

A Machine Learning based uncertainty quantification for compressive strength of high-performance concrete

Nam VU-BAC^{a,*}, Tuan LE-ANH^{a,b}, Timon RABCZUK^c

^a Faculty of Civil Engineering, Ho Chi Minh University of Technology (HCMUT), Ho Chi Minh City 740500, Vietnam

^b Vietnam National University Ho Chi Minh City, Ho Chi Minh City 740500, Vietnam

^c Institute of Structural Mechanics, Bauhaus-University Weimar, Weimar D-99423, Germany

*Corresponding author. E-mail: vbnam@hcmut.edu.vn

© Higher Education Press 2025

ABSTRACT High performance concrete (HPC) properties depend on both its constituent materials and their interaction. This study presents a machine learning framework to quantify the effects of constituents on HPC compressive strength. We first develop a stochastic constitutive model using experimental data and subsequently employ an uncertainty quantification method to identify key parameters in relation to the compressive strength of HPC. The resultant sensitivity indices indicate that fly ash content has the strongest influence on compressive strength, followed by concrete age at test and blast surface slag content.

KEYWORDS uncertainty quantification, machine learning, Artificial Neural Networks, compressive strength of concrete, dependent variables

1 Introduction

High-performance concrete (HPC) has been used widely in concrete construction industry for decades. Additional cementitious materials, i.e., blast furnace slag, fly ash and chemical admixtures (e.g., superplasticizers) are incorporated into the four fundamental ingredients of conventional concrete, namely fine and coarse aggregates, Portland cement and water to make the HPC material [1]. As a consequence, a high strength, high workability, dimensional stability, and durability material has been obtained.

Mechanical properties of the concrete are paramount importance in structural design. They have significant effect on the capacity of structures, the reliability of service-proven structural systems, etc. Therefore, a comprehensive understanding of concrete design methods is necessary to obtain the required properties which ensure structural safety. Particularly, compressive strength is one of the crucial properties of the material and it has been studied for many years. Most concrete design methods are based on experiments or semi-

experiments. Although experimental or semiexperimental results are accurate, they are quite expensive from economic point of view. Besides concrete are also designed based upon empirical methods. However, as the HPC is a mixture of many ingredients with different proportions and the mixing techniques are not indicated properly, a predictive strength from the available data is not completely reliable [2]. For instance, around ten different ingredients are mixed together to design the modern concrete and more and more ingredients tends to be added to enhance material characteristics. Consequently, the adjusted properties of the resulting concrete are increased so that the empirical methods no longer can be used to design the concrete.

The concrete material is stochastic in nature and its properties are affected by different factors (e.g., the above-mentioned constituent parameters). Many experimental studies indicated that concrete strength development is affected by different constituents [3]. Although the experimental data can be used to establish the material development as a function of the constituents, the relationship between the compressive strength and additional cementitious materials (i.e., fly ash, blast furnace, etc.) is still lacked. For the sake of

better designing the concrete mixture, the influence of the concrete composition on the strength should be investigated.

According to Yeh [4], effects of some ingredient parameters on the compressive strength of the HPC are considered. A traditional approach has been performed using experimental data where one constituent parameter is varied while keeping the others fixed. By applying the same technique for all parameters, the effects of constituents on the compressive strength of low-and high-strength concrete can be estimated. However, the approach does not take interactions between constituent parameters into account leading to unsatisfied results. To construct a model of accounting for the HPC strength where the interactions between the constituent parameters are considered, design of experiments has to be employed.

Metamodeling has been used widely to replace complex computational models once experimental data are significantly large. Metamodels adopting experimental training data have been used to search for the optimum proportion mixture design. Statistical models are widely used for predicting the compressive strength tests. For instance, relationship between water-cement proportion and the compressive strengths of concrete was established by Ref. [5] using a regression technique. However, there are several factors influencing the compressive strength and the compressive strength predictions are not accurate if input variable is only the water-cement proportion. Machine Learning (ML) and Artificial Neural Networks (ANN) are regularly employed in constructing meta models. Several methods such as Neural Network (NN), XGBoost, Random Forest (RF), Support Vector Machine, k-Nearest Neighbors etc. have been performed for numerical predictions or regressions [6]. For instance, Mai et al. [7] constructed various ML models such as ANN, RF and Categorical Gradient Boosting (CatBoost) to search for the best predictive model for the compressive strength of fiber reinforced self-compacting concrete. Noureldin et al. [8] proposed a framework for supervised machine learning technique to analyze the seismic performance of frame structures where soil-structure interaction was taken into account. Then, sensitivity analysis was performed to detect the key parameters affecting the soil-structure interaction.

To date, ANNs have been employed extensively to predict the compressive strength of concrete. The ANNs consists of three layers, i.e., the input layer, the hidden layers and the output layer. Each layer includes a couple of neurons and at each of the neurons there is an associated activation function. Due to the presence of the activation function in each layer, ANN is able to learn the complex relationship between the inputs and output [9]. Although multi layers allow learning the nonlinear input-output relationship, it requests tuning several parameters

and hyperparameters, e.g., batch size and the number of epochs [6]. Furthermore, it is burdensome to determine the proper network structure of ANN and to define the activation function, etc.

Ziolkowski and Niedostatkiewicz [10] utilized machine learning for concrete mix design where destructive tests were performed to obtain a large number of databases. Then, the optimal ANN architecture is determined to predict the compressive strength of the concrete resulting from different mix recipes. Lin and Wu [9] built an ANN architecture and the resultant formulae with the corresponding synaptic weights and the thresholds to predict the compressive strength of concrete in accordance with the proportion of concrete mix ingredients. Experimental data are then used to verify and validate their ANN model. Öztaş et al. [11] show applicability of NN in designing the compressive strength and slump of high strength concrete (HSC). Input variables such as the water to binder proportion, water content, fine aggregate proportion, fly ash content and superplasticizer etc. are fed into the NN model to predict the compressive strength and the slump of HSC. Other approaches, which are robust against noisy data such as Gaussian Process, polynomial, Penalized Spline regression models etc. are regularly used.

Polynomial regressions and ANN were employed by Yeh [1,3,4] to predict the compressive strength of the HPC and to assess qualitatively the effects of concrete constituents on this response. Although, good performance of the ANN models was indicated, overfitting could be ignored as the author is only based on coefficients of determination ($COD-R^2$) or root-mean-square error ($RMSE$). It is shown that the proposed approach in this study does not quantify the effects of the mixture constituents on the compressive strength. Furthermore, the constituents are dependent on each other and their interaction effects of the compressive strength have not been tackled so far. Therefore, global sensitivity analysis (SA) can be an appropriate tool for assessment of the meta models.

Global SA based on the Analysis of variances (ANOVA) decomposition (i.e., the functional analysis of variance decomposition) was introduced by Ref. [12]. The total sensitivity measurements were then proposed by Ref. [13]. Among sensitivity analysis methods, variance-based methods have been examined by numerous studies and they have been considered as versatile and effective ones. The variance-based methods known as the computer experiment are replacement tools for the experimental design's analysis. If Monte Carlo simulations are used the methods can thus take the whole factor distribution into account [14].

This study aims to provide a framework consisting of:

- 1) making random samples for dependent variables,
- 2) building effective mathematical metamodels that

approximate the experimental data, and 3) performing SA in a global context. To this end, in Section 2 the experimental data for the HPC is approximated using metamodeling (i.e., penalized spline methodology). Then, Latin hypercube sampling (LHS) described briefly in Section 3 is used to generate random inputs. Transformation of independent input variables to dependent ones is then performed to prepare sample points. Section 4 presents different metamodels. An improvement of variance-based method presented in Section 5 is finally employed to estimate effects of the dependent inputs on the model output quantitatively followed by Section 6 that explain sequential steps of uncertainty quantification procedure. In Section 7 numerical results, i.e., the COD and the sensitivity indices are discussed before we make remarks in conclusion section.

