

Structural performance assessment of GFRP elastic gridshells by machine learning interpretability methods

Soheila KOOKALANI^{a*}, Bin CHENG^a, Jose Luis Chavez TORRES^b

^a Department of Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

^b Department of Civil Engineering, Technical University of Loja, Loja 110102, Ecuador

*Corresponding author. E-mail: soheila_kookalani@sjtu.edu.cn

© Higher Education Press 2022

ABSTRACT The prediction of structural performance plays a significant role in damage assessment of glass fiber reinforcement polymer (GFRP) elastic gridshell structures. Machine learning (ML) approaches are implemented in this study, to predict maximum stress and displacement of GFRP elastic gridshell structures. Several ML algorithms, including linear regression (LR), ridge regression (RR), support vector regression (SVR), K-nearest neighbors (KNN), decision tree (DT), random forest (RF), adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), category boosting (CatBoost), and light gradient boosting machine (LightGBM), are implemented in this study. Output features of structural performance considered in this study are the maximum stress as $f_1(x)$ and the maximum displacement to self-weight ratio as $f_2(x)$. A comparative study is conducted and the Catboost model presents the highest prediction accuracy. Finally, interpretable ML approaches, including shapely additive explanations (SHAP), partial dependence plot (PDP), and accumulated local effects (ALE), are applied to explain the predictions. SHAP is employed to describe the importance of each variable to structural performance both locally and globally. The results of sensitivity analysis (SA), feature importance of the CatBoost model and SHAP approach indicate the same parameters as the most significant variables for $f_1(x)$ and $f_2(x)$.

KEYWORDS machine learning, gridshell structure, regression, sensitivity analysis, interpretability methods

1 Introduction

Nowadays, demands for sustainable structures have been increased. Selecting the appropriate materials and systems according to the life cycle assessment and environmental impact is one strategy for constructing an environmentally friendly structure. The gridshell structure is a sustainable and lightweight lattice roof with the ability to cover a large span. Gridshell is constructed by deform a flat grid without in-plane shear rigidity that creates a double-curvature structure. Several studies have been performed in the case of gridshell structure analysis [1,2]. The accurate analysis of gridshell structure in terms of stress and displacement is critical for breakage reduction.

Finite element analysis (FEA) is usually implemented for structural analysis. However, FEA is a complex and time-consuming process. Therefore, a substituted method with a fast and easy computational process is required. In recent years, data-driven techniques have been developed as substitutions to time-consuming simulation processes. Data-driven approaches solve structural engineering problems through a low computational cost and accurate process [3,4].

The main contributor to data-driven methods is machine learning (ML), which is a subset of artificial intelligence. There are several effective ML applications in structural engineering [5–8]. Mangalathu and Jeon [9] established data-driven ML approaches by employing lasso regression for beam-column joints. Yao et al. [10] discovered that a two-class support vector regression (SVR) presents superior estimation accuracy than

one-class SVR and logistic regression in their investigation of mapping landslide susceptibility. Chopra et al. [11] studied the efficiency of ML models, including decision tree (DT), random forest (RF), and neural networks, to estimate the concrete compressive strength. The results of their study indicate that the neural network model had the highest efficiency, followed by the RF method. Das et al. [12] presented a data-driven physics-informed approach for concrete crack estimation. The suggested technique can estimate the expected service life of infrastructure before it has to be maintained using real-time monitoring data.

Mangalathu et al. [13] implemented ML models such as XGBoost, AdaBoost, CatBoost, and LightGBM to determine the mode of seismic failure. Guo et al. [14] compared several ML models, including logistic regression, classical Naive Bayesian classifier, K-nearest neighbors method, some state-of-the-art ensemble methods, support vector machine, multilayer perceptron neural networks, and some tree-based classifiers, for soil liquefaction prediction. Consequently, the ensemble learning methods presented the most reliable results. Huang and Burton [15] applied ML methods to detect the in-plane failure modes of RC frames with infills and observed that support vector machine and adaptive boosting algorithms had reasonable accuracy. Nunez and Nehdi [16] presented a gradient boosting regression tree model for determining the carbonation depth of recycled aggregate concrete with various mineral additives.

This paper performs a comparative study on several ML models, including linear regression (LR), ridge regression (RR), SVR, K-nearest neighbors (KNN), DT, RF, adaptive boosting (AdaBoost), extreme gradient boosting (XGBoost), category boosting (CatBoost), and light gradient boosting machine (LightGBM), for predicting the structural performance of gridshells. Therefore, the model with the highest accuracy is obtained. In this study, a dataset comprising 400 samples is prepared by FEM. The grid search approach and K-fold cross-validation (CV) are applied for finding the optimum parameters of each ML model. The first-order and total-effect SA are performed for different ML approaches. It is crucial to understand why an ML model produces a particular estimation and what features lead to that estimation. Therefore, shapely additive explanations (SHAP), accumulated local effects (ALE), and partial dependence plot (PDP) are required, to comprehend the behaviour of an ML method. The significance of input features is investigated on estimating the structural performance of glass fiber reinforcement polymer (GFRP) gridshell structures by PDP, ALE, and SHAP methods.

The following is how the paper is structured: Section 2 introduces the ML models, including LR, RR, SVR, KNN, DT, RF, AdaBoost, XGBoost, CatBoost, and

LightGBM, followed by performance indexes, hyper parameters tuning, and SA. Then the Interpretable ML approaches, including PDP, ALE, and SHAP, are presented in Section 3. In Section 4, these methods are implemented to two numerical examples for stress prediction and displacement to self-weight ratio prediction. Finally, Section 5 presents the conclusions.

2 Machine learning model development

This study investigates several ML algorithms, including LR, RR, SVR, KNN, DT, RF, AdaBoost, XGBoost, CatBoost, and LightGBM, in order to find the best ML model. Scikit-learn [17], an ML package written in the Python language, is used to create the models. Scikit-learn contains a wide library of ML methods. Moreover, performance indexes, an efficient method for hyper parameters tuning, and SA are presented.

2.1 Machine learning models

2.1.1 Linear regression

LR is a supervised ML approach that identifies a linear relationship between dependent and independent parameters [18]. The simplest technique to determine the dependence of output variables on the input features is LR. The optimum set of coefficients for the parameters is obtained by decreasing the least square error between the actual and predicted value for the coefficients. The output of an LR is a linear combination of the variables as follows:

$$f(X) = \beta_0 + \sum_{i=1}^M X_i \beta_i, \quad (1)$$

where β_0 and β_i are the intercept and the regression coefficient, respectively; M is the number of parameters. The regression coefficients are calculated as below:

$$\beta_{LR} = \arg \min_{\beta} \left[\sum_{j=1}^N (y_j - f(x_j))^2 \right], \quad (2)$$

where N and y_j denote the number of instances and the target output, respectively.

2.1.2 Ridge regression

RR is comparable to the LR except that it reduces the model variance of LR [19]. Most ML approaches have a bias-variance trade-off. RR is a popular approach for shrinking the coefficients by minimizing the sum of squares [20]. Minimizing a penalized cost function calculates the ridge coefficients as below:

$$\beta_{\text{Ridge}} = \arg \min_{\beta} \sum_{j=1}^N \left(y_j - \beta_0 - \sum_{i=1}^M X_{ji} \beta_i \right)^2 + \lambda \sum_{i=1}^M \beta_i^2, \quad (3)$$

where λ determines the coefficient reduction. The RR coefficients approach zero, as λ increases.

2.1.3 Support vector regression

SVR uses hyperplanes with maximum separation to partition the data space based on support vector machines. The output of regression is defined as follows:

$$f(x) = \phi(x)^T \omega + b, \quad (4)$$

where $\phi(x)$ is a function that transfers the input to the high-dimensional space, and b denotes the bias of the model. Minimum values of ω are found by minimizing a convex optimization problem to ensure that the $f(x)$ is flat [21]:

$$\text{Minimise } \frac{1}{2} \|\omega\|^2 + C \sum_{j=1}^N (\xi_j + \xi_j^*), \quad (5)$$

where C is the box constraint; ξ_j and ξ_j^* are slack factors. Consequently, SVM prediction can be expressed as:

$$f(x, a_i, a_i^*) = \sum_{j=1}^N (a_i - a_i^*) K(x_i, x) + b, \quad (6)$$

where α_i and α_i^* represent Lagrangian multipliers; $K(x_i, x)$ denote the kernel function. The presented equations indicate that the kernel function, values of ε , and C can be used to tune the SVM prediction.

2.1.4 K-nearest neighbors

The output variable is predicted by KNN as the mean of multiple surrounding values, where k refers to the number of employed neighbors [22]. The critical concept of KNN is that KNN gives more weight to the K-nearest samples that are closer to new data point x in the training dataset. The conditional probability of x can be calculated by the following formula:

$$P(Y = m | X = x) = \frac{1}{K} \sum_{i \in N_k} I(y_i = m), \quad (7)$$

where $I(y_i = m)$ is a parameter that serves as an indicator; if a given observation belongs to the m th tag, it returns 1; otherwise, it returns 0; N_k is the number of instances.

2.1.5 Decision tree

DT is a supervised ML approach that may be utilized for

the regression model. Based on the training samples, this method improves a predictive model in a tree-like graph [23]. The tree structure encompasses the input parameters of the dataset as the internal nodes, decision rules as branches, and the output as leaf nodes. A regression tree is a set of ML approaches that create estimation models by splitting the feature space into high-dimensional spaces. By dividing the feature space into D areas, R_1, \dots, R_d , the regression problem can be described as:

$$f(x) = \sum_{d=1}^D c_d I(x \in R_d), \quad (8)$$

where c_d denotes the average of the observations.

2.1.6 Random forest

RF is an ML technique that consists of a large number of decision trees. Breiman [24] developed RF, which combines the bagging approach with the selection of random input parameter strategy. The bagging approach creates an individual tree by replacing the training dataset with a random sample (bootstrap sample). This technique controls the overfitting by decreasing the modification implemented to each tree. Furthermore, rather than choosing all of the parameters, a random subset of them is employed. In particular, RF is an enhanced variant of the bagging approach. The final model predicted output is derived by averaging the outcomes of individual decision trees as follows:

$$Y = \frac{1}{B} \sum_{j=1}^B Y_b(X'), \quad (9)$$

where B refers to the number of decision trees; Y_b denotes each decision tree; X' is unknown instances.

