Journal of Science and Transport Technology
Journal homepage: https://jstt.vn/index.php/en

Interpretable Machine Learning Model for Evaluating Flexural Strength of Ultra High-Performance Concrete

Panagiotis G. Asteris[1], Quang Hung Nguyen[2], Trung Hieu Vu[3], Ly Thi Tran[4,*]
[1]Computational Mechanics Laboratory, School of Pedagogical and Technological Education, Marousi, Athens, 15122 Greece
[2]Thuyloi University, Hanoi 100000, Vietnam
[3]RIMAS Research Group (Resilience & Innovative Materials for Smart Infrastructures), University of Transport Technology, 54 Trieu Khuc, Thanh Liet, Hanoi, Vietnam
[4]University of Transport Technology, Hanoi 100000, Vietnam

Article info
Type of article:
Original research paper

DOI:
https://doi.org/10.58845/jstt.utt.2026.en.6.1.215-242

***Corresponding author:**
Email address:
lytt@utt.edu.vn

Received: 19/08/2025
Received in Revised Form: 21/11/2025
Accepted: 02/12/2025

Abstract: Ultra-high-performance concrete (UHPC) mix design remains experimentally expensive because many ingredients interact nonlinearly to govern flexural behavior. An interpretable machine-learning pipeline was developed to predict UHPC flexural strength from literature-derived mixes (317 observations, 14 input variables). Nine regression models were screened under a rigorous Monte-Carlo protocol (1,000 random 70/30 splits). Tree-based boosting dominated: on a representative split, CatBoost achieved R^2test=0.928, RMSE = 1.980 MPa, MAE = 1.454 MPa, MAPE = 8.386%, with XGBoost close behind; Random Forest and Gradient Boosting formed a reliable second tier, while linear, SVR, and KNN underfit. Global and local interpretability (SHAP and PDP) revealed a stable hierarchy of drivers: steel fiber content and curing time were strongly beneficial; coarse aggregate content was deleterious and nearly monotonic; water became increasingly harmful at high dosages; superplasticizer exhibited an interior “sweet spot”; cement and silica fume were favorable (silica fume above ~100 kg/m³); sand was weakly positive; limestone powder was near-neutral. Guided by mean|SHAP|, the feature set was reduced from 14 to 9 variables with only a modest trade-off (R^2test =0.916, RMSE = 2.143 MPa, MAE = 1.522 MPa, MAPE = 9.07%). External verification on an independent dataset confirmed generalization and preserved the correct nonlinear response to steel fibers in the practical 0-2% range. A lightweight GUI operationalizes the nine-input model, enabling rapid “what-if” exploration and reducing measurement burden by 36%. The results deliver both accuracy and transparency, distilling actionable rules for UHPC tailored to flexure-critical applications: prioritize steel fibers and adequate curing, cap coarse aggregate, and maintain water/superplasticizer within stable windows while using cement and silica fume to tune the matrix.

Keywords: Ultra-High-Performance Concrete, Flexural Strength, CatBoost, SHAP, Partial Dependence Plots.

1. Introduction

Ultra-high-performance concrete (UHPC) has been recognized as a family of fiber-reinforced cementitious composites engineered with very low water-to-binder ratios, dense particle packing, refined microstructure, and superior mechanical performance and durability [1]. Characteristic compressive strengths above 120 MPa at 28 days have been widely reported, while discrete fibers have been incorporated to bridge cracks and to sustain post-cracking tensile capacity, enabling use in repair, rehabilitation, and new construction [2]. These attributes have been attributed to deliberate mix design in which granular packing has been optimized, porosity has been lowered, and steel or polymer fibers have been dispersed within a compact matrix to suppress microcracking and to enhance energy absorption [3, 4].

UHPC mixtures have typically been proportioned under strict rheology control at low water-to-binder ratios, with particle size distributions across cement, supplementary cementitious materials (SCMs) notably silica fume and nano-silica and fine aggregates optimized to maximize packing density; fiber type, geometry, and dosage have been selected to tailor tensile and flexural response [1, 4, 5]. The presence of coarse aggregate has often been reduced or eliminated to improve homogeneity and the interfacial transition zone; however, brittleness has been observed if fiber bridging is inadequate, superplasticizer dosages have required windowing to avoid segregation, and curing regimes have been shown to condition both early-age and long-term gains [4, 6]. As a result, a multivariate design space has been confronted in practice, within which nonlinear interactions among binder fractions, aggregate gradation, admixture content, fiber parameters, and curing schedule have been repeatedly documented [7].

Although targeted experiments have clarified mechanisms such as silica-fume-induced densification, nano-silica seeding, or fiber-shape effects on crack bridging comprehensive factorial exploration has been found to be costly in time, labor, and materials, especially as the number of controllable variables grows [5]. To address this challenge, data-driven modeling has been adopted increasingly, and machine learning (ML) methods have been leveraged to learn complex response surfaces directly from curated mix property datasets [8–10]. On structured, tabular materials data, tree-ensemble models including Gradient Boosting, Random Forests, XGBoost, LightGBM, and CatBoost have been shown to capture nonlinearities and interactions while maintaining competitive accuracy and computational efficiency [11, 12]. Model interpretability has been strengthened through Shapley Additive Explanations (SHAP) [13] and partial-dependence plots (PDP) [14], by which variable attributions and response shapes have been exposed and compared against micromechanical expectations (e.g., beneficial roles of steel fibers and adequate curing; adverse effects of excessive water or coarse aggregate) [15–19]. In several UHPC studies, practical deployment has been facilitated by user-facing tools, through which trained models have been placed in engineers' hands for rapid "what-if" screening during proportioning and specification [20, 21].

Despite these advances, persistent gaps have been identified. Many datasets have been assembled from heterogeneous sources with inconsistent measurement conventions, specimen geometries, and curing protocols; anomalous entries and outliers have not always been screened systematically, and generalization has consequently been degraded [22–24]. Variable sets have often been chosen ad hoc, with weak or rarely used contributors retained and practical levers omitted, thereby inflating variance and blurring attribution [16, 18, 22]. Most importantly for serviceability and toughness, flexural behavior has been explored less systematically than compressive strength in ML settings, despite strong experimental evidence that steel-fiber dosage and geometry exert nonlinear, saturating

influences on flexural response, and that coarse aggregate can disrupt homogeneity and fiber dispersion [24–28]. In addition, robust anomaly detection and rank-based exploratory checks have remained underused, even though isolation-based screening and nonparametric correlations offer practical safeguards for small-to-moderate UHPC datasets.

Within this context, an interpretable pipeline focused on UHPC flexural strength has been pursued. A curated feature set composed of routinely measured quantities cement and SCMs (e.g., silica fume, nano-silica, slag, fly ash), limestone powder, fine/coarse aggregates, water, superplasticizer, steel/polystyrene fibers, and curing time has been emphasized to ensure usability and auditability in mix design and quality control [15, 22, 29]. Model development has been carried out using strong baselines across the tree-ensemble family (Gradient Boosting, Random Forests, XGBoost, LightGBM, CatBoost), under a disciplined evaluation protocol based on repeated Monte-Carlo train/test splits, so that distributions of accuracy metrics rather than single-split values are reported [30]. Hyperparameters have been optimized using modern search strategies to mitigate overfitting and to improve robustness [31]. Data reliability has been enhanced by isolation-based anomaly screening and rank-correlation diagnostics during exploratory analysis, so that spurious leverage is reduced before fitting [25, 26]. Interpretability has been delivered with SHAP for global and local attributions and with PDP for response-shape visualization and uncertainty banding across practically relevant dosage ranges.

Through this approach, a consistent, mechanics-aligned picture has been obtained. Steel-fiber volume and curing time have emerged as dominant positive drivers of flexural strength; coarse aggregate within the sampled range has shown a monotonic negative effect; cement and silica fume have contributed modest gains; water above a stable window and excessive superplasticizer have been associated with declining performance patterns in agreement with micromechanics and with classic UHPC proportioning heuristics [25, 27]. Accuracy and stability have been found to be highest for boosting-based learners (CatBoost, XGBoost), with Random Forest and Gradient Boosting forming a durable second tier, consistent with broader findings on structured tabular problems in materials engineering [30]. For practical adoption, a lightweight graphical user interface has been provided so that rapid “what-if” mix exploration can be conducted during preliminary design and performance screening without displacing confirmatory laboratory tests. External experimental trends on steel-fiber dosage have been used for qualitative verification, and the learned nonlinear slope with saturation at ≈2-3 % volume has been preserved when compared against independent data [7].

Following this introduction, the dataset together with the analysis is presented in Section 2. Section 3 describes the 9 machine learning algorithms used in this study. The results are then developed in Section 4: model performance is assessed on repeated splits and Monte Carlo simulations is used for evaluating the reliability of ML models (4.1), representative prediction outcomes are examined (4.2), and a global interpretation is provided in Section 4.3, where SHAP/PDP analyses are consolidated with a lightweight GUI demonstration and out-of-sample verification.

2. Database analysis for Machine Learning model

The dataset used for training the machine learning models consists of 317 data points from [26], each containing 14 input variables related to the composition of concrete. These features, including cement content, fly ash content, slag content, silica fume content, nano silica content, limestone powder content, sand content, coarse aggregate content, quartz powder content, water content, superplasticizer content, polystyrene fiber

content, steel fiber content, and curing time, serve as the predictors for flexural strength, the target variable.

Table 1 provides detailed statistics for each of the 13 input variables in the dataset. The mean represents the average value for each feature, while the standard deviation indicates the spread or variability of the data. The minimum and maximum values show the range of the dataset, and the Q25, Q50, and Q75 represent the 25th, 50th (median), and 75th percentiles, respectively, which help to understand the distribution of the data. Cement content has a wide range, from 317.0 kg/m³ to 1079.0 kg/m³, with a mean of 711.7 kg/m³. The spread is substantial, indicating variability in cement content across the dataset. Fly ash content and slag content both show high variability, with standard deviations of 96.6 kg/m³ and 126.7 kg/m³, respectively, but the median values (Q50) are close to zero for many samples. Nano silica content and silica fume content are typically low, with the mean values being 9.2 kg/m³ and 109.0 kg/m³, respectively. This suggests that these ingredients are used sparingly in the concrete mix. Sand content and coarse aggregate content have high mean values (1047.4 kg/m³ and 118.6 kg/m³) and show significant variation across the samples, with sand content ranging from 408.0 kg/m³ to 1503.4 kg/m³. Water content shows a mean value of 180.4 kg/m³, with a relatively low standard deviation (45.5 kg/m³), indicating that water content does not vary as widely as some other materials. Steel fiber content and superplasticizer content have relatively low means (0.1% and 29.3%) and show less variability compared to other ingredients like cement content and sand content.

Table 1. Statistical Summary of Variables in the UHPC Dataset

	Unit	Mean	Std	Min	$Q_{25\%}$	$Q_{50\%}$	$Q_{75\%}$	Max
Cement content	kg/m^3	711.7	171.0	317.0	606.4	699.0	850.3	1079.0
Fly ash content	kg/m^3	43.1	96.6	0.0	0.0	0.0	0.0	475.0
Slag content	kg/m^3	50.3	126.7	0.0	0.0	0.0	0.0	475.0
Silica fume content	kg/m^3	109.0	101.7	0.0	0.0	74.5	216.0	273.0
Nano silica content	kg/m^3	9.2	15.2	0.0	0.0	0.0	24.0	43.7
Limestone powder content	kg/m^3	55.2	104.9	0.0	0.0	0.0	0.0	272.9
Sand content	kg/m^3	1047.4	278.3	408.0	1000.0	1056.0	1273.4	1503.4
Coarse aggregate content	kg/m^3	118.6	285.1	0.0	0.0	0.0	58.0	990.0
Quartz powder content	kg/m^3	16.6	51.5	0.0	0.0	0.0	0.0	175.9
Water content	kg/m^3	180.4	15.0	144.0	173.0	177.0	185.0	228.0
Superplasticizer content	kg/m^3	29.3	12.1	8.0	21.0	24.0	43.9	52.0
Polystyrene fiber content	%	0.1	0.3	0.0	0.0	0.0	0.0	2.0
Steel fiber content	%	1.9	2.3	0.0	0.0	2.0	2.0	17.0
Curing time	days	50.5	121.6	1.0	7.0	28.0	28.0	730.0
Flexural strength	MPa	19.1	7.6	5.6	12.8	18.9	23.8	41.5

The variability in these features is important as it can influence the predictions made by machine learning models, and the preprocessing steps may involve scaling or normalization of these features to ensure consistent input values across the dataset.

Fig. 1 shows the distribution of each input variable along with its relationship to flexural strength. These plots include scatter plots and histograms that depict how each feature correlates with the target variable. Cement content (Fig. 1a) shows a positive correlation with flexural strength. As the cement content increases, flexural strength tends to rise as well. This is consistent with the correlation coefficient of 0.36 in Fig. 2, indicating a moderate positive relationship. Cement content is

therefore an important factor in determining concrete strength. Fly ash content (Fig. 1b) shows a weak positive correlation with flexural strength. The scatter plot suggests a slight increase in flexural strength with fly ash content, but the trend is not as strong as that for cement content. The correlation coefficient of 0.18 in Fig. 2 confirms that fly ash content has a minimal impact on flexural strength. Slag content (Fig. 1c) also shows a weak positive relationship with flexural strength. Like fly ash, slag content contributes slightly to increasing flexural strength, but the effect is not significant. The correlation coefficient of 0.26 in Fig. 2 indicates a mild positive correlation.

