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

Optimizing CNN, SVM, and MLP for Prediction of Compressive Strength of Concrete using Grid Search Optimization

Raghvendra Kumar[1], Hoang Ha[2], Nguyen Duc Son[3], Long Hoang Nguyen[4,*]
[1]Department of Computer Science and Engineering, Gandhi Institute of Engg. & Technology University Gunupur, Odisha and 765022, India
[2] University of Transport and Communications, Lang Thuong, Dong Da, Hanoi, Vietnam
[3]Geotechnical and Artificial Intelligence research group, University of Transport Technology, 54 Trieu Khuc, Thanh Liet, Hanoi 100000, Vietnam
[4]University of Transport Technology, 54 Trieu Khuc, Thanh Liet, Hanoi 100000, Vietnam

Article info
Type of article:
Original research paper

DOI:
https://doi.org/10.58845/jstt.utt.2025.en.5.4.198-217

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

Received: 21/09/2025
Received in Revised Form: 24/11/2025
Accepted: 14/12/2025

Abstract: In civil engineering, the accurate prediction of concrete compressive strength (CS) is crucial for evaluation of material performance and structural design. In the present study, the main objective is to optimize the performance of three machine learning (ML) models including Support Vector Machine (SVM), Convolutional Neural Network (CNN), Multi-Layer Perceptron Neural Network (MLP) using Grid Search Optimization technique for improving the prediction accuracy of CS of concrete. For doing this, a total of 236 data points were collected from the “Red River Surface Water Plant” project, a major infrastructure initiative in Vietnam were collected and used to create training (70%) and testing (30%) datasets used for training and testing the models. For validation and comparison of the models, the popular validation metrics such as R^2, RMSE, MAE, and Taylor diagram were used. In addition, the Partial dependence plots (PDP) technique was used to validate the importance of each input variable used in the modeling. Analysis of the results illustrates that the optimized CNN, SVM, and MLP models significantly outperformed the single CNN, SVM, and MLP models, especially the optimized CNN model is the best compared with the optimized SVM and optimized MLP models, achieving an R^2 of 0.92, RMSE of 3.86 (MPa), and MAE of 3.09 (MPa). PDP analysis further revealed that key variables including cement content, coarse aggregates, and water-cement ratio have the most influential effects on the CS. The finding of this study highlights the advantages of combining deep learning with systematic hyperparameter optimization to capture complex, nonlinear relationships in concrete mix designs.

Keywords: Concrete, Compressive strength, CNN, SVM, MLP, Grid Search Optimization, Vietnam.

1. Introduction

In civil engineering, the compressive strength (CS) of concrete is one of the most essential parameters which directly reflects the material’s ability to withstand structural loads and plays a important role in the design, analysis, and safety

assessment of infrastructure [1]. Thus, the accurate prediction of the CS of concrete is crucial for ensuring structural integrity, optimizing material composition, reducing construction costs, and minimizing environmental impact by avoiding overuse of materials [2]. Traditionally, the destructive laboratory testing methods have been often used to estimate the CS of concrete. Common practical techniques include destructive compression testing of standard specimens such as cubes or cylinders, performed at specific curing times (e.g., 7, 14, or 28 days) [3]. Although these techniques are reliable, they are constrained by their cost, time requirements, and dependency on rigorous quality control during sample preparation and curing conditions [3]. Non-destructive techniques, including ultrasonic pulse velocity and rebound hammer, have also been employed, offering faster assessments [4]. Nevertheless, these methods suffer from issues such as calibration complexity, sensitivity to surface conditions, and relatively lower accuracy compared to destructive methods. Therefore, these practical approaches, while widely accepted, are often limited in efficiency and real-time applicability.

To overcome these above limitations, traditional statistical and regression-based modeling techniques have been developed to predict the CS of concrete based on a set of simple input variables [5]. Popovics and Ujhelyi [6] developed an empirical equation indicating the relationship between the CS of concrete and water-cement ratio. Sánchez-Mendieta et al. [7] analyzed the relationships between the CS of porous concretes and various parameters such as porosity, density and permeability. Othman et al. [8] analyzed the relationship between the CS of foamed concrete and density. Zhong et al. [9] studied the correlation between the CSC and dielectric properties based on aggregate particle size. Abazarsa et al. [10] estimated the CS of Portland cement concrete based on the ultrasonic testing, synthetic aperture radar, and rebound hammer. Chen et al. [11] evaluated the correlation of the CS of planting concrete with aggregate size and water/cement. Even though these traditional statistical techniques have some merits, these techniques often fall short in capturing nonlinear and complex relationships inherent in concrete mixture design, limiting their generalizability across varied datasets and conditions. Additionally, their performance depends highly on the assumption of linearity and multicollinearity among variables, that are not always valid in real-world concrete behavior.

In recent years, ML based models, which are based on the algorithms to identify relationships between inputs and outputs within the data [12], were developed and used for prediction of the properties of the materials [13, 14]. Unlike the traditional and statistical techniques, the ML models can handle complex, non-linear relationships from large and high-dimensional datasets. In addition, they are able to automatically uncover hidden patterns and interactions among variables, which makes them particularly well-suited for complex problems like the prediction of the CS of concrete [15]. In literature, there are many ML-based models were utilized for prediction of the CS of concrete. Shafighfard et al. [16] developed and applied fifteen ML models including stacked model, extremely randomized tree regressor, recurrent neural networks, histogram-based bradient boosting machines, adaboost, gradient boosting machines, support vector machine, xgboost, artificial neural networks, extra tree regression, random forest, light gradient boosting machine, k-nearest neighbors, catboost, radial basis function networks, and bagging regressor, for prediction of the CS of high-performance alkali-activated concrete. Sun et al. [17] predicted the CS of the coral aggregate concrete using the hybrid model of backpropagation neural network and genetic algorithm. Abdellatief et al. [18] investigated various ML models namely extreme gradient boosting, support vector regression, and random forest for prediction of the CS of ultra-high-

performance geopolymer concrete. Song et al. [19] compared different ML models such as decision tree, artificial neural networks, gene expression programming for prediction of the CS of self-compacting concrete with high amount of fly ash. Zeng et al. [20] applied convolutional neural network and adaboost, artificial neural networks, and support vector machines for prediction of the CS of concrete. The mentioned literature reviews showed that despite the promising results of these ML models, most existing literatures either depend on default parameters of the models or use trial-and-error approaches for optimization, that might not give the best performance of the models. It reveals a clear research gap in applying structured optimization strategies to enhance model performance of the ML models.

In this study, the main aim is to improve deep learning and ML models such as convolutional neural network (CNN), support vector machines (SVM), and Multi-Layer Perceptron Neural Network (MLP)using grid search optimization technique for prediction of the CS of concrete. The main novelty of this work is that the grid search optimization technique was used to tune the hyperparameters of the models and selected the best parameters for giving the best performance of the models. A practical database collected from the “Red River Surface Water Plant” project, a major infrastructure initiative in Vietnam were used for generation of the datasets for the modeling. For validation and comparison, the popular validation metrics such as R^2, RMSE, MAE, and Taylor diagram were used. In addition, the Partial dependence plots (PDP) technique was used to validate the importance of each input variable used in the modeling. Python software was used for data processing and modeling.

2. Materials and Methods

2.1. Data used

Data utilized in this study was collected from the “Red River Surface Water Plant” project, a major infrastructure initiative in Vietnam involving large-scale concrete use in structural and hydraulic elements. From this project, a total of 236 data points were collected. The data were compiled from laboratory reports and quality control documentation maintained throughout the construction and testing phases. All samples were tested for the CS using standard procedures, ensuring data reliability and uniformity in measurement. In the data, in addition to the 28-days CS of concrete (MPa) (output variable), a set of input variables such as age of concrete (days), water-to-cement ratio, crushed sand (kg/m³), cement content (kg/m³), natural sand (kg/m³), coarse aggregate (kg/m³), water content (kg/m³), superplasticizer admixture (kg/m³), slump ratio (mm/mm), and aggregate-to-cement ratio were selected for prediction of the CS. These input variables were selected based on their known physical and chemical influence on concrete properties and strength development. More specifically, age of concrete is a fundamental parameter as concrete strength increases with time due to continued hydration; the most significant strength gain typically occurs within the first 28 days. Cement content directly influences the formation of calcium silicate hydrates, the primary binding phase in concrete, and therefore plays a central role in strength development. Coarse aggregate contributes to the structural skeleton of concrete and its load-bearing capacity, while the gradation and shape of the particles affect the interlocking and bond characteristics. Natural sand and crushed sand are used in varying proportions as fine aggregates, and their characteristics impact workability, compaction, and ultimately the strength and durability of concrete. Crushed sand, with angular particles, often enhances interparticle friction and bond strength compared to smoother natural sand. Water content is one of the most critical parameters as it directly influences the water-to-cement ratio, which is inversely related to compressive strength. Excess water increases porosity and reduces the density of the hydrated cement matrix, leading to lower strength. On the other hand, superplasticizer admixture enables the

reduction of water content while maintaining or improving workability, allowing for stronger concrete at lower water-to-cement ratios. Slump ratio, a measure of concrete's workability, indirectly relates to water content and mixture cohesion; it affects how well the concrete can be compacted and how consistently it can be placed, both of which influence the final strength. Finally, the aggregate-to-cement ratio is a key factor in defining the overall balance between the binding phase and the skeleton material; a higher ratio may reduce cost but can dilute the cement matrix, affecting the strength negatively if not properly optimized.