2.1.7 Adaptive boosting

The AdaBoost model creates a strong learner from a series of weak learners to enhance the performance of the estimated model [25,26]. Form of the learner is defined as follows:

$$F_T(x) = \sum_{t=1}^T f_t(x). \quad (10)$$

In the AdaBoost model, the sample that was incorrectly predicted is given a higher weight in the preceding step [27,28]. The performance of model is then improved by minimizing the error in the current step. AdaBoost starts with identical weights and evaluates the mean square error of estimation, assigning greater weight to those with the greatest error and repeating the procedure until the output converges, as follows:

$$E_t = \sum_i E[F_{t-1}(x_i) + a_t h(x_i)], \quad (11)$$

where $E(\cdot)$ denotes an error function; $F_{t-1}(x)$ indicates the learner created by the prior training; $f_t(x) = a_t h(x)$ refers to the weak learner, which made a contribution to the strong learner [29]. Finally, AdaBoost combines several weak learners in order to create a single strong learner.

2.1.8 Extreme gradient boosting

XGBoost is a boosting method in which weak learners are iteratively given higher weights. The idea is to integrate the weak learners in order to make a more accurate forecast. Gradient boosting was first presented by Chen et al. [30], who conducted a regression model using the direction gradient vector of the misfit function that is comparable to gradient-descent techniques. Gradient boosting uses a parallel tree version that is known as XGBoost. In this method, the factors are iteratively modified to optimize the objective function by reducing the residual of the past step [31]. In order to avoid overfitting, the XGBoost approach adds regularization to the objective function as well as a loss function. The objective function of XGBoost can be expressed as below:

$$Obj = \sum_{i=1}^n L(\hat{y}_i, y_i) + \sum_{t=1}^k \omega(f_t), \quad (12)$$

where L denotes the loss function for the bias of the model; ω indicates the regular term that is utilized to reduce the complexity of the model.

2.1.9 Category boosting

The CatBoost [32] investigates categorical input parameters as a new gradient boosting technique. This approach employs symmetric decision trees that speeds up the inference process when using pre-trained weak learning models. The CatBoost method is capable of achieving improved performance for highly noisy data with diverse characteristics and complicated relationships. CatBoost sorts all instances at random and then assigns a value to each category characteristic. The priority weight coefficients and priority factor are applied to limit the influence of low-frequency category instances and noise on the distribution of data, which can be expressed as below:

$$\hat{x}_k^i = \frac{\sum_{j=1}^n I_{\{x_j^i = x_k^i\}} \cdot y_j + \beta p}{\sum_{j=1}^n I_{\{x_j^i = x_k^i\}} + \beta}, \quad (13)$$

where p and β represent a prior value and the weight of it, respectively.

2.1.10 Light gradient boosting machine

The LightGBM [33] is on the basis of decision tree approaches. This algorithm generates leaf-wise rather than depth-wise that obtains higher accuracy in more complicated trees. LightGBM implements two innovative methodologies, including Exclusive Feature Bundling and Gradient-based One-Side Sampling, in order to describe its performance and distinguish it from other gradient boosted decision trees. Its basic concept is to linearly integrate M weak regression trees into a strong one, which can be calculated as:

$$F(x) = \sum_{m=1}^M f_m(x), \quad (14)$$

where $f_m(x)$ and $F(x)$ indicate the output of the m th weak regression tree and the final output, respectively. The leaf-wise strategy with depth limitation and the histogram technique are two main enhancements of the LightGBM algorithm.

2.2 Performance indexes

Two performance indexes, namely the root mean square error ($RMSE$), and coefficient of determination (R^2), are employed in this study. Equations for $RMSE$ and R^2 are as below:

$$RMSE = \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} (Y_{pi} - Y_i)^2}, \quad (15)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N_t} (Y_{pi} - Y_i)^2}{\sum_{i=1}^{N_t} (Y_{pi} - \bar{Y}_{pi})^2}, \quad (16)$$

where Y_i and Y_{pi} indicate the real and estimated values of the i th observation, respectively; N_t is the number of testing models; \bar{Y}_{pi} is the average of the output. The lowest value of $RMSE$ and the highest value of R^2 indicate the best-performing ML approach.

2.3 Hyper parameters tuning

Once the dataset is prepared and the ML technique has been chosen, the next step is to specify the model parameters, which are crucial to the success of the model. In this study, the hyper parameters are found using the combination of grid search approach and K -fold CV in

order to avoid overfitting. The potential ranges of each parameter are first established as grids based on literatures. The model is then repeatedly trained using all possible combinations of the parameter grids, and the performance is evaluated using the K -fold CV. The K -fold CV is a method for predictive accuracy measurement that reduces the bias associated with data that is randomly selected for training and testing. It divides the dataset into K equal-sized subsets, and runs a loop of K rounds utilizing $K - 1$ subsets for training and the remaining one for validation. The average of the K rounds is measured to indicate the performance of the model. In this paper, 10-fold CV as a commonly used value is implemented that divides the data into ten groups to prevent overfitting problems.

2.4 Sensitivity analysis

Saltelli et al. [34] presented a variance-based SA method to study how changes in model input values impact model output values. This method determines the interaction between the input features and the output factor by keeping the values of the input parameters constant and adjusting the value of one input parameter [35–37].

2.4.1 First-order sensitivity indices

The measurement vector can be stated as $y = f(x_1, x_2, \dots, x_k)$ in the multivariate k -input model. The first-order index is calculated as follows:

$$S_i = \frac{V_{x_i} [E_{x_{-i}}(y|x_i)]}{V(y)}, \quad (17)$$

where $V_{x_i} [E_{x_{-i}}(y|x_i)]$ calculates the impact of the variable x_i on the output; when x_i is kept constant, $E_{x_{-i}}(y|x_i)$ indicates the variance of the mean value $E(y)$; $V(y)$ refers to the unconditional variance of y .

2.4.2 Total-effect sensitivity indices

Higher order indices of coupling terms have to be extended to assess the whole variance of the output, because the first order index only evaluates a part of variation of the output arising from the variance of the input variable x_i . As a result, the total effect S_T is utilized to evaluate the influence of the input variable x_i to the variance of the output. The total effect index can be measured as below:

$$S_T = 1 - \frac{V_{x_{-i}} [E_{x_i}(y|x_{-i})]}{V(y)}, \quad (18)$$

where $E_{x_i}(y|x_{-i})$ and $V_{x_{-i}} [E_{x_i}(y|x_{-i})]$ indicate the mean value of y when all variables except x_i are kept constant and its variance, respectively. It has to be noted that the

difference between S_i and S_T represents the interaction of the input variables with x_i .

3 Interpretable ML approaches

The developed ML model may be capable of making accurate predictions; however, it is still a black box model that cannot provide an explicit explanation of the mechanical or physical background of the problem, which may result in the ML model losing credit. In this study, the model is interpreted using three approaches, including PDP, ALE, and SHAP, to overcome this obstacle. The significance of a feature is determined by evaluating the increase in estimation error after modifying the factor values. Consequently, a factor is significant if the large error is obtained after this procedure, otherwise, it is not significant if little change has occurred.

3.1 Partial dependence plot

Friedman [38] proposed the PDP to investigate the marginal impact of an assumed parameter on the output by displaying the mean value of outcomes for various parameters. The PDP can be employed to determine the correlation between the feature and the objective. The partial dependence, denoted by f_S , on a portion of features x_S , can be expressed as:

$$f_S(x_S) = E_{x_C} [f(x_S, x_C)] = \int f(x_S, x_C) dP(x_C), \quad (19)$$

where x_S denotes the factors for PDP and x_C refers to the other factors. PDP can be created for data $\{X_i, i = 1, \dots, n\}$, as follows:

$$\bar{f}_S(x_S) = \frac{1}{n} \sum_{i=1}^n f(x_S, x_{iC}). \quad (20)$$

Independence of the input parameters is an essential assumption in PDP. Otherwise, when features are strongly related, they will be influenced by artificial data samples that are impossible in reality, causing the estimated feature effect to be greatly biased.

3.2 Accumulated local effects

ALE is a non-biased variant to PDP that defines the average feature effect of an ML algorithm [39]. If the features are highly associated, $f_S(x_S)$ is averaged over all features in PDP, resulting in biased results. This problem may be solved by constraining data to a particular grid. To block the influence of correlated features, ALE averages the modification in estimations. ALE simplifies the complicated prediction model by relying on only one or two factors. ALE can be explained as:

$$f_{x_i, ALE}(x_S) = \int_{z_{0,1}}^{x_S} E_{x_C|x_S} [\hat{f}^S(x_S, x_C) | x_S = z_S] dz_S - c$$

$$= \int_{z_{0,1}}^{x_S} \int_{x_C} \hat{f}^S(z_S, x_C) P(x_C | z_S) dx_C dz_S - c, \quad (21)$$

where c represents a fixed value; $\hat{f}^S(x_S, x_C) = \frac{\delta \hat{f}(x_S, x_C)}{\delta x_S}$ denotes the local influences of x_S on $\hat{f}(\cdot)$ at (x_1, x_S) ; $z_{0,1}$ refers to a selected value smaller than the least observation; $P(x_C | x_S)$ refers to the density. The center of ALE graphs is zero that works well for correlated data and aids visualization. Nevertheless, grid or interval selection can affect the charts and hide data variability.

3.3 Shapely additive explanations

Lundberg and Lee [40] proposed the SHAP technique for evaluating model prediction, which is based on conditional expectation and game theory. SHAP is utilized to evaluate the impact of distinct input features on each output. In general, SHAP aids in the ranking of the features that account for interaction efforts. To construct an interpretable model, SHAP investigates an additive feature attribution technique. The output model, which is described as a linear function, is the total of the actual values related to each parameter. The interpretable framework can be defined as below:

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^M \phi_i x'_i, \quad (22)$$

where $x = (x_1, x_2, \dots, x_p)$ represents the M input variables; p denotes the quantity of instances; x'_i refers to the simplified inputs. Using a mapping function, $x = hx(x')$, the parameter x' transfers to the x . Moreover, ϕ_0 and ϕ_i denote a fixed value and the influence of each parameter, respectively.

