

Article

Torsional Capacity Prediction of Reinforced Concrete Beams Using Machine Learning Techniques Based on Ensembles of Trees

Diana S. O. Bernardo ¹, Luís F. A. Bernardo ^{2,*} , Hamza Imran ³  and Tiago P. Ribeiro ^{2,4} ¹ Instituto Superior Técnico, University of Lisbon, 1049-001 Lisbon, Portugal² Department of Civil Engineering and Architecture, University of Beira Interior, 6201-001 Covilhã, Portugal³ Department of Environmental Science, College of Energy and Environmental Science, Alkarkh University of Science, Baghdad 10081, Iraq⁴ Tal Projecto Lda., 1350-252 Lisbon, Portugal

* Correspondence: lfb@ubi.pt

Abstract: For the design or assessment of framed concrete structures under high eccentric loadings, the accurate prediction of the torsional capacity of reinforced concrete (RC) beams can be critical. Unfortunately, traditional semi-empirical equations still fail to accurately estimate the torsional capacity of RC beams, namely for over-reinforced and high-strength RC beams. This drawback can be solved by developing accurate Machine Learning (ML) based models as an alternative to other more complex and computationally demanding models. This goal has been herein addressed by employing several ML techniques and by validating their predictions. The novelty of the present article lies in the successful implementation of ML methods based on Ensembles of Trees (ET) for the prediction of the torsional capacity of RC beams. A dataset incorporating 202 reference RC beams with varying design attributes was divided into testing and training sets. Only three input features were considered, namely the concrete area (area enclosed within the outer perimeter of the cross-section), the concrete compressive strength and the reinforcement factor (which accounts for the ratio between the yielding forces of both the longitudinal and transverse reinforcements). The predictions from the used models were statistically compared to the experimental data to evaluate their performances. The results showed that ET reach higher accuracies than a simple Decision Tree (DT). In particular, The Bagging Meta-Estimator (BME), the Forests of Randomized Trees (FRT), the AdaBoost (AB) and the Gradient Tree Boosting (GTB) reached good performances. For instance, they reached values of R^2 (coefficient of determination) in the range between 0.982 and 0.990, and values of $cvRMSE$ (coefficient of variation of the root mean squared error) in the range between 10.04% and 13.92%. From the obtained results, it is shown that these ML techniques provide a high capability for the prediction of the torsional capacity of RC beams, at the same level of other more complicated ML techniques and with much fewer input features.



Citation: Bernardo, D.S.O.; Bernardo, L.F.A.; Imran, H.; Ribeiro, T.P. Torsional Capacity Prediction of Reinforced Concrete Beams Using Machine Learning Techniques Based on Ensembles of Trees. *Appl. Sci.* **2023**, *13*, 1385. <https://doi.org/10.3390/app13031385>

Academic Editor: Ricardo Branco

Received: 28 December 2022

Revised: 12 January 2023

Accepted: 17 January 2023

Published: 20 January 2023

Keywords: machine learning; reinforced concrete beams; torsional capacity; Decision Tree; Bagging; Random Forests; Adaptive Boosting; Gradient Boosting



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Reinforced concrete (RC) structures often incorporate linear members for which the torsional effects can be critical for designing or for assessing the actual cross-sectional resistance. This situation is very common in bridges and building structures which incorporate beams and columns under high eccentric loadings, or any structural system whose equilibrium depends on reinforced concrete elements resisting torsion. For the assessment of the capacity of such members, an accurate estimation of the torsional capacity in their critical cross-sections is usually required. To do so, structural designers often apply the provisions from codes of practice, which incorporate simplified models and semi-empirical

equations, allowing to estimate the resistance to internal forces for a given RC cross-section. As far as torsion is concerned, and unlike what is observed for the bending capacity, recent studies have reported that current codes of practice can still fail to estimate the torsional capacity of RC beams. This conclusion arises from comparing the estimates from codes and the results from experiments [1–3]. For RC beams with somewhat high reinforcement ratios, unsafe estimates for the torsional capacity were also reported. This can explain why structural failures related to primary torsional effects still continue to be reported in the literature [4].

The aforementioned problem has been well-known by the technical community for a long time. For this reason, efforts have been made in the past decades to propose alternative and more accurate models for RC members under torsion. One such model is the skew-bending theory, proposed by Hsu in 1968 [5] from the observation of the experimental torsional failure pattern in tested RC beams, and further developed by other authors [6]. For common and small RC rectangular cross-sections, simple equations were derived from this model to compute the torsional capacity, which were incorporated in some reference codes of practice, namely the American building code prior to 1995 [7] and the current Eurasian code [8].

Nowadays, most of current reference codes of practice for RC structures (such as the American, Canadian and European codes [9–12]) base their provision rules for torsion on the space truss analogy. This more rational model was proposed by Raush in 1929 [13] and further refined by other authors mainly in the second half of the last century [14–17] based on the results of extensive experimental programs. When compared with the skew-bending theory, the space truss analogy provides simpler equations for the design of RC beams under torsion and for members with a much wider range of geometrical cross-sections. With the aim of obtaining simple rules for design to be incorporated in the provisions from codes of practice, the original equations from both the skew-bending theory and space truss analogy have been simplified by using some empirical hypotheses. As a result, codes of practice incorporate many semi-empirical equations, which can still fail to provide always accurate and safe estimates for the torsional capacity of RC beams, as referred before. Hence, more refined or alternative models for RC beams under torsion are still needed and have been proposed in the past two decades.

Among the refined models proposed in the literature, some of them are still based on the space truss analogy. Some of these refined models allow to predict very well the torsional capacity of RC beams with a wide range of design attributes and torsional loadings [18–23]. Alternative and more advanced analytical or numerical reliable models, including models based on the finite element method, have also been proposed [24–26]. However, many of these models need advanced calculation procedures and demanding computational efforts just to compute the actual torsional capacity of RC beams. Hence, they cannot be easy to use by most of practitioners.

In recent years, reliable models based on several Machine Learning (ML) techniques have been proposed in the literature to solve diverse engineering problems, as alternatives to other and more complex analytical or numerical models. These models are able to provide predictions through a self-learning process based on collected or existing databases. In parallel, during the last decades, a huge amount of data has been accumulated in the field of civil engineering due to the advance in experimental programs, monitoring, data acquisition, and processing. For these reasons, ML-based models have been successfully used in complex civil engineering problems, including geotechnical, materials and structural [27–39], and some of them already focus on the torsional capacity of RC beams, including externally strengthened and combined RC beams [40–48]. The referred studies on the torsional capacity of RC beams are still very limited and they generally apply different ML techniques. This is due to the fact that several ML techniques have been proposed in the literature, which are able to be used to solve problems in the field of civil engineering, including for structural concrete [46].

Despite the successes achieved with the application of ML approaches in several previous studies, some key challenges still exist which prevent these models from being widely used to solve engineering problems. Perhaps the main key challenge is related with the selection process of the most appropriate and effective ML models to solve a particular civil engineering problem. For instance, some studies indicated that the best performances were reached with extreme gradient boosting (XGBoost) for problems involving the shear strength of RC beams [49,50], whereas others have mentioned other ML models to be more performant for the same types of problems [41–55], namely Artificial Neural Networks (ANN). This includes the analysis of RC beams under torsion, where the shear effect is predominant [40,41,43–46,48]. Ensembles of Trees (ET) have also been used in such problems, although to a lesser extent and with limited success [44]. Explicit ML techniques, such as the M5P model tree [47], were also successfully used. The problem related with the selection process of the most appropriate effective ML model for RC beams under torsion needs more studies to compare the performance of different ML models. Particularizing for RC beams under torsion, another key challenge is related to the number of features to be considered as input variables for developing the ML models. The more selected number of input variables, the higher the prediction accuracy. However, the more complex the algorithm and the larger the computational cost. Hence, a compromise must be reached for practice.

Decision Trees (DT) are among the most commonly used ML models in very different problems, and they show several distinct advantages. Many algorithms require data normalization before model building. Such variable transformations are not required with DT since the tree structure will remain the same despite the transformation. They also implicitly perform feature selection and are not sensitive to outliers. In addition, DT usually require less input variables when compared to other ML models, such as ANN, to achieve similar performances. For these reasons, DT were used as ML-based models for this study.

This study aims to contribute to compare and find the best ML techniques to be applied to RC beams under pure torsion. Although ET can be considered some of the most effective and popular ML algorithms, an extensive examination of the literature indicates that studies have yet to successfully use this approach to predict the torsional capacity of RC beams. Therefore, an attempt has been made to investigate this potentiality, and five ET models were developed to study the feasibility of applying these techniques for the quick estimation of the torsional capacity of RC beams. To the best of the authors' knowledge, only one previous study used two such techniques, but applied to externally bonded FRP (Fiber Reinforced Polymer) RC Beams under torsion [44]. A dataset including 202 RC beams tested under pure torsion and found in the literature was used for both the training and testing stages. Three input variables were used: the cross-section area (area enclosed within the outer perimeter of the cross-section), the concrete compressive strength, and the reinforcement factor (factor accounting for the ratio between the yielding forces of both the longitudinal and transverse reinforcements). The predictions from the five ML models are statistically compared to each other, to the experimental data, and to other ML models, to evaluate their performance and reliability.

This article was structured in the following manner. Section 2 gives an overview of the used methodology, the dataset, the input and output variables, the splitting of the dataset in training and testing sets, the used ML models, the cross-validation, the hyperparameter tuning procedures, the metrics used to evaluate the performance of the models, and the used programming languages and software to implement the models. Section 3 presents the results of the models and Section 4 discusses the comparison of these results with the ones obtained in previous studies. In Section 5, a sensitivity analysis is performed to check the importance of each input variable in the prediction of the torsional capacity of RC beams. Finally, Section 6 summarizes the main conclusions.

2. Materials and Methods

2.1. Research Methodology

The dataset was randomly divided into two sets: one for training and another for testing. Each model was trained using the training set and was tested using the testing set. The appropriate hyperparameters of each model were determined by using a five-fold cross-validation procedure in the training set. After the hyperparameters were optimized, the testing set was used to verify the performance of each model. If the performance of the model is satisfactory, it is deemed as a final predictive model. The steps used in this study are depicted in the flowchart illustrated in Figure 1 (adapted from [30]).

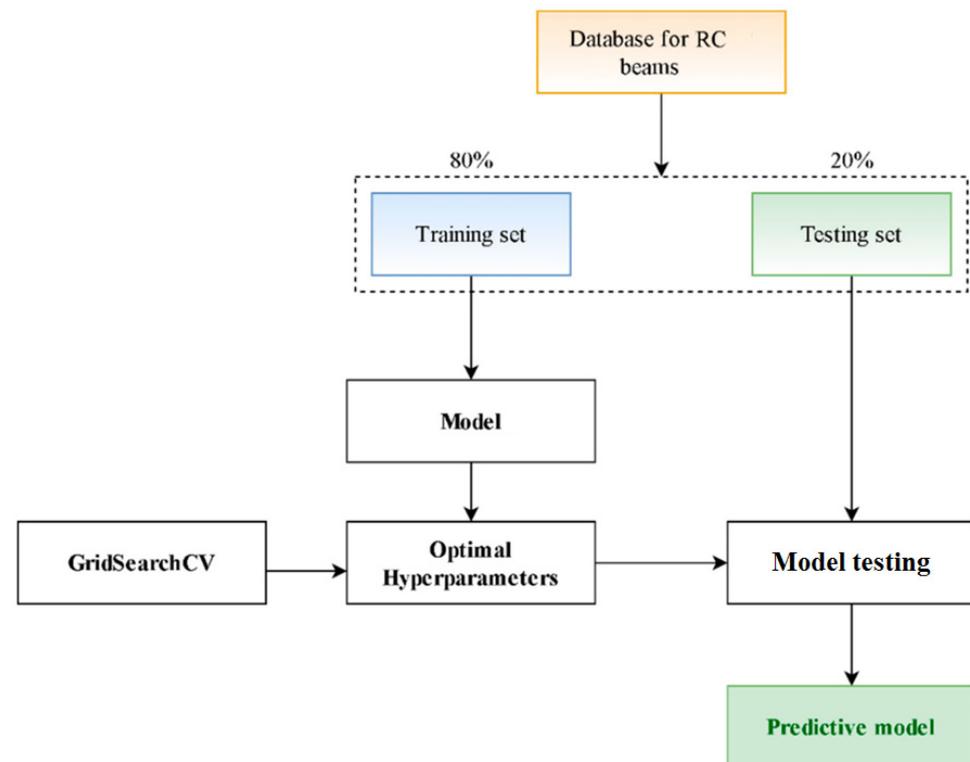


Figure 1. Flowchart with the research methodology.