Table 1. Initial analysis of the data used

Variables	abbreviation	unit	std	min	25%	50%	75%	max
Age of Concrete	AOC	(days)	15.644	11.278317	1	7	7	28
Cement content	CEC	(kg)	428.699	78.882	220	380	440	490
Coarse aggregate	COA	(kg)	1082.301	57.695	997	1025	1085	1125
Natural sand	NAS	(kg)	368.832	332.833	0	0	409	740
Crushed sand	CRS	(kg)	415.553	346.732	0	0	371	739.4
Water content	WAC	(l)	162.676	24.35	135	146	150	180.5
Superplasticizer Admixture	SUA	(l)	4.247	1.523	0	3.42	4.28	5.47
Slump ratio	SLR	(cm)	14.609	2.956	4	12	15	17
Water to cement ratio	WCR	-	0.412	0.135	0.28	0.32	0.36	0.47
Aggregate to cement ratio	ACR	-	4.572	1.22	3.09	3.77	4.19	5.01
Compressive strength of concrete	CSC	(MPa)	43.466	14.3811	10.18	33.1075	45.465	55.225

Table 1 presents a detailed statistical analysis of the data used in this study. It reveals a rich and varied dataset with good representation across all relevant mix and curing parameters. Fig. 1 presents the plots illustrating the distribution of each variable used in the data collected, offering a comprehensive visual summary of their statistical characteristics and the density of their observed values. Overall, the distribution plots show that the data used in this study contains well-distributed and diverse observations across all variables. Fig. 2 indicates a correlation matrix showing the pairwise Pearson correlation coefficients between the variables used in this study. The correlation coefficient varies between -1 and 1. A value near 1 signifies a strong positive linear association, while a value near -1 indicates a strong negative linear association. Values close to 0 imply little to no linear relationship between the variables.

In general, the correlation matrix in Fig. 2 provides valuable insights into the variable interactions and their influence on the CS of concrete. The strong correlations between the CS and certain variables (CEC, WCR, SUA, ACR) justify their inclusion in the predictive modeling process. Meanwhile, other variables with weak or complex relationships (WAC, COA) highlight the importance of using nonlinear and advanced ML models, which can uncover intricate dependencies that linear methods may overlook.

Database collected was divided into two parts including training dataset (70%) used for training the models and testing dataset (30% remaining) used for validating the model.

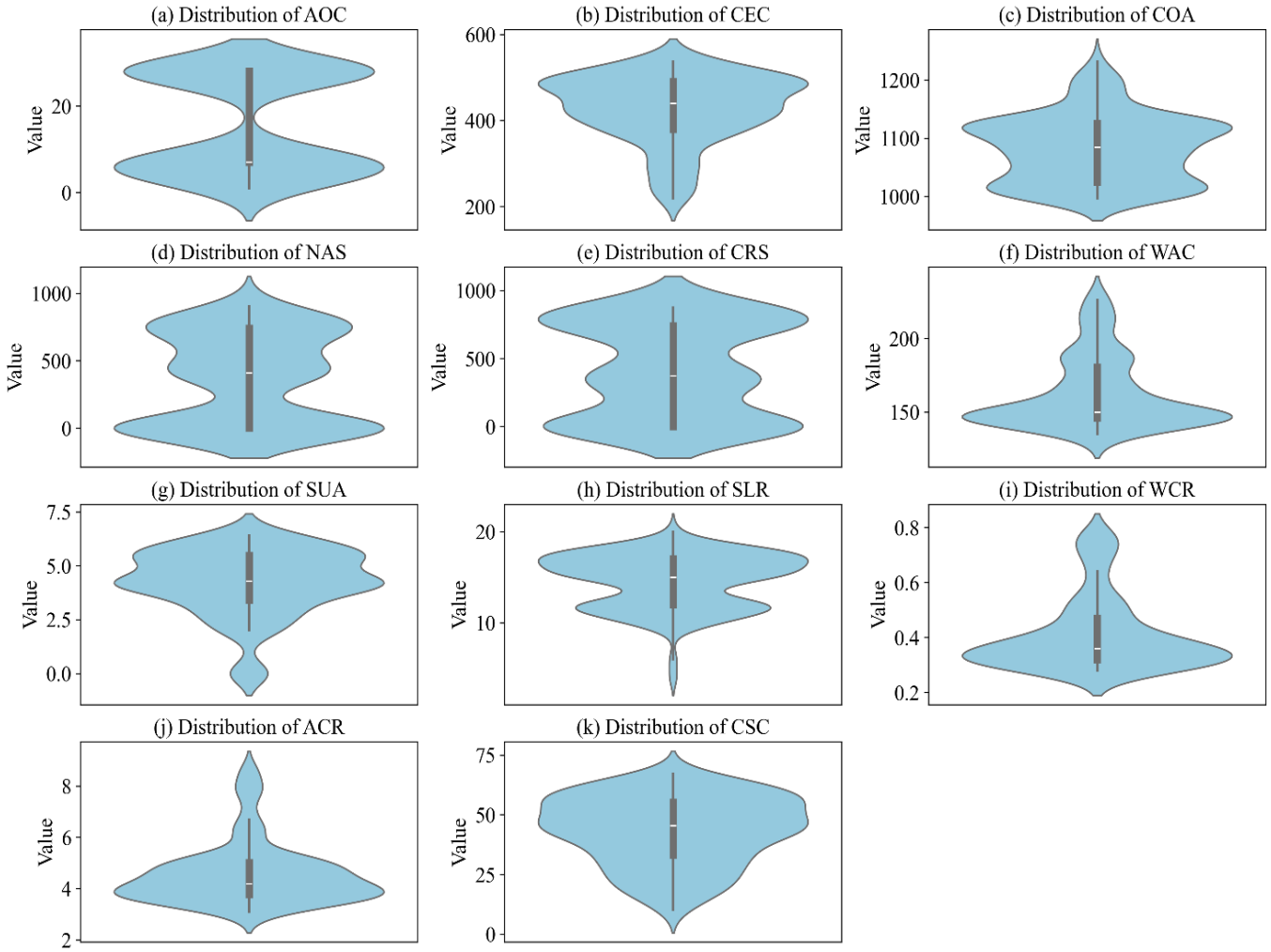

Fig. 1. Data distribution of the variables used in the modeling

2.2. Methods used

2.2.1. Convolutional Neural Network (CNN)

CNN were originally introduced by LeCun et al. [21]. The main principle of the CNN lies in its ability to extract local patterns and hierarchical representations from the input data through a series of convolutional layers, activation functions, pooling operations, and fully connected layers [22]. CNN can be used and applied for both classification and regression problems. In prediction of the CS of concrete of this study – a type of regression problem, the one-dimensional form of CNN is applied, where the input variables (e.g., cement content, water-to-cement ratio, superplasticizer dosage, etc.) are considered as a 1D feature map. The convolutional layers apply kernels or filters that slide over the input data to detect meaningful patterns- such as interactions between material properties - which affect the CS of concrete. These patterns are passed through activation functions like the Rectified Linear Unit (ReLU) to introduce non-linearity and then optionally pooled to reduce dimensionality and enhance feature generalization. The extracted features are finally passed through fully connected (dense) layers to compute the CS of concrete (output). The CNN regression model can be trained by the main mathematical formulations as below [21]:

Convolution Operation:

$$z_i = \sum_{j=1}^{k} x_{i+j-1} \cdot \omega_j + b \quad (1)$$

where x is the input feature vector, ω is the kernel (filter) weight vector, b is the bias term, and z_i is the output of the convolution at position i.

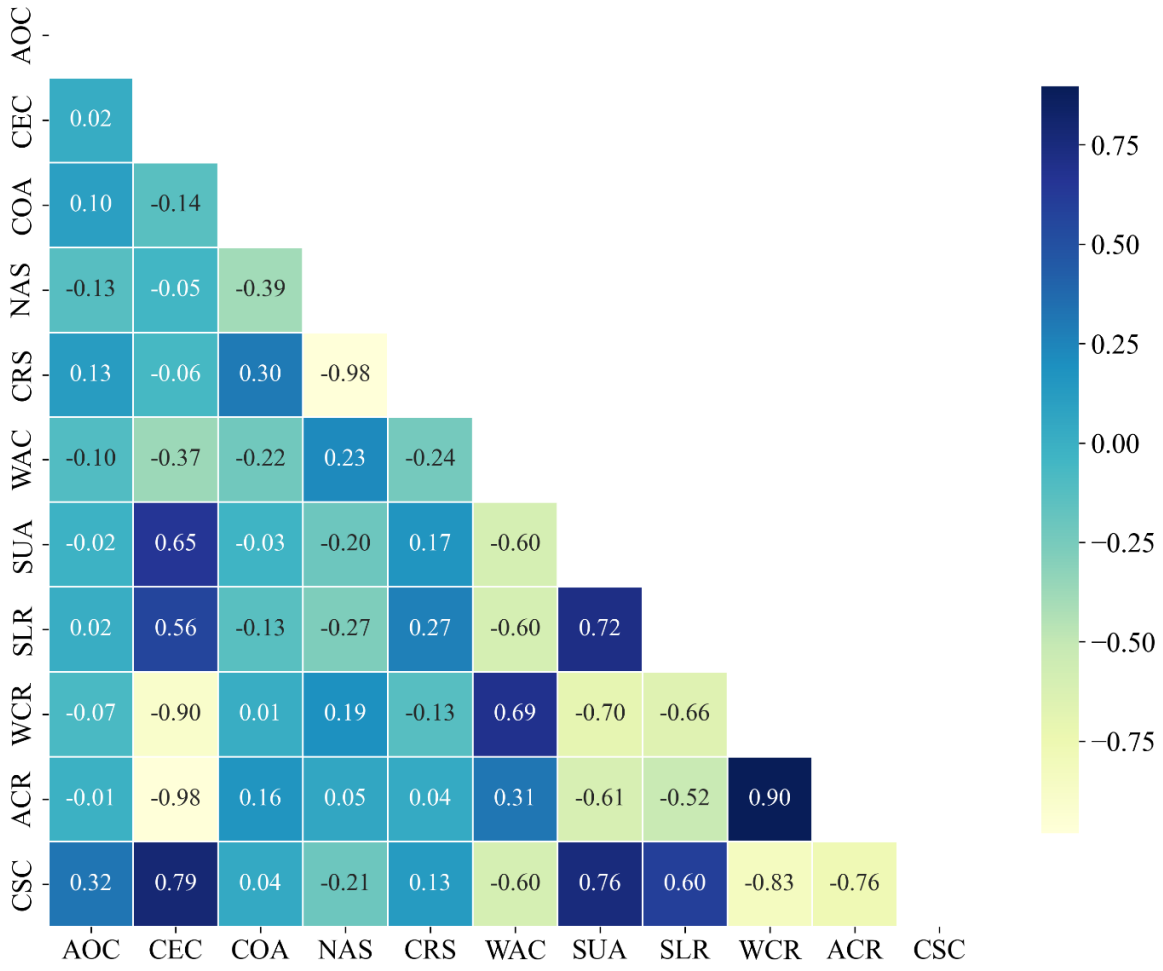

Fig. 2. Correlation analysis of the variables used in the modeling

Activation Function (ReLU):

$$a_i = \max(0, z_i) \quad (2)$$

ReLU introduces non-linearity by passing only positive values and zeroing out the negative outputs.