4 Numerical examples

The quality and the quantity of samples affect the

performance of ML approaches. Therefore, preparing a proper dataset is an essential task for this process. To this aim, parametric design method can be implemented, which is a practical method to create a wide range of forms by modifying features and calculating the results. Initially, the values of the features are specified to generate shell structures. The grid is generated on the continuous shell using the compass approach in the second step. This approach generates a regular quadrilateral grid on any shells. The grid is created using a compass in this approach. Initially, two crossing curves are first generated on the shell. After that, a mesh length is specified, which serves as the radius of the compass. The grid spacing is defined from the point of intersection in each axis. The intersection of two arcs determines the knots. This technique is repeated until the entire shell is covered with mesh. Implementing this process leads to generating different shapes by adjusting the values of design variables with variable heights, curvatures, and border forms. In this study, eight input variables, including height (H_1, H_2, H_3), width (D_1, D_2, D_3), length (S), and grid size (G), are considered, as illustrated in Fig. 1(a). The structural shape of the gridshell can be characterize as a function of these variables:

$$X = \{H_1, H_2, H_3, D_1, D_2, D_3, S, G\}. \quad (23)$$

Table 1 presents the ranges of these parameters. 400 samples are generated based on the mentioned method and within the specified ranges of the features. The structural performances of the generated samples are obtained by the FEA, including the maximum displacement and stress. Based on the derived geometries, FEA of gridshells are built and beam component B32 is used to simulate the members, allowing the axial forces, shear, and bending moments to be precisely calculated [41]. The structural members in the gridshells are defined as circular GFRP tubes with a density of 1850 kg/m^3 and Young's modulus of 26 GPa and a cross-section of 4 mm in wall thickness and 25 mm in outer radius. Furthermore, the swivel scaffold connections of the structures are simulated to join the elements at their intersections,

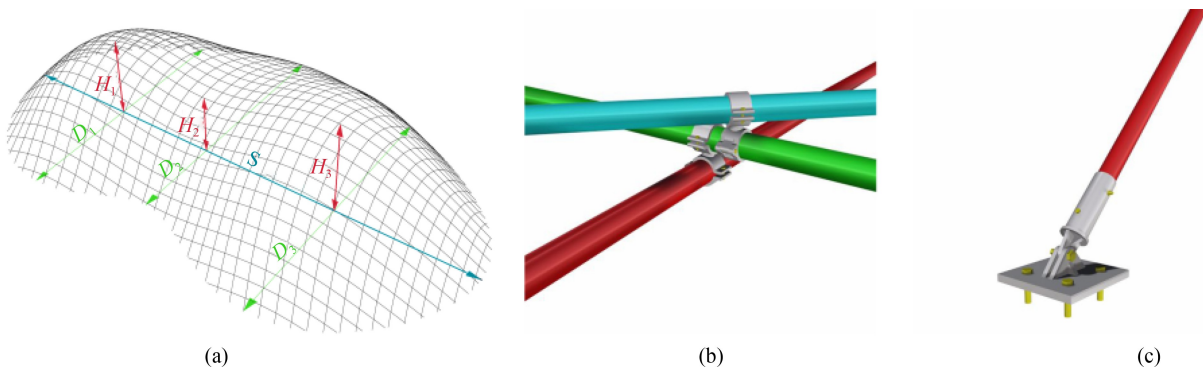


Fig. 1 Gridshell simulation: (a) design variables of gridshells; (b) swivel scaffold connection; (c) pin anchorage for beam ends.

which inhibits out-of-plane rotations and relative translations between vertices while allowing comparative in-plane rotation as illustrated in Fig. 1(b). pinned supports are designed for beam ends to connect the gridshells to the ground as shown in Fig. 1(c) [42]. The weight of equipment is assumed to be 2 kN/m² in the FEA and the structural self-weight is defined by a gravitational acceleration of 9.8 N/kg. The lengths of beam elements are limited to 200 mm [43]. Afterward, two structural performance factors of gridshells, including stress prediction $f_1(x)$ and displacement to self-weight ratio prediction $f_2(x)$, are specified to evaluate the ML models. Finally, a table is established, including the design factors and the outputs. The dataset is randomly split into training and testing set with a 70% training ratio and a 30% testing ratio. The training set is utilized for training ML algorithms and the test set is employed to assess the efficiency of the algorithms.

Figure 2 depicts the correlation matrix for the input parameters. Each correlation coefficient in the matrix represents the strength of interaction between two

Table 1 Statistical attributes of dataset

attribute	unit	minimum	maximum	average
H_1	m	4	8	6
H_2	m	4	8	6
H_3	m	4	8	6
D_1	m	14	18	16
D_2	m	13	22	17.5
D_3	m	16	20	18
S	m	32	37	34.5
G	m	0.5	3	1.75
$f_1(x)$	MPa	3.22	27.82	15.52
$f_2(x)$	mm/kN	0.39	3.94	2.165

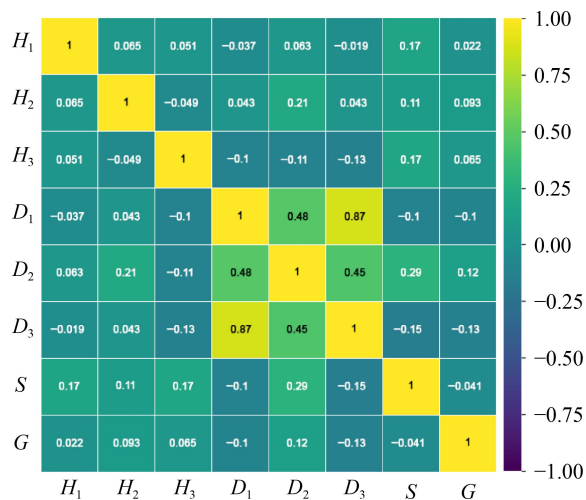


Fig. 2 Correlation matrix for input variables.

parameters. It is shown that the variable D_3 is greatly associated (a correlation of 0.87) with the variable D_1 . Besides, the correlation coefficient between D_1 and D_2 is 0.48. There are no clear correlations for other parameters. Various ML models, such as LR, RR, SVR, KNN, DT, RF, and gradient boosting approaches, namely AdaBoost, XGBoost, CatBoost, LightGBM, have been studied to find the superior ML method.

4.1 Stress prediction

This research aims to consider a characteristic of structural analysis as output. Since the damage occurs in the overstressed members, it is essential to keep the stress in the elements under control. Therefore, the first output is maximum stress. The stress in the gridshell elements can be expressed as follows:

$$\sigma_x = \frac{F_x}{A} \pm \frac{M_y}{W_y} \pm \frac{M_z}{W_z}, \quad (24)$$

$$\tau_y = \frac{F_y}{A}, \quad (25)$$

$$\tau_z = \frac{F_z}{A}, \quad (26)$$

where σ and τ denote the nominal and shear stress, respectively; A is the area of member cross-section; F represents forces; W indicates the bending modulus of sections; M refers to the inner moments of sections. Consequently, the first output can be defined as:

$$f_1(x) = \sigma_{\text{vmax}} = \left(\sqrt{\sigma_x^2 + 3\tau_y^2 + 3\tau_z^2} \right)_{\text{max}}, \quad (27)$$

where σ_{vmax} denotes the maximum stress.

4.1.1 Hyper parameters fine-tuning

Fine-tuning the hyper parameters of the ML models is accomplished by grid search approach and 10-fold CV in order to avoid overfitting. Consequently, the values with the best performance are selected as the hyper parameters of the ML approaches. Table 2 demonstrates the optimum values of hyper parameters for each ML algorithm.

4.1.2 Sensitivity analysis

The first-order (S_i) and total effect (S_{Ti}) sensitivity indices of the input variables are calculated based on the regression models as shown in Table 3 and Fig. 3. It can be seen that the input variables are independent since the first-order and total-effect indices are almost identical for all of the ML methods except KNN and DT models. G is

Table 2 Optimal hyper parameters for $f_1(x)$

model	optimal configuration
LR	N/A
RR	alpha = 1
SVM	kernel = rbf, C = 30, degree = 1, epsilon = 0.1
KNN	leaf_size = 10, n_neighbors = 1, p = 3
DT	max_depth = 8, min_samples_leaf = 1, min_samples_split = 2, random_state = 2
RF	max_depth = 8, max_features = 3, n_estimators = 500
AdaBoost	learning_rate = 0.1, n_estimators = 200, random_state = 0
XGBoost	colsample_bytree = 0.4, learning_rate = 0.08, max_depth = 2, n_estimators = 1000
CatBoost	depth = 6, iterations = 2000, learning_rate = 0.1
LightGBM	colsample_bytree = 0.9, learning_rate = 0.1, max_depth = 4, n_estimators = 500

Table 3 First-order and total effect sensitivity indices for $f_1(x)$

regression model	S	H_1	H_2	H_3	D_1	D_2	D_3	S	G	Σ
LR	S_i	0.0059	0.0023	0.0289	0.0112	0.2408	0.0600	0.0175	0.6302	0.9971
	S_T	0.0064	0.0023	0.0310	0.0112	0.2439	0.0589	0.0170	0.6300	1.0012
RR	S_i	0.0059	0.0023	0.0289	0.0116	0.2468	0.0609	0.0173	0.6232	0.9972
	S_T	0.0063	0.0023	0.0310	0.0116	0.2499	0.0599	0.0168	0.6230	1.0012
SVR	S_i	0.0009	0.0020	0.0602	0.0109	0.4013	0.0903	0.0099	0.4174	0.9932
	S_T	0.0014	0.0022	0.0632	0.0110	0.4056	0.0903	0.0100	0.4189	1.0029
KNN	S_i	0.0144	0.0136	0.0177	0.0360	0.1624	0.0135	0.0405	0.0378	0.3363
	S_T	0.2808	0.2872	0.4072	0.2561	0.7242	0.2060	0.2857	0.3552	2.8028
DT	S_i	0.0080	0.0169	0.0103	0.0004	0.0231	0.0293	0.0048	0.7881	0.8814
	S_T	0.0529	0.1290	0.0144	0.0002	0.0556	0.0714	0.0061	0.9246	1.2546
RF	S_i	0.0004	0.0085	0.0062	0.0098	0.0745	0.0259	0.0014	0.8587	0.9859
	S_T	0.0039	0.0175	0.0040	0.0102	0.0909	0.0370	0.0044	0.8707	1.0389
AdaBoost	S_i	0.0010	0.0162	0.0015	0.0002	0.0307	0.0198	0.0003	0.8706	0.9406
	S_T	0.0014	0.0578	0.0081	0.0021	0.0434	0.0357	0.0009	0.9308	1.0805
XGBoost	S_i	0.0009	0.0255	0.0109	0.0212	0.1151	0.0185	0.0040	0.7835	0.9799
	S_T	0.0115	0.0378	0.0116	0.0250	0.1230	0.0243	0.0100	0.7877	1.0312
CatBoost	S_i	0.0030	0.0044	0.0177	0.0111	0.0777	0.0193	0.0044	0.8281	0.9660
	S_T	0.0096	0.0174	0.0165	0.0218	0.1233	0.0370	0.0171	0.8741	1.1171
LightGBM	S_i	0.0061	0.0157	0.0099	0.0290	0.1056	0.0195	0.0003	0.7407	0.9271
	S_T	0.0262	0.0292	0.0135	0.0415	0.1531	0.0200	0.0116	0.7945	1.0900

the most influential variable and H_1 has a negligible effect on $f_1(x)$. Thus, the low accuracy might be obtained in the prediction of outputs when the influence of G is not considered. D_2 ranked second among all influencing variables. H_1 and S have the minimum influence on $f_1(x)$. It is found that these two variables are insignificant. Therefore, a dataset can be prepared considering the significant variations of G and D_2 for increasing the accuracy and decreasing the number of samples in future studies.