2.2. Dataset

For this study, the dataset compiled by Bernardo et al. [3] and also used by Henedy et al. [47] was considered. In the former study, a literature review was performed to compile the main features and torsional capacity of RC rectangular beams tested under pure torsion until failure [5,6,17,56–68]. In total, 202 RC beams were compiled to build the dataset. Detailed information about the criteria used to discard some of the found tested beams for the dataset can be found in [3].

Table A1 in Appendix A summarizes the main geometrical and mechanical properties of the reference RC beams. The meaning of each parameter can be found in the nomenclature. Figure 2 presents the histograms with the distribution of the considered main key parameters in this study for the 202 RC beams from the dataset. From the analysis of the data presented in Table A1 and from a visual analysis from Figure 2, it can be stated that 142 and 60 beams were built with normal-strength (up to 50 MPa) and high-strength concrete (over 50 MPa, according to [11]), respectively. The average concrete compressive strength (f_c) ranges between 14 MPa and 110 MPa. The total reinforcement ratio, which represents the sum of the longitudinal plus the transverse reinforcement ratio ($\rho_{tot} = \rho_l + \rho_t$), ranges between a minimum of 0.37% and a maximum of 6.36%. For most beams, ρ_{tot} ranges between 1% and 2%. The yielding stress of the longitudinal reinforcement (f_{ly}) ranges between 308.8 MPa and 723.9 MPa. The yielding stress of the transverse reinforcement

(f_{ty}) ranges between 285 MPa and 714.8 MPa. For most beams, the yielding stress ranges between 300 MPa and 500 MPa.

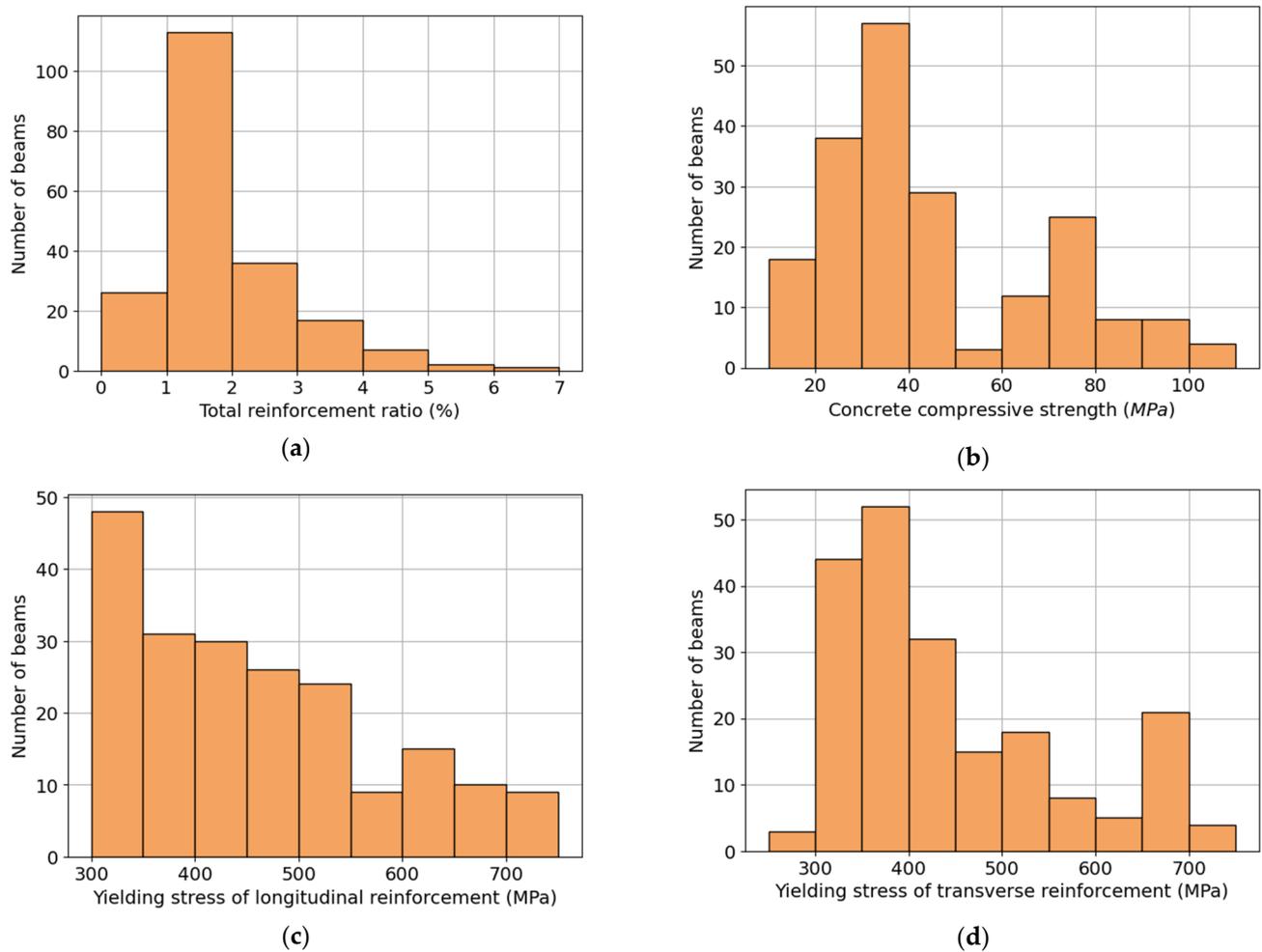


Figure 2. Histograms of key parameters for the RC beams: (a) total reinforcement ratio, (b) concrete compressive strength, (c) yielding stress of longitudinal reinforcement, (d) yielding stress of transverse reinforcement.

2.3. Input and Output Variables

Based on results from previous studies [3,47], the following three input variables were used to characterize each RC beam and predict the torsional capacity T_{R_Exp} : the cross-section area A_c (area enclosed within the outer perimeter of the cross-section), the average concrete compressive strength f_c , and the reinforcement factor $A_l f_{ly} \frac{A_t f_{ty}}{s}$ (factor accounting for the ratio between the yielding forces of both the longitudinal and transverse reinforcements). To compute the last input variable, the units of each parameter are the same as the ones given in Table A1 in Appendix A. The reinforcement factor combines both the ratio of the longitudinal to the transverse torsional reinforcement, as well as the ratio of their yielding stresses. Hence, it can be used to characterize beams with balanced (which is a usual design criterion for pure torsion) or unbalanced (for which the reinforcement factor is different from unity) yielding forces of the longitudinal and transverse reinforcements. For this reason, the authors considered that there is no need to consider separately the effects of longitudinal and transverse reinforcement in developing the ML models in this study, as it was carried out in previous studies using ANN ML models (for instance, [40,41,43,48]) which usually require more input variables. The reinforcement factor was firstly proposed by Rahal [1] as an input variable and further used successfully in recent studies from the

authors [3,47] using nonlinear regression and MP5 models to predict the torsional capacity of RC beams under pure torsion.

Figure 3 presents the histograms of the input variables (the concrete compressive strength was added again), as well as the output variable (torsional capacity), for the 202 reference RC beams from the database. Figure 3 shows that for most of the beams, the reinforcement factor ranges between 1.1×10^6 and 2×10^8 , and the torsional capacity ranges between 9.0 kNm and 200 kNm.

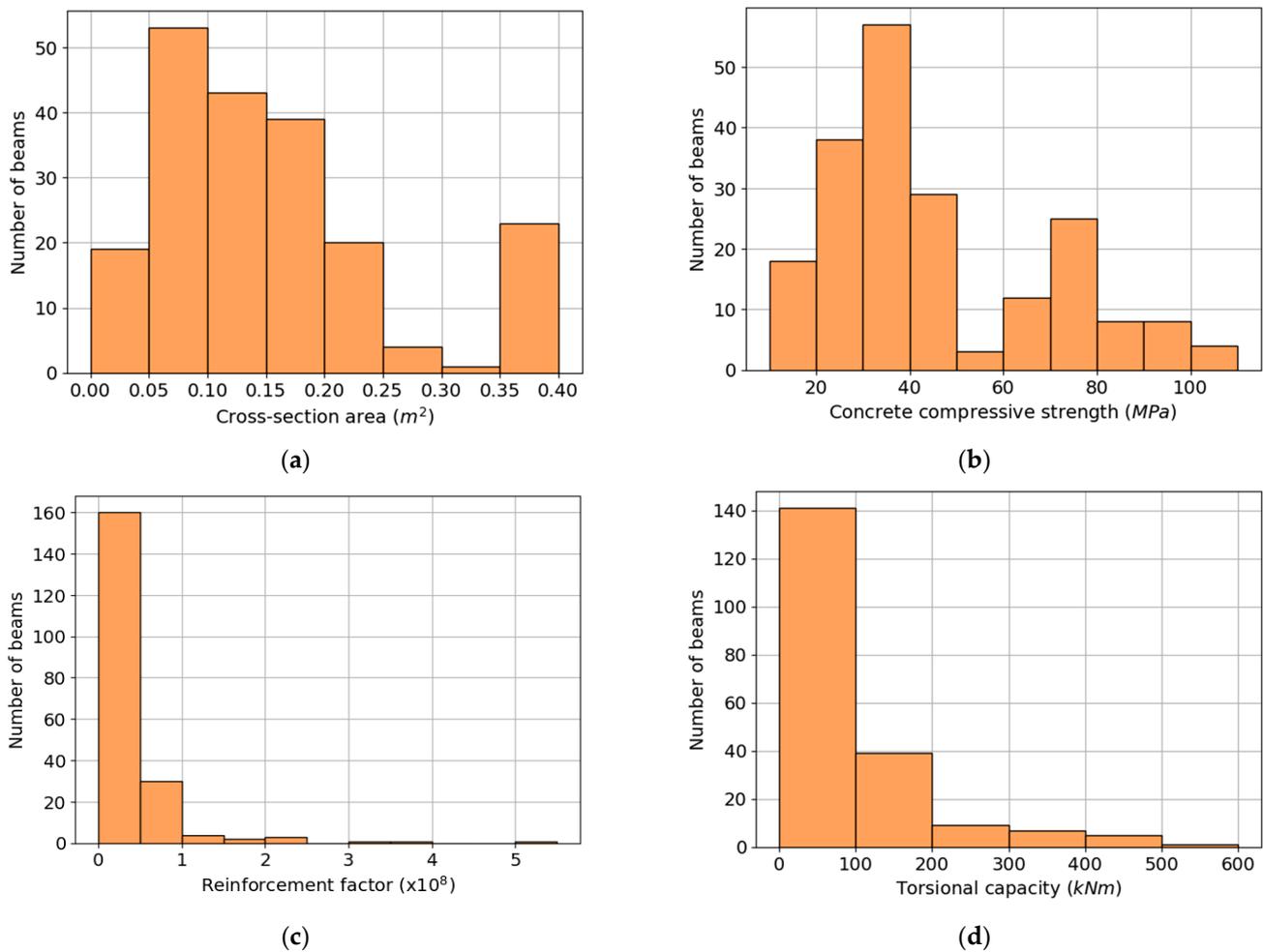


Figure 3. Histograms of input and output variables: (a) cross-section area, (b) concrete compressive strength, (c) reinforcement factor, and (d) torsional capacity.

As it can be noticed from Figures 2 and 3, some of the predictors and outcome variables do not obey the normal distribution curve. However, DT and ET methods do not require feature scaling to be performed as they are not sensitive to the variance in the data [55], as previously mentioned in Section 1. For this reason, data normalization was not performed in this study.

Following a previous study by Wakjira et al. [32], the correlation between each pair of parameters is also important to be analyzed. Figure 4 and Table 1 show the scatter plot and the Pearson correlation coefficient between pairs of the inputs and the output variables, respectively. Both Figure 4 and Table 1 show that there is a strong correlation between some of the input variables, namely the cross-section area and the reinforcement factor, and the output, whereas there is a moderate correlation between the concrete compressive strength and the output. In general, low degrees of correlation exist between pairs of input variables, indicating that these will not cause multicollinearity problems on the models [69]. A high absolute value of the Pearson correlation coefficient between pairs of inputs could

affect the accuracy of the model and the interpretation of the effects of the inputs on the output [33].