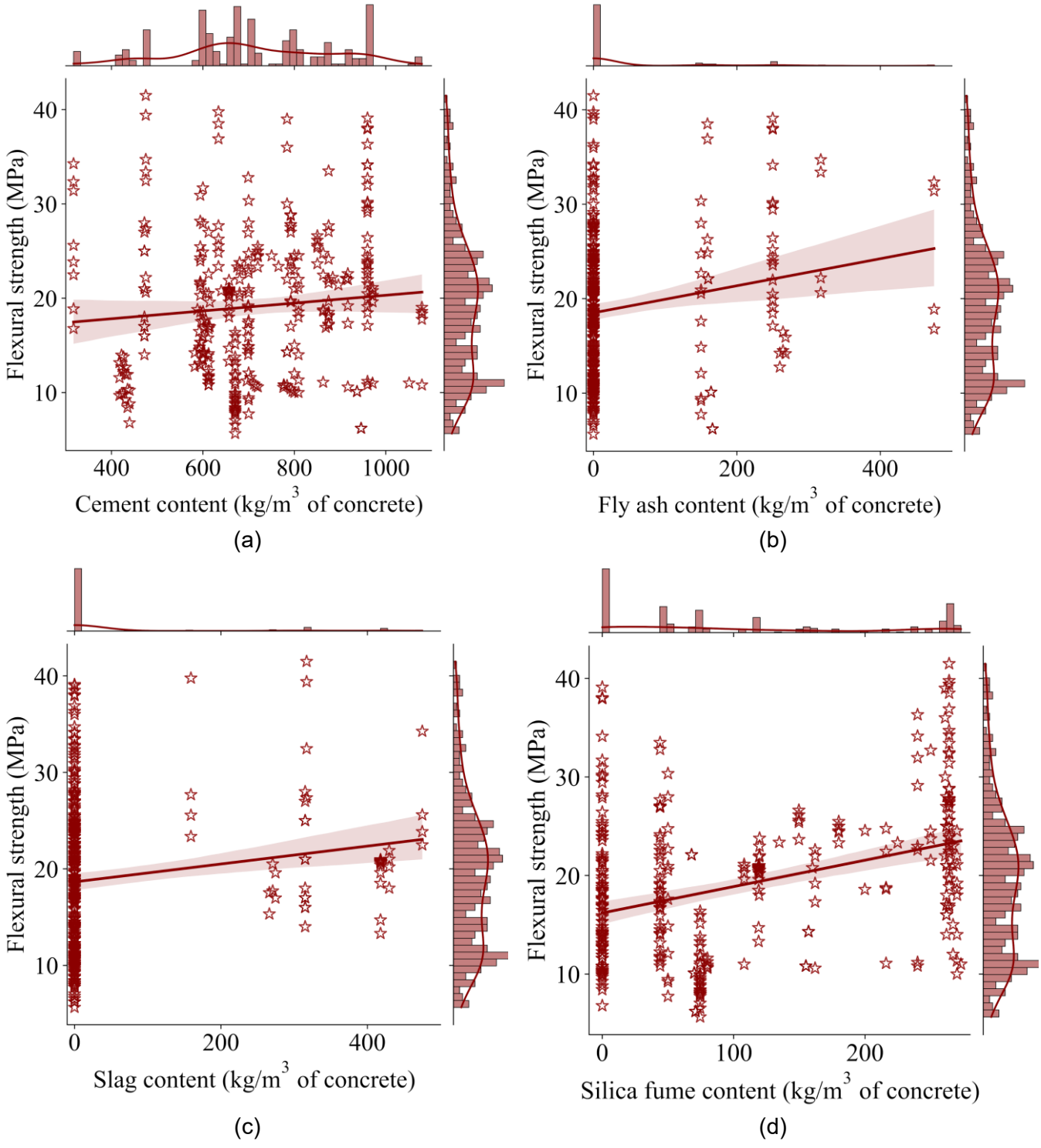


Fig. 1. Distribution samples of each input variable and Flexural strength of UHPC

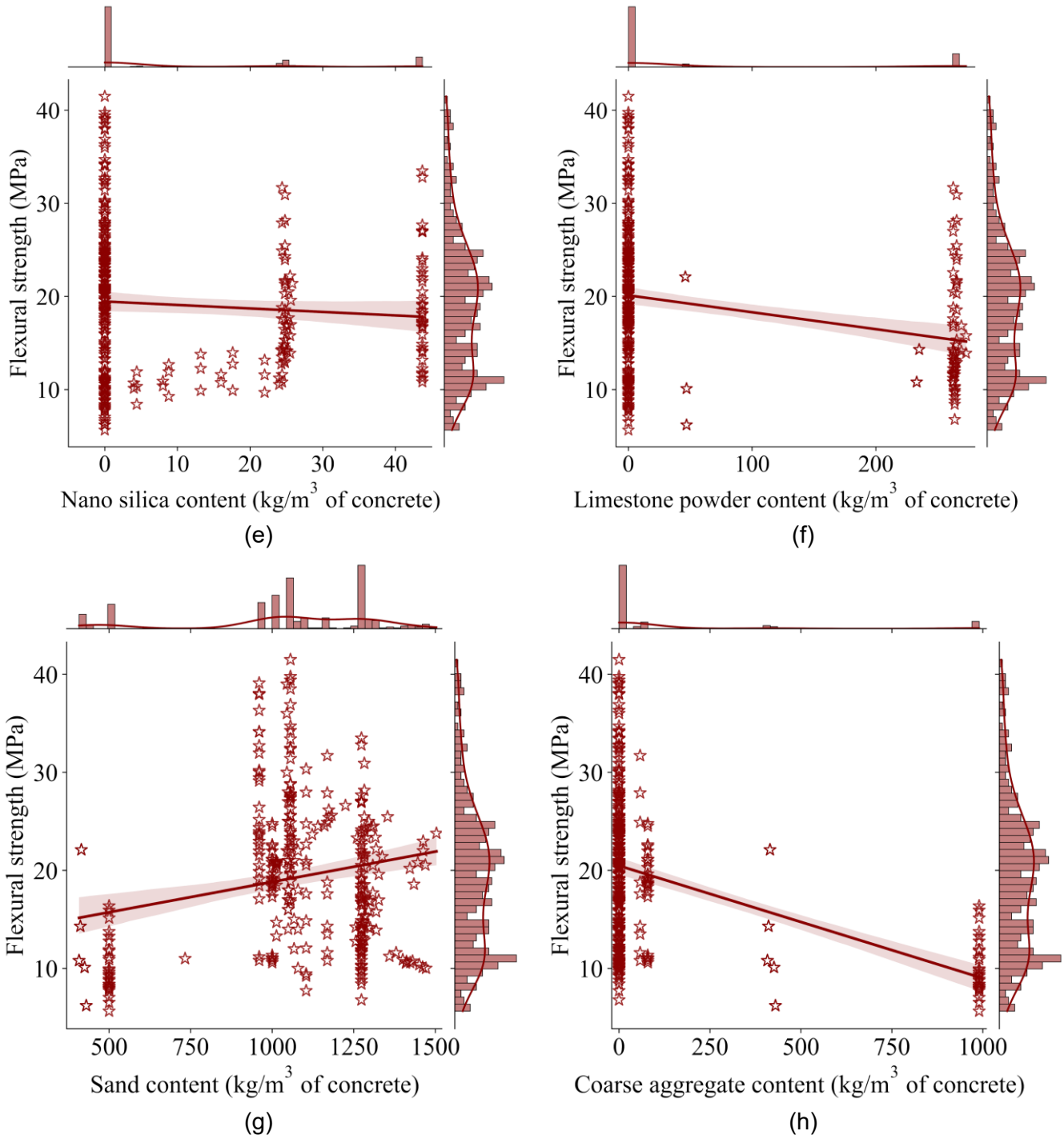


Fig. 1. (continued)

Silica fume content (Fig. 1d) has a weak positive effect, although it is less pronounced than cement content. The correlation coefficient of 0.19 in Fig. 2 indicates a slight positive relationship, suggesting that silica fume content has a modest effect on flexural strength. Nano silica content (Fig. 1e) exhibits a weak negative correlation with flexural strength, as reflected in the scatter plot. As nano silica content increases, flexural strength tends to decrease slightly. The correlation coefficient of-0.16 in Fig. 2 suggests a very weak negative relationship. Limestone powder content (Fig. 1f) shows a weak positive effect on flexural strength, similar to silica fume. The correlation coefficient of 0.24 in Fig. 2 indicates a slight positive relationship. Sand content (Fig. 1g) exhibits almost no linear relationship with flexural strength, with the scatter plot showing a slight

upward trend. However, the correlation coefficient of-0.07 in Fig. 2 suggests that sand content does not significantly influence flexural strength. Coarse aggregate content (Fig. 1h) shows a negative correlation with flexural strength. As the amount of coarse aggregate increases, flexural strength decreases, which is confirmed by the correlation coefficient of-0.79 in Fig. 2, indicating a strong negative relationship. Quartz powder content (Fig. 1i) shows a weak positive correlation with flexural strength, with the scatter plot indicating a slight increase in strength as quartz powder content rises. The correlation coefficient of 0.32 in Fig. 2 supports this weak positive relationship. Water content (Fig. 1j) exhibits a negative correlation with flexural strength. Higher water content leads to lower flexural strength, as reflected in the correlation coefficient of-0.43 in Fig. 2.

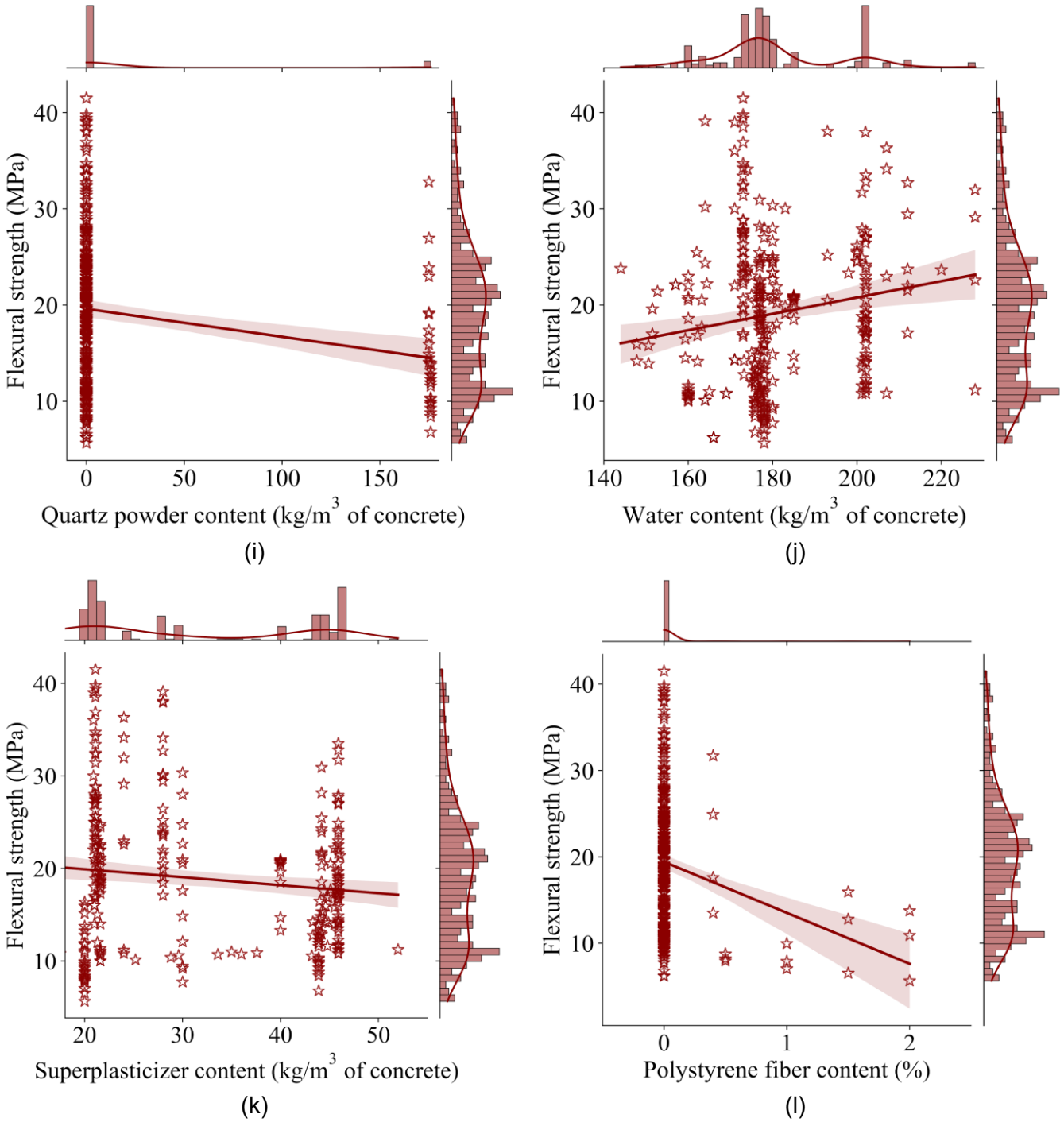
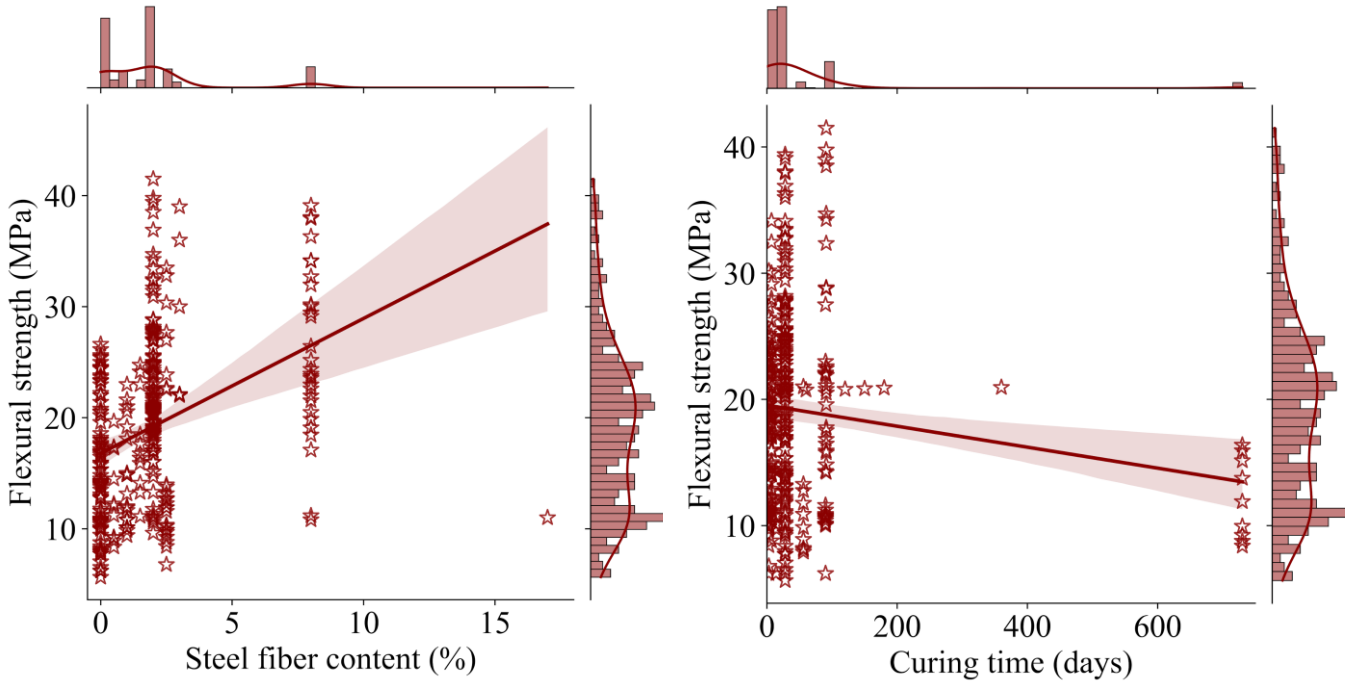


Fig. 1. (continued)



(m) (n)

Fig. 1. (continued)

Superplasticizer content (Fig. 1k) shows a weak positive correlation with flexural strength. The correlation coefficient of 0.25 in Fig. 2 suggests a slight positive effect, where increasing superplasticizer content can improve flexural strength. Polystyrene fiber content (Fig. 1l) exhibits a negative correlation with flexural strength. As the content of polystyrene fibers increases, the flexural strength tends to decrease, which is supported by the correlation coefficient of-0.21 in Fig. 2.