Pooling (e.g., Max Pooling):

$$p_i = \max(a_i, a_{i+1}, ..., a_{i+k-1}) \quad (3)$$

which reduces the size of the feature map while retaining important features.

Fully Connected Layer:

$$y = \sum_{i=1}^{n} a_i \cdot \omega_i + b \quad (4)$$

where a_i are the flattened activations from the previous layer, and y is the final prediction of compressive strength.

The CNN model in this study was trained using the Mean Squared Error (MSE) loss function, defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (5)$$

where y_i is the actual CSC and $\hat{y}_i$ is the predicted value.

In this study, the optimization of the CNN was performed using the Adam optimizer, and hyperparameters such as learning rate, kernel size, the number of filters, and batch size were fine-tuned using grid search optimization to enhance prediction accuracy.

2.2.2. Support Vector machines (SVM)

SVM were initially developed by Vapnik [23]. It can be applied to solve both classification and regression problems. The main principle of SVM for regression problems is to find a function that approximates the relationship between input variables and the target output (the CS), such that the predictions fall within a specified epsilon-insensitive margin of the actual values [24]. Unlike

traditional regression models that minimize the prediction error directly, SVM attempts to fit the best possible hyperplane within a threshold (ε) while minimizing model complexity. This is achieved by identifying a subset of training data points, known as support vectors, which lie outside the epsilon boundary and contribute to defining the regression function. The SVM optimization problem can be formulated as follows [23]:

Minimize the objective function:

$$\min \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) \quad (6)$$

subject to:

$$\begin{aligned} & y_i - \left(\omega \cdot x_i + b\right) \le \varepsilon + \xi_i \\ & \left(\omega \cdot x_i + b\right) - y_i \le \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \ge 0 \end{aligned} \quad (7)$$

where ω is the weight vector, b is the bias term ξ_i and ξ_i^* are slack variables for deviations beyond ε, and C is a regularization parameter that controls the trade-off between the flatness of the function and the amount up to which deviations larger than ε are tolerated. The decision function in SVR for a linear case is:

$$f(x) = \omega \cdot x + b \quad (8)$$

For nonlinear relationships, which are typical in concrete strength prediction, SVR uses a kernel function $K\left(x_i, x_j\right)$ to project the input data into a higher-dimensional space where linear regression is more effective. Common kernel functions include the Radial Basis Function (RBF), polynomial, and linear kernels. In this study, important hyperparameters of SVM such as kernel functions, regularization parameter *C*, and epsilon - gamma (ε) were optimized using grid search optimization for the best model performance.

2.2.3. Multilayer Perceptron Neural Network (MLP)

MLP is one of the most fundamental architectures in artificial neural networks [25]. An MLP includes of an input layer, one or more hidden layers, and an output layer, where each layer is composed of interconnected nodes or neurons. The main principle of MLP lies in hierarchical representation learning, where each layer transforms the input data into increasingly abstract representations through weighted connections and activation functions [26]. In this study, MLP is applied to capture the complex and nonlinear interactions among the ten input variables to predict the CS of concrete. Each neuron in the MLP computes a weighted sum of its inputs, adds a bias term, and applies a nonlinear activation function (typically ReLU or sigmoid). The MLP model operates based on the following main mathematical expressions [27]:

Weighted Sum and Activation:

$$h_j = \phi\left(\sum_{i=1}^{n} x_i \cdot \omega_{ij} + b_j\right) \quad (9)$$

where x_i are input features, ω_{ij} are the weights connecting input neuron *i* to hidden neuron j, b_j is the bias term, and ϕ is the activation function (ReLU).

Output Computation:

$$y = \sum_{j=1}^{m} h_i \cdot \omega_j^{(out)} + b^{(out)} \quad (10)$$

where h_j are the outputs from the hidden layer, $\omega_j^{(out)}$ are weights from hidden to output layer, and $b^{(out)}$ is the output bias. For regression tasks, a linear activation is typically used at the output node.

Loss Function (Mean Squared Error):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \quad (11)$$

where y_i is the actual CSC and $\hat{y}_i$ is the predicted value.

Weight updates are calculated using the gradients of the loss function with respect to weights and biases. Using gradient descent:

$$\omega = \omega - \eta\frac{\partial MSE}{\partial \omega} \quad (12)$$

where η is the learning rate.

In this work, the MLP architecture was optimized using grid search optimization, allowing the systematic tuning of key hyperparameters such as batch size, learning rate, the number of neurons per layer, number of hidden layers, and activation functions. This ensures that the MLP achieves optimal performance for the regression task, adapting effectively to the real-world variability of the concrete dataset.

2.2.4. Grid search optimization

Grid search optimization is a fundamental technique in ML and statistical modeling used to systematically tune hyperparameters in predictive algorithms [28]. It originated from the broader field of combinatorial optimization and has been widely adopted due to its simplicity and effectiveness in finding optimal model configurations [29]. The main principle behind grid search optimization is to exhaustively search through a manually specified subset of the hyperparameter space, evaluating model performance at each combination [30].

Even though grid Search does not involve advanced mathematical formulations like gradient-based optimization methods, its evaluation strategy can be mathematically described. Let θ be a vector of hyperparameters, and let $\Gamma(\theta)$ be a loss function such as the Mean Squared Error (MSE) computed on the validation set. Then the optimization objective can be expressed as:

$$\theta^* = \arg\min_{\theta \in \Theta} \Gamma(\theta) \tag{13}$$

where Θ represents the discrete grid of all possible hyperparameter combinations, and θ^* denotes the optimal configuration that minimizes the validation loss.

In this work, the MSE is used as the loss function for regression, defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{14}$$

where y_i is the actual CSC and $\hat{y}_i$ is the predicted value from the model using a specific hyperparameter setting.

2.2.5. Validation metrics

In this study, the prediction of the CS of concrete using CNN, SVM, and MLP models was evaluated using three widely popular metrics [31-33]: coefficient of determination (R^2), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), which provide comprehensive insights into how well the predicted values from each model match the actual observed values. More specifically, R^2 measures the proportion of the variance in the dependent variable (CS) that is predictable from the independent variables (input variables) [34, 35]. It provides a value between 0 and 1 (or even negative for poor models), where a higher R^2 indicates better model performance. A value of R^2 =1 represents perfect predictions, while values close to zero or negative indicate poor fit. The R^2 is calculated as [31]:

$$R^2 = \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2} \tag{15}$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and $\hat{y}_i$ is the mean of the actual values.

RMSE is a metric that quantifies the square root of the average of the squared differences between predicted and actual values [36, 37]. This metric is particularly useful for evaluating the overall magnitude of prediction error and is expressed in the same units. RMSE is defined as [38, 39]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2} \tag{16}$$

MAE calculates the average of the absolute differences between predicted and actual values [40, 41]. It gives equal weight to all errors, regardless of their direction or magnitude, making it more robust in the presence of outliers than RMSE. MAE provides a direct, interpretable measure of the average error in predictions, indicating how far off the model typically is from actual values. The formula for MAE is [39, 42]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{17}$$

Each of these metrics has its advantages and disadvantages. R^2 is intuitive and provides a clear measure of goodness-of-fit, but it may be misleading when used alone, especially with non-linear models or in the presence of outliers. RMSE is sensitive to large deviations, which is helpful for emphasizing major prediction errors, but it can overstate the impact of a few large outliers. On the other hand, MAE offers a more balanced view of average prediction error but does not differentiate between high and low-magnitude errors as effectively as RMSE. By using all three metrics in combination, this study ensures a robust and multi-faceted evaluation of model performance.

In addition to R^2, RMSE, and MAE, Taylor diagram was also used to compare the models. It is a powerful graphical tool introduced by Taylor [43] to simultaneously visualize and compare the performance of multiple models against observed data using three key statistical metrics: R^2, RMSE, and MAE. The main principle of the Taylor diagram lies in its ability to represent multiple statistical properties in a single two-dimensional polar plot. In this study, the diagram allows for a concise yet powerful visualization of how closely each model's predictions match the actual CS of concrete values, providing insights not just into accuracy but also into the pattern similarity and spread of the predicted data.

2.2.7. Partial Dependence Plots (PDP)

PDP is powerful tool for interpreting complex ML models, especially when the models are considered "black boxes". The concept of PDP was first introduced by Jerome Friedman in the context of generalized additive models and later popularized in the domain of interpretable ML to explain nonlinear, non-parametric models [44]. The main principle of PDP lies in isolating the marginal effect of a specific input variable on the model's output. For example, if the goal is to understand how cement content affects the CS of concrete, the PDP will show how the predicted CS changes when cement content varies across its range while all other input features are averaged out. This is particularly useful for understanding the direction, strength, and nonlinearity of the influence of individual features on the outcome in complex models. In this study, PDPs were plotted for the best model obtained from the comparison of three optimized models (CNN, SVM, and **ANN**) to assess whether they captured physically meaningful trends.