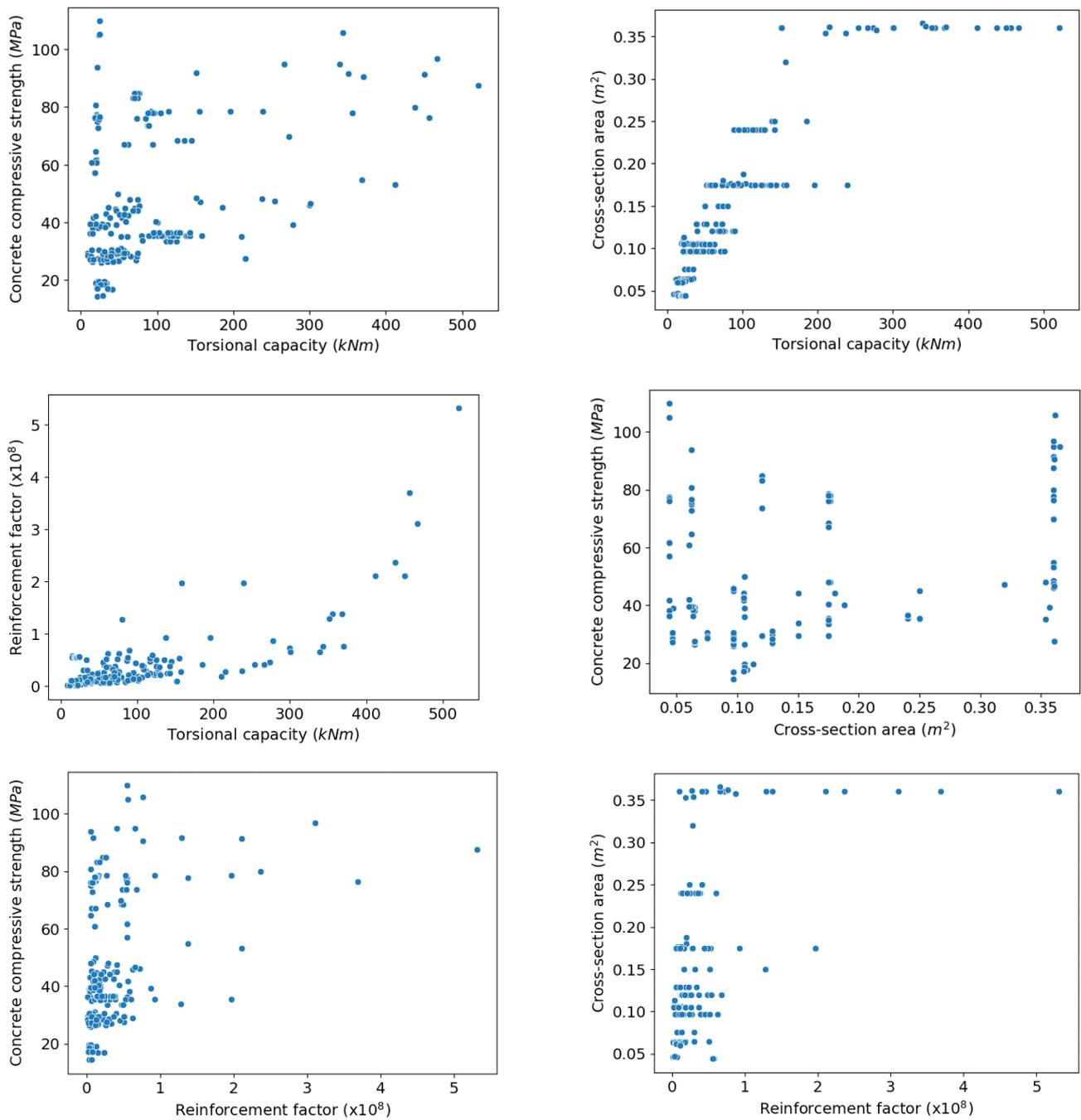


Figure 4. Scatter plot between pairs of variables.

Table 1. Pearson correlation coefficient.

	T_{R_Exp}	f_c	A_c	$A_l f_{ly} \frac{A_l f_{ly}}{s}$
T_{R_Exp}	-	0.4168	0.8801	0.7644
f_c	0.4168	-	0.2461	0.3637
A_c	0.8801	0.2462	-	0.4647
$A_l f_{ly} \frac{A_l f_{ly}}{s}$	0.7644	0.3637	0.4647	-

2.4. Data-Splitting Procedure

To implement the ML models, the database was divided into a training set and a testing set. Each model was developed using the training set and evaluated using the testing set. As much as possible, a statistically significant association was ensured between the variables of the training and testing sets when dividing the database into subsets. Of the 202 samples of the database, 80% (161) was used for training and 20% (41) was used for testing. This division was made based on the ratio of the data-splitting procedure commonly used in several previous related studies. In particular, in [70] the authors justified the use of 70/30 or 80/20 splitting procedures. The statistical measures of the torsional capacity and input variables are given in Tables 2 and 3, for the training and testing set, respectively.

Table 2. Statistical measures of the variables for the training set.

Statistics	T_{R_Exp} (kNm)	f_c (MPa)	A_c (m ²)	$A_t f_{ty} \frac{A_t f_{ty}}{s}$
Mean	88.7	46.7	0.153	3.5×10^7
Minimum	9.0	14.3	0.044	1.1×10^6
Maximum	521.3	109.8	0.366	5.3×10^8
Standard deviation	96.9	23.0	0.094	6.2×10^7
Median	57.0	36.5	0.120	1.7×10^7
Kurtosis	6.0	−0.2	−0.029	32.3
Skewness	2.4	0.9	0.958	5.1

Table 3. Statistical measures of the variables for the testing set.

Statistics	T_{R_Exp} (kNm)	f_c (MPa)	A_c (m ²)	$A_t f_{ty} \frac{A_t f_{ty}}{s}$
Mean	105.8	46.6	0.156	4.0×10^7
Minimum	9.1	16.8	0.044	1.2×10^6
Maximum	437.9	93.9	0.361	2.4×10^8
Standard deviation	109.9	22.7	0.098	5.3×10^7
Median	61.6	39.6	0.120	2.5×10^7
Kurtosis	2.5	−0.9	0.309	7.3
Skewness	1.8	0.6	1.159	2.7

2.5. Development of Models

In this article, five ML models were implemented using the same database and inputs. The first model implemented was a simple Decision Tree (DT) regressor. A DT can be unstable since small variations in the data might result in a completely different generated tree. A single tree usually exhibits high variance and tends to overfit, not generalizing well to new data [71]. This problem is attenuated by using several trees within an ensemble. Ensemble methods seek to combine the predictions of several base estimators to improve generalizability, accuracy and robustness over a single estimator. The other four models implemented in this study involve Ensemble Tree (ET) methodologies. For the sake of the readers, the following subsections briefly present an overview of each implemented method. Training data exist in the form of a training set $\{(x_i, y_i)\}_{i=1}^n$, in which $x_i \in \mathbb{R}^p$ represents the input features and $y_i \in \mathbb{R}$ represents the torsional capacity for p inputs and n samples.

2.5.1. Decision Trees

Decision Trees (DT) predict the value of a variable by learning decision rules inferred from the features. The deeper the tree, the more complex the decision rules and the fitter the model. DT require little data preparation when compared to other techniques that require

data normalization and other transformations. It also uses a white box model, allowing an easy explanation of the conditions by Boolean logic. However, DT can create over-complex trees that overfit and do not generalize well to the data [71]. This problem was avoided by setting a maximum depth for the trees, as referred in [71]. The input data is recursively partitioned into two smaller regions called nodes, based on a series of decision rules until a stopping criterion is reached (in this case, the maximum depth of the tree). Samples with similar target values are grouped together, and then the response is modelled by the mean of the y variable in each region. A tree can be seen as a piecewise constant approximation. The variable and split point are chosen in order to achieve the best fit. Let the data at node m be represented by R_m with n_m samples. For each candidate split $\theta = (j, t_m)$ consisting of a feature j and a threshold t_m , the data is partitioned into $R_m^{left}(\theta)$ and $R_m^{right}(\theta)$ subsets such that:

$$R_m^{left}(\theta) = \{(x, y) \mid x_j \leq t_m\} \tag{1}$$

$$R_m^{right}(\theta) = R_m \setminus R_m^{left}(\theta) \tag{2}$$

The quality of a candidate split of node m is then computed using a loss function H :

$$G(R_m, \theta) = \frac{n_m^{left}}{n_m} H\left(R_m^{left}(\theta)\right) + \frac{n_m^{right}}{n_m} H\left(R_m^{right}(\theta)\right) \tag{3}$$

The parameters are selected in order to minimize the loss function:

$$\theta^* = \arg \min_{\theta} G(R_m, \theta) \tag{4}$$

This is carried out recursively for the two subsets until the maximum depth is reached.

2.5.2. Bagging Meta-Estimator

A Bagging Meta-Estimator (BME) belongs to the set of Ensemble Methods, specifically to the family of averaging methods, in which several estimators are built independently and then their predictions are averaged to form a final prediction. Averaging methods are used to improve stability and reduce the variance of the base estimator by introducing randomization into its construction procedure. As they provide a way to reduce overfitting, these methods work best with strong and complex models, in this case, fully developed trees. BME uses a bootstrapping technique to build each individual estimator on random subsets of samples that are repeatedly drawn from the original training dataset with replacement [71]. A model is fitted for each bootstrap sample, giving a prediction $\hat{f}^b(x)$ for $b = 1, \dots, B$. The final estimate is then given by:

$$\hat{f}(x) = \frac{1}{B} \sum_{b=1}^B \hat{f}^b(x) \tag{5}$$

2.5.3. Forests of Randomized Trees

Forests of Randomized Trees (FRT) are also an averaging methods. The prediction of the ensemble is given as the averaged prediction of the individual regressors. Each tree is built from a sample drawn with replacement from the training set. The sources of randomness decrease the variance of the estimator by reducing the correlation between trees. In this algorithm, only one subset of features is randomly selected out of the total, and the best split feature from the subset is used to split each node in a tree, unlike in Bagging, where all features are considered for splitting a node [71].

2.5.4. Adaptive Boosting

Adaptive Boosting or AdaBoost (AB) belongs to the set of Ensemble Methods, specifically to the family of boosting methods, in which the base estimators are built sequentially, and the goal is to reduce the bias of the combined estimator. The motivation is to combine

several weak models to generate a powerful ensemble. In contrast with averaging methods, boosting methods usually work best with weak models, namely shallow trees in this case. A sequence of weak learners is fitted on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted sum to produce the final prediction. The data modifications at each boosting step consist of applying weights $\omega_1, \omega_2, \dots, \omega_N$ to each of the training samples (x_i, y_i) [71]. Initially, all the weights are all set to $\omega_i = 1/N$, so that the first step simply trains a weak learner in the usual manner. For each successive iteration, the sample weights are individually modified, and the algorithm is reapplied to the weighted data. The training examples that were incorrectly predicted by the model induced at the previous step have their weights increased, whereas the weights are decreased for the ones predicted correctly. As iterations proceed, examples that are difficult to predict receive an increasing influence. Each subsequent weak learner is forced to concentrate on the examples that were missed by the previous ones in the sequence.

