



## OPEN DSCostPred: a double-stacking model for construction cost prediction

Chen-Ping Liu<sup>1</sup>, Xin-Gen Sun<sup>2</sup> & Jian-Hua Guan<sup>3</sup>✉

The prediction of construction project cost plays a core role in engineering construction projects. However, the current prediction involves a multi-dimensional and dynamically variable system, and each major category can be further subdivided into many specific factors. Meanwhile, variables' relationships present a complex network of nonlinearity and interaction, which seriously affected the prediction accuracy. To solve this problem, we proposed a dual-stacking construction cost prediction method based on variable stacking and model stacking (DSCostPred). This method emphasizes that classifying variables and applying different algorithms respectively can avoid the impact of variables' functional differences. First, the variables are pre-classified to avoid mutual interference among them. Then, to learn the attribute and function positioning, as well as the complex interaction among them, different types of models are utilized to learn the variables. In algorithm design, to achieve the organic combination of multiple attributes and multiple models, a variable stacking is introduced into stacking ensemble learning to form collaborative predictions with model stacking. This method was compared with the classical method on real data, and the results show the superior performance. In addition, the ablation experiments and SHAP analysis also demonstrated the feasibility of the double-stacking idea we proposed.

**Keywords** Construction cost prediction, Dual-stacking method, Variable stacking, Model stacking

While the rapid expansion of China's construction and real estate sectors since the start of the 21st century has driven urbanization and improved living conditions, the total output of construction industry has decelerated since 2013. In response to this market slowdown and fierce competition, firms are universally seeking to boost competitiveness through stricter cost management. Cost control in construction project is implemented through cost prediction in the initial planning and feasibility study stages, with the existing prediction methods primarily falling into two categories: machine learning-based, and deep learning-based<sup>1-4</sup>.

Machine learning-based construction cost prediction employs a variety of algorithms, including Support Vector Machine (SVM), BP neural networks, XGBoost, and Random Forest<sup>5-7</sup>. Classically, on the basis of SVM, Miao et al.<sup>8</sup> proposed a Least Squares Support Vector Machine (LSSVM) for construction cost prediction, which achieved a relative error of less than 7%, thereby demonstrating high accuracy and stability. Similarly, on the basis of SVM, Wang et al.<sup>9</sup> also proposed a new model called PCA-LSSVM for the cost prediction of residential construction. Through robust validation with 290 cases from 25 companies, PCA-LSSVM proved highly effective in estimating costs. However, although SVM has achieved certain results, it has many deficiencies in handling heterogeneous features. Therefore, a more comprehensive model is urgently needed. In 2022, combining the extreme gradient boosting method with random forest (XGBoost-RF), artificial neural network (ANN) and SVM, Wu et al.<sup>10</sup> conducted a construction cost prediction on 90 construction projects in Iraq. The results demonstrated that the inflation rate is the most important indicator, and the XGBoost-RF was the top-performing model with a mean absolute error of just 0.25%. By integrating PCA for dimensionality reduction with a BP network for nonlinear fitting, Liu et al.<sup>11</sup> introduced an improved model PCA-BP to distinguish between controllable and uncontrollable factors in construction project. The model first applies PCA to preprocess large-scale construction data for improved training, then uses the BP network for deep analysis and prediction. Their results demonstrate that the approach boosts prediction accuracy, enables dynamic management, and improves project success rates and efficiency. To address BP's tendency to converge to local minima and its slow convergence speed, Feng et al.<sup>12</sup> developed a GA-BP model, in which the Genetic Algorithm (GA) can optimize the BP. The model's generalization ability was assessed using 18 train cases and 2 test cases, validating its effectiveness for

<sup>1</sup>Department of Architectural Engineering, Hunan Defense Industry Polytechnic, Xiangtan 411207, China. <sup>2</sup>School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, China. <sup>3</sup>School of Computational Science and Electronics, Hunan Institute of Engineering, Xiangtan 411105, China. ✉email: 25038@hnie.edu.cn

project cost estimation. To tackle the issues of low accuracy and efficiency caused by high project complexity and uncertainty, Zheng et al.<sup>13</sup> developed a prediction system based on 14 secondary indicators and proposed a Bird Swarm Algorithm-based Random Forest model (BSA-RF). The results on construction data from Xinyu, China showed that the model outperformed both traditional and recent methods in accuracy and efficiency, offering a reliable reference for cost management.

In deep learning-based methods, construction costs are predicted using algorithms such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Generative Adversarial Network (GAN), and Convolutional Neural Network (CNN). Classically, Shi et al.<sup>14</sup> evaluated the performance of LSTM, GRU, and Transformer in construction cost prediction. The results showed that Transformer is the most accurate model, LSTM performed reliably but less accurately, GRU trained faster but was less precise. Li et al.<sup>15</sup> developed a deep neural network (DNN) model that uses project and item features to predict engineering costs. The model achieved high accuracy, with errors as low as 4.203% for total cost and between 2.98% and 4.52% for unit prices, demonstrating its effectiveness for cost management. Feng<sup>16</sup> posited that traditional cost prediction methods are inadequate for managing complex data structures and multimodal features. Therefore, an intelligent model incorporating subtractive clustering, a self-learning mechanism, and a convolutional neural network (CNN) was constructed. This integrated framework applies clustering for data optimization, a self-learning mechanism for parameter tuning, and a CNN for deep feature extraction from diverse data types (images, text, and numerical values), enabling highly accurate predictions. To address the challenge of early-stage cost prediction, Liu et al.<sup>17</sup> introduced a hypergraph deep learning-based framework. This framework first defines a hypergraph that represents cost factors and their relationships. Subsequently, it then employs a deep learning model for end-to-end prediction, and finally quantitatively reveal the importance of each cost factor. To address the challenge of limited and unreliable data in early-stage cost prediction, Hong et al.<sup>18</sup> developed a method called CTGANs for data augmentation. By training an ANN on this synthetic data, their model effectively addressed data scarcity and imbalance. This approach notably reduced RMSE by about 66% and increased predictive effectiveness from 0% to 15.09% compared to a baseline model using only original data.

Based on the analysis of the aforementioned methods, this paper suggests that the core challenge in construction cost prediction is the correct identification of variable functions and attributes, and the effective learning of the complex interaction among them: (1) There are both continuous and categorical variables in construction cost prediction. If only one or a type of model is using without distinction, it may lead to the loss of data information, because these variables need to be preprocessed to eliminate the differences and noise interference; (2) Construction cost variables fall into distinct functional categories (e.g., area, materials, equipment), each with its own logic. A unified modeling approach struggles with this diversity: it must learn excessively complex relationships, often at the cost of interpretability, and fails to respect the specificities of each variable type. Although this goal can be achieved by using neural networks, it lacks interpretability. Consequently, we advocate for a multi-model framework that can integrate multiple models and classify different attribute variables, thereby improving both accuracy and robustness.

In this study, to enhance model interpretability and address the aforementioned prediction challenges, we propose a dual-stacking model for construction cost prediction (DSCostPred) which incorporates both model stacking and variable stacking. Specifically, to avoid mutual interference within variables, the variables are first pre-classified by functional clustering. Then, different types of models are employed to learn the distinct roles and complex interactions of these variables. In terms of algorithm design, a novel variable stacking mechanism is integrated into the ensemble learning framework to achieve an organic synthesis of multiple attributes and models. The main contributions are as follows:

- (1) To reduce the burden of variable differences on model predictions, a vertical variable segmentation is conducted and pre-classify variables according to their attributes and functions;
- (2) Improvements were made to the stacking ensemble learning, not only stacking on models but also on variables or data, providing a methodology for construction cost prediction;
- (3) Experiments and SHAP analyses on a dataset containing 332 samples from China have demonstrated the effectiveness and interpretability of our method.

## Materials and methods

### Data collection and statistics

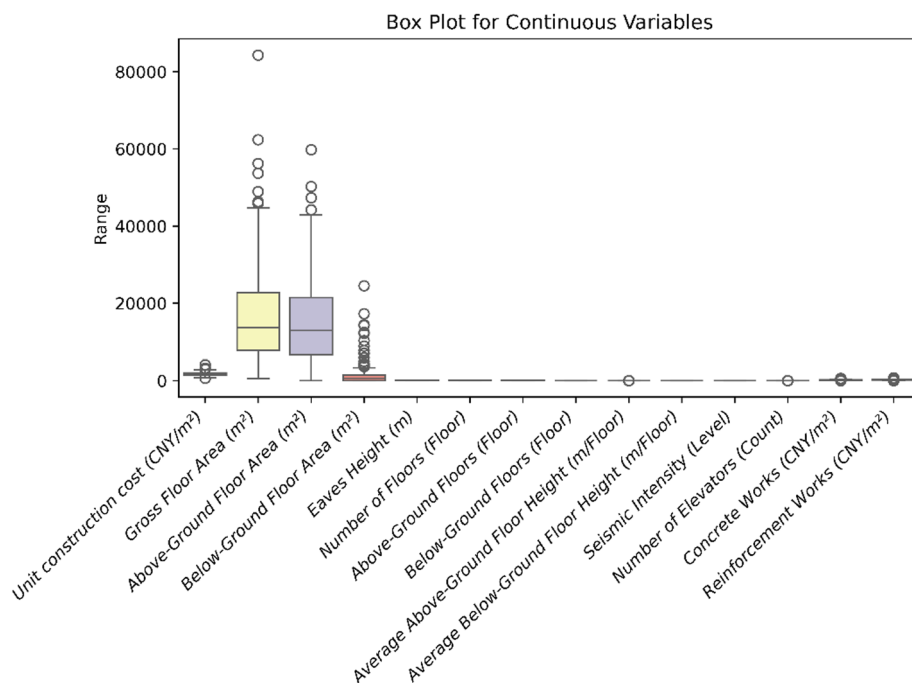
#### *Data collection and preprocessing*

For the dataset used in this study, from the Construction Cost Platform (<https://chaoshi.zjtcn.com/searchclassify?pg=1&l=26>), we collected a dataset including 412 nominal records of residential projects from 21 provinces in China from 2019 to 2022. All the data are nominal data. This data contains 14 continuous variables and 12 categorical variables. The names, units, and types of the variables are shown in Table 1 (where “Type = C” represents continuous variables, “Type = IC” represents integer continuous variables, and “Type = D” represents categorical variables; the categorical variables have no units). Then, these data are preprocessed to obtain data that can be directly used by the algorithm (Supplementary file 1). For detailed processing rules, please refer to Supplementary file 2.

