IBRACON Structures and Materials Journal
Revista IBRACON de Estruturas e Materiais
ISSN 1983-4195 ismj.org

ORIGINAL ARTICLE

Machine learning models to predict the punching shear strength of reinforced concrete flat slabs

Modelos de aprendizado de máquina para previsão da resistência à punção de lajes lisas de concreto armado

Francisco Eudázio Suriano da Silva Júnior[a]
Wellison José de Santana Gomes[a]

[a]Universidade Federal de Santa Catarina – UFSC, Department of Civil Engineering, Florianópolis, SC, Brasil

Received 21 May 2022
Accepted 10 October 2022

Abstract: Punching shear failure is caused by shear stress concentration in the slab-column connection of reinforced concrete flat slabs. As it is a brittle failure, it is crucial to understand how this mechanism works and to correctly predict the resistance of slabs subjected to it. In this paper, machine learning-based models were developed and compared to predict the punching shear resistance of reinforced concrete interior slabs without shear reinforcement. The models were based on 373 experimental data of interior slabs. Artificial neural network, decision tree, random forest and extreme gradient boosting algorithms were employed. The input variables considered herein were the effective depth of the slabs, flexural reinforcement ratio, effective width of the columns, concrete compressive strength and steel yield strength, and the target variable was the punching shear strength. The results for the punching shear resistances obtained by the developed models, as well as those obtained by employing models presented in five reinforced concrete design codes, were compared to the experimental data. All machine learning models showed coefficient of determination above 0.95 for test data. As for the design code models, large discrepancies were observed between them, with the Brazilian code showing more accuracy than the others in predicting the failure load of the slabs.

Keywords: predictive models, structural safety, design provisions, computational modeling, XGBoost.

Resumo: A ruptura por punção é causada pela concentração de tensões de cisalhamento na ligação laje-pilar de lajes lisas de concreto armado. Por se tratar de uma ruptura frágil, é fundamental entender como esse mecanismo funciona e prever corretamente a resistência das lajes submetidas a ele. Neste artigo, modelos baseados em aprendizado de máquina foram desenvolvidos e comparados para prever a resistência à punção de lajes internas de concreto armado sem armadura de cisalhamento. Os modelos foram baseados em 373 resultados experimentais de lajes apoiadas sobre pilar intermediário. Os algoritmos de rede neural artificial, árvore de decisão, floresta aleatória e *extreme gradient boosting* foram empregados. As variáveis de entrada aqui consideradas foram a altura útil das lajes, taxa de armadura de flexão, largura efetiva dos pilares, resistência à compressão do concreto e tensão de escoamento do aço, e a variável alvo foi a resistência à punção. Os resultados das resistências ao cisalhamento obtidos pelos modelos desenvolvidos, bem como aqueles obtidos pelo emprego de modelos apresentados em cinco códigos de projeto de concreto armado, foram comparados com os dados experimentais. Todos os modelos de aprendizado de máquina apresentaram coeficiente de determinação acima de 0,95 para dados de teste. Quanto aos modelos de normas de projeto, foram observadas grandes discrepâncias entre eles, com a norma brasileira apresentando maior precisão que as demais na previsão da carga de ruptura das lajes.

Palavras-chave: modelos preditivos, segurança estrutural, disposições de projeto, modelagem computacional, XGBoost.

How to cite: F. E. S. Silva Júnior and W. J. S. Gomes, "Machine learning models to predict the punching shear strength of reinforced concrete flat slabs", *Rev. IBRACON Estrut. Mater.*, vol. 16, no. 4, e16405, 2023, https://doi.org/10.1590/S1983-41952023000400005

Corresponding author: Francisco Eudázio Suriano da Silva Júnior. E-mail: jrsuri@outlook.com
Financial support: The authors would like to thank CAPES for the financial support during this research.
Conflict of interest: Nothing to declare.
Data Availability: The data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/jrsuri/punching-shear-of-flat-concrete-slabs and the original version is available in datacenterhub at https://datacenterhub.org/dataviewer/view/neesdatabases:db/aci_445_punching_shear_collected_databank/.

1 INTRODUCTION

Flat slabs are systems where the slabs are supported directly on the columns. Although simple in appearance, the flat slab system presents a complex behavior, especially in the slab-column connection, since punching failure develops in a brittle manner and with limited deflections, and is followed by an almost complete loss of the load carrying capacity [1].

According to Muttoni [2], the design of flat slabs is mostly governed by serviceability conditions (with relatively large deflections in service) and by the ultimate limit state of punching shear (also called two-way shear). The design standards for concrete structures present models to try to predict the shear resistance of flat slabs. However, several studies in the literature suggest that the standards differ significantly in predicting punching shear strength.

In the paper by Sanabria et al. [3] two slabs were studied and it was found that the results obtained using the formulations presented in the Brazilian, European and North American standards varied by up to 70% for the same slab, with most of the results being far from the experimental data. Avagyan [4] studied the punching shear resistance of four slabs with shear reinforcement according to the European, North American, Armenian and Russian design models, and found that the strength of the slabs varied by up to 35% between the design provisions. Silva et al. [5] observed significant differences between the North American, Brazilian and European standards regarding the prediction of the punching shear strength of four flat slabs, with results varying by up to 75% for the same slab. Issa and Ismail [6] studied the punching shear strength of 257 slabs using the European, North American, British, Egyptian, Japanese and European design formulas. The authors observed large discrepancies between the models, and the European standards were the most accurate in predicting the punching shear resistance of the slabs, while the North American and Japanese were the less accurate.

Experimental tests and numerical simulations can be used to better understand the mechanical behavior of structural elements. As a complement of these two approaches, models based on machine learning (ML) are increasingly employed. In Xu and Saleh [7] the state of the art and trends for the use of ML in structural engineering problems are presented. The authors stated that the areas of structural reliability and safety will, in the near future, be profoundly changed by the incorporation of ML. According to the authors, this is being made possible in part by the advent of big data, the collection and storage of large amounts of data and the development of powerful algorithms to probe it.

By using self-learning features, ML extracts the complicated relationship between input and output data and then uses this relationship to make predictions, without being explicitly programmed to do so [8]. ML is actually a “black box” that maps the relationship between inputs and outputs, so it can be used for classification and regression, without complicated mathematical derivations. It usually has better performance compared to traditional models, making it a good complement to traditional structural mechanics approaches [9]. As disadvantages, ML-based models require many observational data to be built and are dependent on the quality and distribution of the data. Furthermore, they can be slow to train and they usually lack the extrapolation capacity and the physical meaning or interpretability that traditional numerical models usually have.

Some recent studies that employed ML in structural engineering problems can be highlighted. Nguyen et al. [10] used the extreme gradient boosting algorithm (XGBoost or XGB) to predict the punching shear strength of interior slabs, using data from 497 slabs available in the literature. The XGB model was compared to models based in artificial neural networks (ANN) and random forest (RF), as well as design code models, showing superior accuracy and with coefficient of determination (R^2) around 0.96 for test data. It was also observed that the effective depth of the slab was the variable with the greatest impact on the resistance.

Mangalathu et al. [11] used 380 experimental results and the models were based on linear regression techniques and ML methods. These models were compared with each other and with design models. The XGB-based model was the most accurate in predicting punching shear, with R^2 around 0.85 for test data. However, in conflict with the conclusions of Nguyen et al. [10], it was observed that the material properties had more influence on the punching shear strength than the geometric properties of the slab.

Lu et al. [12] studied some approaches for the feature selection step to determine the best way to estimate the punching shear strength of RC slabs reinforced with SFRC (steel fibers). For this, the authors used 140 experimental results to build models based on trees, obtaining models with R^2 between 0.95 and 0.98. Ly et al. [13] developed two hybrid models that combined ANN and optimization techniques to predict the shear strength of reinforced concrete beams reinforced with SFRC, where 463 experimental results were used and models with R^2 of the order of 0.95 were obtained. Gomes [14] used a structural reliability approach to compare shallow and deep ANNs in the solution of four reliability problems, showing that while both types of ANN produce good results, deep networks usually outperform shallow ones. Feng et al. [9] used ensemble learning and developed a model capable of predicting the resistance and failure mode of circular columns subjected to cyclic loading, which is both a regression and classification problem. In their paper, 254 experimental results were used and the model based on the adaptive boosting (AdaBoost) algorithm was the one with the highest R^2 (0.98).

The results were compared to design code predictions, showing superior accuracy. In the literature, other papers regarding application of ML to structural problems are available, including [15]–[20].

Despite usually providing more accurate models than those present in design codes, the use of ML in structural engineering problems is still in a maturing process and needs to be further investigated, which is one of the reasons for this study. Another reason is that, as already mentioned, some references in the literature show that the design standards for punching shear in RC slabs have large discrepancies between them. Thus, in the present paper, the ABNT NBR 6118 [21], ACI 318 [22], Eurocode 2 [23], BS 8110 [24] and DIN 1045-1 [25] design models were compared. In particular, no comparisons involving ML models for punching shear and the NBR 6118 [21] model were found in the literature so far, so that the present paper expands the discussion about previously studied models and includes the model presented in the Brazilian standard.

The present paper is organized as follows. A summary of the ML algorithms employed in this study is presented in section 2; a brief background of the structural problem is presented along with design provisions in section 3; the experimental dataset used herein is discussed in section 4, along with data cleaning and preprocessing; the development of the ML models is presented in section 5; the results of the developed ML models are presented in section 6; discussions about design standard models and comparisons are made in section 7; and the conclusions are drawn in section 8.

2 MACHINE LEARNING ALGORITHMS OVERVIEW

Four regression algorithms are used in this research: ANN, decision tree (DT), RF and XGBoost. The basic concepts behind each technique are presented below.

