Civil Engineering Infrastructures Journal, 2026, 59(1): 139-160
DOI: 10.22059/ceij.2024.380171.2118

RESEARCH PAPER

Integrating Machine Learning and Genetic Expression Programming for Enhanced Punching Shear Strength Prediction

Mahmoudian, A.[1], Tajik, N.[2], Darabi, A.H.[3], Mohammadzadeh Taleshi, M.[4], Marmarchinia, S.[2], Asghari, A.[5*] and Mirghaderi, R.[6]

[1] M.Sc., Department of Civil Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran.
[2] Ph.D. Candidate, Department of Civil, Structural and Environmental Engineering, State University of New York at Buffalo, USA.
[3] M.Sc., School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran.
[4] Ph.D. Candidate, Civil and Environmental Engineering Department, University of Nevada, Reno.
[5] Associate Professor, School of Civil Engineering, College of Engineering, University of Tehran, Iran.
[6] Professor, School of Civil Engineering, College of Engineering, University of Tehran, Iran.

Received: 30 Jul. 2024; Revised: 1 Sep. 2024; Accepted: 24 Sep. 2024

ABSTRACT: Estimating the punching shear strength of Reinforced Concrete (RC) flat slabs is critical in structural engineering due to potential catastrophic failures. This study introduces advanced data-driven methods, including Machine Learning (ML), Deep Learning (DL), and Genetic Expression Programming (GEP), to improve predictions of punching shear strength. Analyzing a dataset of 380 test samples, the research evaluates various models such as linear regression, stochastic gradient descent, ridge regression, decision trees, K-nearest neighbors, random forests, adaptive boosting, Extreme Gradient Boosting (XGBoost) for ML, alongside Artificial Neural Networks (ANNs) for DL, and GEP for deriving explicit equations. Significant enhancements in model performance were achieved through rigorous hyperparameter tuning, notably with the XGBoost model, which attained an coefficient of determination (R^2) score of 0.98, surpassing other models and existing code-based predictions. The study uses SHapley values to interpret model predictions, highlighting the significant impact of slab depth on punching shear strength, especially in the XGBoost model. Additionally, the GEP method derives explicit equations that accurately represent the relationship between input features and punching shear strength. This research highlights the advantages of advanced computational models and offers new insights into the factors influencing punching shear strength in RC slabs.

Keywords: Machine Learning, Artificial Neural Networks, Extreme Gradient Boosting, Punching Shear Strength, Genetic Expression Programming.

1. Introduction

Concrete structures are the most prevalent types of structures worldwide, with extensive research conducted on the load capacity of concrete elements such as

* Corresponding author E-mail: abazar.asghari@ut.ac.ir

beams, columns, and slabs. Flat slabs, critical elements in Reinforced Concrete (RC) structures, are directly supported by concrete columns without any intervening beams. These slabs are deemed cost-effective due to the reduced construction time compared to traditional slabs and are particularly advantageous in scenarios where higher headroom or lower story heights are required. Ensuring the structural safety of various components is paramount in structural engineering, highlighting the importance of detecting visible signs of failure before the actual collapse of members.

However, RC flat slabs, lacking beams as a primary load path between slabs and columns, are prone to brittle punching failures, often occurring without prior visible warning signs and resulting in sudden collapses, a situation that is critically concerning for civil engineers. This vulnerability is primarily attributed to the concentrated column reactions in the flat slabs. The collapse of RC flat slabs typically results from a combination of concrete crushing and the extensive spread of flexural fractures and punching shear.

Experimental tests conducted in the literature have led to the derivation of several punching shear strength formulations, which serve as the primary method for designing RC flat slab punching shear strength. However, the accuracy of these models, as proposed in various codes, has not been thoroughly investigated, casting doubts on their reliability (Americaan Concrete Institute, 2019; Canadian Standards Association, 2014; European Committee for Standardization, 2004; Standards Australia, 2011).

The literature on the punching shear behavior of RC slabs has primarily investigated the influence of various parameters, including the compressive strength of concrete, type of concrete, yield strength of reinforcement, reinforcement ratio, slab configurations, and conditions of support and load (Birkle and Dilger, 2008; Elstner and Hognestad, 1956; Guandalini et al., 2009; Ozden et al., 2006; Regan, 1986; Rizk et al., 2011; Theodorakopoulos and Swamy, 2002).

Notably, Liang et al. (2023) proposed a hybrid model known as the Symbolic Regression (SR) to estimate the punching shear resistance of Fiber Reinforced Polymer (FRP)-RC slabs, utilizing results from 154 experimental samples. Similarly, Liu et al. (2024) developed an explainable Extreme Gradient Boosting (XGBoost) model to predict the punching shear strength of flat slabs constructed from various types of Fiber-Reinforced Concrete (FRC), based on a comprehensive database of 251 flat slabs that include normal strength, high-performance, and ultra-high-performance FRC slabs. Furthermore, Alotaibi et al. (2021) assessed the performance of several machine learning algorithms, including Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs), for estimating the punching shear capacity of FRC slabs. The results show that ANN-based models perform best, with the slab's effective depth being the most influential factor. Consequently, these studies provide a vast amount of datasets which are beneficial for having a data-driven insight into the problem.

Nowadays, the integration of computer science skills such as Machine Learning (ML), Gene Expression Programming (GEP), Finite Element Analysis (FEM), and probability analysis with other fields of expertise like structural engineering has proven to be successful (Al-Bayati, 2023; Anjali et al., 2023; Li and Li, 2023; Marmarchinia et al., 2025; Palomino Ojeda et al., 2023; Tavasoli, 2023), and there have been various models in numerous studies which have been conducted for damage assessment in structures (Afzali et al., 2023; Aminian et al., 2011; Bypour et al., 2024; Gandomi et al., 2015; Hamidia et al., 2024; Jamshidian and Hamidia, 2023; Mirzahosseini et al., 2019; Naderpour et al., 2024; Tajik et al., 2024; Taleshi et al., 2024; Zaker Esteghamati, 2024; Zaker Esteghamati and Huang, 2023).

However, investigating the punching

shear strength of RC flat slabs without transverse reinforcement has not been taken seriously in the literature over the years. For instance, in the study of Abambres and Lantsoght (2020), only the important parameters in a structure on the shear capacity prediction of one-way slabs under concentrated loads, which are the width of the slab, the effect of the beam span-to-depth ratio, and concrete compressive strength, have been introduced. According to the literature, some Deep Learning (DL) models have been developed with acceptable accuracies, such as the DL models of Tran and Kim (2021); where a total number of 218 sample data were used to develop the ANN models for predicting the punching shear strength of two-way RC slabs, where the coefficient of determination (R^2) score of the implemented model was 0.995.

Mangalathu et al. (2021) investigation of ML models, revealed that the R^2 score accuracy of the XGBoost model (one of the well-known ML models that have been used in predicting the punching shear capacity) is 0.98 for the estimation of punching shear capacity of RC slabs. Additionally, some studies focus on implementing ML and metaheuristic methodologies into flat slab design, such as the work by Alkhawaldeh (2024), which combined the Light Gradient Boosting Machine (LGBM) and Locust Swarm Algorithm (LSA) to improve punching shear strength predictions in flat slabs. This study addresses a crucial gap in the current understanding and prediction of punching shear strength in RC flat slabs.

While traditional code-based methods offer a standardized and generally reliable approach, their predictive accuracy, although good, can still be significantly enhanced. This limitation presents an opportunity for improvement in ensuring infrastructure safety, underscoring the need for more precise and robust predictive models. The primary goal of this research is to develop and validate advanced data-driven models, including ML, DL, and GEP techniques, to predict punching shear strength with higher accuracy than current code formulations.

By comparing these models' performance against existing codes, this study aims to demonstrate the superiority of modern computational methods in capturing the complex interactions that govern punching shear failure. It has been tried to use the grid-search method for tuning the hyperparameters of the ML model (Jiang and Xu, 2022) for capturing higher accuracy of prediction for the existing 380 experimental sample dataset. Then, to propose an explicit formula to calculate the punching shear capacity of conventional concrete slab, GEP is implemented, which provides the function between inputs and outputs in a given dataset.

The first section of this study aims to present and describe various ML models suitable for certain applications, highlighting the advantages and disadvantages of each algorithm. Following this, the study delves into the utilized dataset and describes the implemented methods, including linear regression, Stochastic Gradient Descent (SGD), ridge regression, SVR, K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest (RF), Adaptive Boosting (AdaBoost), XGBoost, and ANNs. This section also introduces the evaluation metrics used to measure the accuracy of these models.

Given the opaque nature of ML and DL algorithms, subsequent sections endeavor to demystify the impact of each model feature on the final predictions through the use of Shapley values. Additionally, the study explores the generation of mathematical equations for predicting the target value using GEP. A separate section is dedicated to comparing the outputs generated by ML algorithms with those of conventional design codes and formulations, demonstrating the effectiveness of these modern tools. Finally, the study concludes with a summary and remarks on the findings presented in the last section.

2. Overview of ML

2.1. ML and DL Methods

ML methods are now widely used to make logical predictions for unseen datasets by training with a limited amount of data. Civil engineers have seized this opportunity, striving to enhance their research by incorporating these techniques. Among these, linear regression is a well-known algorithm that trains a model as a linear combination of dependent variables.

This study investigates nine linear regression models, including linear regression, SGD, ridge regression, SVR, KNN, DT, RF, AdaBoost, and XGBoost, to predict the final punching shear capacity of RC flat slabs without transverse reinforcement. In the case of simple linear regression, just one feature and one target are involved.

However, multiple linear regressions are used since the dataset has several features and a single dependent variable. But in the famous ML libraries in the Python programming language, linear regression specifically means using the normal equation for regression tasks. As in Eq. (1), the normal equation is a closed-form solution used to determine the parameters (W) that minimize the cost function.

$$W = (X^{\mathrm{T}}X)^{-1}.(X^{\mathrm{T}}y) \quad (1)$$

where W: is the vector of regression coefficients, X: is the matrix of input features, X^T: is the transpose of the feature matrix, and y: is the vector of target values. The term $(X^TX)^{-1}$: represents the inverse of the matrix product X^TX.

SGD is a numerical method used to decrease the amount of loss function. In fact, SGD works as an optimizer for a model that helps to converge the model. Moreover, SGD is based on trial and error, which is why it has relatively high calculation costs and errors. Support vector regression is one of the reputed regression algorithms used to predict independent variables (Parbat and Chakraborty, 2020).

KNN is a popular ML non-parametric algorithm, and the output is estimated as the weighted average of the KNN (Mangalathu et al., 2021). Model ensembling is another method that refers to the process of employing several models to get an improved prediction performance. In ensemble models like random forests, each base model is a decision tree, and the result of the random forest model is the aggregate output of these decision trees. In random forests, all the base models are constructed independently using different subsamples of the data (Figure 1).

The RF model efficiently handles tabular data with numerical or categorical features that have fewer than hundreds of categories. Unlike other linear models, RF can capture nonlinear interactions between features and targets. AdaBoost is an ensemble of many decision tree models, each of which is a weak learner and is slightly better than random guessing. However, this algorithm carries the gradient of previous trees to the next ones to improve the error of the previously mentioned trees.

