Contributions to the analysis of prestressed steel and concrete composite beams using machine learning algorithms

Beatriz M. Meneses[1], João B.M. Sousa Jr.[1]

[1]*Dept. Structural Engineering, Federal University of Ceará*
Campus do Pici - Bloco 728, 60440-900, Fortaleza/CE, Brazil
beatrizmatos@alu.ufc.br, joaobatistasousajr@ufc.br

Abstract. In the practice of civil construction, one of the structural alternatives used so that beams can withstand the required loads in a more efficient and economical way are the prestressed composite beams, especially steel and reinforced concrete. Posttensioned composite steel-concrete beams (PSCCB) have a greater range of elastic behavior, as well as yield and ultimate load values. In continuous beams, there is a reduction of cracking in the hogging moment region. They may also have better fatigue performance and employ lighter steel sections. The present paper aims to create a large database for the analysis of PSCCB, based on a previously developed nonlinear finite element model, which performs static non-linear analysis of PSCCB's, considering the partial interaction between steel and concrete. With a database developed from a variation of the parameters in the numerical models, three Machine Learning models are developed and trained with the objective of predicting the beam ultimate load, deflection, and final tendon force. The implemented procedures is compared, whenever possible, with experimental and numerical results available in classic literature, mainly as a reference of test data to evaluate the success of Machine Learning algorithms.

Keywords: prestressed composite beams, nonlinear analysis, machine learning.

1 Introduction

The advantages of steel and concrete composite beams over conventional concrete structures include reduced dead weight, increased span length with smaller deformations, and greater efficiency and precision of execution [1]. On the other hand, analyses of this type of structure are more complex, since the individual characteristics of each constituent material and the interaction between them are taken into account. In addition, it is necessary to consider a partial interaction between the concrete slab and the metal profile, which causes longitudinal displacements between the materials [2].

Some works studied the contribution of external prestressing cables to further increase the load capacity of the structural system. Saadatmanesh et al. [3] mentions that the main advantages of using external prestressing are: smaller deformations in service, control of cracks in regions of negative moment, greater load-carrying capacity, improvement in elastic behavior, better fatigue resistance, savings in the total construction value as well as reduction in the weight of the beam element. In the parametric study by Almeida et al. [4], it was found that increasing the initial prestressing force has no significant effect on the overall behavior and that the influence of the length of the prestressing cable was significant. Regarding the span size of PSCCB's, longer beams need a higher prestressing setting than shorter beams to achieve the same improvement in overall behavior and high ultimate moment resistance. Sousa Jr. et al. [2] developed a 10-dof finite element for nonlinear analysis of PSCCB's by external tendons. They show the influence of considering partial interaction for the correct simulation of the prestressed composite beam both in terms of stiffness and ultimate strength and the importance of the nonlinear geometric effects of the tendon in the distribution of moments and deformations along the beam.

Most of the studies available in the literature that analyze the structural behavior of PSCCB's with partial interaction focus on experimental analyses and/or the implementation of finite elements for numerical analysis of structural behavior [3, 5–9]. Despite the effectiveness of these numerical models in representing the physical and geometric nonlinear behavior of composite beams when compared with experimental results, the time cost and complexity of implementing these elements has made it difficult for these studies to be more widely disseminated for use by anyone interested in the behavior, analysis, and design of PSCCB's.

The use of Machine Learning (ML) algorithms in structural engineering has been growing steadily in recent years. ML is the most successful branch of artificial intelligence (AI) in current technology, with applications in diverse areas of study. This computational tool works based on algorithms that teach the ”machine” to make predictions and estimates of a given objective based on training on a set of available data. In structural engineering, ML has been employed in several areas [10], but there is still a large field within structural engineering that has not yet been explored by ML algorithms due to the lack of training data. ML models to predict deflections of composite beams, most commonly those developed by neural network algorithms, have been successful in predicting the behavior of composite structures. A line of work using neural networks to predict deflections in composite bridges considering partial interaction between the components was developed and when compared to finite element analyses, it had a relatively lower computational cost [11, 12]. Sakr and Sakla [13] built a large database varying different design parameters of the composite beam with the aim of training and verifying the two neural network models. However, no studies of prestressed composite beams using machine learning algorithms have yet been found.

