Proceedings of the IASS 2024 Symposium
Redefining the Art of Structural Design
August 26-30, 2024, Zurich Switzerland
Philippe Block, Giulia Boller, Catherine DeWolf,
Jacqueline Pauli, Walter Kaufmann (eds.)

Structural analysis of concrete shells using deep learning methods

Maxime POLLET*, Paul SHEPHERD[a], Will HAWKINS[b], Eduardo COSTA[c]

* University of Bath
Department of Architecture and Civil Engineering, University of Bath, Bath, UK
mp2333@bath.ac.uk

[c] University of the West of England

Abstract

The structural behaviour of concrete shells is complex, which typically makes their design and production more difficult than prismatic structures. The Finite Element (FE) method is often used for the structural analysis of shells, but obtaining accurate results can be computationally expensive. The present research investigates the use of deep learning techniques to estimate rapidly and accurately the structural behaviour of concrete shells. While these models require a large initial time investment to generate a training dataset and to fit the models, they can then make predictions in a few seconds. Using a flooring thin-shell system as a test-case, a dataset of 20,000 shells with varying spans, heights, thicknesses, and material properties was generated. Linear FE analysis was used to determine the stresses and the buckling factor of the shells under a load case combining self-weight and live loads. Two types of deep learning models, a Multilayer Perceptron (MLP) and a Convolutional Neural Network (CNN) were trained to predict the stress and the buckling behaviour of shells. The results obtained highlight the ability of deep learning models to predict rapidly and accurately the stresses and the buckling factor of concrete shells, as the errors measured are consistently below 2%.

Keywords: Concrete, Shells, Structural analysis, Finite Element Analysis, Buckling, Stress, Machine Learning, Deep Learning, Multilayer Perceptron, Convolutional Neural Network

1. Introduction

Surrogate models are built using techniques from the field of statistical learning to find functions that map the inputs of a simulation model to its outputs [1]. These data-driven models are built by "learning" from the results of physical experiments, or from digital simulations, such as Finite Element (FE) analysis. They require a large initial time investment to collect the simulation samples and build a dataset but can be significantly quicker than a simulation once the model is built [2]. While various types of surrogate models exist, recent research focuses on models from the deep learning field [3,4]. In the field of biomechanics, Multilayer Perceptrons (MLPs) were successfully trained using FE data for various applications such as the mechanical behaviour of the aorta [5], of arterial walls [6], and of the liver [7]. In structural engineering, MLPs have been trained to predict the structural behaviour of truss structures [8], and of steel joints [9]. The use of more complex model architectures has also been explored. Convolutional Neural Networks (CNNs), which make use of image data, have been used to predict stress fields [10,11]. The outputs of mesh-based simulations were predicted using Graph Neural Networks (GNNs), which can learn from data structured in graphs [12]. Physics-Informed Neural Networks (PINNs) were developed to embed the laws of physics in deep learning models [13]. The PINN architecture was also adapted to surrogate modelling in the field of solid mechanics [14]. Generative Adversarial Networks (GANs), which rely on the competition of two networks, have been used successfully for the prediction of stress fields [15,16] and for the prediction of buckling modes [17]. Regarding the structural system of interest, several studies [16,17,18] have used deep learning models

to predict the structural behaviour of thin-walled structures, more specifically for stress and buckling prediction. However, as per the authors' knowledge, the structural behaviour of concrete shells has never been predicted using deep learning.

The present research seeks to adapt deep learning techniques to predict the structural behaviour of concrete shells. The capability to estimate rapidly and accurately the buckling factor and the stresses of concrete shells could have several applications in their design and production. For example, these estimations could be used to guide form-finding and design optimisation, which typically requires many iterations, in which various shapes are explored and assessed.

Shells are non-prismatic structures with shapes designed to maximise membrane action. Therefore, transforming their shapes into a data structure that can be used as input to a deep learning model is a key task for predicting the structural behaviour of shells. In this study, two different deep learning model architectures, MLPs and CNNs, which use two different data structures as input and, as a result, two different methods to represent shells shapes, are investigated. They are trained to predict the stress fields and the buckling factor under a load case combining self-weight and live loads, in order to determine their relative performance and suitability for this novel application. A concrete thin-shell flooring system is used as a test case. This structure is inspired from the flooring system developed by the ACORN research project (Figure 1), with which this research is aligned, and which can achieve a 48% reduction in embodied carbon relative to an equivalent flat slab [19].

