XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING

An approach for displacement field prediction in structures combining the Finite Element Method and Deep Learning Techniques

Mateus de Paula Ferreira[1], Elisa Dominguez Sotelino[1],

[1]*Dept. of Civil and Environmental Engineering, Pontifícia Universidade Católica do Rio de Janeiro*
Rua Marquês de São Vicente, 225, 22451-900, Rio de Janeiro, Brazil
mdepaulaferreira@aluno.puc-rio.br, sotelino@puc-rio.br

Abstract. Recent advances in machine learning have brought new possibilities for computational mechanics. This paper proposes a hybrid approach combining the Finite Element Method (FEM) and Deep Learning (DL) to predict displacement fields in elastic structures. The core idea is to replace the classical resolution methods for the system of equations by a neural networking surrogate capable of mapping stiffness matrices, load vectors and boundary conditions directly into nodal displacements. Three DL architectures – a Multilayer Perceptron (MLP), a Convolutional Neural Network based on ResNet18, and a Vision Transformer (ViT) – were evaluated in terms of accuracy, generalization using a dataset of two-dimensional plane stress elements structures. A curriculum learning strategy was employed to improve performance on increasingly complex boundary conditions and geometries. The results show that ResNet18 and ViT outperform the MLP, achieving R^2 values above 0.9 in some cases. For unstructured meshes, the use of the Cuthill–McKee algorithm for the reshape of matrices, significantly improved performance, with ViT reaching $R^2 \approx 1.00$. These findings indicate that the hybrid FEM-DL approach is promising for fast and accurate displacement predictions, with potential applications in large-scale or real-time structural analysis.

Keywords: Deep Learning, Finite Element Method, Computational Mechanics, Vision Transformers.

1 Introduction

The Finite Element Method (FEM) has revolutionized the analysis of structural systems by enabling accurate simulations of mechanical, thermal, and multi-physics problems (Zienkiewicz and Taylor [1]). FEM guarantees convergence when adequately formulated, but this comes at a computational cost, the more complex the processed models. For large-scale structures, such as bridges or oil platforms, the time required for a single analysis can become prohibitive (Badarinath et al. [2]).

In parallel, Deep Learning (DL) has shown remarkable performance in learning complex relationships from data. In computational mechanics, DL models have been explored as surrogates for expensive FEM solvers (Bolandi et al. [3][4]; Dong et al. [5]). These methods can predict stress or displacement fields directly, bypassing robust numerical methods.

In this study, another application of DNN for predicting nodal displacements in 2D elastic beams is introduced in combination with the Finite Element Method. In this approach, FEM is applied to obtain the stiffness matrix of a structural system and its corresponding vector of nodal loads and boundary conditions. Three models – Multilayer Perceptron (MLP), Convolutional Neural Network ResNet18 and Vision Transformer (ViT) – are tested as alternative solvers for the system of equations, with the main benefit being the instantaneous acquisition of mechanical analysis results.

2 Combining FEM and Deep Neural Networks

The construction of a structural FEM model involves generating the global stiffness matrix $\mathbf{K}$ and the nodal force vector $\mathbf{f}$. The displacement vector $\mathbf{u}$ is obtained by solving the system of the Eq. (1) with the imposition of the prescribed boundary conditions $\bar{\mathbf{u}}$.

$$\mathbf{f} = \mathbf{K}\mathbf{u} \tag{1}$$

In other words, a structure model is fully characterized once its set $\{\mathbf{K}, \mathbf{f}, \bar{\mathbf{u}}, \mathbf{u}\}$ is known. Indeed, all the numerical schemes applied to determine $\mathbf{u}$ can be seen as an operator φ that maps $\mathbf{K}$, $\mathbf{f}, \bar{\mathbf{u}}$ in the displacement field (Eq. (2)).

$$\mathbf{u} = \varphi(\mathbf{K}, \mathbf{f}, \bar{\mathbf{u}}) \tag{2}$$

Considering the availability of data generated by FEM simulations — in this case, sets of tuples in the form of $\{\mathbf{K}, \mathbf{f}, \bar{\mathbf{u}}, \mathbf{u}\}$ — the φ function can be approximated by a DNN model $\mathcal{N}$ – Eq. (3) – which learns its parameters from FEM results databases.

$$\mathbf{U} = \boldsymbol{\varphi}(\mathbf{K}, \mathbf{f}, \bar{\mathbf{u}}) \approx \boldsymbol{\mathcal{N}}(\mathbf{K}, \mathbf{f}, \bar{\mathbf{u}}) \tag{3}$$

