BER the international journal of bridge engineering management and research

Prestressed UHPC Beam Flexural Capacity Prediction Improvement with Constraint-Guided Neural Networks

Hailong Qu[1]; Le Teng[2]; Yanping Zhu[3,*]; Zhongke Qu[4]; and Hantao Wu[4]

Submitted: 31 March 2025 **Accepted:** 11 June 2025 **Publication date:** 10 July 2025

DOI: 10.70465/ber.v2i3.31

Abstract: This study develops a constraint-guided neural network (CGNN) approach to address prediction accuracy degradation in machine learning models when tested on unseen data after training with limited datasets, while simultaneously enhancing the generalization capabilities of conventional mechanical equations. The proposed model predicts flexural capacity of prestressed ultrahigh-performance concrete (UHPC) beams using a small experimental dataset collected from literature. Physical constraints incorporated into the model include geometrical relationships, reinforcement ratio limitations, and strength correlations. Comprehensive performance comparisons demonstrate that the CGNN model outperforms traditional machine learning methods—including back propagation neural networks, categorical boosting, extreme gradient boosting, random forests, and decision trees—as well as conventional predictive equations. The CGNN exhibits superior performance across multiple metrics: mean ratio, coefficient of variation, mean relative error, coefficient of determination, maximum ratio, and minimum ratio, while maintaining consistent accuracy between training and testing phases. Model interpretability is achieved through SHAP (SHapley Additive exPlanations), and a user-friendly graphical user interface has been developed to facilitate adoption by UHPC design practitioners.

Author keywords: Constraint-guided neural networks; UHPC; prestressed beams; flexural capacity; machine learning

Introduction

Ultrahigh-performance concrete (UHPC) material, developed by the particle packing method and mixed with steel fibers, shows dense microstructure, high compressive strength of over 120 MPa and strain hardening behavior in tension.[1–5] With these excellent mechanical and durability properties, UHPC has been an alternative for solving civil infrastructure challenges, such as resilience and sustainability, in the context of climate change.[6,7,8–10] To promote its application in engineering, experimental investigation and validation is a normal routine to check structural load-bearing capacity, stiffness and stability,[11,12] although analytical equations and numerical simulations are developed and considered effective to save time and cost.[13–15] Steel-reinforced UHPC beams with different variables have been widely tested for response evaluation and comparison because the bending moment is frequently encountered in buildings and bridges.[16–19] However, the UHPC beams prestressed with steel strands are not tested comparably with the UHPC beams without strands; therefore, they have limited strength data available in the literature.[20–22] Some provisions specify varying equations for predicting flexural strength of reinforced UHPC beams and the equations are established from strain compatibility, stress–strain relationship and forces equilibrium.[23–25] Despite various studies on the mechanical properties of UHPC beams,[26] experimental data for prestressed UHPC beams remains limited, posing challenges for traditional prediction methods that rely on large datasets. Meanwhile, existing prediction approaches, primarily based on simplified mechanical equations, struggle to fully capture the complex nonlinear behavior characteristics of prestressed UHPC beams. Therefore, there is an urgent need for a prediction method that can function effectively with limited data while integrating the advantages of both physical knowledge and data-driven approaches. Recently emerging physics-informed machine learning methods offer new solutions to this problem, particularly approaches that combine physical constraints with neural networks, which show promise for achieving high-precision predictions under small-sample data conditions. Therefore, it is of importance and interest to develop other flexural strength prediction methods.

Physics-informed neural networks (PINNs) were first proposed by Raissi et al.,[27] and they integrate domain-specific knowledge as informative priors or inductive biases

*Corresponding Author: Yanping Zhu. Email: yzhu@mtech.edu
[1]School of Civil Engineering, Chongqing University, Chongqing, 400045, China
[2]Professor, School of Materials Science and Engineering, Southeast University, Nanjing, 211189, Jiangsu, China
[3]Assistant Professor, Civil Engineering Department, Montana Technological University, Butte, 59701, Montana, USA
[4]School of Human Settlements and Civil Engineering, Xi'an Jiaotong University, Xi'an, 710049, China

Discussion period open till six months from the publication date. Please submit separate discussion for each individual paper. This paper is a part of the Vol. 2 of the International Journal of Bridge Engineering, Management and Research (© BER), ISSN 3065-0569.

to constrain the neural network outputs, ensuring consistency with physical laws and improving both generalization and interpretability.[28,29] In PINNs, the differences in the training process arise from the addition of physical constraints to the loss functions during iterations. The prior knowledge includes physical, empirical, observational, and mathematical insights of the data. The prediction accuracy of most machine learning techniques depends on the quality and quantity of the training data.[30] Embedding theoretical constraints in PINN may improve prediction accuracy and generalization ability, especially with small observational datasets where overfitting usually happens. PINNs provide additional supervision information from these constraints to reduce the model's dependence on large datasets for accuracy and avoid overfitting in small datasets. Rahman et al.[31] developed a PINN to predict early-age cement hydration kinetics, and the model achieves exceptional accuracy and convergence with only 5% of the training data. The excellent generalization ability is also confirmed by Wan et al.[32] in predicting the chloride diffusion coefficient in concrete using PINN with 5% and 10% noise data as observed data. Despite these promising applications, existing PINN implementations for concrete structures often focus on specific material properties rather than structural performance prediction. Furthermore, there remains limited exploration of how different types of physical constraints affect model performance in structural engineering applications. The embedding of theoretical constraints in PINNs shows potential for improving prediction with small datasets, but the optimal selection and weighting of these constraints for prestressed UHPC beam applications has not been systematically investigated.

Inspired by the combined loss function in the PINN, this study develops a constraint-guided neural network (CGNN) to predict the flexural capacity of prestressed UHPC beams with a small dataset. The data is acquired from past experiments in the literature, and the dataset is small due to limited, highly costly, labor-intensive, and time-consuming tests. The main constraints in the CGNN include geometry relationships, reinforcement ratio limitations, and the relationship between tensile and compressive strength. These constraints stem from observations and empirical insights of the dataset. The prediction accuracy and advantage of the CGNN are highlighted through comparison with conventional machine learning models (BPNN, categorical boosting [CatBoost], extreme gradient boosting [XGBoost], random forest [RF], and decision tree [DT]) as well as model equations. The impact of the constraints on the CGNN prediction is further discussed. With model interpretability from SHAP (SHapley Additive exPlanations), a graphical user interface (GUI) is developed for use.

Machine Learning Overview

Back propagation neural network (BPNN)

BPNN demonstrates excellent predictive performance among classic machine learning algorithms. In Fig. 1a, BPNN consists of an input layer, hidden layer, and output layer.[33] Neurons in each layer transmit information through weighted connections, using the backpropagation algorithm to update network parameters. BPNN calculates output through forward propagation, and then adjusts network weights and biases through backward propagation of gradients. After training, BPNN forms a nonlinear mapping model and can be used to fit complex continuous functions.

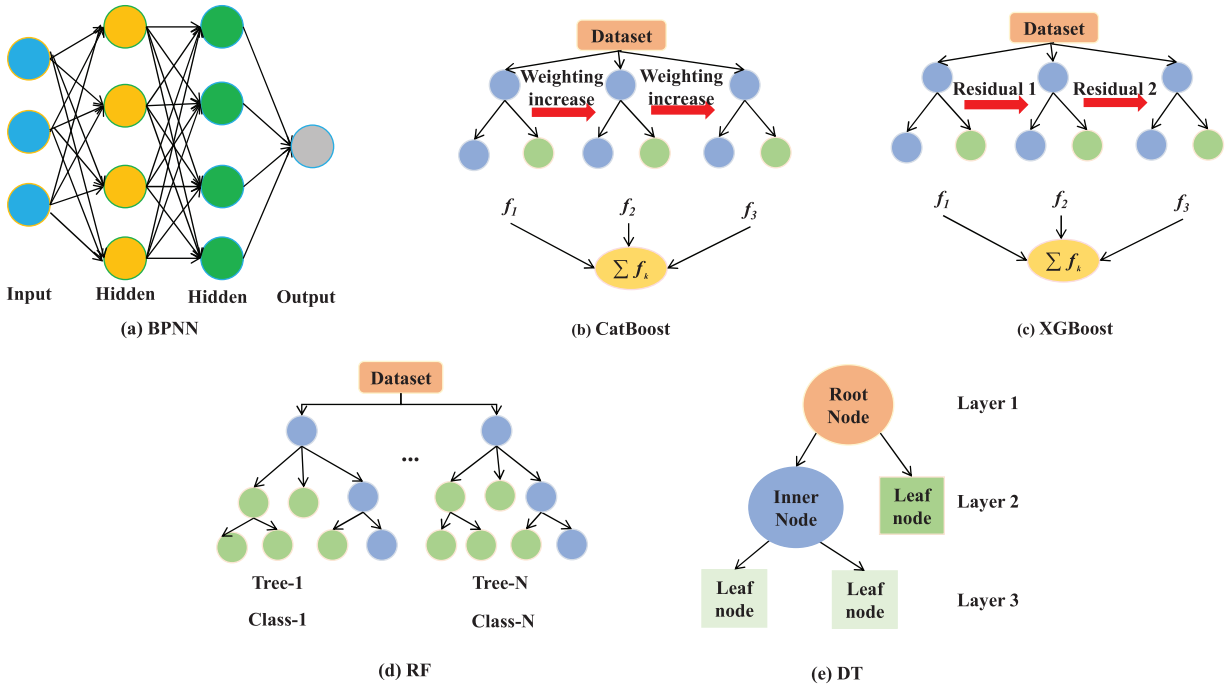


Figure 1. Architecture of machine learning models

During forward propagation, the output of each neuron is calculated by the activation function as follows:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{1}$$

where x_i represents input, w_i represents weight, b represents bias term, f is the activation function, n represents the number of input variables, and y represents the neuron's output.