Figure 1: ACORN's full-scale segmented concrete thin-shell prototype © Paul Shepherd

2. Methods

2.1. General overview

A dataset of 20,000 thin-shells with various spans, heights, thicknesses, and material properties was computationally generated (sections 2.2.1. and 2.2.2.). An FE analysis was then performed for each to determine the stresses and the buckling factor when subjected to a combination of self-weight and live loads (section 2.2.3.). These results were compiled into a stress dataset and a buckling dataset, which were used to train MLPs and CNNs to predict stresses and buckling factors (sections 2.3.1. and 2.3.2.). The tuning of the hyperparameters of the models was done in two steps. First, a random search was performed to explore various model configurations. Using these results as a starting point, the models were then further improved through manual tuning (section 2.3.2.). Finally, the performances of the best MLP and the best CNN were compared (section 2.3.3.).

2.2 Constitution of the dataset

2.2.1. Structural system

While the dataset used to train the surrogate models comprises samples which are all different, they still share certain characteristics. In this study, the structural system of interest is adapted from the segmented thin-shell floor developed in the scope of the ACORN research project [19]. However, rather than an assemblage of segments, the system is a single monolith, which significantly simplifies the structural analysis. This monolith spans over a square area between four columns, and its corners are trimmed, thus forming a flat support. As an example, one meshed sample is shown in Figure 2.

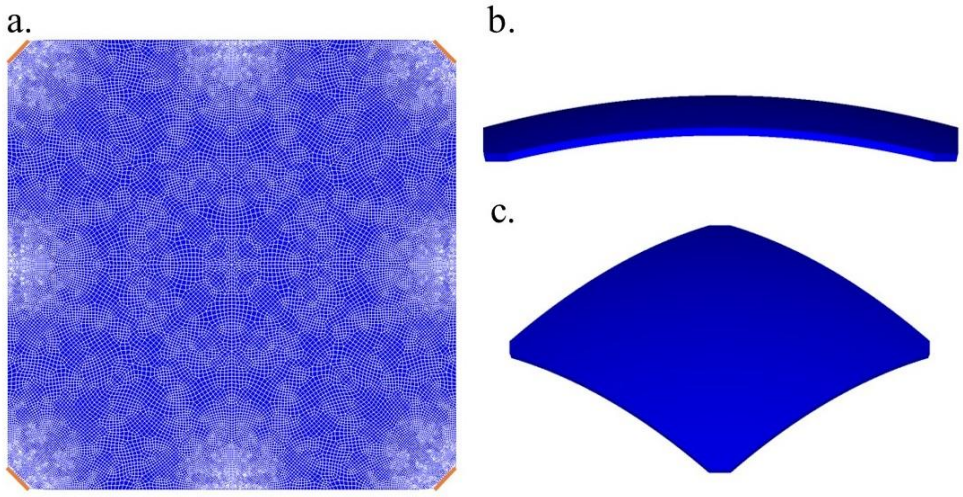

Figure 2: Thin-shell sample with a. a plan view (meshed) with supports in orange, b. an elevation, and c. an isometric view

The material of all samples is unreinforced concrete with a Poisson's ratio of 0.2. Regarding the loads, the self-weight of the concrete is taken as 2,500 kg/m^3. The superimposed dead load on top of the thin-shells, and the live loads are respectively equal to 1 kN/m^2 and 2.5 kN/m^2 of projected flat floor area (typical for UK office buildings [20]). All loads are considered to act in the vertical direction and were combined with the load factors prescribed by Eurocodes [21] for the Ultimate Limit State (ULS). Thus, a load factor of 1.35 was used for the self-weight and the superimposed dead load, and a load factor of 1.5 was used for live loads.

2.2.2. Design space and sampling plan

The dataset comprises 20,000 thin-shells with various shapes and material properties, which were sampled from a design space determined by the span (from 2m to 10m), height (from 1/12th to 1/8th of the span), thickness (from 1/150th to 1/50th of the span), and Young's modulus (from 27.0 GPa to 44.0 GPa). An important task in the constitution of a dataset is the selection of samples from the design space. Sampling plans are based on the general idea that a surrogate model needs to be built upon a uniform spread of datapoints to achieve a uniform level of accuracy over the design space [2]. A key requirement for the choice of training samples is therefore to ensure that they are "space-filling". In this study, the Sobol sequence with scrambling [22] was used to select 20,000 samples from the design space. The shapes of the 20,000 thin-shells were then form-found using the Updated Reference Strategy method [23].