Therefore, this work aims to generate a database on the structural behavior of prestressed composite beams with partial interaction from the numerical model for nonlinear analysis of prestressed composite beams developed by authors who are partners in the research line in a FE program [2, 14, 15]. From this database, different ML methods will be developed and trained in order to predict some important responses of PSCCB’s in nonlinear analysis.

Structural responses, such as the failure load, the maximum vertical displacement of the beam when it reaches the failure load and the final tendon force will be estimated. Finally, the developed ML methods are validated by evaluating their performance when real data from analyses of PSCCB’s with partial interaction are used as test data for the models.

2 Machine Learning Algorithms

Machine Learning (ML) is a multidisciplinary computational tool that consists of teaching a ”machine” to predict the correct answer that one wants to obtain from an initial database. This initial database, also called instances, is divided into features, which are the attributes that represent instances, and labels, which consist of the target value. When an ML model presents adequate features of the initial data and a good algorithm performance, it can generalize a response from examples that are not in the initial database. The initial data is separated into test and training data. Initially, the algorithm uses the test data to teach the model and the test data is later used to evaluate the model’s performance. Three main steps are necessary to build a good ML model: choosing and processing the initial data, training the data, and finally evaluating the model.

In this paper, some ML algorithms that use the predictive regression technique will be used in order to compare the performance of each one and obtain a model that results in reliable structural behavior of prestressed composite beams with a lower computational processing cost. The next sections present the algorithms that will be used in this paper.

2.1 Linear Regression

Linear regression is one of the simplest supervised learning models, which consists of estimating the values of the parameters/coefficients so that the model has a good fit to the data. The intercept (b) and the weights ($\mathbf{w} = (w_0, w_1, ..., w_n)$) are estimated to result in a predicted value (y):

$$y = b + w_0 x_0 + w_1 x_1 + ... + w_n x_n \tag{1}$$

where $\mathbf{x} = (x_0, x_1, ..., x_n)$ are the features of the model.

There are different methods for estimating intercept and weight values from training data, depending on different adjustment criteria, objectives, and different ways of controlling model complexity. The learning algorithm tends to find the best parameters that optimize an objective function, usually by minimizing a function (loss function) between the estimated values and the real values. Two linear regression methods will be discussed in this section: Ridge and Lasso Regression. Lasso regression (LR) is used when there is a high correlation between the model’s input variables and when few parameters have a very high relevance, while Ridge regression (RR) is used when many parameters have medium or low relevance. RR finds w and b that minimize the sum of the quadratic differences between the real values and those estimated using the mean square error (MSE), but adds a penalty (α) to increase the variation of the weight values. The hyperparameter α controls the regularization of the model. Ridge regularization uses L2 regularization (squared value of the weight), which minimizes the sum of the squares of w to eliminate the weights of the least influential features. Regularization is a penalty given to the parameters and prevents overfitting by restricting the model, usually reducing its complexity. In the case of RR, increasing

α simplifies the model. LR uses regularization of L1 (absolute weight value) for training, in which the weights relative to the least influential variables tend to 0, generating a model that selects the features of greatest relevance to predict the target value.

2.2 Support Vector Machine

Support Vector Machine (SVM) algorithms were first used to solve classification problems, but were later developed for regression problems, *support vector regression* (SVR) [16]. The SVR algorithm finds a function that best fits the data points within a decision boundary using linear regression. The best-fit line is the hyperplane that has the maximum number of data points within a threshold value ε. Since in real-world the data are not linearly separable, it is impossible to find a separating hyperplane. In this case, SVR algorithm has procedures for using a penalty parameter to control the trade-off between maximizing the hyperplane margins and minimizing the total slack distance ξ when the data point is on the wrong side of the margin. Then, kernel functions are used to map the original data that does not follow a linear plane into a new space where the data is grouped linearly. The most commonly used kernel functions are linear and non-linear polynomials, RBF, and sigmoid function. Both kernel functions and penalty parameters have significant effects on the performance of SVR models.