CatBoost

In Fig. 1b, CatBoost is a gradient boosting-based ensemble learning method that specifically optimizes the handling of categorical features and gradient bias issues.[34] Each weak learner is based on DTs trained through a special ordered boosting scheme. CatBoost learns the relationship between input and output data by minimizing an objective function with regularization. After training, CatBoost forms an efficient and robust additive model and can handle complex classification or regression problems. The prediction function at the mth iteration of the model is expressed as

$$F_m(x) = F_{m-1}(x) + \alpha_m h_m(x) \tag{2}$$

where $F_m(x)$ represents the model's predicted value at the mth iteration, $h_m(x)$ represents the prediction value of the mth tree, and α_m is the learning rate.

XGBoost

In Fig. 1c, XGBoost is a gradient boosting tree-based ensemble learning method that forms a strong learner by sequentially constructing multiple weak learners.[35] Each weak learner focuses on correcting the residuals of previous models. XGBoost learns the relationship between input and output data by minimizing a regularized objective function. After training, XGBoost forms an efficient additive model and can handle complex classification or regression problems. The model's objective function consists of loss and regularization terms as follows:

$$Obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{k=1}^{K} \Omega\left(f_k\right) \tag{3}$$

where $l\left(y_i, \hat{y}_i\right)$ represents the loss function, used to measure the difference between the predicted value $\hat{y}_i$ and the true value y_i, $\Omega\left(f_k\right)$ represents the complexity penalty term of the kth tree, n represents the number of samples, and K represents the number of trees.

RFs

In Fig. 1d, RF consists of multiple DTs, each constructed through random sampling of training data and features.[36] Each DT is an independently trained classifier or regressor, and results are integrated through voting or averaging. RF learns the relationship between input and output data through bagging and random feature selection. After training, RF forms a powerful ensemble model and can handle complex classification or regression problems as well. The final prediction is determined by the ensemble function:

$$F(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x) \tag{4}$$

where $f_i(x)$ represents the prediction result of the ith DT, N represents the total number of DTs, and x represents the input feature vector. For classification problems, the final result typically uses majority voting; for regression problems, the average value is taken.

DT

In Fig. 1e, DT consists of a root node, internal nodes, and leaf nodes.[37] Each non-leaf node represents a feature test condition and connects to other nodes through branches. DT learns the rule relationship between input and output data through recursive partitioning. After training, DT forms a decision path from root to leaf and can model complex classification or regression problems. The splitting at each node is determined by an information gain function, defined as

$$Gain(D, a) = Info(D) - \sum_{v} \frac{|D_v|}{|D|} Info(D_v) \tag{5}$$

where D represents the dataset, a represents the feature, D_a represents the subset of data after splitting by feature a, Info represents entropy (for classification problems) or variance (for regression problems), and |D| represents the number of samples in the dataset.

CGNN

Due to the stagewise DT structure adopted by the Boost model, the output is typically piecewise constant. This characteristic makes model interpretation complicated and enhances the "black box" nature of the model. In contrast, the BPNN model uses a multilayer perceptron structure and nonlinear activation functions to achieve continuous function mapping between input variables and output. However, due to data limitations, the BPNN model usually has lower prediction accuracy.[38] As shown in Fig. 2, CGNN is a method that incorporates physical laws as constraints into neural networks.[27] Fig. 2 illustrates the comprehensive architecture of the CGNN framework: the scatter plot in the upper left corner depicts the distribution characteristics of the input dataset, while the colored nodes in the central region represent the multilayer neural network structure (blue nodes denote the input layer, yellow and green nodes correspond to the two hidden layers, respectively, and purple nodes represent the output layer). The three physical constraints (C1 geometric relationship, C2 reinforcement ratio limitation, and C3 strength correlation) enclosed within the green rectangular box on the right side are evaluated through the constraint evaluator (purple circle R) to quantify the degree of constraint violation. The composite loss function displayed in the orange box at the bottom integrates data fitting errors with physics-based constraint losses, thereby

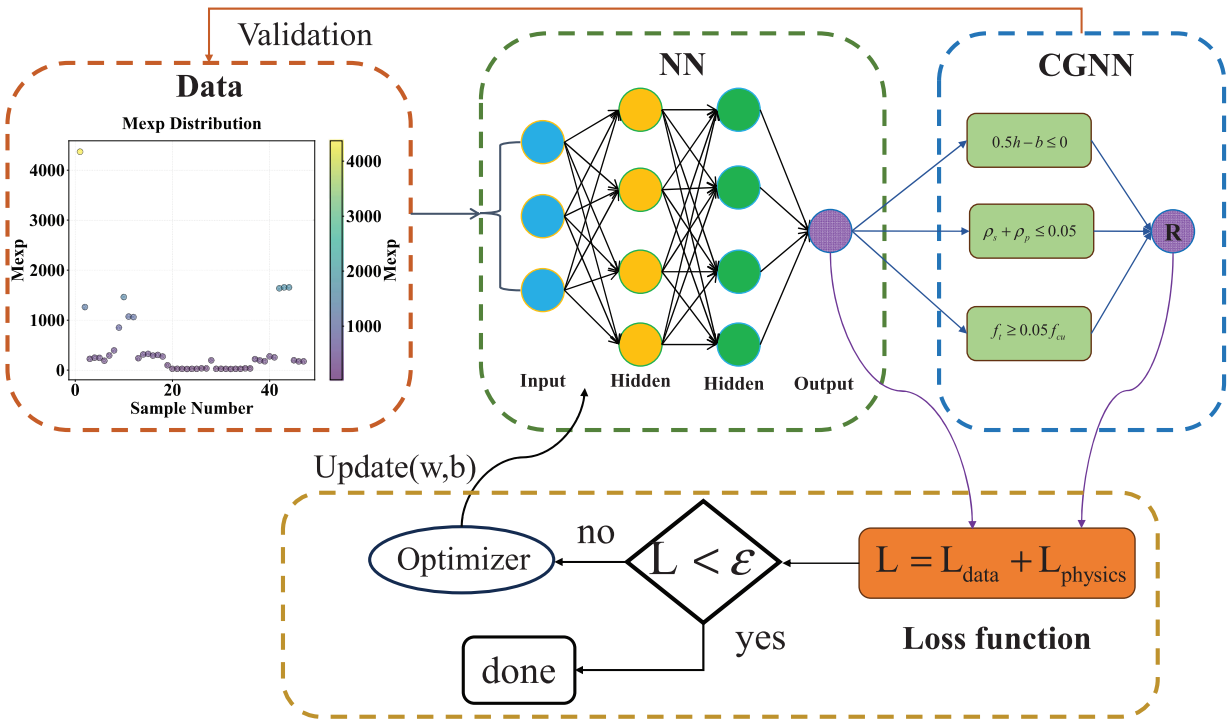


Figure 2. CGNN architecture

Table 1. Steps of the CGNN for flexural capacity prediction

Flowchart CGNN-based prediction of flexural capacity of UHPC beam
Input: Take eight characteristic parameters of UHPC beams including geometry, material properties, and reinforcement, as input dataset X
1. Design and construct a multilayer neural network structure as the basic framework of the prediction model
2. Define three physical constraint conditions: C_1, C_2, and C_3
3. In each iteration, first obtain the predicted flexural capacity through the neural network, then calculate the data loss coupling with the physical constraint loss as the total loss
4. Evaluate whether the prediction results satisfy physical laws to justify the constraint loss for the model prediction modification
5. Update the neural network parameters using backpropagation algorithm based on the total loss, and repeat this iteration process until model convergence

achieving an organic unification of data-driven learning and physics-informed guidance. The entire architecture operates through forward propagation to obtain predicted values while simultaneously conducting constraint evaluation, followed by backpropagation to concurrently optimize both prediction accuracy and physical consistency, which constitutes the fundamental innovation of CGNN compared to conventional neural networks. The model's total loss function is shown as follows:

$$L_{total} = L_{data} + \lambda L_{physics} \tag{6}$$

where L_{data} represents the data-driven loss term, $L_{physics}$ represents the physics-constrained loss term, and λ is the weight coefficient used to balance the importance of the two types of losses. Table 1 shows the steps of the CGNN for flexural capacity prediction.