2.1 Artificial Neural Network – ANN

Artificial neural networks emerged from the idea of mathematically modeling the functioning of biological neurons, whose discussions began in the 1940s. In a generic way, neural networks are the implementation of connections between inputs, mathematical functions and outputs in computer code [26]. For Albon [27], neural networks can be visualized as a series of connected layers that form a network that connects features at one end to target values at the other. At the heart of neural networks is the unit (also called a neuron). The unit takes one or more inputs, multiplies each input by a parameter (also called a weight), adds these values to a constant (bias), and feeds them to an activation function. The output is then passed on to the next layers in the network, if any, or presented as the output of the ANN if no more deeper layers are present.

Aggarwal [28] explains that the most basic architecture of a neural network is known as perceptron. The perceptron contains two layers of nodes (neurons), one corresponding to inputs and one to output, with a single node in the latter. The number of nodes in the input layer depends on the dimensionality of the problem. Each input node is connected to the output and for each connection a weight is assigned. Just as learning in biological systems is done by modifying the intensity of synapses, learning in the perceptron is done by adjusting the weights of connections between inputs and outputs whenever an incorrect prediction is made. The function employed by the perceptron at each node is called an activation function. Arbitrary functions like logistic, sigmoid, hyperbolic tangent or rectified linear unit (ReLU) can be used.

The perceptron algorithm cycles through all the training samples and iteratively adjusts the weights until convergence is reached. Each cycle is called an epoch. During the learning process, a loss function (or cost function) is employed to evaluate how poorly the model is performing at each epoch. Mohri et al. [29] explain that a loss function measures the difference between a predicted label and a true one. Examples of loss functions are the mean absolute error (MAE) and the root mean squared error (RMSE). According to Aggarwal [28], the perceptron algorithm starts with a random vector of weights. The algorithm then performs the initial predictions and updates the weight vector using optimization algorithms (such as gradient descent) and a learning rate to minimize the error defined by the loss function.

In practice, one or more hidden layers are used between the inputs and the output. In this case, the network is a multilayer perceptron. Neural networks with many hidden layers are considered deep networks and their application is called deep learning [27]. In networks with hidden layers, the gradient descent algorithm cannot be used to calculate the error gradient for hidden neurons [30]. Consequently, the contribution of each neuron to the overall error of the network must be calculated by using a backpropagation algorithm.

Examples of single and multilayer perceptrons are shown in Figure 1.

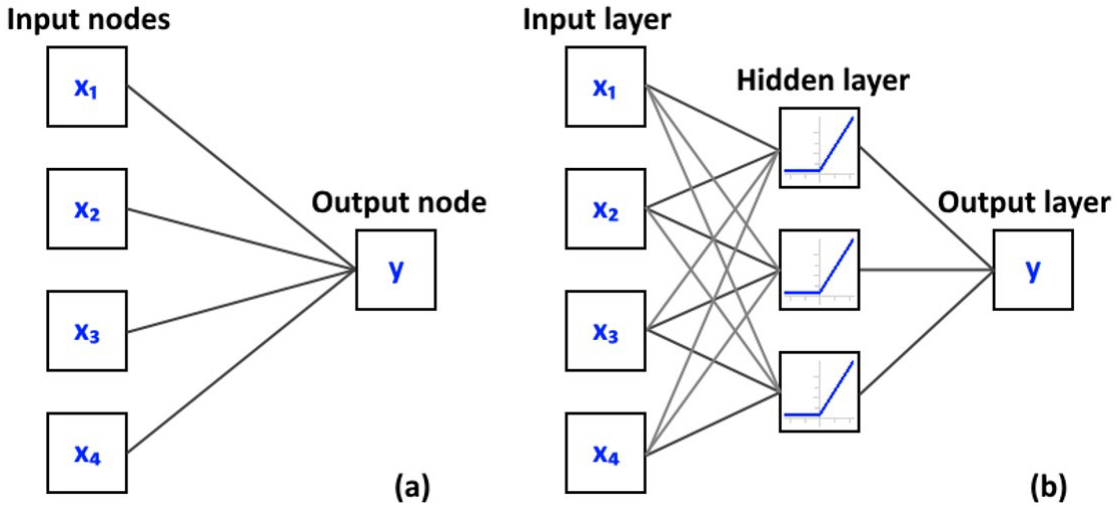

Figure 1. Illustration of single layer (a) and multilayer perceptron (b).

ANNs have been widely used in civil and structural engineering problems, as in [8]–[10], [13], [14] and [31]–[33].

2.2 Decision Tree – DT

Decision trees are models based on a tree-like structure where a series of decisions are connected [27]. One of the reasons for their popularization is that, unlike ANNs, they are more interpretable, intuitive and faster to train.

According to Aggarwal [28], the algorithm for building a decision tree has two types of nodes, called internal and leaf nodes. Each leaf is labeled with the dominant class at that node. A special internal node is the root node that corresponds to the entire feature space. The generic decision tree induction algorithm starts with the complete training dataset at the root node and partitions the data into lower level nodes based on a split criterion. Only nodes that contain a mix of different classes need to be split further. Eventually, the decision tree algorithm stops the tree from growing based on a stopping criterion. A generic example of a decision tree is presented in Figure 2.

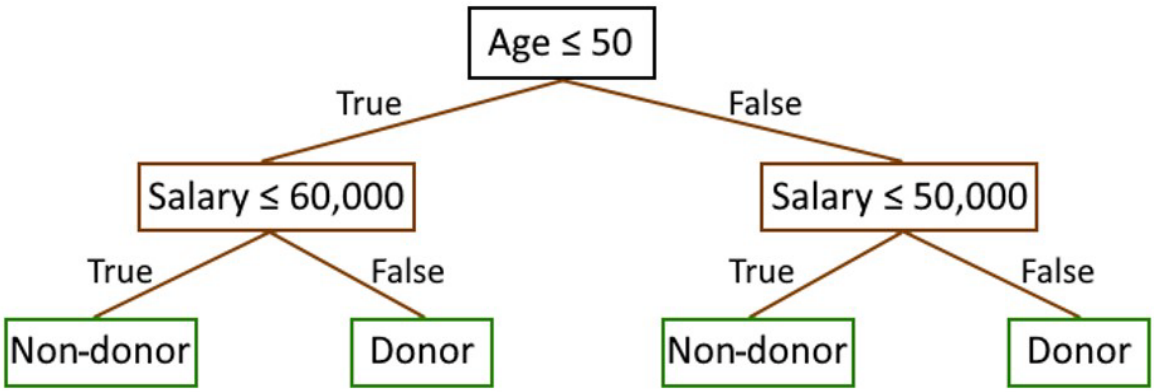

Figure 2. Generic example of a decision tree.

The simplest stopping criterion is one where all training examples in the leaf belong to the same class. One problem is that building the decision tree at this level may lead to overfitting, in which the model fits the random noise of the training data. In this case, this tree will have high variance and will not generalize well to unknown data. To try to reduce the loss of precision associated with overfitting, the classifier uses a pruning mechanism to remove nodes.

In general, simpler models (shallow decision trees) are preferable to more complex models (deep decision trees) if they produce the same error in the training data. Thus, a common stopping criterion is the use of a maximum number of levels (or maximum depth) that the model can reach.

Decision trees can be used for regression or classification and have found applications in different areas of structural engineering, as in [9], [11] and [12]. However, in recent studies, researchers tend to adopt enhanced versions of the classical decision tree algorithm, like random forests and boosting techniques.

2.3 Random Forest – RF

The random forest algorithm emerged as an evolution of decision trees. In general, the data split procedure in decision trees tends to lead to overfitting, when the model performs well with the training data but does not generalize as well to unknown data. The random forest tries to fix this problem by generating several decision trees (estimators), and was developed by Breiman [34].

Random forests are defined as a set of decision trees where randomness is explicitly inserted in the process of splitting the nodes of each tree. The idea is to use this randomness to generate less correlation between the components of the set, allowing each tree to specialize in a slightly different way from the others [28]. The result is the average of the individual results of each tree.

Bootstrapping is one of the sources of randomness in RF. According to Aggarwal [28], in the bootstrap method, the labeled data is sampled randomly and uniformly, with replacement, to create a bootstrapped training dataset that might possibly contain duplicates. The labeled data of size n can be sampled n times with replacement. This results in a training data with the same size as the original labeled data. However, the bootstrapped dataset typically contains duplicates and also misses some points in the original data. Therefore, each estimator in the random forest is trained using a slightly different dataset.

The results of the RF are usually more accurate than those of the simple decision tree and the model is resistant to noise and outliers, which made RF a popular technique nowadays. Recent structural engineering studies include [9]–[12] and [35].

2.4 Extreme Gradient Boosting – XGBoost

Tree boosting is the construction of decision tree models in sequence, iteratively, where a model is based on the results of the previous one. According to Aggarwal [28], the basic idea is to focus on the errors (residuals) and try to adjust the weights of each incorrectly classified instance to improve the next model. In gradient boosting, as the name suggests, the adjustment of the weights is performed through the gradient of the prediction error.

Extreme gradient boosting is a recent technique, developed in 2014 and published by Chen and Guestrin [36] as an evolution of traditional gradient boosting. It was developed to not only have high accuracy, but also low risk of overfitting. According to Chen and Guestrin [36], simple gradient boosting contains three main elements: a loss function to be optimized, a classification and regression tree (CART) for making predictions and a model for adding trees in sequence to minimize the loss function. As an improvement of this algorithm, XGBoost adds regularization to the objective function to avoid overfitting, along with many other enhancements. Russell and Norvig [37] also point out that XGBoost aims at being efficient, carefully organizing memory to avoid cache issues and allowing parallel computing across multiple machines, making it an easily scalable algorithm.