4.1.3 Regression models for stress prediction

The regression plots of different ML models are demonstrated in Fig. 4, which confirm the high accuracy of the CatBoost model. The average R^2 and $RMSE$ values obtained by 10-fold CV are shown in Table 4. The CatBoost model presents the lowest $RMSE$ (1.124) and the highest R^2 (0.930). As a result, the CatBoost is considered in the next subsection for interpretable methods.

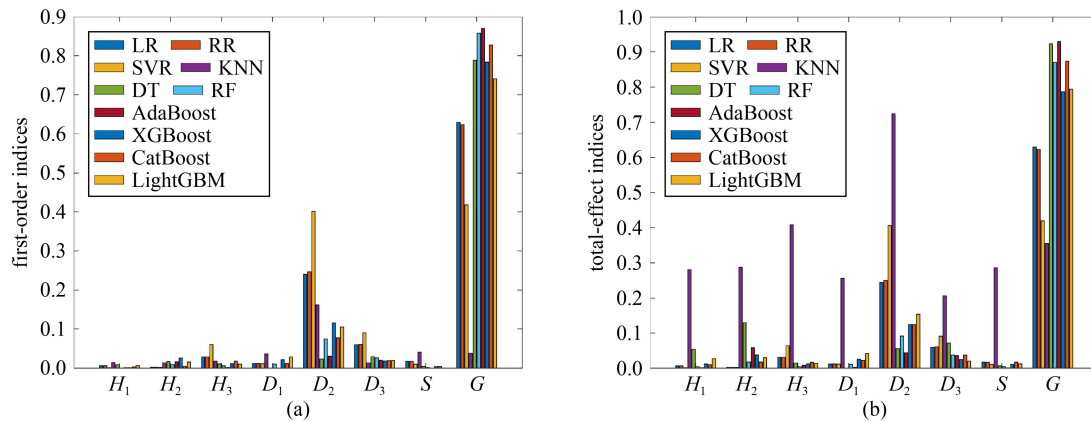


Fig. 3 Sensitivity analysis of $f_1(x)$: (a) first-order; (b) total-effect.

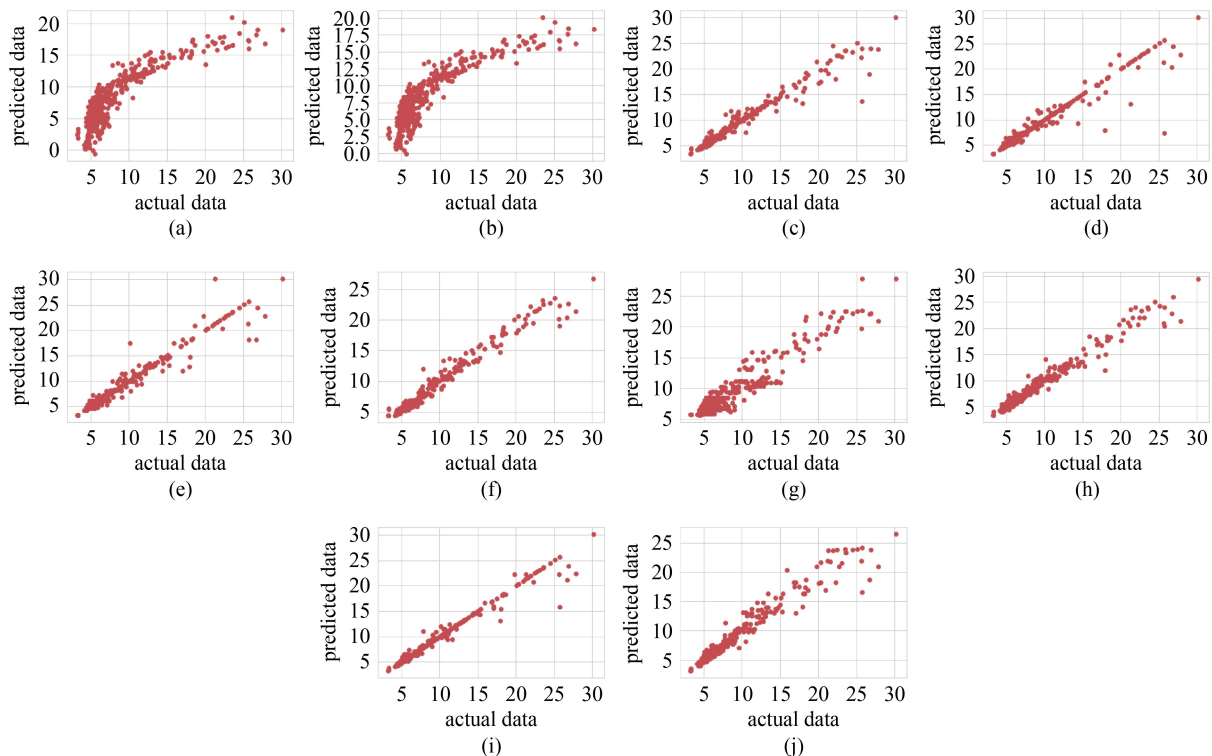


Fig. 4 Regression plot of $f_1(x)$: (a) LR; (b) RR; (c) SVR; (d) KNN; (e) DT; (f) RF; (g) AdaBoost; (h) XGBoost; (i) CatBoost; (j) LightGBM.

Table 4 Performance of ML based regression models for $f_1(x)$ prediction of gridshells

regression method	average R^2	average $RMSE$
LR	0.124	2.515
RR	0.120	2.514
SVR	0.371	2.727
KNN	0.822	1.682
DT	0.779	1.750
RF	0.871	1.342
AdaBoost	0.593	1.824
XGBoost	0.905	1.126
CatBoost	0.930	1.124
LightGBM	0.890	1.278

4.1.4 Interpretable methods for stress prediction

Based on the contribution of each feature to each tree in the model, Fig. 5 displays the importance of the parameters in the development of the CatBoost model. The input parameter G is the most important feature, followed by the parameter D_2 . The least important variable is H_3 , followed by the variable S . However, it is impossible to determine if an input variable has negative or positive effects on the relative importance plots.

The PDPs and ALE plots are the only way to determine this influence, as revealed in Figs. 6 and 7, respectively. Each graph depicts the variation in the CatBoost method

estimation compared to the average of the estimation when the input parameter is changed. Moreover, the

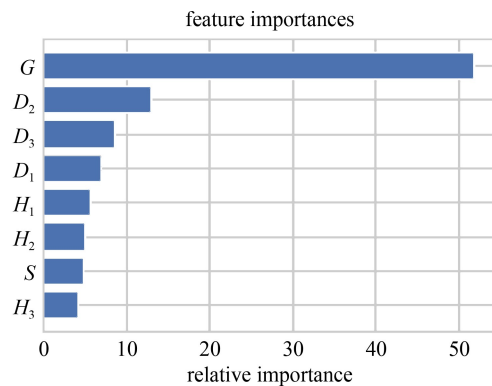


Fig. 5 CatBoost importance factor for $f_1(x)$.

graphs present the threshold of a change in the input variable that causes the model prediction to change.

The PDP aims to display the average model outcome of $f_1(x)$ in terms of distinct values of the estimator throughout the whole range of parameters, whereas, the ALE presents the average outcome over a specific range of parameters. The outcomes prove that the ALE and PDP plots have almost identical trends. The parameter G is crucial and an increase in G , decreases the value of $f_1(x)$, as shown in Figs. 6 and 7. Moreover, the second most important parameter, D_2 , negatively impacts $f_1(x)$ prediction. The $f_1(x)$ value decreases as D_2 increases. Besides, the $f_1(x)$ value is increased when D_3 exceeds the middle value. H_2 does not significantly affect the output until it reaches 75%. The input parameters H_3 and S have the least impact on $f_1(x)$, which is also shown in Fig. 5.

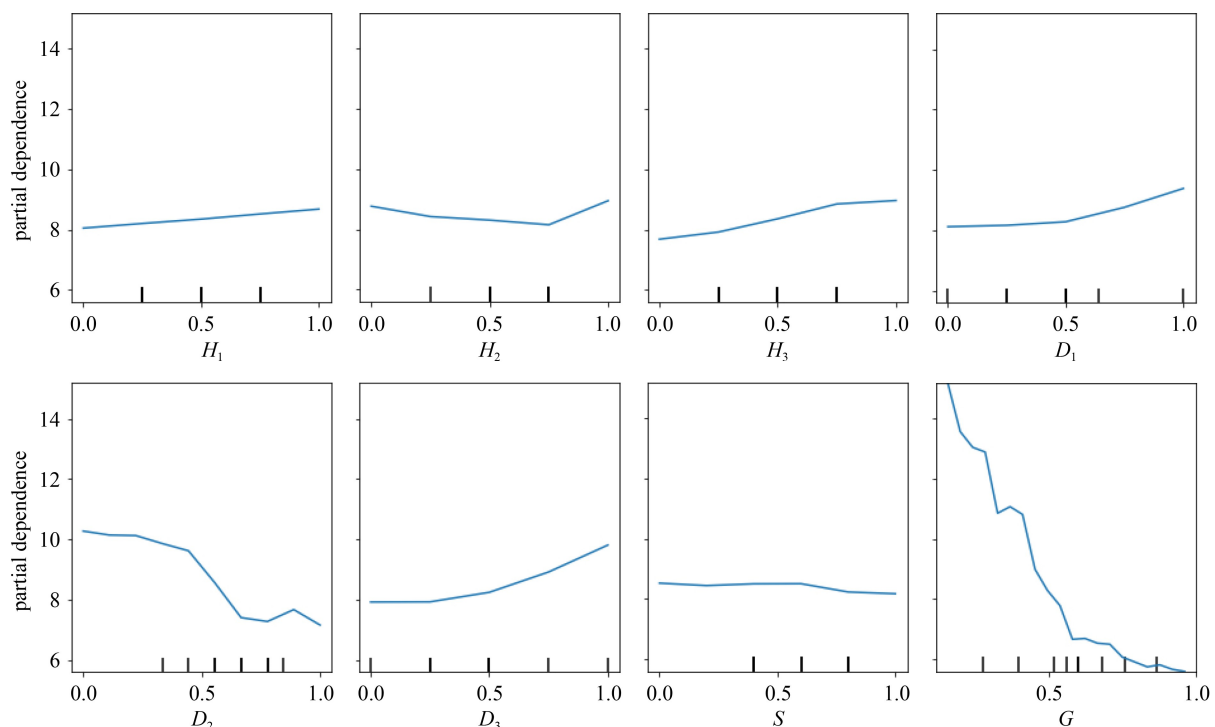


Fig. 6 PDP of input variables for $f_1(x)$.