2.2.3. Finite Element model

All linear FE analyses were performed using the Python implementation of ANSYS [24]. Three layers of eight-noded elements (SOLID185 [25]) were used through the thickness. The superimposed dead load and the live loads were applied on the nodes located on the extrados (upper) face of the thin-shells. To avoid stress singularities that can be caused by fully fixed nodes, elastic supports, which were modelled using ANSYS' SURF154 elements [25], were used and their stiffness was set to 3.5*10^{14} Pa. The density of the mesh was chosen after a mesh convergence study. The mesh chosen is composed of four layers of 33,681 nodes, which form three layers of 33,248 elements. The density is refined at the corners and at the mid-span edges because these areas were found to have a higher stress gradient, as shown in Figure 2. The same mesh topology was used for all samples in the dataset, after scaling the nodes coordinates to fit the span, height, and thickness of each thin-shell. The FE analyses were used to determine the three principal stresses σ_1, σ_2 and σ_3, as well as the eigenvalue buckling factor, defined as the factor of safety against buckling for a given loading.

2.3. Training of the deep learning models

2.3.1. Multilayer Perceptron

The first type of surrogate model investigated for the prediction of the structural behaviour of concrete thin-shells is the MLP. The input layer of this model describes the shape and material properties of each shell. The geometrical descriptors include the span, the height, and the thickness of each sample. Additional descriptors were obtained through Principal Component Analysis (PCA), which is commonly used for feature selection in machine learning [26]. It was used to extract features from the mesh node coordinates of each thin-shell, by projecting it to a lower dimensional space, which principal axes represent the directions of maximum variance in the data. Even though 150 principal components were initially selected to ensure that enough features were extracted, it was found that the first three principal components respectively explained 95.71%, 1.79%, and 0.44% of the variance in the data respectively. However, to ensure that an important feature would not be left aside, the number of PCA features used as input for the MLP was used as a variable in the hyperparameter tuning process (see section 2.3.3.). The only material property used as input to the MLP is the Young's modulus of each sample. The output layer is different for the buckling prediction MLP and for the stress prediction MLP. For buckling prediction, the output layer of the MLP is a single value, the buckling factor. For stress prediction, the dimensionality of the output layer had to be reduced, because predicting the three principal stresses for each node would have resulted in 404,172 output neurons, which would have had a high computational cost. Instead, a low-dimensional representation of the stresses was computed using PCA and the MLPs were trained to predict the first 1,000 components. However, the error metrics were computed after transforming the low-dimensional outputs of the MLP back to the original stress space.

2.3.2 Convolutional Neural Network

The second type of model investigated is the CNN. Unlike MLPs, CNNs are designed to learn and extract features from images. The first part of their architecture typically includes convolutional and pooling layers, which transform an input image into a feature map and reduce the dimensions of the data. The output matrix of these layers is then flattened and fed into the second part of their architecture, which is composed of fully connected layers, like the ones in a MLP [27]. For each sample in the dataset, a 99x99 pixels height map of the thin-shell was created. Each pixel was assigned the height of the thin-shell at the centre of the pixel, except for the pixels located out of the structure in the corners, which were each assigned a value of 0. These height maps constitute the only input to the convolutional part of the CNN. However, additional scalars are used as inputs to the fully connected layers, which include the span, the height, the thickness, and the Young's modulus of each sample. These values are concatenated to the flattened matrix that is fed into the fully connected layers. Given that the last part of the CNN architecture is similar to a MLP, the output layer is the same as described for the MLP for buckling and stress prediction.