3. Results and discussion

3.1. Optimization of the models using grid search optimization

Each model was systematically optimized using the grid search algorithm, which is widely recognized for its efficiency in identifying optimal hyperparameter configurations through an exhaustive search over a predefined parameter space. The defined hyperparameter spaces and the optimal values obtained for each model are summarized in Tables 2 to 4.

Table 2. Hyper-parameters of SVM optimized and selected

No	Hyper-parameters	Values	Best values
1	kernel	['rbf', 'Poly', 'linear']	rbf
2	C	[1, 10, 50, 100]	100
3	gamma	['scale', 'auto']	auto

Table 3. Hyper-parameters of MLP optimized and selected

No	Hyper-parameters	Values	Best values
1	hidden_layer_sizes	[(50,), (100,), (100, 50)]	(100, 50)
2	activation	['relu', 'tanh']	relu
3	alpha	[0.0001, 0.001]	0.001
4	solver	['adam']	adam

With SVM, the optimization process explored a variety of kernel functions (including *rbf*, *poly*, and *linear*), regularization parameters (*C*), and kernel coefficients (*gamma*). The best configuration was determined to be an *rbf* kernel with *C* = *100* and *gamma* = *auto* (Table 2). For MLP underwent

tuning across multiple architectural and training parameters, including the hidden layer structure, activation function, regularization parameter (*alpha*), and solver type. The optimal model featured a two-layer architecture with 100 and 50 neurons, the *ReLU* activation function, *alpha = 0.001*, and the *Adam* solver (Table 3). Related with CNN, a set of hyperparameters was optimized such as the number of filters, kernel size, number of dense units, batch size, dropout rate, activation function, and learning rate. The most effective CNN architecture included 32 filters with a kernel size of 3, 128 units in the dense layer, a batch size of 4, a dropout rate of 0.2, the *ReLU* activation function, and a learning rate of 0.001 (Table 4).

Table 4. Hyper-parameters of CNN optimized and selected

No	Hyper-parameters	Values	Best values
1	filters	[16, 32, 64, 128]	32
2	kernel_size	[1, 2, 3]	3
3	dense_units	[16, 32, 64, 128]	128
4	batch_size	[2, 4, 8, 16]	4
5	activation	['relu', 'tanh']	relu
6	dropout_rate	[0.2]	0.2
7	learning_rate	[0.001, 0.0005]	0.001

Table 5 presents a detailed comparison of performance metrics: R^2, RMSE, and MAE both before and after grid search optimization. The results clearly indicate a significant improvement in predictive performance across all models after hyper-parameter tuning. More specifically, CNN achieved the highest predictive performance, with an R^2 score of 0.95 (training) and 0.918 (testing) after optimization, compared to 0.89 and 0.84 before optimization. SVM, after tuning, improved its testing R^2 from 0.75 to 0.9025. MLP also benefited notably from optimization, with R^2 increasing from 0.84 to 0.9040 on the test set. In addition, RMSE and MAE values for all models were substantially reduced following optimization, confirming better generalization capability and lower prediction errors. For instance, the testing RMSE for CNN dropped from 5.34 to 3.06, and testing MAE decreased from 4.21 to 2.30.

In short, grid Search optimization significantly improved the performance of all three models, with CNN consistently outperforming SVM and MLP. This validates the use of deep learning combined with systematic hyperparameter tuning as an effective approach for predicting concrete compressive strength. More detail comparison of the models is presented in the following sections.

3.2. Optimized model validation and comparison

Validation and comparison of the optimized CNN, SVM, MLP models are further presented and shown in Figs. 3, 4, 5, and 6 on both training and testing datasets.

Fig. 3 illustrates the scatter plots of predicted versus actual CS values for the SVM, MLP, and CNN models during both training and testing phases. Fig. 3a shows SVM achieved an R^2 of 0.98 during training, showing a strong fit between predicted and actual values. However, in the testing phase shown in Fig. 3b, the R^2 decreased to 0.90, indicating a moderate drop in performance and suggesting some level of overfitting. Nonetheless, the testing accuracy still demonstrates solid predictive capability.

Figs. 3c,d indicate MLP achieved an R^2 of 0.97 in training and 0.90 in testing, which are slightly lower than those of SVM in training but equal in testing. This result suggests that MLP has comparable generalization ability to SVM but may be slightly more stable due to a smaller gap between training and testing R^2 values. Figs. 3e,f indicate CNN attained an R^2 of 0.95 in training and the highest R^2 in testing at 0.92. While its training performance is slightly lower than SVM and MLP, CNN generalizes better to unseen data, as evidenced by its leading testing R^2.

Table 5. Comparison of model performance using Grid search optimization

Metric	Dataset	Without Grid search optimization			With Grid search optimization		
		SVM	MLP	CNN	SVM	MLP	CNN
R^2	Training	0.78	0.84	0.89	0.98	0.97	0.95
	Testing	0.75	0.63	0.84	0.902	0.902	0.92
RMSE	Training	6.80	5.85	4.94	2.30	2.68	3.11
	Testing	6.72	8.17	5.38	4.21	4.18	3.86
MAE	Training	5.45	4.42	3.93	1.40	2.10	2.59
	Testing	5.33	6.10	4.41	2.81	2.92	3.09

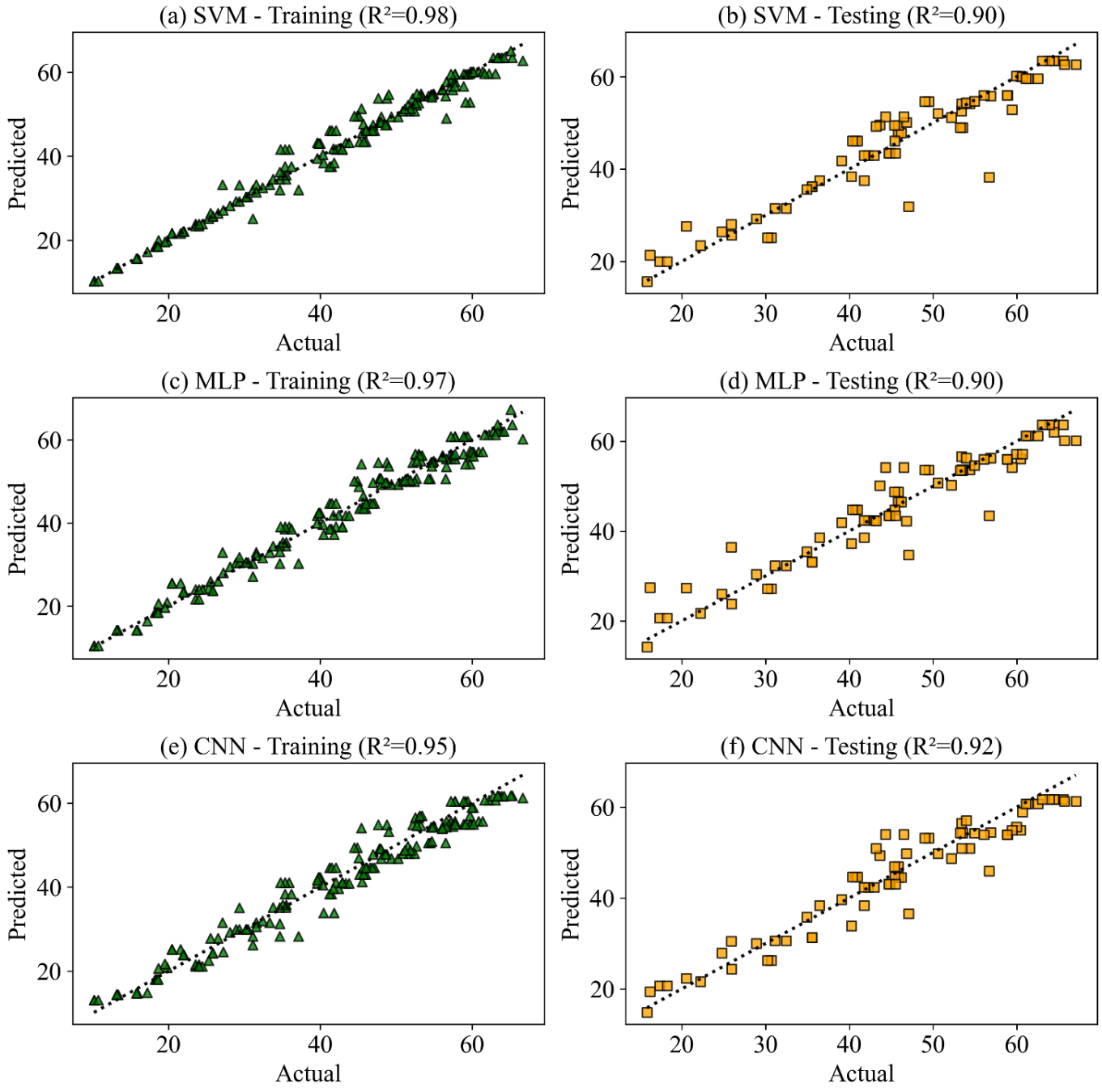

Fig. 3. R^2 values of the models: (a) SVM training, (b) SVM testing, (c) MLP training, (d) MLP testing, (e) CNN training, and (f) CNN testing