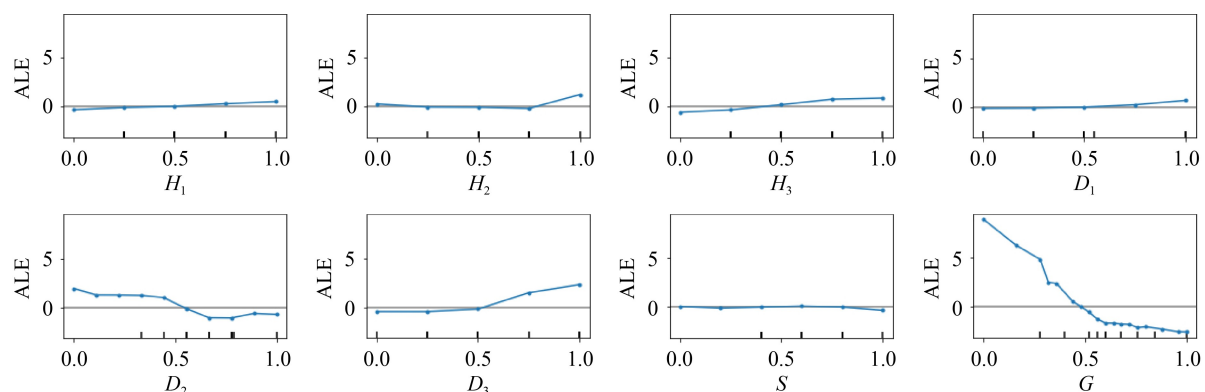


Fig. 7 ALE of input variables for $f_1(x)$.

There is a linear relationship between H_1 and $f_1(x)$. Generally, H_1 , H_2 , H_3 , D_1 , and D_3 have positive effects, while D_2 , S , and G negatively affect $f_1(x)$.

The SHAP summary plot is shown in Fig. 8(a), with each point representing a Shapely value for the parameters. Figure 8(a) shows that each row contains the same quantity of samples. The Shapely values and the input variables constitute the abscissa and ordinate of this plot, respectively. The variables are organized in descending order of importance, while the most important variable is located at the top. Multiple samples of a factor with the identical SHAP value are scattered along the horizontal axes to represent the samples with a certain SHAP value. Red colour denotes a high variable value, whereas blue colour indicates a low variable value. The range of values that raises SHAP value and thus the related estimation is indicated by the red colour.

It can be seen that increasing variables G , D_2 , and S leads to the SHAP value reduction and $f_1(x)$ is also decreased. In contrast, increases in D_3 , H_3 , D_1 , H_2 , and H_1 lead to an increase in the value of $f_1(x)$. Each row has the same number of points in Fig. 8(a). The global significance factor is depicted in Fig. 8(b) as the mean of the absolute of the SHAP value for per factor. SHAP determines that the input parameter G is the most important parameter, comparable to the conclusions from the SA, CatBoost significance variable, ALE, and PDP.

Figure 9 depicts that SHAP presents the dependence graph that is a scatter plot of the SHAP value of a parameter with other parameters. The colour in Fig. 9 reflects the interaction impact of other variables with horizontal axis values. It can be seen that most of them have non-linear behaviour. The effect of D_2 on H_1 is shown in Fig. 9(a). It is evident that for all values of H_1 except 7 m, The SHAP value increases with the increment in D_2 . Besides, the positive effect mostly can be observed in the case of D_1 and D_2 , as shown in Fig. 9(d). Figure 9(g) demonstrates that D_2 almost

negatively affect the SHAP value for input parameter S . The positive effect of D_2 is increased with the increment in G , based on Fig. 9(h). The remaining plots show that for the other variables, the non-linearity is increased. It has to be noted that PDP and ALE plots cannot be used for such inferences.

4.2 Displacement to self-weight ratio prediction

The second output is the maximum displacement to the self-weight ratio of the gridshell. The self-weight of a gridshell can be defined as:

$$W = \sum_{i=1}^k \rho A_i l_i, \quad (28)$$

where l_i refers to the length of the member; A_i denotes the cross-section of the member; ρ represents the density of the material. The displacement can be measured as:

$$d_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \quad (29)$$

where x_i , y_i , z_i are the displacement along the x -, y -, and z -axis. Thus, the second output can be defined as bellow:

$$f_2(x) = \frac{d_{i\max}}{W}, \quad (30)$$

where $d_{i\max}$ is the maximum nodal displacement.

4.2.1 Hyper parameters fine-tuning

The hyper parameters selection of each model has a significant impact on the performance of the model. The optimum hyper parameters are chosen by a combination of grid search method and 10-fold CV to prevent overfitting. Table 5 presents optimum values of each hyper parameters.

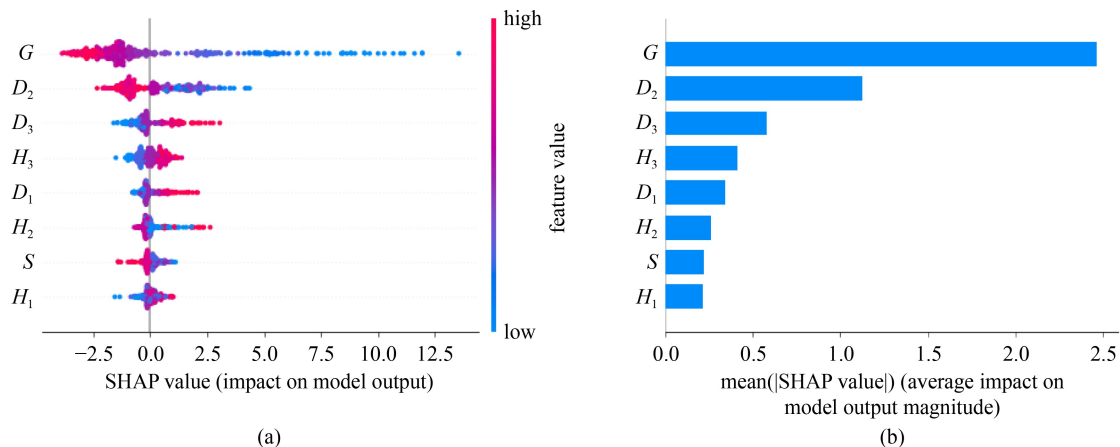


Fig. 8 SHAP plots for $f_1(x)$ of gridshells: (a) Shapely value; (b) global importance factor.

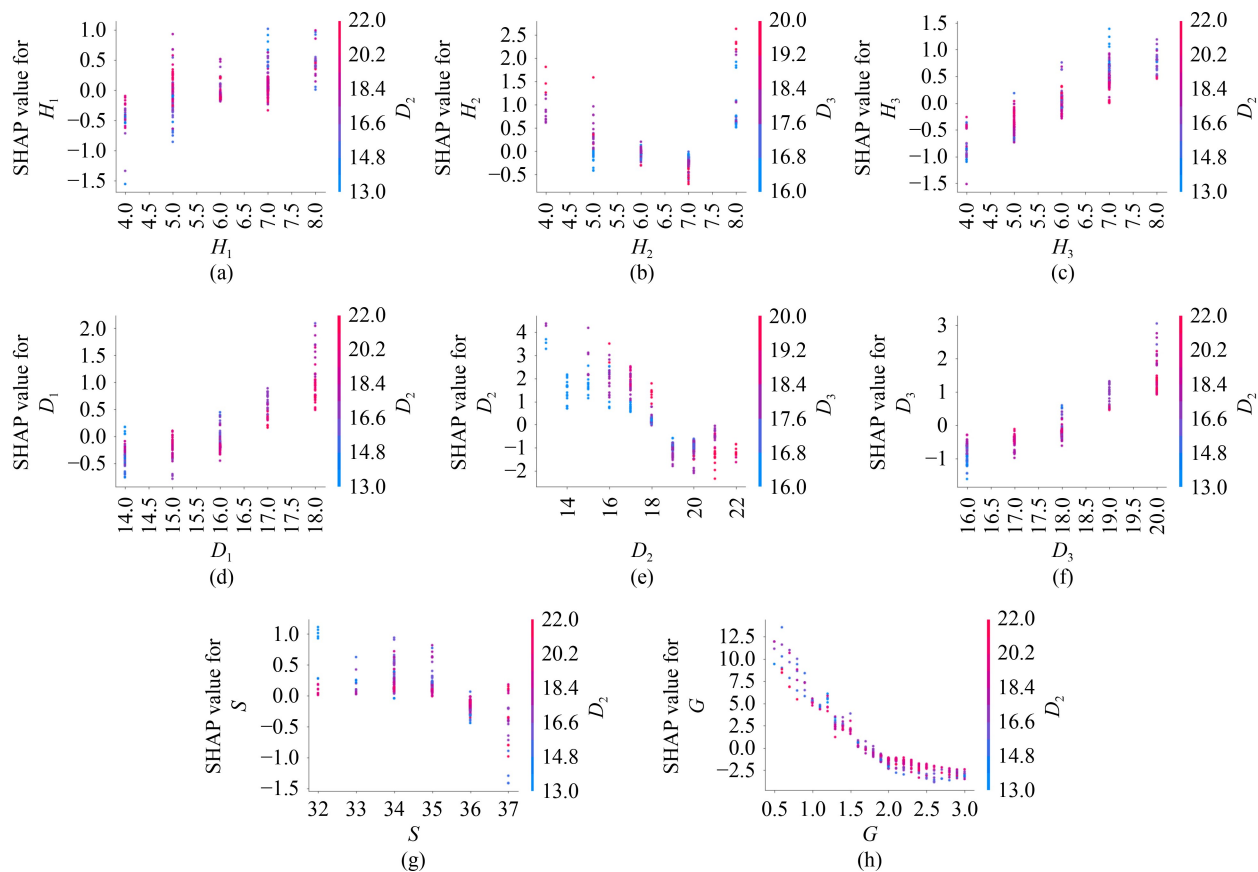


Fig. 9 SHAP partial dependence plots for $f_1(x)$: (a) H_1 ; (b) H_2 ; (c) H_3 ; (d) D_1 ; (e) D_2 ; (f) D_3 ; (g) S ; (h) G .