2 Data description

Experimental data (around 1000 mixtures) collected from different sources by Yeh [1,15] were examined in this study. These HPC mixtures, made with ordinary Portland cement, fly ash, blast furnace slag, and superplasticizer were assembled into a data set. The HPC specimens were compacted on a vibrating table, with moist curing maintained for 24 h. After demolding, the concrete was cured in water at temperature of 20 °C (see description in Ref. [4]). According to Ref. [1], specimens with different sizes and shapes were used in different tests. However, based on guidelines for experimental design, all compressive strengths measurements were adjusted to their equivalent values for 150 mm cylinders.

As mentioned above, the HPC is composed of different constituents. Each constituent is represented by a random input variable in the predictive mechanical model, as the input values are of different magnitudes. In this study, the compressive strength of the HPC is considered as a function of eight variables, i.e., 1) cement (kg/m³); 2) fly ash (kg/m³); 3) blast furnace slag (kg/m³); 4) water (kg/m³); 5) superplasticizer (kg/m³); 6) coarse aggregate (kg/m³); 7) fine aggregate (kg/m³); 8) age at test (d). Table 1 summarizes the histograms, corresponding fitted distributions, and ranges of the variables within a stochastic modeling framework.

3 Sampling method

LHS, due to its simplicity and effectiveness, is used to generate quasi-random variables for high-dimensional problems. As the training points provided by this method are scattered entirely throughout the design space, the computational effort is then significantly reduced [16]. The generated independent variables are then transformed

into dependent variables using Gaussian copula and Cholesky factorization. More details of this method can be found in Ref. [17].

4 Metamodels

4.1 Penalized spline regression

Recalling the n -dimensional model function $y = f(\mathbf{x})$ with $\mathbf{x} = [x_1, x_2, \dots, x_n]$ being the vector of n independent variables. Correspondingly, if we define the n -component smoothing vector $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_n]$, at a design point $(\mathbf{x}^{(j)}, y^{(j)})$ the additive model is rewritten as follows:

$$y^{(j)} = \beta_0 + \sum_{i=1}^n f_i(x_i^{(j)}) + \epsilon_j, \quad (1)$$

where $x_i^{(j)}, j = 1, \dots, N$, with N being the number of realizations, denotes the j th realization of the i th variable. Defining Ξ_i knots $\xi_{1i} < \dots < \xi_{\Xi_i i}$ for the i th variable and choosing the degree of the truncated power basis of p . Correspondingly, if we introduce vectors of coefficients $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$ with their respective i th rows $\beta_i = [\beta_{1i}, \dots, \beta_{p_i}]$ and $\zeta_i = [\zeta_{1i}, \dots, \zeta_{\Xi_i i}]$ the mixed mode representation can be written as:

$$f(\mathbf{x}, \boldsymbol{\beta}, \boldsymbol{\zeta}) = \beta_0 + \sum_{i=1}^n \left(\beta_{1i} x_i + \dots + \beta_{p_i} x_i^{p_i} + \sum_{l=1}^{\Xi_i} \zeta_{li} (x_i - \xi_{li})_+^p \right). \quad (2)$$

Let define $\mathbf{X}$ and $\mathbf{Z}$ as linear spline bases with their respective i th rows $\mathbf{X}_i = [1 \ x_i, \dots, x_i^{p_i}]$ and $\mathbf{Z}_i = [(x_i - \xi_{1i})_+^p, \dots, (x_i - \xi_{\Xi_i i})_+^p]$. Equation (2) is rewritten in terms of matrix form as:

$$\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \mathbf{Z} \boldsymbol{\zeta} + \boldsymbol{\epsilon}, \text{Cov}(\boldsymbol{\epsilon}) = \sigma_\epsilon \mathbf{I}. \quad (3)$$

Consequently, the approximated model is shown as:

$$\hat{\mathbf{y}} = \beta_0 + \sum_{i=1}^n \left(\beta_{1i} x_i + \dots + \beta_{p_i} x_i^{p_i} + \sum_{l=1}^{\Xi_i} \zeta_{li} (x_i - \xi_{li})_+^p \right), \quad (4)$$

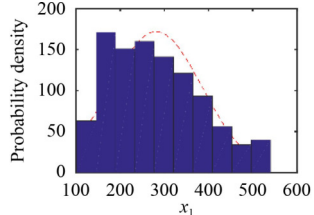
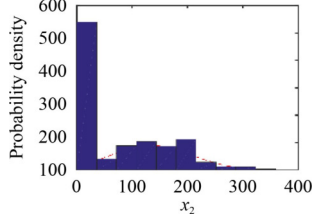
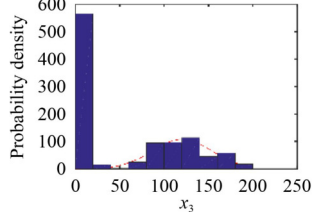
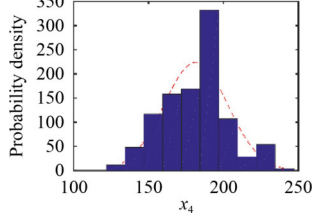
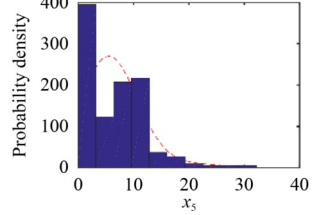
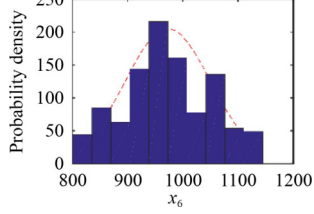
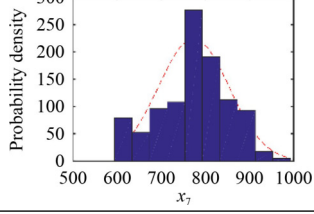
$$\hat{\mathbf{y}} = \mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{Z} \hat{\boldsymbol{\zeta}} = \mathbf{C}(\mathbf{C}^T \mathbf{C} + \boldsymbol{\Lambda})^{-1} \mathbf{C}^T \mathbf{y}, \quad (5)$$

with

$$\mathbf{C} \equiv [\mathbf{X} | \mathbf{Z}] \quad \text{and} \quad \boldsymbol{\Lambda} = \text{diag}(0, 0, 0, \lambda_1^2 \mathbf{1}_{\Xi_1 \times 1}, \dots, \lambda_n^2 \mathbf{1}_{\Xi_n \times 1}). \quad (6)$$

The error between the mechanical model (experimental data) and the approximated regression spline model is then minimized subjected to the constraints on the coefficients $\boldsymbol{\beta}$ and $\boldsymbol{\zeta}$, see Ref. [18] for details. Using Lagrange multiplier method with the penalty function

Table 1 Statistical description of the input variables

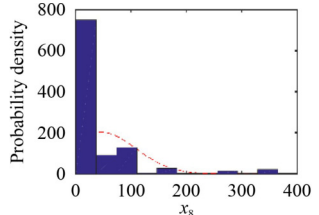
Input	Minimum	Maximum	Truncated distribution	Probability
Cement (x_1)	71	600	Gaussian	
Fly ash (x_2)	0	175	Gaussian mixture	
Blast furnace slag (x_3)	0	395	Gaussian mixture	
Water (x_4)	120	228	Truncated Gaussian	
Superplasticizer (x_5)	0	20.8	Gaussian mixture	
Coarse aggregate (x_6)	730	1322	Gaussian	
Fine aggregate (x_7)	486	968	Gaussian	

$\lambda(\cdot)$ (i.e., smoothing parameter) for the i th predictor, the coefficients $\hat{\beta}$ and $\hat{\zeta}$ are defined as the minimizer of

$$\sum_{j=1}^N \{y^{(j)} - f(x^{(j)}, \beta, \zeta)\}^2 + \sum_{i=1}^n \lambda_i^2(\xi_{l,i}) \zeta_{l,i}^2. \quad (7)$$

Generalized cross-validation (GCV) is minimized to

(Continued)

Input	Minimum	Maximum	Truncated distribution	Probability
Age at test (x_g)	1	365	Gaussian mixture	

obtain the smoothing parameter $\hat{\lambda}_{GCV}$, see Ref. [19] for details.