This study improves the predictive performance of neural network models by incorporating physical constraints, specifically including three types of engineering constraints: (C1) cross-sectional geometric constraint, ensuring that the beam width does not exceed half of its height; (C2) reinforcement ratio constraint, limiting the total ordinary and prestressed steel reinforcement ratio to not exceed 5%; constraints (C1) and (C2) were suggested in GB 50010-2010[42] for the design of rectangular cross-sections and columns, respectively; (C3) relationship constraint between UHPC compressive and tensile strength, ensuring that the tensile strength is not less than 5% of the compressive strength. These constraints are integrated into the model's loss function through a penalty mechanism, where the model receives no penalty when the engineering constraints are satisfied, but a penalty will be applied accordingly if the engineering

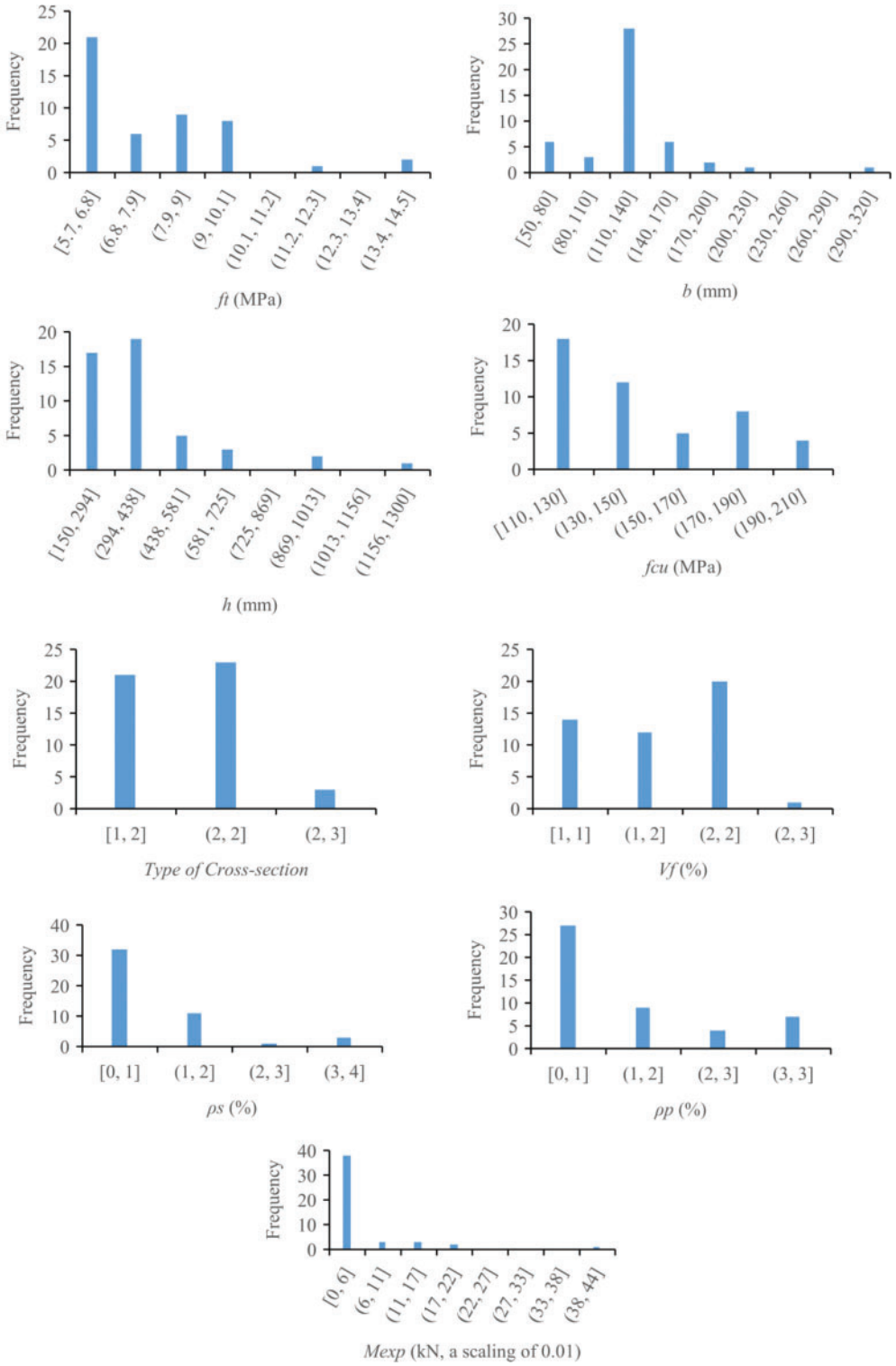


Figure 3. Features distribution of concrete beams

constraints are violated, thereby gradually adjusting model parameters during training to make predictions more consistent with engineering principles. This constraint mechanism not only significantly improves the physical reasonableness of model predictions but also enhances the reliability and applicability of the prediction model in practical engineering.

Experimental Database

The established database encompasses prestressed UHPC beams with various cross-sectional configurations (T-shaped, box-shaped, and rectangular), with the distribution characteristics of the parameters shown in Fig. 3. The specimens exhibit significant geometric variations, with cross-sectional widths (b) and heights (h) ranging from

50 to 300 and 150 to 1300 mm, respectively. Regarding material properties, the compressive strength (f_{cu}) primarily ranges from 110 to 200 MPa, tensile strength (f_t) ranges from 5.7 to 14.5 MPa, and steel fiber volume content (V_f) predominantly concentrates between 0.5% and 2%. In terms of reinforcement design, the longitudinal reinforcement ratio (ρ_s) shows a bimodal distribution around 0% and 4.17%, while the prestressed reinforcement ratio (ρ_p) mainly varies between 0% and 3.14%. The experimental flexural capacity (M_{exp}) ranges from 23 to 4370 kN·m, with a high data density in the low value region.

Modeling Parameters and Evaluation

Algorithms parameters

Both data-driven models and the physics informed model employ 80% of the data for training and the remaining 20% for testing. The parameters used for different models and their selection rationale are presented as follows: *BPNN*: First hidden layer: 64 neurons; Second hidden layer: 64 neurons; Activation: ReLU; Learning rate: 0.001; Maximum iterations: 1000; Batch size: 4; and Optimizer: Adam. The choice of 64 neurons per hidden layer follows the empirical rule of setting hidden layer size to 2–16 times the number of input features (8 in this case) to balance model complexity and training efficiency. The ReLU activation function is selected for its computational efficiency and ability to mitigate vanishing gradient problems. The learning rate of 0.001 is a standard setting for neural network training that ensures stable convergence. The small batch size (4) is suitable for small datasets and helps improve the stochasticity of gradient estimation. *CatBoost*: Number of iterations: 1000; Learning rate: 0.1; Maximum tree depth: 6; L2 regularization term: 3.0; Leaf estimation method: Gradient; Minimum data in leaf: 1; Bootstrap type: Bayesian; and Leaf weight update step: 0.03. The 1000 iterations ensure sufficient model convergence, while the learning rate of 0.1 balances training speed and stability. Tree depth is limited to 6 to prevent overfitting, and L2 regularization enhances generalization capability. Bayesian bootstrap is particularly suitable for small datasets as it provides better uncertainty estimation. *XGBoost*: Number of estimators: 1000; Learning rate: 0.1; Maximum tree depth: 6; Subsample ratio: 1.0; Minimum child weight: 1; Minimum split loss: 0; Column sample by tree: 1; and Tree method: exact. These parameters follow XGBoost best practices, with the combination of 1000 estimators and a 0.1 learning rate performing well in most regression tasks. The tree depth of 6 captures complex patterns while avoiding overfitting. *RF*: Number of trees: 1000; Maximum tree depth: 6; Minimum samples split: 2; Minimum samples leaf: 1; Split criterion: Mean Squared Error (MSE); Bootstrap: True; and Max features per leaf: n features. The 1000 trees ensure ensemble stability and accuracy, while depth limitation prevents individual tree overfitting. Bootstrap sampling is a core feature of RFs that enhances model generalization. *DT*: Maximum tree depth: 6; Min samples split: 2; Min samples leaf: 1; Split criterion: MSE; Max features: all. The depth limitation of 6 is crucial for preventing DT overfitting, especially on small datasets. *CGNN*: First hidden layer: 64 neurons; Second hidden layer: 64 neurons; Activation: tanh; Loss function: MSE + 0.1 × physics loss; Optimizer: Adam; Learning rate: 0.001; Batch size: 8; Epochs: 1000; Physical constraints: (C1) Geometry relationship; (C2) Reinforcement ratio limitation; and (C3) Strength relationship. The tanh activation function is chosen because its output range [−1, 1] helps stabilize physics constraint calculations. The physics loss weight of 0.1 was determined through grid search to achieve optimal balance between data fitting and physical consistency. The relatively larger batch size (8) facilitates stable computation of physics constraints.