Table 5 Optimal hyper parameters for $f_2(x)$

model	optimal configuration
LR	N/A
RR	alpha = 100
SVM	kernel = RBF, C = 40, degree = 1, epsilon = 0.01
KNN	leaf_size = 20, n_neighbors = 1, p = 1
DT	max_depth = 8, min_samples_leaf = 1, min_samples_split = 2, random_state = 2
RF	max_depth = 9, max_features = 3, n_estimators = 1000
AdaBoost	n_estimators = 200, random_state = 2, learning_rate = 0.1
XGBoost	colsample_bytree = 0.9, learning_rate = 0.08, max_depth = 4, n_estimators = 1000
CatBoost	depth = 6, iterations = 2000, learning_rate = 0.05
LightGBM	colsample_bytree = 0.9, learning_rate = 0.5, max_depth = 5, n_estimators = 1000

4.2.2 Sensitivity analysis

The impact of influencing variables on $f_2(x)$ is investigated by performing the SA, as shown in Table 6 and Fig. 10. The total-effect indices are higher than first-order indices of KNN and DT models; thus, the variables are correlated in these two models. It is shown that the D_2 is the most sensitive variable for all ML models. Thus, the low accuracy might be obtained in the prediction of outputs when the influence of this variable is not considered. H_1 has the minimum influence, implying that

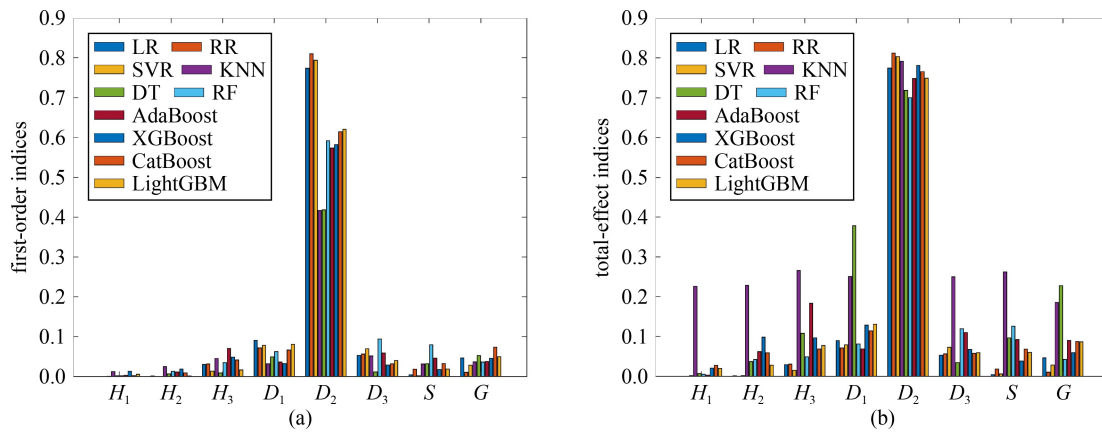
this parameter have relatively little impact on $f_2(x)$. There are not significant differences in the effect of remaining design parameters. Therefore, a dataset can be prepared considering the significant variations of D_2 for increasing the accuracy and decreasing the number of samples in future studies.

4.2.3 Regression models for displacement to self-weight ratio prediction

Figure 11 demonstrates the regression plots of ML

Table 6 First-order and total effect sensitivity indices for $f_2(x)$

regression model	S	H_1	H_2	H_3	D_1	D_2	D_3	S	G	Σ
LR	S_i	0.0000	0.0014	0.0301	0.0903	0.7737	0.0529	0.0043	0.0467	0.9997
	S_T	0.0001	0.0013	0.0292	0.0899	0.7743	0.0529	0.0044	0.0466	0.9992
RR	S_i	0.0000	0.0003	0.0313	0.0718	0.8100	0.0563	0.0177	0.0102	0.9979
	S_T	0.0001	0.0003	0.0307	0.0715	0.8122	0.0563	0.0179	0.0101	0.9994
SVR	S_i	0.0008	0.0002	0.0134	0.0780	0.7937	0.0695	0.0017	0.0286	0.9862
	S_T	0.0020	0.0018	0.0148	0.0789	0.8030	0.0732	0.0060	0.0283	1.0084
KNN	S_i	0.0122	0.0251	0.0453	0.0317	0.4166	0.0517	0.0313	0.0366	0.6508
	S_T	0.2264	0.2290	0.2656	0.2516	0.7916	0.2504	0.2624	0.1860	2.4634
DT	S_i	0.0020	0.0065	0.0089	0.0496	0.4186	0.0112	0.0319	0.0523	0.5813
	S_T	0.0070	0.0367	0.1076	0.3780	0.7181	0.0342	0.0964	0.2282	1.6066
RF	S_i	0.0020	0.0128	0.0347	0.0620	0.5918	0.0942	0.0793	0.0361	0.9133
	S_T	0.0054	0.0418	0.0491	0.0812	0.7003	0.1193	0.1257	0.0422	1.1653
AdaBoost	S_i	0.0032	0.0111	0.0704	0.0358	0.5738	0.0589	0.0462	0.0372	0.8369
	S_T	0.0031	0.0617	0.1842	0.0688	0.7483	0.1094	0.0924	0.0904	1.3587
XGBoost	S_i	0.0129	0.0185	0.0488	0.0322	0.5824	0.0290	0.0169	0.0457	0.7868
	S_T	0.0203	0.0982	0.0965	0.1283	0.7807	0.0677	0.0382	0.0590	1.2893
CatBoost	S_i	0.0025	0.0089	0.0412	0.0660	0.6145	0.0318	0.0323	0.0733	0.8708
	S_T	0.0274	0.0588	0.0687	0.1142	0.7656	0.0574	0.0682	0.0873	1.2480
LightGBM	S_i	0.0057	0.0015	0.0164	0.0806	0.6205	0.0399	0.0181	0.0500	0.8331
	S_T	0.0197	0.0279	0.0774	0.1301	0.7491	0.0592	0.0600	0.0862	1.2099

**Fig. 10** Sensitivity analysis of $f_2(x)$: (a) first-order; (b) total-effect.

models for $f_2(x)$ that confirms the high efficiency of the CatBoost model. Table 7 presents the average $RMSE$ and R^2 obtained by 10-fold CV for assessing the performance of the ML algorithms. The outcomes indicate that the CatBoost method is the best model with the lowest $RMSE$ (0.120) and highest R^2 (0.966). As a result, the CatBoost approach is chosen in this paper to investigate the influence of various interpretable ML algorithms.

4.2.4 Interpretable methods for displacement to self-weight ratio prediction

Figure 12 displays the importance factor generated from

the CatBoost model, with D_2 , D_1 , and D_3 being the most significant variables impacting $f_2(x)$. However, $f_2(x)$ is slightly influenced by the input variables H_1 , G , and H_2 .

Figures 13 and 14 illustrate the direction of the impact of the parameter and the threshold where the estimation variations can be investigated with PDP and ALE graphs, respectively. Even for correlated behaviour, The ALE and PDP plots reveal comparable tendencies. It can be seen that the input factors, with the exception of D_1 , D_2 , and G , have a minor impact on $f_2(x)$. The relationship between $f_2(x)$ and D_3 is non-linear, and there is a rapid change in the $f_2(x)$ value when D_3 surpasses the mean value. The

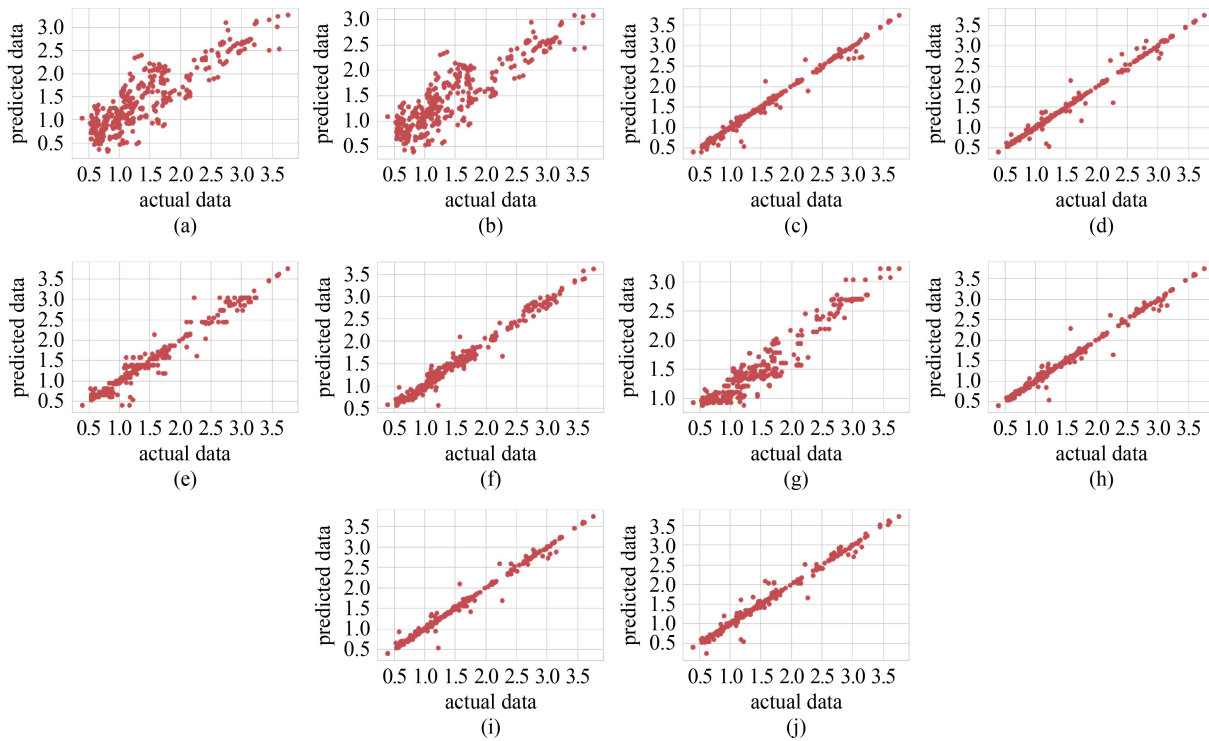


Fig. 11 Regression plot of $f_2(x)$: (a) LR; (b) RR; (c) SVR; (d) KNN; (e) DT; (f) RF; (g) AdaBoost; (h) XGBoost; (i) CatBoost; (j) LightGBM.

Table 7 Performance of ML based regression models for $f_2(x)$ prediction of gridshells

regression method	average R^2	average RMSE
LR	0.649	0.372
RR	0.612	0.396
SVR	0.719	0.332
KNN	0.948	0.140
DT	0.869	0.220
RF	0.941	0.159
AdaBoost	0.788	0.290
XGBoost	0.961	0.129
CatBoost	0.966	0.120
LightGBM	0.931	0.164

$f_2(x)$ is almost identical when H_3 exceeds 75%. The $f_2(x)$ decreases linearly when the S value surpasses the middle value based on the PDP. Generally, H_1 , H_3 , D_1 , D_3 , and G have positive effects, while D_2 , and S negatively affect $f_2(x)$. Although, Fig. 12 ranks the features, the impact of the features can only be determined by the ALE and PDP plots. However, ranking the features on the basis of the ALE or PDP plots is extremely challenging.

Figure 15(a) depicts the overall SHAP values. Red colour represents positive effect, while blue colour specifies negative effect. The positive effect refers to an increase in prediction as the input factor is increased. D_2 has the greatest negative effect in forecasting $f_2(x)$, while

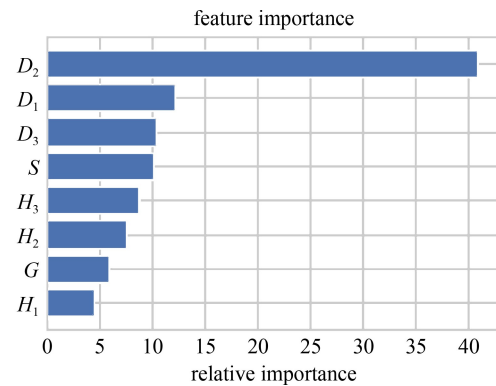


Fig. 12 CatBoost importance factor for $f_2(x)$.