2.3 Decision Tree

Decision Tree (DT) is a popular and easy to use and understand method for Regression and Classification. It is an exploratory method that helps to understand how variables influence the model and is based on operations by control structures, such as *"if/else"*, in its algorithm. The complexity of the model is controlled by the tree maximum depth. The DT has four elements: a root node, two or more branches, decision nodes and leaf nodes (terminal). The root node is the highest decision node in a tree that represents the final objective. The leaf node located at the end of the branch indicates a final decision to be made, while the decision node represents a condition that causes a data set to be split. The split condition can be based on different metrics, such as Gini index, entropy, information gain and MSE (regression problem). The division process is repeated on each derived subset until a division is found that reduces the metrics used or reaches the maximum depth of the tree, which is defined by the person who starts the algorithm. The greater the depth, the greater the possibility of overfit and the higher the computational cost.

2.4 Neural Network (Perceptron)

A Perceptron is a mathematical model for supervised learning, a single-layer neural network that takes multiple inputs and produces a single binary output. For more complex problems, it is necessary to introduce a network with hidden layers. A hidden layer allows the network to reorganize the input data. The multilayer perceptron algorithm can have different numbers of hidden layers. Without hidden layers, it is only capable of representing linear functions. With 1 hidden layer, it can approximate any function that is a continuous mapping from one finite space to another. With 2 hidden layers, it can represent non-continuous functions and arbitrary decision lines. The increase in the number of neurons/layers implies an increase in the complexity of the algorithm. As in linear regression models (ridge and lasso), regularization can be used to make the model have more weights close to 0. The parameter α is used for this purpose; a higher value of α implies greater regularization. The MLP regression algorithm presents different activation functions so that the data from the hidden layer are transformed into the output layer.

3 Results and discussion

Data acquisition will be done using the finite element program for nonlinear analysis of prestressed composite beams with partial interaction, developed by authors who are companions to the research line program [2, 14, 15]. The input parameters for analysis, also called attributes or features, will be varied to generate a diversified database. Two CP 190 RB prestressing tendons were considered with initial prestressing force applied to the tendon set at 10 tf and 15 tf for 12.7 mm and 15.7 mm tendons, respectively. The parameters, as well as the criteria for their variation, are listed below. Regarding the varied parameters that describe the geometry of the model, Figure 1 presents an illustration of a composite beam identifying each variable described below:

- L (mm) - Beam span: Between 15 and 40 times the height of the steel profile (Hs);
- Fcm (MPa) - Compressive strength of concrete: Between 20 and 50 MPa;

- Ec (GPa) - Modulus of elasticity of concrete: $Ec = 5600\sqrt{fcm}$
- Hc (mm) - Height of the concrete section: Between 80 and 250 mm [13];
- Bc (mm) - Width of the concrete section: Between half the height of the steel section (Hs/2) and 25% of the beam span [13];
- Ac (mm²) - Area of the concrete section;
- Hs (mm) - Height of the steel profile: Between 200 and 600 mm according to the Gerdau catalog [17];
- W (mm) - Width of the steel profile flange: limits of the Gerdau catalog [17];
- Tw (mm) - Thickness of the steel profile web: limits from the Gerdau catalog [17];
- Tf (mm) - Thickness of the steel profile flange: limits from the Gerdau catalog [17];
- Dp (mm) - Diameter of the prestressing cable: 12.7, 15.7 mm according to the Belgo Bakaert Arames catalog [18];
- Fp (tf) - 10 tf for 12.7 mm prestressing cables and 15tf for 15.7 mm cables
- LocAsp (mm) - Location of the prestressing cable: Half the distance of the height of the steel profile $\pm$ 50;
- Scon - Longitudinal spacing between connectors: Between 6 times the cable diameter, minimum spacing between connectors ([19]), and the height of the concrete section (Hc).