Thus, this subsequent learning of trees at each step builds up a strong learner. The final prediction is the weighted average of the predictions given by each tree. Because of high adaptability, AdaBoost is more sensitive to outliers data, which is a key requirement in the case of this study (Patil et al., 2018). In XGBoost, trees are grown sequentially, and continuous scores are allocated to each leaf (Chen and Guestrin, 2016). In addition to the mentioned models, in this study, the ANNs models are also being used to improve the accuracy of punching shear strength prediction in the flat slabs.

Figure 2 shows the inter-relationship between ANNs and ML. Consider ANN as a subset of DL, which is a subset of ML, which is a branch of artificial intelligence. It should be noted that ANNs are usually used for large datasets due to their increased model capacity. However, depending on the model type, it can sometimes be useful for small datasets. The main concept behind using the ANN approach is that it learns adaptively from experience and extracts various functions, each appropriate for its purpose.

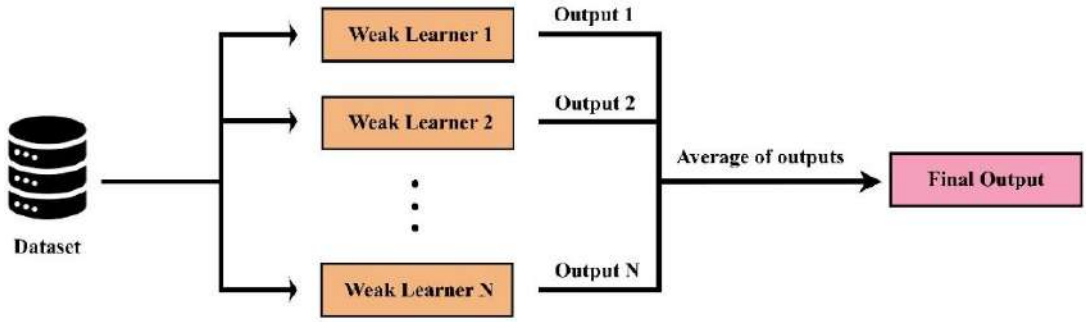


Fig. 1. Schematic of the random forest model

ANN can operate on large quantities of data and learns complex model functions from examples by training on a set of inputs and the corresponding outputs. ANNs can take into account the nonlinear and complex interactions that take place among the variables of a system without the need for assuming the form of the relationship that exists between the independent variables and those that are dependent. This is the primary advantage that ANN has over more conventional modeling techniques (Soleimani-Babakamali and Esteghamati, 2022; Tran and Kim, 2021).

An ANN model is a mathematical tool for imitating human brain functions like learning, reasoning, and performing heavy parallel computations. The smallest unit that makes up an ANN model is referred to as a neuron, and it is developed in three distinct levels (Mehrzad et al., 2023). These layers are the input layer, the hidden layer, and the output layer (Tran and Kim, 2021).

As the grid search parameters in neural networks were reviewed in previous sections, the aforementioned parameters were implemented with the dataset studied in this research, and the following results were achieved. There are various metrics to evaluate the accuracy of a model (Habib and Yildirim, 2022; Habib et al., 2023), which Mean Squared Error (MSE) being the most common. This method, which is shown in Eq. (2), considers the average squared difference between the actual and predicted values as the error. However, the problem with this method is that the MSE value is not understandable to users, so another method, shown in Eq. (3), called R^2 score is used. R^2 score is a statistical measure representing the proportion of the variance for a dependent variable explained by an independent variable or variables in a regression model.

$$\text{MSE} = \frac{\sum(y_i - \hat{y}_i)^2}{n} \quad (2)$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad (3)$$

where *MSE*: is the mean squared error, y_i: is the actual value, $\hat{y}_i$: is the predicted value, n: is the total number of samples, R^2: is the coefficient of determination, y_i: is the actual value, $\hat{y}_i$: is the predicted value, $\bar{y}$: is the mean of the actual values, and N: is the total number of samples.

2.2. Overview of GEP

In artificial intelligence, GEP is a robust and advanced evolutionary algorithm, motivated by natural selection and genetics concepts. Like the process of gene expression in natural living organisms, GEP mimics their process to solve complicated problems.

Since Ferreira (2001), initially presented GEP, it has grown in popularity due to its effectiveness and adaptability. The GEP algorithm can be used to achieve a mathematical function between the features and the targets. In this function, logical operators (AND, IF, …), algebraic operators (+ - × /) and algebraic functions such as trigonometric, exponential, etc., could be used. In order to implement the GEP framework, first a linear chromosome population needs to be formed, which can be single or multi-gene (Figure 3 depicts a chromosome example with *N* genes). One of the variables, targets, or guessed mathematical operators may be placed in each gene position of this chromosome.

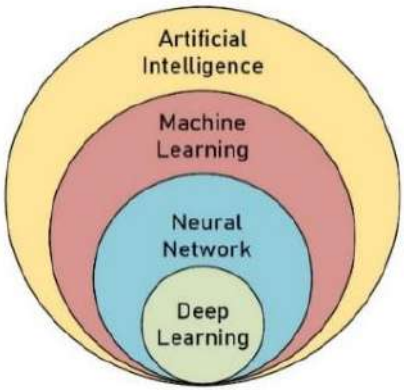


Fig. 2. AI vs. machine learning vs. neural network vs. deep learning

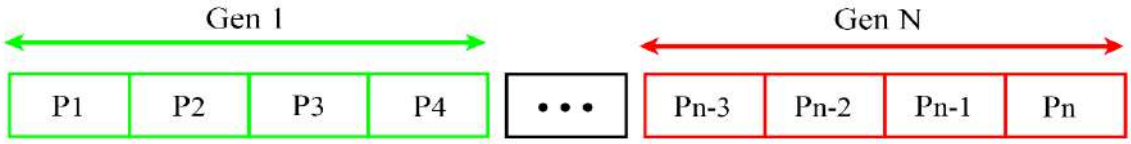


Fig. 3. An example of a chromosome

Some rules have been established by the inventor of the algorithm for choosing the length of the gene and the placement of these variables:

- Every gene has a head and a tail section, and the functions of a gene cannot be assigned to the tail part.
- As seen in Eq. (4), the maximum number of function parameters (n) determines the tail section length (t), while the head section length (h) is user-specified.

$$t = h(n-1) + 1 \quad (4)$$

where t: is the length of the tail section of the gene, h: is the length of the head section specified by the user, and n: is the maximum number of arguments or parameters used by the functions in the gene.

Then, it is time to assess each chromosome's fitness in the generation after chromosomes have been created and inserted into their proper locations. For this purpose, in the GEP algorithm, chromosomes are expressed as Tree Expression (TE), enabling the algorithm to assess and evolve the most promising solutions within each generation. Because of its creative methodology and adherence to genetic principles, GEP is recognized as a strong and trustworthy instrument for handling challenging problems in a variety of intersecting fields (Ferreira, 2001; Mansouri et al., 2021). The flowchart of the GEP process is shown in Figure 4.

2.3. Shapley Values

ML models have been increasingly utilized to address a wide range of problems, yet the manner in which results are derived from these models often remains scrutinized. To aid in the interpretation of results produced by ML models, techniques such as Shapley values have proven to be effective. Based on the principles of game theory, Shapley values significantly enhance the interpretability of ML models. The Shapley Additive Explanations (SHAP) method, which is grounded in the Shapley value theory from cooperative game theory, was initially introduced by Lundberg and Lee (2017). This method has since been recognized for its ability to provide clear explanations for the output of ML models, making it a pivotal tool in understanding and justifying the decision-making processes of these models (Figure 5). The SHAP method employs two key equations to facilitate the interpretation of ML models through an additive feature attribution approach. This approach decomposes a model's output into the sum of contributions from its input variables, enhancing interpretability.

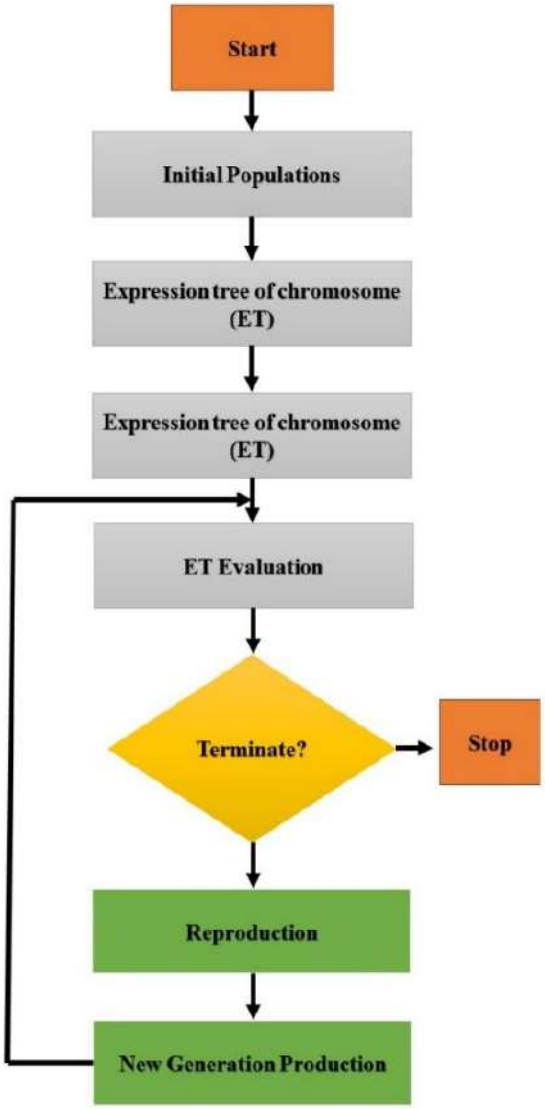


Fig. 4. Flowchart of GEP process

Eq. (5) outlines the basic form of the explanation model, $g(x')$, which uses a simplified version of the input, x', to approximate the original model's behavior, $f(x)$. In this equation (Lundberg and Lee, 2017):

$$f(x) = g(\acute{x}) = \varphi_0 + \sum_{i=1}^{M} \varphi_i x_i' \quad (5)$$

where x': represents the simplified input variables, derived from the original input variables, M: denotes the total number of features, φ_0: is the constant value indicating the model's output in the absence of all inputs, and φ_i: signifies the attributed contribution of each variable to the model's output.

Eq. (6) provides a formula for calculating the contribution of each feature. The parameters in this equation include (Lundberg and Lee, 2017):

$$\varphi_i(f.x) = \sum_{Z' \sqsubseteq x'} \frac{|z'|!\ (M - |Z'| - 1)!}{M!} [f_x(z') - f_x(z' \backslash i)] \quad (6)$$

where $|z'|$: represents the count of non-zero elements within Z', indicating the number of active features in the subset, and $Z' \sqsubseteq x'$ signifies all instances of Z' vectors where their non-zero elements correspond to a subset of the non-zero elements found in x', focusing on specific combinations of features.