Steel fiber content (Fig. 1m) shows a positive correlation with flexural strength. The more steel fibers are added, the greater the flexural strength. This is confirmed by the correlation coefficient of 0.37 in Fig. 2.

Curing time (Fig. 1n) exhibits a weak positive correlation with flexural strength. The correlation coefficient of 0.19 in Fig. 2 indicates that curing time has a minor but positive effect on flexural strength.

Fig. 2 presents the correlation matrix of the input variables and the target variable (flexural strength). This matrix quantifies the strength and direction of the linear relationships between the features. It supports the trends observed in Fig. 1: Cement content has a moderate positive correlation of 0.36 with flexural strength, confirming its significant role in concrete strength. Coarse aggregate content has a strong negative correlation of-0.79 with flexural strength, indicating that more coarse aggregate generally leads to lower flexural strength. Water content shows a negative correlation of-0.43, which aligns with the observation in Fig. 1j that increased water content decreases flexural strength.

The analysis of the statistical summary, the visualizations in Fig. 1, and the correlation matrix in Fig. 2 reveals important insights into how each feature influences the flexural strength of concrete. Variables such as cement content, steel fiber content, and curing time show stronger relationships with flexural strength, while other variables like sand content and coarse aggregate content exhibit weaker or negative correlations.

To evaluate the overall impact of these features and to better understand their influence on the prediction accuracy of the machine learning models by using SHAP (SHapley Additive exPlanations) and PDP (Partial Dependence Plots) which are two important tools in machine learning

for interpreting and explaining model predictions, the 13 input variables will be retained for further analysis and model development. These features will be included in the construction and evaluation of the machine learning model aimed at predicting the flexural strength of UHPC. The next step involves further preprocessing, feature engineering, and applying machine learning techniques to develop a robust predictive model based on these input features.

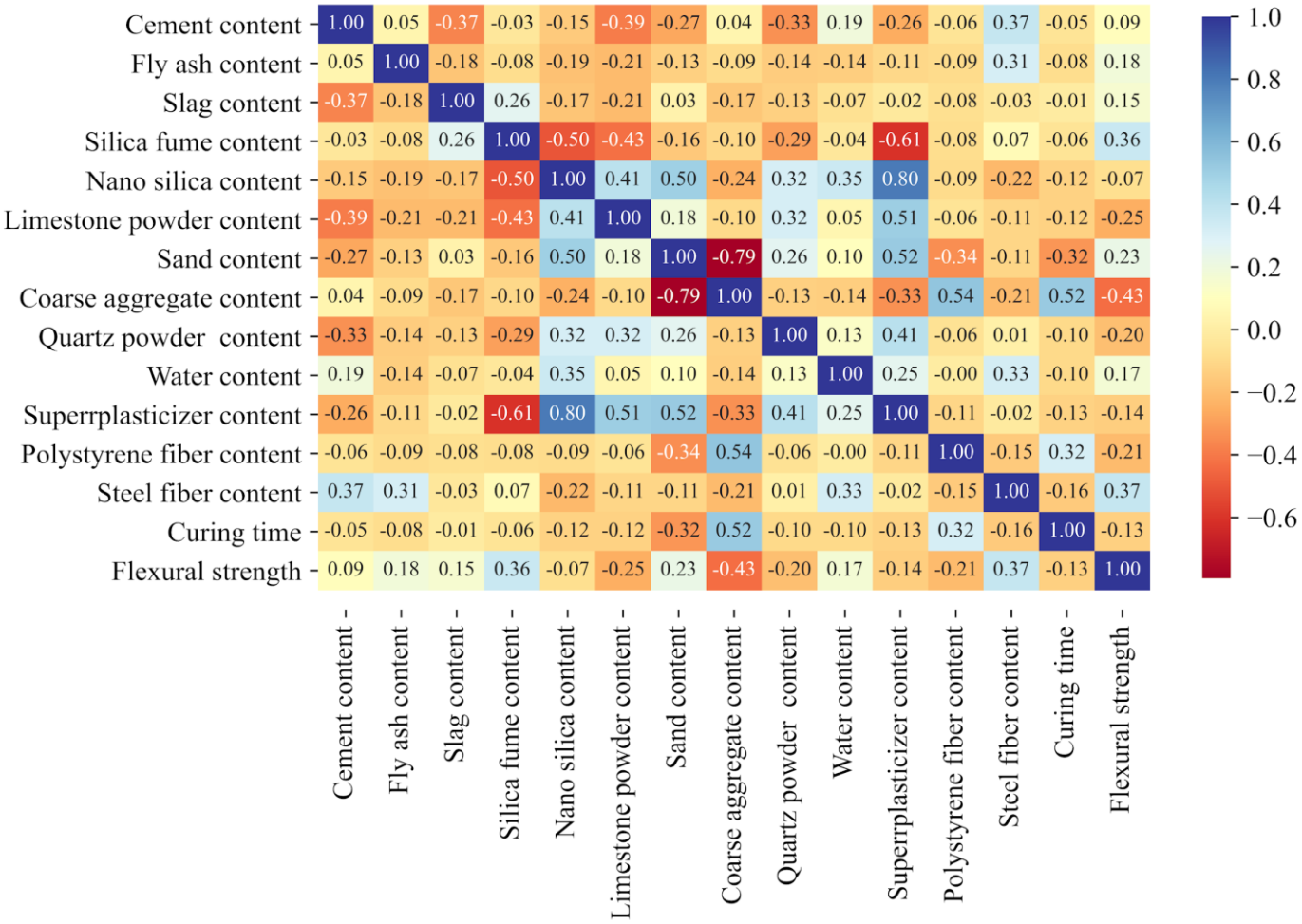


Fig. 2. Correlation Matrix of Input Variables and Flexural Strength

3. Machine Learning algorithms

In this study, nine supervised machine learning algorithms were selected to span a representative spectrum of model families that are suitable for small-to-moderate tabular datasets such as UHPC mixture databases. A multiple linear regression (LR) model was first included as a weak baseline to quantify the performance gains obtained from more flexible non-linear learners. Support vector regression (SVR) and K-Nearest Neighbours (KNN) were then considered as kernel-based and instance-based methods that can capture moderate non-linear relationships while remaining relatively simple and robust for limited data. A single decision tree (DT) was used as a transparent but high-variance benchmark, providing intuitive if-then rules at the cost of reduced generalization.

The remaining five models are ensemble tree-based algorithms: Random Forest (RF), Gradient Boosting (GB), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Categorical Gradient Boosting (CatBoost). Ensemble trees, and in particular gradient boosting variants, have repeatedly delivered state-of-the-art accuracy for concrete strength prediction, clearly outperforming traditional regression or single-model baselines in

terms of R2 and error metrics. Recent studies have shown that optimized XGBoost and related boosting models are highly effective for forecasting the compressive strength of conventional and fly ash concretes, as well as other cementitious mixtures [32–36]. In parallel, CatBoost has been successfully combined with SHAP-based interpretability to design accurate, explainable models for concrete compressive strength and high-performance concrete mixtures (Fu et al., 2025; Zhang and Ren, 2024). Moreover, random forest models have already been applied to predict the flexural strength and flexural behaviour of fiber-reinforced UHPC, confirming the suitability of tree ensembles for this class of materials (Sarmiento-Pupo et al., 2025). On this basis, the present work uses the nine models above to cover a broad range of bias-variance trade-offs and modelling philosophies that are known to be effective for concrete and UHPC datasets, while remaining compatible with SHAP and partial-dependence analyses.

The ML models were evaluated based on their ability to maximize the coefficient of determination R^2 and minimize the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean absolute percentage error (MAPE).

Linear Regression (Linear)

Linear regression is one of the simplest and most widely used algorithms in machine learning. It establishes a relationship between the input features and the target variable by fitting a linear equation to the observed data. This model assumes that there is a linear relationship between the predictors and the target variable, making it interpretable but sometimes less accurate in capturing complex patterns in the data. It is often used as a baseline model for comparison. Linear regression assumes a linear relationship between features and the target variable, which may limit its accuracy on more complex datasets. Additionally, it is sensitive to outliers, which can significantly impact the model's predictions [8].

Support Vector Regression (SVR)

Support Vector Regression (SVR) is a regression technique derived from Support Vector Machines (SVM). SVR attempts to find a hyperplane in a high-dimensional space that best fits the data, while also ensuring that the errors of the model remain within a specified margin. SVR is well-suited for data with complex, non-linear relationships, and is often preferred for its robustness to overfitting when tuned appropriately. SVR aims to find a hyperplane that best fits the data while minimizing errors within a specific margin, making it robust against overfitting [37]. SVR is computationally expensive, particularly for large datasets, and requires careful tuning of hyperparameters like the kernel and epsilon to perform optimally [37]. Moreover, it may not scale well to extremely large datasets or complex feature spaces.

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple, yet powerful algorithm that classifies a data point based on the majority label of its nearest neighbors. In regression, the algorithm predicts the target variable by averaging the values of the nearest neighbors. KNN does not assume any underlying data distribution, making it highly flexible, though it can be computationally expensive for large datasets. KNN can be computationally expensive during prediction, as it requires calculating distances to all training samples. Additionally, its performance may degrade in high-dimensional spaces (the curse of dimensionality) and is sensitive to the choice of K (the number of neighbors) and the distance metric used [38].

Light Gradient Boosting Machine (LightGB)

Light Gradient Boosting Machine **(LightGB)** is a highly efficient gradient boosting framework that is optimized for speed and accuracy. It utilizes a histogram-based approach for decision tree learning, which helps reduce memory usage and improves performance on large datasets. LightGBM has become increasingly popular in

machine learning competitions due to its high efficiency and ability to handle categorical features natively. It has proven effective in machine learning competitions due to its speed and accuracy [39]. While LightGB is powerful, it can overfit if not carefully tuned. It also has limitations when dealing with imbalanced datasets, although these can be addressed with hyperparameter tuning.

Gradient Boosting (GB)

Gradient Boosting is an ensemble learning method that builds a series of weak learners, typically decision trees, in a sequential manner. Each new model attempts to correct the errors of the previous model, resulting in a strong predictive model. Gradient Boosting is known for its high accuracy but can be sensitive to overfitting if not tuned properly. It performs well on both structured and unstructured data [40]. Gradient Boosting can be slow to train due to the sequential nature of its tree-building process. It also requires careful tuning of hyperparameters such as learning rate and number of trees to prevent overfitting, which can be computationally expensive [40].

Random Forest (RF)

Random Forest is another ensemble learning method that constructs multiple decision trees and merges them together to improve the accuracy and control overfitting. Random Forest is particularly useful for handling high-dimensional data and can model complex interactions between features. It is robust and often provides good results with minimal tuning [41]. Despite its strengths, Random Forest models can become computationally expensive when the number of trees is large. Additionally, although it is less prone to overfitting than single decision trees, it can still be sensitive to noise in the data if not properly tuned [41].

CatBoost (CatB)

CatBoost (Categorical Boosting) is a gradient boosting algorithm developed by Yandex, optimized for categorical features. It performs well on datasets with categorical variables without requiring much preprocessing. CatBoost handles categorical features efficiently by using ordered boosting and supports both regression and classification tasks. It performs exceptionally well with minimal preprocessing and is efficient in handling large datasets. CatBoost has become increasingly popular due to its superior performance and ease of use with categorical variables [42]. Despite its efficiency, CatBoost can be harder to tune than other gradient boosting methods due to fewer resources available for hyperparameter optimization. Additionally, its performance on very large datasets may not be as high as other tree-based methods like XGB in certain scenarios [42].

XGB (XGB)

XGB (Extreme Gradient Boosting) is one of the most popular gradient boosting algorithms due to its performance and scalability. It is designed to be highly efficient, flexible, and portable. XGB is widely used in machine learning competitions and has been shown to perform excellently on various structured datasets. XGB is particularly suited for regression and classification tasks on structured data [43]. Like other boosting methods, XGB can overfit if hyperparameters are not well-tuned. Additionally, it can be computationally expensive when working with very large datasets, though its scalability is one of its key strengths [43].

Decision Tree (DT)

Decision Tree is a non-linear model that splits the data into subsets based on feature values, creating a tree structure where each leaf node represents a predicted value. Decision Trees are easy to interpret and visualize, which makes them highly useful for understanding how predictions are made. They are non-linear and can model complex relationships between features and the target variable [44]. Decision Trees are prone to overfitting, especially when the tree depth is too large. They are also less stable, meaning small changes in the data can significantly alter the tree structure. Furthermore, they may struggle with handling missing data or very high-dimensional datasets without proper tuning [44].

Each machine learning algorithm has its

unique strengths and weaknesses, and selecting the most appropriate model depends on the characteristics of the dataset and the problem at hand. Simple models like Linear Regression are computationally efficient but may not capture the complexity of non-linear relationships. On the other hand, more complex models like XGB and LightGBM offer higher accuracy and can handle large datasets effectively, but they require careful hyperparameter tuning to avoid overfitting. Random Forest and Gradient Boosting are robust ensemble methods that often provide excellent performance, while SVR and KNN excel in non-linear relationships but may struggle with larger or high-dimensional data. The performance of these algorithms will be further evaluated based on their performance metrics including R^2, RMSE, MAE and MAPE, which will guide the model selection for predicting flexural strength in UHPC.

4. Data splitting and Monte Carlo cross-validation

To assess the generalization performance of the nine machine-learning models, we adopted a Monte Carlo cross-validation (also referred to as repeated random sub-sampling or shuffle-split cross-validation). In each iteration, the full dataset was randomly partitioned into a training subset containing 70% of the data and a testing subset containing the remaining 30% of the data. The model was fitted on the training subset and evaluated on the testing subset. This procedure was repeated 1,000 times with independently generated random splits, and the reported performance metrics correspond to the mean and dispersion (standard deviation and selected quantiles) over these 1,000 Monte Carlo replications.

The 70/30 train-test ratio was chosen as a compromise between (i) providing a sufficiently large training set to calibrate flexible non-linear models, and (ii) maintaining a relatively large, truly unseen test set to estimate generalization error and compare competing algorithms. Similar 70/30 hold-out splits have been widely adopted in recent machine-learning studies on compressive and flexural strength prediction of concrete and ultra-high-performance concrete with comparable sample sizes, which facilitates a consistent comparison with the existing literature [12,45–47].