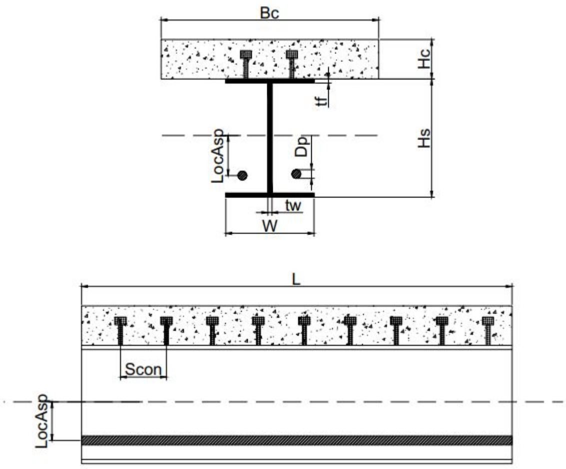

Figure 1. Illustration of a composite beam identifying parameters of the geometry to be varied.

From these features, a considerable number of different instances will be generated from random numbers using a Latin Hypercube Sampling function, respecting the parameter limit criteria. Each group of attributes of each instance will be simulated in the finite element program to generate the results, target values: FatCarg (N) - Load factor at rupture; Deslocmax (m) - Maximum vertical displacement at the moment of rupture; FCabo (N) - Force acting on the prestressing tendon at the end of the analysis.

3.1 Pre-processing of the database

The results were evaluated and those that did not converge were excluded, thus obtaining a database with 110 models. In Python, all the pre-processing of these data was carried out to perform the statistical study, treatment, normalization and cleaning. Table 1 presents all the responses considered important from the descriptive statistics of the generated database. Figure 2 presents a table showing the correlation between all the variables in the database, with a darker shade of blue indicating a greater parameter correlation, that is, a greater association between the parameters studied.

3.2 Training and performance of machine learning models

In this section, training, testing and evaluation of the different models are carried out so that the models with the best performance can be selected and, subsequently, validated with the data available in the literature. Five machine learning methods are developed: RR, LR, Support Vector Machine (SVM), Decision Tree and Neural Networks. First, for each model, the data will be separated between training and testing so that training can be carried out. This step will be carried out both by performing a simple training-test separation (using *train_test_split*) and by cross-validation (*cross_val_score*), which allows extracting data from the same underlying distribution for

Table 1. Statistical study of database.

Parameter	Mean	Stand. Deviation	Min. Value	Max. Value	25%	50%	75%
L (mm)	16553.07	5137.99	4572.00	24963.95	12556.17	17218.60	20426.79
Hc (mm)	171.47	48.77	76.00	249.71	128.55	177.72	214.24
Bc (mm)	836.44	441.83	135.18	1869.41	473.67	822.15	1089.06
Hs (mm)	442.55	110.49	204.11	609.09	355.31	455.74	541.37
W (mm)	219.9	61.60	101.91	322.96	169.77	225.28	271.80
tw (mm)	8.84	2.69	4.38	13.88	6.65	8.73	11.09
tw (mm)	13.4	4.72	5.01	21.59	9.91	13.57	17.24
Dp (mm)	13.91	1.49	12.70	16.00	12.70	12.70	15.70
Fp (tf)	11.89	2.41	9.00	15.00	10.00	10.00	15.00
LocAsp (mm)	-395.71	236.57	-931.05	46.00	-555.58	-381.41	-210.07
Scon (mm)	140.44	38.58	95.00	236.12	105.45	131.55	169.74
FatCarg (kN)	436.61	184.52	79.79	1023.09	288.34	434.57	560.06
Deslocmax (m)	-0.37	0.157	-0.69	-0.052	-0.48	-0.39	-0.25
FCabo (kN)	206.56	66.31	70.57	348.85	163.93	208.01	228.77