Chen and Guestrin [36] point that the tree ensemble model with regularization minimizes the objective function shown in Equation 1, where i is the number of inputs, N is the number of predictions and K is the number of trees. The function l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. The term Ω penalizes the complexity of the model, where T is the number of leaves and γ is the minimum loss reduction required to make a further partition on a leaf node of the tree. The additional L2 regularization term λ helps to smooth the final learnt weights to avoid overfitting. When λ is set to 0, the objective falls back to the traditional gradient tree boosting.

$$\mathcal{L}(\phi) = \sum_i^N l(\hat{y}_i, y_i) + \sum_k^K \Omega(f_k), \text{ where } \Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2 \quad (1)$$

Besides the regularized objective, two additional techniques are used to further prevent overfitting. The first technique is shrinkage, which scales newly added weights by a factor η after each step of tree boosting. Similar to a learning rate in stochastic optimization, shrinkage reduces the influence of each individual tree and leaves space for future trees to improve the model. The second technique is feature subsampling, which is also used in RF in commercial software such as TreeNet, and helps to prevent overfitting even more [36].

According to Chen and Guestrin [36], the XGBoost uses a gain-based score to search for the best node splits when building a tree. Gain can be seen as the improvement in accuracy brought by a feature to the branches it is on. Equation 2

gives the gain at a leaf node during splitting. Nguyen et al. [10] explain that this formula is composed of four terms, which in turn represent the scores of the new left leaf (g_L), new right leaf (g_R), original left and right leaves (h_L, h_R), and the regularization of the additional leaf. The tree ceases growing when the gain becomes smaller than γ. In the ensemble model, the prediction scores of all trees are summed to obtain the final score.

$$Gain = \frac{1}{2}\left[\frac{g_L^2}{h_L+\lambda} + \frac{g_R^2}{h_R+\lambda} - \frac{(g_L+h_R)^2}{h_L+h_R+\lambda}\right] - \gamma \quad (2)$$

As it is a recent technique, there are still few civil engineering studies applying XGBoost, such as [10], [11] and [38]–[42].

3 PUNCHING SHEAR IN FLAT SLABS AND DESIGN CODE MODELS

The flat slab system of reinforced concrete has been used more frequently because it has some advantages when compared to conventional structural systems. Among these advantages, one can mention greater architectural freedom in defining internal environments or future layout changes; simplification of reinforcement and consequent reduction of labor and material costs; ease in the arrangement of installations and simplification of forms and framing. The system also has disadvantages compared to conventional ones, such as higher levels of vertical displacement of the structure, reduction of the global stability and the possibility of failure by punching shear [43].

Punching shear is a type of shear failure that can occur in plate elements subjected to a concentrated load or reaction applied transversally and is characterized by occurring abruptly, which can lead the structure to ruin through progressive collapse. The shear strength of the slab-column connection is one of the most important parameters in the design of flat slabs [44].

Wight and MacGregor [45] mention that the two-way shear involves a truncated cone or pyramid-shaped surface around the column. According to Muttoni [2], most design codes for punching shear base their verifications on a critical section, with the punching shear strength of slabs without shear reinforcement defined as a function of the concrete compressive strength and often of the reinforcement ratio. Some codes also account for size effect, membrane effect, or the ratio of column size to the depth of the slab. This critical section or control perimeter is defined as shown in Figure 3.

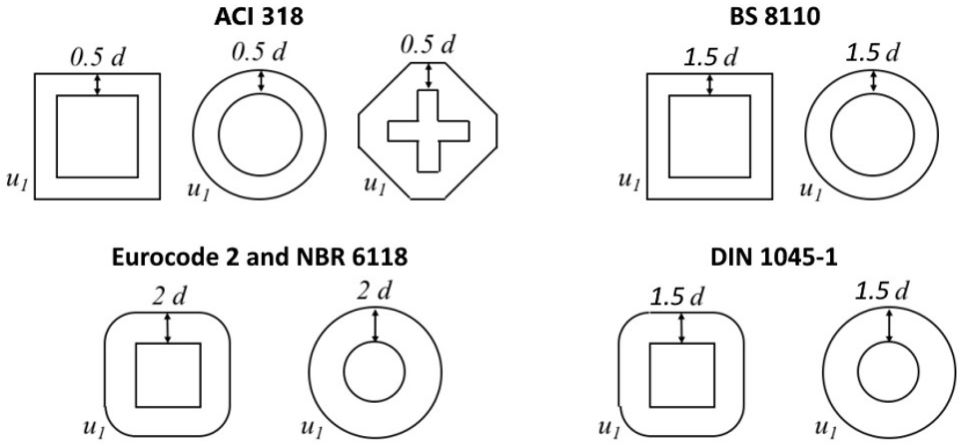

Figure 3. Control perimeters.

The design equations for calculating the punching shear resistance of flat RC slabs without shear reinforcement investigated herein are presented as follows. Note that, to ensure comparison with real failure data, no safety coefficients were introduced in the presented equations. Units are in SI.

3.1 Eurocode 2 (2004)

Eurocode 2 [23] uses Equation 3 for punching shear design, where ρ_l is the longitudinal (flexural) reinforcement ratio, f_c is the compressive strength of the concrete, u_1 is the control perimeter, d is the effective depth of the slab and ξ is the size effect factor, calculated by Equation 4.

$$V_{EC2,p} = 0.18\xi(100\rho_l f_c)^{1/3} u_1 d, \rho_l \leq 2\% \quad (3)$$

$$\xi = 1 + \sqrt{\frac{200}{d}} \leq 2 \quad (4)$$

3.2 ABNT NBR 6118 (2014)

NBR 6118 [21] uses Equations 5 and 6 for design, using the same procedure as Eurocode 2 [23], but without limitations on the reinforcement ratio and size effect factor.

$$V_{NBR,p} = 0.18\xi(100\rho_l f_c)^{1/3} u_1 d \quad (5)$$

$$\xi = 1 + \sqrt{\frac{200}{d}} \quad (6)$$

3.3 ACI 318M (2019)

Equation 7 is used in the design of flat slabs by ACI 318 [22], where α_s is equal to 20 for corner columns, 30 for edge columns and 40 for interior columns, β is the ratio between the largest and smallest sides of the column and λ_s is the modification factor related to the size effect, given by Equation 8.

$$V_{ACI,p} = \lambda_s \sqrt{f_c} u_1 d \min \begin{cases} \frac{1}{3} \\ \frac{1}{6}\left(1 + \frac{2}{\beta}\right) \\ \frac{1}{12}\left(2 + \frac{\alpha_s d}{u_1}\right) \end{cases} \quad (7)$$

$$\lambda_s = \sqrt{\frac{2}{1+0.004d}} \leq 1 \quad (8)$$

3.4 BS 8110 (1987)

The British standard BS 8110 [24] uses Equation 9 for punching shear design. Reinforcement ratio is capped at 3%. For concretes with f_c above 25 MPa, the calculated value must be multiplied according to Equation 10. Equation 11 calculates the size effect factor.

$$V_{BS,p} = 0.79\xi(100\rho_l)^{1/3} u_1 d, \rho_l \leq 3\% \quad (9)$$

$$V_{BS,p} = V_{BS,p}\left(\frac{f_c}{25}\right)^{1/3} \text{ for } f_c \geq 25 \text{ MPa} \quad (10)$$

$$\xi = \sqrt[4]{\frac{400}{d}} \geq 1 \quad (11)$$

3.5 DIN 1045-1 (2008)

Equation 12 is used to calculate the punching shear resistance by the German standard DIN 1045-1 [25]. Equation 4 gives the size effect factor, corresponding to the same formulation as in Eurocode 2 [23].

$$V_{DIN,p} = 0.21\xi(100\rho_l f_c)^{1/3} u_1 d, \rho_l \leq 2\% \quad (12)$$

4 DATA ENGINEERING

The dataset applied in this study is the same used by [11]. This is a database with experimental results of 519 flat slabs tested by several authors since 1938. A preliminary treatment is carried out to reduce the dimensionality of the problem, considering only the most relevant variables that were used in other papers on the subject, such as [10]–[12]. The selected input variables are: average effective depth of the slab (d_{avg}) in X and Y directions, effective column width (b^*), concrete compressive strength (f_c), steel yield strength (f_y) and average flexural reinforcement ratio (ρ_{avg}) in X and Y directions.

In most cases, the average compressive strength at time of slab testing was informed and, thus, this variable could be used directly. However, in some of the experimental tests, only the compressive strength at 28 days was reported. In these cases, strength at the time of slab testing was estimated from the strength at 28 days based on the elapsed time. Specimen details are presented in the original dataset.

In few cases, the longitudinal reinforcement bars of the slabs were arranged with non-uniform spacing along the section, and the reinforcement ratios employed herein had to be adjusted from the reported ones. The reinforcement ratio considered corresponded only to the region that passes through the column.

For the cases of rectangular or circular columns, b^* is the width of the equivalent square area section, given by Equations 13a and 13b [11], where D is the diameter and b_1 and b_2 are the smaller and larger side of the column, respectively. Thus, the number of features is reduced to five and the output variable is the punching shear strength of the slab (P_u).

$$b^* = \frac{\pi D}{4} \text{ (circular section)} \tag{13a}$$

$$b^* = \frac{b_1 + b_2}{2} \text{ (rectangular section)} \tag{13b}$$

As the objective of this study is to predict the punching shear failure load, 84 slabs that did not fail exclusively by this mechanism are excluded. In addition, missing values are identified in f_y for 18 of the samples. These samples are removed from the dataset. Finally, 44 slabs that did not have longitudinal reinforcement in the region of the column are excluded, leaving 373 experimental results. Table 1 shows the descriptive statistics of the dataset after this treatment.