D_1 has the greatest positive impact in forecasting $f_2(x)$, as shown in Fig. 15(a). Generally, D_2 , S , and H_2 negatively predict $f_2(x)$, while D_1 , D_3 , H_3 , G , and H_1 positively predict $f_2(x)$. Figure 15(b) depicts the global importance of the features on the basis of the mean SHAP values in forecasting $f_2(x)$. In predicting $f_2(x)$ of gridshell structures, D_2 is the most important factor, followed by D_1 and D_3 , and H_1 is the least important factor. The global significance feature in CatBoost determines H_3 as the fifth most significant feature, while SHAP detects H_2 as the fifth most important feature due to the basic difference in importance factor assessment. SHAP employs the significance of input features in the output estimations, while CatBoost implements the significance of input features in the construction of decision trees.

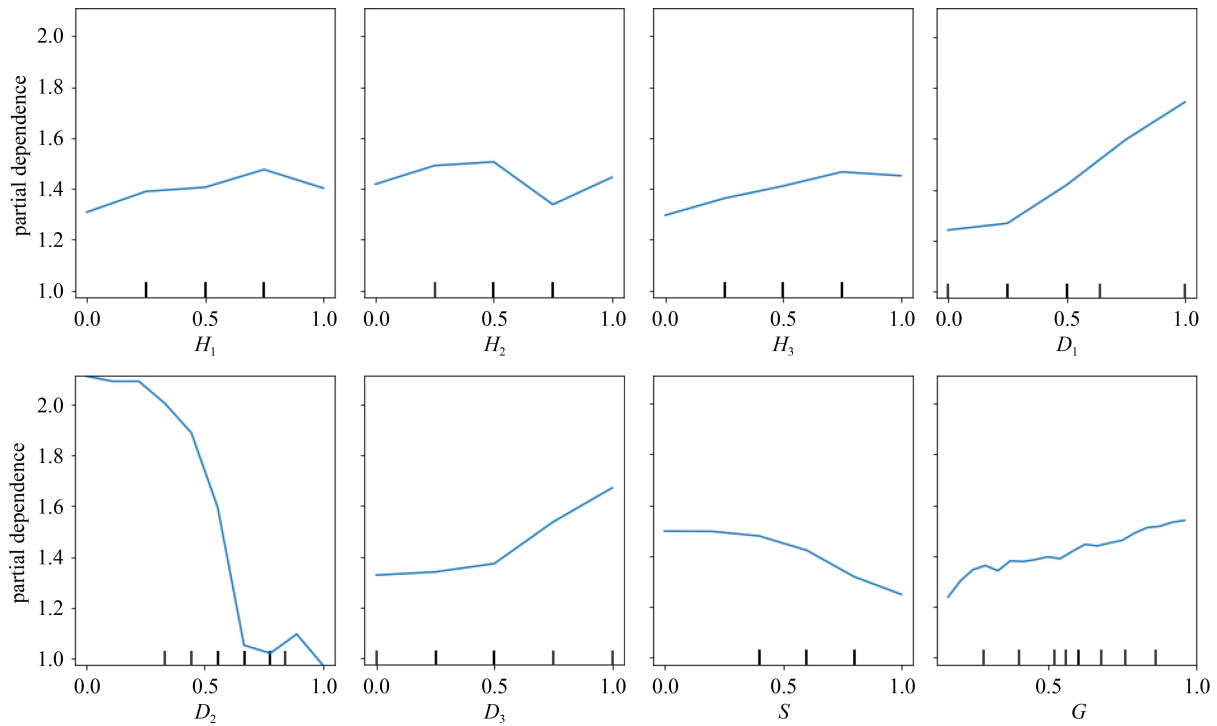


Fig. 13 PDP of input variables for $f_2(x)$.

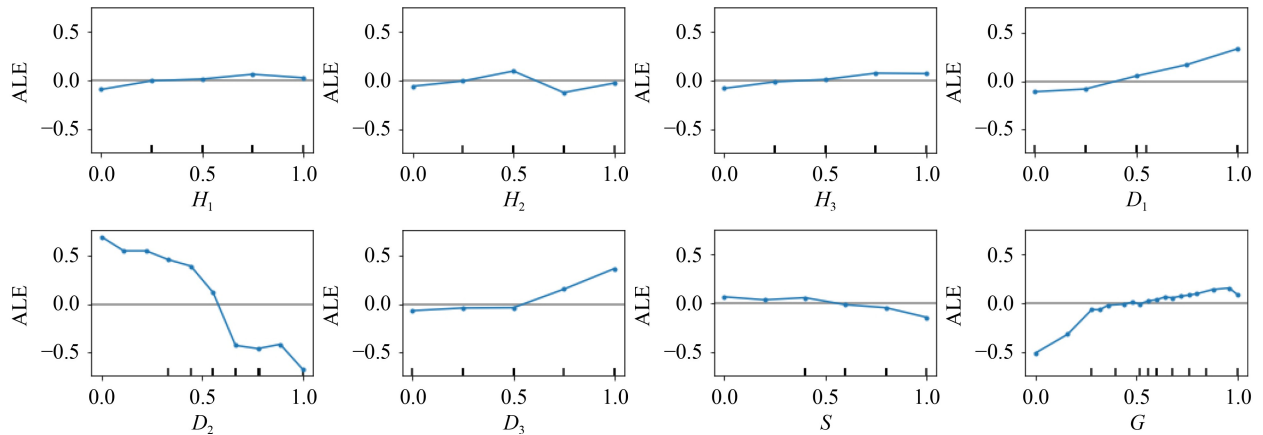


Fig. 14 ALE of input variables for $f_2(x)$.

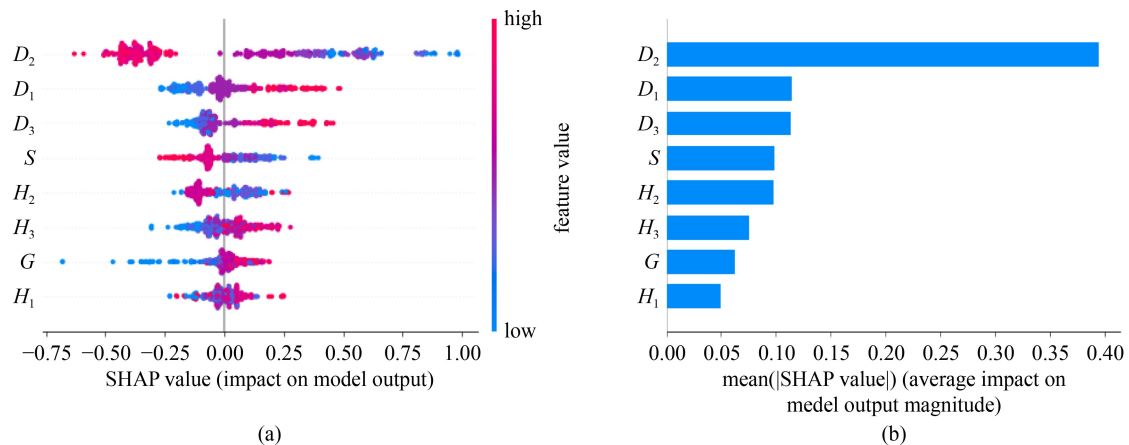


Fig. 15 SHAP plots for $f_2(x)$ of gridshells: (a) Shapely value; (b) global importance factor.

However, the top features selected by SHAP and CatBoost are identical.

The partial dependence plots of SHAP provide extensive insights into the relationships of input parameters, as shown in Fig. 16. It can be seen that the correlations between the features are mostly non-linear. Figure 16(a) displays the effect of SHAP value on H_1 and H_3 variables. It is evident that for H_1 more than 6 m, The SHAP values of H_1 increase with an increment in H_3 . However, the trend is reversed for H_1 less than or equals 6 m. Furthermore, the value of SHAP of H_2 is higher for more D_2 when H_2 is less than or equals 6 m, while the trend is reversed and results in negative SHAP values in the case of H_2 more than 6m, as shown in Fig. 16(b). Figure 16(c) demonstrates that the SHAP value is decreased as the H_3 is increased. The remaining plots show that the other variables frequently interact linearly.

The results indicate that one interpretability approach is not sufficient to entirely explain the behaviour of ML approaches and a collection of interpretable methods is required to comprehensively investigate ML behaviour.

5 Conclusions

This study proposes research on the development of ML

approaches for structural performance prediction of GFRP elastic gridshells subjected to self-weight. To this aim, several ML algorithms, namely LR, RR, SVR, KNN, DT, RF, AdaBoost, XGBoost, CatBoost, and LightGBM, are investigated. FEA of 400 gridshell structures are conducted for dataset preparation to be utilized for the training and testing of the ML algorithms. The input to each ML model consists of eight features, including three height factors, three width factors, length, and grid size. Besides, the output consists of two variables, including the stress and the displacement to self-weight ratio. The best hyper parameters for ML algorithms are determined by a combination of grid search approach and the K -fold CV algorithm. The SA is performed to investigate the influence of input variables on the output and the results are obtained as first-order (S_i) and total-effect (S_T) sensitivity indices. Consequently, little differences are observed between S_i and S_T for most of the ML models, indicating that there is no interaction between input variables. As a result of comparative study on various ML models, the CatBoost model possesses the highest accuracy with R^2 values of 0.930 for $f_1(x)$ and 0.966 for $f_2(x)$ and $RMSE$ values of 1.124 for $f_1(x)$ and 0.120 for $f_2(x)$. LR and RR models shows the lowest accuracy for both outputs.