	L	Hc	Bc	Hs	W	tw	tf	Dp	Fp	LocAsp	Scon	FatCarg	Deslocmax	FCabo
L	1.000000	0.204263	0.217996	0.313661	0.137850	0.204405	0.153779	-0.212975	-0.097744	-0.376311	-0.047660	-0.426930	-0.918292	0.008937
Hc	0.204263	1.000000	0.090530	0.087628	0.052760	-0.019553	-0.118164	-0.017569	0.076171	-0.199537	0.706163	0.126975	-0.147980	0.307139
Bc	0.217996	0.090530	1.000000	0.459050	-0.069984	0.049361	0.021423	-0.096409	-0.113513	-0.910967	0.181056	0.483006	-0.420245	0.510136
Hs	0.313661	0.087628	0.459050	1.000000	-0.216822	0.032047	-0.181945	0.005881	0.056728	-0.516475	0.061735	0.427218	-0.357279	0.454356
W	0.137850	0.052760	-0.069984	-0.216822	1.000000	0.090200	0.183959	0.059524	0.111545	-0.014704	-0.083611	0.084198	-0.142823	0.009072
tw	0.204405	-0.019553	0.049361	0.032047	0.090200	1.000000	0.136578	-0.000362	0.039561	-0.114466	-0.011899	0.035207	-0.242968	-0.009855
tf	0.153779	-0.118164	0.021423	-0.181945	0.183959	0.136578	1.000000	0.074151	0.108470	-0.059028	-0.034766	0.072235	-0.194309	0.013862
Dp	-0.212975	-0.017569	-0.096409	0.005881	0.059524	-0.000362	0.074151	1.000000	0.955123	0.218237	0.020246	0.254307	0.316879	0.430198
Fp	-0.097744	0.076171	-0.113513	0.056728	0.111545	0.039561	0.108470	0.955123	1.000000	0.137765	0.040428	0.223333	0.225829	0.544164
LocAsp	-0.376311	-0.199537	-0.910967	-0.516475	-0.014704	-0.114466	-0.059028	0.218237	0.137765	1.000000	-0.185266	-0.423237	0.544929	-0.621607
Scon	-0.047660	0.706163	0.181056	0.061735	-0.083611	-0.011899	-0.034766	0.020246	0.040428	-0.185266	1.000000	0.260613	0.027861	0.247727
FatCarg	-0.426930	0.126975	0.483006	0.427218	0.084198	0.035207	0.072235	0.254307	0.223333	-0.423237	0.260613	1.000000	0.248404	0.626697
Deslocmax	-0.918292	-0.147980	-0.420245	-0.357279	-0.142823	-0.242968	-0.194309	0.316879	0.225829	0.544929	0.027861	0.248404	1.000000	-0.040162
FCabo	0.008937	0.307139	0.510136	0.454356	0.009072	-0.009855	0.013862	0.430198	0.544164	-0.621607	0.247727	0.626697	-0.040162	1.000000

Figure 2. Correlation between features and labels.

training and evaluation by performing multiple training-test separations. This prevents the model from appearing to perform well only in a simple random training-test separation and the rest of the data from being unrepresentative. For this purpose, the tool *GridSearchCV* was used, which, in addition to performing cross-validation, automates the process of adjusting the parameters of an algorithm, making various combinations of parameters and indicating which model performed best.

In this development, the following parameters were tested for each ML algorithm evaluated, so that the model with the best performance can be selected.