Using 1,000 Monte Carlo replications allows us to substantially reduce the variance of the error estimates that arises from the randomness of any single train-test split and to approximate the full distribution of the performance metrics for each algorithm. Methodological work on Monte Carlo and resampling-based cross-validation recommends using a relatively large number of repetitions when the goal is to obtain stable estimates and confidence intervals for prediction error, rather than relying on a single split or a small number of folds [48]. Given the modest size of our dataset and the moderate complexity of the considered models, the computational cost of 1,000 repetitions remained low, while the benefit in terms of robustness of model ranking and uncertainty quantification was significant.

In this sense, our evaluation protocol can be viewed as a Monte Carlo cross-validation scheme with a fixed 70/30 split ratio and 1,000 random replications, which is statistically equivalent in spirit to repeated k-fold cross-validation but offers more flexibility in the choice of the train-test proportion and provides a richer characterization of the variability in predictive performance.

5. Results and Discussion

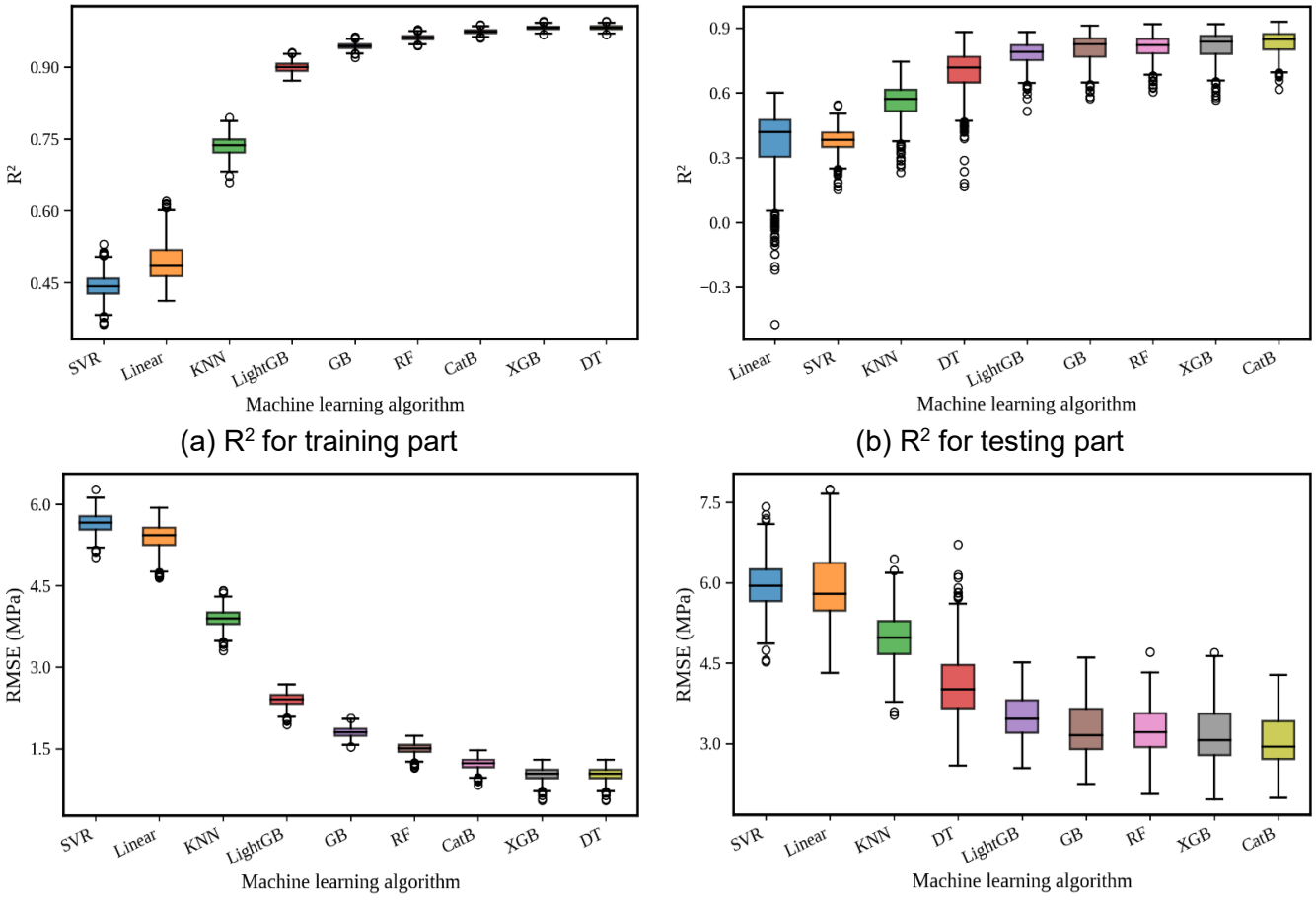
5.1. Evaluating performance of Machine learning model

Using the 317 observations described in Table 1, each experiment adopted a 70/30 split 70 % for training and 30 % for testing and repeated this procedure 1000 times under Monte-Carlo random sub-sampling. At every run, models were refit on the training subset and assessed on the hold-out using R^2, RMSE (MPa), MAE (MPa), and MAPE (%). This design suppresses split idiosyncrasies and exposes both the central tendency and dispersion of model accuracy. The distributional outcomes are visualized as boxplots

in Fig. 3, while the means and standard deviations aggregated over runs are reported in Table 2 and Table 3, respectively.

The evidence in Fig. 3 is unequivocal: tree-based boosting dominates. CatBoost and XGB show the highest median test R^2, the lowest median errors, and the tightest interquartile ranges, indicating not only accuracy but also robustness to resampling. Table 2 quantifies these advantages. On the testing set, CatBoost attains R^2=0.834 with RMSE = 3.041 MPa, MAE = 2.058 MPa, and MAPE = 11.70 %, XGB follows closely with R^2=0.818, RMSE = 3.173 MPa, MAE = 2.052 MPa, and MAPE = 11.40 %. These error magnitudes are meaningfully lower than the rest and, crucially, remain concentrated an observation reinforced by the narrow boxes and short whiskers in Fig. 3.

A second tier is visible just behind the best ML model CatBoost. Random Forest and Gradient Boosting maintain strong performance but with slightly inflated errors and wider spreads. Table 2 shows RF at R^2=0.812, RMSE=3.246 MPa, MAE=2.228 MPa, MAPE = 12.37%, and GB at R^2=0.809, RMSE = 3.268 MPa, MAE = 2.282 MPa, MAPE = 13.05 %. Fig. 3 corroborates this: their medians remain competitive, but the interquartile ranges expand, and occasional outliers appear more frequently than for CatBoost/XGB. LightGB trails this cluster with noticeably higher errors and dispersion, suggesting that its histogram-based splits did not exploit the full structure of this dataset at the given sample size.



(a) R^2 for training part (b) R^2 for testing part

(c) RMSE for training part (d) RMSE for testing part

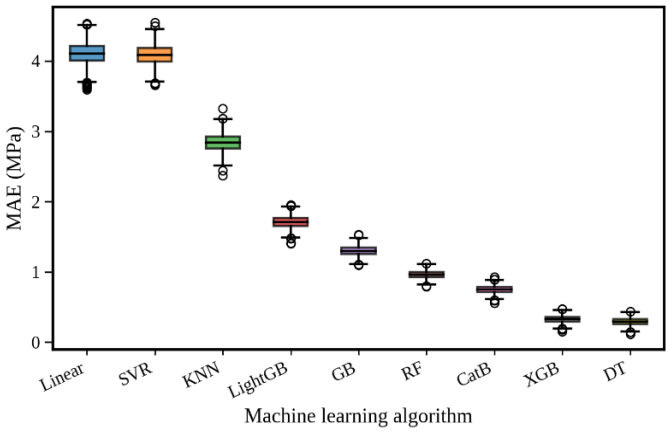
Fig. 3. Using 1000 times of Monte Carlo simulations (MCs) for evaluating the performance of 9 ML models

Models outside the tree-ensemble family underperform for this task. Decision Tree exhibits the classic signature of overfitting: extremely high training R^2 but a sharp decline on testing (Fig. 3 shows the train box nearly pegged at the top while the test box sinks), which Table 2 summarizes with a test R^2 round 0.70 and RMSE ≈ 4.09 MPa. In contrast, Linear regression, SVR, and KNN tend to

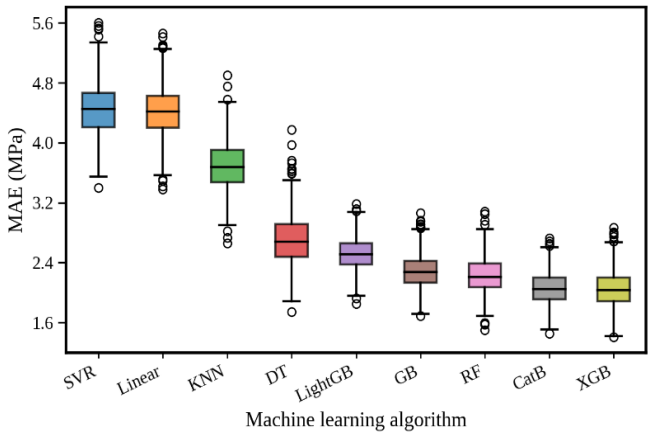
underfit: both train and test R^2 remain low (e.g., Linear R^2=0.375 on test) and error distributions are broad, implying insufficient capacity to capture non-linearities and interactions inherent to UHPC mix strength relationships.

Beyond central accuracy, stability matters. Table 3 shows that CatBoost and XGB also minimize variability across resamples. For CatBoost, the test-set standard deviations are StD(R^2)=0.0510, StD(RMSE)=0.4487 MPa, StD(MAE)=0.2120 MPa, and StD(MAPE)=0.0158; XGB is comparably tight StD(RMSE)=0.5193 MPa. RF and GB exhibit moderate increases in spread, while LightGBM, DT, KNN, SVR, and Linear show the largest standard deviations, confirming the broader boxes and longer whiskers in Fig. 3 and signaling higher sensitivity to the particular train/test partitions.

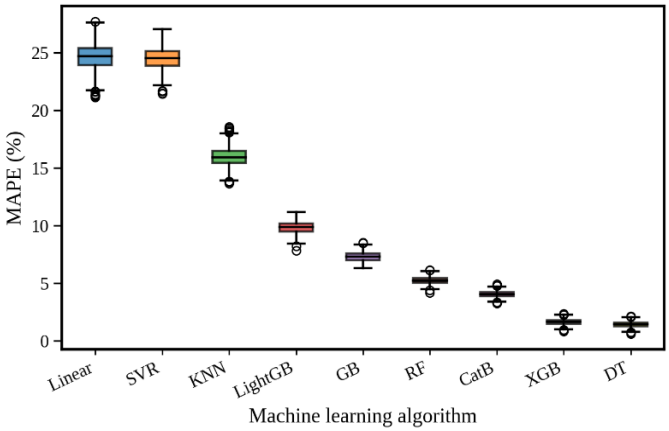
Generalization gaps inferred from Table 2 are consistent with these conclusions. Boosting models achieve very high training R^2 ≈ 0.95-0.98 yet preserve solid test R^2 ≈ 0.81-0.83, indicating capacity without collapse. DT's gap is excessive, betraying brittle fits to noise. By contrast, Linear/SVR/KNN keep gaps small only because their ceilings are low, these ML models fail to model the problem's non-linearity, not because they generalize better.



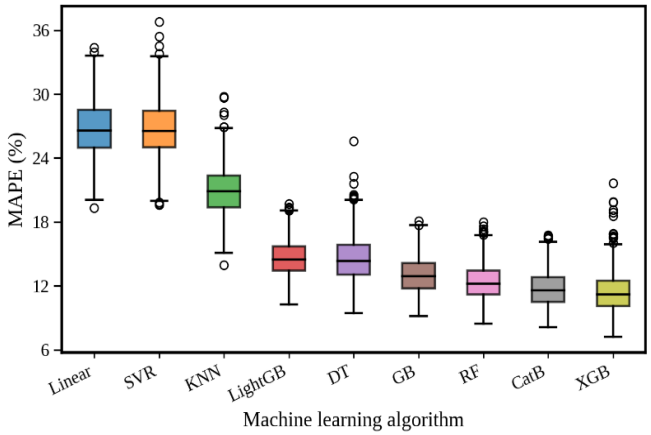
(e) MAE for training part



(f) MAE for testing part



(g) MAPE for training part



(h) MAPE for testing part

Fig. 3. (continued)

CatBoost and XGB are the most accurate and stable; Random Forest and Gradient Boosting form a reliable second tier that remains competitive with slightly higher uncertainty. These four algorithms are therefore selected for deeper analysis in Section 4.2 on predicting UHPC flexural strength.

The random 70/30 partitioning was repeated 1,000 times. At each repetition, the models were re-trained on the training subset and evaluated on the independent test subset, and the resulting R^2, RMSE, MAE and MAPE values were aggregated

to compute their means and standard deviations (Tables 2–3) and empirical distributions (Fig. 3). This procedure corresponds to Monte Carlo cross-validation, a widely used resampling strategy for quantifying prediction error and uncertainty in supervised learning. The number of repetitions (1,000) was selected because training on this relatively small dataset is computationally inexpensive, and such a high number of replications yields smooth and stable estimates of the performance metrics at the two-decimal level. The observed high test-set accuracy and moderate test-set standard deviations for the best models (e.g., CatBoost) further indicate that the dataset size (317 mixtures) is sufficient for the class of learning algorithms considered in this work.