2.3.3. Hyperparameter tuning

A deep learning model is typically difficult to optimize, as many hyperparameters related to its architecture (such as the number of layers for a MLP, or the kernel size for a CNN) and to the fitting procedure (such as the learning rate or the batch size) need to be tuned [28]. Optimal combinations can therefore be difficult to find. In this research, the hyperparameter tuning was performed in a two-step process for each model type – MLP and CNN – and for each prediction task – buckling and stress. In each case, a random search that explored 50 different model configurations was first performed. This approach was found to determine relevant hyper-parameter combinations much faster than the traditional grid search approach [29]. For the MLP, the search space had 11 dimensions: two related to the inputs (number of PCA components used and normalisation), six related to the model itself (number of hidden layers, number of neurons per layer, activation function, use of dropout and batch normalisation layers, and weight initialisation method), and three related to the fitting method (optimisation algorithm, learning rate, and batch size). In the case of the CNNs, the hyperparameters tuned were adapted from [30]. Five were used to define the convolutional part of the CNN: number of convolutional layers, number of kernels per layer, kernel size, pooling method, and pooling size. The hyperparameters

defining the fully connected layers of the CNN and the fitting procedure were the same as the ones used for the MLP. The number of epochs was not included as a hyperparameter, as an early-stopping approach, which interrupts training when the model is not progressing anymore, was preferred. The Mean Squared Error was used as a loss for all model configurations. The few best performing models in each random search were selected for the second part of the hyperparameter tuning process, where these were manually tuned to further improve performance. In the manual tuning phase, more advanced model training methods were implemented, such as learning rate scheduling which allows the learning rate to be reduced dynamically based on validation measurements.

2.3.4. Measuring model performance

The traditional approach to assess the performance of a machine learning model is to divide the dataset into three parts: a training set, a validation set, and a testing set. The training set is used to fit the model, the validation set is used for assessing its accuracy, choosing a model type, and tuning its parameters, and the testing set is used to measure the generalization error of the final model [31]. This approach was adopted, and the dataset of 20,000 samples was split into a training set with 12,800 samples, a validation set with 3,200 samples, and a testing set with 4,000 samples. The performances of the models were measured using several metrics that are commonly used for regression tasks. The ones that were selected for this study are the Mean Absolute Error (MAE), the Root Mean Squared Error (RMSE), and the Mean Absolute Percentage Error (MAPE). They are defined as follows:

$$MAE = \frac{1}{N}\sum_i^N |y_i - \hat{y}_i| \tag{1}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_i^N (y_i - \hat{y}_i)^2} \tag{1}$$

$$MAPE = \frac{1}{N}\sum_i^N \left|\frac{y_i - \hat{y}_i}{y_i}\right| * 100\% \tag{3}$$

In the above equations, N is the number of samples, y_i is a tensor of the actual targets, and $\hat{y}_i$ is a tensor of the predictions. Additional metrics were used for each prediction task. For buckling factor prediction, the $MAPE_{5\%}$ was used, which measures the MAPE for the 5% lowest buckling factors in the test set, which are the most critical for structural design. For stress prediction, the $MAPE_{peak}$ metric was adapted from [5] and used. It is defined as follows:

$$MAPE_{peak} = \frac{1}{N}\sum_i^N \left|\frac{\max(|S|) - \max(|\hat{S}|)}{\max(|S|)}\right| * 100\% \tag{4}$$

Where max(|S|) is the peak stress obtained using FE analysis and max($|\hat{S}|$) is the peak stress obtained using a deep learning model. All the metrics used for stress prediction are computed separately on each of the principal stresses σ_1, σ_2 and σ_3.

3. Results and discussion

3.1. Buckling prediction

The best-performing MLP was obtained using 25 PCA components and normalised inputs. It is composed of 5 hidden layers, each composed of 4096 neurons. The activation function used is the Sigmoid function, and the weights were initialised using values drawn from a uniform distribution. The model was fitted using the Adamax optimisation algorithm [32], an initial learning rate of 0.001, and a batch size of 32. For the CNN, the best results were obtained using 4 convolutional layers, with 32 kernels of size 3x3 per layer. Maximum pooling was used with a pooling size of 7x7. The fully connected layers comprise 3 hidden layers, each composed of 256 neurons. The weights were initialised using a

semi orthogonal matrix, as described in [33]. The CNN was fitted using the Adamax algorithm, with an initial learning rate of 0.01 and a batch size of 16. The performance of both optimal models on the test set is summarised in Table 1.