2.5.5. Gradient Tree Boosting

Gradient Tree Boosting (GTB) is also a boosting method. It builds an additive model in a forward stage-wise fashion allowing the optimization of a differentiable loss function. The existing residuals are used to build new trees sequentially. In each stage, a regression tree is added and fitted on the negative gradient of the loss function, reducing its value and improving the prediction. The prediction \hat{y}_i for a given input x_i is of the form:

$$\hat{y}_i = F_M(x_i) = \sum_{m=1}^M h_m(x_i) \tag{6}$$

where h_m are the weak learners. The constant M corresponds to the number of estimators. GTB is built as follows:

$$F_m(x) = F_{m-1}(x) + h_m(x) \tag{7}$$

where the added tree h_m is fitted in order to minimize a sum of losses L_m , given the previous ensemble F_{m-1} :

$$h_m = \arg \min_h L_m = \arg \min_h \sum_{i=1}^n l(y_i, F_{m-1}(x_i) + h(x_i)) \tag{8}$$

where $l[y_i, F_{m-1}]$ is the loss function measuring how much the predicted value $F(x)$ differs from the true value y . Trees will be built, and each iteration must satisfy the above equation. The initial model F_0 is chosen as the constant that minimizes the loss. Using a first-order Taylor approximation, the value of l can be approximated as:

$$l(y_i, F_{m-1}(x_i) + h_m(x_i)) \approx l(y_i, F_{m-1}(x_i)) + h_m(x_i) \left[\frac{\partial l(y_i, F(x_i))}{\partial F(x_i)} \right]_{F=F_{m-1}} \tag{9}$$

Removing the constant terms,

$$h_m \approx \arg \min_h \sum_{i=1}^n h(x_i) \tag{10}$$

This is minimized if $h(x_i)$ is fitted to predict a value that is proportional to the negative gradient. Therefore, at each iteration, the estimator h_m is fitted to predict the negative gradients of the samples. The gradients are updated at each iteration. This can be considered as a form of gradient descent in a functional space.

Regularization techniques are usually applied during training to reduce overfitting and improve the generalization of the model. A regularization strategy that scales the contribution of each weak learner by a constant factor ν is the following [72]:

$$F_m(x) = F_{m-1}(x) + \nu h_m(x) \tag{11}$$

The parameter ν is called the learning rate because it scales the step length of the gradient descent procedure.

Stochastic Gradient Boosting combines gradient boosting with bootstrap averaging (Bagging) [73]. At each iteration the base estimator is trained on a fraction subsample of the available training data. The subsample is drawn without replacement.

2.6. Cross-Validation

A standard approach for evaluating the performance of ML models with hyperparameter tuning is to divide the dataset into three subsets: training, validation, and testing. The training set is used for the learning process and the evaluation of the performance of the model is done on the validation set. After the best parameters are found, the final evaluation is performed on the test set, with samples that the model has never seen before. Note that the validation score is biased and to obtain a proper estimate of the generalization the score needs to be computed on the test set. However, dividing the data into three subsets reduces the number of samples that can be used for learning, which might result in an inadequately trained model. Cross-validation (CV) [71] is a widely used strategy for avoiding over-reduction of the training set, particularly for small datasets. A test set should still be held out for final evaluation, but the validation set is no longer needed. In this research, k -fold CV was used. This consists of splitting the training set into k smaller sets. Then, for each of the k folds, the model is trained using $k - 1$ of these folds and then it is validated on the remaining fold. The final performance measure is the average of the k performance values computed in this loop. Figure 5 (adapted from [74]) depicts the five-fold CV used in this research for training and also for the hyperparameter selection.

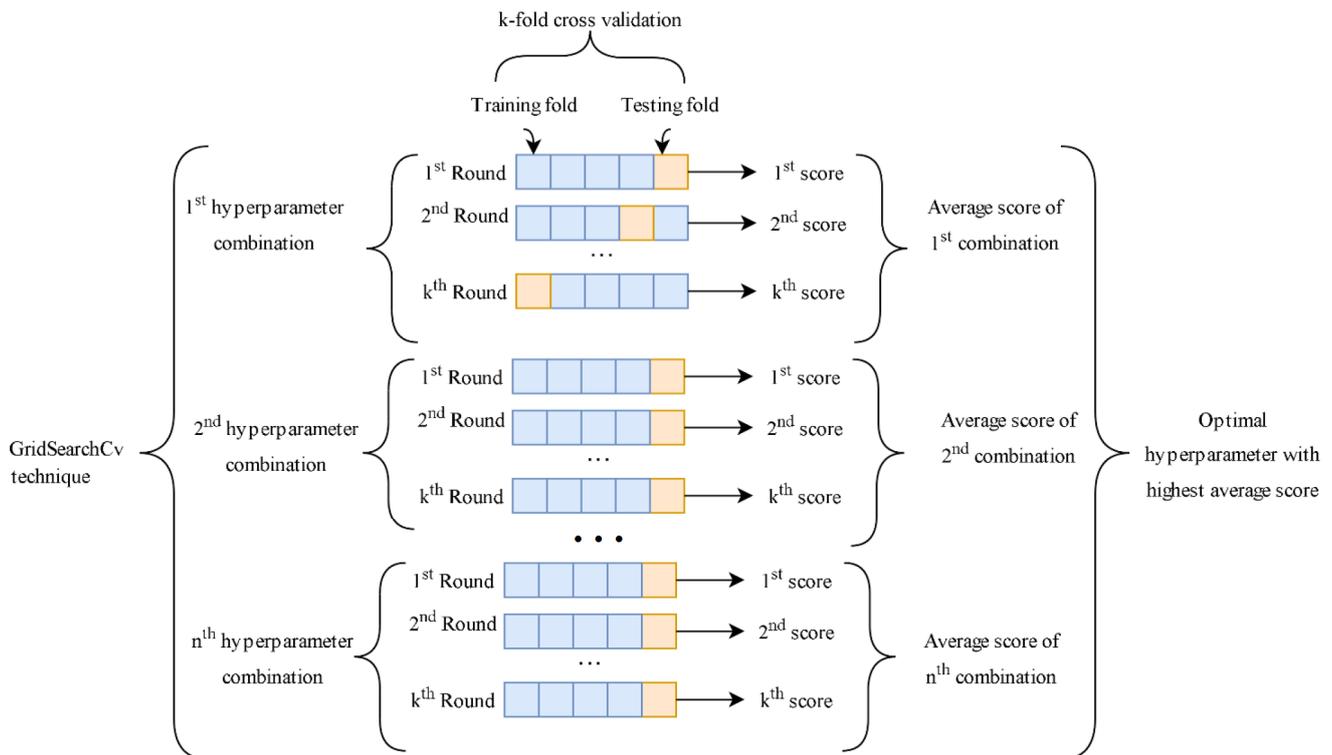


Figure 5. Hyperparameter tuning using five-fold cross validation.

2.7. Hyperparameter Tuning

The tuning of hyperparameters is essential to select the optimal values that improve the performance of the model. One method of automated hyperparameter selection is the grid search technique, which investigates all possible hyperparameter values in a pre-defined domain [75]. The hyperparameters are optimized by a cross-validated grid

search over a parameter grid, considering all possible combinations. It selects the set of hyperparameters with the maximum score on a validation set. At last, a final evaluation of the model is performed in the test set, allowing the understanding of how the model is performing on unseen data.

The proper manner of choosing multiple hyperparameters of an estimator is the grid search. However, grid search techniques can come at the cost of great computing time when having several parameters, with dozens or even hundreds of values each. It is sometimes helpful to plot the influence of a single hyperparameter on the training and validation scores to find if the estimator is overfitting or underfitting for some hyperparameter values.

2.8. Model Performance and Uncertainty Metrics

Various statistical measures were used to evaluate the performance of the ML models. The mean absolute error (*MAE*) is the averaged absolute difference between the actual and the predicted values. The equation for the *MAE* is the following:

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (12)$$

where y_i and \hat{y}_i are the actual and the predicted value of the i -th sample, respectively, for a total of n samples.

The root-mean-squared error (*RMSE*) is calculated as the square root of the average squared errors. The *RMSE* is computed as follows:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (13)$$

The coefficient of variation of the root mean squared error (*cvRMSE*), or the scatter index (*SI*), is the ratio of the *RMSE* to the mean of the actual values $\bar{y} = \sum_{i=1}^n y_i$. It represents the percentage of *RMSE* with respect to the mean of observations and gives the expected error. It can be calculated as follows:

$$cvRMSE = \frac{RMSE}{\bar{y}} \quad (14)$$

The coefficient of determination (R^2) represents the proportion of variation in the dependent variable that is predicted by the independent variables in the model. It is an indicator of goodness of fit and a measure of how well the model predicts the outcome. When the R^2 value is 1, the predicted and the true values are perfectly aligned. R^2 has the following mathematical representation:

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

The mean absolute percentage error (*MAPE*) is an evaluation metric sensitive to relative errors. The smaller its value, the better. It is given by:

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

The root-relative-squared error (*RRSE*) is the square root of the sum of squared errors of a predictive model normalized by the sum of squared errors of a simple model. The lower its value, the better the model. It can be calculated as:

$$RRSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (17)$$

Another performance metric used in this study is the variance accounted for (VAF), with the best possible score of 1. The following equation was used to compute this metric.

$$VAF = 1 - \frac{VAR(y_i - \hat{y}_i)}{VAR(y_i)} \quad (18)$$

The model uncertainty is defined as the inability of the model to effectively express the torsional capacity. In this research, the predictive model uncertainty related to beam i is equal to the ratio between the experimental and the predicted torsional capacity:

$$M_i = \frac{T_{R_Exp_i}}{T_{R_Pred_i}} \quad (19)$$

where M_i is the model uncertainty of the i -th beam sample. The mean and standard deviation of the model uncertainty are represented by σ_M and μ_M , respectively. It is better to obtain a model with a mean μ_M close to 1 and a standard deviation σ_M close to 0.

2.9. Programming Languages and Software

The *Python* programming language combined with the *Scikit-learn* library was used to build the models for the torsional capacity prediction. *Scikit-learn* [76] is an open-source ML library in *Python*, containing many algorithms and methods, such as classification, clustering, and regression, in addition to being used in data processing and model evaluation.

3. Model Results and Discussion

3.1. Parameters Description of the Regression Functions

For the Decision Tree (DT), the module *sklearn.tree* provides the function *DecisionTreeRegressor*. This function supports four different criteria to measure the quality of a split via the parameter *criterion*: the MSE, the MSE with improvement score by Friedman, the MAE and the Poisson deviance. The size of the regression tree can be controlled by specifying the parameter *max_depth*, which limits the number of nodes in the tree. The maximum number of features to consider when looking for the best split is given by the parameter *max_features*.

For the Bagging Meta-Estimator (BME), the module *sklearn.ensemble* provides the function *BaggingRegressor*, taking as input an estimator along with parameters specifying the strategy to draw random subsets. The base estimator used was a DT regressor. The parameter *n_estimators* is the number of base estimators in the ensemble. The parameters *max_samples* and *max_features* control the size of the subsets in terms of samples and features to train each base estimator, while *bootstrap* and *bootstrap_features* control whether samples and features are drawn with or without replacement.

For the Forests of Randomized Trees (FRT), the module *sklearn.ensemble* provides the function *RandomForestRegressor*. The parameter *n_estimators* is the number of trees in the forest. This function also supports the same four criteria to measure the quality of a split via the parameter *criterion* as the *DecisionTreeRegressor*. When building trees, the best node split can be found either from all input features or a random subset of size *max_features*. Bootstrap samples are used by default and the fraction of samples to train each base estimator is given by the parameter *max_samples*.

For the AdaBoost (AB), the module *sklearn.ensemble* provides the function *AdaBoostRegressor*, taking as input a user-specified estimator from which the ensemble is built along with other parameters. The base estimator used was a DT regressor. The learning rate can be set via the parameter *learning_rate*, which controls the contribution of each tree in the final combination. This strongly interacts with the parameter *n_estimators*. Empirical evidence suggests that small values of *learning_rate* favor better test error and lead to better model generalization [71]. Smaller values of *learning_rate* require larger numbers of weak learners to maintain a constant training error, and this comes at the cost of greater computing time. It is recommended to set the learning rate to a small constant and choose

$n_estimators$ by early stopping [71]. This function also supports four different loss functions when updating the weights after each boosting iteration via the parameter *loss*: the linear, the square and the exponential.

Finally, for the Gradient Tree Boosting (GTB) the module *sklearn.ensemble* provides the function *GradientBoostingRegressor*, which uses gradient boosted trees. It supports four different loss functions to be optimized via the parameter *loss*: the squared error, the absolute error, the huber and the quantile. The Huber loss function is a combination of the first two loss functions. It applies the squared error loss for small deviations from the actual value and the absolute error loss for large deviations. There is also a parameter *alpha* that dictates the threshold between these losses. This function also supports two different criteria to measure the quality of a split via the parameter *criterion*: the MSE and the MSE with improvement score by Friedman. The size of the individual trees can be controlled by the parameter *max_depth*. The *subsample* parameter represents the fraction of samples to be used for fitting the individual base learners. The learning rate can be set via *learning_rate* and the number of estimators can be set via *n_estimators*. The number of features to consider when looking for the best split can be given by *max_features*.