Table 1. Descriptive statistics of the dataset.

	d_{avg} (mm)	ρ_{avg} (‰)	b^* (mm)	f_c (MPa)	f_y (MPa)	P_u (kN)
Mean	110.73	12.98	180.95	32.80	461.63	384.30
Std. dev.	66.50	6.50	97.25	18.62	118.23	458.86
Min.	29.97	3.25	39.90	8.66	250.00	24.00
25%	76.20	8.44	109.96	22.13	359.00	165.00
50%	107.00	11.75	159.59	28.05	462.00	265.00
75%	121.56	15.19	225.00	35.34	530.00	405.00
Max.	668.50	50.10	707.64	118.70	749.00	4915.00

A common step in the data preprocessing for ML models is data scaling. In the input data, the parameters usually have very different magnitudes and units from each other, which can lead the model to incorrectly assign greater importance to variables with higher numerical values. To prevent this problem, data is scaled so that the features are of the same order of magnitude (with values close to zero, usually between 0 and 1 or between -1 and 1). In this study, three scaling methods from the Python scikit-learn (sklearn) library are applied: StandardScaler, Normalizer and MinMaxScaler. According to the documentation of the sklearn library [46], the StandardScaler transforms the data through the z-score technique, setting null mean and unit standard deviation; the Normalizer normalizes the data by rescaling them to unitary norm; and MinMaxScaler scales the data so that the values are always between 0 and 1. For each ML algorithm, the scaling method that obtained the best results is used. Figure 4 shows the pairplots of the scaled variables after applying the StandardScaler.

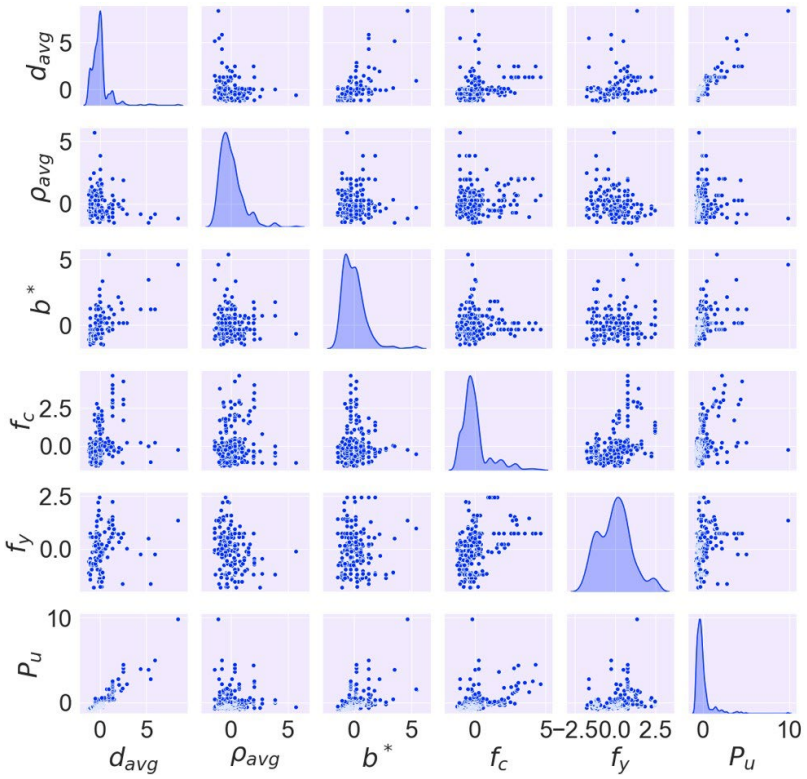

Figure 4. Pairplot.

A pairplot can be interpreted as a symmetric matrix of graphs, where kernel density estimation (KDE) graphs (equivalent to smoothed histograms) are plotted on the main diagonal and, in the other cells, the relationship between each variable and the others is observed through scatter plots. In this way, it is possible to analyze, in a preliminary way, if there is correlation between variables and what type of distribution they follow. Another way to get insights from data is by plotting a heatmap, which corresponds to a representation of data in the form of a diagram in which data values are represented with colors. Heatmap for the features in the dataset (with correlation values) is shown in Figure 5.

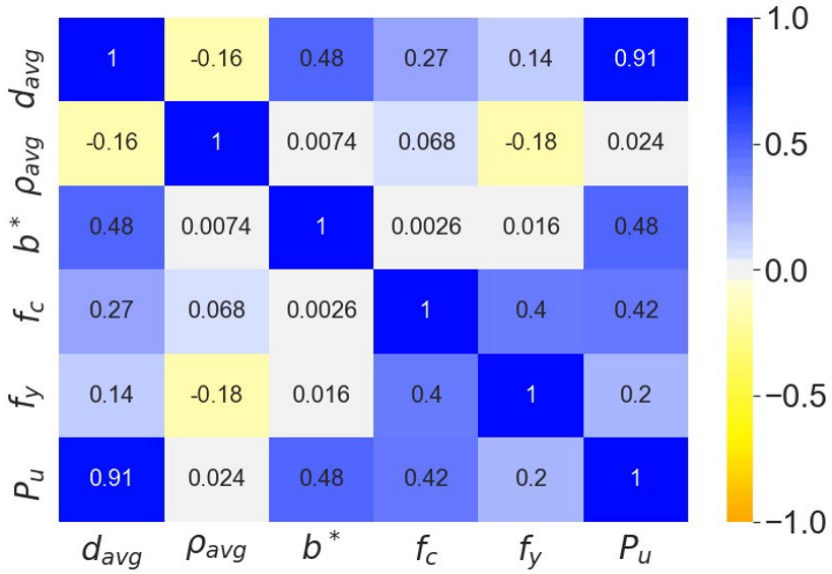

Figure 5. Heatmap.

Through Figures 4 and 5 it is observed that there is a strong correlation between the effective depth and the ultimate load. It is noticed that there is always some degree of positive correlation between individual parameters and the strength of the slabs: when increasing the width of the column, the strength of the concrete, the yield strength of the steel, the effective depth or the flexural reinforcement ratio, the punching shear strength also increases. It is also observed that, apparently, the geometric properties of the slabs (d_{avg} and b^*) show a proportional behavior, which can be explained by the fact that, in experimental tests, larger slabs (with higher widths) are usually supported on larger columns (larger b^*).

Before running the ML algorithms, it is common to split the data into training and test sets. In this way, models are developed based on a portion of the dataset (training set) and tested with the remaining data (test set), to evaluate their accuracy with unknown data and avoid overfitting. For all models of this paper, the data are split in the proportion 70% training and 30% test, which is a common ratio used in the literature. Data scaling is performed only after this split to avoid issues like data leakage.

The repeated holdout technique is used for each algorithm. According to Aggarwal [28], the holdout consists of randomly dividing the data into two disjoint sets, corresponding to the training and test data. The repeated holdout, as the name suggests, repeatedly does this to find the best random state for the training/test split for a given problem. Random state is the seed (or starting point in random number generators), which is a value specified to control randomness and to generate reproducible results.

5 MODEL ENGINEERING

Python libraries sklearn for RF and DT, XGBoost for XGB and Tensorflow/Keras for ANN are used. Hyperparameters must be set to the ML algorithms. Hyperparameters are free parameters that are not determined by the learning algorithm, but rather specified as inputs to it [29]. Table 2 shows the settings used for each algorithm, with initial values based on [10] and [11] and subsequently adjusted via hyperparameter tuning with grid search. Grid search is the process of trying combinations of values in given intervals and seeing which performs best on the validation data [37]. Other hyperparameters take default values from their respective libraries.

Table 2. Hyperparameters used for each ML algorithm.

Model	Parameters
ANN	Number of hidden layers = 2, number of neurons in each hidden layer = 94, activation function = ReLU, optimizer = Adam, learning rate = 0.3, loss function = RMSE, epochs = 50, random state = 0
DT	Maximum depth = 10, random state = 0
RF	Number of estimators = 69, random state = 0
XGB	Number of estimators = 72, learning rate = 0.22, maximum depth = 5, random state = 0

The coefficient of determination (R^2) quantifies the fraction of variability in the series that is explained by the regression [28]. It is therefore desirable for this coefficient to be as close to 1 as possible. It provides an indication of goodness of fit and a measure of how well unseen samples are likely to be predicted by the model. Another metric is the RMSE, which measures the average error between predictions. It is desirable for this metric to be as close to zero as possible. Equations 14 and 15 give the R^2 and RMSE calculations, where Y and $\hat{Y}$ are the actual and predicted values by the model, N_t is the number of samples and $\bar{Y}$ is the mean.

$$R^2 = 1 - \frac{\sum_{i=1}^{N_t}(\hat{Y}_i - Y_i)^2}{\sum_{i=1}^{N_t}(\hat{Y}_i - \bar{Y}_i)^2} \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2} \tag{15}$$

There are several rules in the literature for trying to choose the optimal number of hidden neurons in ANNs. However, most predicting research fields are heuristic in nature, and there is no generally accepted theory to determine how many hidden neurons are needed to approximate any given function [47]. Therefore, the use of systematic experimentation to discover what works best for a specific dataset may be the most general approach. In this paper, the number of hidden neurons is chosen so that R^2 is maximized and RMSE is minimized for test data. Figure 6 indicates that the optimal number of neurons to meet these conditions is 94 for the problem at hand.