Table 2. Performance mean values of Machine Learning Models on Flexural Strength Prediction for UHPC after 1000 Monte Carlo simulations

Ranking	ML models	Training dataset				Testing dataset			
		R^2	RMSE (MPa)	MAE (MPa)	MAPE (%)	R^2	RMSE (MPa)	MAE (MPa)	MAPE (%)
1	CatBoost	0.974	1.224	0.748	4.047	0.834	3.041	2.058	11.704
2	XGB	0.981	1.028	0.328	1.636	0.818	3.173	2.052	11.400
3	RF	0.961	1.500	0.962	5.248	0.812	3.246	2.228	12.369
4	GB	0.943	1.800	1.299	7.303	0.809	3.268	2.282	13.047
5	LightGBM	0.900	2.397	1.708	9.823	0.782	3.502	2.521	14.606
6	DT	0.981	1.026	0.293	1.427	0.700	4.090	2.707	14.534
7	KNN	0.735	3.895	2.839	15.959	0.560	4.988	3.691	20.985
8	SVR	0.442	5.657	4.086	24.492	0.380	5.949	4.445	26.721
9	Linear	0.493	5.389	4.099	24.619	0.375	5.934	4.426	26.750

Table 3. Performance StD values of Machine Learning Models on Flexural Strength Prediction for UHPC after 1000 Monte Carlo simulations

ML models	Training dataset				Testing dataset			
	R^2	RMSE (MPa)	MAE (MPa)	MAPE (%)	R^2	RMSE (MPa)	MAE (MPa)	MAPE (%)
CatBoost	0.0043	0.0981	0.0523	0.0026	0.0510	0.4487	0.2120	0.0158
XGB	0.0043	0.1189	0.0510	0.0025	0.0625	0.5193	0.2448	0.0179
RF	0.0052	0.0954	0.0554	0.0030	0.0485	0.4170	0.2316	0.0160
GB	0.0059	0.0884	0.0679	0.0038	0.0582	0.4804	0.2215	0.0165
LightGB	0.0106	0.1200	0.0835	0.0052	0.0513	0.3865	0.2113	0.0164
DT	0.0043	0.1192	0.0520	0.0025	0.0898	0.5765	0.3182	0.0203
KNN	0.0198	0.1605	0.1253	0.0079	0.0761	0.4441	0.3137	0.0224
SVR	0.0247	0.1751	0.1424	0.0092	0.0535	0.4355	0.3426	0.0250
Linear	0.0413	0.2428	0.1616	0.0115	0.1421	0.6494	0.3264	0.0261

5.2. Prediction of UHPC flexural strength based on ML models

This section presents the predictive results for the four shortlisted models Gradient Boosting (GB), Random Forest (RF), Extreme Gradient Boosting (XGB) and CatBoost using the representative 70/30 train/test split that underlies Figs. 4-6. The corresponding train/test metrics are summarized in Table 4 and are used here for all numerical statements.

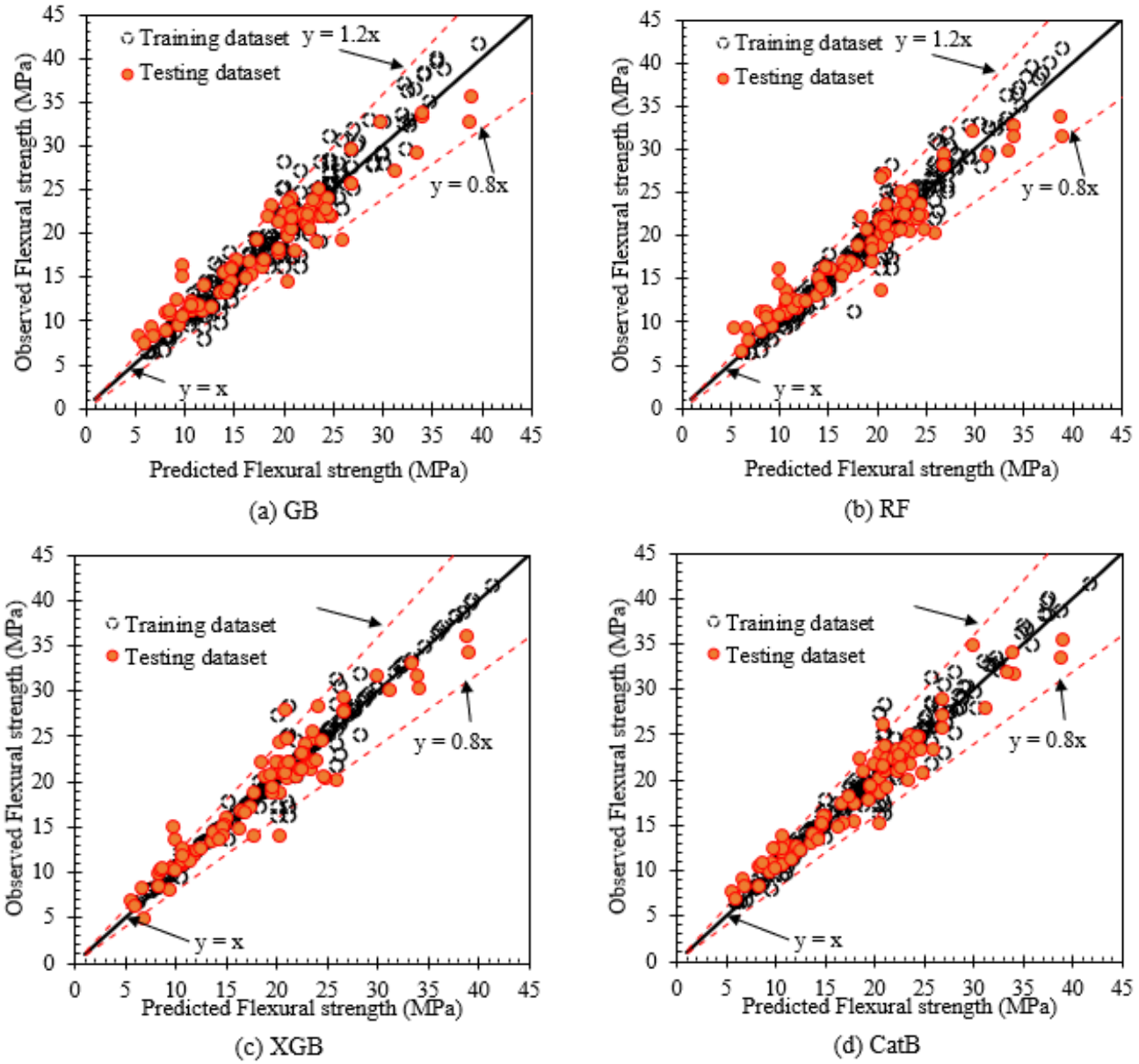


Fig. 4. Observed flexural strength vs flexural strength value predicted by (a) GB, (b) RF, (c) XGB, and (d) CatBoost

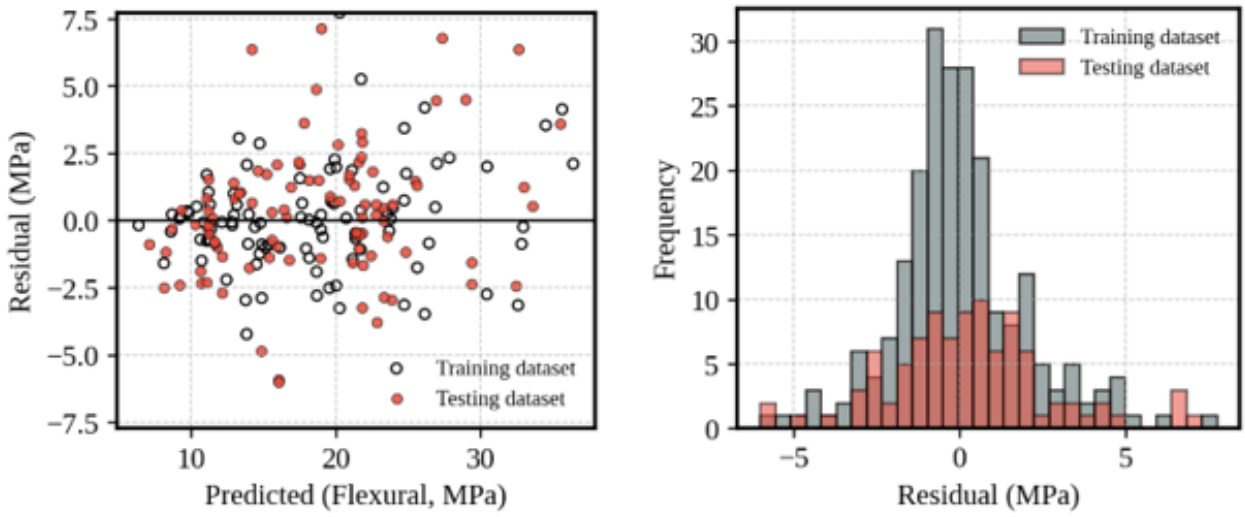
On the testing set, CatBoost delivers the best accuracy with R^2=0.928, RMSE = 1.980 MPa, MAE = 1.454 MPa, and MAPE = 8.386 %. XGB ranks second (R^2=0.903, RMSE = 2.298 MPa, MAE = 1.562 MPa, MAPE = 9.038 %). RF follows (R^2=0.885, RMSE = 2.507 MPa, MAE = 1.791 MPa, MAPE = 10.574 %), and GB is close to RF (R^2=0.891, RMSE = 2.439 MPa, MAE =1.820 MPa, MAPE = 10.951 %). On the training set, all four models achieve high goodness of fit CatBoost R^2=0.966, XGB R^2=0.974, RF R^2=0.954, GB R^2=0.933 with low errors (RMSE ≈ 1.23-1.98 MPa; MAE ≈ 0.43-1.44 MPa). The combination of high R^2 and low errors on both splits indicates that nonlinearity and interactions in the data were captured effectively.

Fig. 4 compares predicted and observed flexural strength. The solid line shows y=x (perfect agreement) and the dashed lines mark a ±20 % band. For CatBoost, points lie very close to y=x across the full range (~5-45 MPa). The testing points remain tightly packed within the ±20 % band, consistent with the small RMSE and MAPE in Table 4. XGB shows a similar pattern with slightly broader dispersion and a few underestimates at the highest strengths. RF and GB also align with

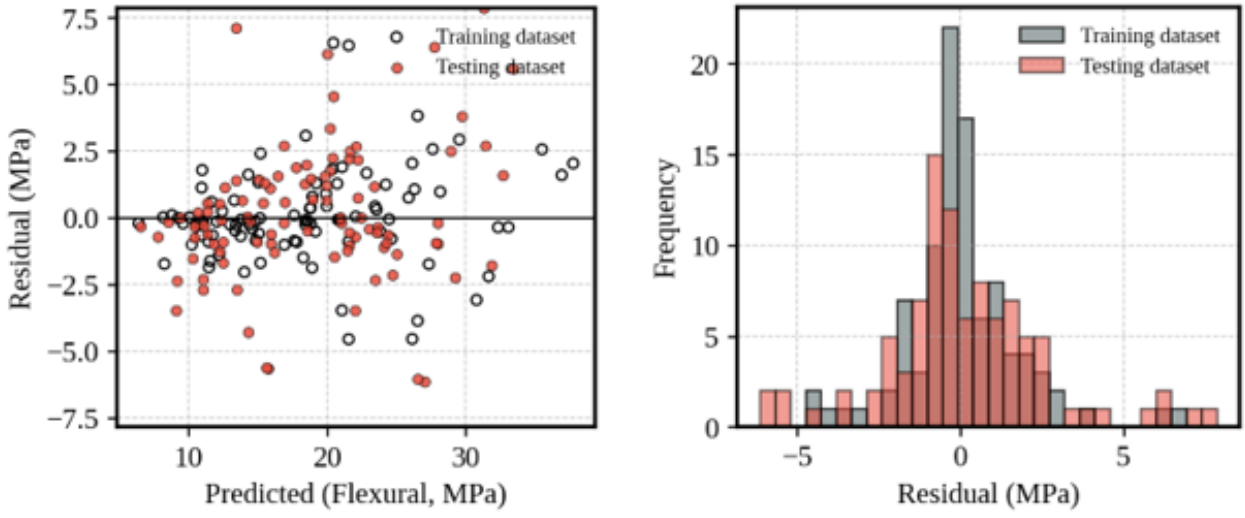
y=x, but the clouds are wider; a mild negative bias appears at the upper end, in line with their larger MAE/MAPE. The overlap between training and testing clouds is most evident for CatBoost and XGB, which matches their strong test R^2.

Residual predicted plots are centered around zero for all models, which indicates negligible global bias. The tightest residual spread occurs for CatBoost and XGB; the spread grows modestly with predicted strength for GB and RF, revealing light heteroscedasticity. Residual histograms confirm these findings: CatBoost and XGB produce narrow, near-symmetric distributions around 0 MPa, whereas RF and GB show broader histograms with slightly heavier negative tails consistent with the small underestimation seen in Fig. 4 at high strengths.

Training-testing gaps are modest for all four models. The drop in R^2 from train to test equals 0.038 for CatBoost (from 0.966 to 0.928), 0.071 for XGB (from 0.974 to 0.903), 0.069 for RF (from 0.954 to 0.885), and 0.042 for GB (0.933→0.891). In absolute terms, the RMSE increase from train to test is +0.57 MPa for CatBoost, +1.07 MPa for XGB, +0.86 MPa for RF, and +0.46 MPa for GB. Considering the typical flexural strength level of about 18-19 MPa (Table 1), test-set RMSE ≈ 2.0-2.5 MPa and MAPE ≈ 8-11 % represent practically useful accuracy for mix-design screening and performance checking.