Although the data provided in Supplementary file 1 can already be used by the algorithm, it still fails to meet the experimental requirements because there are some outliers (Fig. 1). To reduce the impact of outliers on model performance, all outliers were removed in this study, and ultimately an experimental dataset containing 332 samples was obtained.

Variable	Type	Unit	Variable	Type	Unit
Unit construction cost	C	CNY/m <sup>2</sup>	Foundation Type	D	--
Gross Floor Area	C	m <sup>2</sup>	Earthwork	D	--
Above-Ground Floor Area	C	m <sup>2</sup>	Foundation Pit Support	D	--
Below-Ground Floor Area	C	m <sup>2</sup>	Masonry Works	D	--
Eaves Height	C	m	Waterproofing Works	D	--
Number of Floors	IC	Floor	Exterior Wall Material	D	--
Above-Ground Floors	IC	Floor	Interior Finishing Material	D	--
Below-Ground Floors	IC	Floor	Roof Decoration	D	--
Average Above-Ground Floor Height	C	m/Floor	Heating Works	D	--
Average Below-Ground Floor Height	C	m/Floor	Number of Elevators	IC	Count
Structural Type	D	--	Fire Protection Works	D	--
Seismic Intensity	IC	Level	Concrete Works	C	CNY/m <sup>2</sup>
Civil Air Defense	D	--	Reinforcement Works	C	CNY/m <sup>2</sup>

**Table 1.** Variable description.



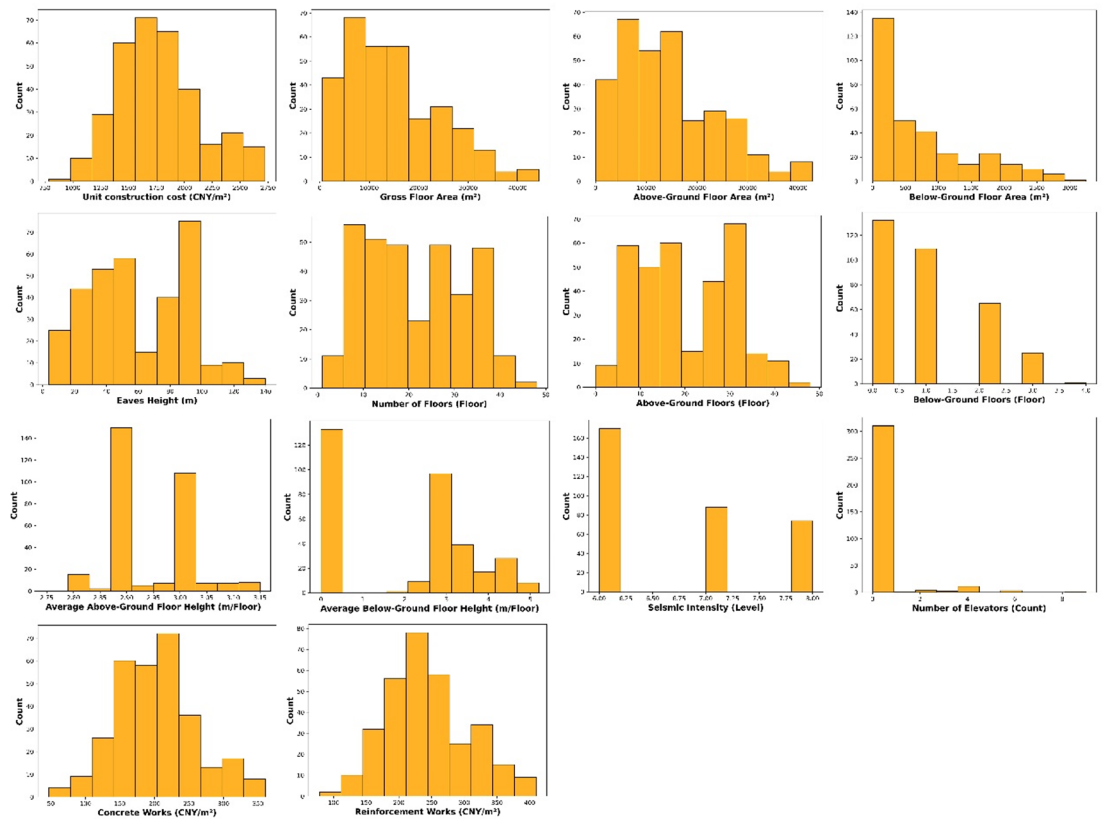
**Fig. 1.** Outliers plot of the continuous variables. Only 14 continuous variables are listed, as this issue is applicable to categorical variables.

*Statistical analysis*

To gain an intuitive understanding of the data, data statistics was conducted. Figure 2 shows all continuous variables’ data distribution, Fig. 3 shows all categorical variables’ data distribution, and Supplementary file 3 lists the distribution statistics of all variables, including mean, variance, minimum value, Q1, median, Q3, and maximum value.

**Double-stacking model-based construction cost prediction**

Figure 4 shows the process of DSCostPred. This method takes the classic stacking ensemble learning as its main structure. The approach of stacking is to first construct multiple different types of first-level learners (i.e., base learners), and use them to obtain first-level prediction results. Then, based on these first-level prediction results, a second-level learner (i.e., meta-learner) is constructed to obtain the final prediction result. The motivation of stacking can be described as follows: If a first-level learner mistakenly learns a certain area of the feature space, then a second-level learner can appropriately correct this error by combining the learning behaviors of other first-level learners. Specifically, in the training of base learners, the K-fold cross-validation is used to divide the



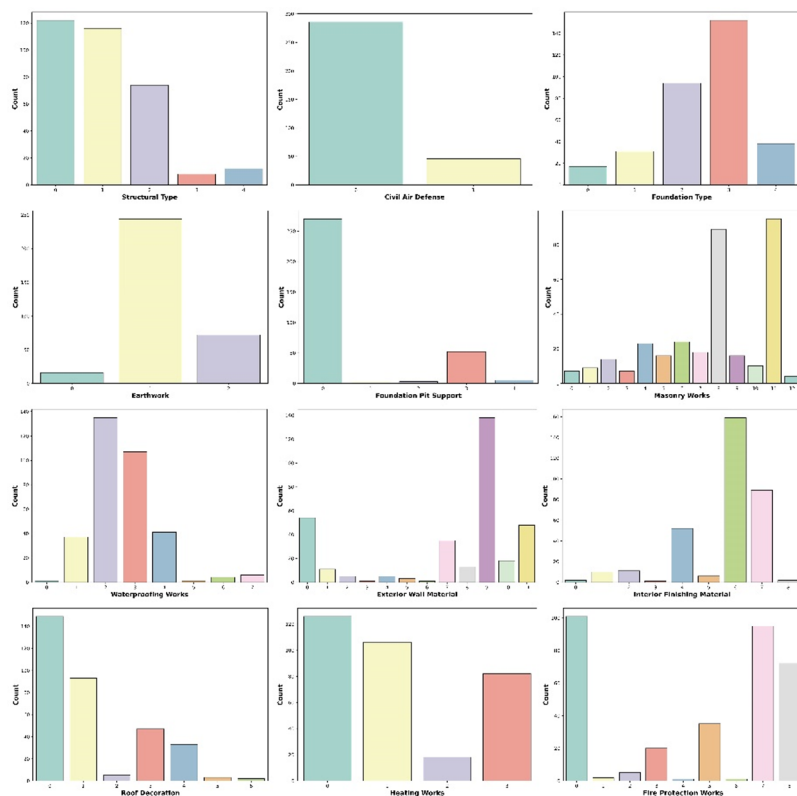
**Fig. 2.** Frequency of distribution plots of continuous variables.

training data into  $K$  parts, and then  $K$  base learners are adopted to learn different folds respectively. In this way,  $K$  results can be obtained. Then these results will be input as metadata into a single meta-learner for prediction. This framework is very powerful and robust. However, we believe that this framework may not be the best when dealing with multiple types of variables, because although it uses multiple models at the base-learners stage, it does not distinguish them, and these models are generally homogeneous. This leads to the situation when different variables are input into the model, it may achieve good results on some variables while not on others. To solve this problem, we introduced variable stacking in this framework. As shown in Fig. 4, variable stacking mainly modifies the base learning stage of stacking ensemble learning. It is composed of the vertical variable segmentation in Step1 and the vertical feature concatenation in Step4. Through the variable segmentation in Step1, the data is split into  $K$  parts, each containing different variables. Then, the base learning stage in stacking ensemble learning can be replicated  $K$  times. Meanwhile, during different base learning processes, the data is allowed to be replicated and input into different models for training. This enables the data to be fully utilized, corresponding to the “repeat” in Step2. The cross-validation in Step2 is the operation of stacking ensemble learning itself. Different from our vertical variable segmentation, it performs horizontal data variable segmentation for each piece of data. Afterwards, all the copied and partitioned data can be input into various types of models to train and obtain features with sufficient information. These features are finally concatenated in Step4 to form comprehensive features for use by the meta-learner. In simple terms, our improvement is similar to the transformation from the attention mechanism to multi-head attention mechanism in classic model Transformer in the field of natural language processing. The specific details are as follows.

#### *Variable attributes and functions-based vertical variable segmentation*

The prediction of unit construction cost involves many variables. In this paper, it includes 25 variables (see Table 1). In fact, these variables can be classified into the following categories based on their attributes and functions. Meanwhile, among these variables, there may be situations such as hierarchy, interaction effects, coexistence of qualitative and quantitative factors. In such a complex situation, we cannot expect to obtain high-precision results with a simple model. Therefore, a large amount of data processing and data standardization work is often carried out before prediction. However, this work is very complex and time-consuming. To simplify the process, we propose to vertically segment the input data according to attributes and function of variable, thus prevent the complex relationships and categories of variables from interfering with model training. This operation can be obtained through  $K$ -means clustering<sup>19</sup> by take feature variable as sample.