The overall methodology for this study involves training a DNN model capable of predicting the displacement field in a structure without the need for matrix inversion or any other numerical method. The advantage of this approach is that the neural network receives the same inputs that are typically used for solving the problem through other numerical algorithms. Moreover, in contrast to other DNN models reported in the literature that usually rely on primary information such as node coordinates and elasticity modulus values, the proposed model has the structure information "encoded" in the input. Leveraging these inherent properties of the system allows the DNN to focus on learning patterns from higher-level structural features (stiffness coefficients, load vectors, and boundary conditions) instead of learning a new representation for this and then mapping it to the displacement field, potentially improving both the accuracy and efficiency of the solution. A diverse 2D mesh dataset, where used with each structural domain discretized using quadrilateral finite elements in plane stress state. This choice simplifies the mechanical formulation while maintaining all essential characteristics of the stiffness matrix and displacement field required for validating the proposed FEM–DL framework. The methodology, however, remains general and can be adapted to other element formulations such as plane strain or solid elements.

3 Methodology

3.1 Neural architecture

Three deep learning models were tested to identify the most effective architecture for the displacement prediction task:

- ***Multilayer Perceptron (MLP)***: The MLP is a fully connected network that receives the flattened $\mathbf{K}$, $\mathbf{f}, \bar{\mathbf{u}}$ as input. It consists of three hidden layers with [512, 256, 128] neurons, ReLU activation, and dropout regularization. Although simple, this architecture serves as a baseline for comparison.

- ***Convolutional Neural Network – ResNet18***: The ResNet18 architecture, known for its residual connections, was chosen to exploit the spatial structure of the input tensors. The residual blocks help avoid the vanishing gradient problem and allow the network to capture hierarchical patterns in the stiffness matrix and load distribution. This architecture is well-suited to structured meshes and grid-like data.

- ***Vision Transformer (ViT)***: The ViT model applies self-attention mechanisms to non-overlapping patches of the input matrices, learning global relationships across all nodes simultaneously. Unlike CNNs, which rely on local receptive fields, the ViT can model long-range interactions, which is advantageous for complex or irregular structures.

All models were trained with and without L2 weight decay, resulting in configurations like MLP-L0 (no regularization) and MLP-L2 (with regularization).

3.2 Dataset generation and parameter sampling

Datasets were generated by randomly varying geometric and material parameters within realistic design ranges, as shown in **Fig 1**. Each sample corresponds to a complete FEM analysis (including boundary conditions and nodal forces), which was exported to a standardized format for training.

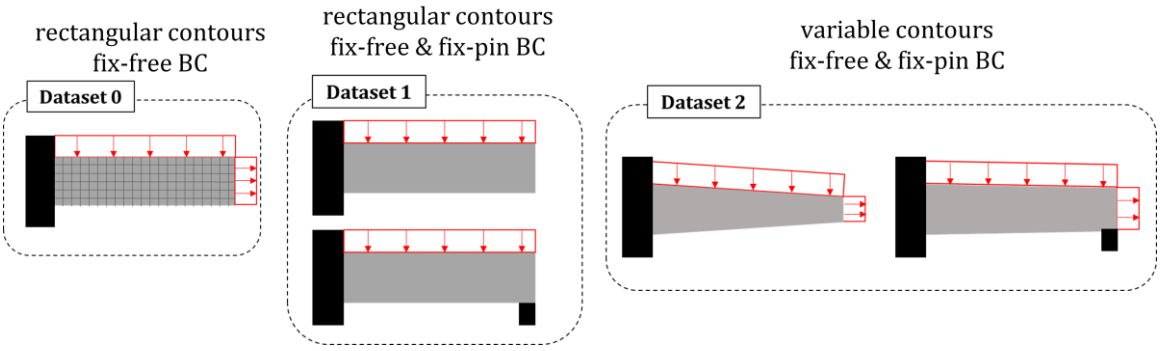


Fig 1 Overview of the datasets used for model training and evaluation, illustrating different geometries and boundary conditions

- ***Dataset 0***: The first dataset includes 2D rectangular cantilever beam configurations. The length, height, and thickness of the beams varied randomly.
- ***Dataset 1***: The second dataset extends Dataset 1 by introducing pinned supports and mixed BCs.
- ***Dataset 2***: The third dataset comprises non-rectangular domains (trapezoidal and irregular contours).

To ensure sufficient variability for effective learning, geometric and material parameters were randomly selected within realistic design limits.