- Ridge Linear Regression - Tested hyperparameters: 0.1, 1, 10, 50, 100, 200, 500, 1000, 10000, 500000.
- Lasso Linear Regression - Tested hyperparameters: 0.1, 1, 10, 50, 100, 200, 500, 1000, 10000, 500000.
- SVM Linear Regression - Tested kernels: linear, poly and rbf; Tested hyperparameters: 0.0001, 0.01, 0.1, 1, 10, 50.
- Decision tree - maximum depths tested: 3, 5, 6, 7, 8, 9, 10.
- Perceptron Neural Network - Activation functions tested: logistic, tanh and relu; Tested hyperparameters: 0.001, 0.01, 1, 10;

Table 2 presents the evaluation metrics for predicting the Load Factor. The model with the best performance was the neural network with R^2=0.86, using the logistic function and the hyperparameter alpha=10. The method with the worst performance was the decision tree with R^2=0.28. Table 3 presents the evaluation metrics for predicting Maximum Displacement. The model with the best performance was the Ridge linear regression with R^2=0.92, using the hyperparameter alpha=1. The method with the worst performance was the Lasso linear regression with

R^2=0.37. Table 4 presents the evaluation metrics for predicting the final tendon force. The model with the best performance was the Lasso linear regression with R^2=0.93, using the hyperparameter alpha=0.1. The method with the worst performance was the SVM with R^2=0.04.

Table 2. Performance of Load factor model

Model	Best train score (R^2)	Best test score (R^2)	Best param. GRID	Best Grid score (R^2)
Ridge	0,89	0,80	alfa=1	0,85
Lasso	0,89	0,79	alfa=1	0,85
SVM	0,88	0,85	C=50; Kernel = linear	0,85
Decision Tree	0,73	0,02	max_depth=3	0,28
MLP	0,99	0,92	function=logistic; alfa=10	0,86

Table 3. Performance of Maximum displacement model

Model	Best train score (R^2)	Best test score (R^2)	Best param. GRID	Best Grid score (R^2)
Ridge	0,95	0,82	alfa=1	0,92
Lasso	0,41	0,39	alfa=0.1	0,37
SVM	0,84	0,71	C=0.1; Kernel = linear	0,79
Decision Tree	0,99	0,82	max_depth=6	0,83
MLP	0,97	0,78	function=tanh; alfa=0.01	0,91

Table 4. Performance of final tendon force model

Model	Best train score (R^2)	Best test score (R^2)	Best param. GRID	Best Grid score (R^2)
Ridge	0,95	0,76	alfa=0.1	0,92
Lasso	0,95	0,63	alfa=0.1	0,93
SVM	0,12	0,12	C=50; Kernel = linear	0,04
Árvore de decisão	0,99	0,80	max_depth=10	0,89
MLP	0,99	0,79	function=relu; alfa=10	0,75

The method with the highest computational cost while training the model was the neural network method (MLP). Interestingly, of the three models, MLP has a best perform only for load factor prediction. For maximum displacement, the method performed as well as Ridge regression, and for cable force prediction, the method had a high computational cost for the penultimate worst evaluation metric (R^2=0.75). This shows that in this study, it is inefficient to select the neural network model for evaluation, since it has a high computational cost compared to the other methods that also perform well.

4 Conclusions

This work contributes to the development of a database for the analysis of prestressed steel and concrete composite beams considering partial interaction based on a finite element model already developed by partner researchers and validated as a reliable and robust option for the analysis of composite beams.

From the varied database considering different parameters of beam geometry, prestressing and degree of iteration at the interface, the relevance and correlation of these parameters in relation to some structural responses such as load factor at rupture, maximum beam slip at the moment of rupture and final force in the prestressing cable at the end of the analysis were shown. With the database and the target values, 5 machine learning models were trained, obtaining good performance scores. In the end, it was possible to evaluate that the Ridge linear regression method had better efficiency, presenting on average an R^2=0.8 and a lower computational cost when compared to the trained neural network model.

The work had a limitation due to numerical convergence failures in the generation of a database when varying the prestressing force. As a contribution to future works, we suggest to develop a database varying the prestressing force in the cable so that it is possible to predict structural responses using different prestressing cables and prestressing degrees better assist the designs of composite-prestressed beams considering partial iteration.