2.4. Dataset

Achieving a precise ML model entails having a clean dataset. Dataset plays a vital role in ML models. The preprocessing of the dataset is more significant than the dataset itself, which requires great accuracy. In this study, 380 samples, each with six independent variables and one dependent variable, have been investigated (Mangalathu et al., 2021). Since the target variable under study is a discrete variable, it is clear that regression-based models should be used, and these algorithms are briefly introduced in the previous section.

To evaluate the model effectively, not all available data is presented to it during training. A portion of the dataset, typically 10-30%, is reserved as test data. This allows for the evaluation of the model's performance after the learning process is complete. It is crucial that the test data be randomly selected to ensure it is representative of the entire dataset, facilitating a comprehensive evaluation of the model. The normal distribution, also known as the Gaussian distribution, is a symmetric probability distribution centered around the mean, which indicates that data points close to the mean are more common than those far from it.

This study presents the normal distribution of each feature and target to highlight this aspect, as shown in Figure 6.

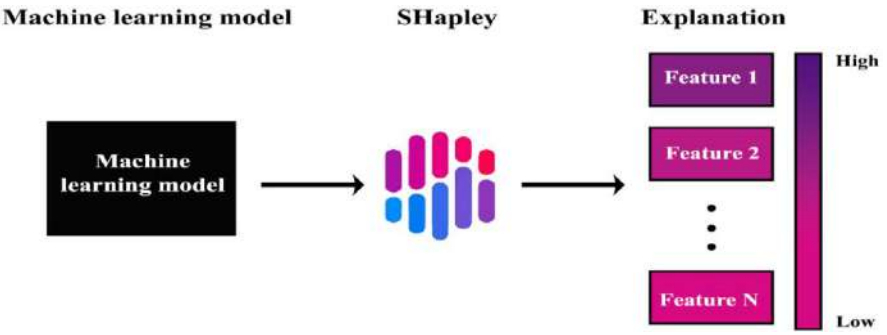
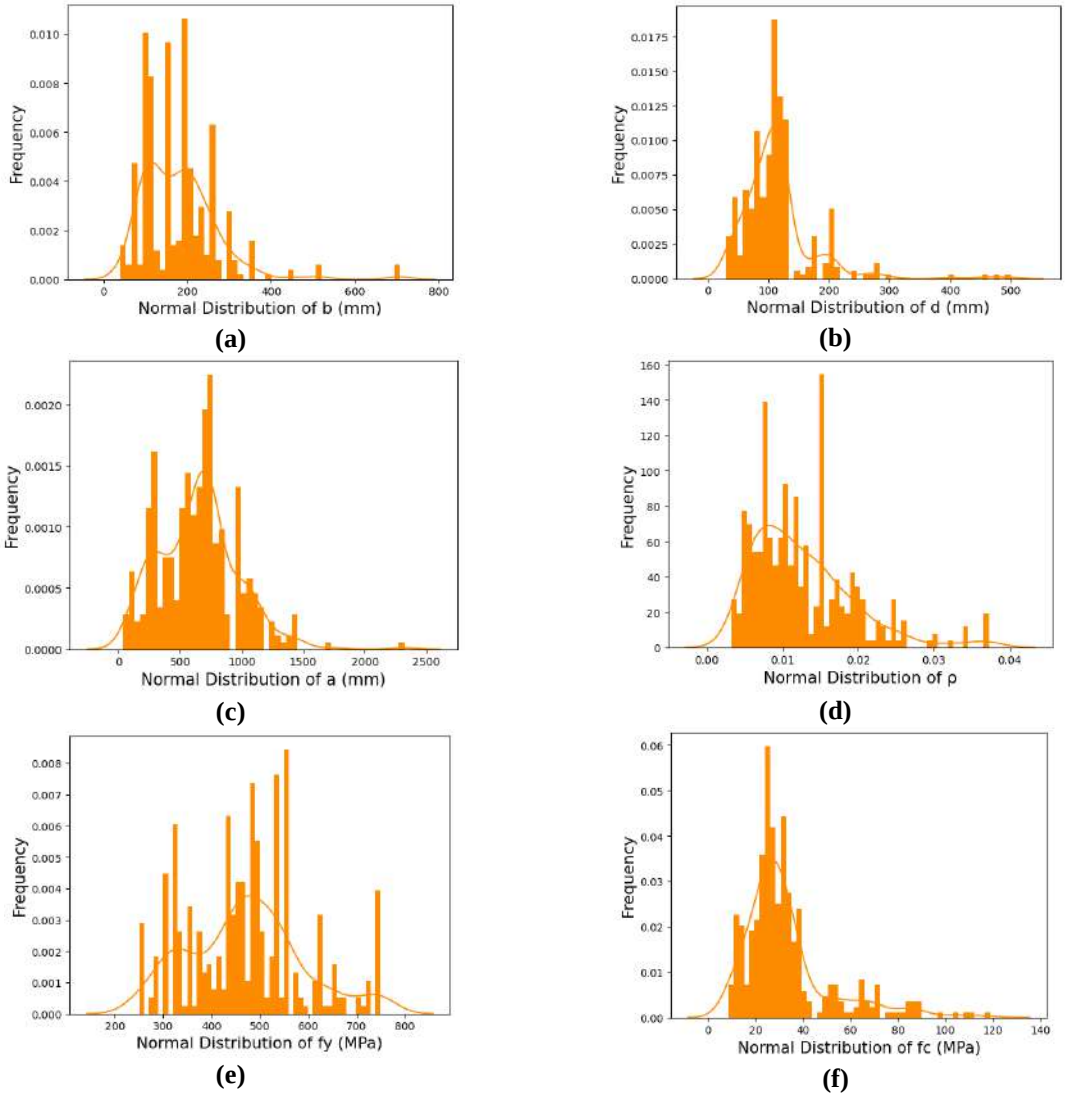
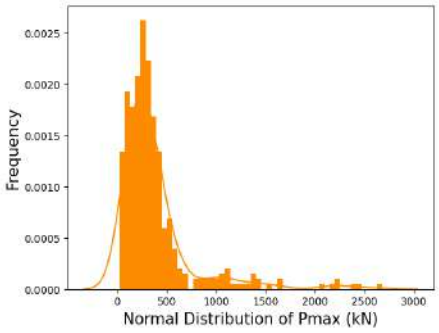


Fig. 5. Schematic of the Shapley values method

2.5. Grid Search

Grid search is designed to conduct hyperparameter tuning systematically by automatically going through a possible set of hyperparameter values during learning. In a grid search, each hyperparameter is given a series of values, and the program will then iterate through every hyperparameter value combination possible to train models (Jiang and Xu, 2022). Hyperparameter tuning and grid search were applied to optimize model performance in this study, which helped to achieve high accuracy in machine-learning models (Figure 7). It should be noted that not all parameters of a model are considered hyperparameters. However, the most important ones, which probably significantly impact the model's accuracy, are considered hyperparameters.





(g)

Fig. 6. Normal distribution of features and outputs: a) *b*; b) *d*; c) *a*; d) ρ; e) f_y; f) f_c; and g) P_{max}

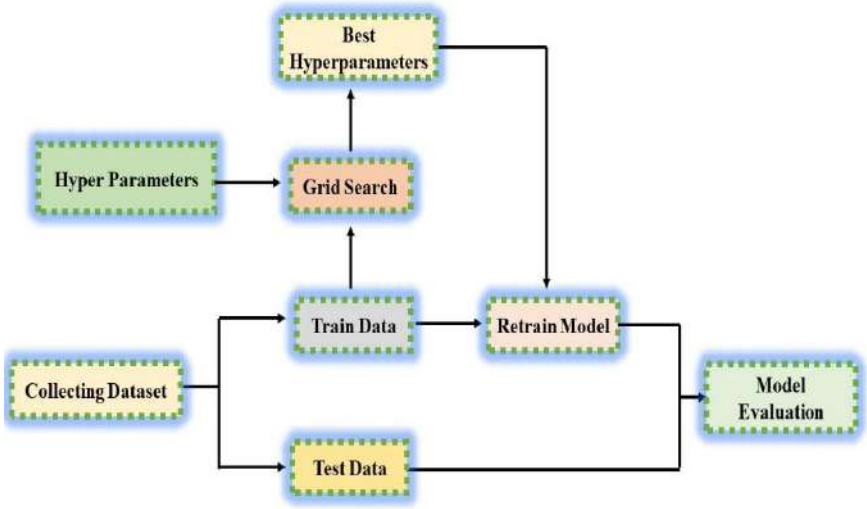


Fig. 7. Flowchart of grid search

In the SGD model, 81 models used different values for the four arguments in Table 1. Among them, the penalty argument that prevents model overfitting and the learning rate determining the speed of reaching the minimum point in the cost function are more important. As shown in Table 2, 120 different models have been tested for KNN; the most important argument of KNN is the number of neighbors in predicting punch shear strength values. It should be noted that selecting a small number of neighbors, such as one or two, leads to model overfitting.

Tables 3 and 4 show that 50 different models have been used for DT and RF algorithms, each having different values for the arguments mentioned. Criterion is the function to measure the quality of a split, and max_depth expresses the maximum depth of a decision tree in which the chosen numbers are suitable and widely used. Also, n_estimators is the number of trees used in the random forest.

Tables 5 and 6 show that 648 models for XGBoost and 60 models for AdaBoost with different values for the mentioned arguments have been tested. The number of trees in the ensemble often increased until no further improvements were seen. The boost parameter specifies the type of learner. In most cases, this is either a tree or a linear function. In the case of trees, the model will consist of an ensemble of trees. As shown in Table 7, 162 models with different parameters for different arguments have been used in ANNs.

Table 1. SGD grid search parameters

Penalty	Max_iter	Learning_ rate	Alpha
l1	1000	Constant	0.000 1
l2	1500	Optimal	0.001
Elasticnet	2000	Invscaling	0.01

Table 2. KNN grid search parameters

Neighbors	Weights	Algorithm	P
3	Uniform	Auto	1
4	Distance	Ball_tree	2
5	-	Kd_tree	3
6	-	Brute	-
7	-	-	-

The first and second columns show the number of neurons in each layer. Another important parameter is the batch size, which indicates how much data is entered into the model in each iteration. Then, the type of optimizer, the most well-known of which is Adam, is specified, and finally, the activation function is added to the model after the last layer.