Table 1: Performance of deep learning models for buckling factor prediction

Metric		MAE	RMSE	MAPE (%)	$MAPE_{5\%}$ (%)
Model type	MLP	0.371	0.526	1.12	1.79
	CNN	0.0644	0.0948	0.228	0.642

They both succeeded in modelling the buckling behaviour of concrete thin-shells accurately, with MAPE values below 2%. The CNN performed significantly better than the MLP according to all metrics, with a MAPE of 0.228%. Regarding the prediction of the lowest buckling factors - which are the most critical in the context of structural design – both models were found to perform slightly worse, with $MAPE_{5\%}$ of 1.79% for the MLP and 0.642% for the CNN. The Mean Squared Error, which was used as a loss in the model training, could potentially be responsible for this difference, since the squaring of each error effectively weights large errors more heavily than small ones. As a result, the lowest buckling factors could potentially have a lower impact than the highest buckling factors on model fitting. Nevertheless, the $MAPE_{5\%}$ of both models is still relatively low, as it is below 2% for the MLP and below 1% for the CNN. Further investigation was conducted to determine if the magnitude of the errors is correlated to the dimensions of the design space. It was found that the thickness-to-span ratio and absolute percentage error have a Pearson correlation coefficient of -0.347 for the MLP and -0.442 for the CNN, which suggests that the buckling factor of thin-shells with a low thickness relative to the span are more difficult to predict for both models. All the other dimensions of the design space were not found to have a significant correlation with the absolute percentage error.

3.2. Stress prediction

For stress prediction, the best-performing MLP was obtained using 142 PCA components and normalised inputs. It is composed of 5 hidden layers, each composed of 1024 neurons. The activation function used is the Rectified Linear Unit function, and the weights were initialised using a semi orthogonal matrix. The model was fitted using the Stochastic Gradient Descent optimisation algorithm, an initial learning rate of 0.01, and a batch size of 32. The best CNN was obtained using 4 convolutional layers, with 4 kernels of size 7x7 per layer. Average pooling was used with a pooling size of 9x9. The fully connected layers comprise 4 hidden layers, each composed of 512 neurons. The weights were initialised using values drawn from a uniform distribution. The CNN was fitted using the Adamax algorithm, with an initial learning rate of 0.01 and a batch size of 16. The performance of both models on the test set is summarised in Table 2.

Table 2: Performance of deep learning models for stress prediction

Metric		MAE (kPa)			RMSE (kPa)			$MAPE_{peak}$ (%)		
Principal stress component		σ_1	σ_2	σ_3	σ_1	σ_2	σ_3	σ_1	σ_2	σ_3
Model type	MLP	4.72	2.84	9.31	20.6	10.8	30.9	0.631	0.761	0.402
	CNN	2.04	1.43	3.71	5.68	4.38	7.68	0.435	0.414	0.186

The results obtained for the MLP and the CNN demonstrate their capability to predict accurately stress fields, as the $MAPE_{peak}$ is below 1% for all principal stresses. For both models, the largest MAE and the lowest $MAPE_{peak}$ are measured on σ_3, the principal stress component with the highest compressive stresses. However, according to all the metrics, the CNN performs better than the MLP. Indeed, for the CNN, the $MAPE_{peak}$ is below 0.5% for all stress components, which is the case for only one principal

stress component for the MLP. An example of a stress field predicted by both models is shown in Figure 3.