Acknowledgements. The authors gratefully acknowledge agencies FUNCAP, CNPq and CAPES for the financial support.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] J. Nie, J. Fan, and C. Cai. Stiffness and deflection of steel-concrete composite beams under negative bending. *Journal of Structural Engineering*, vol. 130, n. 11, pp. 1842–1851, 2004.
[2] J. B. M. Sousa Jr., E. Parente Jr, E. M. F. Lima, and M. V. X. Oliveira. Beam-tendon finite elements for post-tensioned steel-concrete composite beams with partial interaction. *J. Construct. Steel Research*, vol. 159, pp. 147–160, 2019.
[3] H. Saadatmanesh, P. Albrecht, and B. M. Ayyub. Experimental study of prestressed composite beams. *J. Struct. Engineering*, vol. 115, n. 9, pp. 2348–2363, 1989.
[4] M. M. R. Almeida, de A. S. C. Souza, de A. T. Albuquerque, and A. Rossi. Parametric analysis of steel-concrete composite beams prestressed with external tendons. *J. Construct. Steel Research*, vol. 189, pp. 107087, 2022.
[5] B. M. Ayyub, Y. G. Sohn, and H. Saadatmanesh. Prestressed composite girders under positive moment. *Journal of structural engineering*, vol. 116, n. 11, pp. 2931–2951, 1990.
[6] S. Chen. Experimental study of prestressed steel–concrete composite beams with external tendons for negative moments. *Journal of Constructional Steel Research*, vol. 61, n. 12, pp. 1613–1630, 2005.
[7] W. Lorenc and E. Kubica. Behavior of composite beams prestressed with external tendons: Experimental study. *J. of Construct. Steel Research*, vol. 62, n. 12, pp. 1353–1366, 2006.
[8] T. Lou and T. L. Karavasilis. Numerical assessment of the nonlinear behavior of continuous prestressed steel-concrete composite beams. *Engineering Structures*, vol. 190, pp. 116–127, 2019.
[9] A. A. El-Sisi, A. I. Hassanin, H. F. Shabaan, and A. I. Elsheikh. Effect of external post-tensioning on steel–concrete composite beams with partial connection. *Engineering Structures*, vol. 247, pp. 113130, 2021.
[10] H.-T. Thai. Machine learning for structural engineering: A state-of-the-art review. In *Structures*, volume 38, pp. 448–491. Elsevier, 2022.
[11] Z. Tadesse, K. Patel, S. Chaudhary, and A. Nagpal. Neural networks for prediction of deflection in composite bridges. *Journal of Constructional Steel Research*, vol. 68, n. 1, pp. 138–149, 2012.
[12] R. Gupta, S. Kumar, K. Patel, S. Chaudhary, and A. Nagpal. Rapid prediction of deflections in multi-span continuous composite bridges using neural networks. *Int. J. Steel Struct.*, vol. 15, pp. 893–909, 2015.
[13] M. A. Sakr and S. S. S. Sakla. Long-term deflection of cracked composite beams with nonlinear partial shear interaction — a study using neural networks. *Engineering Structures*, vol. 31, pp. 2988–2997, 2009.
[14] M. V. X. Oliveira. Análise numérica de vigas mistas protendidas. 2017. 84 f. Mestrado em engenharia civil, PPG em Engenharia Civil: Estruturas e Construção Civil, UFC, 2017.
[15] E. M. F. Lima. Análise numérica de vigas mistas protendidas com interação parcial. 2018. 96 f. Mestrado em engenharia civil, PPG em Engenharia Civil: Estruturas e Construção Civil, UFC, 2018.
[16] C. Cortes and V. Vapnik. Support-vector networks. *Mach Learn*, pp. 273–297, 1995.
[17] GERDAU. Perfis estruturais gerdau: Tabela de comparações entre perfis, 2022.
[18] B. B. Arames. Fios e cordoalhas para concreto protendido, 2022.
[19] *NBR 8800:2008*. ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2008.