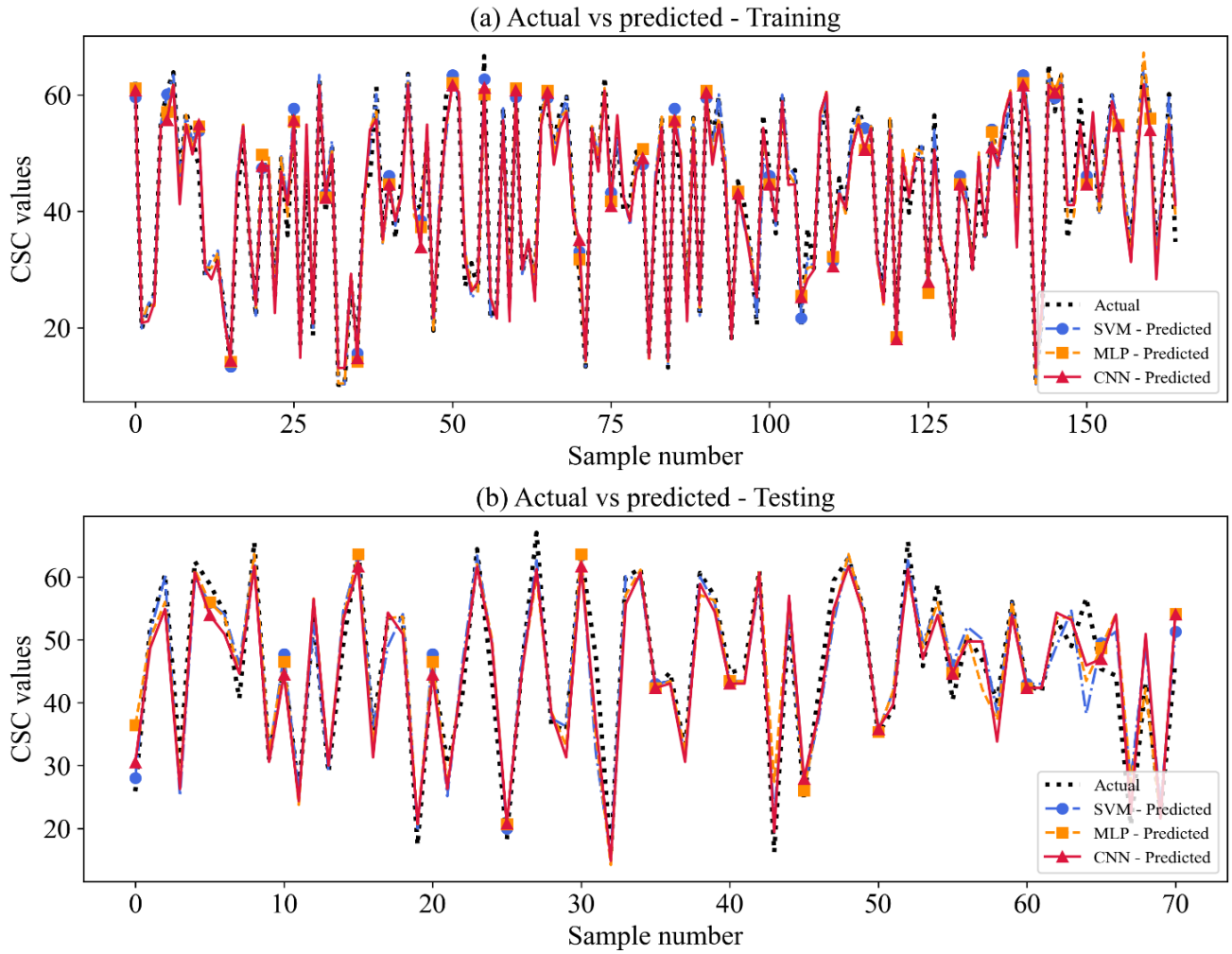

Fig. 4. Actual vs predicted values of the models: (a) training and (b) testing

Fig. 4 provides a visual comparison between the actual CS of concrete values and the predicted CS values from the SVM, MLP, and CNN models across the training and testing datasets. In the training phase (Fig. 4a), all three models: SVM, MLP, and CNN closely follow the actual, demonstrating their capacity to learn complex relationships within the data. The MLP and CNN curves almost overlap the actual values consistently, while SVM shows slightly more variation and divergence in some segments. This suggests that MLP and CNN are better at minimizing residual errors during training, likely due to their higher capacity for nonlinear feature mapping. In the testing phase (Fig. 4b), a more distinct difference between the models emerges. CNN remains the most faithful to the actual values across most samples, indicating its superior generalization capability. MLP also shows robust performance but with slightly larger deviations than CNN in certain high-gradient regions. In contrast, SVM exhibits more noticeable fluctuations and under- or over-estimations, particularly in the mid-range values.

Fig. 5 provides a comparative analysis of the three models SVM, MLP, and CNN using the values of three validation metrics: R^2, RMSE, and MAE on both training and testing datasets. Fig. 5a illustrates the R^2 values, SVM exhibits the highest training performance with an R^2 of 0.98, followed closely by MLP at 0.97, and CNN at 0.95, respectively. However, in the testing dataset, CNN outperforms the other models with the highest R^2 of 0.92, while both SVM and MLP plateau at 0.90.

Fig. 5b presents the RMSE values, where SVM records the lowest training error (2.30)

compared to MLP (2.68) and CNN (3.11). Yet on the testing set, the pattern reverses: CNN achieves the lowest RMSE (3.86), followed by MLP (4.18) and SVM (4.21), respectively. In terms of MAE, Fig. 5c shows that SVM again demonstrates the smallest training error (1.40), with MLP and CNN at 2.10 and 2.59, respectively. However, for the testing dataset, CNN slightly leads with a MAE of 3.09, marginally outperforming MLP (2.92) and SVM (2.81).

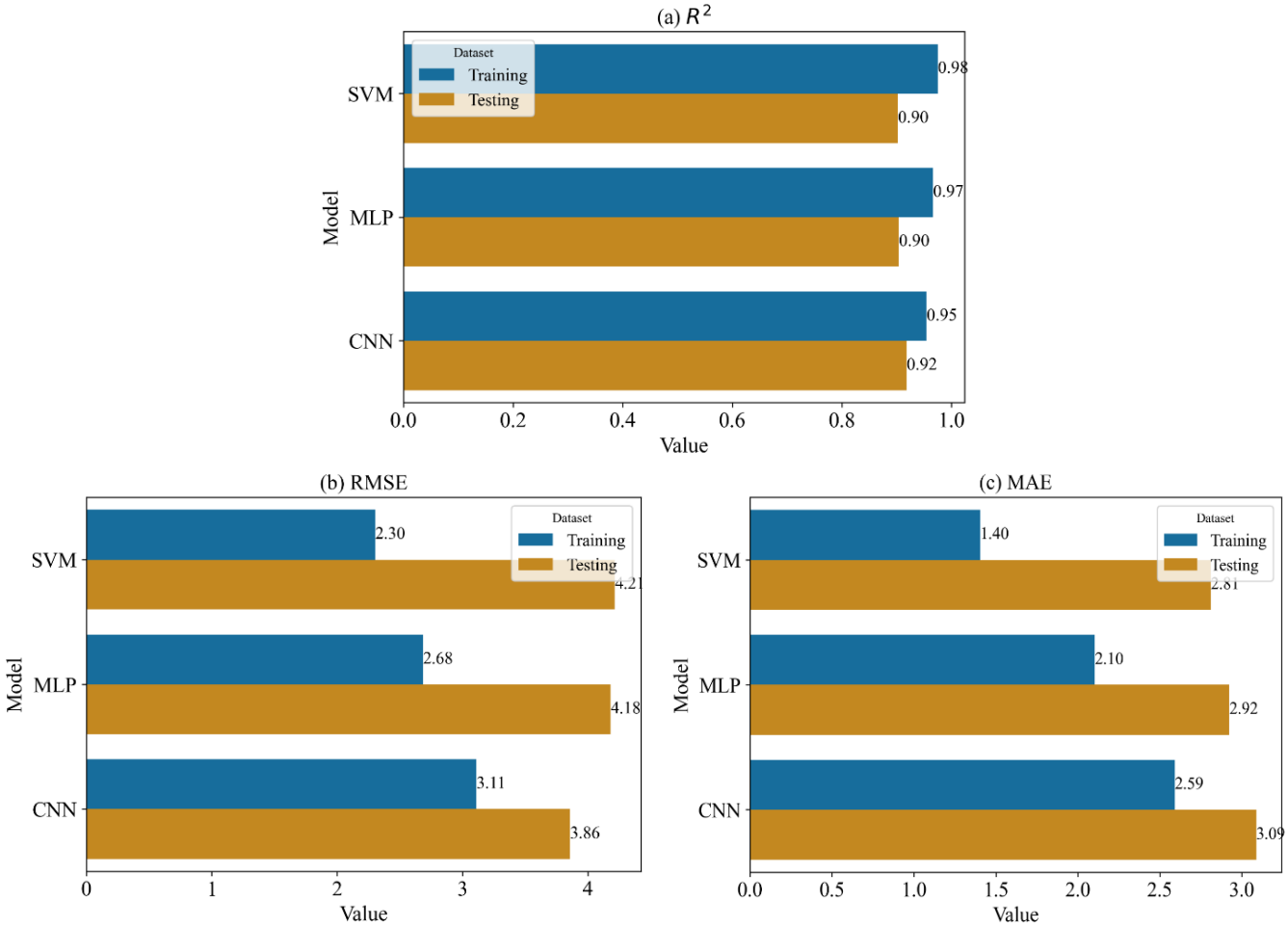

Fig. 5. Comparison of the validation metrics of the models: (a) R^2, (b) RMSE, and (c) MAE

Fig. 6 presents a comprehensive Taylor diagam analysis to visually assess and compare the predictive performance of SVM, MLP, and CNN models based on RMSE, MAE, and R^2 values for both training and testing datasets. In the training phase, Figs. 6a,c show that the SVM model performs with the lowest RMSE and MAE values (closer to the reference point), followed closely by MLP, and with CNN slightly further out. However, when transitioning to the testing phase (Figs. 6b, d) CNN becomes the most accurate, exhibiting lower RMSE and MAE values than SVM and MLP. Figs. 6e,f show the R^2-based Taylor diagrams of the models. In Fig. 6e, all models show high correlation with actual values, with SVM achieving the closest proximity to the reference, consistent with its higher training R^2 (0.98). Conversely, in the testing phase (Fig. 6f), CNN clearly outperforms, aligning more closely with the reference in terms of correlation and R^2 (0.92), surpassing both MLP and SVM, which maintain an R^2 of 0.90.