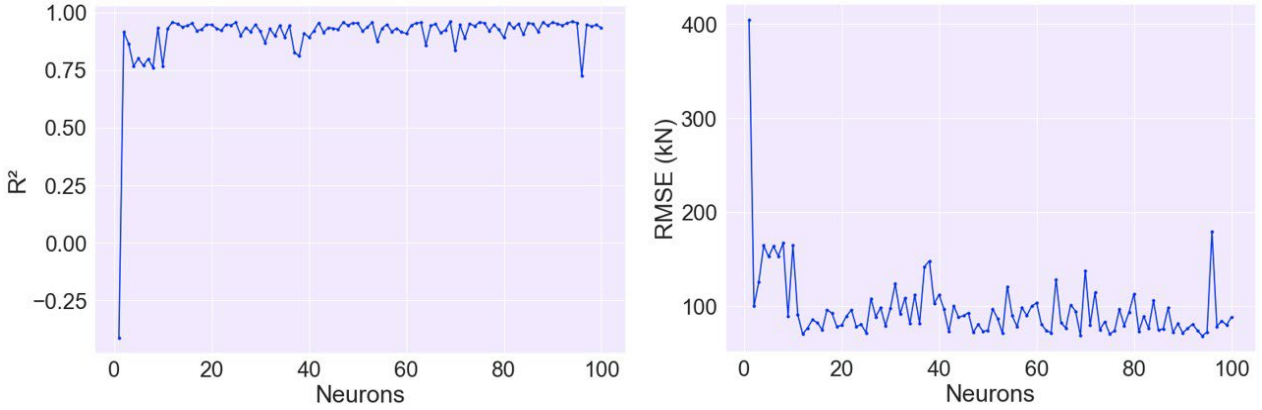

Figure 6. Experimentation of the number of neurons in the ANN.

A similar process is done to find the optimal maximum depth for the DT and number of estimators for the RF and XGB algorithms. Results are shown in Figures 7, 8 and 9, where optimal values are 10 levels for DT, 69 trees for RF and 72 trees for XGB. In Figures 7 and 8, it is observed that the prediction accuracy for test data starts to decrease after certain values of maximum tree depth (DT) and number of estimators (RF), which is a sign of overfitting. Figure 7 also indicates that the maximum possible depth for the DT is 18.

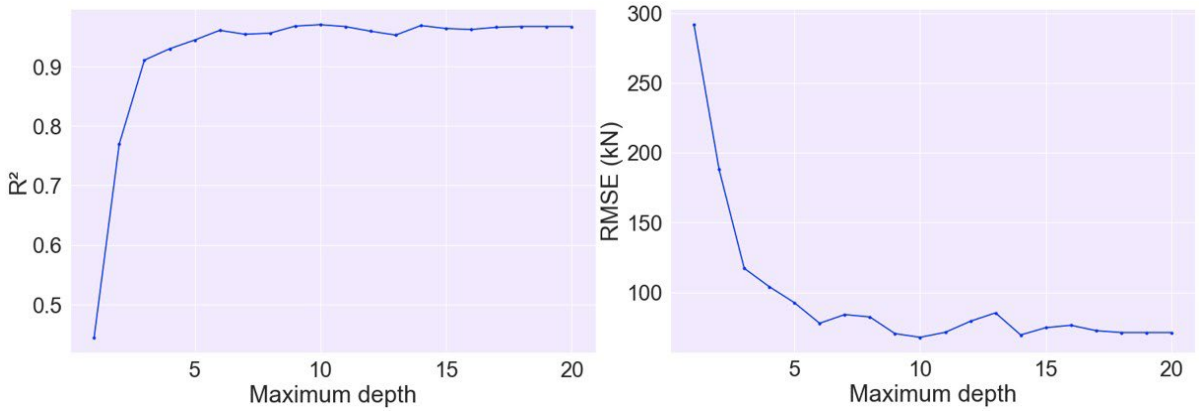

Figure 7. Experimentation of the maximum depth in the DT.

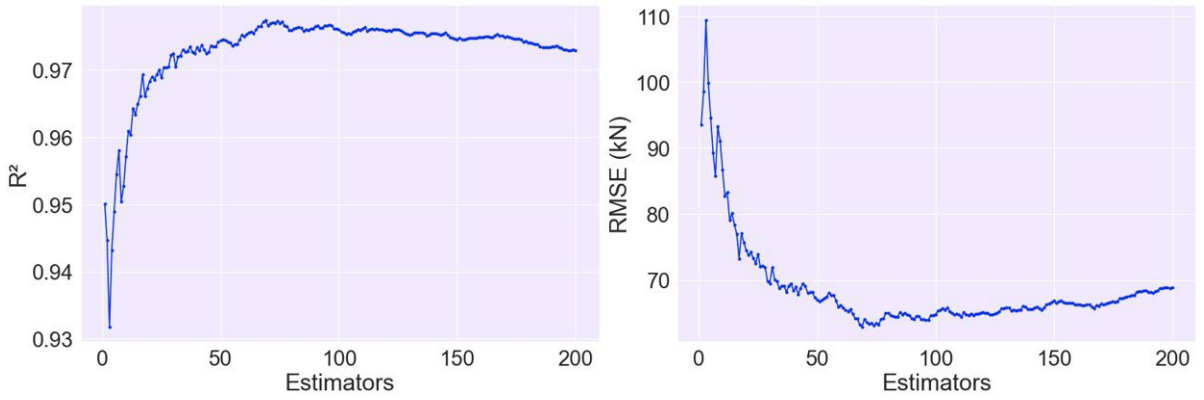

Figure 8. Experimentation of the number of estimators in the RF.

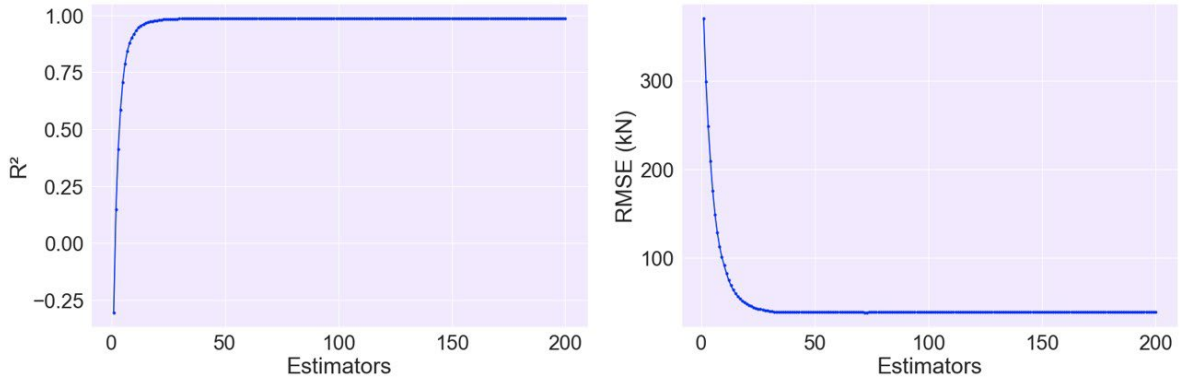

Figure 9. Experimentation of the number of estimators in the XGB.

6 RESULTS AND DISCUSSIONS

Figure 10 shows the results of the regression models (regplots) for training and test data and for the entire dataset. Results are presented from highest to lowest R^2 for test data.

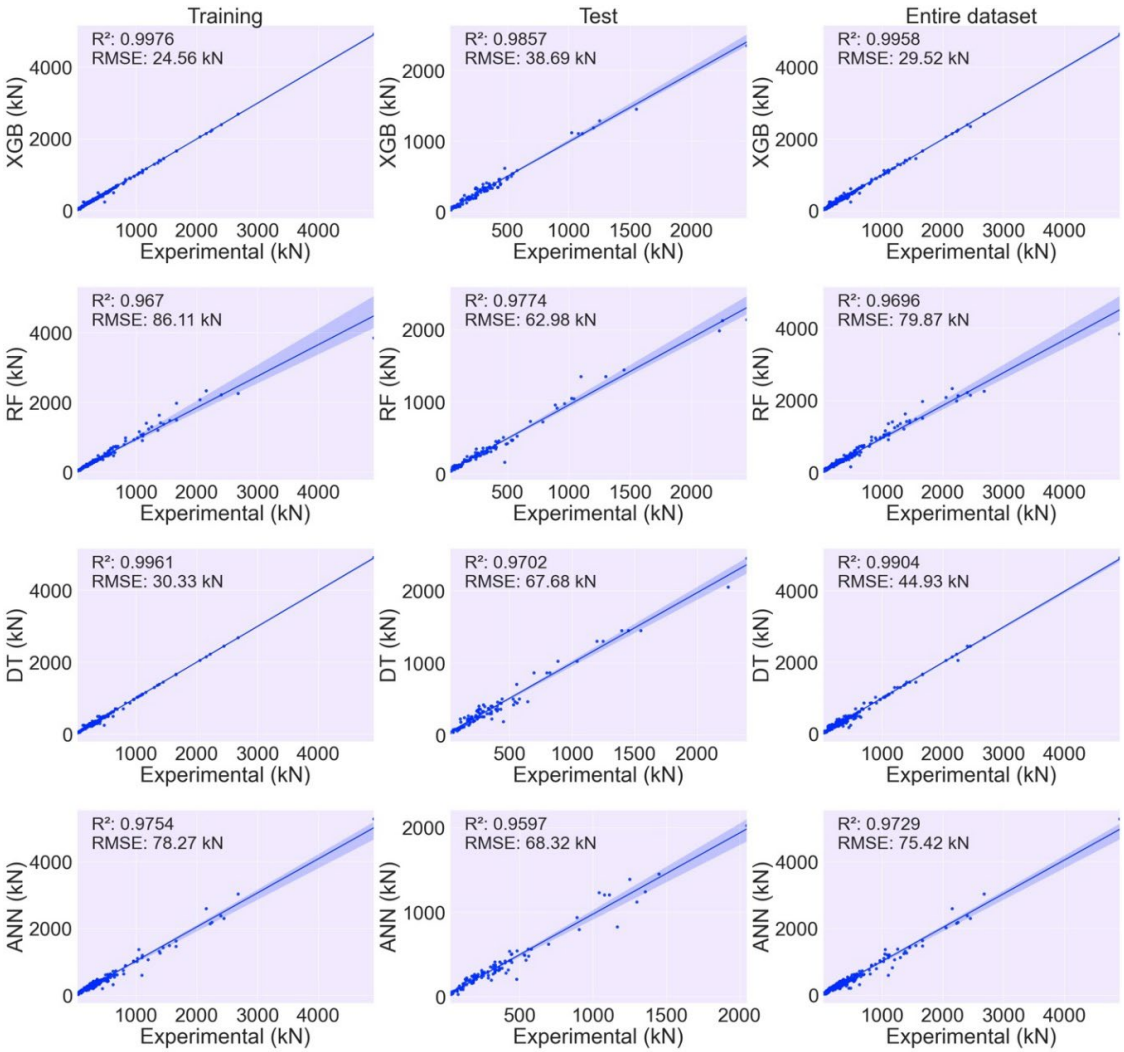

Figure 10. Regplots.