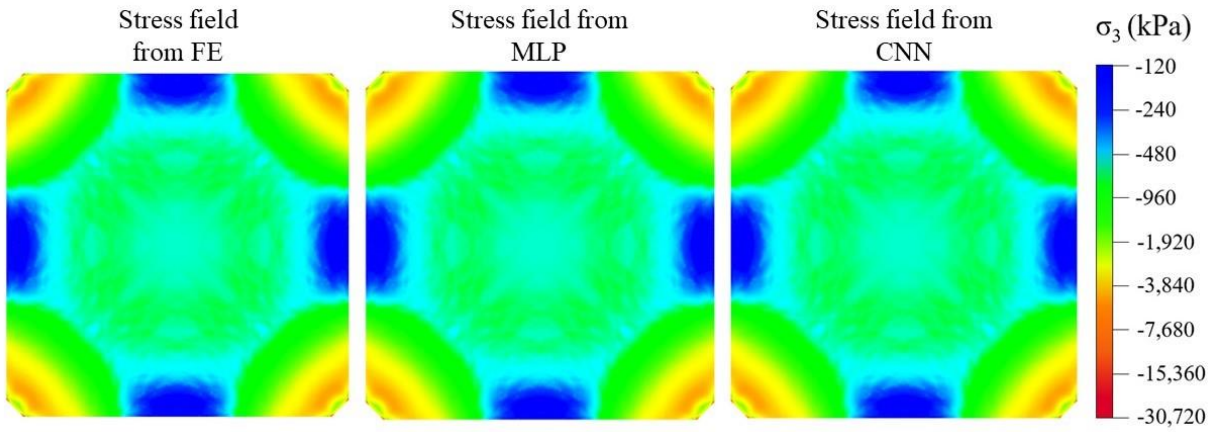

Figure 3: An example of stress field prediction

Further investigation revealed that the $MAPE_{peak}$ is correlated with the span. For the MLP, Pearson correlation coefficients of 0.321, 0.268, and 0.301 were measured for the correlation of the span with the $MAPE_{peak}$ of σ_1, σ_2 and σ_3 respectively. In the case of the CNN, these correlations are respectively equal to -0.246, -0.012, and -0.449. This suggests that the MLP is better at predicting the stress fields of the thin-shells with the lowest spans, while the CNN is more accurate with the highest spans. The Pearson correlation coefficients of the other dimensions of the design space were not found to be significant. Additionally, the spatial distribution of errors was investigated. The MAE was computed for each node across all thin-shells and plotted on a flat normalised projection of the mesh. As shown in Figure 4, the highest MAE values are measured for the nodes that are near the corners of the thin-shells. Additionally, the better performance of the CNN compared to the MLP is clearly visible. The same behaviour was observed for all principal stresses regardless of the relative position through the thickness of the thin-shells.

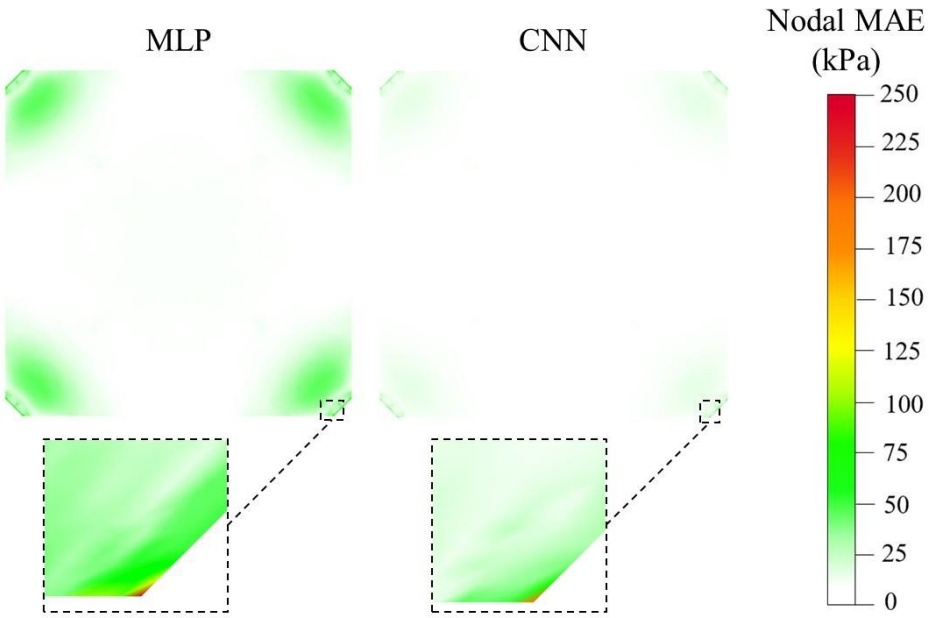

Figure 4: Nodal MAE for σ_3 in the intrados (lower) face of the thin-shells

4. Conclusion

This research presented a deep learning-based approach to determine the stresses and the buckling factor of concrete thin-shells. Two types of deep learning model were investigated, MLP and CNN. It was found that both models were able to predict the buckling behaviour accurately, as the MAPE values measured are consistently below 2%. Regarding stress prediction, the same conclusions can be made since both models have $MAPE_{peak}$ values below 1%. For both prediction tasks, the best performing CNN was found to be more accurate than the best performing MLP. The capability to estimate rapidly and accurately the critical buckling load and the stresses of concrete shells could enable several applications in the design and production of concrete shells. For design, these estimations could be used to guide form-finding. In the production phase, the impact of geometric imperfections caused by manufacturing processes could be assessed before the concrete sets, thus enabling potential corrections to be applied. Future research will focus on the use of different model types, such as GNNs, which can learn from graph-structured data, and mixed-fidelity models, which use a mix of low-fidelity and high-fidelity simulations for model training. Additionally, geometric imperfections will be introduced in the thin-shells to investigate the potential use of deep learning models in the manufacturing context and for the prediction of more complex stress fields. The FE model will also be improved with the modelling of geometric and material nonlinearities.