4.2 Artificial Neural Networks

ANNs are designed to model complex, nonlinear relationships in data. An ANN consists of layers of neurons that transform inputs using weights, biases, and activation functions. For a neuron in layer l , the output is computed as:

$$z_j^{(l)} = \sum_{i=1}^{n^{(l-1)}} w_{ij}^{(l)} a_i^{(l-1)} + b_j^{(l)}, a_j^{(l)} = \phi(z_j^{(l)}), \quad (8)$$

where $w_{ij}^{(l)}$ is the weight connecting neuron i in layer $l-1$ to neuron j in layer l , $b_j^{(l)}$ is the bias term, $\phi(z)$ is the activation function (e.g., ReLU: $\phi(z) = \max(0, z)$).

Learning occurs through backpropagation, an iterative optimization process that minimizes a loss function, such as Mean Squared Error (MSE) in our case:

$$\mathcal{L} = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2, \quad (9)$$

where m is the number of training samples, y_i the actual value, and $\hat{y}_i$ the predicted value. Backpropagation computes gradients of the loss with respect to weights and biases using the chain rule. The weight update is performed as:

$$w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}}, b_j^{(l)} \leftarrow b_j^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial b_j^{(l)}}, \quad (10)$$

where η is the learning rate. Multiple iterations are required to progressively reduce the loss, as the optimization process starts far from the optimal solution.

To prevent overfitting, L2 regularization adds a penalty term $\lambda \sum w^2$ to the loss function, while dropout randomly deactivates neurons during training to enhance generalization. In this study, the ANN architecture included three hidden layers (400-400-300 neurons) with ReLU activations. The model was trained using MSE as the loss function, and early stopping was employed to halt training once validation performance stopped improving.

4.3 XGBoost

XGBoost is a machine learning algorithm based on gradient-boosted decision trees. It builds models iteratively, where each tree corrects the residual errors of the previous model. The prediction at iteration k is given by:

$$\widehat{y}_i^{(k)} = \widehat{y}_i^{(k-1)} + f_k(x_i), \quad (11)$$

where $f_k(x_i)$ is the k th tree, trained to minimize the residual error. The overall objective function is:

$$\mathcal{L} = \sum_{i=1}^m l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k), \quad (12)$$

where $l(y_i, \hat{y}_i)$ is the loss function (e.g., squared error), and $\Omega(f_k) = \gamma T + \frac{1}{2} \lambda |w|^2$ is the regularization term, penalizes large trees with many leaf nodes (T) or high weights $|w|^2$ representing increased model complexity. Here, T represents the number of leaf nodes, and $|w|^2$ is the L2 regularization term for leaf weights. XGBoost incorporates additional optimizations such as parallel processing, sparsity awareness, and tree pruning to enhance computational speed and accuracy. In this study, Bayesian optimization was used to fine-tune hyperparameters like learning rate, tree depth, and regularization terms. This approach ensured efficient modeling of nonlinear interactions among HPC constituents while maintaining strong generalization to test data.

ANNs are adept at capturing high-dimensional, nonlinear relationships through dense neuron connections, while XGBoost handles feature interactions and residual errors efficiently with interpretable tree-based models. Due to these differing strengths, these approaches have proven effective for accurate prediction of HPC compressive strength.

5 Sensitivity analysis

5.1 The ANOVA decomposition

Let consider a mathematical model $y = f(\mathbf{x}) \in \mathbb{L}^2$ with $\mathbf{x} = [x_1, x_2, \dots, x_n]$ being in a unit hypercube domain. We can then expand $f(\mathbf{x})$ as follows:

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^n f_i(x_i) + \sum_{j>i}^n f_{ij}(x_i, x_j) + f_{1\dots n}(x_1, \dots, x_n), \quad (13)$$

with $\mathbf{x}_v = \{x_1, \dots, x_m\}$, $E[f_{i_1, \dots, i_m}(\mathbf{x}_v) | \mathbf{x}_{\sim i_q}] = 0$, $i_q = i_1, \dots, i_m$ and $\mathbf{x}_{\sim i_q} = \mathbf{x}_v \setminus \{x_{i_q}\}$. Note that the functions are orthogonal yielding $E[f_i \times f_j] = V_i \delta_{ij}$ with $\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases}$ see Refs. [14,20].

$$\begin{aligned} f_0 &= E[y], \\ f_i &= E_{x \sim i}[y|x_i] - E[y], \\ f_{ij} &= E_{x \sim ij}[y|x_i, x_j] - f_i - f_j - E[y]. \end{aligned} \quad (14)$$

The higher order terms are written in the same manner.

$$\begin{aligned} V_i &= V[f_i(x_i)] = V_{x_i}[E_{x \sim i}[y|x_i] - E[y]] \\ V_{ij} &= V[f_{ij}(x_i, x_j)] = V_{x_i, x_j}[E_{x \sim ij}[y|x_i, x_j]] \\ &\quad - V_{x_i}[E_{x \sim i}[y|x_i]] - V_{x_j}[E_{x \sim j}[y|x_j]] \end{aligned} \quad (15)$$

As the inputs are independent, Eq. (13) is unique. The total variance of $f(\mathbf{x})$ is then derived by

$$V_y = \sum_{i=1}^n V_i + \sum_{j>i}^n V_{ij} + \dots + V_{1, \dots, n}, \quad (16)$$

where V_i denotes the partial variance of f due to x_i alone and $V_{i_1, \dots, i_m}$ known as the partial variance of due to the interactions between x_i and x_j . If we divide both sides of Eq. (16) by V_y , that yields

$$\sum_{i=1}^n S_i + \sum_{j>i}^n S_{ij} + \dots + S_{1, \dots, n} = 1. \quad (17)$$

The Sobol's sensitivity index is obtained consequently

$$S_{i_1, \dots, i_m} = \frac{V_{i_1, \dots, i_m}}{V_y}, \quad (18)$$

where $S_i = V_i/V_y$ estimates the first-order effect of x_i on y and $S_{i_1, \dots, i_m}$ estimates the high-order effect of joint contribution of $x_1, \dots, x_m$ on y . The total effect of all input variables $\mathbf{x}_{\sim i}$ on y reads

$$ST_i = S_i + \sum_{j \neq i}^n S_{ij} + \dots + S_{1, \dots, n}. \quad (19)$$

It is indicated by Ref. [14], the unicity of the ANOVA decomposition is ensured only if the input variables are independent. In other words, the Sobol's sensitivity index can be affected if the input variables depend on one another. Extended formulas of the variance-based sensitivity indices proposed by Ref. [21] will be presented in the following sections where the main and total effects of the dependent input variables on the model response are considered.