Table 3. DT grid search parameters

Criterion	Splitter	Max_depth
Squared_error	Best	5
Friedman_mse	Random	10
Absolute_error	-	15

Table 4. RF grid search parameters

N_estimators	Criterion	Max_depth
100	Squared_error	5
200	Absolute_error	10
300	-	15
400	-	20

Table 5. AdaBoost grid search parameters

N_estimators	Learning_rate	Loss
50	0.001	Linear
80	0.01	Square
100	0.1	Exponential
150	1	-
250	-	-

Table 6. XGBoost grid search parameters

N_ estimators	Max_ depth	Bace_ score	Learning_rate
100	2	0.4	0.05
200	4	0.5	0.1
400	6	0.6	0.2
600	8	0.7	-
800	10	0.8	-
1000	15	1	-

Table 7. ANNs grid search parameters

Neurons in Layer 1	Neurons in Layer 2	Batch size	Optimizer	Active function
16	16	64	Adam	ReLU
32	32	128	Sgd	Linear
64	64	256	Rmsprop	-

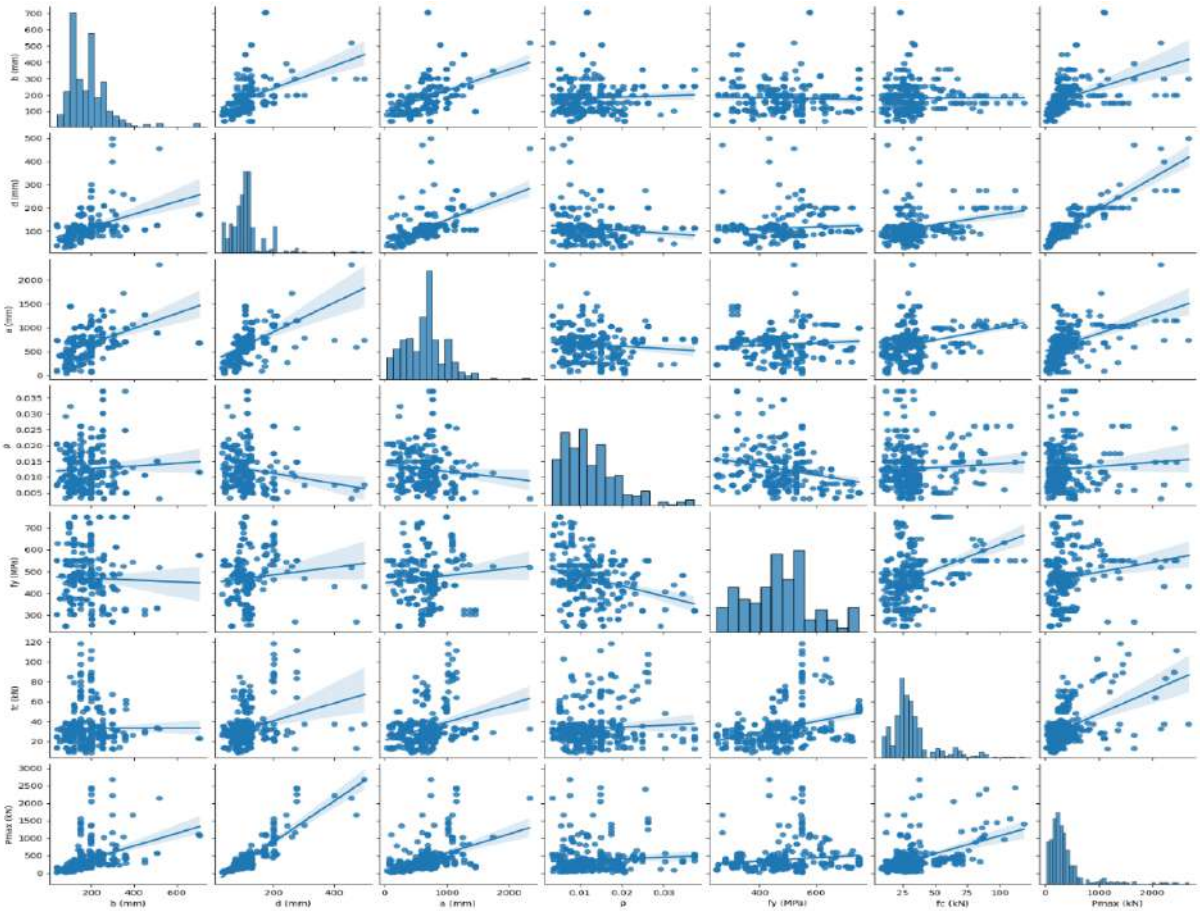


Fig. 8. Flowchart of grid search

3. ML and DL Models of Punching Shear Strength

3.1. Features

The important parameters used in this study to calculate the punching shear strength (V_n) are divided into two categories: (I) Parameters related to materials include compressive strength of slab concrete and slab flexural reinforcement yield strength, and (II) Parameters related to geometry include the effective flexural depth of slab (d), which expresses the average value of the effective flexural depth of the slab in two orthogonal directions, the shear span (a), which shows the distance between the slab supports and the face of the column, the slab reinforcement ratio (ρ), which is an estimate of the slab reinforcement ratio it is orthogonal in two directions and the equivalent width of the column (b).

Because punching shear occurs in a critical perimeter (b_0), it is necessary to find this value using different codes. The critical perimeter is located at a certain distance from the face of the column, for which different codes have stated different values. For example, ACI-318-14 (American Concrete Institute, 2014), CSA A23.3-19 (Canadian Standards Association, 2019), and AS 3600 (Standards Australia, 2011) consider the critical section at half of the effective flexural depth of the slab from the face of the column, while Eurocode 2 (European Committee for Standardization, 2004) considers it twice the depth from the face of the column. However, this study follows ACI-318-14 (American Concrete Institute, 2014).

For a better understanding of the relation between the features, Figure 8 is used. This scatter plot indicates that the closer the slope of the regressor is to the angle of 45 degrees, the more the two features are related to each other. Also, the exact relation is expressed through the correlation matrix shown in Figure 9. To be more precise, the slope of the regressor drawn in the scatter plot shows that the relationship between d and P_{max} is close to an angle of 45 degrees, indicating the strong relation between these two features. A correlation matrix is a matrix that shows the relationship and correlation of dataset features with each other. This matrix's number of rows and columns equals the number of features of the dataset. Each cell is marked with a color ranging from minus one to plus one. The closer this number is to minus one, it means that these two features are inversely related to each other, and the closer this number is to plus one, it means that two features are directly related to each other.

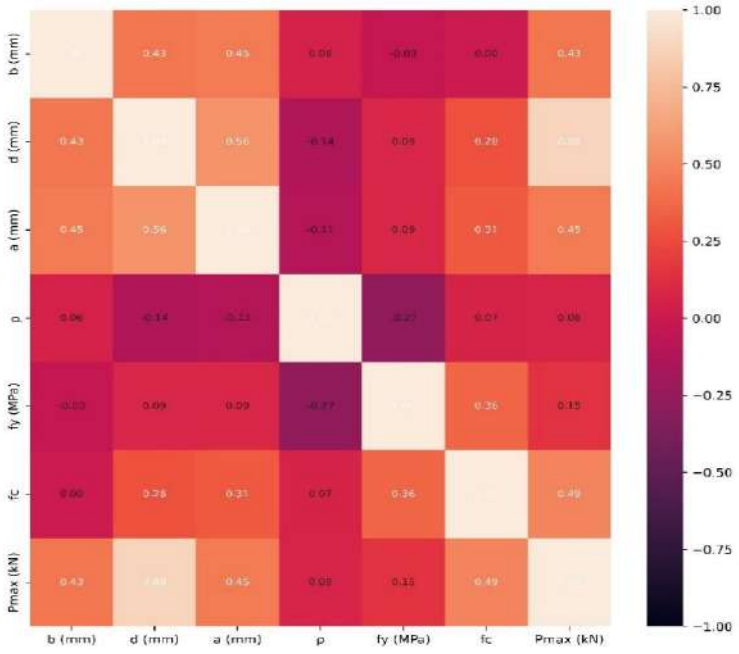


Fig. 9. Features and output correlation matrix

This matrix is symmetric, and the main diagonal of this matrix is equal to one because each feature naturally has a maximum correlation with itself. In this study, the correlation matrix is a matrix that has seven rows and seven columns. For example, in Figure 9, the maximum compressive strength that enters the column (P_{max}) has a positive correlation (+ 0.88) with the effective depth of the slab (d) and is marked with red color. This shows that when d increases, so does P_{max}.

3.2. ML and DL with Grid Search Results

The first method used in this study is linear regression based on a normal equation shown in Figure 10. Because this method is derived from an explicit equation, there is no need for trial and error and special hyperparameter tuning. Of course, it can achieve different accuracies by changing the training data size or the normalization method, but the difference in these accuracies is insignificant in this study. With a train size of 0.8 and the StandardScaler normalization method, the accuracy of this model on test data has been achieved at 0.88, as shown in Figure 10.

The next method used in this article is the SGD method. This method is based on trial and error and has various hyperparameters that can be changed, especially the learning rate. This method's best possible accuracy can be achieved by evaluating different hyperparameter values using a grid search, as shown in Table 8 and Figure 11. Although KNN is mostly used for classification tasks, it is also used in regression problems.

As demonstrated in Figure 12, applying an appropriate grid search, shown in Table 9, to this model can achieve high accuracy. In this model, according to the number of neighbors and the type of distance, desired neighbors are selected, and their average label is considered the output label. According to the KNN grid search results in Table 9, the optimal number of neighbors in this dataset is 4, and the power distance (p) type is 2.

The decision tree model usually works very well in training samples, but its test sample accuracy is not the same as the training sample accuracy. Actually, this does not imply that overfitting has occurred in this model; rather, it indicates that its accuracy on the training data is much higher. The grid search results in the DT model showed that the max_depth would probably not have a remarkable effect on the accuracy achieved in this dataset, as shown in Table 10. Furthermore, applying the grid search method for the DT model resulted in an accuracy of 0.95, as shown in Table 10.

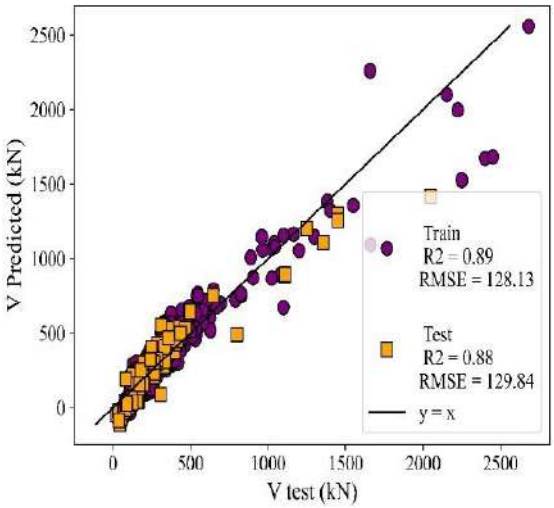


Fig. 10. The best linear regression model results

Table 8. Results of grid search for SGD

R^2 score train	R^2 score test	Alpha	Penalty	Learning_rate
0.89	0.88	0.1	l1	Constant

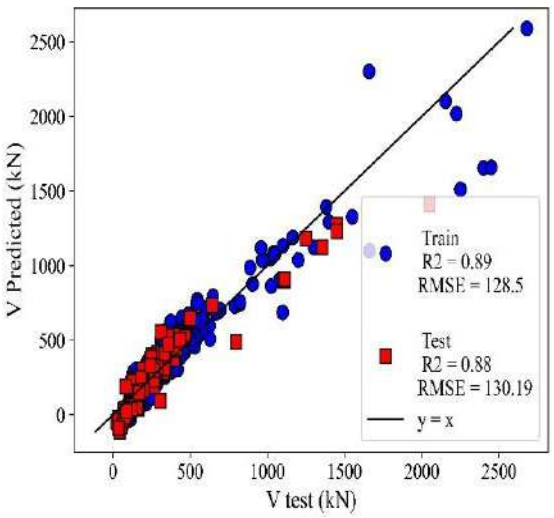


Fig. 11. The best SGD regression model results

Table 9. Results of grid search for KNN

R^2 score train	R^2 score test	Neighbors	Weights	Algorithm	*P*
1	0.93	2	Distance	Auto	2

Random forest, AdaBoost, and XGBoost models are based on ensemble learning. In this field, techniques have been proposed that use several models in a combined and simultaneous way to make decisions to increase the model's power in estimating the data output. As demonstrated in Table 11, similar to DT, the max_depth parameter in RF models has little effect; however, this data set clearly shows that squared error is the best criterion value, and applying the grid search method to the RF model resulted in an accuracy of 0.95, as shown in Figure 14.