3.2. Hyperparameter Optimization

The hyperparameters of each model were optimized using a grid search process and a five-fold cross-validation was performed in the training set. For all methods, an exhaustive search over specified parameter values was performed using the function *GridSearchCV* from the module *sklearn.model_selection*. For the BME model, the parameters that control whether samples and features are drawn with or without replacement were set according to their default stage (samples drawn with replacement and features drawn without replacement). Since averaging methods work best with strong and complex models, the trees for these methods were not given a maximum depth. For the AB and the GTB models, the learning was set to 0.01.

For each model, the tuned values for each hyperparameter are shown in Table 4. The coefficient of determination (R^2) was used as the statistical error to obtain hyperparameters with maximum accuracy while minimizing overfitting. Since there are only three inputs, the maximum number of features when splitting a node was considered to be the default value for all the models, which means all of them. The maximum depth for the DT, AB and GTB models was searched in the range between 3 and 15. The number of estimators was searched in the range between 10 and 100 for the BME and the FRT models, and in the range between 100 and 1500 for the AB and GTB models, since for these ones a low value for the learning rate was chosen. The fractions of the maximum number of samples, the alpha and the subsample were searched in the range 0.1–1.0.

Table 4. Hyperparameters for each model.

Model	Criterion	Max Depth	n Estimators	Max Samples	Loss	α	Subsample
DT	Poisson	8	-	-	-	-	-
BME	MAE	-	66	0.97	-	-	-
FRT	MAE	-	56	-	-	-	-
AB	Friedman MSE	6	1250	-	Square	-	-
GTB	Friedman MSE	3	1500	-	Huber	0.94	0.67

A total of 161 data samples was used to train all the models, and 41 samples were used to test them. The five-fold cross-validation results and statistical breakdown are shown in Table 5, where the coefficient of variation (COV) based on the average R^2 and the standard deviation (STD) are presented for each model. There is no noticeable fluctuation in the results of the five-folds, and the overall accuracy remains very good for all the models. In particular, the FRT model showed to be excellent with the smallest value for the COV of 0.6857%.

Table 5. Cross-validation measurement results.

Model	Statistics	Folds					STD	Average	COV%
		1	2	3	4	5			
DT	R^2	0.9711	0.9255	0.9677	0.9706	0.9538	0.0173	0.9577	1.8064
BME		0.9626	0.9719	0.9761	0.9489	0.9554	0.0101	0.9630	1.0488
FRT		0.9577	0.9691	0.9719	0.9576	0.9563	0.0066	0.9625	0.6857
AB		0.9673	0.9562	0.9812	0.9707	0.9667	0.0081	0.9684	0.8364
GTB		0.9650	0.9662	0.9891	0.9722	0.9759	0.0087	0.9737	0.8935

The validation curve for the coefficient of determination as a function of the maximum depth of the tree for the chosen hyperparameters is represented in Figure 6 for the DT, AB and GTB models. For this, *validation_curve* from the module *sklearn.model_selection* was used. As referred earlier, for the BME and FRT model the trees were not given a maximum depth, since they are averaging methods. Figure 6 shows that, for all models, the R^2 quickly reaches a plateau, and the results stop significantly improving beyond a critical number of maximum depths. It can also be observed that the models are not overfitting, since the training and validation scores are both high. This way, the used values of *max_depth* were deemed as satisfactory.

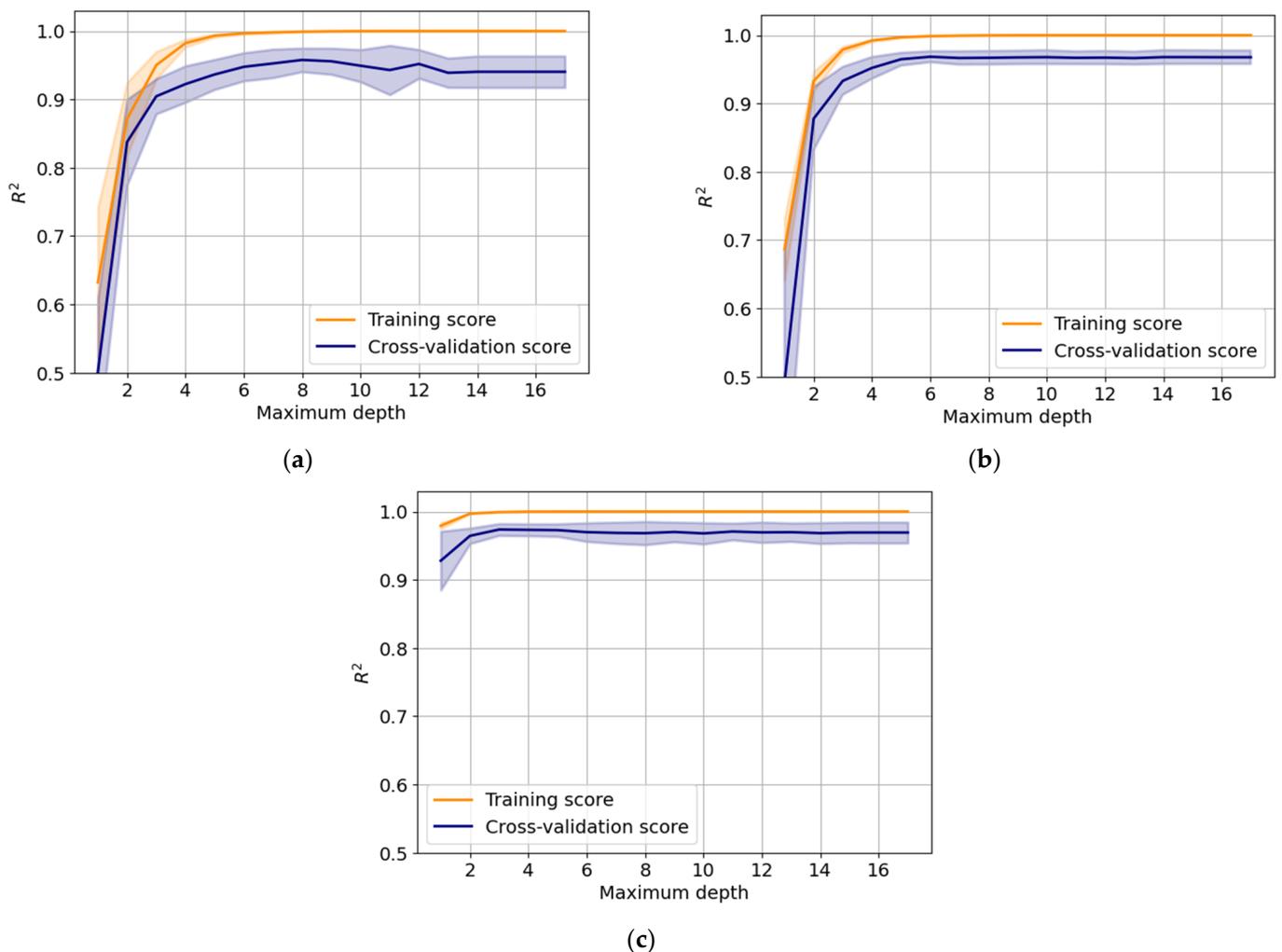


Figure 6. Validation curve for the maximum depth of the trees: (a) DT, (b) AB and (c) GTB.

A validation curve for the number of estimators was calculated to check the performance of the models (except for the DT since it only has one estimator). The validation curve for the R^2 as a function of the number of estimators for the chosen hyperparameters is represented in Figure 7. As it can be observed, the R^2 quickly reaches a plateau, and the results stop significantly improving beyond a critical number of trees. The larger the better, but also the longer it will take to compute. This way, the used values of $n_estimators$ were deemed as satisfactory. It can also be observed from the plot that the models are not overfitting, since the training validation scores are both high.

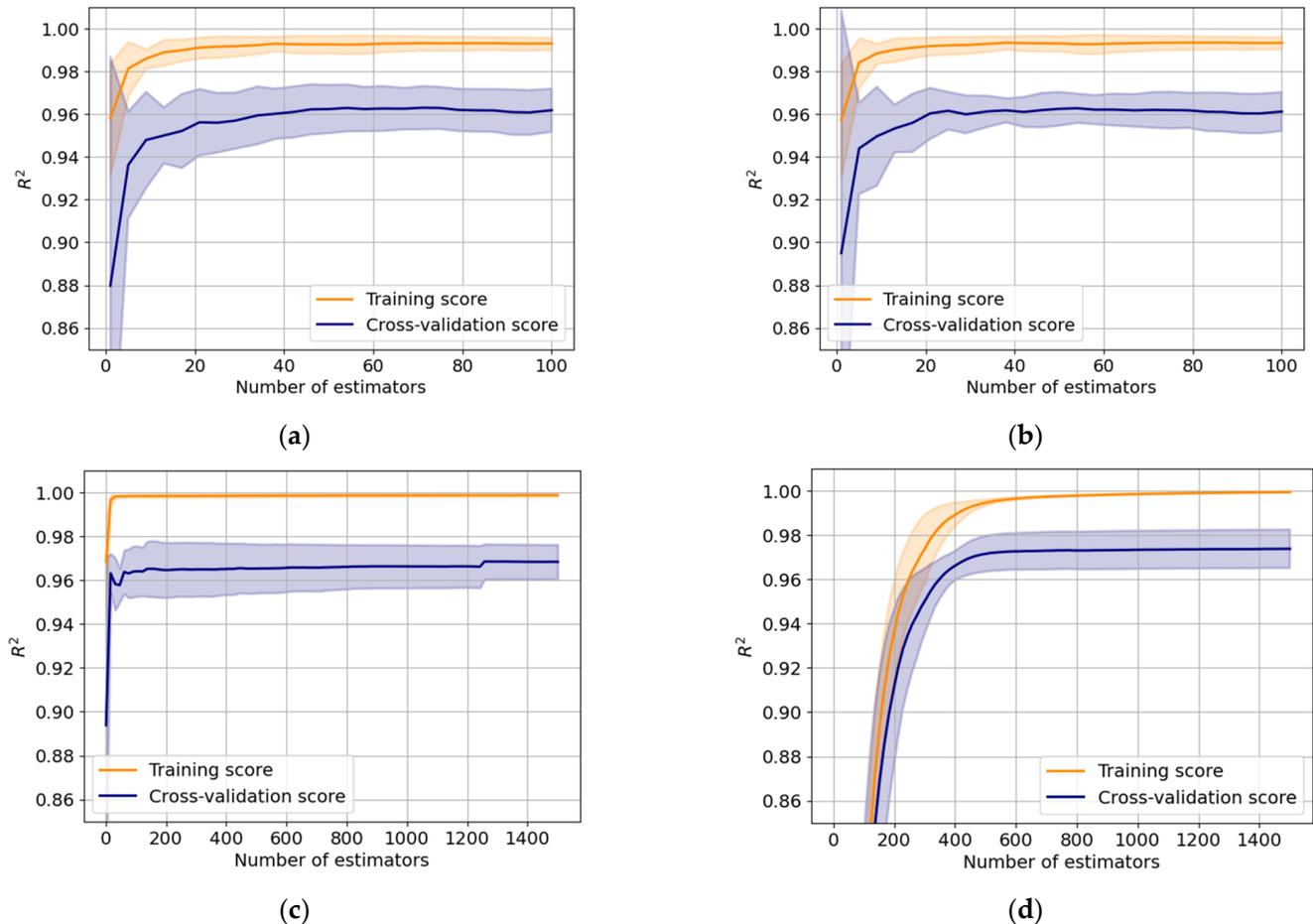


Figure 7. Validation curve for the number of estimators: (a) BME, (b) FRT, (c) AB and (d) GTB.

A learning curve shows the validation and training score of an estimator for several numbers of training samples. It is used to find how much the model benefits from adding more training data and whether the estimator suffers more from a variance or a bias error. The learning curves for all the models with the respective hyperparameters chosen are represented in Figure 8. The function `learning_curve` from the module `sklearn.model_selection` was used. Overall, for small amounts of data, the training score is much greater than the validation score. Adding more training samples increases generalization.