5.2 Formulas for the first-order and the total indices

If we split the input variables $\mathbf{x} = (x_1, \dots, x_n)$ into 2 subsets arbitrarily, i.e., the subset $\mathbf{w} = [x_{i_1}, \dots, x_{i_n}]$ and its complementary one $\mathbf{z} = [x_{i_{n+1}}, \dots, x_{i_n}]$, the total variance of $f(\mathbf{x})$ in Eq. (16) can be rewritten as:

$$V = V_w[E_z f(\mathbf{w}, \hat{\mathbf{z}})] + E_w[V_w(f(\mathbf{w}, \hat{\mathbf{z}}))], \quad (20)$$

where the subset $\mathbf{z}$ is sampled using the joint probability density function (PDF) $p(\mathbf{x}, y)$ while $\hat{\mathbf{z}}$ is sampled using the conditional PDF $p(y, \hat{\mathbf{z}}|y)$. The first-order and the total-effect sensitivity indices for the subsets are described by Ref. [21] as follows:

$$\begin{aligned} S_w &= \frac{V_w[E_z(f(\mathbf{w}, \hat{\mathbf{z}}))]}{V}, \\ ST_z &= \frac{E_z[V_y(f(\hat{\mathbf{w}}, \mathbf{z}))]}{V}, \\ ST_w &= \frac{E_w[V_z(f(\hat{\mathbf{w}}, \mathbf{z}))]}{V}. \end{aligned} \quad (21)$$

The first-order index is estimated numerically by:

$$S_w = \frac{1}{V} \left[\int_{\mathbb{R}^l} p(\mathbf{w}) d\mathbf{w} \left[\int_{\mathbb{R}^{n-1}} f(\mathbf{w}, \hat{\mathbf{z}}) p(\mathbf{w}, \hat{\mathbf{z}}|\mathbf{w}) d\hat{\mathbf{z}} \right]^2 - f_0^2 \right], \quad (22)$$

with $f_0 = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)$ and $p(\mathbf{w})$ denoting a probability density function (PDF). It can be expanded further as follows

$$\begin{aligned} S_w &= \frac{1}{V} \left[\int_{\mathbb{R}^l} p(\mathbf{w}) d\mathbf{w} \left[\int_{\mathbb{R}^{n-1}} f(\mathbf{w}, \hat{\mathbf{z}}) p(\mathbf{w}, \hat{\mathbf{z}}|\mathbf{w}) d\hat{\mathbf{z}} \right. \right. \\ &\quad \cdot \left. \int_{\mathbb{R}^{n-1}} f(\mathbf{w}, \hat{\mathbf{z}}') p(\mathbf{w}, \hat{\mathbf{z}}'|\mathbf{w}) d\hat{\mathbf{z}}' \right] - f_0^2 \right], \end{aligned} \quad (23)$$

where the combined vectors $(\mathbf{w}, \mathbf{z})$, $(\mathbf{w}', \mathbf{z}')$, and $(\mathbf{w}, \mathbf{z})$, $(\mathbf{w}, \mathbf{z}')$ are generated using the joint PDF $p(\mathbf{w}, \mathbf{z})$ along with the conditional PDF $p(\mathbf{w}, \mathbf{z}'|\mathbf{w})$. The first-order indices shown by Eq. (23) can be then obtained by

$$\begin{aligned} S_w &= \frac{1}{V} \left[\int_{\mathbb{R}^l} f(\mathbf{w}, \mathbf{z}) p(\mathbf{w}, \mathbf{z}) d\mathbf{y} d\mathbf{z} \left[\int_{\mathbb{R}^{n-1}} f(\mathbf{w}, \mathbf{z}') p(\mathbf{w}, \mathbf{z}'|\mathbf{w}) d\mathbf{z} \right. \right. \\ &\quad \cdot \left. \int_{\mathbb{R}^{n-1}} f(\mathbf{w}', \mathbf{z}') p(\mathbf{w}, \mathbf{z}') d\mathbf{w}' d\mathbf{z}' \right] \right]. \end{aligned} \quad (24)$$

From both Eqs. (20), (21), and (24), ST_y can be expressed by

$$\begin{aligned} ST_w &= \frac{1}{2V} \int_{\mathbb{R}^{n-1}} [f(\mathbf{w}, \mathbf{z}) - f(\mathbf{w}, \mathbf{z}')]^2 \\ &\quad \cdot f p(\mathbf{w}, \mathbf{z}) p(\mathbf{w}, \mathbf{z}'|\mathbf{w}) d\mathbf{w} d\mathbf{w}' d\mathbf{z}. \end{aligned} \quad (25)$$

For numerical computation, realizations of the variables $\mathbf{z}$, $\mathbf{z}'$, and $\hat{\mathbf{z}}$ have to be generated and the details can be found in Ref. [17].

5.3 Monte Carlo estimates

Monte Carlo (MC) algorithm is used in practice to estimate the first-order and the total-effect indices numerically. Introducing the subsets z' , $\hat{z}'$, and $\hat{w}'$ generated from the conditional PDFs $p(w, z'|y)$, $p(w, \hat{z}'|w)$ and $p(\hat{w}', z|z)$, respectively, the first-order index reads

$$S_w = \frac{\frac{1}{N} \sum_{j=1}^N (f(w_i, z_i) f(w_i, \hat{z}'_i)) - \left(\frac{1}{N} \sum_{j=1}^N f(w_i, z_i) \right)^2}{V}, \quad (26)$$

where the total variance V is approximated numerically by

$$V = \frac{1}{N_{wz}} \sum_{l=1}^{N_{wz}} (f(w_l, z_l))^2 - f_0^2, \quad (27)$$

with

$$f_0 = \frac{1}{N_{wz}} \sum_{l=1}^{N_{wz}} f(w_l, z_l), \quad (28)$$

being evaluated at N_{wz} data points of the combined vector (w, z) . An estimation of the total-effect index yields

$$ST_w = \frac{\frac{1}{N} \sum_{j=1}^N (f(w_j, z_j) - f(\hat{w}'_j, z_j))^2}{2V}. \quad (29)$$

Numerically, the MC estimation for $f(w, z)$, $f(w, \hat{z}')$, $f(\hat{w}', z')$, $f(w', z')$, and $f(w, \hat{z})$ at each data point is estimated using the above-mentioned meta model. It should be noted that n-dimensional multivariate normal distribution and multivariate conditional distribution are thus used to generate dependent variables through MC sampling algorithm. More details of the sampling method and the MC estimates of the sensitivity indices can be found in Ref. [17]. A schematic flow chart of the sensitivity analysis procedure shows the sequential steps linking experimental data preparation, surrogate model construction using ANN/ML, and sensitivity estimation.

6 Flow chart

The pipeline shown in Fig. 1 for predicting HPC compressive strength involves the following steps: 1) data preprocessing, 2) implementation of machine learning models (ANN/XGBoost or Penalized Spline regression), hyperparameter tuning, 3) training, evaluation, and visualization, 4) estimating the sensitivity indices.