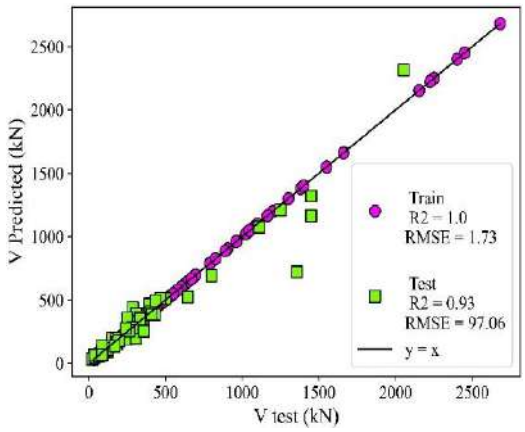


Fig. 12. The best KNN regression model results

Table 10. Results of grid search for DT

R^2 score train	R^2 score test	Criterion	Splitter	Max_ depth
1	0.95	Friedman-mse	Random	15

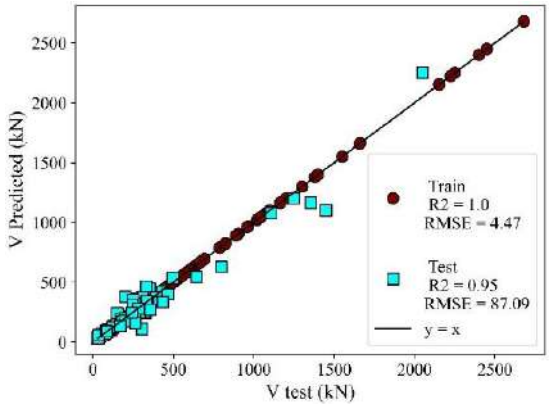


Fig. 13. The best DT regression model results

Table 11. Results of grid search for RF

R^2 score train	R^2 score test	N_ estimators	Criterion	Max_depth
0.99	0.95	100	SquarEd_error	10

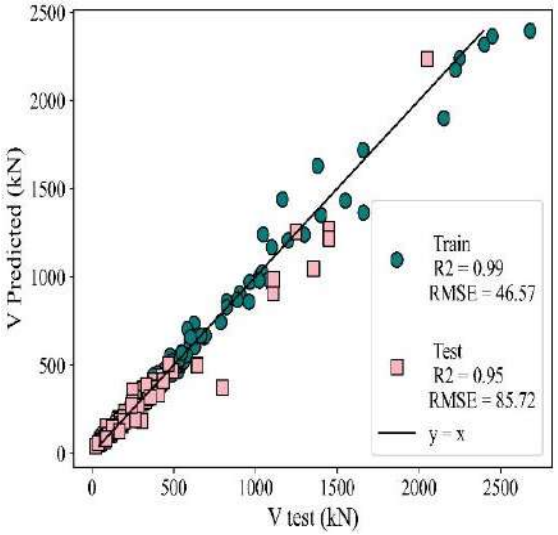


Fig. 14. The best RF regression model results

Applying grid search has led to various results in the AdaBoost models, as shown in Table 12, but it is easy to understand that the best value for the learning rate is 0.1. As shown in Figure 15, the grid search method achieved the best AdaBoost model accuracy of 0.91. In the XGB algorithm, as many models as possible have been tested to achieve the best results. After seeing the results in Table 13, it can be concluded that the best input values for the arguments of N_estimators, max_depth and Learning_rate are 300, 2, and 0.05, respectively. As seen in Figure 16, the best accuracy achieved in XGB is 0.98. The findings from these models demonstrate that the XGB model outperforms the ANNs, exhibiting higher accuracy on this specific dataset.

Table 12. Results of grid search for AdaBoost

R^2 score train	R^2 score test	N_estimators	Learning_rate	Loss
0.94	0.91	150	0.1	Exponential

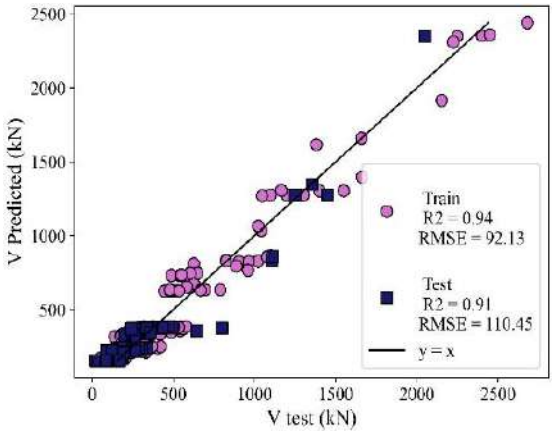


Fig. 15. The best AdaBoost regression model results

Table 13. Results of grid search for XGB

R^2 score train	R^2 score test	N_estimators	Max_depth	Learning_rate
0.9975	0.9802	300	2	0.05

Despite achieving commendable accuracy with ML models, further investigations were conducted using neural network models varying in neuron counts, activation functions, and optimizers. According to the results presented in Table 14 and Figures 17 and 18, the accuracy attained by neural networks falls short of that achieved by the best ML model. It was also determined that the Adam optimizer and the ReLU activation function are the most effective for this dataset.

3.3. GEP Model Proposal

A section of this study proposes a precise model for predicting the punching shear strength of RC flat slabs, using the GEP approach. The model inputs adhere to the standards commonly employed in ML practices. Through the application of the GEP technique, an empirical model was constructed to forecast the punching shear strength of RC flat slabs, as elucidated in Eq. (7).

$$
\begin{aligned}
V &= V_1 + V_2 + V_3 + V_4 \\
V_1 &= d_3(d_4 + d_1 - d_2) + \frac{d_5 d_1}{39.25} \\
V_2 &= d_3(d_4 + d_0 - 3.67) + \frac{d_5}{10.69} \\
V_3 &= -8.91 d_3 d_1 + \frac{d_0}{e^{d_0}} \\
V_4 &= d_3 {d_1}^2 + \frac{d_0 d_1}{247.15}
\end{aligned}
\tag{7}
$$

where V: is the predicted punching shear strength of the RC flat slab, while V_1, V_2, V_3, and V_4: are intermediate sub-expressions generated by the GEP model. The parameters d_0, d_1, d_2, d_3, d_4, and d_5: represent the normalized input variables used in the GEP model. The term e: is the base of the natural logarithm.

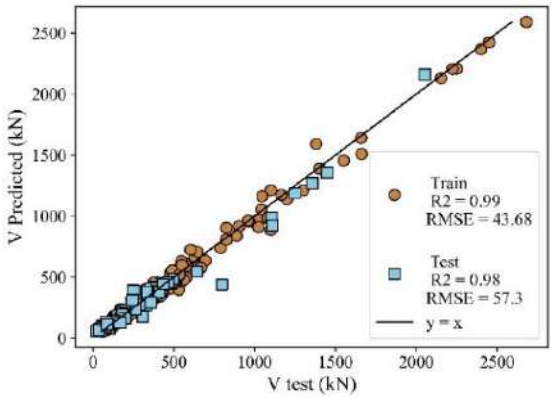


Fig. 16. The best XGB regression model results

Table 14. Results of grid search for ANNs

R^2 score train	R^2 score test	Layer 1	Layer 2	Batch size	Optimizer	Activatefuntion
0.97	0.96	32	64	32	Adam	ReLU

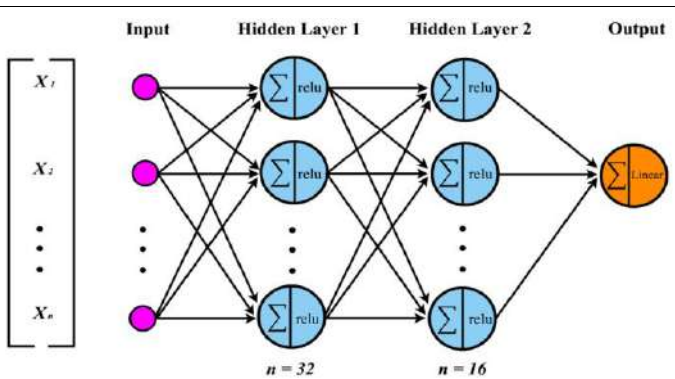


Fig. 17. The architecture of ANNs regression model

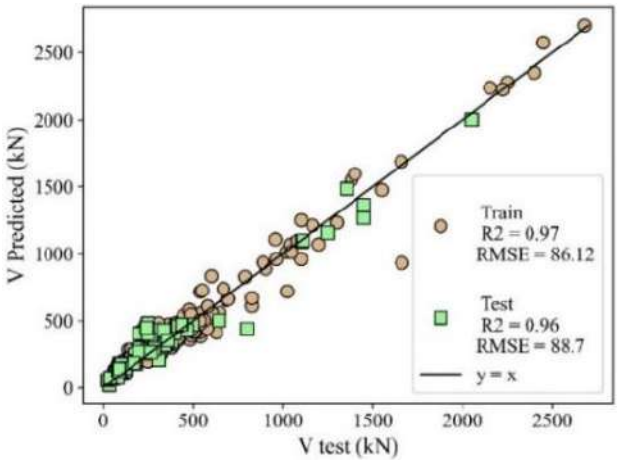
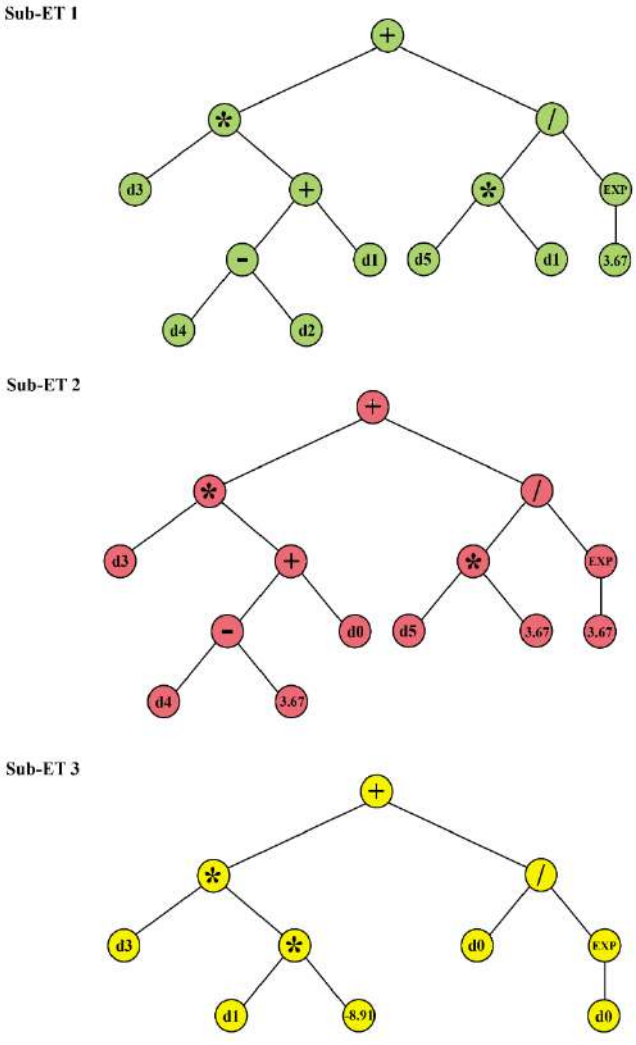


Fig. 18. The best ANNs regression model results

Figure 19 shows the expression tree of the estimation model. Figure 20 demonstrates its results, and Table 15 details the operational and functional specifics of the proposed model.