In  $K$ -means clustering, the initial clustering number  $K$  needs to be determined manually. To determine the number of  $K$ , we set  $K = \{3, 4, 5, 6, 7\}$  and evaluated the results using silhouette score. Meanwhile, to ensure the stability of the selected  $K$ , bootstrap method is used to conduct this process 50 times because of the inherent



**Fig. 3.** Frequency of distribution plots of categorical variables. Please refer to Supplementary file 2 for the names corresponding to the numbers of horizontal coordinate.

randomness of K-means clustering. The results are shown in Fig. 5, where the red line represents the average contour coefficient, and the blue and green lines represent the upper and lower bounds of the 95%CI, respectively.

As can be seen from the red line, the clustering result gets the best when  $K = 4$ , because its contour coefficient is the largest. Meanwhile, the blue line and the green line reflect that the clustering result is most stable when  $K = 4$ , because the distance between the blue line and the green line is the closest at this point. Then, the  $K$  which has the largest silhouette score was used as the final value. The results of the variable division are shown in Table 2.

These four types of variables respectively reflect the cost of construction projects from four aspects. The variables belonging to cluster1 are the general outline and foundation of cost prediction, determining the magnitude of the cost and usually positively correlated with it. Almost any cost prediction model must first take these variables into account to determine the basic range of costs. The variables belonging to cluster2 describe the connection mode between the building and the foundation, and they are the “foundation” of the project. Their costs belong to the concealed but crucial part and are an important source of cost fluctuations. In areas with complex geological conditions, the cost of foundation engineering may account for a large proportion of the total cost. Accurate description of the basic structure is crucial for prediction accuracy. The variables belonging to cluster3 determine the main construction and installation costs and construction technical plans, and are the key to distinguishing cost levels. The variables belonging to cluster4 directly affect the unit price of individual projects and are the focus of refined prediction and cost optimization. When constructing a cost prediction model, these four types of variables complement each other and none can be missing. An excellent model needs to integrate this information to accurately capture all cost drivers from macro to micro levels.

After vertical data segmentation, they will subsequently be respectively input into 4 stacking ensemble learning sessions for training. However, the type of model used in each stacking ensemble learning session is different, including those based on logistic regression, trees, neural networks, and so on.

#### Stacking ensemble learning

The dual-stacking model proposed in this paper is based on the traditional stacking ensemble learning<sup>20,21</sup>, in which variable stacking is added to fully extract the features required for construction cost prediction from different functional variables. Stacking ensemble learning is an effective ensemble method, in which predictions generated by various machine learning algorithms are used as input features for the second-layer learning algorithm. Then the second-layer algorithm, after training, can optimize the predictions of the combined model to form new predictions. Stacking generally consists of two layers: (a) a series of base model used to analyze data from multiple aspects; (b) the model obtained by taking the output of the base model as the training set, that is, the meta-model. The following is an introduction to these two parts respectively.

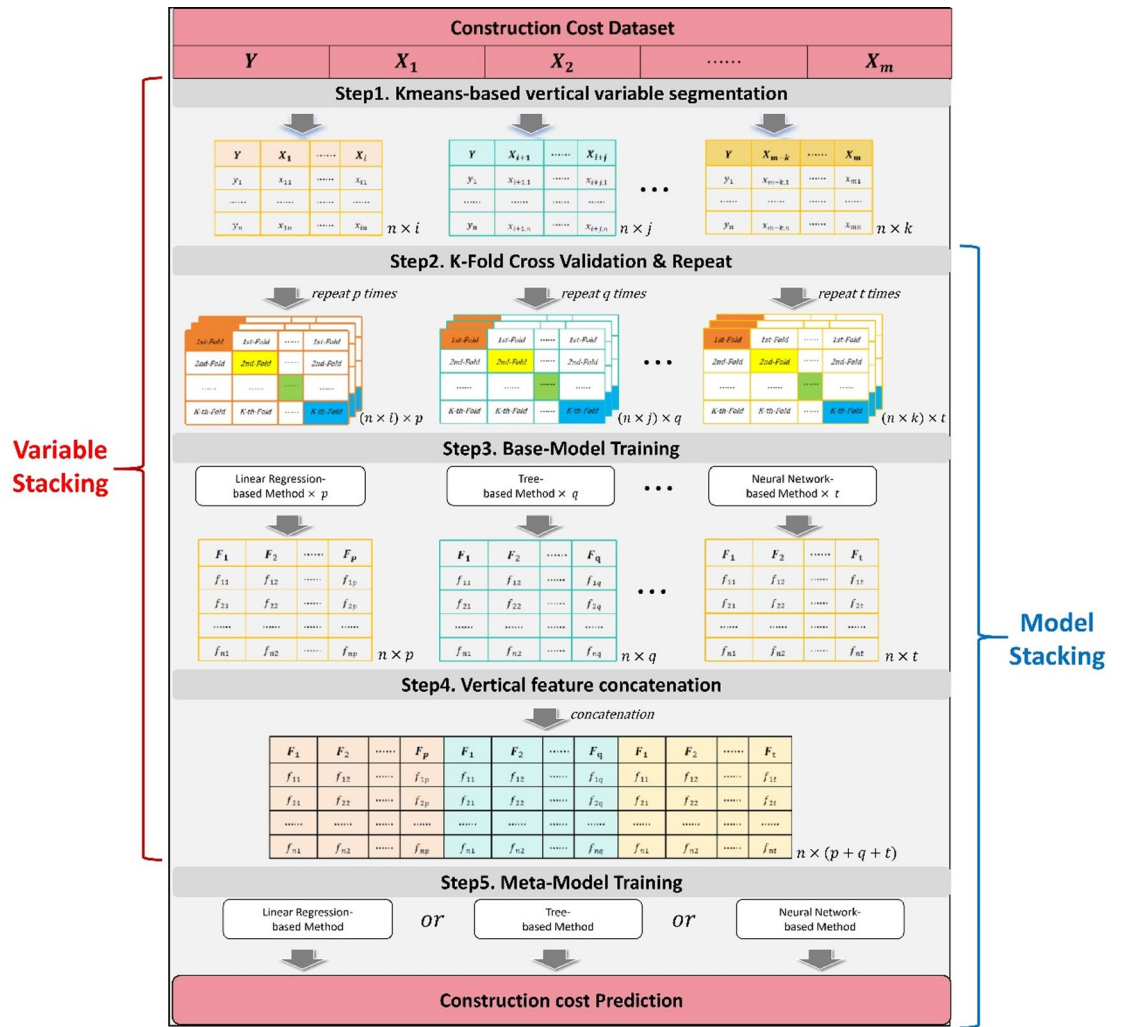


Fig. 4. The flowchart of DSCostPred.

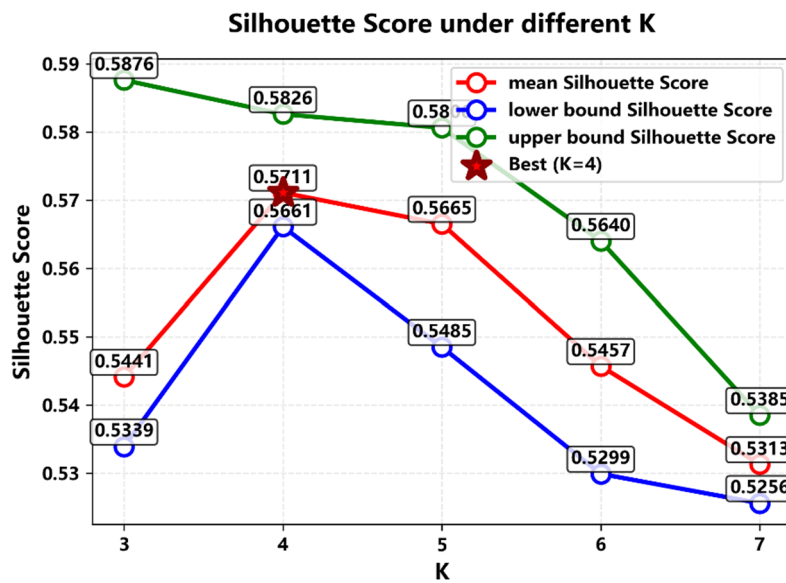


Fig. 5. The silhouette score under different K.

Clusters	Variables
Cluster1	Gross Floor Area ( $m^2$ ), Above-Ground Floor Area ( $m^2$ ), Below-Ground Floor Area ( $m^2$ ), Number of Floors ( $Floor$ ), Above-Ground Floors ( $Floor$ ), Below-Ground Floors ( $Floor$ ), Average Above-Ground Floor Height ( $m/Floor$ ), Average Below-Ground Floor Height ( $m/Floor$ ), Concrete Works ( $CNY/m^2$ ), Reinforcement Works ( $CNY/m^2$ ), Eaves Height ( $m$ ), Number of Elevators ( $Count$ )
Cluster2	Structural Type, Seismic Intensity ( $Level$ ), Civil Air Defense
Cluster3	Foundation Type, Earthwork, Foundation Pit Support, Masonry Works, Waterproofing Works, Fire Protection Works, Heating Works
Cluster4	Exterior Wall Material, Interior Finishing Material, Roof Decoration

**Table 2.** The result of vertical variable segmentation.

(1) Base model.