7 Numerical results

In this section, the proposed machine learning framework is applied to the experimental data of the HPC described in Section 2 to measure the SA indices. The input variables are dependent and the correlation matrix C between the input variables is shown by:

$$C = \begin{bmatrix} 1 & -0.28 & -0.40 & -0.08 & 0.09 & -0.11 & -0.22 & 0.08 \\ -0.28 & 1 & -0.32 & 0.11 & 0.04 & -0.28 & -0.28 & -0.04 \\ -0.40 & -0.32 & 1 & -0.26 & 0.38 & -0.01 & 0.08 & -0.15 \\ -0.08 & 0.11 & -0.26 & 1 & -0.66 & -0.18 & -0.45 & 0.28 \\ 0.09 & 0.04 & 0.38 & -0.66 & 1 & -0.27 & 0.22 & 0.19 \\ -0.11 & -0.28 & -0.01 & -0.18 & -0.27 & 1 & -0.18 & 0.00 \\ -0.22 & -0.28 & 0.08 & -0.45 & 0.22 & -0.18 & 1 & -0.16 \\ 0.08 & -0.04 & -0.15 & 0.28 & -0.19 & 0.00 & -0.16 & 1 \end{bmatrix}.$$

Note that the components of the correlation matrix are computed using Pearson correlation formula, see Ref. [22] or details. The correlation among the input variables is illustrated in Fig. 2. From the correlation matrix and the plot, we realize that a pair of inputs x_4 and x_5 ($C_{45} = -0.66$) and another one x_4 and x_7 ($C_{47} = -0.45$) are strongly correlated.

Scatter plots of the compressive strength with respect to (w.r.t.) each individual input variable is plotted in Fig. 3. At a glance, we realize that the fly ash (x_2) and the age at test (x_8) may affect the compressive strength of the HPC the most. The effect of the input variables on the

compressive strength will be quantified in the sequel.

In addition to the regression models discussed in the manuscript, we implemented two additional machine learning approaches, an ANN and XGBoost, to enhance prediction accuracy of HPC compressive strength. The ANN consists of three hidden layers with 400, 400, and 300 nodes, respectively, using ReLU activation functions. We included 30% dropout between layers and L2 regularization to prevent overfitting. The XGBoost model's hyperparameters were optimized using Bayesian optimization, resulting in a learning rate of 0.159, maximum tree depth of 4 and 202 estimators, and

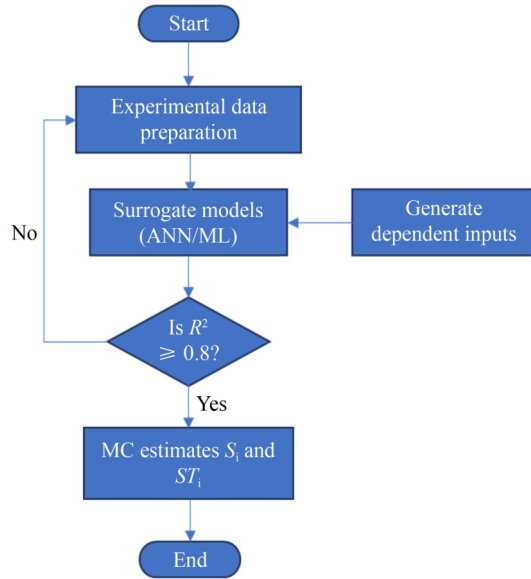


Fig. 1 Schematic flow chart of sensitivity analysis procedure.

regularization parameters $\alpha = 0.59$ and $\lambda = 0.046$, see Table 2 for details. Both models were evaluated by splitting the data into training and validation sets (0.3) to ensure generalizability on out-of-distribution data. XGBoost outperformed the ANN on the validation data set, achieving a lower $RMSE$ and higher R^2 score, demonstrating better generalization capability for predicting concrete strength. These results suggest both models, particularly XGBoost, are effective at accurately mapping how multiple concrete ingredients—particularly the interactions between cement, fly ash, and blast furnace slag—influence the final compressive strength. For more theoretical and implementation details of each model, see Refs. [23,24].

Both models were evaluated on training and testing sets using metrics computed with scikit-learn. Metrics Used: mean_squared_error for MSE and $RMSE$; r2_score for R^2 ; mean_absolute_error for MAE . Tools and Software Used: Data Preprocessing: Pandas (1.5.3), NumPy (1.24.4), and scikit-learn (1.2.2); ANN Implementation: TensorFlow

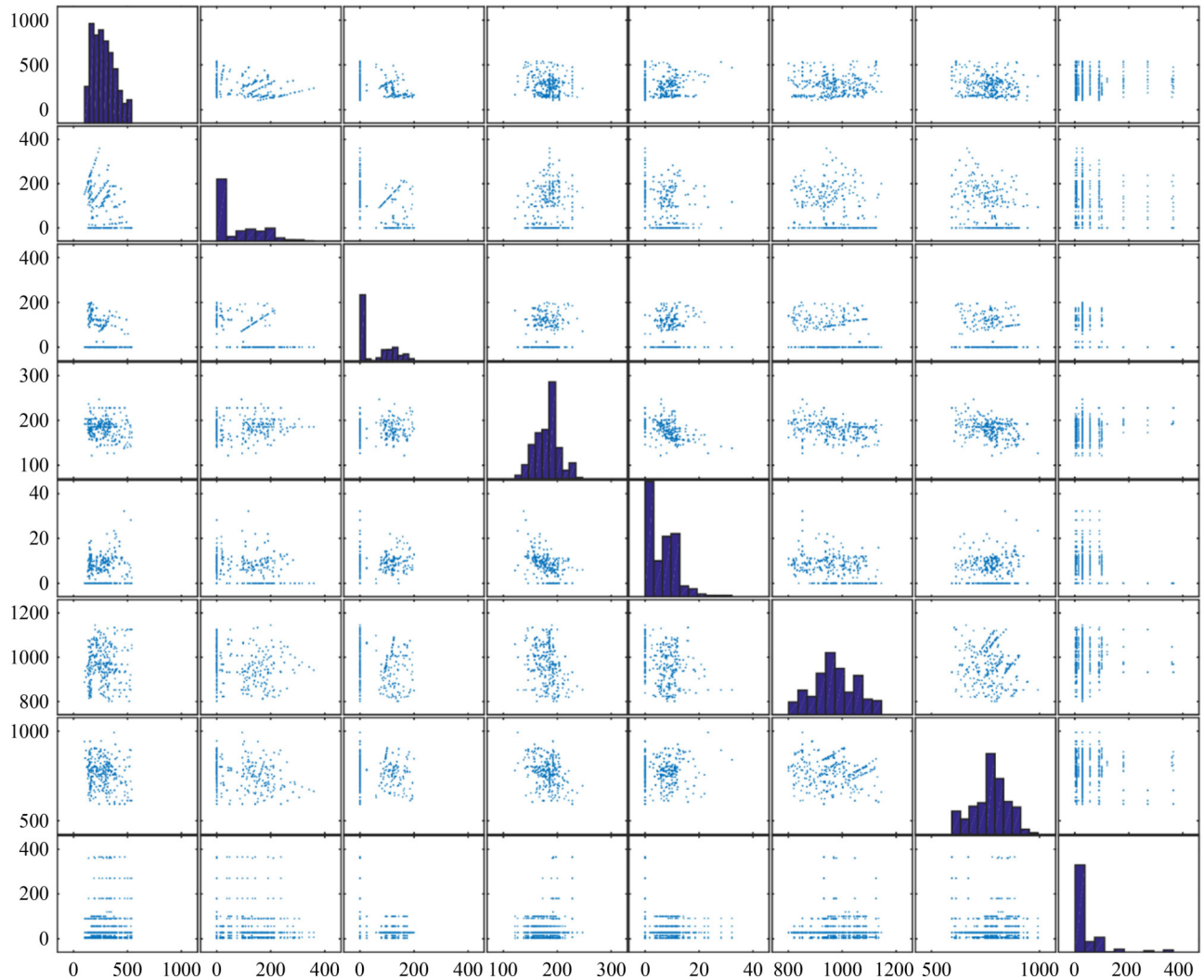


Fig. 2 Plots of input distributions and interaction between the training points.