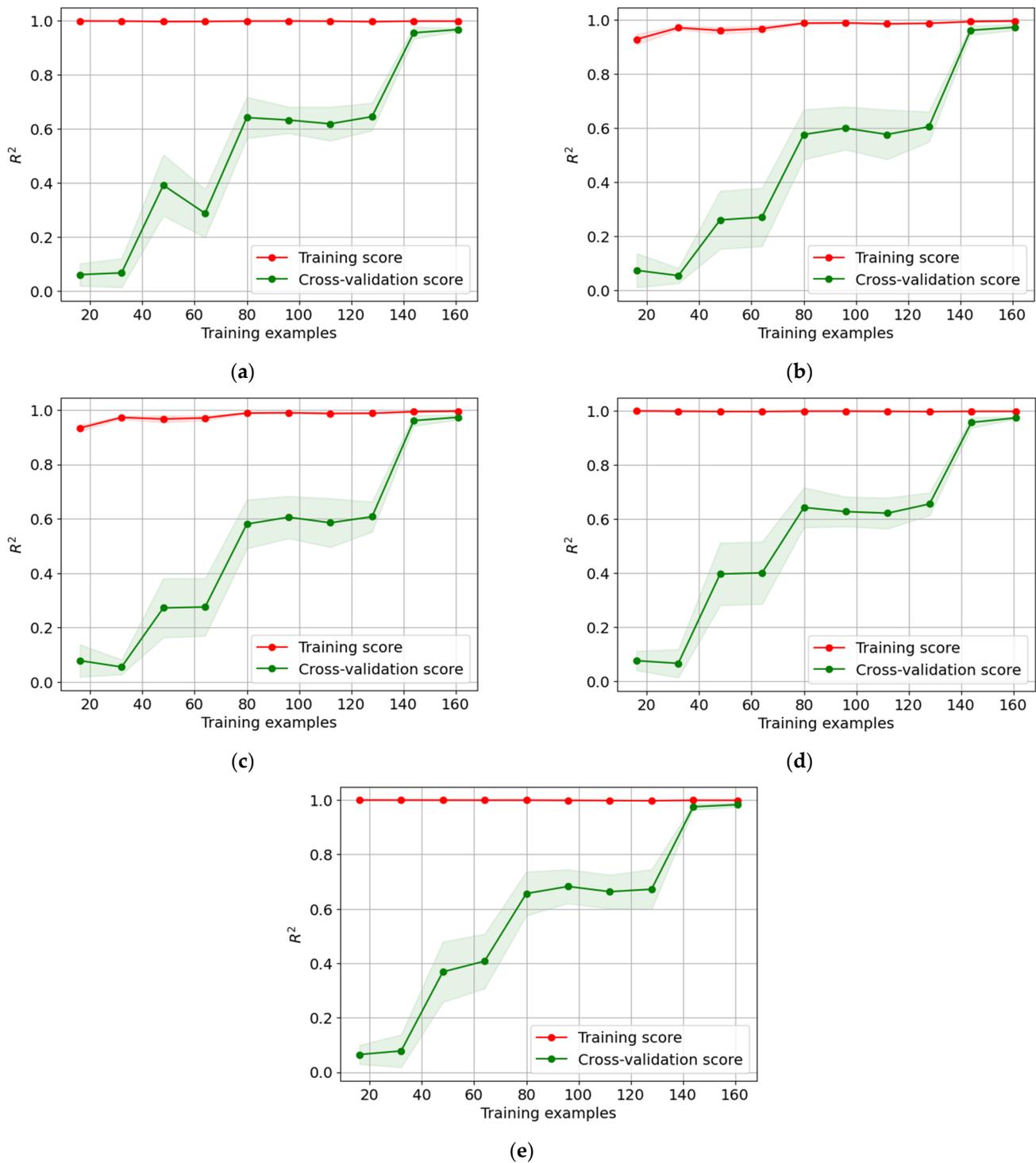


Figure 8. Learning curves: (a) DT, (b) BME, (c) FRT, (d) AB and (e) GTB.

3.3. Performance on Testing Set

The prediction performance of the proposed methods can be tested after the best hyperparameters have been identified. The performance metrics of the models are reported in Table 6. The prediction findings for the testing set are shown in Figure 9, with the x and y axes representing the experimental and predicted torsional capacity, respectively.

Table 6. Performance metrics for the testing results.

Model	MAE (kNm)	RMSE	cvRMSE (%)	R ²	MAPE (%)	RRSE	VAF
DT	10.713	17.749	16.78	0.973	12.18	0.163	0.975
BME	7.616	12.064	11.41	0.988	8.83	0.111	0.989
FRT	7.434	11.461	10.83	0.989	8.72	0.106	0.990
AB	9.901	14.725	13.92	0.982	10.83	0.136	0.983
GTB	6.856	10.618	10.04	0.990	7.21	0.098	0.991

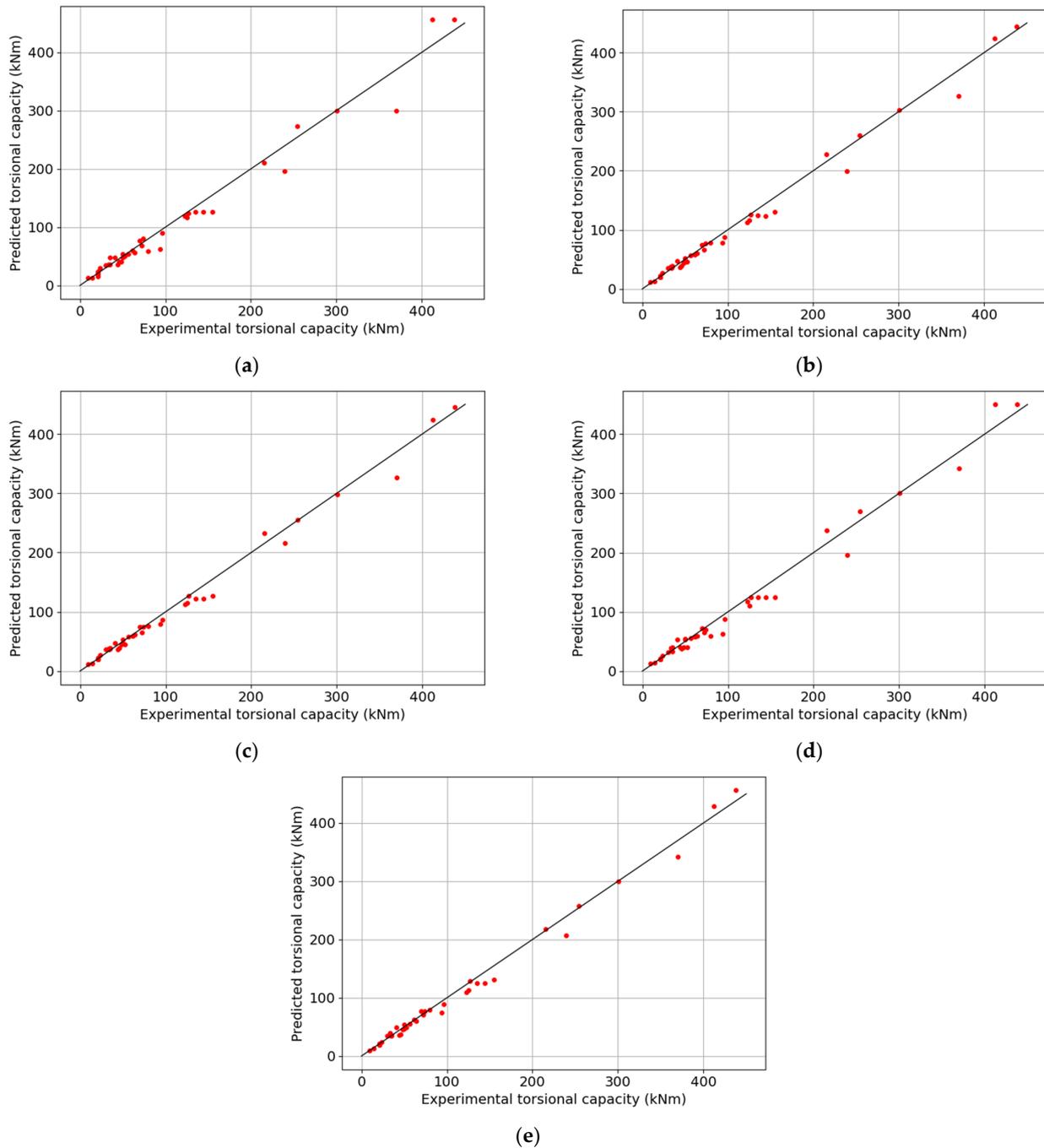


Figure 9. Predictions results: (a) DT, (b) BME, (c) FRT, (d) AB and (e) GTB.

Overall, the difference between the predicted and actual torsional capacity (*MAE*) is low, with an average between 6.9 and 10.7 kNm. The low values obtained for *RMSE*, *cvRMSE*, *MAPE* and *RRSE*, and the high values obtained for *R²* and *VAF* are very acceptable

and show that the used models based on Ensembles of Trees can be considered effective in estimating the torsional capacity of RC beams. From Table 6, the best models in terms of the $cvRMSE$ are the BME, FRT and GTB models, with values of 11.41%, 10.83% and 10.04%, respectively.

Figure 10 presents the histograms of the predicted torsional capacity T_{R_Exp} from all the models compared with the actual torsional capacity T_{R_Pred} . It also presents its normal distribution, for which the mean value and standard deviation are presented in Table 7. The normal distribution shows that the error is dispersed randomly.

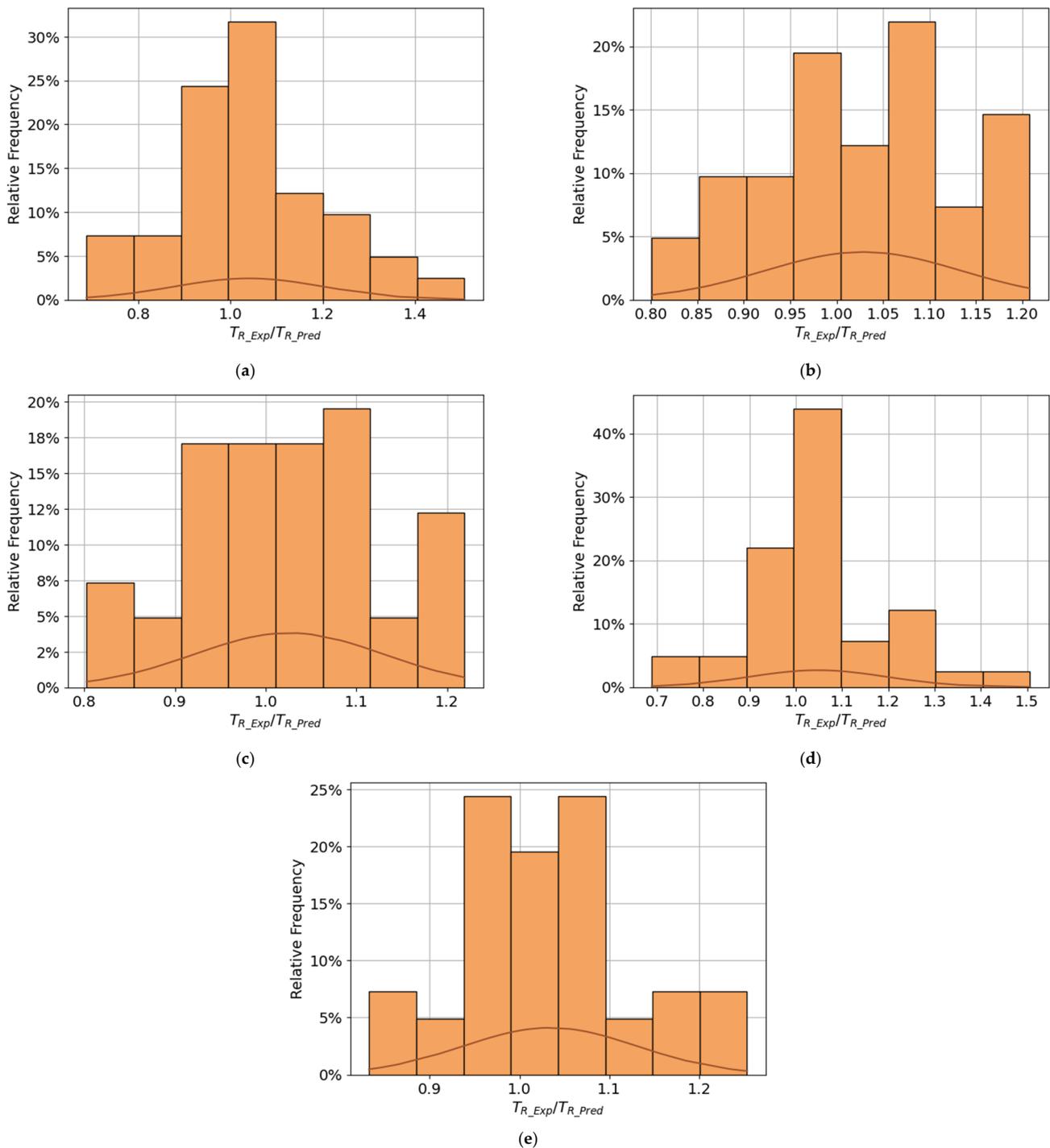


Figure 10. Histogram for T_{R_Exp}/T_{R_Pred} for testing results: (a) DT, (b) BME, (c) FRT, (d) AB and (e) GTB.