3.4. Feature Importances

In order to identify the black box of ML models, especially the XGB model, which exhibits the highest accuracy among the models in this study, SHapley values are employed. Figures 21 and 22 reveal that the slab depth (*d*) is the most significant feature in this model, whereas the feature *fy* (MPa) ranks lowest in terms of importance.



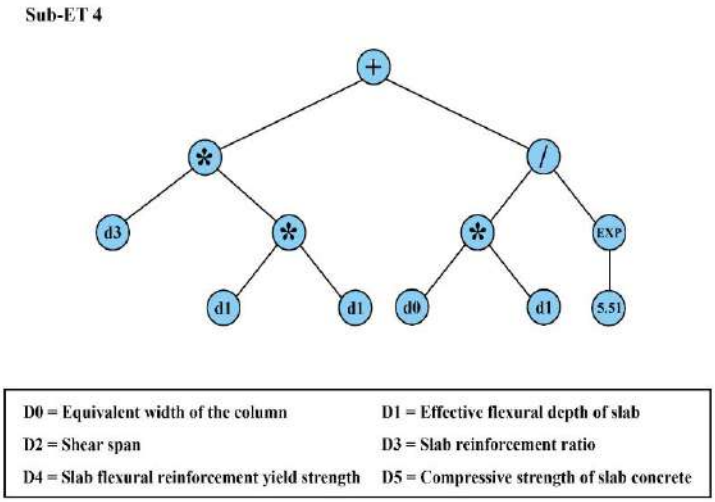


Fig. 19. Expression Tree (ET) of the GEP model proposed

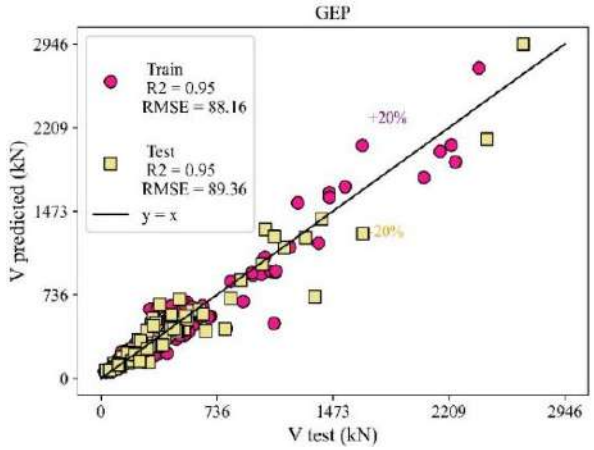


Fig. 20. GEP model results

Table 15. GEP model parameters

Function	(+ - × /) , pow, sqrt, exp, ln, sin, arctan, tanh, not
Number of generations	63919
Chromosomes	30
Head size	8
Number of genes	3
Linking function	Addition
Mutation	0.44
IS transposition	0.1
RIS transposition	0.1
One-point recombination rate	0.2
Two-point recombination rate	0.3
Gene recombination	0.2
Gene transposition	0.1

While Figure 21 presents the SHapley values separately, Figure 22 shows the average impact of each feature on the model's predictions.

3.5. Empirical Design Code Equations

The codes used in this study to calculate punching shear strength in two-way slabs without transverse reinforcement are ACI 318-14 (American Concrete Institute, 2014), ACI 318-19 (American Concrete Institute, 2019), ACI 440.1R-06 (American Concrete Institute, 2006), Eurocode 2 (European Committee for Standardization, 2004), and CSA A23.3-14 (Canadian Standards Association, 2014).

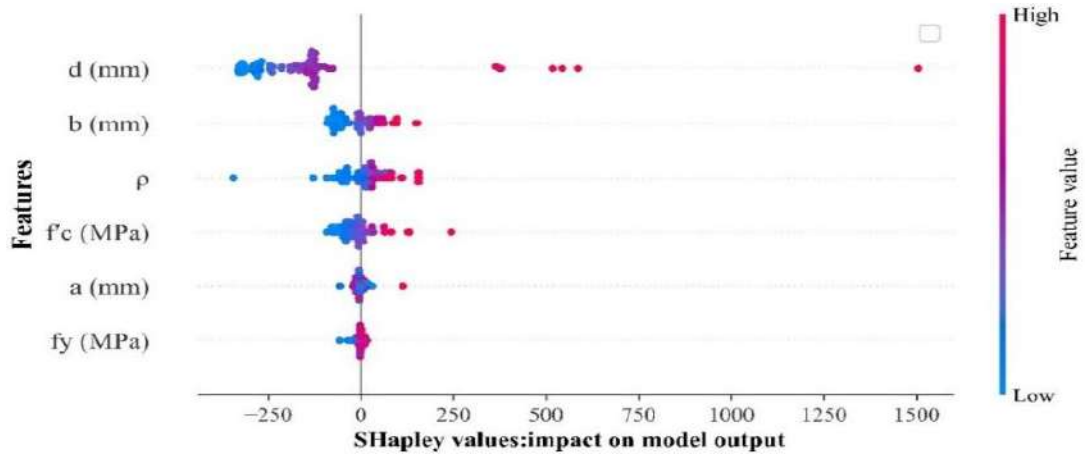


Fig. 21. The SHapley values of the XGB model

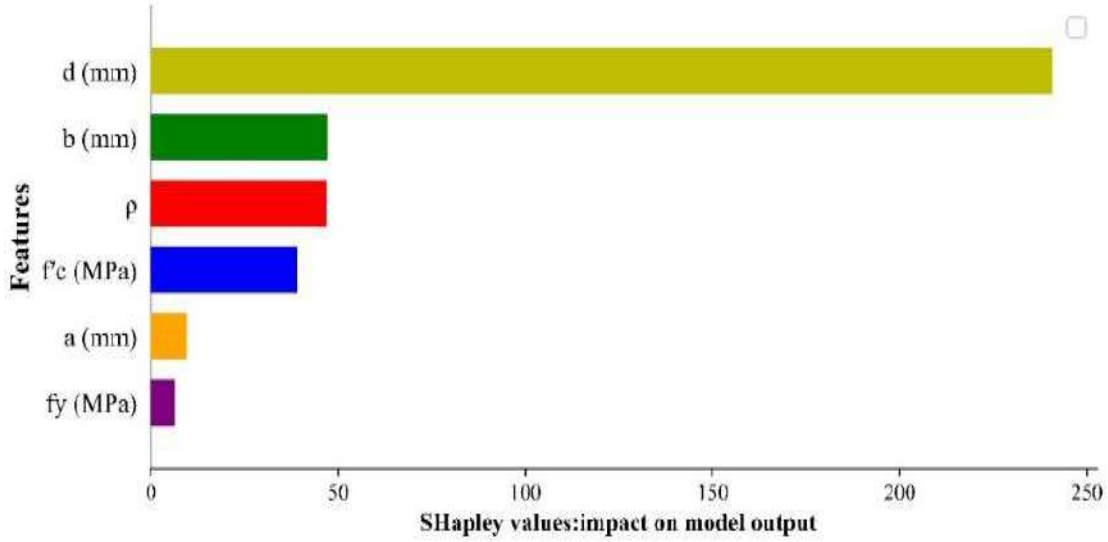


Fig. 22. The mean SHapley values of the XGB model

In ACI 318-14 (American Concrete Institute, 2014), the following Eq. (8) is used to find the punching shear strength, where b_0 (mm), d (mm), f_c' (MPa) have an effect.

$$V_n = \frac{1}{3}\sqrt{f_c'}\, b_0 d \tag{8}$$

Based on ACI 318-19 (American Concrete Institute, 2019), Eq. (9) is an updated form of Eq. (8) with three other parameters called α_s (a constant dependent on supporting column location and its value is 40 for interior columns, 30 for edge columns, and 20 for corner columns; since this study considers only interior columns, α_s = 40), β (The ratio of the long side to the short side of the column, concentrated load, or reaction area) and λ_s (which follows: $\lambda_s = \sqrt{2/(1+0.004d)} \leq 1$).

$$V_n = \min\left[\frac{1}{3}\cdot\frac{1}{6}\left(1+\frac{2}{\beta}\right)\cdot\frac{1}{12}\left(2+\frac{\alpha_s d}{b_0}\right)\right]\lambda_s\sqrt{f_c'}\, b_0 d \tag{9}$$

where V_n: is the nominal punching shear strength, λ_s : is the size effect factor, f_c': is the concrete compressive strength, b_0: is the critical perimeter around the column, and d: is the effective depth of the slab. The parameter α_s : is a constant related to the column location, and β: is the ratio of the long side to the short side of the column, concentrated load area, or reaction area. The operator min: indicates that the minimum value among the bracketed expressions is used in the calculation.

ACI 440.1R-06 (American Concrete Institute, 2006), using the variable k in Eq. (10), also enters the ratio of the modulus of

elasticity of steel to concrete:

$$V_n = 0.8\sqrt{f_c'}\, k b_0 d \quad (10)$$

where k: is $\sqrt{(n\rho)^2 + 2n\rho} - n\rho$. $n = E_s/E_c$. In Eq. (11), the Eurocode 2 (European Committee for Standardization, 2004) uses f_{ck} instead of f_c' as the indicator of characteristic cylinder strength ($f_{ck} = f_c' - 1.6$).