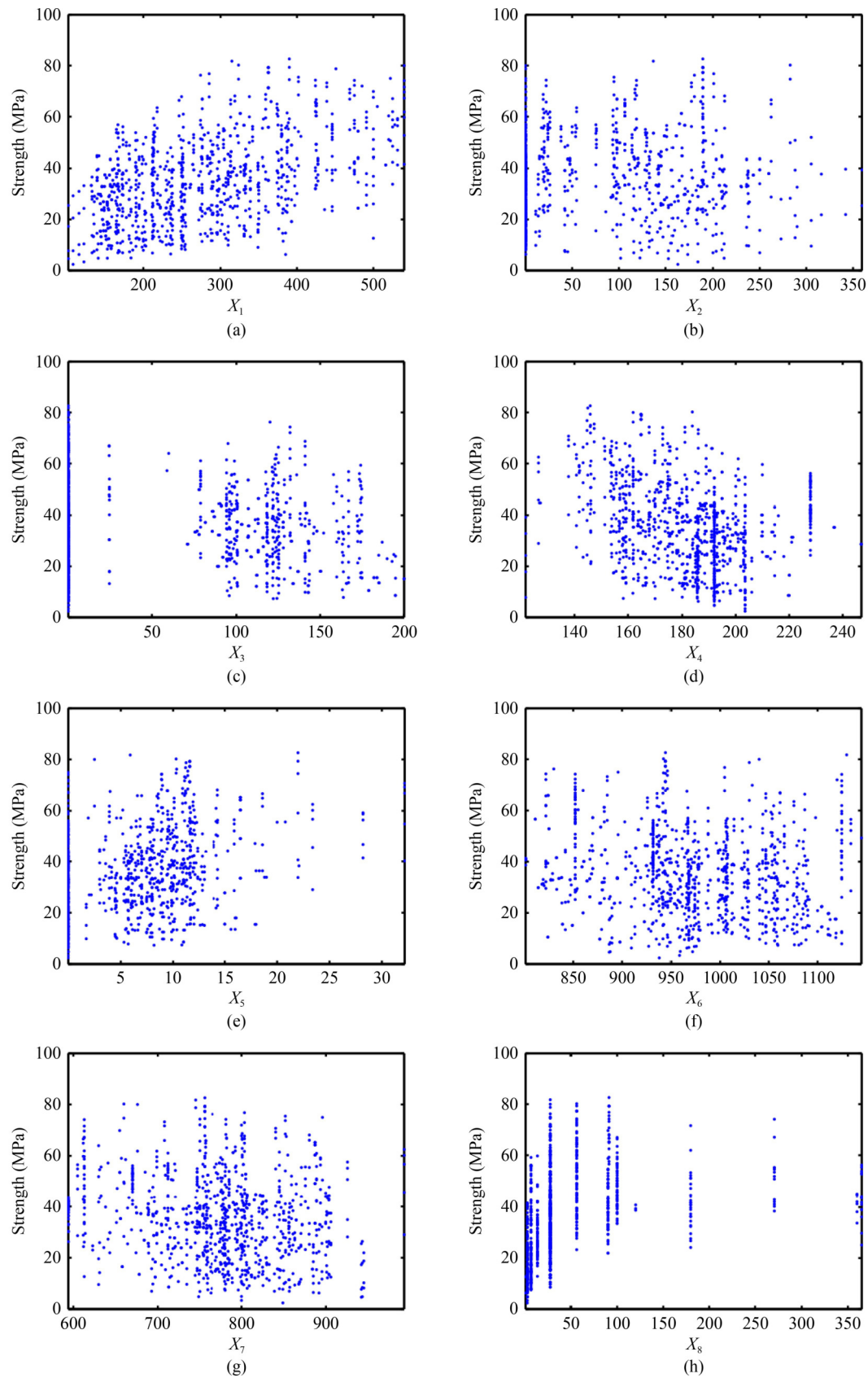


Fig. 3 Scatter points of the compressive strength w.r.t. the input variables: (a) X_1 ; (b) X_2 ; (c) X_3 ; (d) X_4 ; (e) X_5 ; (f) X_6 ; (g) X_7 ; (h) X_8 .

(2.13.0) and Keras (2.13.1); XGBoost Implementation: xgboost (1.7.6); Hyperparameter Tuning: bayes_opt (1.4.2) for Bayesian optimization (XGBoost) and manual tuning with best practices (ANN); Visualization: Matplotlib (3.7.1) and Seaborn (0.12.2) for plotting parity plots, learning curves, and feature importance. History convergences of the loss functions over the number of epoch iterations are shown in Fig. 4. The obtained results are shown in Table 3 as follows.

Table 3 and Figs. 5(a) and 5(b) show that the results obtained by the ANN, particularly $R^2_{\text{ANN}} = 0.95(0.88)$ and $RMSE_{\text{ANN}} = 3.72(5.66)$ are in good agreement with the previous results shown in Ref. [4]: $R^2_{\text{REF}} = 0.94(0.93)$ and $RMSE_{\text{REF}} = 3.01(4.32)$ for training and testing data, respectively. Slightly discrepancies between the results are due to different hyperparameters used in our and their models.

To avoid carrying out a large number of experiments requested by the SA, metamodeling is employed in this study. As reported by Ref. [1], a quadratic regression model fails to approximate nonlinear relationships in the data (i.e., $R^2 < 0.8$). Thus, both quadratic [25,26] and penalized spline regression models, shown in Section 4, are constructed to approximate the experimental data on which the SA will be performed. The approximation

ability of the two methods, indicated by either the COD (R^2) or the $RMSE$ [27], is shown in Table 4. The quadratic regression model, with $R^2_{\text{quad}} < 0.8$, clearly fails to approximate the data, while the penalized spline regression model, with $R^2_{\text{spl}} = 0.92$, fits the data almost perfectly. Therefore, the penalized spline regression model is employed in this study. The training points and the approximated regression model are plotted w.r.t. the two variables x_2 , x_3 and the ones x_2 , x_8 in Figs. 6(a) and 6(b), respectively.

The SA is then performed to measure the sensitivity indices of the input variables on the compressive strength. First, a large number of samples of dependent input variables are generated using the technique presented in Section 3. To this end, random variables following n -dimensional multivariate normal distribution and conditional normal distribution are generated using MC algorithms. Next, the predicted output values at the samples are obtained using the penalized spline regression model. Finally, the MC estimates are used to evaluate the sensitivity indices. The first-order and the total-effect sensitivity indices for the cases of independent inputs and dependent inputs are shown in Fig. 7. The results demonstrate that the sensitivity indices differ when dependencies among the inputs are

Table 2 ANN and XGBoost model structures

Model	Layers/Parameters	Description
ANN	input layer	8 features, fully connected
	hidden layer 1	400 nodes, ReLU activation, L2 regularization
	hidden layer 2	400 nodes, ReLU activation, L2 regularization
	hidden layer 3	300 nodes, ReLU activation, L2 regularization
	dropout	30% dropout after each hidden layer
	output layer	1 node, linear activation
XGBoost	learning rate	0.159 (tuned via Bayesian Optimization)
	max depth	4
	number of estimators	202
	regularization (α , λ)	0.59, 0.046