The main function of the base model is to learn the complex relationships in the data from multiple perspectives, extract and provide diverse and complementary “meta-features”, reduce the risk of overfitting in model training and improve the generalization ability of the model. The selection of base models is usually not simply about pursuing accuracy, but rather follows the principle of diversity, that is, to choose models with different operating principles as much as possible, such as linear models, tree models, kernel models, and neural networks. This is known as model stacking. Furthermore, since the data we used was split based on variable attributes and functions, we further optimized on the basis of model stacking and thus proposed variable stacking. In the previous training of base models, the input data for each set of base models was horizontally sliced data, and each set of data carried all variables indiscriminately. Then, a set of base models was used for training, which required the base models to have a strong learning ability. If the base model fails to learn effective features, then accurate results still cannot be achieved based on model stacking. With the assistance of variable stacking, the data is cut vertically (Table 2) to several parts, and each part is processed by different model. This not only enables diversified learning of various parts of the data but also allows for tailor-made solutions. This is known as variable stacking. For model selection, this paper adopts linear regression, ridge regression, LASSO regression, kernel ridge regression, random forest, and multi-layer perceptron. The following introduces them respectively:

(i) Linear regression.

For a linear equation  $Y = XW$ , the loss function of linear regression<sup>22</sup> is expressed as:

$$J(W) = \|Y - \hat{Y}\|^2 = (Y - XW)^T (Y - XW) \quad (1)$$

By setting the derivative to 0, the solution can get as follows:

$$W = (X^T X)^{-1} X^T Y \quad (2)$$

Here,  $W$  describes the influence weights of all variables in  $X$  to the construction cost. However, when there is multicollinearity among variables,  $X$  is a degenerate matrix, thus making it impossible to solve correctly.

(ii) Ridge regression.

Ridge regression<sup>23</sup> is an improved algorithm specifically designed to handle multicollinearity problems (where features are highly correlated), in which a regularization term  $\alpha \|W\|_2^2$  is added to the loss function:

$$J(W) = \|Y - \hat{Y}\|^2 + \alpha \|W\|_2^2 \quad (3)$$

The solution are as follows:

$$W = (X^T X + \alpha I)^{-1} X^T Y \quad (4)$$

where  $\alpha$  is the penalty coefficient used to adjust the penalty intensity of  $W$ .

(iii) LASSO regression.

LASSO regression<sup>24</sup> is another improved version with 1-norm  $\alpha \|W\|_1$  as regularization term:

$$J(W) = \|Y - \hat{Y}\|^2 + \alpha \|W\|_1 \quad (5)$$

Since this loss function is not continuously distinguishable, it can be solved by using the coordinate descent method or the minimum angle regression method.

(iv) Kernel ridge regression.

Kernel Ridge Regression (KRR)<sup>25</sup> introduces kernel techniques in ridge regression to solve the nonlinear problems. The objective function is as follows:

$$J(W) = \|Y - f(X)\|^2 + \alpha \|f\|_2^2 \tag{6}$$

where  $f = \sum_{i=1}^n w_i k(x_i, x_j)$ ,  $k(\cdot, \cdot)$  is the kernel function,  $w_i$  is the coefficient,  $x_i, x_j$  are the data samples. The solution is as follows:

$$W = (K + \alpha I) Y \tag{7}$$

where  $K$  is a matrix with dimension of  $n \times n$ ,  $K_{i,j} = k(x_i, x_j)$ .

(v) Random forest.

Random forest (RF)<sup>13</sup> is an ensemble learning method that enhances the predictive ability by combining multiple decision trees. The prediction result of RF is as follows:

$$f(x) = \frac{1}{k} \sum_{i=1}^k \varphi_i(x) \tag{8}$$

where  $\varphi_i(x)$  is the predicted result of decision tree. In regression tasks, the typically used decision tree is the Classification and Regression Tree (CART) with following loss function:

$$\min_{j,s} \left[ \min_{c_1} \sum_{x \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x \in R_2(j,s)} (y_i - c_2)^2 \right] \tag{9}$$

$$\begin{cases} R_1(j,s) = \{x | x^j \leq s\} \\ R_2(j,s) = \{x | x^j \geq s\} \\ c_1 = ave[y_i | x \in R_1(j,s)] \\ c_2 = ave[y_i | x \in R_2(j,s)] \end{cases} \tag{10}$$

where,  $x^j$  is the  $j$ -th variable;  $S$  is the  $x^j$  value that minimizes the sum of the squared errors of the two divided regions;  $R_1$  and  $R_2$  are the smallest division regions;  $c_1$  and  $c_2$  are the average values of the prediction results of the two regions respectively;  $y_i$  is the predicted result.

(vi) Multi-layer perceptron.

A multi-layer perceptron (MLP)<sup>26</sup> refers to a neural network with at least three layers (an input layer, a hidden layer, and an output layer), as shown in Fig. 6.

Theoretically, MLP can simulate any complex function by set enough layer and neuron. The following is the calculation formula on a certain neuron:

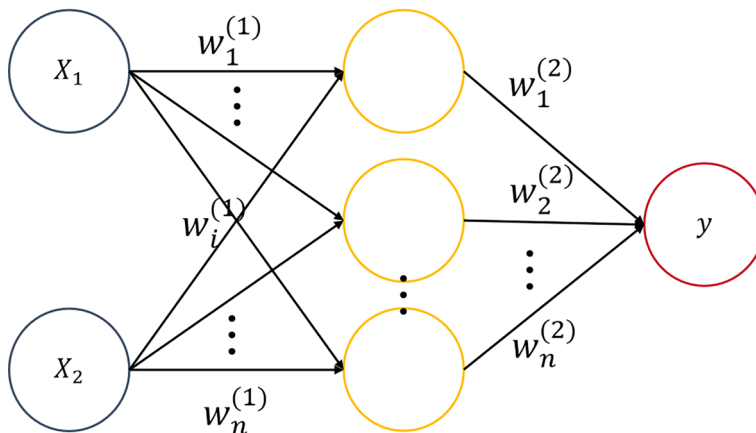


Fig. 6. Schematic diagram of MLP.

$$u = \sum_{i=1}^n w_i x_i + b \quad (11)$$

$$y = \text{activation}(u) \quad (12)$$

where  $x_i$  is  $i$ -th neuron's output in previous layer,  $w_i$  is the connection weight,  $b$  is the bias term,  $\text{activation}(\cdot)$  is the activation function endowing model with the ability of nonlinear mapping.

## (2) Meta model.

The purpose of the meta-model is to construct the relationship between the predicted results and the true results of multiple base models. It combines the advantages of the base models in the optimal way, make up for the deficiencies of individual models, and thereby achieve performance that surpasses any single model. In practice, simple and robust models are usually chosen as meta-models. In this article, we use the random forest as the meta-model because it naturally conforms to the role of meta-model, that is, to determine the final result through "swarm intelligence":

- (i) RF has low requirements for data distribution and does not need to standardize or normalize features. Because the tree model is based on threshold splitting, the scale change of features will not affect the splitting result, and thus can adapt to the differentiated output of different models;
- (ii) RF uses bootstrap sampling for training, further increasing the diversity of perspectives and forcing the model to learn more robust results;
- (iii) The features in the input data of the meta-model are independent of each other because they are obtained from different base models. If one wants to learn the nonlinear relationships and interaction effects among these features, it is also necessary to manually construct the interaction terms among these features. RF can naturally capture this complex nonlinear relationship and interaction effect between features and targets, because the decision tree in it essentially divides the feature space through a series of if-then rules.

## Results

In this section, we first compared it with other classic models to comprehensively evaluate the performance of DSCostPred. Then, we conducted an ablation experiment on it to verify the effectiveness of variable stacking and to explore the predictive effects under different base model and meta-model selections.

### Experimental setup

#### Training settings

In this study, the number of samples obtained for the experiment after preprocessing is 332. For this dataset, we divided it into a training set and a test set in an 8:2 ratio. Meanwhile, to avoid misunderstandings, we particularly note that the cross-validation in Step2 of flowchart is independent of the data partitioning here. Cross-validation in Step2 is to further divide the data and put input multiple base models. That is to say, in DSCostPred, during the training phase, the training set here is redivided, and during the testing phase, the test set here is also redivided.

To highlight the performance of the method, we compared it with several classic methods, including LSSVM, PCA-LSSVM, BSA-RF, PSO-BP, and CTGAN. To ensure the fairness of the comparison, we used their original parameters as much as possible, and for those using optimization algorithms, we explored their best performance on this data using the optimization algorithms (Supplementary file 4). All methods are trained and tested on the same training set and test set.

#### Evaluation metrics

The predicted values of regression models are often difficult to calculate precisely, so the key lies in demonstrating the closeness degree between the predicted values and the true values. This paper evaluates the model by using six indicators:  $R^2$ , mean absolute error (MAE), root mean square error (RMSE), mean absolute percentage error (MAPE), and symmetric mean absolute percentage error (sMAPE). Their formulas are as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \text{ where } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (13)$$

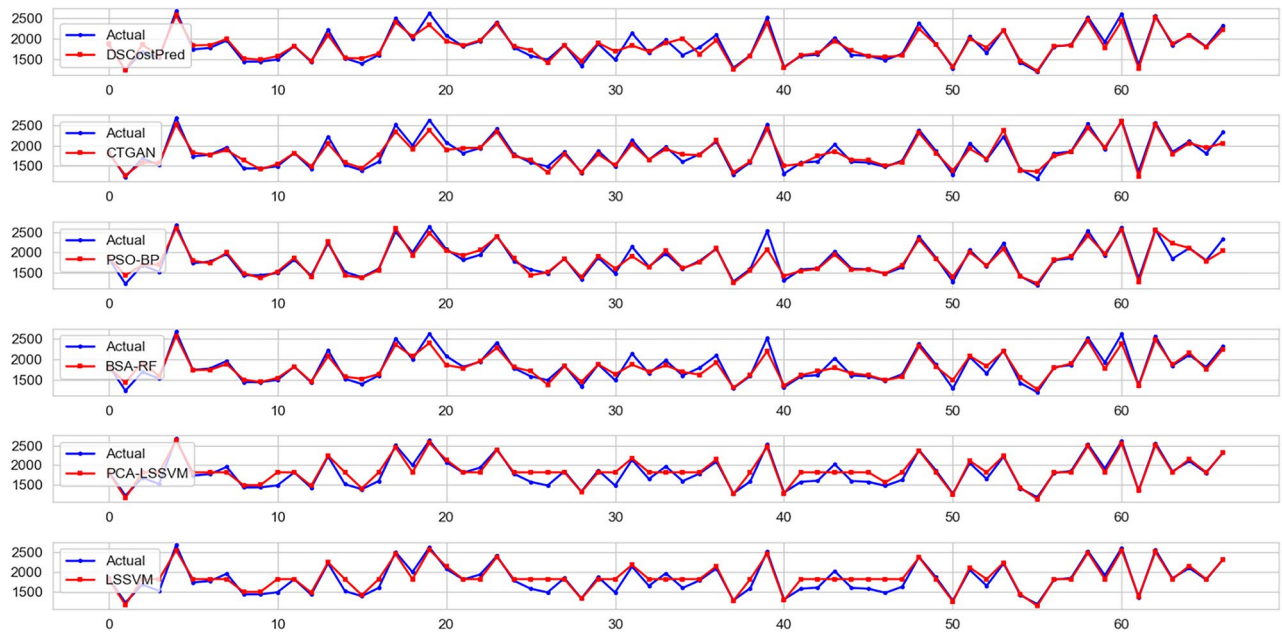
$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (14)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (15)$$