All hyperparameter selections are based on preliminary experiments and best practices from the literature, and were optimized through cross-validation to ensure optimal performance on the given dataset.

Performance metrics

To comprehensively evaluate the performance of the CGNN model in predicting the flexural bearing capacity of prestressed UHPC beams, this study employs six statistical indicators for quantitative analysis. These indicators include mean ratio (ξ_m), coefficient of variation (ξ_{cov}), mean relative error (ε_s), coefficient of determination (R^2), maximum ratio (ξ_{Max}), and minimum ratio (ξ_{Min}). These evaluation metrics reflect the model's predictive performance from different perspectives. Among them, the mean ratio and coefficient of determination mainly reflect the overall prediction accuracy of the model; ideally, the mean ratio is close to 1 and the coefficient of determination is close to 1. The coefficient of variation and mean relative error reflects the dispersion degree of the prediction results, with lower values indicating more stable predictions. The maximum and minimum ratios reflect the extreme cases of prediction results, which can be used to evaluate the reliability range of model predictions. The specific calculation formulas for each indicator are detailed in Eqs. (7)–(12):

$$\xi_m = \frac{1}{n}\sum_{i=1}^{n}\frac{M_{pred,i}}{M_{exp,i}} \tag{7}$$

$$\xi_{cov} = \frac{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{M_{pred,i}}{M_{exp,i}} - \xi_m\right)^2}}{\xi_m} \tag{8}$$

$$\varepsilon_s = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{M_{pred,i} - M_{exp,i}}{M_{exp,i}}\right| \times 100\% \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(M_{exp,i} - M_{pred,i}\right)^2}{\sum_{i=1}^{n}\left(M_{exp,i} - \overline{M}_{exp}\right)^2} \tag{10}$$

$$\xi_{Max} = max\left(\frac{M_{pred,i}}{M_{exp,i}}\right) \tag{11}$$

$$\xi_{Min} = min\left(\frac{M_{pred,i}}{M_{exp,i}}\right) \tag{12}$$

where n is the number of samples, $M_{pred,i}$ and $M_{exp,i}$ represent the predicted flexural capacity and experimental flexural capacity of the ith sample, respectively, and $\overline{M}_{exp}$ is the mean value of experimental results.

Results and Discussion

Performance of the model

Fig. 4 shows a direct comparison of the observed and predicted bending capacity by different models. From the performance comparison in Fig. 5, CGNN with physical constraints shows significant improvements in all performance metrics compared to traditional BPNN. On the test set, CGNN achieves an R^2 of 0.994, while BPNN only reaches 0.906, representing an improvement of nearly 10%. In terms of prediction error, CGNN's ε_s is 8.57%, while BPNN's ε_s is as high as 20.96%, showing a reduction in error of approximately 59%.

Compared to other machine learning algorithms, CGNN also demonstrates significant advantages. Although CatBoost, XGBoost, and DT all achieve high R^2 values (all

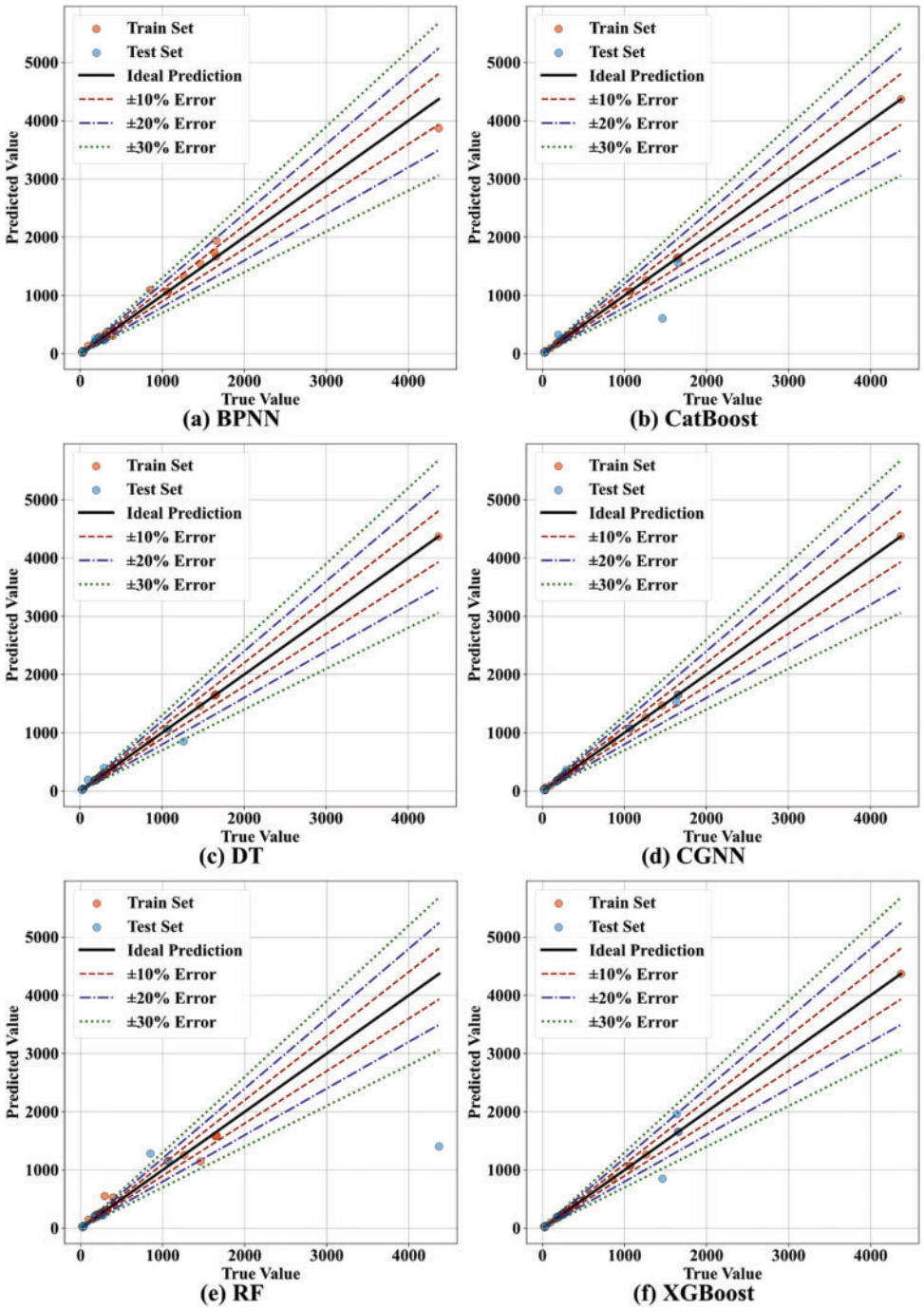
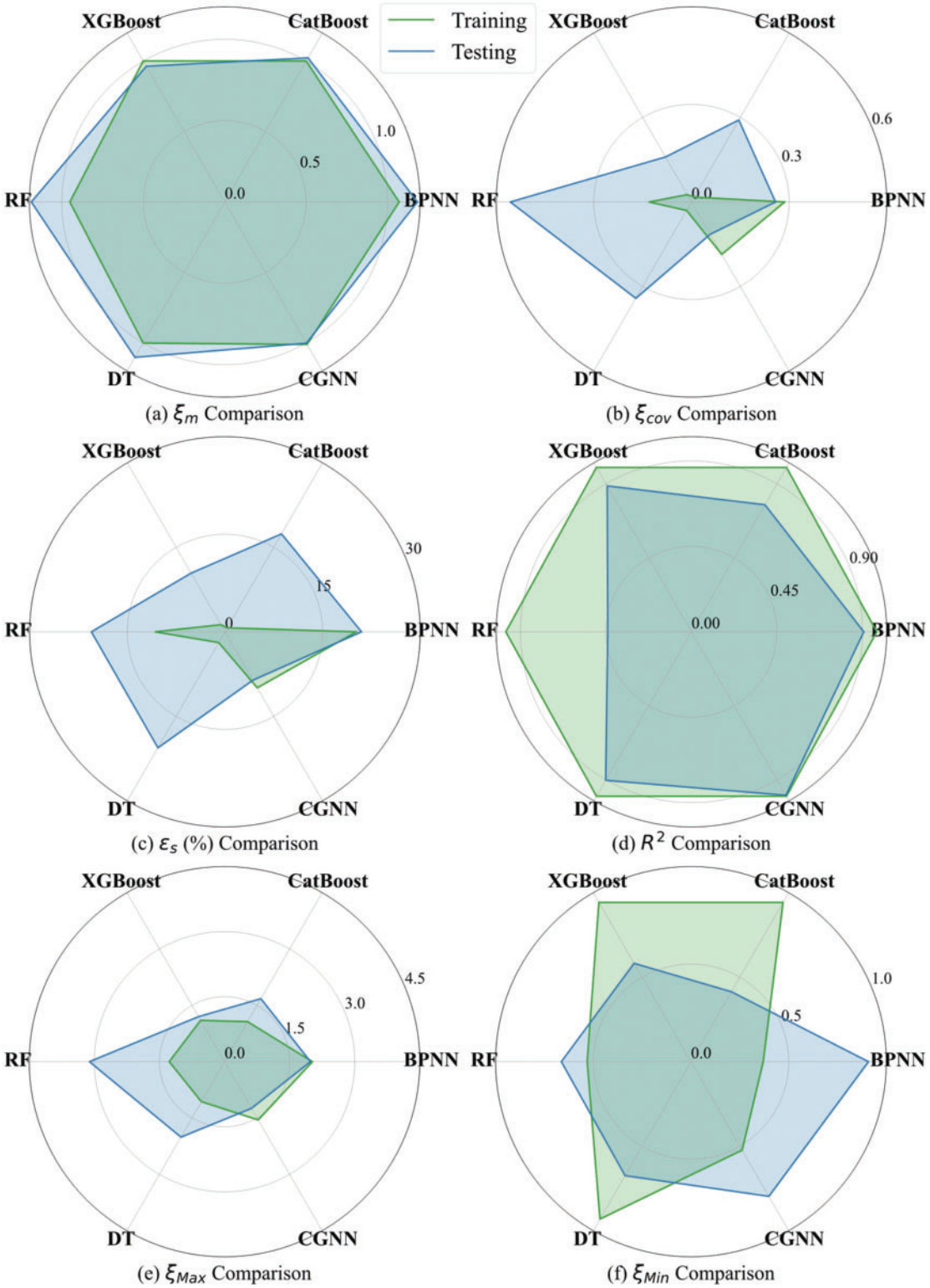