Table 1 Global parameters for varying in dataset

Property	Range
Length	1000 – 2000 mm
Height	250 – 300 mm
Thickness	100 – 500 mm
Young modulus	25– 200 MN/mm²
Poisson ratio	0.30
Load value	1000 – 2000 N/mm

For each configuration, nodal displacements **u** were obtained using a FEM solver. The resulting datasets contained 12,000 samples, enabling the training of deep models.

3.3 Data preparation and normalization

The neural networks require well-conditioned inputs to learn effectively. For this purpose:

- **Normalization of matrices:** The entries of **K** and **f**, which can vary by several orders of magnitude, were scaled to the range [-1, 1].
- **Load vector transformation:** The **f** vector was reshaped into a diagonal matrix of the same dimension as **K**, ensuring that the input channel structure matched the expected input of CNN and ViT models.
- **Concatenation of channels:** The final input to the networks was a 3D tensor, where the first channel corresponds to the normalized stiffness matrix, the second the load matrix and the third the boundary conditions (0 or 1 for free or constrained degrees of freedom).

3.4 Training strategy and curriculum learning

The training pipeline – shown in Fig 2 - was structured with a curriculum learning. This staged approach improves convergence stability and prevents the networks from diverging or overfitting to simple cases, leading to smoother learning dynamics and better performance on complex datasets. This was performed as follows:

- **Phase 1:** Models were first trained on *Dataset 0* (simplest case).
- **Phase 2:** The pre-trained weights were fine-tuned on *Dataset 1*, where boundary conditions are more diverse.
- **Phase 3:** Finally, the networks were tested and fine-tuned on *Dataset 2*, containing variable geometries, boundary conditions and unstructured meshes.

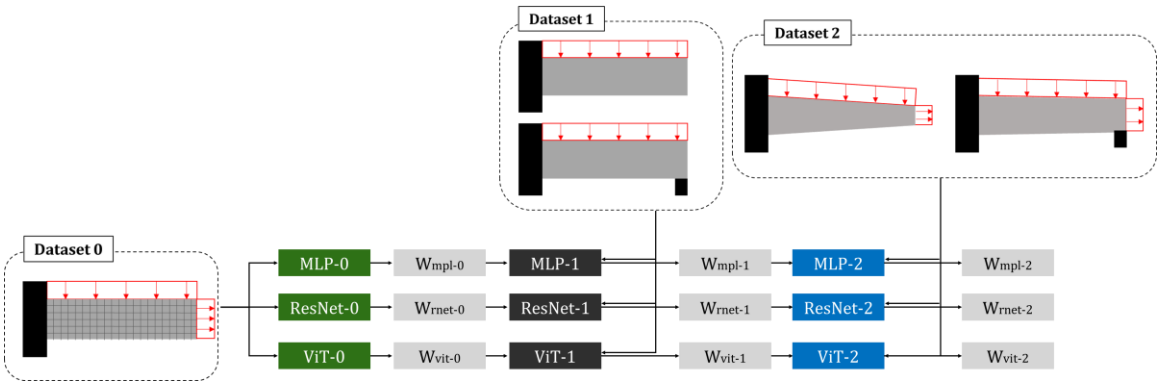


Fig 2 Schematic representation of the training strategy for the three architectures (MLP, ResNet, and ViT) using curriculum learning.

This strategy was combined with the training properties:

- **Loss function:** Mean Squared Error (MSE) was chosen as the primary loss.
- **Optimizer:** Adam optimizer with a learning rate of 0.001.
- **Early stopping:** Training was halted if validation loss did not improve after 20 epochs.
- **Batch size:** 32 samples.

4 Results

The three neural architectures – MLP, ResNet18, and ViT – were evaluated across all datasets, from simple rectangular domains to irregular geometries and unstructured meshes. The performance is reported in terms of the determination coefficient (R^2) and Mean Squared Error (MSE), providing both a relative and absolute measure of predictive accuracy.

4.1 Results on Dataset 0

In the simplest scenario – shown Fig 3 - with regular rectangular geometries and fixed–free boundary conditions, all models performed reasonably well, achieving MSE below 3.0. The ResNet18-L0 configuration was the best-performing model, with $R^2 = 0.85$ and MSE = 1.59, clearly benefiting from convolutional feature extraction. Both MLP and ViT obtained similar results ($R2 \approx 0.79$), indicating that for simple datasets the attention mechanism of the ViT did not provide an advantage over traditional fully connected layers. Interestingly, applying L2 regularization slightly reduced the performance of all models in this dataset, suggesting that overfitting was not a concern in the low-complexity regime. Results are summarized in Table 2.