From Figure 10, XGB is the model with highest R^2 (0.9857) and lowest RMSE (38.69 kN) for test data. It is followed by RF and DT. The RF and DT models show similar performance; however, the RF has less variance (less overfitting or higher accuracy for test data). The DT and XGB models also present similar results, but the DT shows a significantly lower R^2 for the test data. The DT not performing as well to test data as it does to training data can be explained by the tendency to overfitting of the classical decision tree models. It is observed that ANN is the model that shows the lowest R^2 (0.9597) and highest RMSE (68.32 kN) for test data, but these results agree with what was found in the literature, as shown in section 1.

Figure 11 shows histograms of the $P_{u,pred}/P_{u,experimental}$ ratio (using the test sets of each model), with all models having values concentrated around 1. From a structural safety point of view, it is desirable that, if the model makes an incorrect prediction, its prediction must be in favor of safety (that is, that it underestimates the structural resistance). Therefore, ratios closer to 1 and lower standard deviations are desirable. The ANN outperforms the DT and RF models when analyzing this relationship, as opposed to what is observed in Figure 10. It is observed that the XGB (Figure 11d) shows the closest to 1 mean and the lowest standard deviation between the four models, and also the lowest maximum value.

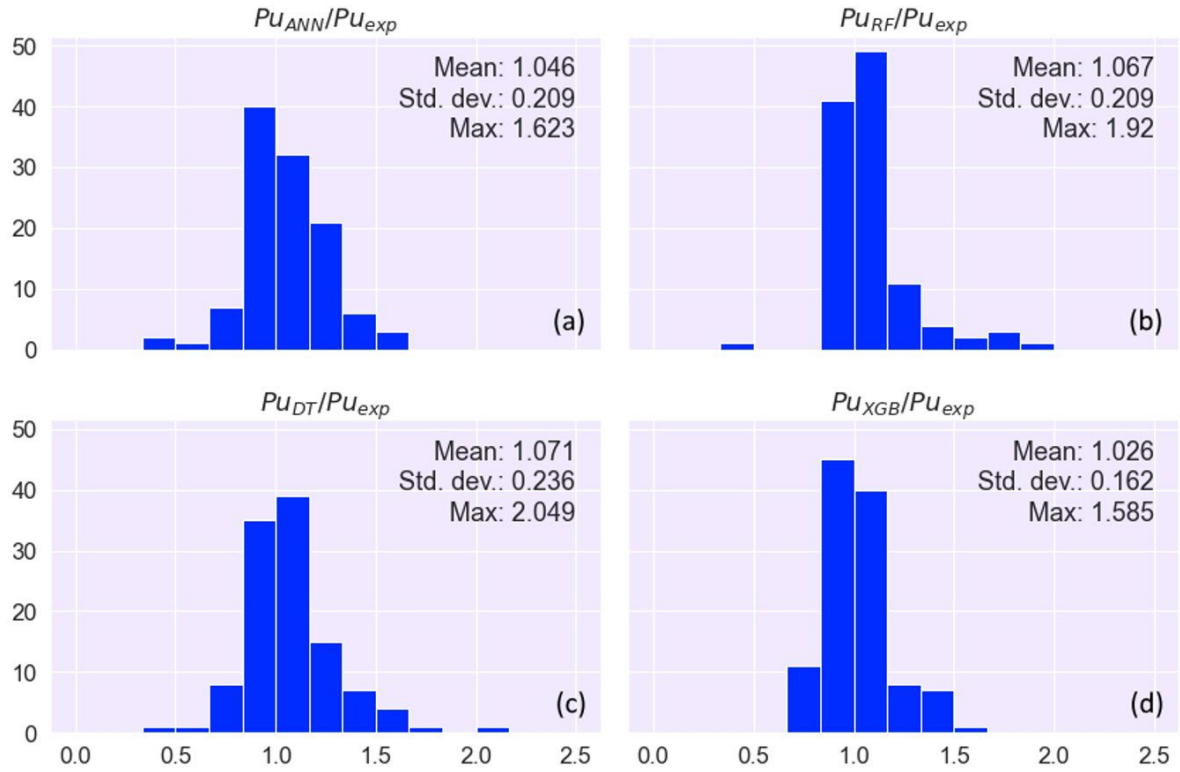

Figure 11. Histograms of the predicted/actual punching shear strength ratios.

In applications related to structural engineering, it is essential to understand whether the model usually leads to values in favor of safety, that is, conservative, or not. In this sense, the method proposed in [48] is employed herein to compare the different ML models. Table 3 presents the calculated conservatism indexes (C) and indicates that all models are considered slightly non-conservative according to the method, with negative values.

Table 3. Conservatism indexes.

Model	C
XGB	-0.0262
ANN	-0.0464
RF	-0.0671
DT	-0.0714

Therefore, considering the metrics in Figures 10 and 11 and Table 3, it is decided, from now on, to use only the XGB model for the purposes of comparison with design standard models in this study.

After selecting the best model, a feature importance analysis is performed to identify the impact of the input variables on the output. The bar graph in Figure 12 shows the feature importance through their gain. The feature importance based on this metric shows the average gain across all splits where a particular feature is used. Therefore, higher gain means higher importance. As presumed in the data engineering step, the effective depth is the most important variable, followed by the column dimension, reinforcement ratio and steel yield strength. The concrete strength is the variable that provides the smallest gain in this model. These results agree with what was observed by [10].

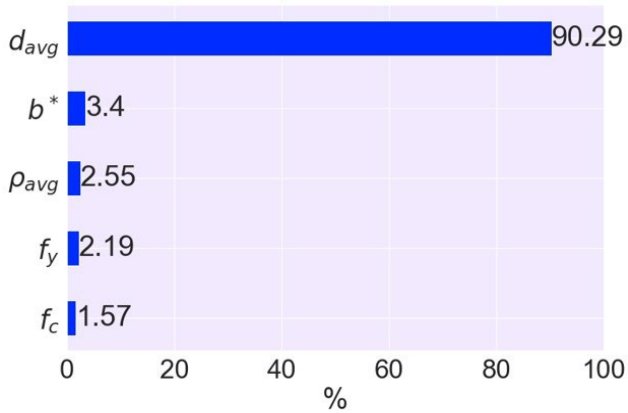

Figure 12. Relative feature importance in the XGB model.

7 COMPARATIVE STUDY OF MODELS

The equations for RC slab design according to the design standards [21]–[25] are presented in section 3. It is noted that none of these codes directly use the column width in their formulas. Instead, a critical perimeter around the column is considered, which depends on its dimensions and the effective depth of the slab. None of the standards uses the f_y parameter in the design, while the American standard also ignores the longitudinal reinforcement ratio. Most standards differ in calculating the size effect factor.

In all evaluations performed with the standards, for a fair comparison with the XGB model, only the 112 slabs used in the XGB test dataset are considered. The regplots, R^2 and RMSE of the design code models and XGB are shown in Figure 13.

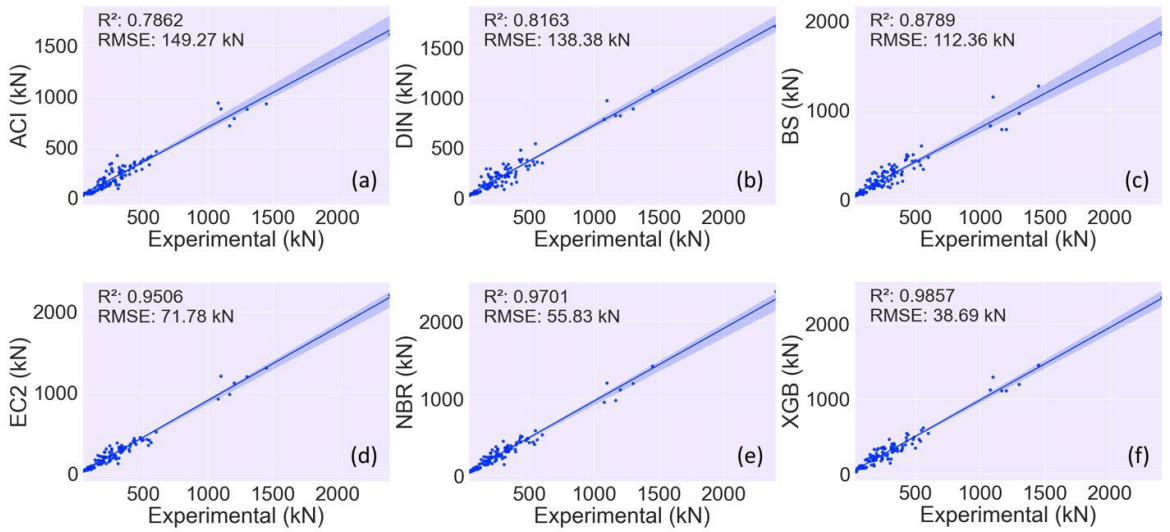

Figure 13. Regplots with design code results.

From Figure 13 it is observed that, although providing accurate results, all design models show inferior accuracy than the XGB. The ACI (Figure 13a) shows the least accurate results, followed by DIN (Figure 13b), BS (Figure 13c), EC2 (Figure 13d) and NBR (Figure 13e). The XGB (Figure 13f) is the model that shows the highest R^2 and the lowest RMSE.