In summary, SVM performs best on the training set but shows signs of overfitting, while CNN provides the most balanced and robust performance, particularly on the testing dataset. The results demonstrate that CNN, despite its slightly higher training error, benefits from a deeper architecture and optimized hyperparameters via

grid search, resulting in superior generalization performance in predicting the CS of concrete. The superior performance of CNN can be attributed to both physical and computational factors. From a computational perspective, CNN's ability to extract spatial hierarchies and nonlinear relationships makes it inherently suitable for handling complex material behavior and multivariate interactions, especially after tuning hyperparameters such as filters, kernel size, and dropout rate. Physically, concrete strength depends on the interrelated effects of multiple ingredients (e.g., cement content, water-cement ratio, aggregate proportions), and CNN's architecture is well-suited to capture these intricate patterns that may not be fully exploited by shallower models like SVM and MLP.

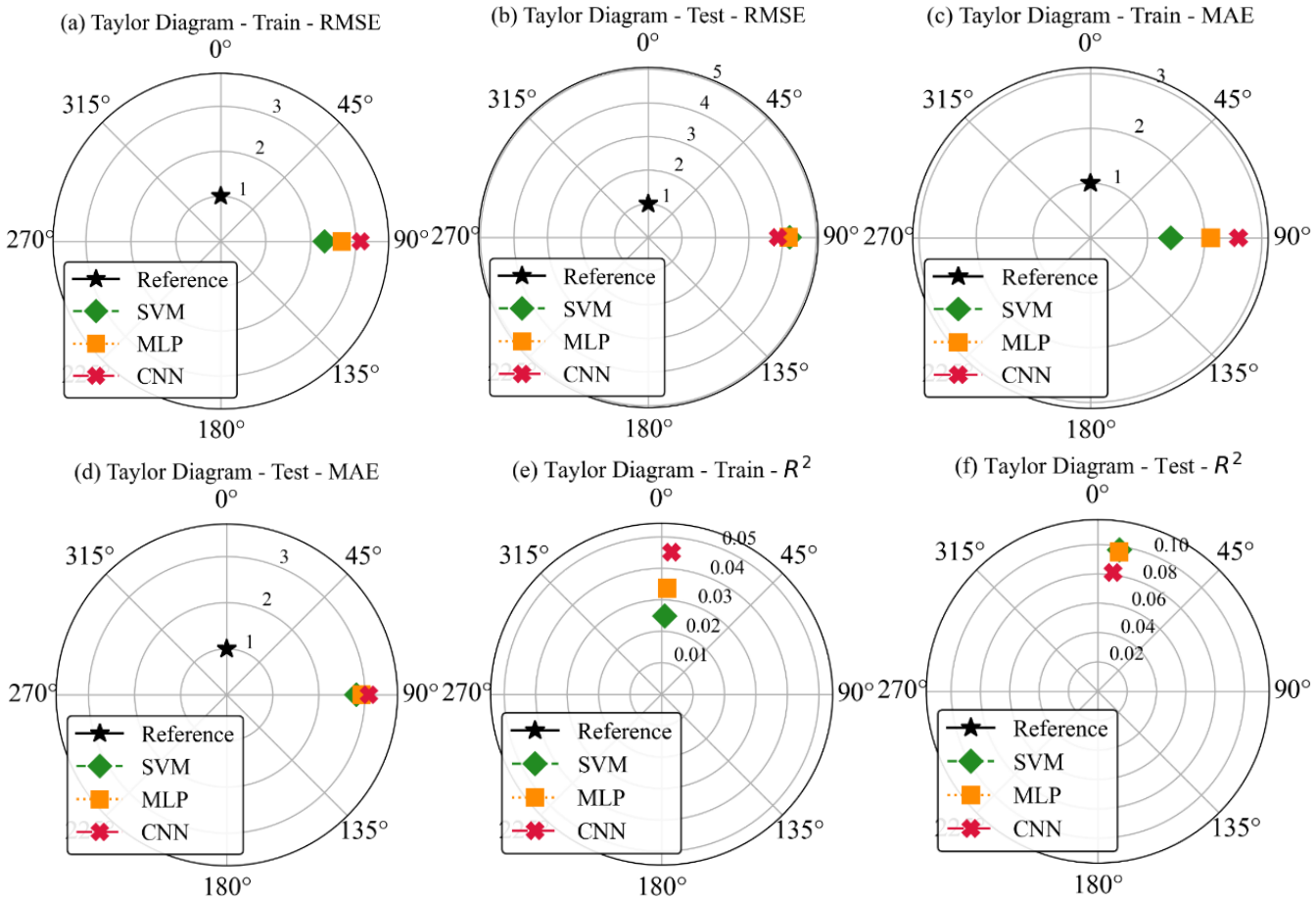

Fig. 6. Taylor diagram analysis of the models: (a) RMSE training, (b) RMSE testing, (c) MAE training, (d) MAE testing, (e) R^2 training, and (f) R^2 testing

3.2. PDP analysis

Validation and comparison of the models showed that optimized CNN is the best model compared with the optimized SVM and MLP for prediction of the CS of concrete. Therefore, CNN model was used for PDP analysis as shown in Fig. 7.

Fig. 7a shows a strong positive relationship between AOC and the CS of concrete. As AOC increases, the predicted CSC increases steadily. This aligns well with fundamental concrete science, where more cement contributes to higher matrix density and bonding strength. Fig. 7b shows the CEC exhibits a negative trend: as CEC increases, the predicted CS of concrete decreases. This may suggest that higher values of CEC, which possibly associated with inefficient cement use, are detrimental to strength. Fig. 7c shows COA initially has a weak influence, followed by a nonlinear increase beyond a threshold value, indicating that adequate coarse aggregate enhances load transfer and compressive resistance once a critical

volume is reached. Fig. 7d a U-shaped curve for NAS: the CS of concrete slightly decreases at first, then increases sharply. This suggests that very low or very high amounts of NAS can be beneficial, likely due to its role in improving compaction and reducing voids. Fig. 7e shows CRS curve shows a positive relationship, especially beyond a midpoint value. It reinforces the idea that crushed sand with angular particles may contribute to better interlocking in the mix.

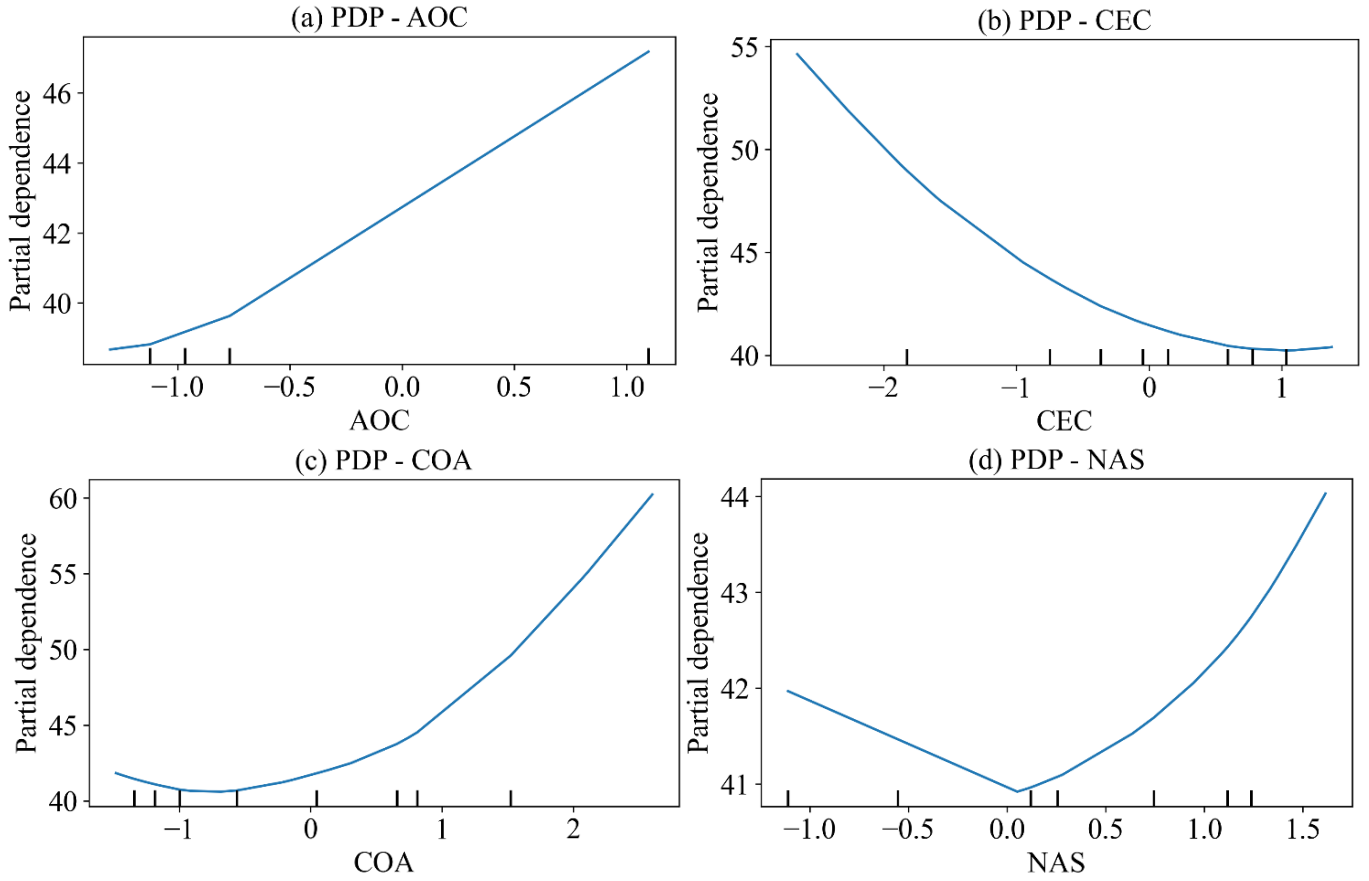

Fig. 7. PDP analysis of the variables using RF model: (a) G, (b) CS, (c) FS, (d) SC

WAC in Fig. 7f shows a nonlinear increase in the CS of concrete with increasing WAC, suggesting that optimal water content relative to aggregate helps hydration and workability, up to a point before oversaturation becomes detrimental.