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (16)$$

Method	$R^2$	MAE	RMSE	MAPE	sMAPE
LSSVM	0.8577	104.8050	144.6685	6.2928	5.9803
PCA-LSSVM	0.8717	98.6708	137.3883	5.9101	5.6352
BSA-RF	0.9051	89.9029	118.1766	4.9337	4.9302
PSO-BP	0.9184	65.3063	109.5734	3.4279	4.3633
CTGAN	0.9143	88.6325	112.2731	4.8106	4.9134
<b>DSCostPred</b>	<b>0.9197</b>	<b>78.2025</b>	<b>108.7037</b>	<b>4.2848</b>	<b>4.2492</b>

**Table 3.** The results of comparative experiment. Significant values are in [bold].



**Fig. 7.** Fitting result.

$$sMAPE = \frac{100}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{(|y_i| + |\hat{y}_i|) / 2} \quad (17)$$

where,  $R^2$  is used to measure the degree about a model fits the data. Its value range is usually between  $-\infty$  and 1. The closer  $R^2$  is to 1, the better the model fitting effect. MAE is used to measure fitting errors, is not sensitive to outliers, and is affected by dimensions. It should be noted that since the prediction target of this study is the unit construction cost, the calculation of MAE is directly based on the unit area cost (CNY/m<sup>2</sup>), and is not affected by the scale of the project, thus can be directly used for project interpretation. RMSE is used to measure performance with large errors and is sensitive to outliers. MAPE can represent errors in the form of relative proportions, which is convenient for intuitive understanding. sMAPE resolves the issue that MAPE imposes a heavier penalty for negative errors (predicted values > true values) than for positive errors.

### Comparative experiment

To highlight the advantages of DSCostPred, we compared it with several classic models (LSSVM, PCA-LSSVM, BSA-RF, PSO-BP, CTGAN) on  $R^2$ , MAE, RMSE, MAPE and sMAPE. The results are shown in Table 3.

It can be seen from the table that our method ( $R^2 = 0.9197$ , RMSE = 108.7037, sMAPE = 4.2492) achieved the best result on  $R^2$ , RMSE, and sMAPE, and was second only to PSO-BP on MAE and MAPE (DSCostPred: MAE = 78.2025, MAPE = 4.2848; PSO-BP: MAE = 65.3063, MAPE = 3.4279). This indicates that compared with PSO-BP, our method will have inaccurate predictions on more data points, but the error range is smaller. However, although PSO-BP only makes inaccurate predictions at a few data points, once it makes a wrong judgment, it will cause a large error, which might be intolerable for construction projects. Meanwhile, we have noticed that the positions of the two methods on MAPE and sMAPE are inconsistent. It is well known that MAPE tends to accommodate methods where the predicted values are generally lower than the true values, while sMAPE avoids such a result. Although the MAPE of PSO-BP is relatively low, we know that if the estimated cost of the project is too low, it will lead to the project having to be interrupted due to lack of funds, which is also unacceptable. All these indicate that our method might be more practical. Figure 7 shows the prediction results

(Since the data is not time series data, for Fig. 7, we only discuss the differences between the predicted values and the true values at each data point, not the trend of the curve). We can observe that the number of data points with predicted values less than the true values in PSO-BP is smaller than that in DSCostPred, and there are more data points with huge errors, which is consistent with our analysis.

Furthermore, to further comprehensively reflect the performance of each method, we additionally conducted paired bootstrap, which calculates the evaluation metrics by randomly sampling the data with replacement, and finally the average value and 95% confidence intervals (95%CI) of the evaluation metrics can be obtained. This process of multiple random samplings is equivalent to disrupting the order and size of the data, and the results obtained through multiple samplings are more convincing than point estimates. Specifically, through a resampled test set sample with replacement (1000 times), we constructed the sampling distribution of  $R^2$ , MAE, RMSE, MAPE, sMAPE and calculated their 95% confidence intervals (95%CI) and means. This method ensures that our evaluation of the model performance is unbiased and statistically robust. The results are shown in Table 4.

It can be known from the table that DSCostPred still achieved outstanding performance advantages (mean  $R^2$  = 0.9160, mean MAE = 73.8160, mean RMSE = 110.0667, mean MAPE = 4.0641, mean sMAPE = 4.0463). It was only slightly inferior to BSA-RF and CTGAN in mean  $R^2$  and mean RMSE (BSA-RF: mean  $R^2$  = 0.9187, mean RMSE = 108.3997; CTGAN: mean  $R^2$  = 0.9241, mean RMSE = 105.0371). However, considering the 95%CI, the BSA-RF were  $R^2$  (95% CI) = [0.8890, 0.9498] and RMSE (95% CI) = [86.0059, 124.0570], the CTGAN were  $R^2$  (95% CI) = [0.9054, 0.9408] and RMSE (95% CI) = [93.2591, 115.1987], while ours were  $R^2$  (95% CI) = [0.8790, 0.9521] and mean RMSE = [84.1785, 128.7718]. Our upper bound of  $R^2$  (95% CI) is greater than BSA-RF and CTGAN, and the lower bound of RMSE (95% CI) is less than BSA-RF and CTGAN, which indicates that our method is more likely to achieve higher performance. In conclusion, although DSCostPred is inferior to BSA-RF and CTGAN in mean  $R^2$  and mean RMSE, the gap is not significant. Its comprehensive performance on other metrics still holds a prominent position, which indicates that DSCostPred has certain statistical superiority.

In addition, it should be noted that since the bootstrap process involves extracting different numbers of samples which come from different regions. This process is also a process that contains noise and bias. Our method can achieve a good performance in this situation, indicating that it may also be able to resist the influence of noise and bias to a certain extent.

### Ablation experiment

To investigate the effectiveness of variable stacking and explore the predictive effects under different base model and meta-model selections, we compared DSCostPred with its variant versions. These variant versions include:

- (1) noVS: This version has removed the variable stacking and uses the original stacking ensemble learning. The input data has also been restored from the vertically cut data to the original horizontally cut data;
- (2) BaseModel-LR: This version removes linear regression from base models;
- (3) BaseModel-RR: This version removes ridge regression from base models;
- (4) BaseModel-LASSO: This version removes LASSO regression from base models;
- (5) BaseModel-KRR: This version removes kernel ridge regression from base models;
- (6) BaseModel-RF: This version removes RF from base models;
- (7) BaseModel-MLP: This version removes MLP from base models;
- (8) Metamodel-LR: This version replaces the meta-model used in DSCostPred with linear regression;
- (9) Metamodel-MLP: This version replaces the meta-model used in DSCostPred with MLP.

Among these variants, (1) replaces the variable stack with the original input method of stacking ensemble learning, with the aim of verifying the validity of the variable stacking; (2) to (7) deletes the corresponding base learners with the aim of verifying the importance of each base learner in the model stacking. (8) to (9) replaced the meta-learner with other classic models to demonstrate the necessity of choosing RF as the meta-learner. The more the effect drops, the greater the contribution of the deleted module in that variant. The results of the ablation experiment are shown in Table 5. Meanwhile, to clarify the contribution of each part to the model,

Method	mean $R^2$ 95%CI	mean MAE 95%CI	mean RMSE 95%CI	mean MAPE 95%CI	mean sMAPE 95%CI
LSSVM	0.8554 [0.7973,0.9050]	104.7040 [88.3549,120.8911]	144.4128 [124.6560,161.7003]	6.2902 [5.1999,7.3377]	5.9767 [4.9897,6.9354]
PCA-LSSVM	0.8697 [0.8154,0.9135]	98.5215 [82.3860,112.5948]	137.0294 [117.6224,153.3676]	5.9031 [4.8451, 6.8284]	5.6281 [4.6483, 6.4732]
BSA-RF	0.9187 [0.8890, 0.9498]	78.1070 [65.3080, 89.3942]	108.3997 [86.0059, 124.0570]	4.2777 [3.5395, 4.9006]	4.2423 [3.5227, 4.8375]
PSO-BP	0.9043 [0.8761, 0.9292]	89.7321 [76.5828, 100.7795]	117.8808 [101.7653,130.4516]	4.9232 [4.2435, 5.5306]	4.9197 [4.2613, 5.5032]
CTGAN	0.9241 [0.9054, 0.9408]	83.9156 [73.3299, 92.8960]	105.0371 [93.2591, 115.1987]	4.6793 [4.0655, 5.2293]	4.6570 [4.0631, 5.1889]
<b>DSCostPred</b>	<b>0.9160</b> <b>[0.8790, 0.9521]</b>	<b>73.8106</b> <b>[60.2145, 86.6098]</b>	<b>110.0667</b> <b>[84.1785, 128.7718]</b>	<b>4.0641</b> <b>[3.3813, 4.7255]</b>	<b>4.0463</b> <b>[3.3615, 4.6843]</b>