Acknowledgements

This work is aligned with the ACORN research project, which was funded by UK Research and Innovation (EP/S031316/1). The authors would also like to acknowledge and thank the members of the ACORN research project for their support in the design of this research, as well as the staff from the Bath Research Computing Team for their help in the implementation of the FE analysis scripts on the Nimbus cloud supercomputer.

References

[1] P. Jiang, Q. Zhou, and X. Shao, *Surrogate Model-Based Engineering Design and Optimization*, 1st ed. Springer Singapore, 2020. doi: https://doi.org/10.1007/978-981-15-0731-1.

[2] A. Keane, A. Forrester, and A. Sobester, *Engineering Design via Surrogate Modelling: A Practical Guide*. Wiley, 2008. doi: 10.2514/4.479557.

[3] R. Phellan, B. Hachem, J. Clin, J. M. Mac-Thiong, and L. Duong, 'Real-time biomechanics using the finite element method and machine learning: Review and perspective', *Med Phys*, vol. 48, no. 1, pp. 7–18, 2020, doi: 10.1002/mp.14602.

[4] X. Qi, G. Chen, Y. Li, X. Cheng, and C. Li, 'Applying Neural-Network-Based Machine Learning to Additive Manufacturing: Current Applications, Challenges, and Future Perspectives', *Engineering*, vol. 5, no. 4. Elsevier Ltd, pp. 721–729, Aug. 01, 2019. doi: 10.1016/j.eng.2019.04.012.

[5] L. Liang, M. Liu, C. Martin, and W. Sun, 'A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis', *J R Soc Interface*, vol. 15, no. 138, Jan. 2018, doi: 10.1098/rsif.2017.0844.

[6] A. Madani, A. Bakhaty, J. Kim, Y. Mubarak, and M. R. K. Mofrad, 'Bridging Finite Element and Machine Learning Modeling: Stress Prediction of Arterial Walls in Atherosclerosis', *J Biomech Eng*, vol. 141, no. 8, Aug. 2019, doi: 10.1115/1.4043290.

[7] O. J. Pellicer-Valero, M. J. Rupérez, S. Martínez-Sanchis, and J. D. Martín-Guerrero, 'Real-time biomechanical modeling of the liver using Machine Learning models trained on Finite Element Method simulations', *Expert Syst Appl*, vol. 143, p. 113083, Apr. 2020, doi: 10.1016/j.eswa.2019.113083.

[8] H. T. Mai, J. Kang, and J. Lee, 'A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior', *Finite Elements in Analysis and Design*, vol. 196, p. 103572, Nov. 2021, doi: 10.1016/J.FINEL.2021.103572.

[9] S. H. Kim, X. Song, C. Cho, and C. H. Lee, 'Strength prediction of steel CHS X-joints via leveraging finite element method and machine learning solutions', *J Constr Steel Res*, vol. 176, p. 106394, 2021, doi: 10.1016/j.jcsr.2020.106394.

[10] Y. Chen, M. C. Goorden, F. J. Beekman, M. Rezasefat, and J. D. Hogan, 'A finite element-convolutional neural network model (FE-CNN) for stress field analysis around arbitrary inclusions', *Mach Learn Sci Technol*, vol. 4, no. 4, p. 045052, Dec. 2023, doi: 10.1088/2632-2153/AD134A.

[11] W. Gao, X. Lu, Y. Peng, and L. Wu, 'A Deep Learning Approach Replacing the Finite Difference Method for in Situ Stress Prediction', *IEEE Access*, vol. 8, pp. 44063–44074, 2020, doi: 10.1109/ACCESS.2020.2977880.

[12] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, 'Learning Mesh-Based Simulation With Graph Networks', 2021. [Online]. Available: https://sites.google.com/view/meshgraphnets

[13] M. Raissi, P. Perdikaris, and G. E. Karniadakis, 'Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations', *J Comput Phys*, vol. 378, pp. 686–707, Feb. 2019, doi: 10.1016/J.JCP.2018.10.045.

[14] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, 'A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics', *Comput Methods Appl Mech Eng*, vol. 379, p. 113741, Jun. 2021, doi: 10.1016/J.CMA.2021.113741.