Figure 4. A direct comparison of the observed and predicted bending capacity (kN) by different models. (a) BPNN, (b) CatBoost, (c) DT, (d) CGGN, (e) RF, and (f) XGBoost



(a) ξ_m Comparison (b) ξ_{cov} Comparison (c) ε_s (%) Comparison (d) R^2 Comparison (e) ξ_{Max} Comparison (f) ξ_{Min} Comparison

Figure 5. Performance indicator values of each prediction model: (a) ξ_m, (b) ξ_{cov}, (c) ε_s(%), (d) R^2, (e) ξ_{max}, and (e) ξ_{min}

above 0.999) on the training set, their performance on the test set is inferior to CGNN: CatBoost's R^2 is 0.773, XGBoost's is 0.887, and DT's is 0.903. Particularly in terms of prediction stability, CGNN's coefficient of variation ($\xi_{cov} = 0.114$) significantly outperforms other algorithms (RF's 0.557, CatBoost's 0.291, and DT's 0.342). RF shows the most unstable performance on the test set, with an R^2 of only 0.441 and a large maximum prediction deviation ($\xi_{Max} = 3.113$).

In Fig. 6, different colors represent different models' mean relative error (ε_s) values, where darker colors indicate higher ε_s values and lighter colors indicate lower ε_s

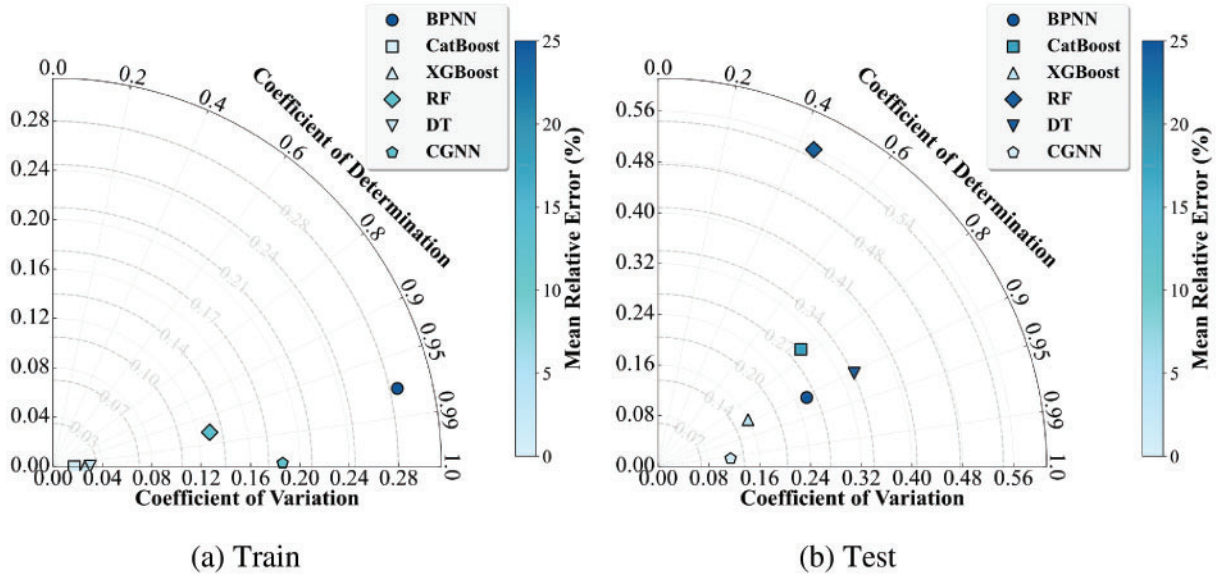


Figure 6. Prediction performance comparison of machine learning models. (a) Train and (b) test

Table 2. Comparison between CGNN and conventional equations

Models	ξ_m	ξ_{cov}	ε_s	R^2	ξ_{Max}	ξ_{Min}
JGJ/T 465-2019[43]	0.86	0.21	14	0.94	1.16	0.35
JTG 3362-2018[44]	1.17	0.17	17	0.99	1.73	0.85
NF P18-710-2016	1.03	0.13	3	0.98	1.36	0.79
CGNN—Train	1.01	0.18	9.93	0.99	1.54	0.52
CGNN—Test	1.00	0.11	8.57	0.99	1.24	0.79

Note: The test set results are the primary evaluation criterion for the model's generalization ability; the training set results are used for diagnosing overfitting and for fair comparison with conventional equations.

values. Additionally, points closer to the center indicate smaller coefficient of variation (ξcov) values, while points closer to the horizontal axis represent higher coefficient of determination (R^2) values. On the training set, CGNN, CatBoost, DT, and XGBoost achieved the highest R^2 (close to 1) and small coefficient of variation errors (close to 0), while maintaining relatively low mean relative errors. RF performed second best, maintaining a small coefficient of variation but slightly lower R^2. BPNN performed relatively poorly, showing the highest coefficient of variation and mean relative error. During the test set validation phase, CGNN demonstrated significant advantages, achieving optimal levels in all three evaluation metrics, that is, a minimal decline in R^2, and improvements both in coefficient of variation and mean relative error. XGBoost ranked second; BPNN, CatBoost, and DT showed more significant performance degradation, while RF performed worst on the test set, with notable deterioration in all three evaluation metrics. CGNN maintained consistently low error levels on both training and test datasets, demonstrating excellent generalization capability, while all other models showed varying degrees of performance degradation from training to testing.