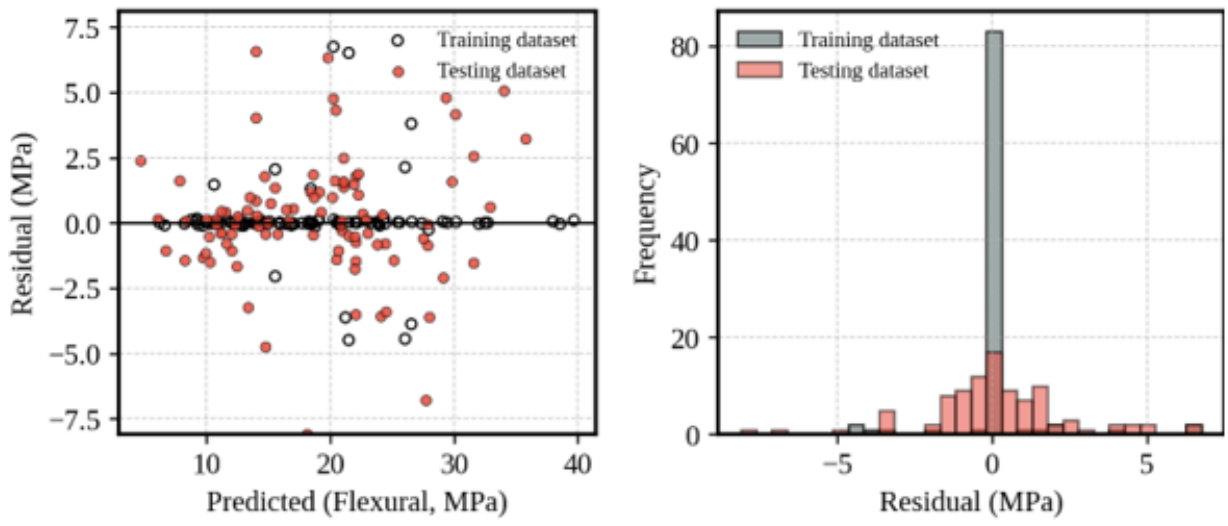


(a) GB model including Residual vs Predicted and Residual Histogram

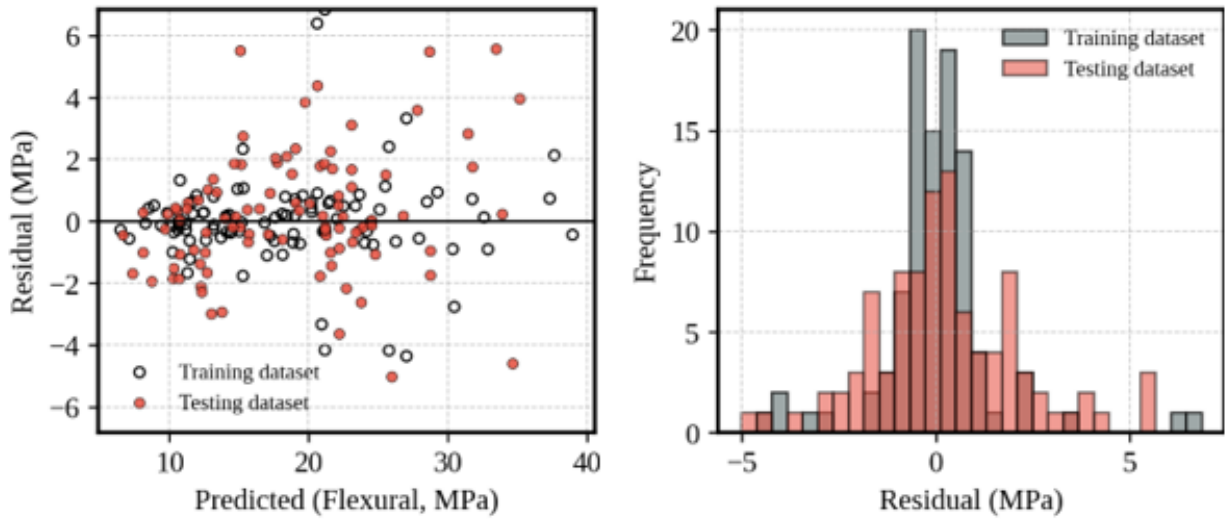


(b) RF model including Residual vs Predicted and Residual Histogram

Fig. 5. Error analysis of flexural strength predicted (a) GB model and (b) RF model



(a) XGB model including Residual vs Predicted and Residual Histogram



(b) CatB model including Residual vs Predicted and Residual Histogram

Fig. 6. Error analysis of flexural strength predicted (a) XGB model and (b) CatBoost model

Table 4. Performance value including R^2, RMSE, MAE and MAPE of ML models

Model	Training dataset				Testing dataset			
	R^2	RMSE (MPa)	MAE (MPa)	MAPE (%)	R^2	RMSE (MPa)	MAE (MPa)	MAPE (%)
CatB	0.966	1.410	0.862	4.541	0.928	1.980	1.454	8.386
XGB	0.974	1.233	0.427	2.092	0.903	2.298	1.562	9.038
RF	0.954	1.650	1.090	5.916	0.885	2.507	1.791	10.574
GB	0.933	1.983	1.438	7.895	0.891	2.439	1.820	10.951

CatBoost offers the most reliable predictions over the entire range because accuracy is high, dispersion is small, and the residual distribution is compact and symmetric. XGB is a strong alternative with very similar behavior and only a slightly larger spread. RF and GB remain viable in practice but show broader residuals and a mild high-end underestimation; these characteristics suggest that additional high-strength samples or a targeted calibration could improve performance in that region.

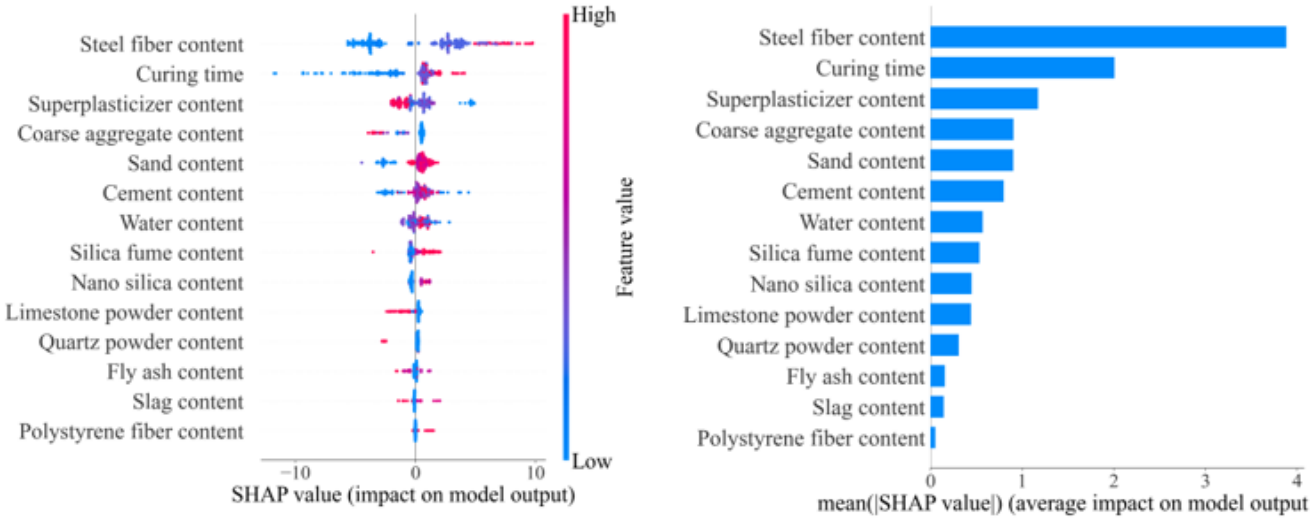
Evidence from Table 4 together with the

visual checks in Figs. 4-6 leads to a consistent ranking: CatBoost → XGB → RF/GB. CatBoost is recommended as the primary model for predicting UHPC flexural strength, with XGB as the secondary choice. RF and GB provide robust benchmarks and can serve as backups when simplicity of training or model diversity is required.

5.3. Global interpretation of four Machine Learning models

A model-agnostic view was first obtained with SHAP across the four learners (GB, RF, XGB, CatBoost) in Fig. 7. The beeswarm plots show the signed contribution of each feature (horizontal spread = effect size; color = feature value), while the right-hand bars rank average importance (mean|SHAP|). A consistent hierarchy emerged: steel fiber content and curing time dominate; cement, superplasticizer, sand, coarse aggregate, and water have medium effects; polystyrene fiber, slag, quartz powder, fly ash, and limestone powder contribute negligibly. Signs were also stable across models steel fiber and curing time positive; coarse aggregate negative; water slightly negative at high values so pruning the five weakest variables is justified without sacrificing explanatory power.

Across the four SHAP summaries, steel fiber content consistently exhibits the largest mean |SHAP|, with curing time systematically ranked second. This indicates that steel fiber content is the single most sensitive input variable for UHPC flexural strength in the present dataset, while curing time is the second most influential driver; all remaining variables exert substantially smaller average effects.



(a) GB

Fig. 7. Global SHAP interpretation of four ML models (GB, RF, XGB, CatBoost): beeswarm plots (left) and mean |SHAP| rankings (right)

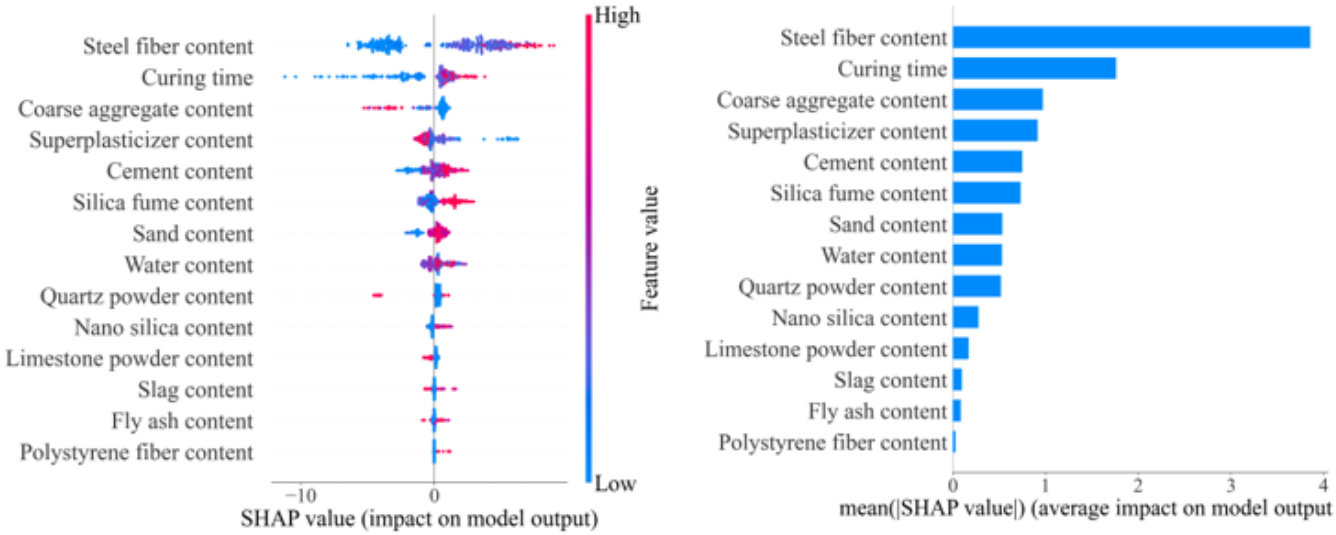
A closer reading of Fig. 7 supports these points. As steel fiber increases, points move strongly to the positive side (red to the right), forming a long right tail; this pattern appears in all four learners. Curing time shows the same positive drift with sizeable spread at very early ages, hinting at interactions with other mix variables. Coarse aggregate contributions are compact and negative across the board, indicating a robust detrimental influence for flexure within the sampled space. The mid-tier group behaves as expected: cement is modestly positive; superplasticizer is centered slightly positive but with two-sided spread (non-monotonicity); sand is weakly positive; water tilts negative at the high end. By contrast, polystyrene fiber, slag, quartz powder, fly ash, and limestone powder cluster around zero with very short bars.

Building on the global picture, Fig. 8 reports local response shapes for the nine retained variables using CatBoost with SHAP-PDP overlays and 95% prediction bands. Steel fiber displays a steep gain from 0 → ≈2-3 % with clear saturation

thereafter; residual uncertainty is small in the core range, indicating reliable gains. Curing time yields large early benefits to ~28-56 days, followed by diminishing returns; bands widen only at very long times due to scarce data. Coarse aggregate shows a monotonic negative profile with a narrow band, confirming a dependable penalization on flexural strength. Cement rises gently (positive but shallow), while silica fume turns beneficial once ≈100 kg/m³ is reached. Superplasticizer is non-monotonic a moderate window is favorable, but very high dosages drift negative consistent with segregation risks. Water has a broad “safe” zone around ~170-190 kg/m³ and becomes increasingly negative above ~190-200 kg/m³. Sand is weakly positive around ~900-1200 kg/m³. Limestone powder remains near-neutral with a slight negative drift at the high end. Vertical scatter at fixed x-values (notably for fiber and curing) indicates interactions (fiber×curing, fiber×superplasticizer), while band widening at extreme dosages marks regions where more experiments would reduce uncertainty.

Taken together, these SHAP–PDP curves confirm that changes in steel fiber volume fraction have the strongest marginal impact on the predicted flexural strength, followed by curing time, whereas the other inputs act as secondary modifiers or penalties (notably coarse aggregate).

5.4. CatBoost GUI implements



(b) RF

Fig. 7. (continued)

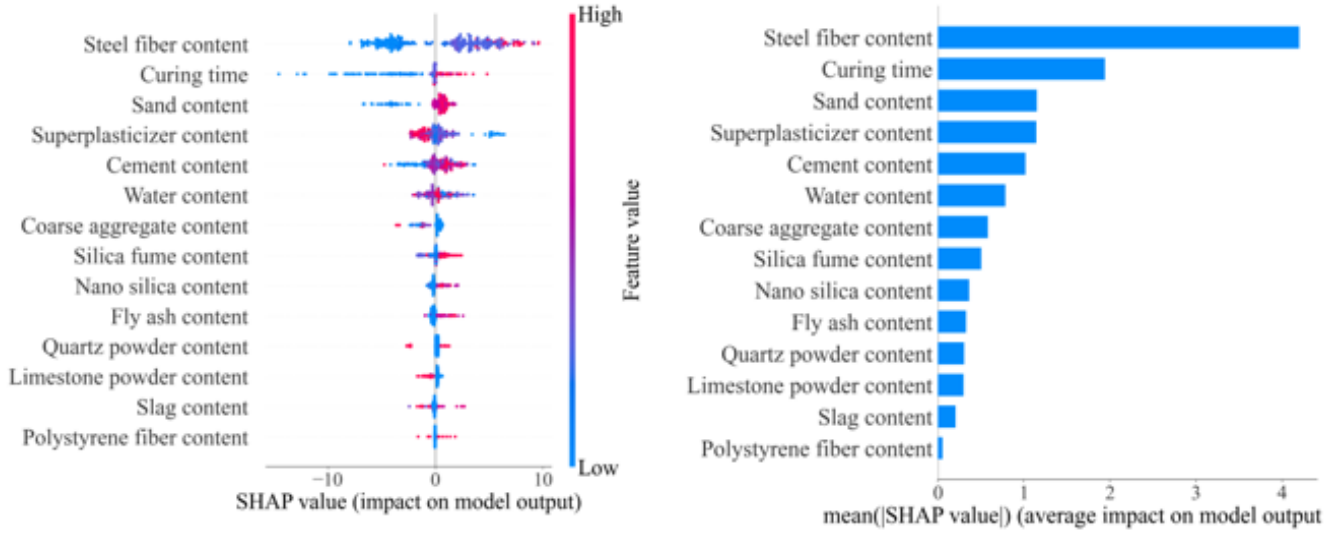
The CatBoost GUI implements the nine-input model for immediate engineering use. Inputs are limited to variables that are routinely measured; outputs include the predicted flexural strength and transparent quality indicators (R^2, RMSE, MAE, MAPE for train/test). On the independent test split, performance remains R^2 = 0.916, RMSE = 2.143 MPa, MAE = 1.522 MPa, MAPE = 9.07% a small trade-off relative to the 14-variable baseline. In practice, the GUI enables rapid what-if exploration (fiber, curing, water, superplasticizer, etc.), reduces measurement burden by 36% (14 → 9 inputs), and shortens the design-trial loop. This is the critical step from analysis to deployable decision support for UHPC mix design. The GUI (python code) and data are available at https://github.com/vanquanTRAN/UHPC-Flexural-strength.