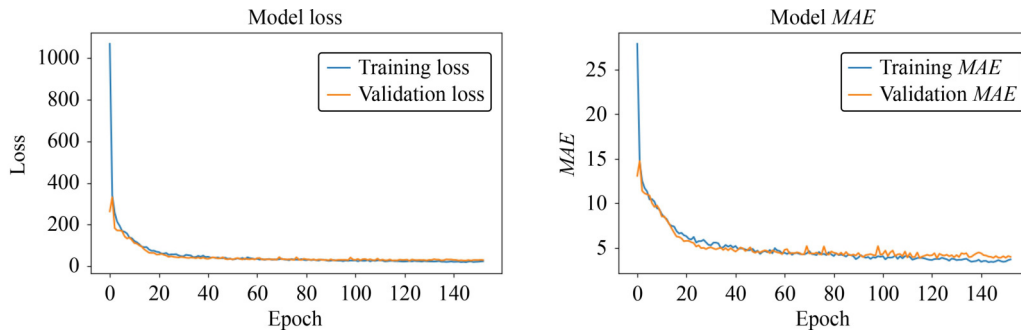


Fig. 4 ANN learning curves.

considered versus when they are ignored. In other words, interactions among the input variables have to be taken into account when measuring their effects on the compressive strength. Note that since the first-order and the total-effect indices here are approximated numerically, they are reduced by the R^2 to account for how well the metamodel explains the experimental data, that is

$$\hat{S}_i = R^2 S_i,$$

$$\widehat{ST}_i = R^2 ST_i.$$

Table 3 Performance metrics of ML models

Model	Dataset	RMSE	R^2
ANN	training	3.72	0.951
	test	5.66	0.882
XGBoost	training	1.64	0.991
	test	4.16	0.933

When measuring the sensitivity index of each individual variable on the compressive strength without considering the correlations (interactions) among them, the most sensitive variables are the coarse aggregate, followed by the age at test and the cement. However, when considering correlations among the input variables, the key influencing variables are fly ash and age at test, as shown in Table 5. The effect of cement on the compressive strength is reduced relatively when considering the dependencies (interactions) among the concrete constituents. This can be explained from practical point of view: when the fly ash is added into the concrete, the amount of the cement in concrete can be reduced [28]. Furthermore, fly ash particles reacting with the cement can produce additional cementitious material, resulting in an increase in long-term compressive strength of the HPC. This suggests that the fly ash plays a more important role for the compressive strength when it interacts the cement matrix.

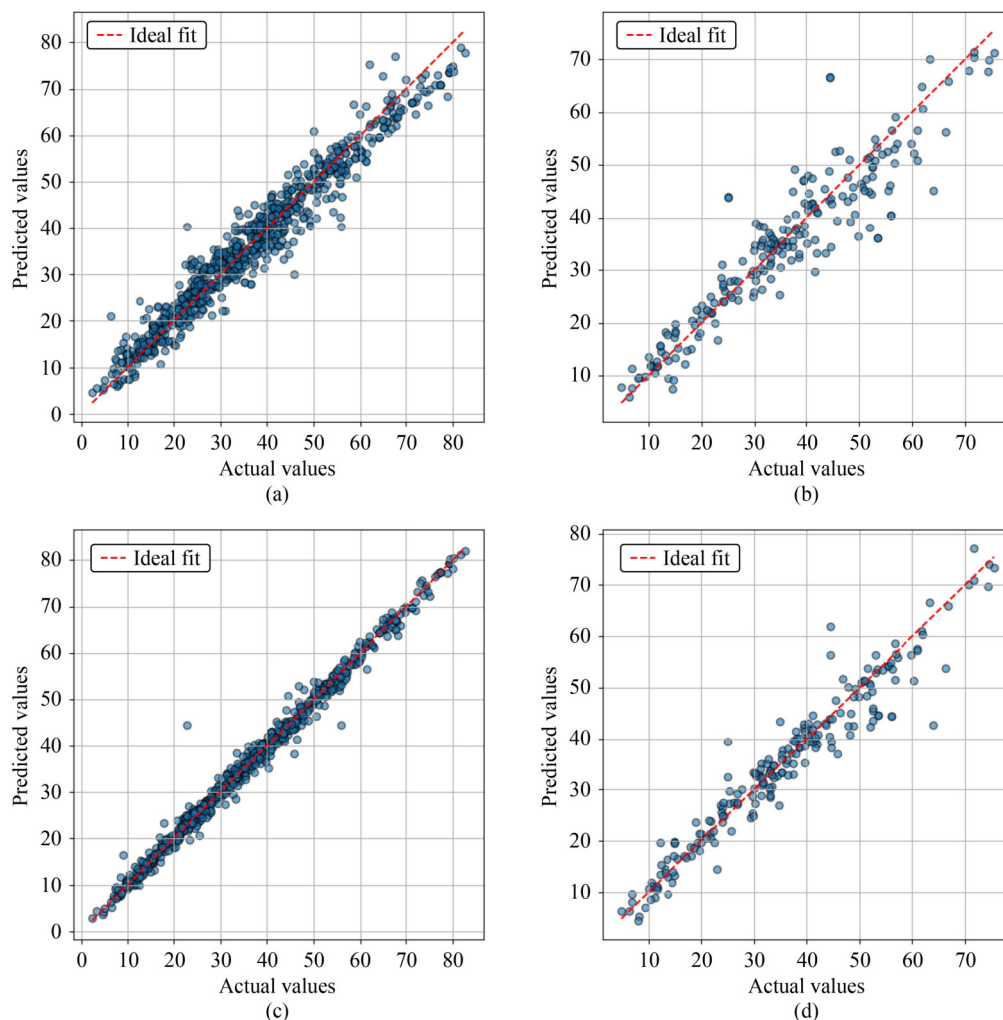


Fig. 5 (a) Parity plots of training data set for ANN; (b) parity plots of test data set for ANN; (c) parity plots of training data set for XGBoost; (d) parity plots of test data set for XGBoost.

Table 4 Comparison of the approximation performance of the quadratic and penalized spline regression models

Quadratic regression model		Penalized spline regression model	
R^2_{quad}	$RMSE$	R^2_{spl}	$RMSE$
0.77	7.93	0.92	4.77

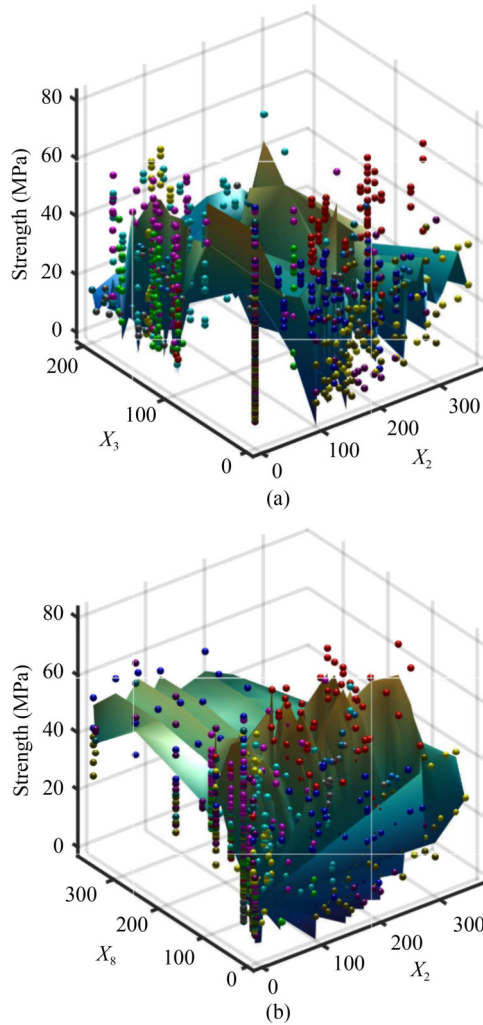


Fig. 6 Plots of the approximated meta models and the training points. The meta models are projected on the two variables (a) x_2 and x_3 ; (b) x_2 and x_8 .