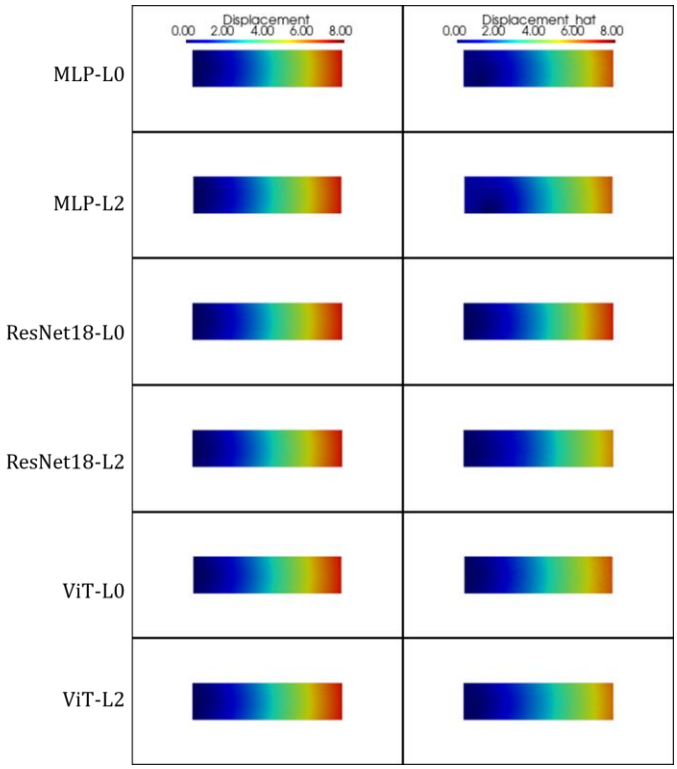


Fig 3 Displacement field for a random sample from Dataset 0.

Table 2 Metrics for Dataset 0

Model	R^2	MSE
MLP-L0	0.79	2.12
MLP-L2	0.78	2.44
ResNet18-L0	**0.85**	**1.59**
ResNet18-L2	0.79	2.16
ViT-L0	0.79	2.11
ViT-L2	0.79	2.12

4.2 Results on Dataset 1

When the complexity increased with the introduction of mixed fixed–free and fixed–pin supports, the models' behaviors diverged – Fig 4. ResNet18-L2 achieved outstanding performance, with $R^2 = 0.94$ and MSE = 0.43, outperforming all other configurations by a wide margin. The regularization L2 played a key role here, helping CNN to generalize across different boundary conditions without memorizing specific patterns. The ViT-L0 also performed strongly ($R^2 = 0.91$ and MSE = 0.65), demonstrating the benefit of attention-based mechanisms for recognizing long-range dependencies between nodes. Conversely, the MLP models suffered a significant performance drop ($R^2 < 0.35$), which highlights their limitations in scenarios with complex boundary conditions and high geometric variability. Results are summarized in Table 3.

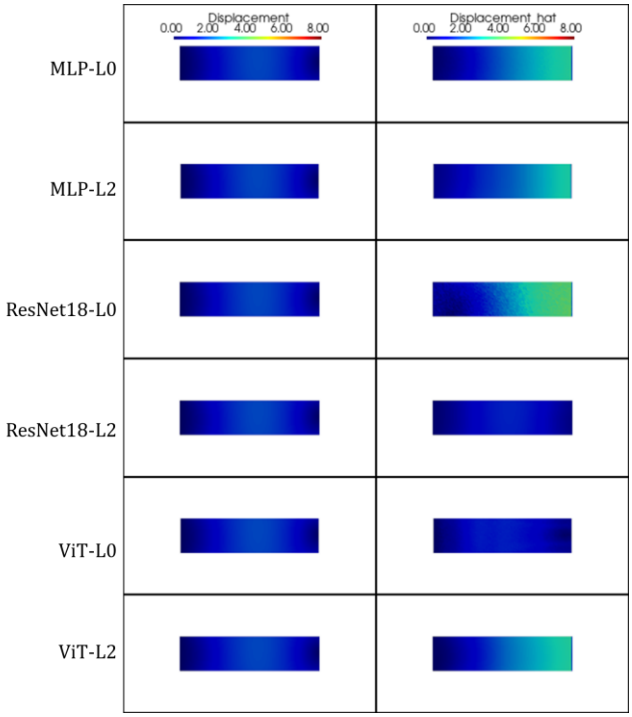


Fig 4 Displacement field for a random sample from Dataset 1

Table 3 Metrics for Dataset 1

Model	R^2	MSE
MLP-L0	0,32	4,99
MLP-L2	0,31	5,29
ResNet18-L0	0,60	3,96
ResNet18-L2	**0,94**	**0,43**
ViT-L0	0,91	0,65
ViT-L2	0,32	4,99