Fig. 7g shows that SUA has a clearly positive influence, with the CS of concrete increasing consistently with SUA levels. These additives may act as pozzolanic or filler materials that refine pore structure and boost long-term strength.

SLR in Fig. 7h shows a mild U-shaped curve, indicating that both very low and very high SLR values are favorable, likely due to the combined effects of silica and calcium reactions forming additional C-S-H gel.

WCR in Fig. 7i is negatively correlated with the CS of concrete, which is well-aligned with established concrete theory: as WCR increases, excess water reduces density and creates voids, leading to lower strength.

ACR in Fig. 7j also shows a negative trend, implying that excessive admixture use may hinder the hydration process or introduce unfavorable chemical interactions if not properly balanced.

In general, the PDP results highlight the complex, nonlinear interactions between input parameters and the CS of concrete. Variables such as AOC, SUA, CRS, and COA contribute positively and significantly to the CS when properly balanced, while high values of WCR, ACR, and CEC have a detrimental effect. These findings not only validate the internal reasoning of the ML model but also confirm fundamental engineering knowledge, thereby increasing confidence in the model's

predictive and interpretive power. Therefore, the PDP analysis serves as a valuable tool not just for model transparency, but also for guiding material design decisions. It confirms that cement dosage, aggregate quality, supplementary materials, and water management remain the cornerstone variables in concrete strength prediction and optimization.

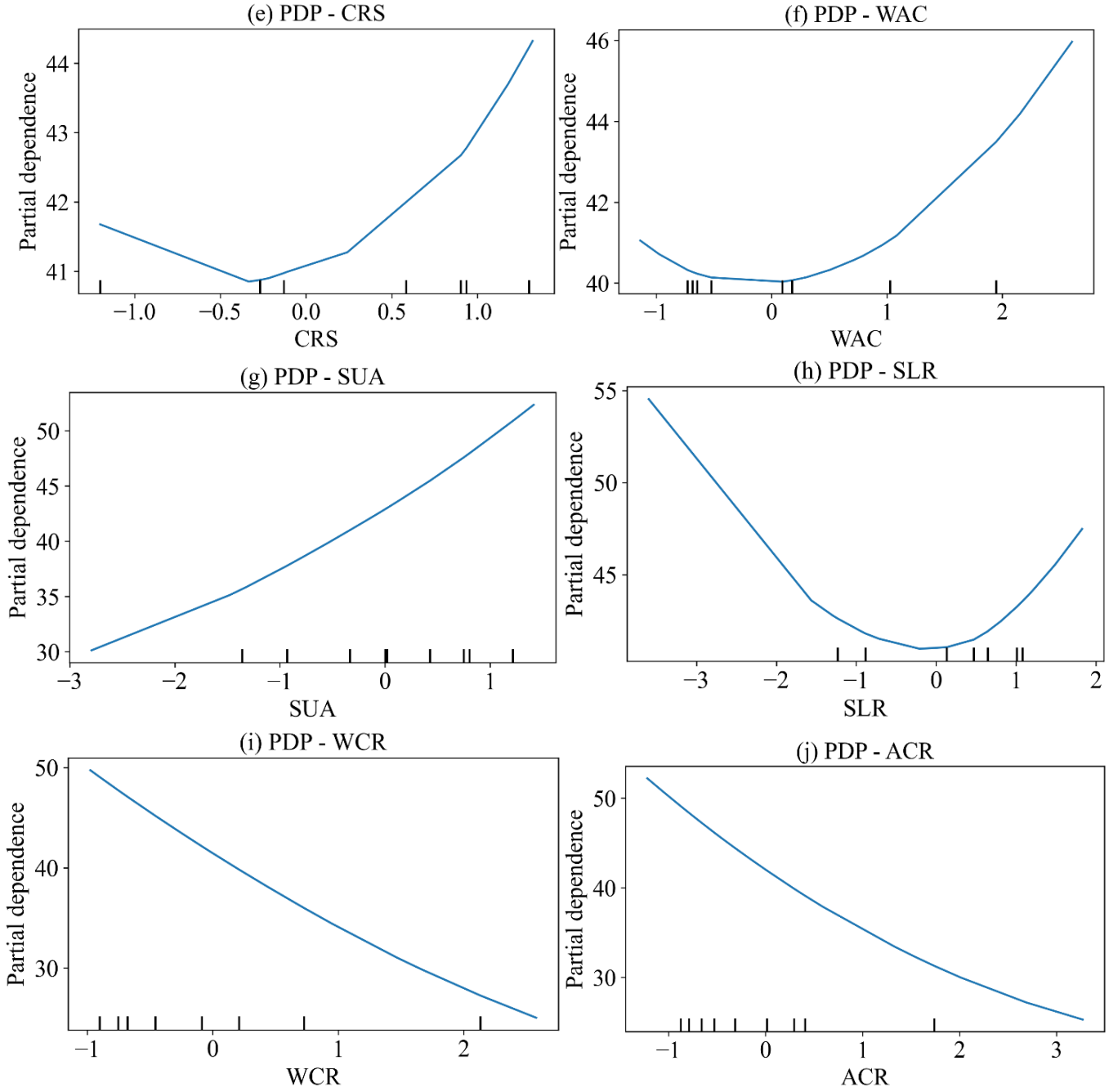

Fig. 7. PDP analysis of the variables using RF model: (e) O, (f) LL, (g) PL, and (h) PI

5. Conclusion

Predicting the CS of concrete remains a critical problem in civil engineering, as it directly influences the design, safety, and durability of structures. In this study, three ML models: CNN, SVM, and MLP were evaluated for the CS of concrete prediction. Importantly, a grid search optimization strategy was applied to fine-tune the hyperparameters of each model, enhancing their predictive accuracy and generalization capabilities. Results of this study show that while all three models benefited from hyperparameter tuning, the CNN model outperformed SVM and MLP across all evaluation metrics. The optimized CNN achieved

an R^2 of 0.92, RMSE of 3.86, and MAE of 3.09 on the testing set. In contrast, although SVM achieved the highest R^2 during training (0.98), its performance dropped to 0.90 on testing, indicating a higher tendency toward overfitting. The MLP model offered a balance between SVM and CNN, but was ultimately surpassed by CNN in both accuracy and consistency. PDP analysis further validated the model's learning behavior by identifying key variables such as AOC, COA, and WCR as dominant factors affecting the CS of concrete.

While the study provides strong evidence supporting the use of CNN with hyperparameter optimization for CSC prediction, some limitations should be addressed in future research. The dataset size, though sufficient for this analysis, can be expanded to include more diverse mix designs, curing conditions, and regional variations. Moreover, more sophisticated optimization techniques such as Bayesian optimization, genetic algorithms, or ensemble strategies could be explored to further enhance model robustness. Finally, integration of uncertainty quantification, feature selection techniques, and explainable AI methods such as SHAP or LIME could provide more actionable insights for engineers and material designers.

In conclusion, this study has demonstrated the potential of deep learning, particularly CNNs optimized via grid search, as a powerful and interpretable tool for predicting the CS of concrete. With further development and data enrichment, such models could become indispensable in modern concrete design, quality control, and performance assessment.

References

[1] S. Paudel, A. Pudasaini, R.K. Shrestha, E. Kharel. (2023). Compressive strength of concrete material using machine learning techniques. *Cleaner Engineering and Technology*, 15, 100661. https://doi.org/10.1016/j.clet.2023.100661

[2] P.G. Asteris, A.D. Skentou, A. Bardhan, P. Samui, K. Pilakoutas. (2021). Predicting concrete compressive strength using hybrid ensembling of surrogate machine learning models. *Cement and Concrete Research*, 145, 106449.https://doi.org/10.1016/j.cemconres.2021.106449

[3] K. Ali-Benyahia, S. Kenai, M. Ghrici, Z.-M. Sbartaï, S.-M. Elachachi. (2023). Analysis of the accuracy of in-situ concrete characteristic compressive strength assessment in real structures using destructive and non-destructive testing methods. *Construction and Building Materials*, 366, 130161. https://doi.org/10.1016/j.conbuildmat.2022.130161

[4] O.D. Atoyebi, O.P. Ayanrinde, J. Oluwafemi. (2019). Reliability Comparison of Schmidt Rebound Hammer as a Non-Destructive Test with Compressive Strength Tests for different Concrete Mix. *Journal of Physics: Conference Series*, 1378, 032096. https://doi.org/10.1088/1742-6596/1378/3/032096

[5] W.Z. Taffese, L. Espinosa-Leal. (2023). Multitarget regression models for predicting compressive strength and chloride resistance of concrete. *Journal of Building Engineering*, 72, 106523. https://doi.org/10.1016/j.jobe.2023.106523

[6] S. Popovics, J. Ujhelyi. (2008). Contribution to the Concrete Strength versus Water-Cement Ratio Relationship. *Journal of Materials in Civil Engineering*, 20(7), 459-463. https://doi.org/10.1061/(ASCE)0899-1561(2008)20:7(459)

[7] C. Sánchez-Mendieta, J.J. Galán-Díaz, I. Martinez-Lage. (2024). Relationships between density, porosity, compressive strength and permeability in porous concretes: Optimization of properties through control of the water-cement ratio and aggregate type. *Journal of Building Engineering*, 97, 110858. https://doi.org/10.1016/j.jobe.2024.110858

[8] R. Othman, R.P. Jaya, K. Muthusamy, M. Sulaiman, Y. Duraisamy, M.M.A.B. Abdullah, A. Przybył, W. Sochacki, T. Skrzypczak, P. Vizureanu, A.V. Sandu. (2021). Relation between Density and Compressive Strength of Foamed Concrete. *Materials*, 14(11), 2967. https://doi.org/10.3390/ma14112967