Table 7. Mean and standard deviation for testing results.

Model	μ	σ
DT	1.040	0.164
BME	1.029	0.106
FRT	1.026	0.105
AB	1.048	0.150
GTB	1.034	0.097

4. Comparison with the Results from Previous Studies

When compared with the results from previous studies using ML models, referred in Section 1 and related mainly with RC beams under pure torsion, the following can be stated.

The research article from Deifalla and Salem [44] was the only one in which the authors applied two models based on ET, namely a Boosted and a Bagged model, to predict the torsional gain capacity of externally bonded FRP-RC beams, and not the full torsional capacity of common RC beams. They used additional inputs incorporating data about the strengthening system and a dataset with 157 beams. Since the predicted variable was different, only the R^2 parameter is discussed here. With their models, the referred authors achieved values of 0.71 and 0.47 for the Boosted and Bagged models, respectively. These values are much lower than the ones achieved in this study. Reference [44] also used other models, namely four models based on a Gaussian Process Regression and five models based on Neural Networks. For all these models, the R^2 ranged between 0.56 and 0.93, which still remains lower than the ones from this study.

Henedy et al. [47] used the same database and the same inputs as in the present research, and applied a M5P model tree and a non-linear regression model to predict the torsional capacity of the RC beams. For the first one and for the testing set, they achieved the following metric performances: $MAE = 8.224$, $RMSE = 13.432$ and $R^2 = 0.981$. When compared with the metric performances in Table 6 (also for the testing set), it can be stated that the models used in this study achieved similar results, although three of the models (BME, FRT and GTB) showed to be slightly better.

Cevik et al. [42] used a Genetic-programming-based model to predict the torsional capacity of RC beams using a dataset with only 76 beams. The authors used 5 variables among 12 available as inputs. For the testing set, the following metric performances were achieved: $RMSE = 12.9$ and $R^2 = 0.95$. Based on the results from Table 6 (also for the testing set), the same three previously referred models (BME, FRT and GTB) were shown to be slightly better in terms of $RMSE$. Furthermore, all models showed R^2 values higher than 0.95.

Finally, [40,41,43,48] used different artificial neural network (ANN) or convolutional neural network (CNN) models to predict the torsional capacity of RC beams, with a much higher number of inputs (11 or 12) and different datasets (with 76 beams for [41,43], 240 beams for [40] and 268 beams for [48]). Regarding the same metric performances used in this study, for reference [40] and for four different ANN models (BP and GA-BP neural networks, optimized BP and GA-BP neural networks), the metric performances ranged between the following values: $MAE = 6.742$ to 11.548 , $RMSE = 10.154$ to 17.758 and $R^2 = 0.846$ to 0.950 . For reference [41], with only one ANN model (hybrid neural network: GA-MLP), the metric performances were as follows: $MAE = 6.94$ and $R^2 = 0.980$. For reference [43] where eleven ANN algorithms were used, the author achieved the following range of values: $R^2 = 0.9496$ to 0.9876 . Finally, in [48], the authors proposed an improved bird swarm algorithm optimized 2D CNN, for which the following metric performance was achieved: $MAE = 2.9875$. When compared with the results from Table 6, it can be stated that the ET models in this study achieved very similar performance metrics as the ones from the ANN models, but with much lesser inputs.

Based on the previous comparisons, it can be stated that ET models, such as the ones used in this study, can be considered predictive models as good as the best models from the previously referred studies to predict the torsional capacity of RC beams.

5. Sensitivity Analysis

The importance of each input variable in the prediction of the torsional capacity of RC beams is determined by performing a sensitivity analysis. This analysis is carried out by removing each predictor from the database, one at a time, and training and testing the proposed models using the resulting dataset. The MAE, RMSE, cvRMSE and R² performance measures resulting from testing were used to analyze the importance of each input variable. These results are presented in Table 8, in which it can be noticed that the cross-section area A_c is the most important variable of the three inputs when predicting the torsional capacity of the RC beams. The concrete compressive strength f_c in its turn is the variable with the least impact on the performance of the models. These results confirm the ones from a previous study from the authors using different models to predict the torsional capacity of RC beams for the same dataset [47].

Table 8. Effect of input variables on the performance of the ML models.

Models	INPUT VARIABLE	Testing Set						
		MAE	RMSE	cvRMSE (%)	R ²	MAPE (%)	RRSE	VAF
DT	A _c , f _c , A ₁ f _{yl} A _t f _{yt} /s	10.713	17.749	16.78	0.973	12.18	0.163	0.975
	f _c , A ₁ f _{yl} A _t f _{yt} /s	27.620	46.919	44.36	0.813	35.85	0.432	0.815
	A _c , A ₁ f _{yl} A _t f _{yt} /s	13.694	22.280	21.06	0.958	14.94	0.205	0.960
	A _c , f _c	19.959	31.359	29.65	0.917	23.09	0.289	0.926
BME	A _c , f _c , A ₁ f _{yl} A _t f _{yt} /s	7.616	12.064	11.41	0.988	8.83	0.111	0.989
	f _c , A ₁ f _{yl} A _t f _{yt} /s	29.184	48.128	45.50	0.804	32.68	0.443	0.806
	A _c , A ₁ f _{yl} A _t f _{yt} /s	10.030	16.123	15.24	0.978	11.06	0.148	0.980
	A _c , f _c	20.905	34.497	32.61	0.899	20.00	0.318	0.910
FRT	A _c , f _c , A ₁ f _{yl} A _t f _{yt} /s	7.434	11.461	10.83	0.989	8.72	0.106	0.990
	f _c , A ₁ f _{yl} A _t f _{yt} /s	28.391	44.576	42.14	0.832	35.61	0.410	0.833
	A _c , A ₁ f _{yl} A _t f _{yt} /s	9.974	16.468	15.57	0.977	11.01	0.152	0.979
	A _c , f _c	20.586	31.897	30.16	0.914	20.20	0.294	0.925
AB	A _c , f _c , A ₁ f _{yl} A _t f _{yt} /s	9.901	14.725	13.92	0.982	10.83	0.136	0.983
	f _c , A ₁ f _{yl} A _t f _{yt} /s	32.522	54.019	51.07	0.753	36.96	0.497	0.755
	A _c , A ₁ f _{yl} A _t f _{yt} /s	12.225	21.030	19.88	0.963	12.22	0.194	0.964
	A _c , f _c	18.927	29.762	28.14	0.925	20.83	0.274	0.931
GTB	A _c , A ₁ f _{yl} A _t f _{yt} /s	6.856	10.618	10.04	0.990	7.21	0.098	0.991
	f _c , A ₁ f _{yl} A _t f _{yt} /s	29.846	46.988	44.42	0.813	45.15	0.433	0.815
	A _c , A ₁ f _{yl} A _t f _{yt} /s	13.664	23.794	22.49	0.952	14.52	0.219	0.952
	A _c , f _c	24.378	41.416	39.155	0.855	21.97	0.381	0.896

6. Conclusions

An investigation on the use of ML methods based on the Ensemble of Trees (ET) for the prediction of the torsional capacity of RC beams was conducted in this study. A database with 202 RC beam samples was created and randomly divided into testing and training sets. Using a five-fold cross-validation procedure paired with a grid search strategy, optimal hyperparameters for all the models were found based on the training dataset. The testing dataset was used to validate the performance of the built models. From the obtained results, the following main conclusions that can be drawn:

- A simple Decision Tree (DT) regressor predicts the torsional capacity of RC beams with satisfactory accuracy. The model has a R² value of 0.973 and a cvRMSE value of 16.78% for the testing set;
- It was shown that ET reach higher accuracies than a simple DT. The Bagging Meta-Estimator (BME), the Forests of Randomized Trees (FRT), the AdaBoost (AB) and the Gradient Tree Boosting (GTB) reached values of R² in the range between 0.982 and 0.990, and values of cvRMSE in the range between 10.04% and 13.92%. These results indicate that the prediction capability of these four models can be trusted with high confidence, in particular for the GTB and FRT models;

- The cross-section area A_c was shown to be the most important variable of the three inputs when predicting the torsional capacity of the RC beams, while the concrete compressive strength f_c was shown to be the variable with the least impact;
- With low error measurements and mean values μ_M near unity, the results showed that these four ET methods (BME, FRT, AB and GTB) can be considered predictive models as good as the best models from the referred previous studies to predict the torsional capacity of RC beams.

With only 202 samples, the dataset used to develop the prediction models can be considered somewhat small, which represents a limitation of this investigation. The accuracy of the models can be improved in the future by using a larger dataset.

Finally, upon request to the corresponding author, the Python scripts developed in this article can be made available, namely for researchers and practitioners.

Author Contributions: D.S.O.B.—methodology, software, formal analysis, investigation, data curation, and writing—original draft preparation. L.F.A.B.—conceptualization, validation, supervision, data acquisition, and writing—original draft preparation. H.I.—validation, and writing—review and editing. T.P.R.—validation, and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

A_c	area enclosed within the outer perimeter of the cross-section
A_l	total area of longitudinal reinforcement
A_t	area for one bar of the hoop
f_c	average compressive strength of concrete
f_{ly}	average yield strength of longitudinal reinforcement
f_{ty}	average yield strength of transverse reinforcement
s	longitudinal spacing between hoops
T	torsional moment
T_R	torsional moment resistance
T_{R_Exp}	experimental torsional moment resistance
T_{R_Pred}	theoretical torsional moment resistance
t	thickness of the wall for hollow cross-sections
s	longitudinal spacing between stirrups
x	smaller dimension of the cross-section
x_1	smaller dimension of hoops
y	larger dimension of the cross-section
y_1	larger dimension of hoops
ρ_l	ratio of longitudinal reinforcement: $\rho_l = A_l/xy$
ρ_t	ratio of transverse reinforcement: $\rho_t = A_tu/As$

Appendix A

Table A1. Geometric and mechanical properties of the reference RC beams.