As a result, the CatBoost is implemented for

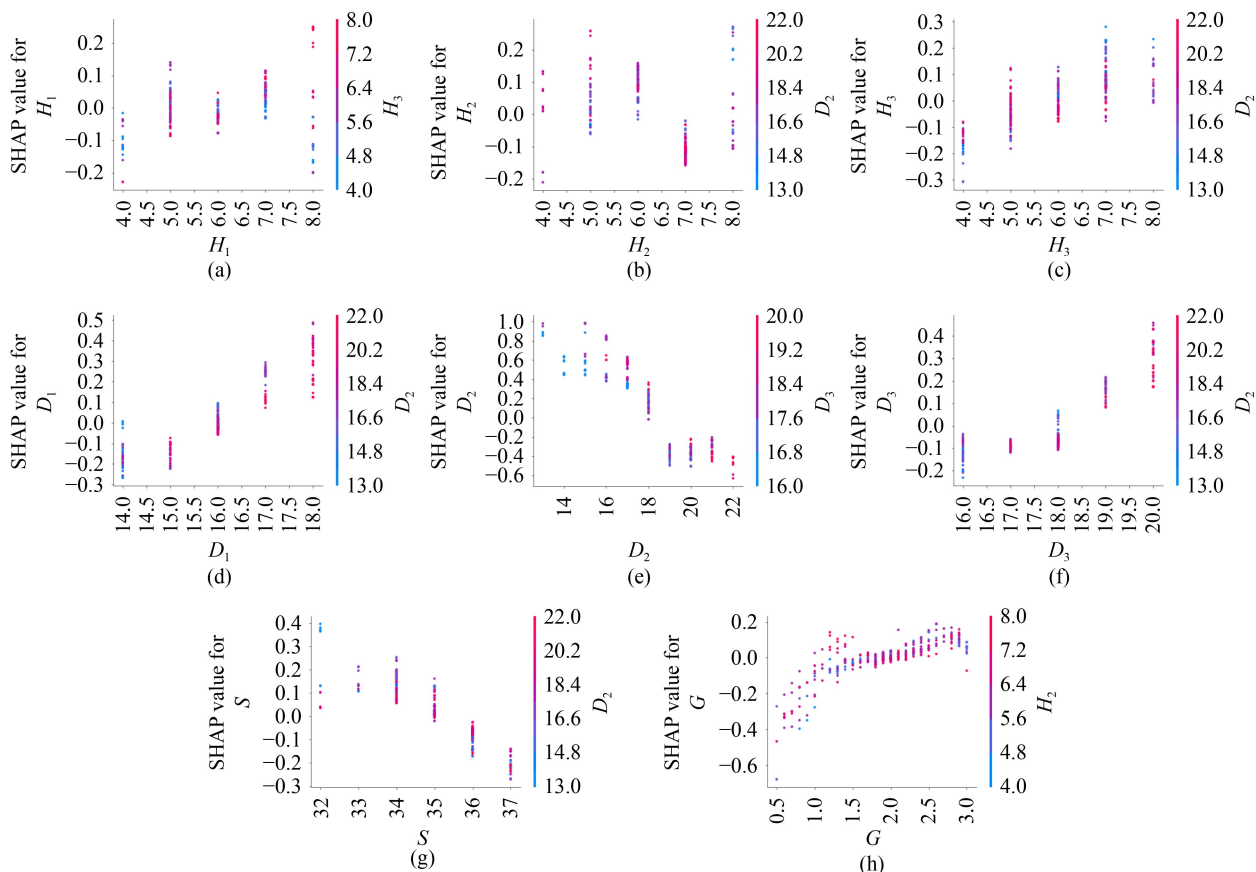


Fig. 16 SHAP partial dependence plots for $f_2(x)$: (a) H_1 ; (b) H_2 ; (c) H_3 ; (d) D_1 ; (e) D_2 ; (f) D_3 ; (g) S ; (h) G .

interpretable approaches. Explaining the prediction of ML models is essential for the proper use of these methods. This paper describes different ML method explanation approaches, including the PDP, ALE, and SHAP. The interpretable ML approaches rank the input features and explain the trend in the estimation. It should be highlighted that a single method is not enough for all interpretability problems; however, SHAP can comprehensively explain prediction models and the importance of input parameters. In general, SHAP fulfils more criteria, which is a reason for its popularity. On the other hand, SHAP is unable to provide an answer to the estimation threshold or how to decrease or increase a specific prediction by input factors modifications. PDP and ALE approaches are able to overcome these drawbacks. However, the ALE approach is an unbiased alternative and faster than PDP. As a result, PDP, ALE, and SHAP approaches are required for comprehensive interpretable ML models. SHAP values reveal that the most significant variables are G and D_2 for $f_1(x)$ and $f_2(x)$, respectively, with negative impacts on estimating the results. The most significant features that are obtained by the SA and the feature importance plots of the CatBoost method, are similar to those obtained by the SHAP method.

Although the conclusions drawn from this study are based on gridshell structures, the method utilized in this paper can be implemented in structural performance estimation for other types of structures. Future research will focus on expanding the proposed methodology for structural performance prediction of gridshell structures considering the complex environmental loads.

Acknowledgements The research work was supported by the National Natural Science Foundation of China (Grant No. 51978400) and the National Key Research and Development Program of China (No. 2021YFE0107800). The support is gratefully acknowledged.

References

1. Tayeb F, Caron J F, Baverel O, Du Peloux L. Stability and robustness of a 300 m² composite gridshell structure. *Construction & Building Materials*, 2013, 49: 926–938
2. Kaveh A, Servati H. Neural networks for the approximate analysis and design of double layer grids. *International Journal of Space Structures*, 2002, 17(1): 77–89
3. Fan W, Chen Y, Li J, Sun Y, Feng J, Hassanin H, Sareh P. Machine learning applied to the design and inspection of reinforced concrete bridges: Resilient methods and emerging applications. *Structures*, 2021, 33: 3954–3963
4. Xu Y, Zhang M, Zheng B. Design of cold-formed stainless steel circular hollow section columns using machine learning methods. *Structures*, 2021, 33: 2755–2770
5. Bekdaş G, Yücel M, Nigdeli S M. Estimation of optimum design of structural systems via machine learning. *Frontiers of Structural and Civil Engineering*, 2021, 15(6): 1–12
6. Sharafati A, Naderpour H, Salih S Q, Onyari E, Yaseen Z M. Simulation of foamed concrete compressive strength prediction using adaptive neuro-fuzzy inference system optimized by nature-inspired algorithms. *Frontiers of Structural and Civil Engineering*, 2021, 15(1): 61–79
7. Teng S, Chen G, Wang S, Zhang J, Sun X. Digital image correlation-based structural state detection through deep learning. *Frontiers of Structural and Civil Engineering*, 2022, 16(1): 1–12
8. Lin S, Zheng H, Han C, Han B, Li W. Evaluation and prediction of slope stability using machine learning approaches. *Frontiers of Structural and Civil Engineering*, 2021, 15(4): 821–833
9. Mangalathu S, Jeon J S. Classification of failure mode and prediction of shear strength for reinforced concrete beam–column joints using machine learning techniques. *Engineering Structures*, 2018, 160: 85–94
10. Yao X, Tham L G, Dai F C. Landslide susceptibility mapping based on Support Vector Machine: A case study on natural slopes of Hong Kong, China. *Geomorphology*, 2008, 101(4): 572–582
11. Chopra P, Sharma R K, Kumar M, Chopra T. Comparison of machine learning techniques for the prediction of compressive strength of concrete. *Advances in Civil Engineering*, 2018, 2018: 1–9
12. Das S, Dutta S, Putcha C, Majumdar S, Adak D. A data-driven physics-informed method for prognosis of infrastructure systems: Theory and application to crack prediction. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems. Part A, Civil Engineering*, 2020, 6(2): 04020013
13. Mangalathu S, Jang H, Hwang S H, Jeon J S. Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls. *Engineering Structures*, 2020, 208: 110331
14. Guo H, Zhuang X, Chen J, Zhu H. Predicting earthquake-induced soil liquefaction based on machine learning classifiers: A comparative multi-dataset study. *International Journal of Computational Methods*, 2022, 2142004
15. Huang H, Burton H V. Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning. *Journal of Building Engineering*, 2019, 25: 100767
16. Nunez I, Nehdi M L. Machine learning prediction of carbonation depth in recycled aggregate concrete incorporating SCMs. *Construction & Building Materials*, 2021, 287: 123027
17. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011, 12: 2825–2830
18. Liang H, Song W. Improved estimation in multiple linear regression models with measurement error and general constraint. *Journal of Multivariate Analysis*, 2009, 100(4): 726–741
19. Hastie T, Tibshirani R, Friedman J, Friedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2009
20. Hoerl A E, Kennard R W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 1970, 12(1): 55–67
21. Smola A J, Schölkopf B. A tutorial on support vector regression. *Statistics and Computing*, 2004, 14(3): 199–222

22. Cover T M, Hart P E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 1967, 13(1): 21–27
23. Dietterich T G. Ensemble methods in machine learning. In: *International Workshop on Multiple Classifier Systems*. Berlin: Springer, 2000, 1–15
24. Breiman L. Random forests. *Machine Learning*, 2001, 45(1): 5–32
25. Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997, 55(1): 119–139
26. Zhang C, Ma Y. *Ensemble Machine Learning: Methods and Applications*. Berlin: Springer Science & Business Media, 2012
27. Schapire R E, Singer Y. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 1999, 37(3): 297–336
28. Schapire R E. Explaining Adaboost. In: *Empirical Inference*. Berlin: Springer, 2013, 37–52
29. Freund Y, Schapire R, Abe N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 1999, 14(771–780): 1612
30. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H. Xgboost: Extreme gradient boosting. *R Package Version 0.4–2*. 2015, 1–4
31. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA: Association for Computing Machinery, 2016, 785–794
32. Dorogush Veronika A, Ershov V, Gulin A. CatBoost: gradient boosting with categorical features support. 2018, arXiv:1810.11363
33. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T Y. Light GBM: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 2017, 30: 1–9
34. Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M, Tarantola S. *Global Sensitivity Analysis*. John Hoboken, NJ: Wiley & Sons, 2008
35. Vu-Bac N, Lahmer T, Keitel H, Zhao J, Zhuang X, Rabczuk T. Stochastic predictions of bulk properties of amorphous polyethylene based on molecular dynamics simulations. *Mechanics of Materials*, 2014, 68: 70–84
36. Vu-Bac N, Zhuang X, Rabczuk T. Uncertainty quantification for mechanical properties of polyethylene based on fully atomistic model. *Materials (Basel)*, 2019, 12(21): 3613
37. Liu B, Vu-Bac N, Zhuang X, Rabczuk T. Stochastic multiscale modeling of heat conductivity of Polymeric clay nanocomposites. *Mechanics of Materials*, 2020, 142: 103280
38. Friedman J H. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 2001, 29(5): 1189–1232
39. Apley D W, Zhu J. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 2020, 82(4): 1059–1086
40. Lundberg S M, Lee S I. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 2017, 30: 1–10
41. Xiang S, Cheng B, Zou L, Kookalani S. An integrated approach of form finding and construction simulation for glass fiber-reinforced polymer elastic gridshells. *Structural Design of Tall and Special Buildings*, 2020, 29(5): e1698
42. Xiang S, Cheng B, Kookalani S, Zhao J. An analytic approach to predict the shape and internal forces of barrel vault elastic gridshells during lifting construction. *Structures*, 2021, 29: 628–637
43. Xiang S, Cheng B, Kookalani S. An analytic solution for form finding of GFRP elastic gridshells during lifting construction. *Composite Structures*, 2020, 244: 112290