[9] Y. Zhong, Y. Wang, B. Zhang. (2024). Research on correlation between dielectric properties and compressive strength of concrete based on aggregate particle size. *Journal of Building Engineering*, 97, 110844. https://doi.org/10.1016/j.jobe.2024.110844

[10] M. Abazarsa, K. Raisi, T. Yu. (2024). Estimation of compressive strength of Portland cement concrete using synthetic aperture radar, ultrasonic testing, and rebound hammer. *Proceedings Volume 12950, Nondestructive Characterization and Monitoring of Advanced Materials, Aerospace, Civil Infrastructure, and Transportation XVIII; 1295006 (2024)*. https://doi.org/10.1117/12.3010516

[11] J. Chen, W. Du, G. Zhao, M. Shi, B. Xue. (2022). Effect of Aggregate Size and Water/Cement on Compressive Strength and Physiological Performance of Planting Concrete. *Materials*, 15(19), 6685. https://doi.org/10.3390/ma15196685

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

[13] H. Ha, H.V. Trong, T.L. Huyen, D.D. Nguyen, I. Prakash, B.T. Pham. (2025). Investigation of the Gaussian Process with Various Kernel Functions for the Prediction of the Compressive Strength of Concrete. *Engineering, Technology & Applied Science Research*, 15(1), 19992-19997. https://doi.org/10.48084/etasr.9125

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

[15] B.T. Pham. (2025). BGG-REPT and ROF-REPT: ensemble machine learning models for the prediction of compressive strength of concrete. *Innovative Infrastructure Solutions*, 10(2), 69. https://doi.org/10.1007/s41062-025-01869-3

[16] T. Shafighfard, F. Kazemi, N. Asgarkhani, D.-Y. Yoo. (2024). Machine-learning methods for estimating compressive strength of high-performance alkali-activated concrete. *Engineering Applications of Artificial Intelligence*, 136, 109053. https://doi.org/10.1016/j.engappai.2024.109053

[17] Z. Sun, Y. Li, Y. Li, L. Su, W. He. (2024). Investigation on compressive strength of coral aggregate concrete: Hybrid machine learning models and experimental validation. *Journal of Building Engineering*, 82, 108220. https://doi.org/10.1016/j.jobe.2023.108220

[18] M. Abdellatief, Y.M. Hassan, M.T. Elnabwy, L.S. Wong, R.J. Chin, K.H. Mo. (2024). Investigation of machine learning models in predicting compressive strength for ultra-high-performance geopolymer concrete: A comparative study. *Construction and Building Materials*, 436, 136884. https://doi.org/10.1016/j.conbuildmat.2024.136884

[19] H. Song, A. Ahmad, F. Farooq, K.A. Ostrowski, M. Maślak, S. Czarnecki, F. Aslam. (2021). Predicting the compressive strength of concrete with fly ash admixture using machine learning algorithms. *Construction and Building Materials*, 308, 125021. https://doi.org/10.1016/j.conbuildmat.2021.125021

[20] Z. Zeng, Z. Zhu, W. Yao, Z. Wang, C. Wang, Y. Wei, Z. Wei, X. Guan. (2022). Accurate prediction of concrete compressive strength

based on explainable features using deep learning. *Construction and Building Materials*, 329, 127082. https://doi.org/10.1016/j.conbuildmat.2022.127082

[21] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4), 541-551. https://doi.org/10.1162/neco.1989.1.4.541

[22] E. Cetinic, T. Lipic, S. Grgic. (2020). Learning the Principles of Art History with convolutional neural networks. *Pattern Recognition Letters*, 129, 56-62. https://doi.org/10.1016/j.patrec.2019.11.008

[23] V.N. Vapnik. (2000). The Nature of Statistical Learning Theory. Springer New York, NY. https://doi.org/10.1007/978-1-4757-3264-1

[24] A.M. Abd, S.M. Abd. (2017). Modelling the strength of lightweight foamed concrete using support vector machine (SVM). *Case Studies in Construction Materials*, 6, 8-15. https://doi.org/10.1016/j.cscm.2016.11.002

[25] H.-G. Ni, J.-Z. Wang. (2000). Prediction of compressive strength of concrete by neural networks. *Cement and Concrete Research*, 30(8), 1245-1250. https://doi.org/10.1016/S0008-8846(00)00345-8

[26] A. Öztaş, M. Pala, E. Özbay, E. Kanca, N. Çagˇlar, M.A. Bhatti. (2006). Predicting the compressive strength and slump of high strength concrete using neural network. *Construction and Building Materials*, 20(9), 769-775. https://doi.org/10.1016/j.conbuildmat.2005.01.054

[27] F. Murtagh. (1991). Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5), 183-197. https://doi.org/10.1016/0925-2312(91)90023-5

[28] H. Alibrahim, S.A. Ludwig. (2021). Hyperparameter Optimization: Comparing Genetic Algorithm against Grid Search and Bayesian Optimization. *2021 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1551-1559.https://doi.org/10.1109/CEC45853.2021.9504761

[29] Q. Huang, J. Mao, Y. Liu. (2012). An improved grid search algorithm of SVR parameters optimization. *2012 IEEE 14th International Conference on Communication Technology*, pp. 1022-1026. https://doi.org/10.1109/ICCT.2012.6511415

[30] F.J. Pontes, G.F. Amorim, P.P. Balestrassi, A.P. Paiva, J.R. Ferreira. (2016). Design of experiments and focused grid search for neural network parameter optimization. *Neurocomputing*, 186, 22-34. https://doi.org/10.1016/j.neucom.2015.12.061

[31] D.D. Nguyen, M.D. Nguyen, I. Prakash, N.V. Huong, H.V. Le, B.T. Pham. (2025). Prediction of safety factor for slope stability using machine learning models. *Vietnam Journal of Earth Sciences*, 47(2), 182-200. https://doi.org/10.15625/2615-9783/22196

[32] B.T. Pham, M. Amiri, M.D. Nguyen, T.Q. Ngo, K.T. Nguyen, H.T. Tran, H. Vu, B.T.Q. Anh, H.V. Le, I. Prakash. (2021). Estimation of shear strength parameters of soil using Optimized Inference Intelligence System. *Vietnam Journal of Earth Sciences*, 43(2), 189-198. https://doi.org/10.15625/2615-9783/15926

[33] V.-H. Nhu, B.T. Pham, D.T. Bui. (2023). A novel swarm intelligence optimized extreme learning machine for predicting soil shear strength: A case study at Hoa Vuong new urban project (Vietnam). *Vietnam Journal of Earth Sciences*, 45(2), 219-237. https://doi.org/10.15625/2615-9783/18338

[34] V.-H. Phan, H.-B. Ly. (2024). RIME-RF-RIME: A novel machine learning approach with SHAP analysis for predicting macroscopic permeability of porous media. *Journal of Science and Transport Technology*, 4(1), 58-71. https://doi.org/10.58845/jstt.utt.2024.en.4.1.58-71

[35] D.D. Nguyen, H.P. Nguyen, D.Q. Vu, I. Prakash, B.T. Pham. (2023). Using GA-ANFIS machine learning model for forecasting the load bearing capacity of driven piles. *Journal of Science and Transport Technology*, 3(2), 26-33. https://doi.org/10.58845/jstt.utt.2023.en.3.2.26-33

[36] S. Pal, L.H. Trang, V.T. Hieu, D.D. Nguyen, D.Q. Vu, I. Prakash. (2024). Investigation of Support Vector Machines with Different Kernel Functions for Prediction of Compressive Strength of Concrete. *Journal of Science and Transport Technology*, 4(2), 55-68. https://doi.org/10.58845/jstt.utt.2024.en.4.2.55-68

[37] I. Prakash, D.D. Nguyen, N.T. Tuan, T.V. Phong, L.V. Hiep. (2024). Landslide Susceptibility Zoning: Integrating Multiple Intelligent Models with SHAP Analysis. *Journal of Science and Transport Technology*, 4(1), 23-41. https://doi.org/10.58845/jstt.utt.2024.en.4.1.23-41

[38] M.D. Nguyen, D.D. Nguyen, H.N. Hai, A.H. Sy, P.N. Quang, L.N. Thai, D.N. Cong, I. Prakash, H.V. Le, B.T. Pham. (2024). Estimation of recompression coefficient of soil using a hybrid ANFIS-PSO machine learning model. *Journal of Engineering Research*, 12(3), 358-368. https://doi.org/10.1016/j.jer.2023.10.018

[39] P.T. Binh, D.D. Nguyen, Q.-A.T. Bui, M.D. Nguyen, T.T. Vu, I. Prakash. (2022). Estimation of load-bearing capacity of bored piles using machine learning models. *Vietnam Journal of Earth Sciences*, 44(4), 470-480. https://doi.org/10.15625/2615-9783/17177

[40] M.V. Le, I. Prakash, D.D. Nguyen. (2023). Predicting Load-Deflection of Composite Concrete Bridges Using Machine Learning Models. *Journal of Science and Transport Technology*, 3(4), 43-51. https://doi.org/10.58845/jstt.utt.2023.en.3.4.43-51

[41] I. Prakash, T.A. Nguyen. (2023). Predicting the Maximum Load Capacity of Circular RC Columns Confined with Fibre-Reinforced Polymer (FRP) Using Machine Learning Model. *Journal of Science and Transport Technology*, 3(4) 25-42. https://doi.org/10.58845/jstt.utt.2023.en.3.4.25-42

[42] H.-B. Ly, P.G. Asteris, B.T. Pham. (2020). Accuracy assessment of extreme learning machine in predicting soil compression coefficient. *Vietnam Journal of Earth Sciences*, 228–336. https://doi.org/10.15625/0866-7187/42/3/14999

[43] K.E. Taylor. (2001). Summarizing multiple aspects of model performance in a single diagram. *Journal of Geophysical Research: Atmospheres*, 106(D7), 7183-7192. https://doi.org/10.1029/2000JD900719

[44] J.H. Friedman. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451