**Table 4.** The results of comparative experiment under paired bootstrap. Significant values are in [bold].

Method	R <sup>2</sup>	MAE	RMSE	MAPE	sMAPE	SHAP
DSCostPred	<b>0.9197</b>	<b>78.2025</b>	<b>108.7037</b>	<b>4.2848</b>	<b>4.2492</b>	--
noVS	0.8931	93.1625	125.4364	5.0732	5.0822	0.0781
BaseModel-LR	0.9054	86.7039	117.9678	4.8057	4.7640	0.0385
BaseModel-RR	0.9025	87.4828	119.7969	4.8420	4.7989	0.0463
BaseModel-LASSO	0.9026	87.3423	119.6895	4.8160	4.7621	0.0460
BaseModel-KRR	0.9092	86.1443	115.5498	4.6947	4.6773	0.0283
BaseModel-RF	0.6953	163.3338	211.7348	8.8251	8.7291	0.6043
BaseModel-MLP	0.9128	84.5581	113.2543	4.6489	4.6199	0.0186
MetaModel-LR	0.8627	115.3173	142.1287	6.3547	6.6162	0.0837
MetaModel-MLP	0.8815	105.4832	132.0285	5.9215	5.8830	0.0561

**Table 5.** The results of ablation experiment. Significant values are in [bold].

we also calculated the marginal benefits of each part using SHAP values, which are listed in the last column of Table 5. When calculating the SHAP value, we take the R<sup>2</sup> of DSCostPred as the benchmark, and the absolute value of the difference between the R<sup>2</sup> obtained after removing a certain part and the R<sup>2</sup> of DSCostPred as the gain.

From Table 5, it can be seen that variable stacking is an effective improvement, as noVS (R<sup>2</sup>=0.8931, MAE=93.1625, RMSE=125.4364, MAPE=5.0732, sMAPE=5.0822) shows significantly worse performance than DSCostPred (R<sup>2</sup>=0.9197, MAE=78.2025, RMSE=108.7037, MAPE=4.2848, sMAPE=4.2492). Specifically, the R<sup>2</sup> of noVS is clearly lower, while its MAE, RMSE, MAPE and sMAPE are all higher, indicating that variable stacking substantially enhances prediction accuracy. Moreover, it can also be seen from the SHAP column that, except for BaseModel-RF, the order of magnitude of SHAP contributions in other parts is the same, and noVS ranks first among them, which proves the effectiveness of the variable stacking we proposed. The reason why it is effective might be that it adds more diverse options and more samples to the model. These two advantages are closely related to the vertical variable segmentation and the repeat operation in Step2, which is precisely one of the core operations of variable stacking.

Then, for the ablation of the base model, we can see that the selection of RF is very important because the performance of BaseModel-RF drops significantly (R<sup>2</sup>=0.6953, MAE=163.3338, RMSE=211.7348, MAPE=8.8251, sMAPE=8.7291). This is because RF is suitable for learning both continuous and categorical variables, so the internal features of the data can be learned in any group. Meanwhile, we noticed that the performance of BaseModel-MLP (R<sup>2</sup>=0.9128, MAE=84.5581, RMSE=113.2543, MAPE=4.6489, sMAPE=4.6199) showed almost no decline. In this study, we believe there might be two reasons: (i) Insufficient model training. MLP requires a large amount of data for training to fully exert its function, but the data we use is only a few hundred, which cannot meet the requirements of model training. (ii) The role of MLP can be replaced by the combination of other models. The greatest advantage of MLP is that it can fit complex nonlinear mappings, but KRR and RF also have this advantage. Therefore, even if MLP is removed, the missing part can still be compensated by KRR and RF.

Finally, for the meta-model, we did not ablate all models but only selected LR and MLP as representatives. It can be seen that after changing the meta-model from RF to LR and MLP, the fitting results have decreased significantly, which indicates that using RF as the meta-model is a suitable choice.

Meanwhile, to make the performance comparison more convincing, we also conducted experiments and evaluations using paired bootstrap. The results are shown in Table 6.

As can be seen from Table 6, even under strict evaluation conditions, the results of the ablation experiment still support the previous conclusion, which further proves that each part of our model is effective. In conclusion, the addition of variable stacking is effective. Compared with the traditional stacking ensemble learning, variable stacking can increase the accuracy of cost prediction. This also confirms that it is very important for us to perform vertical segmentation on the data and handle different types of variables separately using different models. Meanwhile, the selection of the base model and the meta-model is effective, and RF plays a crucial role in both the base model and the meta-model.

### Feature importance analysis using SHAP analysis

To capture the influence of variables on the unit construction cost prediction, SHAP analysis is used to calculate the SHAP value of all variables. This kind of analysis is crucial for identifying which variables have the greatest impact on the results, enhancing the understanding of complex correlations in the dataset and providing a valuable perspective on the internal operation of the model. This in-depth analysis of variable importance adds additional transparency to the model's decision-making process.

According to the SHAP analysis, we provided the importance representations of all variables, and the results are shown in Fig. 8. From this, it can be known that variables such as "Concrete Works" and "Reinforcement works" are the most important, which is consistent with reality. "Concrete works" and "Reinforcement works", as the structural skeleton and cost core of buildings, directly determine the strength, stability and seismic resistance capacity of buildings. Any design changes involving structural types, seismic grades, and floor heights will directly and significantly translate into the increase or decrease of "Concrete works" and "Reinforcement works". Such results show that our model has obtained results consistent with reality. "Exterior Wall Material",

Method	mean R <sup>2</sup> 95%CI	mean MAE 95%CI	mean RMSE 95%CI	mean MAPE 95%CI	mean sMAPE 95%CI
<b>DSCostPred</b>	<b>0.9160</b> [0.8790, 0.9521]	<b>73.8106</b> [60.2145, 86.6098]	<b>110.0667</b> [84.1785, 128.7718]	<b>4.0641</b> [3.3813, 4.7255]	<b>4.0463</b> [3.3615, 4.6843]
noVS	0.8918 [0.8468, 0.9287]	93.0272 [79.3158, 106.5625]	124.9999 [105.6811, 140.5020]	5.0650 [4.2742, 5.7833]	5.0748 [4.3154, 5.7822]
BaseModel-LR	0.9043 [0.8731, 0.9324]	86.7430 [73.7710, 98.2825]	117.7272 [99.6173, 131.8137]	4.8064 [4.1139, 5.4833]	4.7660 [4.1118, 5.4114]
BaseModel-RR	0.9014 [0.8693, 0.9317]	87.4931 [74.8363, 99.4459]	119.5109 [101.1635, 134.1561]	4.8412 [4.1432, 5.5117]	4.7993 [4.1355, 5.4521]
BaseModel-LASSO	0.9017 [0.8698, 0.9322]	87.3652 [73.9435, 99.3307]	119.3554 [99.6716, 134.3912]	4.8153 [4.0885, 5.4846]	4.7630 [4.0594, 5.3968]
BaseModel-KRR	0.9083 [0.8772, 0.9351]	86.1360 [73.5236, 97.9445]	115.3182 [98.3442, 128.7968]	4.6922 [4.0953, 5.3036]	4.6762 [4.0591, 5.2862]
BaseModel-RF	0.6922 [0.6146, 0.7676]	163.5837 [141.0519, 187.2785]	211.6943 [180.3030, 238.9372]	8.8385 [7.7600, 10.0242]	8.7415 [7.6538, 9.8842]
BaseModel-MLP	0.9121 [0.8843, 0.9383]	84.5187 [72.5730, 95.8079]	112.9000 [95.1871, 126.3889]	4.6452 [3.9956, 5.2353]	4.6176 [3.9907, 5.1898]
MetaModel-LR	0.8609 [0.8233, 0.8913]	115.4187 [101.7302, 128.2040]	142.1844 [125.1481, 156.2966]	6.3614 [5.7183, 6.9921]	6.6243 [5.9314, 7.3047]
MetaModel-MLP	0.8801 [0.8463, 0.9071]	105.6289 [93.3731, 117.9452]	132.0458 [115.3295, 147.0451]	5.9270 [5.2624, 6.6057]	5.8896 [5.2378, 6.5491]

**Table 6.** The results of ablation experiment under paired bootstrap. Significant values are in [bold].



**Fig. 8.** The SHAP value of all variables.

“Interior Finishing Material” and “Heating Works”, are at the top of the list, which is also consistent with reality. They have a huge impact on the unit construction cost because of their highly cost elasticity. The choice of “Exterior Wall Material” and “Interior Finishing Material” can have a difference of several times or even tens of times on the unit cost, thus be the key variable for cost control. Heating projects (such as floor heating and