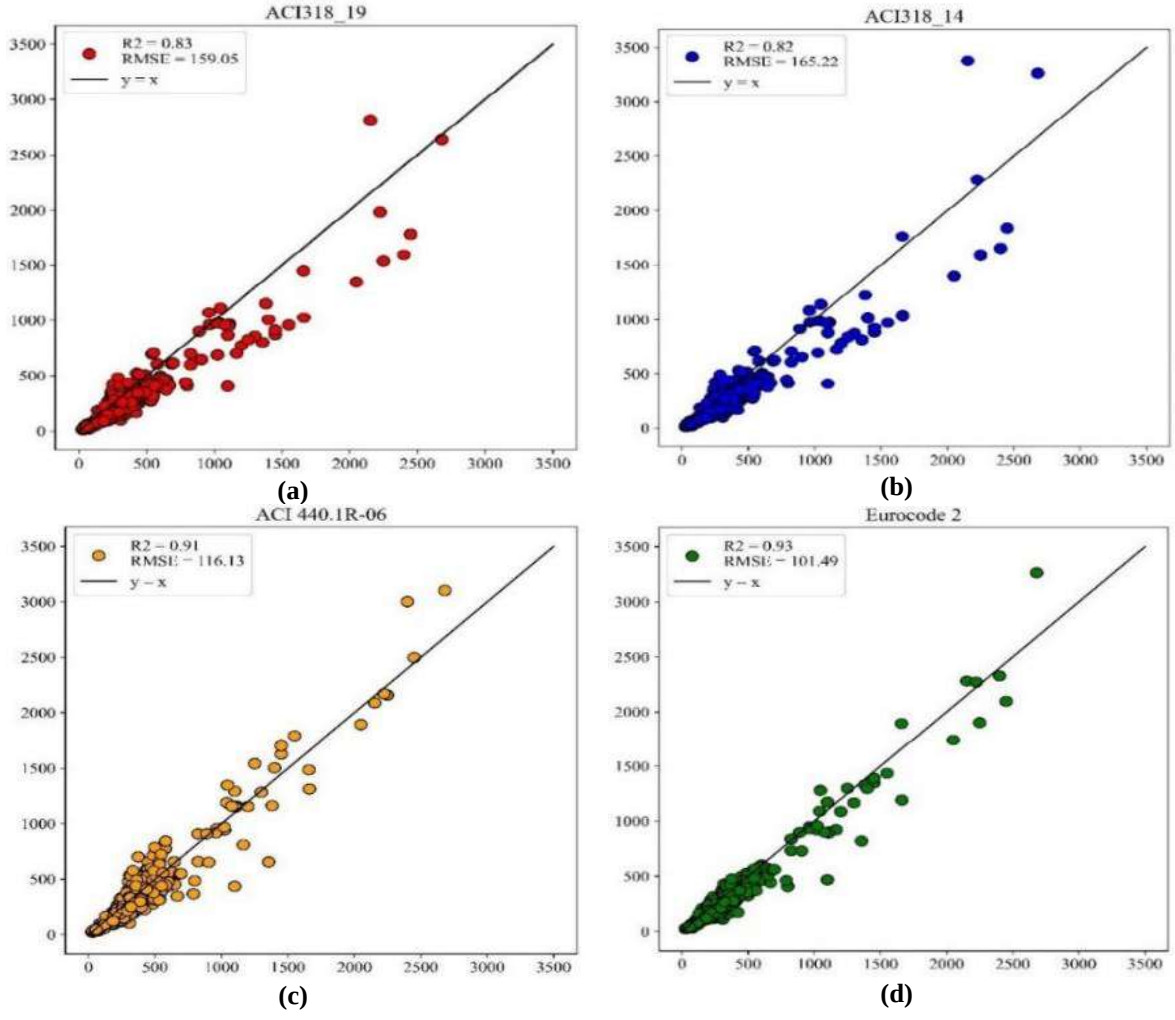
$$V_n = 0.18\,\xi\,\sqrt[3]{100\rho f_{ck}}\; b_0 d \quad (11)$$

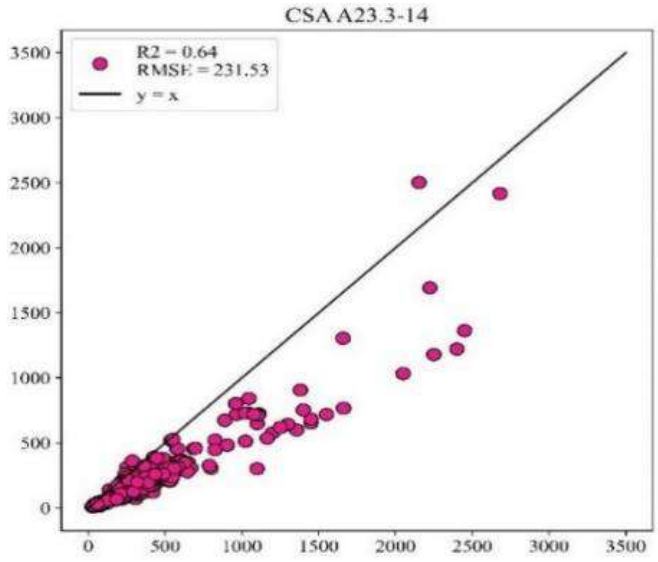
where V_n: is the nominal punching shear strength, ξ: is the size effect factor, ρ: is the reinforcement ratio, f_{ck}: is the characteristic cylinder compressive strength of concrete, b_0: is the critical perimeter around the column, and d: is the effective depth of the slab. The term $\sqrt[3]{100\rho f_{ck}}$: represents the cube root of the product $100\rho f_{ck}$.

It also introduces a new variable related to geometry, and this variable and its limit is as follows $\xi = \left(1 + \sqrt{\frac{200}{d}}\right) \le 2$. CSA A23.3-14 (Canadian Standards Association, 2014) also uses the normal density of concrete (for normal density concrete, $\lambda = 1$) in relation to punching shear strength in two-way slabs without transverse reinforcement with Eq. (12).

$$V_n = \min\left[0.38\,.\,0.19\left(1 + \frac{2}{\beta}\right).\left(0.19 + \frac{\alpha_s d}{b_0}\right)\right]\lambda\phi_c\sqrt{f_c'}\, b_0 d \quad (12)$$

where φ_c = 0.65 and α_s = 4 for interior columns, 3 for edge columns, and 2 for corner columns; since this study considers only interior columns, α_s = 4. After introducing the above equations, the R^2 score graphs of punching shear strength test values (v test) and punching shear strength results from codes equations (v predicted) are shown in Figure 23.





(e)

Fig. 23. Comparison of the R^2 score graphs of the punching shear strength test values (v test) vs. punching shear strength results from codes equations (v predicted): a) ACI 318-19 (American Concrete Institute, 2019); b) ACI 318-14 (American Concrete Institute, 2014); c) ACI 440.1R-06 (American Concrete Institute, 2006); d) Eurocode 2 (European Committee for Standardization, 2004); and e) CSA A23.3-14 (Canadian Standards Association, 2014)

4. Conclusions

Several experimental and a few ML investigations have been conducted due to the significance of the brittle punching shear failure of RC flat slabs without transverse reinforcement. However, the improvement of ML models, which have a lower cost of computations (time and hardware) than DL models, has not been investigated in the studies of ML models or DL models. Therefore, this study aims to provide improved ML and DL models with grid search for punching shear strength prediction in RC flat slabs without transverse reinforcements to make supplementary design proposals. In addition, GEP is implemeted in this study in order to propose an explicit formula to calculate the punching shear strength of concrete slabs. The following are the conclusions derived from this study:

- This study aimed to optimize the accuracy of ML and DL models by applying a grid search to various algorithm parameters. In this effort, 959 machine-learning models and 162 ANN models were evaluated to identify the most effective model. Due to the limited number of experimental tests available, the methodology of this study utilizing a range of models and refining their hyperparameters with optimization tools-provides a more accurate estimation of the punching shear strength of slabs compared to similar studies.
- Each model has a unique accuracy on the test data and a unique accuracy on the training data, but in ranking the best models, one should pay attention to the accuracy of the test data and rank the models accordingly. This is because the test data did not affect the training of the model, and the model did not see them. Therefore, it can be a good criterion for evaluating the model. In this study, the best model achieved from the test data is the XGB model with an R^2 score of 0.98; therefore, the XGB model is introduced as the best model in this study.
- An explicit formula based on GEP for calculating punching shear strength of concrete slabs is proposed, which its R^2 score on test data obtained 0.95.
- After comparing the punching shear obtained from Eurocode 2 (European Committee for Standardization, 2004), ACI, and Canadian codes with the actual

values of punching shear strength available in the dataset, the best R^2 score achieved in Eurocode 2 (European Committee for Standardization, 2004) was 0.93, followed by ACI 440.1R-06 (American Concrete Institute, 2006) with 0.91, while the best R^2 score of ML and DL models, was achieved XGB accounting for 0.98. Therefore, the values obtained from the codes are far from the actual values of the punching shear strength compared to ML and neural networks models. Therefore, it is suggested that in the new editions of these codes, more statistical calculations should be performed on the actual data of punching shear strength in order to achieve a better formula closer to reality.

5. Declaration

Artificial intelligence tools were used only for minor language editing purposes, including checking grammar and spelling. No AI tools were used for data analysis, model development, result generation, or scientific interpretation in this study.

6. References

Abambres, M. and Lantsoght, E.O.L. (2020). "Neural network-based formula for shear capacity prediction of one-way slabs under concentrated loads", *Engineering Structures*, 211, 110501, https://doi.org/10.1016/j.engstruct.2020.110501

Afzali, M., Hamidia, M. and Safi, M. (2023). "Data-driven strength-based seismic damage index measurement for rc columns using crack image-derived parameters", *Measurement*, 218, 113155, https://doi.org/10.1016/j.measurement.2023.113155.

Al-Bayati, A.F. (2023). "Shear strength of reinforced concrete squat walls", *Civil Engineering Journal*, 9(2), 273-304, https://doi.org/10.28991/CEJ-2023-09-02-03.

Alkhawaldeh, S.M.A. (2024). "Enhancing flat slab design: machine learning and metaheuristic approaches to predict punching shear strength", *Asian Journal of Civil Engineering*, 25(3), 2459-2469, https://doi.org/10.1007/s42107-023-00919-4.

Alotaibi, E., Mostafa, O., Nassif, N., Omar, M. and Arab, M.G. (2021). "Prediction of punching shear capacity for fiber-reinforced concrete slabs using neuro-nomographs constructed by machine learning", *Journal of Structural Engineering*, 147(6), https://doi.org/10.1061/(ASCE)ST.1943-541X.0003041.

American Concrete Institute (ACI). (2006). *ACI 440.1R-06: Guide for the design and construction of structural concrete reinforced with FRP bars*, American Concrete Institute, https://www.iranfrp.ir/wp-content/uploads/2018/12/13.pdf.

American Concrete Institute (ACI). (2014). *Building code requirements for structural concrete and commentary (ACI 318-14)*, American Concrete Institute, https://www.concrete.org/Portals/0/Files/PDF/Previews/318-14_preview1.pdf.

American Concrete Institute (ACI). (2019). *Building code requirements for structural concrete and commentary (ACI 318-19)*, American Concrete Institute, https://mattia.ir/wp-content/uploads/2020/10/ACI-318R-19.pdf.

Aminian, P., Javid, M.R., Asghari, A., Gandomi, A.H. and Esmaeili, M.A. (2011). "A robust predictive model for base shear of steel frame structures using a hybrid genetic programming and simulated annealing method", *Neural Computing and Applications*, 20(8), 1321-1332, https://doi.org/10.1007/s00521-011-0689-0.