To complement the accuracy and stability analysis, the computational cost of the main models was also measured. All experiments were carried out on a standard workstation equipped with an Intel® Core™ i7-6820HQ CPU @ 2.70 GHz, 32 GB RAM, a 238 GB SSD, and an NVIDIA Quadro M1000M (4 GB), running a 64-bit operating system. No GPU acceleration was used for training.

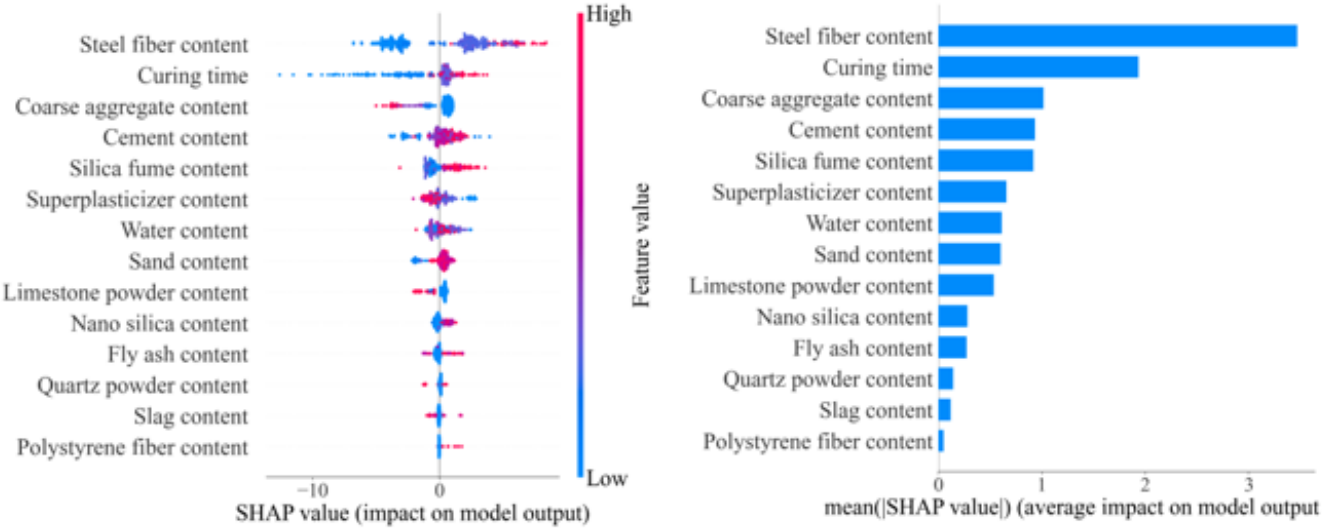
For a single 70/30 train–test experiment (one

Monte Carlo repetition), the wall-clock and CPU times required to train and evaluate the four most competitive ensemble models were as follows: CatBoost (true time wall = 1.555111 s, CPU time = 4.859375 s), Gradient Boosting (true time wall = 0.139806 s, CPU time = 0.078125 s), Random Forest (true time wall = 0.187439 s, CPU time = 0.187500 s), and XGBoost (true time wall = 0.119021 s, CPU time = 0.828125 s). The remaining baselines (linear regression, SVR, KNN, and a single decision tree) trained essentially instantaneously on the 317-sample UHPC dataset and were not computationally limiting.

Extrapolating these figures to the full Monte Carlo scheme with 1,000 random 70/30 splits shows that the total offline training and evaluation cost remains moderate: on this CPU-only machine, the most expensive model (CatBoost) requires only a few tens of minutes of wall-clock time, while the other ensembles complete in a few minutes. Once a single model has been selected and fitted (CatBoost in this work), however, the cost of computing the flexural strength for a new UHPC mixture is negligible: a single forward pass through the trained model takes well below one millisecond on the same hardware. Consequently, the proposed model can be embedded in the accompanying GUI and used in an interactive design setting without any perceptible delay for the end user.



(c) XGB



(d) CatB

Fig. 7. (continued)

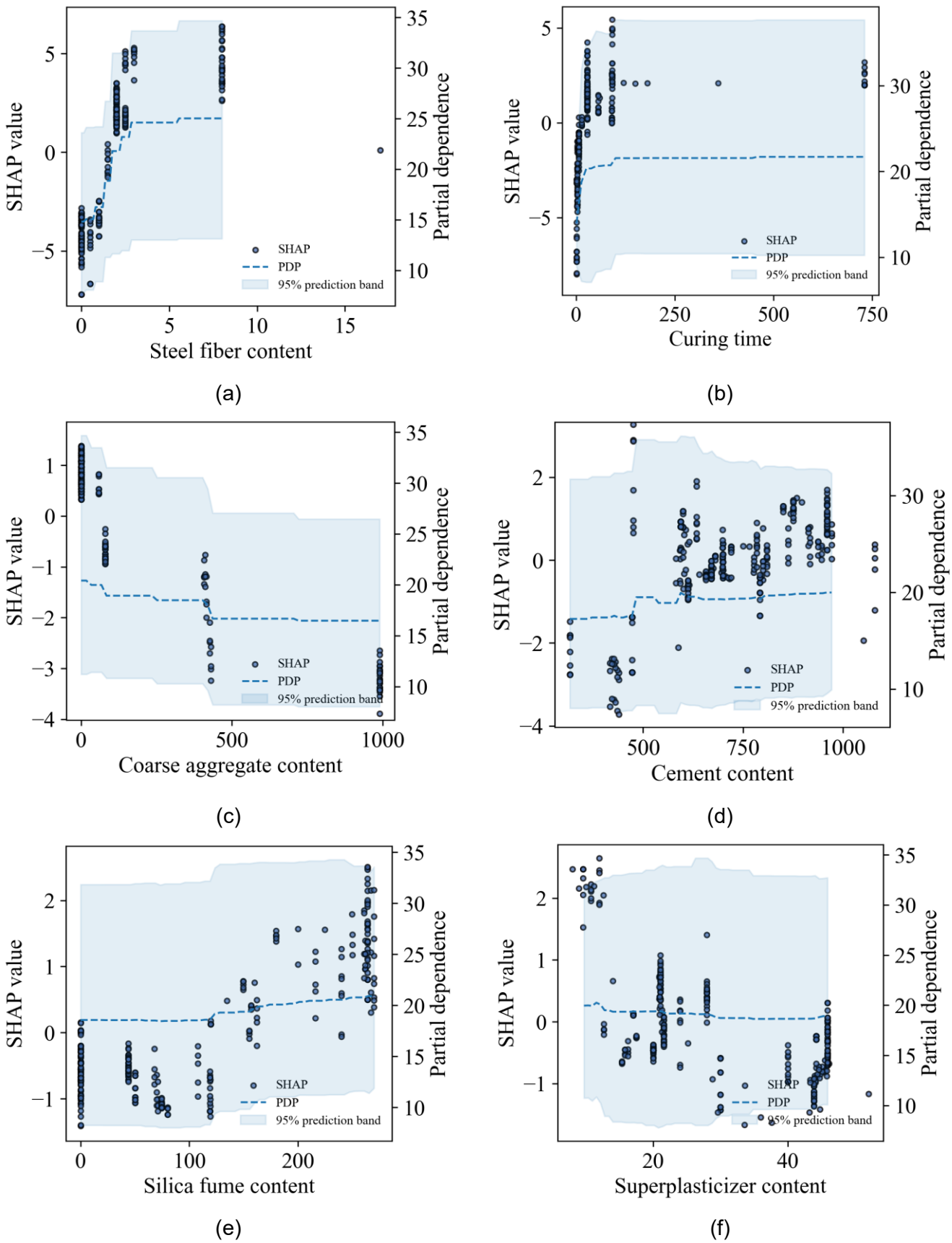
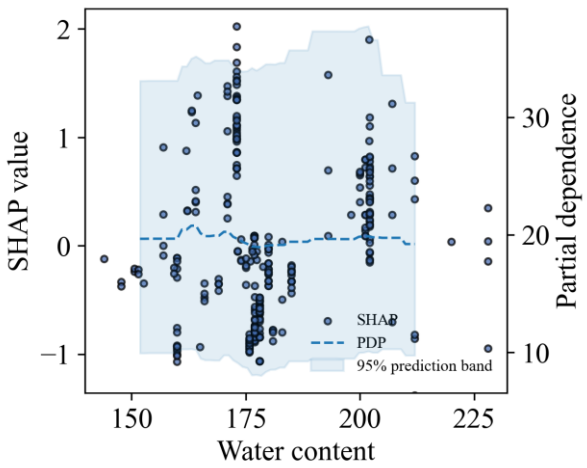
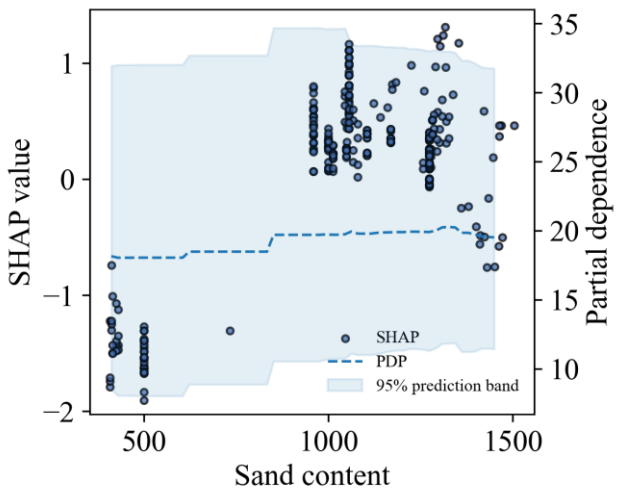


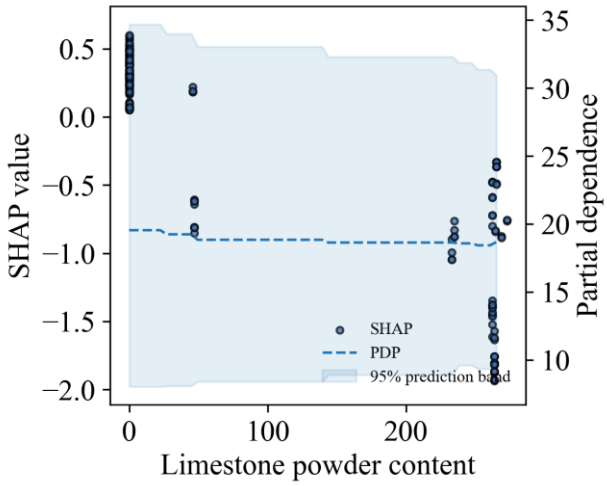
Fig. 8. SHAP-PDP analysis (with 95% prediction band) for the nine retained variables of the CatBoost model: (a) Steel fiber, (b) Curing time, (c) Coarse aggregate, (d) Cement, (e) Silica fume, (f) Superplasticizer, (g) Water, (h) Sand, (i) Limestone powder



(g)



(h)



(i)

Fig. 8. (continued)

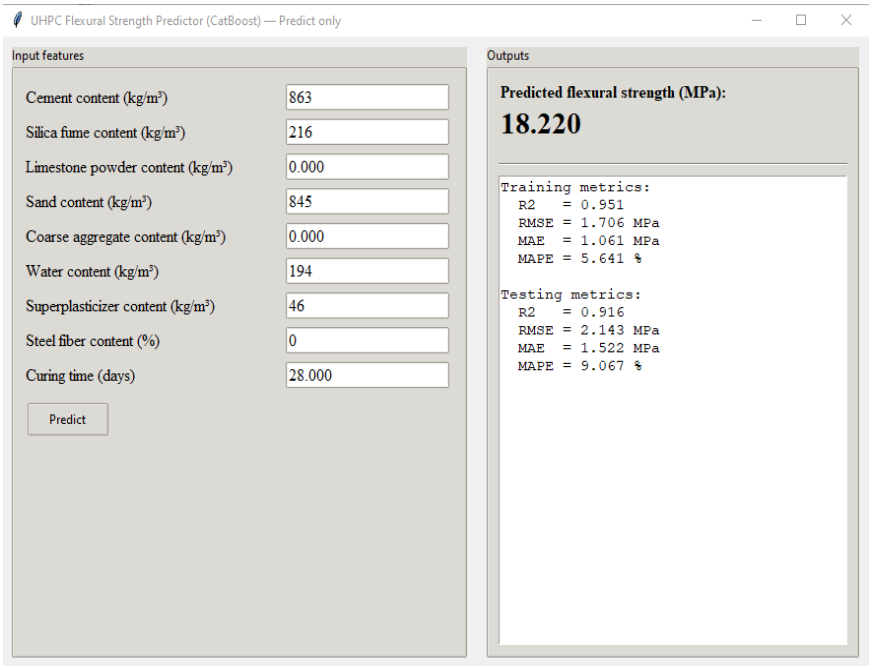


Fig. 9. GUI for Catboost model prediction of UHPC flexural strength

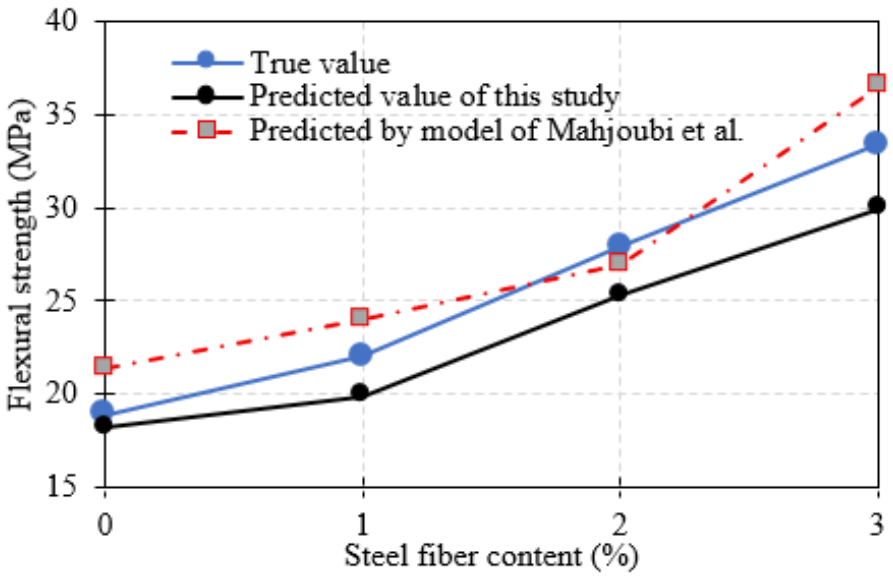


Fig. 10. Comparison of UHPC flexural strength versus steel fiber content: CatBoost (9 inputs) vs. true values [49] and Mahjoubi et al. [26]

The dataset of Wu et al. [49] is external to model training and is used solely for out-of-sample verification. Across 0-2% steel fiber, CatBoost tracks the true values closely and preserves the correct nonlinear slope; a mild underprediction appears near ~3%. In contrast, the Mahjoubi et al. model [26] overpredicts at higher fiber contents. This external comparison confirms that the nine-input CatBoost generalizes beyond the training corpus and better preserves the mechanics-consistent shape of the fiber-strength relation in the practical operating range.

