-021-00314-5.
24. Chen, T.; Chen, H. Universal Approximation to Nonlinear Operators by Neural Networks with Arbitrary Activation Functions and Its Application to Dynamical Systems. *IEEE Trans. Neural Netw.* **1995**, *6*, 911–917. https://doi.org/10.1109/72.392253.
25. Hornik, K.; Stinchcombe, M.; White, H. Multilayer Feedforward Networks Are Universal Approximators. *Neural Netw.* **1989**, *2*, 359–366. https://doi.org/10.1016/0893-6080(89)90020-8.
26. Zhu, M.; Feng, S.; Lin, Y.; et al. Fourier-DeepONet: Fourier-Enhanced Deep Operator Networks for Full Waveform Inversion with Improved Accuracy, Generalizability, and Robustness. *Comput. Methods Appl. Mech. Eng.* **2023**, *416*, 116300. https://doi.org/10.1016/j.cma.2023.116300.
27. Spathopoulos, S.C.; Stavroulakis, G.E. Springback Prediction in Sheet Metal Forming, Based on Finite Element Analysis and Artificial Neural Network Approach. *Appl. Mech.* **2020**, *1*, 97–110. https://doi.org/10.3390/applmech1020007.