Performance comparison with conventional model

Through comparative analysis with the results computed from existing equations in the standards, the proposed CGNN model also demonstrates excellent predictive performance as evidenced in Table 2. The mean ratio (ξ_m) of the model on training and test sets are 1.01 and 1.00, respectively, very close to the ideal value of 1.0, outperforming JGJ/T 465-2019 (0.86) and JTG 3362-2018 (1.17), and comparable to NF P18-710-2016 (1.03). In terms of coefficient of variation (ξ_{cov}), the CGNN model achieves 0.18 and 0.11 for training and test sets, respectively, with the test set performance superior to JGJ/T 465-2019 (0.21) and JTG 3362-2018 (0.17), and comparable to NF P18-710-2016 (0.13), indicating good prediction stability. Regarding prediction range, the model's predictions show reasonable distribution, with ranges of 0.52–1.55 and 0.80–1.24 for the training and test sets, respectively, which are more moderate compared to JGJ/T 465-2019 (0.35–1.16) and JTG 3362-2018 (0.85–1.73). The CGNN model achieves R^2 values of 0.999 and 0.994 on the training and test sets, with mean relative errors (ε_s) of 9.93% and 8.58%, outperforming JGJ/T 465-2019 (14%) and JTG 3362-2018 (17%), though slightly higher than NF P18-710-2016 (3%). These results

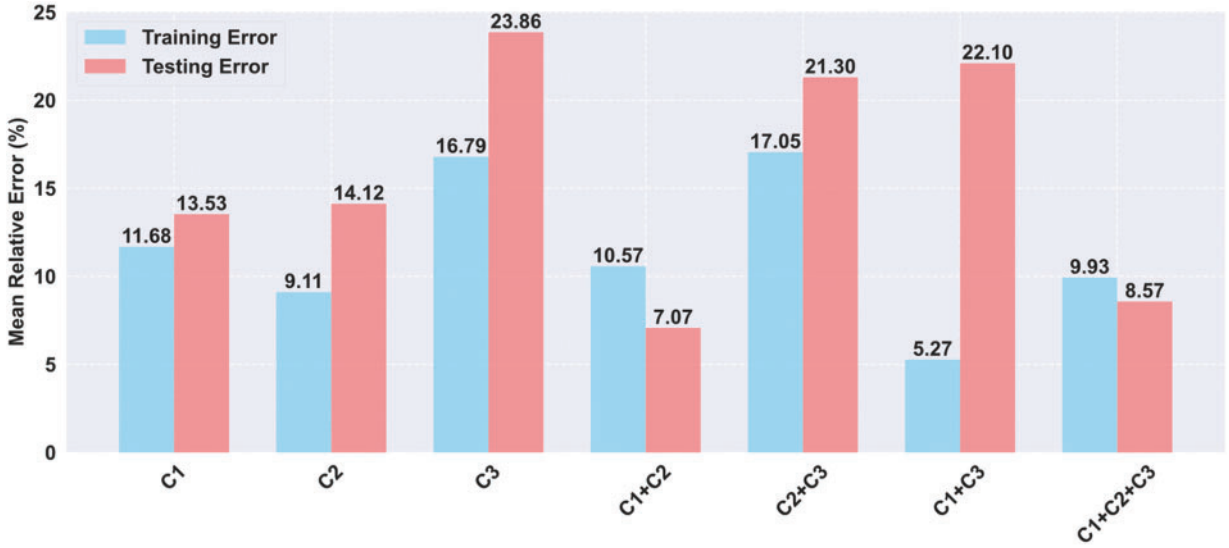


Figure 7. Comparison of training and testing errors of different model configurations

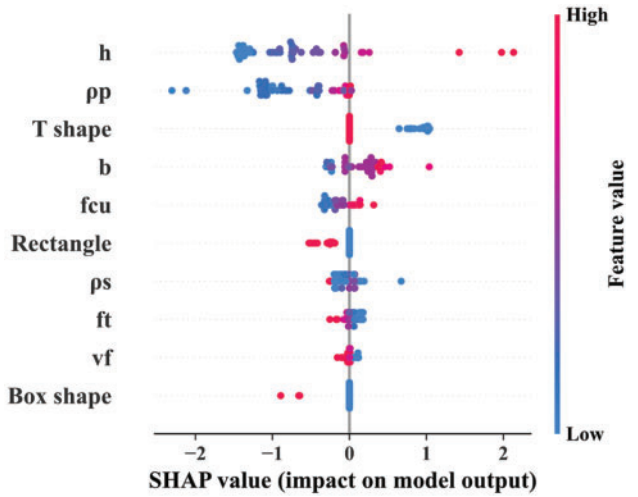


Figure 8. SHAP summary plot for CGNN

thoroughly demonstrate that the proposed CGNN model performs well in prediction accuracy, stability, and reliability, making it an effective supplement and alternative to existing equations.

CGNN constraints discussion

The three physical constraints are decomposed in this section to discuss their individual and combined impact on the CGNN model performance, and their training and testing errors are visualized and compared in Fig. 7. The combination of C1 and C2 demonstrates high prediction performance and stability, with a testing error as low as 7.07%. In contrast, the models with a single constraint show inferior performance, particularly when C3 is tested alone, resulting in a test error of up to 23.86%. Although the model incorporating all three constraints does not have the best performance, it exhibits good generalization ability with a small difference between training error (9.93%) and test error (8.57%). These experimental results reveal that the selection of physical constraints influences CGNN model performance, and that an appropriate constraint combination enhances both prediction accuracy and generalization ability. Despite the slightly superior performance of the dual-constraint condition, this study ultimately adopted the three-constraint combination, considering the completeness of the physical problem and model rationality. This decision was made because the three constraints represent different physical characteristics of the problem, and the complete constraints better ensure consistency between model prediction and actual physical laws, while the test error of 8.57% remains within an acceptable range.

Model explainability

Based on previous model evaluations, the CGNN algorithm demonstrated excellent predictive performance on both training and test sets. Therefore, CGNN was selected for model interpretability analysis using the game theory-based method—SHAP. Fig. 8 shows the SHAP summary plot, where the horizontal axis represents SHAP values (indicating the feature's impact on prediction results, with negative values decreasing predictions and positive values increasing predictions), while the vertical axis lists all features ordered by importance from high to low. Each point represents a sample, with colors ranging from blue to red indicating low to high feature values, and the horizontal position showing the impact of that sample's feature on the prediction result.

In terms of importance ranking, h, ρ_p, and T-shape are the most influential features, followed by b and f_{cu}. In contrast, box shape and fiber content have the least impact on predictions. h shows the widest distribution of SHAP values (−1.5 to 2), with color distribution indicating that larger heights (red points) tend to have stronger positive impacts. The SHAP values of ρ_p range from −2.5 to 0. As the reinforcement ratio increases, its negative impact on

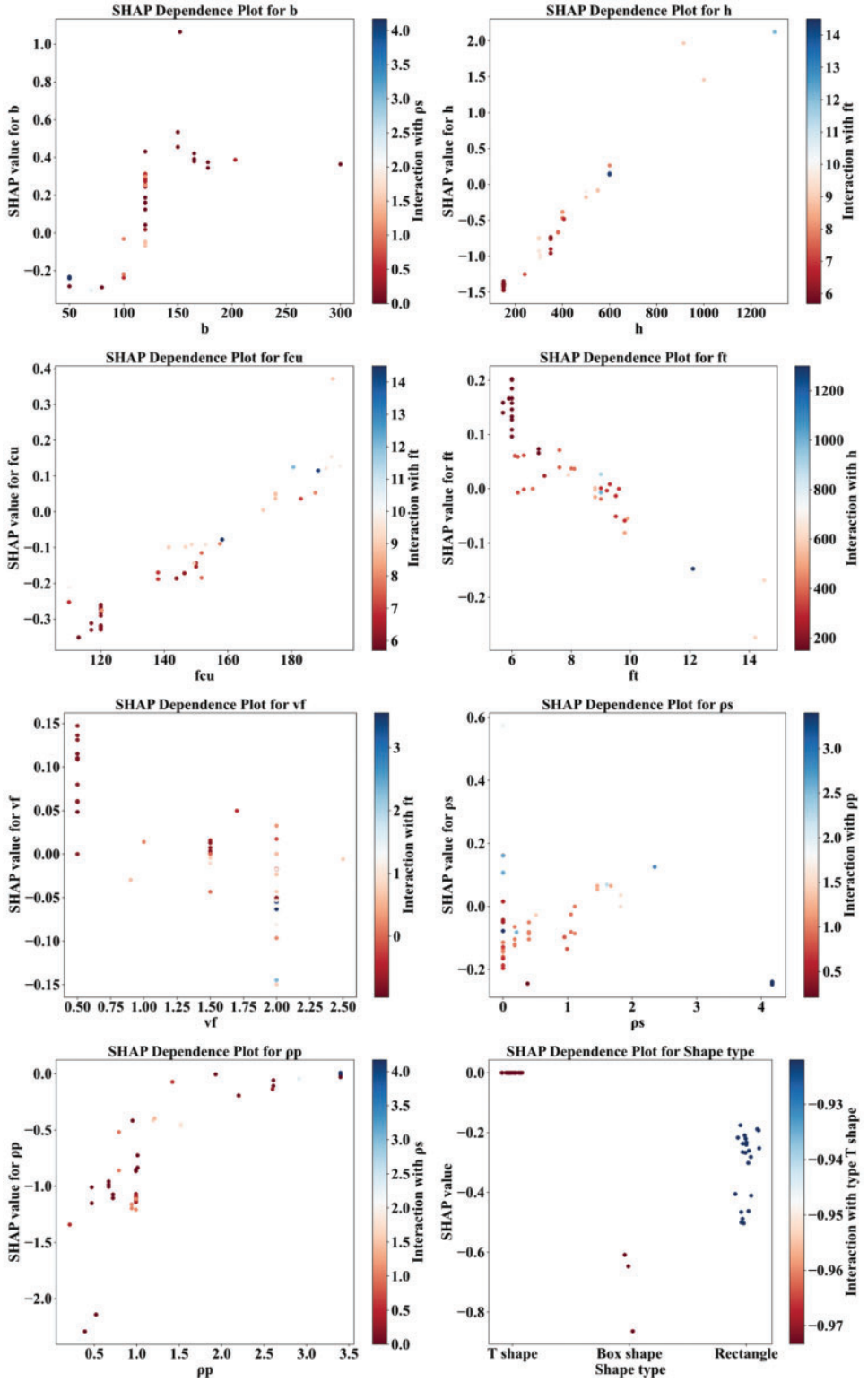


Figure 9. SHAP-based sensitivity analysis

the model output gradually weakens, showing an upward trend. T-shape mainly shows positive influence, concentrated between 0 and 1. Both b and f_{cu} demonstrate moderate impacts with relatively concentrated SHAP value distributions. Rectangle has relatively minor influences, and box shape shows slight negative impacts. The lowest-ranked features—f_t and V_f —have minimal impact, with their SHAP values clustered near zero, indicating their limited roles in model predictions.