8 Conclusions

A machine learning methodology is presented in this study to quantify the effects of concrete constituents on the compressive strength of the HPC. The dependencies (interactions) between input variables, represented by the correlation matrix, are taken into account. The numerical example indicates that the effect of one input on the compressive strength is contributed not only by itself but also by its interactions with other variables. In addition to sensitivity measurement, a stochastic constitutive law representing the relationship between the compressive

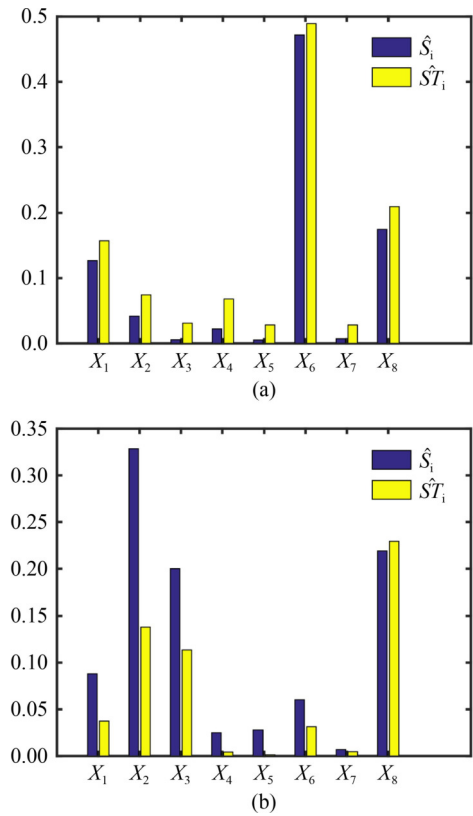


Fig. 7 The first-order and the total-effect sensitivity indices in case: (a) the inputs are considered as independent variables; (b) the inputs are considered as dependent variables.

Table 5 First-order and total-effect sensitivity indices computed on the penalized quadratic spline regression model

SA indices	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
$\hat{S}_i$	0.09	0.32	0.20	0.03	0.04	0.05	0.01	0.22
$\widehat{ST}_i$	0.04	0.14	0.11	0.00	0.00	0.03	0.00	0.23

strength and the eight input variables has been constructed using a penalized spline regression model within a stochastic modeling framework. Consequently, the stochastic nature of the input variables has been included to account for the uncertain mechanisms in both virtual simulations and practical experiments.

Acknowledgements We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

Competing interests The authors declare that they have no conflict of interest.

References

1. Yeh I C. Design of high-performance concrete mixture using neural networks and nonlinear programming. *Journal of Computing in Civil Engineering*, 1999, 13(1): 36–42
2. Kasperkiewicz J, Racz J, Dubrawski A. HPC strength prediction

- using artificial neural network. *Journal of Computing in Civil Engineering*, 1995, 9(4): 279–284
3. Yeh I C. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 1998, 28(12): 1797–1808
4. Yeh I C. Analysis of strength of concrete using design of experiments and neural networks. *Journal of Materials in Civil Engineering*, 2006, 18(4): 597–604
5. Abrams D. *Design of Concrete Mixtures*. Chicago, IL: Structural Materials Laboratory, Lewis Institute, 1918
6. Urbas U, Zorko D, Vukašinović N. Machine learning based nominal root stress calculation model for gears with a progressive curved path of contact. *Mechanism and Machine Theory*, 2021, 165: 104430
7. Mai H V T, Nguyen M H, Trinh S H, Ly H B. Optimization of machine learning models for predicting the compressive strength of fiber-reinforced self-compacting concrete. *Frontiers of Structural and Civil Engineering*, 2023, 17(2): 284–305
8. Noureldin M, Ali T, Kim J. Machine learning-based seismic assessment of framed structures with soil–structure interaction. *Frontiers of Structural and Civil Engineering*, 2023, 17(2): 205–223
9. Lin C J, Wu N J. An ANN model for predicting the compressive strength of concrete. *Applied Sciences*, 2021, 11(9): 3798
10. Ziolkowski P, Niedostatkiewicz M. Machine learning techniques in concrete mix design. *Materials*, 2019, 12(8): 1256
11. Öztaş A, Pala M, Özbay E, Kanca E, Çag'lar N, Bhatti M A. Predicting the compressive strength and slump of high strength concrete using neural network. *Construction and Building Materials*, 2006, 20(9): 769–775
12. I.M. Sobol'. Sensitivity analysis for non-linear mathematical models, mathematical modelling and computational experiment. *Mathematical Modeling & Computational Experiment*, 1993, 4: 407–411
13. Saltelli A, Homma T. Importance measures in global sensitivity analysis of model output. *Reliability Engineering and System Safety*, 1996, 55(2): 1–17
14. Saltelli A, Annoni P, Azzini I, Campolongo F, Ratto M, Tarantola S. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Computer Physics Communications*, 2010, 181(2): 259–270
15. Yeh I C. *Concrete Compressive Strength*. 2007 (available at the website of UC Irvine Machine Learning Repository)
16. Stein M. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 1987, 29(2): 143–151
17. Vu-Bac N, Lahmer T, Zhuang X, Nguyen-Thoi T, Rabczuk T. A software framework for probabilistic sensitivity analysis for computationally expensive models. *Advances in Engineering Software*, 2016, 100: 19–31
18. Ruppert D, Wand M P, Carroll R J. *Semiparametric Regression* (No. 12). Cambridge: Cambridge University Press, 2003
19. Vu-Bac N, Silani M, Lahmer T, Zhuang X, Rabczuk T. A unified framework for stochastic predictions of mechanical properties of polymeric nanocomposites. *Computational Materials Science*, 2015, 96: 520–535
20. Papaioannou I, Straub D. Variance-based reliability sensitivity analysis and the FORM α . *Reliability Engineering and System Safety*, 2021, 210: 107496
21. Kucherenko S, Tarantola S, Annoni P. Estimation of global sensitivity indices for models with dependent variables. *Computer Physics Communications*, 2012, 183(4): 937–946
22. Vu-Bac N, Rafiee R, Zhuang X, Lahmer T, Rabczuk T. Uncertainty quantification for multiscale modeling of polymer nanocomposites with correlated parameters. *Composites. Part B, Engineering*, 2015, 68: 446–464
23. Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY: Association for Computing Machinery, 2016, 785–794
24. Chollet F. *Deep Learning with Python*. Shelter Island, NY: Manning Publications, 2018
25. 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
26. Vu-Bac N, Lahmer T, Zhang Y, Zhuang X, Rabczuk T. Stochastic predictions of interfacial characteristic of polymeric nanocomposites (PNCs). *Composites. Part B, Engineering*, 2014, 59: 80–95
27. Forrester A, Sobester A, Keane A. *Engineering Design via Surrogate Modelling: A Practical Guide*. Hoboken, NJ: John Wiley & Sons Ltd, 2008
28. American coal ash association. *Fly Ash Facts for Highway Engineers*. Technical report FHWA-IF-03-019. 2003