The histograms of the $P_{u,pred}/P_{u,experimental}$ ratios are shown in Figure 14. Based on the results presented, there is a tendency for the ACI (Figure 14a) to underestimate the strength of the slabs, being a conservative model, due to the method used to calculate the control perimeter (only 0.5d away from the column face). The DIN (Figure 14b), BS (Figure 14c) and EC2 (Figure 14d) standards, in general, also show conservative results. These standards tend to underestimate the slabs resistances, as expected. The NBR (Figure 14e) shows better estimates than the other standards, as the NBR does not limit the size effect factor (ξ) and the reinforcement ratio in its formulas, obtaining results closer to reality than the other standards. Between the design models, the NBR shows the closest to 1 average error. Thus, the recommendation of NBR is the most indicated by the authors regarding the size effect.

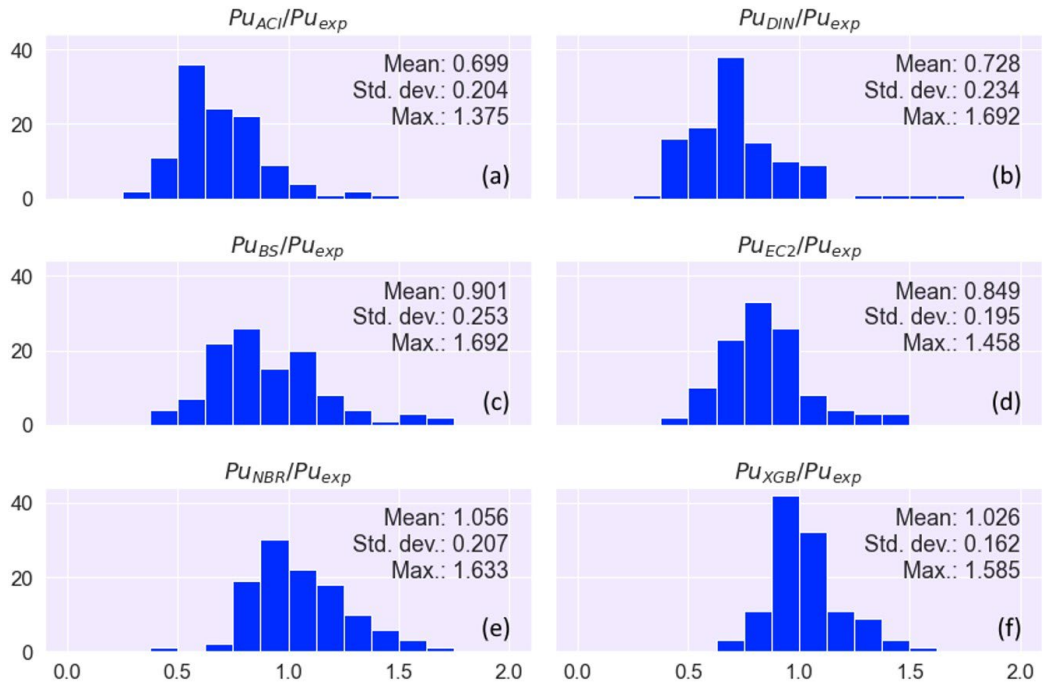

Figure 14. Histograms of the predicted/actual punching shear strength ratios with design code results.

From Figure 14, the ACI, DIN, BS, EC2 and NBR models estimate, on average, failure loads 30.1% lower, 27.2% lower, 9.9% lower, 15.1% lower and 5,6% higher than the real ones, respectively. On the other hand, the XGB estimates values, on average, 2.6% higher than the experimental ones.

Table 4 shows the descriptive statistics of the ultimate load for each model. The NBR and XGB showed similar statistics for the 112 slabs, with the NBR being closer to the actual mean and the XGB closer to the experimental standard deviation.

Table 4. Descriptive statistics of the results.

	Exp. (kN)	**ACI (kN)**	**DIN (kN)**	**BS (kN)**	**EC2 (kN)**	**NBR (kN)**	**XGB (kN)**
Mean	**320.02**	225.95	233.85	275.50	278.07	322.18	323.87
Std. dev.	**324.29**	228.28	240.08	259.18	299.74	310.96	320.02
Min.	**34.00**	19.81	20.80	33.39	24.24	42.54	30.96
25%	**159.00**	91.06	102.52	132.84	122.20	153.55	173.34
50%	**244.00**	169.28	201.92	208.00	210.35	252.88	248.70
75%	**346.25**	280.44	282.00	326.18	320.13	370.04	352.55
Max.	**2400.00**	1609.56	1710.26	1839.25	2201.13	2386.79	2348.72

Table 5 shows the results for conservatism indexes (C) calculated according to the method presented in [48], with results for the design code models. Despite presenting the lowest conservatism index and having predicted, on average, values greater than the real ones, it cannot be said that the NBR was against safety, as there may be other factors that influence resistance that were not addressed in this study. For example, according to Sousa and Debs [49], the aggregate interlock effect is not accounted in the ACI, EC2 and NBR standards, which is known to be an important factor in the shear resistance of concrete structures.

Table 5. Conservatism indexes with design code results.

Model	C
ACI	0.3011
DIN	0.2715
EC2	0.1510
BS	0.0989
NBR	-0.0557
XGB	-0.0262

8 CONCLUSIONS

In this paper, four ML models were developed to predict the punching shear strength of RC flat slabs based on 373 experimental results. After a comparative analysis between them, the model based on extreme gradient boosting (XGB) was considered the best one, with higher R^2, lower RMSE and lower error ratio.

The XGB model was compared with models of five design standards: ABNT NBR 6118 [21], ACI 318 [22], Eurocode 2 [23], BS 8110 [24] and DIN 1145-1 [25]. It was observed that the standard models, although showing satisfactory results in general, presented lower accuracy not only in comparison with the XGB, but also with the other developed ML models. The design code models were compared to the experimental data, where the one that came closest to the real results was the NBR, while the ACI was the most distant from the actual resistances. It was found that the ACI, EC2, BS and DIN tend to underestimate the strength of the slabs, providing values on average lower than the real ones, being conservative.

As a result, the following conclusions are drawn:

1. The XGB model had the best performance between the developed ML models and it was closer to the real results than the studied design code models, showing good ability to predict the punching shear strength of flat RC slabs, with R^2 equal to 0.9857 and RMSE of 38.69 kN for the test data;
2. Between the five input variables of the XGB model, it was found that the effective depth of the slab is the most important, and the properties of concrete and steel are the least relevant;
3. Between the studied models from design codes, there were large discrepancies in the results and the NBR 6118 [21] model showed the best overall performance in predicting the failure load of the slabs, as well as the lowest conservatism, followed by Eurocode 2 [23], BS 8110 [24], DIN 1145-1 [25] and ACI 318 [22].

ACKNOWLEDGEMENTS

The authors would like to thank CAPES for the financial support during this research.

REFERENCES

[1] D. S. Lourenço, E. A. P. Liberati, M. G. Marques, L. C. Almeida, and L. M. Trautwein, "Reinforced concrete flat slabs with openings at different distances from the column," *Rev. IBRACON Estrut. Mater.*, vol. 14, no. 1, e14111, 2021, http://dx.doi.org/10.1590/s1983-41952021000100011.

[2] A. Muttoni, "Punching shear strength of reinforced concrete slabs without transverse reinforcement," *ACI Struct. J.*, vol. 105, no. 4, pp. 440–450, 2008. Accessed: May 20, 2022. [Online]. Available: https://www.researchgate.net/publication/37455726_Punching_Shear_Strength_of_Reinforced_Concrete_Slabs_without_Transverse_Reinforcement

[3] R. A. Sanabria, L. M. Trautwein, and L. C. Almeida, "Análise numérica da resistência à punção em lajes de concreto armado sem armadura de cisalhamento," in *An. 59º CBC*, Bento Gonçalves, Nov., 2017, pp. 1–19.

[4] H. Avagyan, "Punching shear resistance of slab with shear reinforcement according to Armenian and foreign building standards," *Teh. Vjesn.*, vol. 26, no. 1, pp. 279–282, 2019, http://dx.doi.org/10.17559/TV-20170705153240.

[5] R. J. C. Silva, D. R. C. Oliveira, N. G. B. Albuquerque, F. E. S. Silva Júnior, and P. V. Sacramento, "Computational modeling of flat slabs: Influence of ribs and flexural reinforcement on shear strength," *Lat. Am. J. Solids Struct.*, vol. 17, no. 6, pp. 1–16, 2020, http://dx.doi.org/10.1590/1679-78256128.

[6] M. S. Issa and E. Ismail, "Punching strength of conventional reinforced concrete flat slabs," *HBRC J.*, vol. 17, no. 1, pp. 77–91, 2021, http://dx.doi.org/10.1080/16874048.2021.1901401.

[7] Z. Xu and J. H. Saleh, "Machine learning for reliability engineering and safety applications: Review of current status and future opportunities," *Reliab. Eng. Syst. Saf.*, vol. 211, pp. 1–50, 2021, http://dx.doi.org/10.1016/j.ress.2021.107530.

[8] M. Su, Q. Zhong, H. Peng, and S. Li, "Selected machine learning approaches for predicting the interfacial bond strength between FRPs and concrete," *Constr. Build. Mater.*, vol. 270, pp. 1–16, 2021, http://dx.doi.org/10.1016/j.conbuildmat.2020.121456.

[9] D. Feng, Z. Liu, X. Wang, Z. Jiang, and S. Liang, "Failure mode classification and bearing capacity prediction for reinforced concrete columns based on ensemble machine learning algorithm," *Adv. Eng. Inform.*, vol. 45, pp. 101–126, 2020, http://dx.doi.org/10.1016/j.aei.2020.101126.