4.3 Results on Dataset 2

For variable geometries and irregular contours, all models struggled to generalize, leading to a dramatic performance collapse. None of the tested architectures achieved a positive R^2; for example, ResNet18-L2, the most stable model in earlier datasets, only reached R^2 = -1439.08 and MSE = 38.99. The ViT models performed even worse, with highly negative R^2 values and MSEs exceeding 200.0. This indicates that the lack of explicit connectivity or positional encoding severely hampers the ability of the networks to map stiffness and loads to displacements in irregular meshes. It became evident that traditional data reshaping (flattening or resizing to a grid) is insufficient to capture the structural relationships in these scenarios. Results are summarized in Table 3.

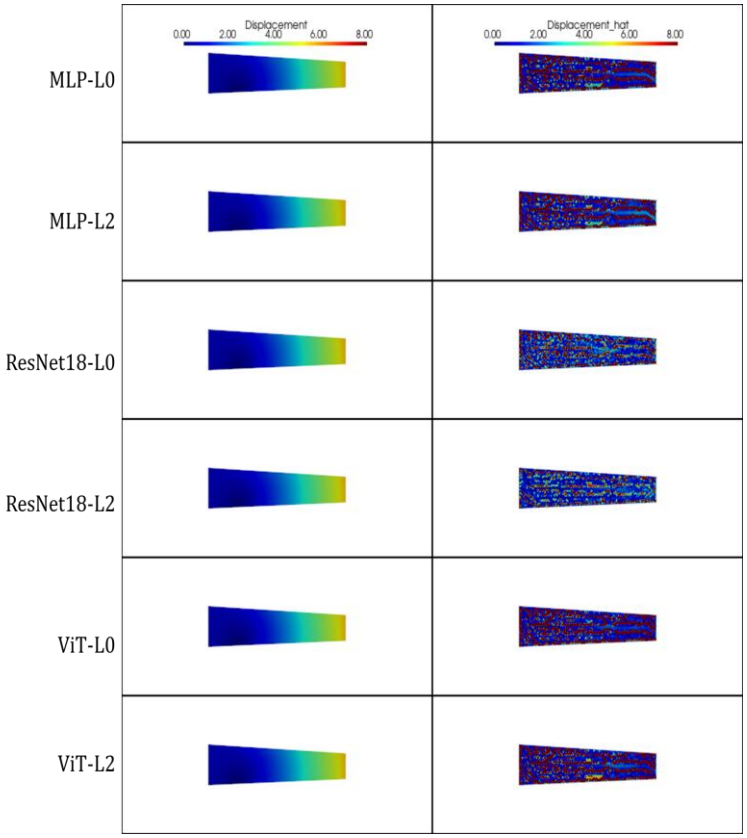


Fig 5 Displacement field for a random sample from Dataset 2

Table 4 Metrics for Dataset 2

Model	R^2	MSE
MLP-L0	-5834.76	164.20
MLP-L2	-6092.38	173.02
ResNet18-L0	-1562.87	42.32
ResNet18-L2	**-1439.08**	**38.99**
ViT-L0	-8594.55	232.04
ViT-L2	-8735.12	235.87

4.4 Results on unstructured meshes with Cuthill–McKee

To overcome the limitations observed in Dataset 2, the Cuthill–McKee algorithm was employed. The improvement observed – Fig 6 – stems from its ability to reduce the bandwidth of the stiffness matrix, effectively reordering the nodes so that connected degrees of freedom are positioned closer in the matrix representation. This reordering restores a quasi-structured spatial relationship in the input tensors, allowing convolutional filters and attention mechanisms to better exploit local and topological correlations that were otherwise dispersed in the original unstructured configuration. The ViT-L0 achieved nearly perfect results, R^2 = 1.00 and MSE = 0.01, essentially reproducing the FEM results with negligible error. ResNet18 configurations also improved significantly, maintaining MSE values close to 1.0. Even the MLP baseline, which had previously performed poorly, achieved R^2 = 0.77 and MSE = 1.21, highlighting the crucial role of data ordering in all tested models. Results are summarized in Table 4.