The sensitivity relationships between various features and predictions were visualized through SHAP dependence plots, as shown in Fig. 9. The horizontal axis represents the actual parameter values, while the SHAP values on the vertical axis reflect the contribution of each parameter

to the prediction results (positive values indicate enhancement effects, negative values indicate weakening effects). The color of the scatter points reflects the intensity of other features. From Fig. 9: h shows the strongest positive correlation, with the maximum slope especially in the range of 200–600 mm, indicating that increasing height is the most effective for capacity improvement in this range, which aligns with findings by Shafieifar et al.[25]; b shows a rapid increase in SHAP values in the range of 50–150 mm before leveling off, suggesting limited benefits from increasing width beyond a certain value; f_{cu} exhibits a linear positive correlation, with SHAP values transitioning continuously from negative to positive as values increase, consistent with Yang and Fang's[39] research on the flexural performance of prestressed UHPC beams; f_t shows significant fluctuations in the 6–8 MPa range and demonstrates negative effects beyond 10 MPa; V_f exhibits distinct non-monotonic characteristics in the range of 0.5%–2.5%, with an optimal fiber content of 1.5%, corroborating Zhang et al.'s[40] findings on fiber volume effects on UHPC beam flexural hardening behavior; both ρ_s and ρ_p show negative effects at low reinforcement ratios, gradually improving as ratios exceed certain thresholds, a phenomenon also verified in Feng et al.'s[39] study on UHPC beam flexural behavior under varying reinforcement ratios; the influence of cross-sectional shape shows obvious discrete characteristics, with T-shape having higher SHAP values than rectangle, suggesting that T-shape might be the optimal choice in this structural system.

These SHAP-based sensitivity analysis results reveal the linear and nonlinear characteristics and degrees of various design parameters' impacts on structural capacity. These findings provide guidance for optimizing structural design and serve as a basis for further improving the CGNN model. Given the limitation of the experimental data, this machine learning-based analysis method demonstrates good generalizability and practical value.

GUI

The primary audience for this CGNN model comprises practicing engineers with limited expertise in PINN methodologies. Unlike traditional mechanical calculation methods, the CGNN model possesses complexities that deter UHPC designers from utilizing it. To address this issue, a GUI was developed internally using Python, aimed at providing a user-friendly interface for engineering professionals.

Fig. 10 shows the intuitive design of the GUI, which is freely available to all users. The GUI (accessible at https://github.com/TTTANDDD/PINN-Based-Flexural-Prediction-of-PS-UHPC-Beams/tags) prioritizes optimal user experience and employs a concise input interface for predicting the flexural capacity of prestressed UHPC beams. It simplifies the process for engineers to access predictive results by allowing input of beam parameters and utilizing a pre-trained CGNN model for predictions. Beyond its predictive capabilities, the GUI offers a resource-efficient alternative to traditional experimental testing. It provides rapid and approximate preliminary estimates, resulting in significant savings in financial resources, specimen preparation, time, and labor. Additionally, it serves as a valuable resource for those interested in flexural capacity data of prestressed UHPC beams, supporting detailed research endeavors.

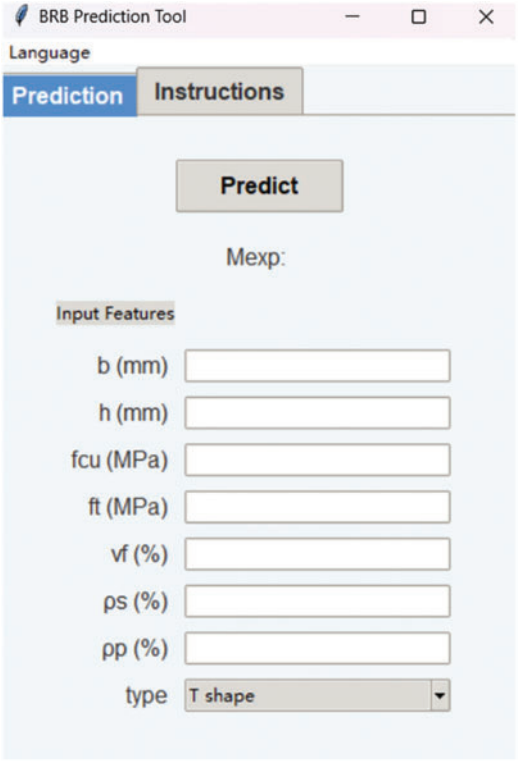


Figure 10. Graphical user interface

To utilize this GUI toolbox, users need only input relevant details. Once these details are entered, they can click the "Predict" button. The estimated flexural capacity is then displayed, providing a user-friendly and efficient prediction interface. This streamlined process ensures that even engineers with limited CGNN expertise can easily use the GUI to make accurate predictions.

Model limitations and future perspectives

Although the CGNN model demonstrates excellent performance in predicting the flexural capacity of prestressed UHPC beams, certain limitations exist and future improvements can be pursued. First, the training dataset is relatively small and primarily derived from laboratory conditions, which may not fully capture the diverse scenarios encountered in actual engineering applications; thus, the dataset can be expanded by incorporating more experimental data and engineering cases to enhance the model's generalization capability. Second, the model only incorporates three basic physical constraints, failing to fully describe the complex mechanical behavior of prestressed UHPC beams, and the selection of constraint weights remains largely empirical; therefore, additional physical constraints can be introduced, such as temperature effects, and more scientific constraint weight optimization methods can be developed to strengthen the model's theoretical foundation. Additionally, the model's

applicability is restricted to the parameter ranges covered by the dataset, and the reliability of predictions beyond these ranges requires further validation. Finally, considering these limitations may affect the model's effectiveness in certain specialized engineering scenarios, it is recommended to develop more comprehensive uncertainty quantification methods to provide more reliable references for engineering design.

Conclusion

This study develops a CGNN method for flexural capacity prediction of prestressed UHPC beams with a small dataset. The physical constraints include geometrical relationship, reinforcement ratio limitation, and strength relationship, and the effectiveness of the constraints is confirmed from the decomposition discussion. The CGNN shows the best prediction performance in both training and testing datasets and solves the performance inconsistency (i.e., high accuracy in the training dataset but low accuracy in the testing dataset) observed in other machine learning models with the small datasets. Compared to BPNN, the testing mean relative error and R^2 improved by 59% and 10%, respectively; compared to the second ranked model (i.e., XGBoost), these values are 17.5% and 10.7%, while they are comparable with the best conventional equation from NF P18-710-2016. The interpretability from SHAP highlights the significance of UHPC beam height and prestressed reinforcement ratio for structural analysis and design.

Acknowledgments

This study acknowledges the research start-up support from Montana Technological University.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability statement

Some or all data, models, or code that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] Graybeal B. Flexural behavior of an ultrahigh-performance concrete I-girder. *J Bridge Eng*. 2008;13(6):602–610. doi:10.1061/(asce)1084-0702(2008)13:6(602).

[2] Wille K, Parra-Montesinos GJ. Effect of beam size, casting method, and support conditions on flexural behavior of ultra-high-performance fiber-reinforced concrete. *ACI Mater J*. 2012;109(3):379–388. doi:10.14359/51683829.

[3] Graybeal B, Brühwiler E, Kim BS, Toutlemonde F, Voo YL, Zaghi A. International perspective on UHPC in bridge engineering. *J Bridge Eng*. 2020;25(11):04020094. doi:10.1061/(asce)be.1943-5592.0001630.

[4] Hung CC, El-Tawil S, Chao SH. A review of developments and challenges for UHPC in structural engineering: behavior, analysis, and design. *J Struct Eng*. 2021;147(9):03121001. doi:10.1061/(asce)st.1943-541x.0003073.