[10] H. D. Nguyen, G. T. Truong, and M. Shin, "Development of extreme gradient boosting model for prediction of punching shear resistance of r/c interior slabs," *Eng. Struct.*, vol. 235, pp. 1–14, 2021, http://dx.doi.org/10.1016/j.engstruct.2021.112067.

[11] S. Mangalathu, H. Shin, E. Choi, and J. Jeon, "Explainable machine learning models for punching shear strength estimation of flat slabs without transverse reinforcement," *J. Build. Eng.*, vol. 39, pp. 1–10, 2021, http://dx.doi.org/10.1016/j.jobe.2021.102300.

[12] S. Lu, M. Koopialipoor, P. G. Asteris, M. Bahri, and D. J. Armaghani, "A novel feature selection approach based on tree models for evaluating the punching shear capacity of steel fiber-reinforced concrete flat slabs," *Materials*, vol. 13, no. 17, pp. 1–20, 2020, http://dx.doi.org/10.3390/ma13173902.

[13] H. Ly, T. Le, H. T. Vu, V. Q. Tran, L. M. Le, and B. T. Pham, "Computational hybrid machine learning based prediction of shear capacity for steel fiber reinforced concrete beams," *Sustainability*, vol. 12, no. 7, pp. 1–34, 2020, http://dx.doi.org/10.3390/su12072709.

[14] W. J. S. Gomes, "Shallow and deep artificial neural networks for structural reliability analysis," *J. Risk Uncertain Eng. Syst. Part B Mech. Eng.*, vol. 6, no. 4, pp. 1–8, 2020, http://dx.doi.org/10.1115/1.4047636.

[15] M. Bilgehan and P. Turgut, "The use of neural networks in concrete compressive strength estimation," *Comput. Concr.*, vol. 7, no. 3, pp. 271–283, 2010, http://dx.doi.org/10.12989/cac.2010.7.3.271.

[16] K. Yan and C. Shi, "Prediction of elastic modulus of normal and high strength concrete by support vector machine," *Constr. Build. Mater.*, vol. 24, no. 8, pp. 1479–1485, 2010, http://dx.doi.org/10.1016/j.conbuildmat.2010.01.006.

[17] J. Chou, C. Chiu, M. Farfoura, and I. Al-Taharwa, "Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques," *J. Comput. Civ. Eng.*, vol. 25, no. 3, pp. 242–253, 2011, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000088.

[18] J. Chou, C. Tsai, A. Pham, and Y. Lu, "Machine learning in concrete strength simulations: Multi-nation data analytics," *Constr. Build. Mater.*, vol. 73, pp. 771–780, 2014, http://dx.doi.org/10.1016/j.conbuildmat.2014.09.054.

[19] P. Prasanna et al., "Automated Crack Detection on Concrete Bridges," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 591–599, 2014, http://dx.doi.org/10.1109/TASE.2014.2354314.

[20] K. THIYAGARAJAN, S. KODAGODA, N. ULAPANE, "Data-driven machine learning approach for predicting volumetric moisture content of concrete using resistance sensor measurements," in *Proc. 11th ICIEA*, Hefei, June, 2016, pp. 1288-1293, http://dx.doi.org/10.1109/ICIEA.2016.7603783.

[21] Associação Brasileira de Normas Técnicas, *Projeto de Estruturas de Concreto – Procedimento*, ABNT NBR 6118, 2014.

[22] American Concrete Institute, *Building Code Requirements for Structural Concrete and Commentary*, ACI 318M, 2019.

[23] European Committee For Standardization, *Design of Concrete Structures – Part 1-1: General Rules and Rules for Buildings*, Eurocode 2, 2004.

[24] British Standards Institution, *Structural Use of Concrete - Part 1: Code of Practice for Design and Construction*, BSI BS 8110, 1997.

[25] German Institute for Standardization, *Plain, Reinforced and Prestressed Concrete Structures-Part 1: Design and Construction*, DIN 1045-1, 2008.

[26] M. E. Fenner, *Machine Learning with Python for Everyone*, 1st ed. London: Pearson Educ., 2020.

[27] C. Albon, *Python Machine Learning Cookbook*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2018.

[28] C. C. Aggarwal, *Data Mining: The Textbook*, 1st ed. New York, NY, USA: Springer, 2015, http://dx.doi.org/10.1007/978-3-319-14142-8.

[29] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, 2nd ed. London: MIT Press, 2018.

[30] J. D. Kelleher, B. M. Namee, and A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics. Algorithms, Worked Examples, and Case Studies*, 2nd ed. London: MIT Press, 2020.

[31] W. J. S. Gomes, "Shallow and deep artificial neural networks for structural reliability analysis," *J. Risk Uncertain Eng. Syst. Part B Mech. Eng.*, vol. 5, no. 4, pp. 1–8, 2019, http://dx.doi.org/10.1115/1.4044040.

[32] M. Z. Naser, "Machine learning assessment of fiber-reinforced polymer-strengthened and reinforced concrete members," *ACI Struct. J.*, vol. 117, no. 6, pp. 237–251, 2020, http://dx.doi.org/10.14359/51728073.

[33] E. Darvishan, "The punching shear capacity estimation of FRP strengthened RC slabs using artificial neural network and group method of data handling," *J. Rehabil. Civ. Eng.*, vol. 9, no. 1, pp. 102–113, 2021, http://dx.doi.org/10.22075/jrce.2020.20335.1407.

[34] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, http://dx.doi.org/10.1023/A:1010933404324.

[35] O. B. Olalusi and P. O. Awoyera, "Shear capacity prediction of slender reinforced concrete structures with steel fibers using machine learning," *Eng. Struct.*, vol. 227, pp. 1–13, 2021, http://dx.doi.org/10.1016/j.engstruct.2020.111470.

[36] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, San Francisco (CA), Aug., 2016, pp. 785-794, http://dx.doi.org/10.1145/2939672.2939785.

[37] S. Russel and P. Norvig, *Artificial Intelligence – A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson Education, 2021.

[38] J. Duan, P. G. Asteris, H. Nguyen, X.-N. Bui, and H. Moayedi, "A novel artificial intelligence technique to predict compressive strength of recycled aggregate concrete using ICA-XGBoost model," *Eng. Comput.*, vol. 37, no. 4, pp. 3329–3346, 2020, http://dx.doi.org/10.1007/s00366-020-01003-0.

[39] D. Sathyan, D. Govind, C. B. Rajesh, K. Gopikrishnan, G. Aswath Kannan, and J. Mahadevan, "Modelling the shear flow behaviour of cement paste using machine learning – XGBoost," *J. Phys. Conf. Ser.*, vol. 1451, no. 1, pp. 1–8, 2020, http://dx.doi.org/10.1088/1742-6596/1451/1/012026.

[40] M. Wang, D. Huang, G. Wang, and D. Li, "SS-XGBoost: a machine learning framework for predicting newmark sliding displacements of slopes," *J. Geotech. Geoenviron. Eng.*, vol. 146, no. 9, pp. 1–17, 2020, http://dx.doi.org/10.1061/(ASCE)GT.1943-5606.0002297.

[41] N. M. Shahani, X. Zheng, C. Liu, F. U. Hassan, and P. Li, "Developing an XGBoost regression model for predicting young's modulus of intact sedimentary rocks for the stability of surface and subsurface structures," *Front. Earth Sci.*, vol. 9, pp. 1–13, 2021, http://dx.doi.org/10.3389/feart.2021.761990.

[42] A. Kaveh, S. M. Javadi, and R. M. Moghanni, "Shear strength prediction of FRP-reinforced concrete beams using an extreme gradient boosting framework," *Period. Polytech. Civ. Eng.*, vol. 66, no. 1, pp. 18–29, 2022, http://dx.doi.org/10.3311/PPci.18901.

[43] G. S. Santos, W. G. Nicácio, A. W. Lima, and G. S. S. A. Melo, "Punching strengthening in flat plates of reinforced concrete with carbon fiber reinforced polymer (CFRP)," *IBRACON Struct. Mater. J.*, vol. 7, no. 4, pp. 592–625, 2014.

[44] M. H. Oliveira, M. J. M. Pereira Filho, D. R. C. Oliveira, M. P. Ferreira, and G. S. S. A. Melo, "Punching resistance of internal slab-column connections with double-headed shear studs," *Rev. IBRACON Estrut. Mater.*, vol. 6, no. 5, pp. 681–714, 2013., http://dx.doi.org/10.1590/S1983-41952013000500002.

[45] J. K. Wight and J. G. MacGregor, *Reinforced Concrete Mechanics and Design*, 6th ed. Upper Saddle River, NJ, USA: Pearson Educ., 2012.

[46] F. Pedregosa et al., "Scikit-learn: machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011. Accessed: May 20, 2022. [Online]. Available: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[47] K. G. Sheela and S. N. Deepa, "Review on methods to fix number of hidden neurons in neural networks," *Math. Probl. Eng.*, vol. 2013, pp. 1–11, 2013, http://dx.doi.org/10.1155/2013/425740.

[48] W. J. S. Gomes and A. T. Beck, "A conservatism index based on structural reliability and model errors," *Reliab. Eng. Syst. Saf.*, vol. 209, pp. 1–15, 2021, http://dx.doi.org/10.1016/j.ress.2021.107456.

[49] A. M. D. Sousa and M. K. El Debs, "Shear strength analysis of slabs without transverse reinforcement under concentrated loads according to ABNT NBR 6118:2014," *Rev. IBRACON Estrut. Mater.*, vol. 12, no. 3, pp. 658–693, 2019, http://dx.doi.org/10.1590/s1983-41952019000300012.

Author contributions: FESSJ: conceptualization, data curation, formal analysis, methodology, writing. WJSG: formal analysis, methodology, writing, supervision.

Editors: Leandro Mouta Trautwein, Guilherme Aris Parsekian.