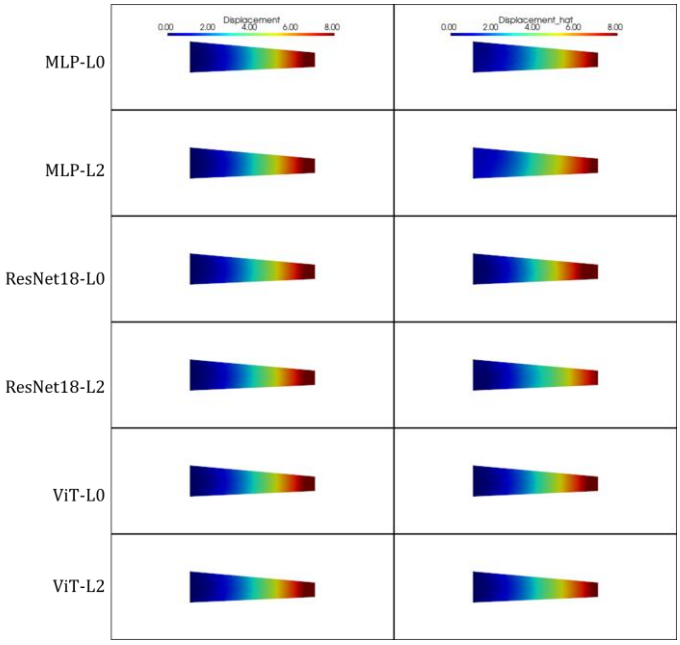


Fig 6 Displacement field for a random sample from Dataset 2 with Cuthill-McKee

Table 5 Metrics for Dataset 2 with Cuthill-McKee

Model	R^2	MSE
MLP-L0	0.77	1.21
MLP-L2	0.71	1.87
ResNet18-L0	0.79	1.11
ResNet18-L2	0.77	1.19
ViT-L0	**1.00**	**0.01**
ViT-L2	0.78	1.15

4.5 Discussion of training behavior

Beyond raw performance metrics, training revealed distinct behaviors for each architecture. ResNet18 and ViT converged rapidly, typically within 50–70 epochs, while the MLP required over 100 epochs and still exhibited signs of underfitting or overfitting depending on dataset complexity. The use of L2 regularization proved particularly beneficial for ResNet18 when boundary conditions varied widely, as in *Dataset 1*, but had a negligible effect on ViT. Although ViT achieved near-perfect accuracy on the reordered unstructured meshes, its training curves suggested a tendency to memorize data patterns, indicating potential overfitting when exposed to smaller datasets.

5 Conclusions

This study proposed a hybrid FEM–DL approach for predicting displacement fields in elastic structures, avoiding the explicit resolution of the FEM system by using Deep Neural Networks as surrogate solvers. Among the evaluated architectures, ResNet18 and ViT achieved the best results, with ResNet18-L2 performing particularly well on structured cases ($R^2 = 0.94$), while ViT-L0 reached near-perfect accuracy ($R^2 = 1.00$) after applying the Cuthill–McKee reordering on unstructured meshes. These results indicate that combining FEM data generation with DL models is a feasible and efficient alternative to classical numerical solvers. However, their performance remains sensitive to input representation and mesh irregularities, pointing to future improvements through graph-based models and physics-informed architectures.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors or has the permission of the owners to be included here.

References

[1] Zienkiewicz, O. C. and Taylor, R. L., "The Finite Element Method", 4th Edition, vol. 1: "*The Finite Element Method: From Intuition to Generality*",Appl. Mech. Rev., 23, no. 23, pp. 249-256, March, 1970.

[2] Badarinath, P. V.; Chierichetti, M.; Kahkhi, F. D. "A machine learning approach as a surrogate for a finite element analysis: Status of research and application to one dimensional systems". *Sensors*, vol. 21, pp. 1–18, 2021.

[3] Bolandi, H. et al. "Bridging finite element and deep learning: High-resolution stress distribution prediction in structural components". *Frontiers of Structural and Civil Engineering*, vol. 16, pp. 1365–1377, 2022b

[4] Bolandi, H. et al. "Deep learning paradigm for prediction of stress distribution in damaged structural components with stress concentrations". *Advances in Engineering Software*, vol. 173, 2022a.

[5] Dong, Y. et al. "DeepFEM: A novel element-based deep learning approach for solving nonlinear partial differential equations in computational solid mechanics". *Journal of Engineering Mechanics*, vol. 149, 2023.