[15] H. Jiang, Z. Nie, R. Yeo, A. B. Farimani, and L. B. Kara, 'StressGAN: A generative deep learning model for two-dimensional stress distribution prediction', *Journal of Applied Mechanics, Transactions ASME*, vol. 88, no. 5, May 2021, doi: 10.1115/1.4049805/1096635.

[16] M. S. Nashed, J. Renno, and M. S. Mohamed, 'Nonlinear analysis of shell structures using image processing and machine learning', *Advances in Engineering Software*, vol. 176, p. 103392, Feb. 2023, doi: 10.1016/J.ADVENGSOFT.2022.103392.

[17] R. Xin, V. T. Le, and N. S. Goo, 'Prediction of the buckling mode of cylindrical composite shells with imperfections using FEM-based deep learning approach', *Advanced Composite Materials*, Mar. 2024, doi: 10.1080/09243046.2023.2224129.

[18] Z. ul R. Tahir, P. Mandal, M. T. Adil, and F. Naz, 'Application of artificial neural network to predict buckling load of thin cylindrical shells under axial compression', *Eng Struct*, vol. 248, p. 113221, Dec. 2021, doi: 10.1016/J.ENGSTRUCT.2021.113221.

[19] R. Oval, M. Nuh, E. Costa, O. A. Madyan, J. Orr, and P. Shepherd, 'A prototype low-carbon segmented concrete shell building floor system', *Structures*, vol. 49, pp. 124–138, Mar. 2023, doi: 10.1016/J.ISTRUC.2023.01.063.

[20] BSI, 'UK National Annex to Eurocode 1: Actions on structures — Part 1-1: General actions'. British Standards Institution, 2008.

[21] BSI, *BS EN 1990:2002 +A1:2005: Eurocode - Basis of structural design*. British Standards Institution, 2005. doi: 10.1007/978-3-642-41714-6_51753.

[22] M. H. Kalos and P. A. Whitlock, 'Monte Carlo Methods: Second Edition', *Monte Carlo Methods: Second Edition*, pp. 1–203, Jul. 2009, doi: 10.1002/9783527626212.

[23] K. U. Bletzinger and E. Ramm, 'A General Finite Element Approach to the form Finding of Tensile Structures by the Updated Reference Strategy':, *International Journal of Space Structures*, vol. 14, no. 2, pp. 131–145, Nov. 1999, doi: 10.1260/0266351991494759.

[24] A. Kaszynski, 'pyansys: Pythonic interface to MAPDL'. Zenodo, 2021.

[25] ANSYS Inc., 'ANSYS Mechanical APDL Element Reference', 2012.

[26] F. Song, Z. Guo, and D. Mei, 'Feature selection using principal component analysis', *Proceedings - 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization, ICSEM 2010*, vol. 1, pp. 27–30, 2010, doi: 10.1109/ICSEM.2010.14.

[27] K. O'Shea and R. Nash, 'An Introduction to Convolutional Neural Networks', *Int J Res Appl Sci Eng Technol*, vol. 10, no. 12, pp. 943–947, Nov. 2015, doi: 10.22214/ijraset.2022.47789.

[28] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[29] J. Bergstra and Y. Bengio, 'Random Search for Hyper-Parameter Optimization', *Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012, Accessed: Mar. 17, 2022. [Online]. Available: http://scikit-learn.sourceforge.net.

[30] W. Zhu, W. Yeh, J. Chen, D. Chen, A. Li, and Y. Lin, 'Evolutionary convolutional neural networks using ABC', *ACM International Conference Proceeding Series*, vol. Part F148150, pp. 156–162, 2019, doi: 10.1145/3318299.3318301.

[31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning Data Mining, Inference, and Prediction*, 2nd ed. Springer, New York, NY, 2017. doi: https://doi.org/10.1007/978-0-387-84858-7.

[32] D. P. Kingma and J. L. Ba, 'Adam: A Method for Stochastic Optimization', *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec. 2014, Accessed: Mar. 29, 2024. [Online]. Available: https://arxiv.org/abs/1412.6980v9

[33] A. M. Saxe, J. L. McClelland, and S. Ganguli, 'Exact solutions to the nonlinear dynamics of learning in deep linear neural networks', *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, Dec. 2013, Accessed: Mar. 29, 2024. [Online]. Available: https://arxiv.org/abs/1312.6120v3