Anjali, K., Yesilyurt, S.N., Samui, P., Yildirim Dalkilic, H. and Mert Katipoglu, O. (2023). "Determination of dbtt of functionally graded steels using artificial intelligence", *Civil Engineering Infrastructures Journal*, 57(1), 189-203, https://doi.org/10.22059/ceij.2023.349334.1902

Birkle, G. and Dilger, W.H. (2008). "Influence of slab thickness on punching shear strength", *ACI Structural Journal*, 105(2), https://doi.org/10.14359/19733.

Bypour, M., Mahmoudian, A., Tajik, N., Taleshi, M.M., Mirghaderi, S.R. and Yekrangnia, M. (2024). "Shear capacity assessment of perforated steel plate shear wall based on the combination of verified finite element analysis, machine learning, and gene expression programming", *Asian Journal of Civil Engineering*, 25(7), 5317-5333, https://doi.org/10.1007/s42107-024-01115-8.

Canadian Standards Aassociation (CSA). (2014). *Design of concrete structures (CSA A23.3-14)*, *In Canadian Standards Association*, https://books.google.com/books/about/a23_3_14_design_of_concrete_structures.html?id=bopfzqeacaaj.

Canadian Standards Association (CSA). (2019). *CSA A23.3-19: Design of concrete structures.* Mississauga, Ontario, Canada.

Chen, T. and Guestrin, C. (2016). "XGBoost: A scalable tree boosting system", *Proceedings of*

the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794, https://doi.org/10.1145/2939672.2939785.

Elstner, R.C. and Hognestad, E. (1956). "Shearing strength of reinforced concrete slabs", *ACI Journal Proceedings*, 53(7), https://doi.org/10.14359/11501.

European Committee for Standardization (CEN). (2004). *EN 1992-1-1: Euro code 2- design of concrete structures-part 1.1: general rules and rules for buildings*, European Committee for Standardization, https://eurocodes.jrc.ec.europa.eu/EN-Eurocodes/eurocode-2-design-concrete-structures?page=1.

Ferreira, C. (2001). "Gene expression programming: a new adaptive algorithm for solving problems", *Complex Systems*, 13(2), 87-129, https://doi.org/10.48550/arXiv.cs/0102027.

Gandomi, A.H., Faramarzifar, A., Rezaee, P.G., Asghari, A. and Talatahari, S. (2015). "New design equations for elastic modulus of concrete using multi expression programming", *Journal of Civil Engineering and Management*, 21(6), 761-774, https://doi.org/10.3846/13923730.2014.893910

Guandalini, S., Burdet, O.L. and Muttoni, A. (2009). "Punching tests of slabs with low reinforcement ratios", *ACI Structural Journal*, 106(1), 87-95, https://doi.org/10.14359/56287.

Habib, A. and Yildirim, U. (2022). "Estimating mechanical and dynamic properties of rubberized concrete using machine learning techniques: A comprehensive study", *Engineering Computations*, 39(8), 3129-3178, https://doi.org/10.1108/EC-09-2021-0527.

Habib, M., Bashir, B., Alsalman, A. and Bachir, H. (2023). "Evaluating the accuracy and effectiveness of machine learning methods for rapidly determining the safety factor of road embankments", *Multidiscipline Modeling in Materials and Structures*, 19(5), 966-983, https://doi.org/10.1108/MMMS-12-2022-0290.

Hamidia, M., Kaboodkhani, M. and Bayesteh, H. (2024). "Vision-oriented machine learning-assisted seismic energy dissipation estimation for damaged RC beam-column connections", *Engineering Structures*, 301, 117345, https://doi.org/10.1016/j.engstruct.2023.117345

Jamshidian, S. and Hamidia, M. (2023). "Post-earthquake damage assessment for RC columns using crack image complexity measures", *Bulletin of Earthquake Engineering*, 21(13), 6029-6063, https://doi.org/10.1007/s10518-023-01745-4.

Jiang, X. and Xu, C. (2022). "Deep learning and machine learning with grid search to predict later occurrence of breast cancer metastasis using clinical data", *Journal of Clinical Medicine*, 11(19), 5772, https://doi.org/10.3390/jcm11195772.

Li, B. and Li, X. (2023). "Study on the test error of silt dynamic characteristic and its influence on the peak ground acceleration", *High Tech and Innovation Journal*, 4(1), 65-74, https://doi.org/10.28991/HIJ-2023-04-01-05.

Liang, S., Shen, Y., Gao, X., Cai, Y. and Fei, Z. (2023). "Symbolic machine learning improved MCFT model for punching shear resistance of FRP-reinforced concrete slabs", *Journal of Building Engineering*, 69, 106257, https://doi.org/10.1016/j.jobe.2023.106257.

Liu, T., Cakiroglu, C., Islam, K., Wang, Z. and Nehdi, M.L. (2024). "Explainable machine learning model for predicting punching shear strength of FRC flat slabs", *Engineering Structures*, 301, 117276, https://doi.org/10.1016/j.engstruct.2023.117276

Lundberg, S. and Lee, S.I. (2017). "A unified approach to interpreting model predictions", https://doi.org/10.48550/arXiv.1705.07874.

Mangalathu, S., Shin, H., Choi, E. and Jeon, J.S. (2021). "Explainable machine learning models for punching shear strength estimation of flat slabs without transverse reinforcement", *Journal of Building Engineering*, *39*, 102300, https://doi.org/10.1016/j.jobe.2021.102300.

Mansouri, I., Güneyisi, E.M. and Mosalam, K.M. (2021). "Improved shear strength model for exterior reinforced concrete beam-column joints using gene expression programming", *Engineering Structures*, 228, 111563, https://doi.org/10.1016/j.engstruct.2020.111563

Marmarchinia, S., Ghiami Azad, A.R., Mirghaderi, S.R. and Karney, B. (2025). "A novel modular system for pipe racks using slip-friction connections", *Journal of Constructional Steel Research*, 226, 109286, https://doi.org/10.1016/j.jcsr.2024.109286.

Mehrzad, A., Darmiani, M., Mousavi, Y., Shafie-Khah, M. and Aghamohammadi, M. (2023). "A review on data-driven security assessment of power systems: trends and applications of artificial intelligence", *IEEE Access*, 11, 78671-78685, https://doi.org/10.1109/Access.2023.3299208.

Mirzahosseini, M., Jiao, P., Barri, K., Riding, K.A. and Alavi, A.H. (2019). "New machine learning prediction models for compressive strength of concrete modified with glass cullet", *Engineering Computations*, 36(3), 876-898, https://doi.org/10.1108/EC-08-2018-0348.

Naderpour, H., Abbasi, M., Kontoni, D.P.N., Mirrashid, M., Ezami, N. and Savvides, A.A. (2024). "Integrating image processing and machine learning for the non-destructive assessment of RC beams damage", *Buildings*, 14(1), 214, https://doi.org/10.3390/buildings14010214.

Ozden, S., Ersoy, U. and Ozturan, T. (2006). "Punching shear tests of normal- and high-strength concrete flat plates", *Canadian Journal of Civil Engineering*, 33(11), 1389-1400, https://doi.org/10.1139/l06-089.

Palomino Ojeda, J.M., Cayatopa-Calderón, B.A., Quiñones Huatangari, L., Piedra Tineo, J.L., Milla Pino, M.E. and Rojas Pintado, W. (2023). "Convolutional neural network for predicting failure type in concrete cylinders during compression testing", *Civil Engineering Journal*, 9(9), 2105-2119, https://doi.org/10.28991/CEJ-2023-09-09-01.

Parbat, D. and Chakraborty, M. (2020). "A python based support vector regression model for prediction of COVID19 cases in India", *Chaos, Solitons and Fractals*, 138, 109942, https://doi.org/10.1016/j.chaos.2020.109942.

Patil, S., Patil, A. and Phalle, V.M. (2018). "Life prediction of bearing by using adaboost regressor", *Proceedings of TRIBOINDIA-2018, An International Conference on Tribology*, https://doi.org/10.2139/ssrn.3398399.

Regan, P.E. (1986). "Symmetric punching of reinforced concrete slabs", *Magazine of Concrete Research*, 38(136), 115-128, https://doi.org/10.1680/macr.1986.38.136.115.

Rizk, E., Marzouk, H. and Hussein, A. (2011). "Punching shear of thick plates with and without shear reinforcement", *ACI Structural Journal*, 108(5), 581-591, https://doi.org/10.14359/51683215.

Soleimani-Babakamali, M.H. and Zaker Esteghamati, M. (2022). "Estimating seismic demand models of a building inventory from nonlinear static analysis using deep learning methods", *Engineering Structures*, 266, 114576, https://doi.org/10.1016/j.engstruct.2022.114576

Standards Australia. (2011). "Reinforced concrete design: in accordance with AS 3600-2009", Sydney, N.S.W.: Cement and concrete aggregates Australia, Standards Australia, https://nla.gov.au/nla.cat-vn5779703.

Tajik, N., Mahmoudian, A., Mohammadzadeh Taleshi, M. and Yekrangnia, M. (2024). "Explainable XGBoost machine learning model for prediction of ultimate load and free end slip of GFRP rod glued-in timber joints through a pull-out test under various harsh environmental conditions", *Asian Journal of Civil Engineering*, 25, 141-157, https://doi.org/10.1007/s42107-023-00764-5.

Taleshi, M.M., Tajik, N., Mahmoudian, A. and Yekrangnia, M. (2024). "Prediction of pull-out behavior of timber glued-in glass fiber reinforced polymer and steel rods under various environmental conditions based on ANN and GEP models", *Case Studies in Construction Materials*, 20, e02842, https://doi.org/10.1016/j.cscm.2023.e02842.

Tavasoli, M. (2023). *Web-based intelligent packaging evaluation (WIPE) platform*, M.Sc. Theses, Michigan State University, https://doi.org/doi:10.25335/6xt3-3x58.

Theodorakopoulos, D.D. and Swamy, R.N. (2002). "Ultimate punching shear strength analysis of slab-column connections", *Cement and Concrete Composites*, 24(6), 509-521, https://doi.org/10.1016/S0958-9465(01)00067-1.

Tran, V.L. and Kim, S.E. (2021). "A practical ANN model for predicting the PSS of two-way reinforced concrete slabs", *Engineering with Computers*, 37(3), 2303-2327, https://doi.org/10.1007/s00366-020-00944-w.

Zaker Esteghamati, M. (2024). "Leveraging machine learning techniques to support a holistic performance-based seismic design of civil structures", *Interpretable Machine Learning for the Analysis, Design, Assessment, and Informed Decision Making for Civil Infrastructure*, 25-49, https://doi.org/10.1016/B978-0-12-824073-1.00008-3.

Zaker Esteghamati, M. and Huang, Q. (2023). "Evaluating the impact of higher-mode and inelastic dynamic responses of concrete frames on the performance of seismic intensity measures", *Structures*, 56, 105029, https://doi.org/10.1016/j.istruc.2023.105029.