Figs. 7-10 jointly prescribe a clear recipe: prioritize steel fiber (≈0-2.5/3%) and adequate curing (≥28 days); limit coarse aggregate for flexure-critical targets; tune the matrix with cement and silica fume while holding water and superplasticizer in stable windows; treat sand and limestone powder as secondary knobs. The nine-input CatBoost remains both accurate and interpretable, and through the GUI actionable for engineers. Where needed (very long curing or extreme dosages), targeted data collection will narrow the remaining uncertainty.

6. Conclusions

The study established an interpretable machine-learning pipeline for predicting UHPC flexural strength. Across nine candidate algorithms evaluated under 1,000 Monte-Carlo resamplings, tree-based boosting consistently prevailed; CatBoost delivered the strongest generalization with a representative test split of R^2=0.928, RMSE=1.980 MPa, MAE=1.454 MPa, and MAPE=8.386%, while the resampling average yielded R^2=0.834.

SHAP/PDP analyses revealed a stable hierarchy of effects: steel fiber content and curing time were strongly positive, coarse aggregate was monotonically detrimental, and water and superplasticizer exhibited dosage windows. Guided by these diagnostics, the input set was reduced from 14 to 9 variables with only a small accuracy trade-off; the deployed GUI based on the 9-input CatBoost maintained R^2test=0.916 and enabled rapid “what-if” assessments for mix design. Out-of-sample verification confirmed that the learned response preserved the mechanics-consistent nonlinear trend with fiber dosage.

The dataset is modest and heterogeneous with incomplete metadata (fiber geometry/rheology/curing), leaving residual confounding and extrapolation risk; SHAP/PDP may also mislead under collinearity or sparsity. The present modeling targets only flexural strength and does not embed explicit physics constraints. Future work will expand and standardize multi-lab

datasets, add calibrated uncertainty and drift detection to the GUI, integrate physics-guided priors and multi-objective optimization (strength-workability-cost-carbon), and extend to multi-task prediction with external blind validation.

Conflict of Interest: The authors declare that there is no conflict of interest.

Funding: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Availability of data and material: Data will be made available on request.

References

[1] C. Shi, Z. Wu, J. Xiao, D. Wang, Z. Huang, Z. Fang. (2015). A review on ultra high performance concrete: Part I. Raw materials and mixture design. *Construction and Building Materials*, 101 741–751. https://doi.org/10.1016/j.conbuildmat.2015.10.088

[2] Z. Wu, C. Shi, K.H. Khayat. (2019). Investigation of mechanical properties and shrinkage of ultra-high performance concrete: Influence of steel fiber content and shape. *Composites Part B: Engineering*, 174, 107021. https://doi.org/10.1016/j.compositesb.2019.107021

[3] Z. Wu, C. Shi, W. He, L. Wu. (2016). Effects of steel fiber content and shape on mechanical properties of ultra high performance concrete. *Construction and Building Materials*, 103, 8–14. https://doi.org/10.1016/j.conbuildmat.2015.11.028

[4] R. Yu, P. Spiesz, H.J.H. Brouwers. (2015). Development of an eco-friendly Ultra-High Performance Concrete (UHPC) with efficient cement and mineral admixtures uses. *Cement and Concrete Composites*, 55, 383–394. https://doi.org/10.1016/j.cemconcomp.2014.09.024

[5] K.H. Khayat, W. Meng, K. Vallurupalli, L. Teng. (2019). Rheological properties of ultra-high-performance concrete — An overview. *Cement and Concrete Research*, 124, 105828. https://doi.org/10.1016/j.cemconres.2019.105828

[6] D.-Y. Yoo, N. Banthia. (2016). Mechanical properties of ultra-high-performance fiber-reinforced concrete: A review. *Cement and Concrete Composites*, 73, 267–280. https://doi.org/10.1016/j.cemconcomp.2016.08.001

[7] Z. Wu, C. Shi, W. He, L. Wu. (2016). Effects of steel fiber content and shape on mechanical properties of ultra high performance concrete. *Construction and Building Materials*, 103, 8–14. https://doi.org/10.1016/j.conbuildmat.2015.11.028

[8] G. James, D. Witten, T. Hastie, R. Tibshirani. (2021). An Introduction to Statistical Learning: with Applications in R. *Springer New York, NY*. https://doi.org/10.1007/978-1-0716-1418-1

[9] A. Benzaamia, M. Ghrici, R. Rbouh, A.A. Ghrici, P.G. Asteris. (2025). Prediction of Chloride Resistance Level in Concrete Using Optimized Tree-Based Machine Learning Models. *Bulletin of Computational Intelligence*, 1(1), 104–117. https://doi.org/10.53941/bci.2025.100007

[10] P.G. Asteris. (2025). Computational Intelligence: From Nature and Aristotle to Meta-Heuristic Algorithms. *Bulletin of Computational Intelligence*, 1(1), 1–2. https://doi.org/10.53941/bci.2025.100001

[11] V.Q. Tran, H.-V.T. Mai, Q.T. To, M.H. Nguyen. (2023). Machine learning approach in investigating carbonation depth of concrete containing Fly ash. *Structural Concrete*, 24(2), 2145-2169. https://doi.org/10.1002/suco.202200269

[12] V.Q. Tran. (2022). Machine learning approach for investigating chloride diffusion coefficient of concrete containing supplementary cementitious materials. *Construction and Building Materials*, 328, 127103. https://doi.org/10.1016/j.conbuildmat.2022.127103

[13] S.M. Lundberg et al. (2020). From local

explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2, 56-67. https://doi.org/10.1038/s42256-019-0138-9

[14] Partial Dependence and Individual Conditional Expectation Plots, Scikit-Learn (n.d.). https://scikit-learn/stable/auto_examples/inspection/plot_partial_dependence.html (accessed October 13, 2021).

[15] Y. Qian, M. Sufian, A. Hakamy, A. Farouk Deifalla, A. El-said. (2023). Application of machine learning algorithms to evaluate the influence of various parameters on the flexural strength of ultra-high-performance concrete. *Frontiers in Materials*, 9, 1114510. https://doi.org/10.3389/fmats.2022.1114510

[16] N.-H. Nguyen, J. Abellán-García, S. Lee, T.P. Vo. (2024). From machine learning to semi-empirical formulas for estimating compressive strength of Ultra-High Performance Concrete. *Expert Systems with Applications*, 237, 121456.https://doi.org/10.1016/j.eswa.2023.121456

[17] A. Kashem, R. Karim, S.C. Malo, P. Das, S.D. Datta, M. Alharthai. (2024). Hybrid data-driven approaches to predicting the compressive strength of ultra-high-performance concrete using SHAP and PDP analyses. *Case Studies in Construction Materials*, 20, e02991. https://doi.org/10.1016/j.cscm.2024.e02991

[18] P. Das, A. Kashem. (2024). Hybrid machine learning approach to prediction of the compressive and flexural strengths of UHPC and parametric analysis with shapley additive explanations. *Case Studies in Construction Materials*, 20, e02723. https://doi.org/10.1016/j.cscm.2023.e02723

[19] J. Abellán-García. (2020). Four-layer perceptron approach for strength prediction of UHPC. *Construction and Building Materials*, 256, 119465. https://doi.org/10.1016/j.conbuildmat.2020.119465.

[20] G. Sun, M. Du, B. Shan, J. Shi, Y. Qu. (2022). Ultra-high performance concrete design method based on machine learning model and steel slag powder. *Case Studies in Construction Materials*, 17, e01682. https://doi.org/10.1016/j.cscm.2022.e01682

[21] B.K. Aylas-Paredes, T. Han, A. Neithalath, J. Huang, A. Goel, A. Kumar, N. Neithalath. (2025). Data driven design of ultra high performance concrete prospects and application. *Scientific Reports*, 15, 9248. https://doi.org/10.1038/s41598-025-94484-2

[22] Y. Yuan, M. Yang, X. Shang, Y. Xiong, Y. Zhang. (2023). Predicting the compressive strength of UHPC with coarse aggregates in the context of machine learning. *Case Studies in Construction Materials*, 19, e02627. https://doi.org/10.1016/j.cscm.2023.e02627

[23] M. Katlav, F. Ergen. (2025). Improved forecasting of the compressive strength of ultra-high-performance concrete (UHPC) via the CatBoost model optimized with different algorithms. *Structural Concrete*, 26, 212–235. https://doi.org/10.1002/suco.202400163

[24] M.I. Khan, Y.M. Abbas, G. Fares, F.K. Alqahtani. (2023). Strength prediction and optimization for ultrahigh-performance concrete with low-carbon cementitious materials – XG boost model and experimental validation. *Construction and Building Materials*, 387, 131606. https://doi.org/10.1016/j.conbuildmat.2023.131606

[25] P.P. Li, Q.L. Yu, H.J.H. Brouwers. (2018). Effect of coarse basalt aggregates on the properties of Ultra-high Performance Concrete (UHPC). *Construction and Building Materials*, 170, 649–659. https://doi.org/10.1016/j.conbuildmat.2018.03.109

[26] S. Mahjoubi, W. Meng, Y. Bao. (2022). Auto-tune learning framework for prediction of flowability, mechanical properties, and porosity

of ultra-high-performance concrete (UHPC). *Applied Soft Computing*, 115, 108182. https://doi.org/10.1016/j.asoc.2021.108182

[27] S. Abbas, A.M. Soliman, M.L. Nehdi. (2015). Exploring mechanical and durability properties of ultra-high performance concrete incorporating various steel fiber lengths and dosages. *Construction and Building Materials*, 75, 429–441. https://doi.org/10.1016/j.conbuildmat.2014.11.017

[28] P.R. Prem, A.R. Murthy, B.H. Bharatkumar. (2015). Influence of curing regime and steel fibres on the mechanical properties of UHPC. *Magazine of Concrete Research*, 67, 988–1002. https://doi.org/10.1680/macr.14.00333

[29] M.I. Khan, Y.M. Abbas, G. Fares, F.K. Alqahtani. (2023). Strength prediction and optimization for ultrahigh-performance concrete with low-carbon cementitious materials – XG boost model and experimental validation. *Construction and Building Materials*, 387, 131606. https://doi.org/10.1016/j.conbuildmat.2023.131606

[30] R. Kohavi. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Vol 2, Morgan Kaufmann Publishers Inc., USA, pp. 1137–1143.*

[31] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl. (2011). Algorithms for hyper-parameter optimization. *Proceedings of the 25th International Conference on Neural Information Processing Systems, Curran Associates Inc., USA, pp. 2546–2554.*

[32] R. Kumar, H.-V.T. Mai. (2022) Prediction and sensitivity analysis of self compacting concrete slump flow by random forest algorithm. *Journal of Science and Transport Technology*, 2(1), 31–41. https://doi.org/10.58845/jstt.utt.2022.en.2.1.31-41

[33] A.-T. Tran, T.-H. Le, M.H. Nguyen. (2022). Forecast of surface chloride concentration of concrete utilizing ensemble decision tree boosted. *Journal of Science and Transport Technology*, 2(1), 42–53. https://doi.org/10.58845/jstt.utt.2022.en.2.1.42-53

[34] M. Hadzima-Nyarko, S.H. Trinh. (2022). Prediction of compressive strength of concrete at high heating conditions by using artificial neural network-based Bayesian regularization. *Journal of Science and Transport Technology*, 2(1), 9–20. https://doi.org/10.58845/jstt.utt.2022.en.2.1.9-20

[35] D.Q. Vu, D.D. Nguyen, Q.-A.T. Bui, D.K. Trong, I. Prakash, B.T. Pham. (2021). Estimation of California Bearing Ratio of Soils Using Random Forest based Machine Learning. *Journal of Science and Transport Technology*, 1(1) 45–58. https://doi.org/10.58845/jstt.utt.2021.en.1.1.45-58

[36] V.Q. Tran, H.Q. Do. (2021). Prediction of California Bearing Ratio (CBR) of Stabilized Expansive Soils with Agricultural and Industrial Waste Using Light Gradient Boosting Machine. *Journal of Science and Transport Technology*, 1(1), 1–8. https://doi.org/10.58845/jstt.utt.2021.en.1.1.1-8

[37] C. Cortes, V. Vapnik. (1995). Support-vector networks. *Machine Learning*, 20, 273–297.

[38] N.S. Altman. (1991). BU-1065MA An Introduction to Kernel and Nearest Neighbor Nonparametric Regression.

[39] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems 30 (NIPS 2017), Curran Associates, Inc.*

[40] J.H. Friedman. (2001). Greedy function approximation: A gradient boosting machine.

The Annals of Statistics, 29, 1189–1232. https://doi.org/10.1214/aos/1013203451

[41] L. Breiman. (2001). Random Forests. *Machine Learning*, 45, 5–32. https://doi.org/10.1023/A:1010933404324

[42] A.V. Dorogush, V. Ershov, A. Gulin. (2018). CatBoost: gradient boosting with categorical features support. *arXiv*. http://arxiv.org/abs/1810.11363

[43] T. Chen, C. Guestrin. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, USA, pp. 785–794.* https://doi.org/10.1145/2939672.2939785.

[44] J.R. Quinlan. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106. https://doi.org/10.1007/BF00116251

[45] Q.H. Nguyen, H.-B. Ly, L.S. Ho, N. Al-Ansari, H.V. Le, V.Q. Tran, I. Prakash, B.T. Pham. (2021). Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil. *Mathematical Problems in Engineering*, 2021(1), 4832864. https://doi.org/10.1155/2021/4832864

[46] V.Q. Tran, V.Q. Dang, L.S. Ho. (2022). Evaluating compressive strength of concrete made with recycled concrete aggregates using machine learning approach. *Construction and Building Materials*, 323, 126578. https://doi.org/10.1016/j.conbuildmat.2022.126578

[47] V.Q. Tran. (2022). Hybrid gradient boosting with meta-heuristic algorithms prediction of unconfined compressive strength of stabilized soil based on initial soil properties, mix design and effective compaction. *Journal of Cleaner Production*, 355, 131683. https://doi.org/10.1016/j.jclepro.2022.131683

[48] Q.-S. Xu, Y.-Z. Liang. (2001). Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, 56(1), 1–11. https://doi.org/10.1016/S0169-7439(00)00122-2

[49] Z. Wu, C. Shi, K.H. Khayat. (2019). Investigation of mechanical properties and shrinkage of ultra-high performance concrete: Influence of steel fiber content and shape. *Composites Part B: Engineering,* 174, 107021. https://doi.org/10.1016/j.compositesb.2019.107021