[5] Shao Y, Nguyen W, Bandelt MJ, Ostertag CP, Billington SL. Seismic performance of high-performance fiber-reinforced cement-based composite structural members: a review. *J Struct Eng*. 2022;148(10):03122004. doi:10.1061/(asce)st.1943-541x.0003428.

[6] Russell HG, Graybeal BA. *Ultra-High Performance Concrete: A State-of-the-Art Report for the Bridge Community*. McLean, VA: United States Federal Highway Administration, Office of Infrastructure; 2013.

[7] Dong Y. Performance assessment and design of ultra-high performance concrete (UHPC) structures incorporating life-cycle cost and environmental impacts. *Construct Build Mater*. 2018;167(Apr):414–425. doi:10.1016/j.conbuildmat.2018.02.037.

[8] Haber ZB, la Varga IDe, Graybeal BA, Nakashoji B, El-Helou R. *Properties and Behavior of UHPC-Class Materials. Rep. No. FHWA-HRT-18-036*. McLean, VA: Federal Highway Administration; 2018.

[9] Lavorato D, Bergami AV, Nuti C, et al. Ultra-high-performance fibre-reinforced concrete jacket for the repair and the seismic retrofitting of Italian and Chinese RC bridges. *Proceedings of the 6th International Conference on Computational Methods in Structural Dynamics and Earthquake Engineering*. vol. 1. Rhodes Island, Greece. 2017:2149–2160.

[10] Shao Y, Parks A, Ostertag CP. Carbon footprint between steel-reinforced concrete and UHPC beams. *J Struct Eng*. 2023;149(3):06023001. doi:10.1061/jsendh.steng-11449.

[11] Shao Y, Billington SL. Impact of UHPC tensile behavior on steel reinforced UHPC flexural behavior. *J Struct Eng*. 2022;148(1):04021244. doi:10.1061/(asce)st.1943-541x.0003225.

[12] Saqif MA, Tai YS, El-Tawil S. Experimental and computational evaluation of the ductility of UHPC beams with low steel reinforcement ratios. *J Struct Eng*. 2022;148(7):04022077. doi:10.1061/(asce)st.1943-541x.0003378.

[13] El-Helou RG, Graybeal BA. Flexural behavior and design of ultrahigh-performance concrete beams. *J Struct Eng*. 2022;148(4):04022013. doi:10.1061/(asce)st.1943-541x.0003246.

[14] de Lima PB, Krahl PA, Silva FA, Cardoso DCT. A modeling strategy for the flexural performance prediction of UHPC beams accounting for variability of properties. *Compos Struct*. 2023;322:117374. doi:10.1016/j.compstruct.2023.117374.

[15] Givonetti M, Mairone M, Asso R, De Luca E, Grateron LAB, Masera D, Marano GC. Non-linear analysis of RC and PRC girder bridges. *Designs*. 2023;7(4):102. doi:10.3390/designs7040102.

[16] Graybeal B. *Structural Behavior of a Prototype Ultra-High Performance Concrete pi-girder. FHWA-HRT-10-027*. McLeanVA: Federal Highway Administration; 2009.

[17] Aaleti S, Petersen B, Sritharan S. *Design Guide for Precast UHPC Waffle Deck Panel System, Including Connections*. FHWA-HIF-13-032, Washington DC: Federal Highway Administration; 2013.

[18] Yoo DY, Banthia N, Kang ST, Yoon YS. Size effect in ultra-high-performance concrete beams. *Eng Fract Mech*. 2016;157:86–106. doi:10.1016/j.engfracmech.2016.02.009.

[19] Fiore A, Berardi L, Marano GC. Predicting torsional strength of RC beams by using evolutionary polynomial regression. *Adv Eng Softw*. 2012;47(1):178–187. doi:10.1016/j.advengsoft.2011.11.001.

[20] Yang IH, Joh C, Kim BS. Flexural strength of large-scale ultra high performance concrete prestressed T-beams. *Can J Civil Eng*. 2011;38(11):1185–1195. doi:10.1139/l11-078.

[21] Sim C, Tadros M, Gee D, Asaad M. Flexural design of precast, prestressed ultra-high-performance concrete members. *PCI J*. 2020;65(6):35–61.

[22] Feng JH, Shao XD, Qiu MH, Li HH, Gao X, Huang ZL. Reliability evaluation of flexural capacity design provision for UHPC beams reinforced with steel rebars/prestressing tendons. *Eng Struct*. 2024;300:117160. doi:10.1016/j.engstruct.2023.117160.

[23] AFNOR. *National Addition to Eurocode 2—Design of Concrete Structures: Specific Rules for Ultra-High Performance Fibre-Reinforced Concrete (UHPFRC)*. NF P 18-710. Paris: French Standard Institute; 2016.

[24] MCS-EPFL. *Recommendation: Ultra-High Performance Fibre Reinforced Cement-Based Composites (UHPFRC)*. Lausanne, Switzerland: Maintenance, Construction and Safety of Structures (MCS)-École polytechnique fédérale de Lausanne (EPFL); 2016.

[25] Shafieifar M, Farzad M, Azizinamini A. A comparison of existing analytical methods to predict the flexural capacity of Ultra High Performance Concrete (UHPC) beams. *Constr Build Mater*. 2018;172:10–18. doi:10.1016/j.conbuildmat.2018.03.229.

[26] Kumar R, Rai B, Samui P. Prediction of mechanical properties of high-performance concrete and ultrahigh-performance concrete using soft computing techniques: a critical review. *Struct Concrete*. 2025;26(2):1309–1337. doi:10.1002/suco.202400188.

[27] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys*. 2019;378:686–707. doi:10.1016/j.jcp.2018.10.045.

[28] Lee S, Popovics JS. Ultrasonic defect detection in a concrete slab assisted by physics-informed neural networks. *NDT & E Int*. 2025;151:103311.

[29] Yang Y, Guo Z, Liu Z. Parameter identification in prestressed concrete beams by incremental beam-column equation and physics-informed neural networks. *Comput Aided Civ Infrastruct Eng Online Early View*. 2025;26:3051. doi:10.1111/mice.13480.

[30] Solhmirzaei R, Salehi H, Kodur V. Predicting flexural capacity of ultrahigh-performance concrete beams: machine learning-based approach. *J Struct Eng*. 2022;148(5):04022031. doi:10.1061/(asce)st.1943-541x.0003320.

[31] Rahman MA, Zhang TJ, Lu Y. PINN-CHK: physics-informed neural network for high-fidelity prediction of early-age cement hydration kinetics. *Neural Comput Appl*. 2024;36:13665–13687.

[32] Wan YT, Zheng WZ, Wang Y. Identification of chloride diffusion coefficient in concrete using physics-informed neural networks. *Constr Build Mater*. 2023;393:132049.

[33] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature*. 1986;323(6088):533–536. doi:10.1038/323533a0.

[34] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A, CatBoost: unbiased boosting with categorical features. In: *Advances in Neural Information Processing Systems 31*. 2018.

[35] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; 2016:785–794.

[36] Breiman L. Random forests. *Mach Learn*. 2001;45:5–32.

[37] Loh WY. Classification and regression trees. *Wiley Interdiscip Rev: Data Min Knowl Dis*. 2011;1(1):14–23. doi:10.1002/widm.8.

[38] Bejani MM, Ghatee M. A systematic review on overfitting control in shallow and deep neural networks. *Artif Intell Rev*. 2021;54(8):6391–6438. doi:10.1007/s10462-021-09975-1.

[39] Yang J, Fang Z. Research on flexural behaviors of prestressed ultra high performance concrete beams. *Chin J Highway Transp*. 2009;22(1):39–46.

[40] Zhang Y, Zhu Y, Qiu J, Hou C, Huang J. Impact of reinforcing ratio and fiber volume on flexural hardening behavior of steel reinforced UHPC beams. *Eng Struct*. 2023;285:116067. doi:10.1016/j.engstruct.2023.116067.

[41] Feng Z, Li C, Yoo DY, Pan R, He J, Ke L. Flexural and cracking behaviors of reinforced UHPC beams with various reinforcement ratios and fiber contents. *Eng Struct*. 2021;248:113266. doi:10.1016/j.engstruct.2021.113266.

[42] China Ministry of Construction. *Code for Design of Concrete Structures (GB50010-2010)*. Beijing: China Architecture and Building Press; 2010.

[43] Ministry of Housing and Urban-Rural Development of the People's Republic of China. *Chinese Standard JGJ/T 465–2019: Standard for Design of Steel Fiber Reinforced Concrete Structures*. Beijing, China; 2019.

[44] Ministry of Transport of the People's Republic of China. *Chinese Standard JTG 3362–2018: Specifications for Design of Highway Reinforced Concrete and Prestressed Concrete Bridges and Culverts*. Beijing, China; 2018.