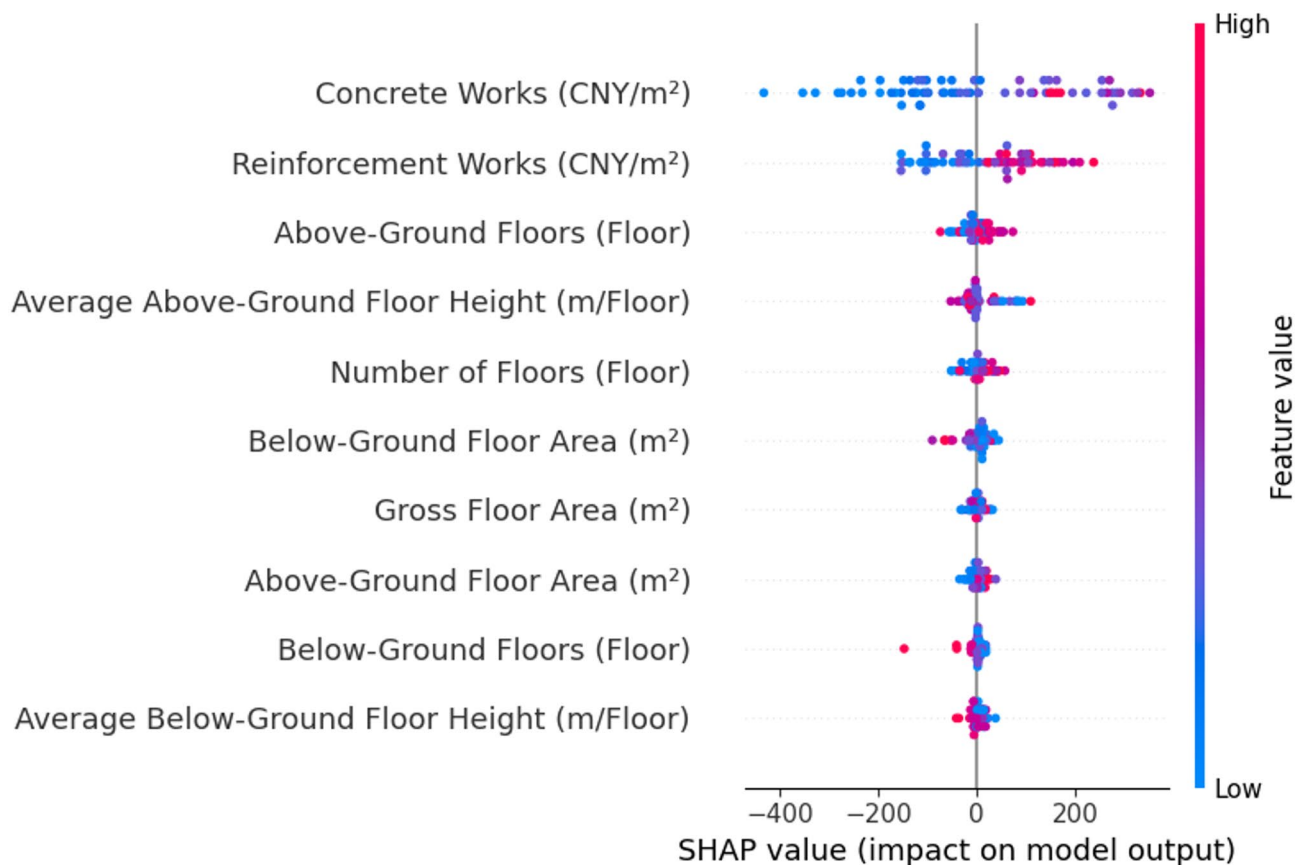
central air conditioning) are costly. Their expenses increase sharply with the complexity of the system, the brand of equipment and energy-saving standards, and they are essential functional expenditures. Therefore, they are the core for balancing cost, quality and selling price, and have a significant and flexible impact on the unit cost.

Meanwhile, we also observed that “Above/Below-Ground Floor Area”, “Number of Floors”, “Above/Below-Ground Floors” ranked in the middle since their influence is systematic rather than direct. Their minor changes can trigger a significant response in cost. For instance, if the floor height increases by 10 centimeters, all vertical structures and exterior wall materials will increase. Such characteristics determine that they will have an impact on the final cost. Finally, we also found that “Earthwork”, “Foundation Pit Support”, “Masonry Works”, “Waterproofing Works” ranked lower because their unit engineering cost is relatively stable (unless extreme geological conditions were encountered). In conclusion, through the SHAP analysis of the variables, it has been effectively proved that our model has certain reliability and interpretability.

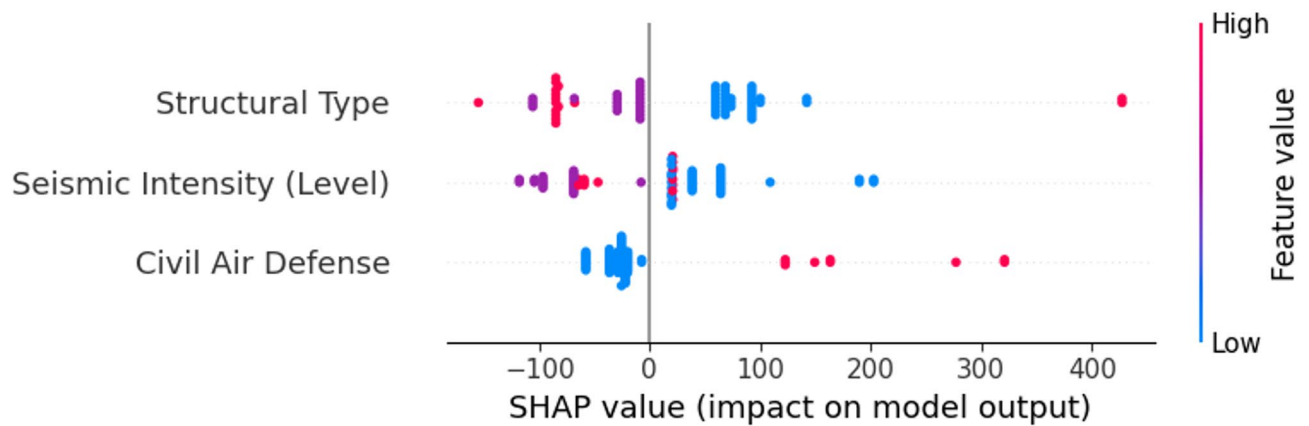
Finally, to enhance the interpretability, we also provide the SHAP analysis results of each variable in four clusters (Figs. 9, 10, 11 and 12).

## Discussion and conclusion

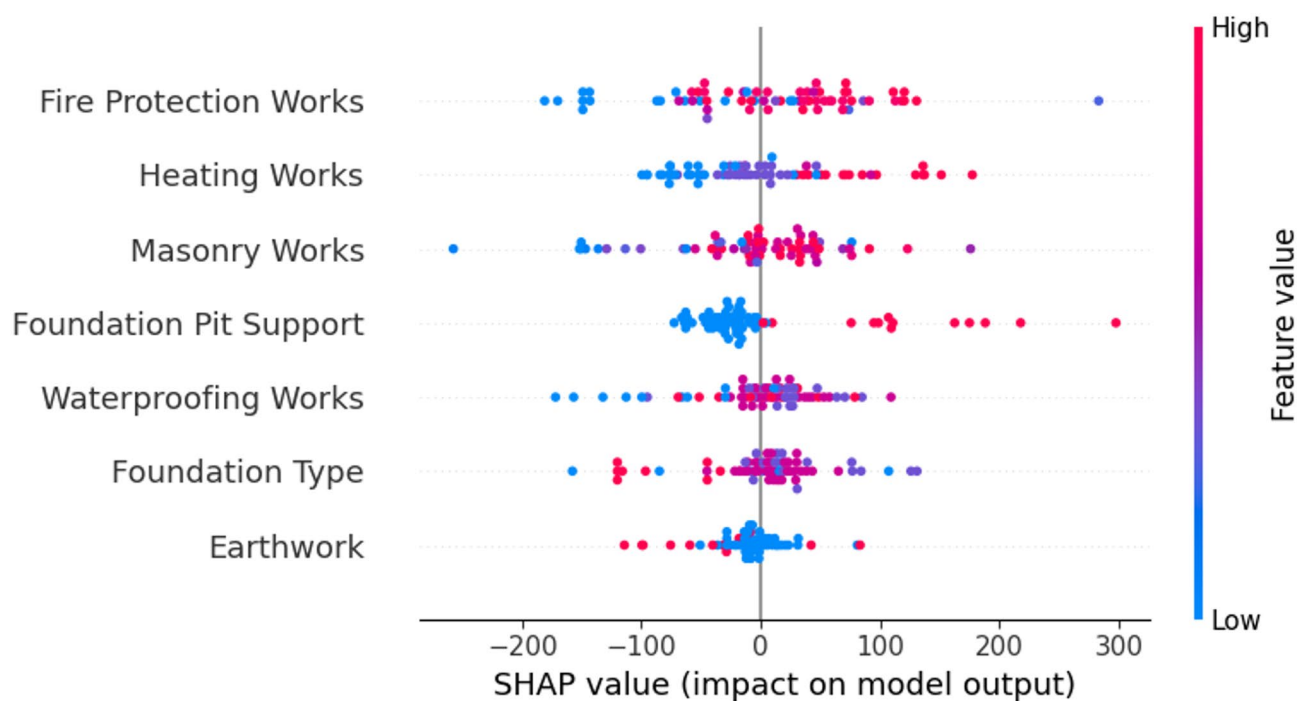
The prediction of construction project cost plays a crucial and core role in engineering construction projects. Before the project is initiated, accurate cost prediction serves as the primary basis for investment decisions, directly determining the economic feasibility of the project and providing a crucial economic judgment criterion for the comparison of different schemes. However, the current prediction of construction project costs involves a multi-dimensional and dynamically variable system, and each major category can be further subdivided into countless specific factors. Meanwhile, the relationships among variables are far from being simple superpositions; rather, they present a complex network of nonlinearity and interaction. This has seriously affected the accuracy of the prediction. To solve this problem, we proposed a dual-stacking project cost prediction method DSCostPred based on variable stacking and model stacking. This method emphasizes that classifying variables based on their functions and attributes and applying different algorithms respectively can avoid the functional differences of variables and the impact brought by complex system interactions. First, pre-classify the variables based on their functions and attributes to avoid mutual interference among variables with different attributes. Then, to learn the attribute and function positioning of variables, as well as the complex interaction among them, different types of models are utilized to learn the variables under different attributes and functions. In terms of algorithm design, to achieve the organic combination of multiple attributes and multiple models in the system, we have incorporated the variables stacking into the stacking ensemble learning.



**Fig. 9.** The SHAP value of variables in cluster 1.



**Fig. 10.** The SHAP value of variables in cluster 2.

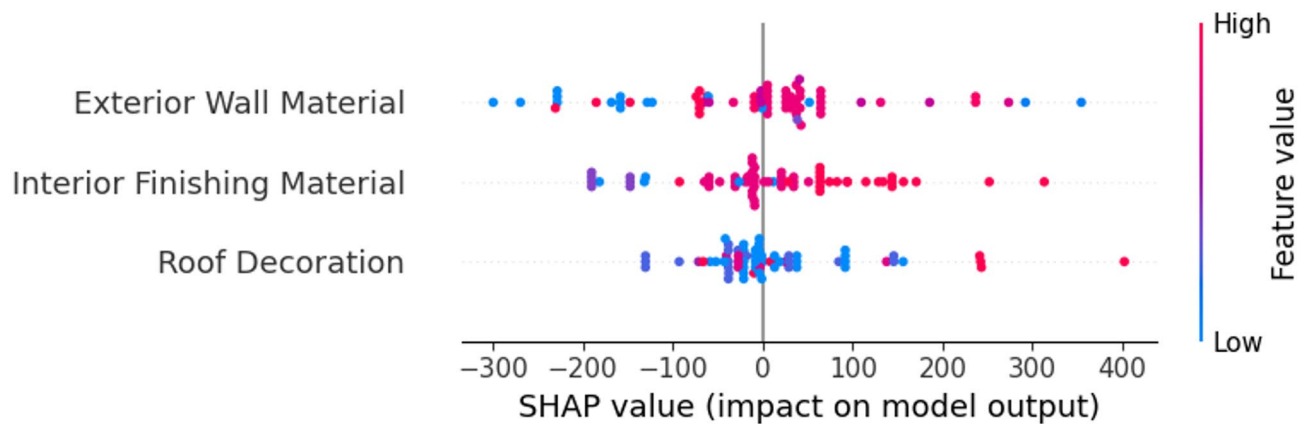


**Fig. 11.** The SHAP value of variables in cluster 3.

This method was experimented on real data. Compared with methods such as LSSVM, PCA-LSSVM, BAS-RE, and PSO-BP, DSCostPred achieved better results. This proves that the method we proposed is effective. In addition, we also conducted ablation experiments and SHAP analysis about variable stacking and model stacking. The experimental results show that variable stacking has a significant improving effect on the prediction of construction project costs. In model stacking, the role of RF is crucial, which is consistent with reality. The variables involved in the task of construction project cost prediction include both discrete and continuous variables, and it is not convenient to be standardized. The RF, as a tree model, splits the tree based on the numerical order of features. Therefore, the numerical size and distribution of features have no impact on model training, and thus are naturally adapted to unstandardized data and discrete data. In conclusion, all these factors are the basis for the superior performance of DSCostPred.

Although the comparison with a series of classic methods, as well as the results of ablation experiments and SHAP interpretability analysis, have demonstrated that DSCostPred is a model with outstanding performance. However, it also has some limitations, which are reflected in two aspects:

- (1) It does not consider the impact of inflation and currency conversion. In the data used in this article, we only focus on the Chinese region, and the data used are from 2019 to 2022. In fact, from 2019 to 2022, although the cost index of China's engineering and construction market experienced phased fluctuations, it remained



**Fig. 12.** The SHAP value of variables in cluster 4.

within a controllable range overall, and the macro price level also remained stable. Therefore, changes in price levels were not sufficient to affect cross-year cost comparisons or model analysis results. Meanwhile, the research did not involve any cross-currency or currency system conversion at the same time. All data were directly statistically analyzed using the RMB values provided by the domestic engineering cost system. This has led to the possibility that we might have overlooked the impacts related to inflation and currency conversion when designing the model;

- (2) It does not consider the impact of different contracts (project types) or regions. In fact, to test the impact of regions and contracts on model performance, we collected 20 residential projects records from Hunan, China (not included in the training set) (with the same contract but different regions), as well as 53 teaching building project data from all over the country (with different contracts). The results obtained on 20 data are  $R^2=0.9181$ , MAE = 43.7996, RMSE = 50.1480, MAPE = 2.5212, sMAPE = 2.5185, and on 53 data are  $R^2=0.6610$ , MAE = 135.7463, RMSE = 187.2793, MAPE = 6.9371, sMAPE = 6.9660. This result indicates that our model has excellent generalization for the prediction of the same contract type in different regions, while its predictive ability for different types of contracts drops sharply. In fact, this is obvious. Because our model was trained based on national residential data, and the new data is based on teaching buildings. There are structural differences between these two types of buildings in terms of fundamental use, design standards and functional requirements, which leads to completely different key features and their weights that determine their costs.

In the future, we will be committed to exploring these two aspects. It would be an attractive outcome if the construction costs of more diverse types of projects could be predicted. In the future, we will attempt to collect data from various types of contracts around the world and try to use larger datasets to train our model in order to achieve more comprehensive performance.

### Data availability

The data supporting this study's findings are available on request from the corresponding author.

Received: 6 November 2025; Accepted: 17 December 2025

Published online: 21 December 2025

### References

- Ghaddhan Abed, Y., Hasan, T. M. & Zehawi, R. N. Machine learning algorithms for constructions cost prediction: a systematic review. *Int. J. Nonlinear Anal. Appl.* **13**, 2205–2218 (2022).
- Tayefeh Hashemi, S., Ebadati, O. M. & Kaur, H. Cost Estimation and prediction in construction projects: a systematic review on machine learning techniques. *SN Appl. Sci.* **2**, 1703 (2020).
- Castro Miranda, S. L., Del Rey Castillo, E., Gonzalez, V. & Adafin, J. Predictive analytics for early-stage construction costs estimation. *Buildings* **12**, 1043 (2022).
- AlTalhoni, A., Liu, H. & Abudayyeh, O. Forecasting construction cost indices: Methods, trends, and influential factors. *Buildings* **14**, 3272 (2024).
- Tipu, R. K. & Suman, Batra, V. Development of a hybrid stacked machine learning model for predicting compressive strength of high-performance concrete. *Asian J. Civil Eng.* **24**, 2985–3000 (2023).
- Tipu, R. K. et al. Predictive modelling of surface chloride concentration in marine concrete structures: a comparative analysis of machine learning approaches. *Asian J. Civil Eng.* **25**, 1443–1465 (2024).
- Tipu, R. K. et al. Optimizing sustainable blended concrete mixes using deep learning and multi-objective optimization. *Sci. Rep.* **15**, 16356 (2025).
- Fan, M. & Sharma, A. Design and implementation of construction cost prediction model based on SVM and LSSVM in industries 4.0. *Int. J. Intell. Comput. Cybern.* **14**, 145–157 (2021).
- Wang, Y. & Sun, N. Cost estimation in residential building project based on LSSVM with PCA. *Proc. 2024 Int. Conf. Big Data Digit. Manag* **1**, 1–8 (2024).
- Yue, W. Research on optimization of construction project cost control in construction enterprises. *Acad. J. Bus. Manag.* **4**, 89–93 (2022).

11. Liu, M. & Chen, H. Research on construction efficiency based on improved PCA-BP neural network model. *Proc. 2024 Int. Conf. Electron. Devices Comput. Sci.* **1**, 354–360 (2024).
12. Feng, G. L. & Li, L. Application of genetic algorithm and neural network in construction cost estimate. *Adv. Mater. Res.* **756**, 3194–3198 (2013).
13. Zheng, Z., Zhou, L., Wu, H. & Zhou, L. Construction cost prediction system based on random forest optimized by the bird swarm algorithm. *Math. Biosci. Eng.* **20**, 15044–15074 (2023).
14. Shi, T. & Shide, K. A comparative analysis of LSTM, GRU, and transformer models for construction cost prediction with multidimensional feature integration. *J Asian Archit. Build. Eng* **2025**, 1–16 (2025).
15. Li, B., Xin, Q. & Zhang, L. Engineering cost prediction model based on DNN. *Sci. Program* **2022**, 3257856 (2022).
16. Feng, N. Intelligent estimation of construction project costs based on subtractive clustering-based self-learning convolutional neural network. *J. Combin. Math. Combin. Comput.* **127**, 7321–7336 (2025).
17. Liu, H., Li, M., Cheng, J. C., Anumba, C. J. & Xia, L. Actual construction cost prediction using hypergraph deep learning techniques. *Adv. Eng. Inf.* **65**, 103187 (2025).
18. Hong, E., Yi, J. S. & Lee, D. CTGAN-based model to mitigate data scarcity for cost estimation in green building projects. *J. Manag. Eng.* **40**, 04024024 (2024).
19. Singh, R. V. & Bhatia, M. S. Data clustering with modified K-means algorithm. In *Proc. 2011 Int. Conf. Recent Trends Inf. Technol.* 717–721 (2011).
20. Güneş, F., Wolfinger, R. & Tan, P. Y. Stacked ensemble models for improved prediction accuracy. *Proc. Static Anal. Symp.* **2017**, 1–19 (2017).
21. Tipu, R. K. et al. Ensemble machine learning models for predicting concrete compressive strength incorporating various sand types, multiscale and multidisciplinary modeling. *Exp. Des.* **8**, 222 (2025).
22. Lin, L., Jiang, W., Chen, B., Yu, J. & Zheng, C. Construction and application of cost prediction model based on multiple linear regression analysis. *Procedia Comput. Sci.* **247**, 617–623 (2024).
23. Mohamad, J. et al. A simulation study of some logistic, Poisson, and multiple ridge regression estimators. *Eur. J. Pure Appl. Math.* **18**, 6162–6162 (2025).
24. Ranstam, J. & Cook, J. A. LASSO regression. *J. Br. Surg.* **105**, 1348 (2018).
25. Pereira, A., Moreno, S., Rodríguez, J. & Villalba, M. Kernel ridge regression estimation of high-frequency signals for nuclear fusion diagnostics. *Fusion Eng. Des.* **219**, 115261 (2025).
26. Tipu, R. K. et al. Efficient compressive strength prediction of concrete incorporating recycled coarse aggregate using newton's boosted backpropagation neural network (NB-BPNN). *Structures* **58**, 105559 (2023).

### Author contributions

CPL and XGS initiated this study and designed algorithms and experiments. JHG performed the experiments, analyzed the results, and drafted the manuscript. CPL and JHG revised the manuscript. All authors have read and approved the final manuscript.

### Funding

This work was supported by Hunan Provincial Natural Science Foundation of China (Grant number: 2023JJ60199).

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-33305-y>.

**Correspondence** and requests for materials should be addressed to J.-H.G.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025