Ref.	Beam	Section Type	x	y	t	x_1	y_1	A_1	A_t/s	f_c	f_{ly}	f_{ty}	T_{R_Exp}
			(m)	(m)	(m)	(m)	(m)	(cm ²)	(cm ² /m)	(MPa)	(MPa)	(MPa)	(kNm)
	B1	Plain	0.254	0.381	-	0.216	0.343	5.07	4.68	27.6	314.0	341.0	22.30
	B3	Plain	0.254	0.381	-	0.216	0.343	11.36	10.16	28.1	327.6	320.0	37.48
	B4	Plain	0.254	0.381	-	0.216	0.343	15.48	14.01	29.2	320.0	323.4	47.30
	B5	Plain	0.254	0.381	-	0.216	0.343	20.39	18.47	30.6	332.4	321.4	56.11
	B6	Plain	0.254	0.381	-	0.216	0.343	25.81	22.58	28.8	331.7	322.8	61.64
	B7	Plain	0.254	0.381	-	0.216	0.343	5.16	10.16	26.0	320.0	318.6	26.87
	B8	Plain	0.254	0.381	-	0.216	0.343	5.16	22.58	26.8	322.1	320.0	32.51
	B9	Plain	0.254	0.381	-	0.216	0.343	11.36	4.66	28.8	319.3	342.8	29.80
	B10	Plain	0.254	0.381	-	0.216	0.343	25.8	4.66	26.5	334.0	342.0	34.40
	C2	Plain	0.254	0.254	-	0.216	0.216	5.07	6.07	26.5	334.0	345.0	15.30
	C4	Plain	0.254	0.254	-	0.216	0.216	11.36	13.11	27.2	336.6	327.6	25.29
	C5	Plain	0.254	0.254	-	0.216	0.216	15.48	17.67	27.2	328.3	329.0	29.69
	C6	Plain	0.254	0.254	-	0.216	0.216	20.39	23.91	27.6	315.9	327.6	34.21
	G2	Plain	0.254	0.508	-	0.216	0.470	7.94	5.91	30.9	323.0	334.0	40.30
	G3	Plain	0.254	0.508	-	0.216	0.470	11.36	8.29	26.8	338.6	327.6	49.56
	G4	Plain	0.254	0.508	-	0.216	0.470	15.48	11.29	28.3	325.5	321.4	64.80
	G5	Plain	0.254	0.508	-	0.216	0.470	20.39	15.05	26.9	331.0	327.6	71.91
	G6	Plain	0.254	0.508	-	0.216	0.470	7.60	5.61	29.9	334.0	350.0	39.10
	G7	Plain	0.254	0.508	-	0.216	0.470	12.00	8.84	31.0	319.3	322.8	52.61
[5]	G8	Plain	0.254	0.508	-	0.216	0.470	17.03	12.32	28.3	322.1	329.0	73.38
	I2	Plain	0.254	0.381	-	0.216	0.343	7.94	7.25	45.2	325	349	36.00
	I3	Plain	0.254	0.381	-	0.216	0.343	11.36	10.16	44.8	343.4	333.8	45.61
	I4	Plain	0.254	0.381	-	0.216	0.343	15.48	14.01	45.0	315.2	326.2	58.02
	I5	Plain	0.254	0.381	-	0.216	0.343	20.39	18.47	45.0	310.3	325.5	70.67
	I6	Plain	0.254	0.381	-	0.216	0.343	25.81	22.58	45.8	325.5	329.0	76.65
	J1	Plain	0.254	0.381	-	0.216	0.343	5.16	4.66	14.3	327.6	346.2	21.45
	J2	Plain	0.254	0.381	-	0.216	0.343	8.00	7.21	14.6	320.0	340.7	29.13
	J3	Plain	0.254	0.381	-	0.216	0.343	11.36	10.16	16.9	388.6	337.2	35.22
	J4	Plain	0.254	0.381	-	0.216	0.343	15.48	14.01	16.8	324.1	331.7	40.64
	K2	Plain	0.152	0.495	-	0.114	0.457	7.74	6.77	30.6	335.9	337.9	23.71
	K3	Plain	0.152	0.495	-	0.114	0.457	12.00	10.42	29.0	315.9	320.7	28.45
	K4	Plain	0.152	0.495	-	0.114	0.457	17.03	15.05	28.6	344.1	340.0	35.00
	M1	Plain	0.254	0.381	-	0.216	0.343	8.00	4.76	29.9	326.2	353.1	30.37
	M2	Plain	0.254	0.381	-	0.216	0.343	11.36	6.77	30.6	329.0	357.2	40.53
	M3	Plain	0.254	0.381	-	0.216	0.343	15.48	9.24	26.8	322.1	326.2	43.80
	M4	Plain	0.254	0.381	-	0.216	0.343	20.39	12.33	26.6	318.6	326.9	49.56
	M5	Plain	0.254	0.381	-	0.216	0.343	25.81	15.63	28.0	335.2	331.0	55.65
	M6	Plain	0.254	0.381	-	0.216	0.343	30.58	15.63	29.4	317.9	340.7	60.06
	N1	Plain	0.152	0.305	-	0.130	0.283	2.84	3.50	29.5	352.4	341.4	9.09
	N1a	Plain	0.152	0.305	-	0.130	0.283	2.84	3.50	28.7	346.2	344.8	8.99

Ref.	Beam	Section Type	<i>x</i> (m)	<i>y</i> (m)	<i>t</i> (m)	<i>x</i> ₁ (m)	<i>y</i> ₁ (m)	<i>A</i> _l (cm ²)	<i>A</i> _{t/s} (cm ² /m)	<i>f</i> _c (MPa)	<i>f</i> _{ly} (MPa)	<i>f</i> _{ty} (MPa)	<i>T</i> _{R_Exp} (kNm)
[5]	N2	Plain	0.152	0.305	-	0.130	0.283	5.16	6.35	30.4	331.0	337.9	14.45
	N2a	Plain	0.152	0.305	-	0.130	0.283	1.61	6.21	28.4	333.1	360.7	13.21
	N3	Plain	0.152	0.305	-	0.130	0.283	4.26	5.08	27.3	351.7	351.7	12.19
	N4	Plain	0.152	0.305	-	0.130	0.283	6.58	7.98	27.3	340.9	355.9	15.69
[15]	T4	Plain	0.500	0.500	-	0.454	0.454	18.10	10.28	35.3	356.7	356.7	138.61
[17]	VB2	Plain	0.440	0.240	-	0.420	0.220	7.01	5.84	26.4	541.4	541.4	42.11
	VB3	Plain	0.440	0.240	-	0.420	0.220	7.01	5.84	39.1	541.4	541.4	46.40
	VB4	Plain	0.440	0.240	-	0.420	0.220	7.01	5.84	49.8	541.4	541.4	48.54
	VM1	Plain	0.294	0.160	-	0.280	0.146	3.00	3.63	39.1	442.4	568.9	13.89
	VM2	Plain	0.440	0.240	-	0.420	0.220	6.60	5.32	36.1	431.6	436.5	39.17
	VM3	Plain	0.587	0.320	-	0.561	0.294	12.84	7.14	40.0	461.0	442.4	100.80
	VQ1	Plain	0.324	0.324	-	0.304	0.304	3.46	2.88	19.0	557.1	557.1	21.11
	VQ3	Plain	0.580	0.186	-	0.560	0.166	4.27	3.05	17.6	432.6	432.6	19.98
	VQ9	Plain	0.806	0.140	-	0.786	0.120	5.08	2.82	19.5	441.4	441.4	21.90
	VS2-VQ2	Plain	0.440	0.240	-	0.420	0.220	3.66	3.05	19.0	432.6	432.6	19.53
	VS3	Plain	0.440	0.240	-	0.420	0.220	5.49	4.55	19.5	432.6	432.6	28.56
	VS4-VQ5	Plain	0.440	0.240	-	0.420	0.220	7.32	6.10	19.0	432.6	432.6	34.32
	VS9	Plain	0.440	0.240	-	0.420	0.220	3.48	2.90	17.6	570.9	570.9	21.56
	VS10-VB1	Plain	0.440	0.240	-	0.420	0.220	6.96	5.80	19.0	570.9	570.9	33.30
	VU1	Plain	0.440	0.240	-	0.420	0.220	3.36	5.60	19.5	441.4	441.4	23.93
	VU2	Plain	0.440	0.240	-	0.420	0.220	5.04	5.60	19.5	441.4	441.4	30.37
	VU3	Plain	0.440	0.240	-	0.420	0.220	6.72	4.18	18.5	441.4	441.4	31.04
VU4	Plain	0.440	0.240	-	0.420	0.220	6.72	2.80	18.5	441.4	441.4	25.96	
[6]	A2	Plain	0.254	0.254	-	0.222	0.222	5.16	7.82	38.2	380.0	285.0	22.58
	A3	Plain	0.254	0.254	-	0.219	0.219	8.00	8.94	39.4	352.4	360.0	27.77
	A4	Plain	0.254	0.254	-	0.219	0.219	11.36	12.42	39.2	351.0	360.0	34.43
	B1r	Plain	0.178	0.356	-	0.146	0.324	2.85	3.87	36.3	360.0	285.0	12.30
	B2	Plain	0.178	0.356	-	0.146	0.324	5.07	7.19	39.6	380.0	285.0	20.80
	B3	Plain	0.178	0.356	-	0.143	0.321	8.00	8.60	38.6	352.4	360.0	25.29
	B4	Plain	0.178	0.356	-	0.143	0.321	11.36	11.76	38.5	351.0	360.0	31.72
	B30.1	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	41.7	620.0	665.0	16.62
	B30.2	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	38.2	638.0	669.0	15.29
	B30.3	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	36.3	605.0	672.0	15.25
[58]	B50.1	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	61.8	612.0	665.0	19.95
	B50.2	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	57.1	614.0	665.0	18.46
	B50.3	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	61.7	612.0	665.0	19.13
	B70.1	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	77.3	617.0	658.0	20.06
	B70.2	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	76.9	614.0	656.0	20.74
	B70.3	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	76.2	617.0	663.0	20.96
	B110.1	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	109.8	618.0	655.0	24.72
	B110.2	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	105.0	634.0	660.0	23.62

Ref.	Beam	Section Type	<i>x</i> (m)	<i>y</i> (m)	<i>t</i> (m)	<i>x</i> ₁ (m)	<i>y</i> ₁ (m)	<i>A</i> _l (cm ²)	<i>A</i> _{t/s} (cm ² /m)	<i>f</i> _c (MPa)	<i>f</i> _{ly} (MPa)	<i>f</i> _{ty} (MPa)	<i>T</i> _{R_Exp} (kNm)
[58]	B110.3	Plain	0.160	0.275	-	0.120	0.235	15.44	8.73	105.1	629.0	655.0	24.77
	B5UR1	Plain	0.203	0.305	-	0.165	0.267	5.16	6.56	39.6	386.0	373.0	19.40
	B7UR1	Plain	0.203	0.305	-	0.165	0.267	5.16	6.56	64.6	386.0	399.0	18.90
	B9UR1	Plain	0.203	0.305	-	0.165	0.267	5.16	6.56	75.0	386.0	373.0	21.10
	B12UR1	Plain	0.203	0.305	-	0.165	0.267	5.16	6.56	80.6	386.0	399.0	19.40
[59]	B12UR2	Plain	0.203	0.305	-	0.165	0.267	5.16	6.95	76.2	386.0	386.0	18.40
	B12UR3	Plain	0.203	0.305	-	0.165	0.267	6.58	7.46	72.9	379.5	386.0	22.50
	B12UR4	Plain	0.203	0.305	-	0.165	0.267	7.74	7.88	75.9	373.0	386.0	23.70
	B12UR5	Plain	0.203	0.305	-	0.165	0.267	8.00	10.13	76.7	380.0	386.0	24.00
	B14UR1	Plain	0.203	0.305	-	0.165	0.267	5.16	6.56	93.9	386.0	386.0	21.00
[60]	H-06-06	Plain	0.350	0.500	-	0.300	0.450	11.92	7.13	78.5	440.0	440.0	92.00
	H-06-12	Plain	0.350	0.500	-	0.300	0.450	20.65	7.10	78.5	410.0	440.0	115.10
	H-07-10	Plain	0.350	0.500	-	0.300	0.450	17.03	7.89	68.4	500.0	420.0	126.70
	H-07-16	Plain	0.350	0.500	-	0.300	0.450	28.39	7.89	68.4	500.0	420.0	144.50
	H-12-12	Plain	0.350	0.500	-	0.300	0.450	20.65	14.19	78.5	410.0	440.0	155.30
	H-12-16	Plain	0.350	0.500	-	0.300	0.450	28.39	14.19	78.5	520.0	440.0	196.00
	H-14-10	Plain	0.350	0.500	-	0.300	0.450	17.03	16.13	68.4	500.0	360.0	135.20
	H-20-20	Plain	0.350	0.500	-	0.300	0.450	34.06	23.46	78.5	560.0	440.0	239.00
	N-06-06	Plain	0.350	0.500	-	0.300	0.450	12.00	7.10	35.5	440.0	440.0	79.70
	N-06-12	Plain	0.350	0.500	-	0.300	0.450	20.65	7.10	35.5	410.0	440.0	95.20
	N-07-10	Plain	0.350	0.500	-	0.300	0.450	17.03	7.89	33.5	500.0	420.0	111.70
	N-07-16	Plain	0.350	0.500	-	0.300	0.450	28.39	7.89	33.5	500.0	420.0	117.30
	N-12-12	Plain	0.350	0.500	-	0.300	0.450	20.65	14.19	35.5	410.0	440.0	116.80
	N-12-16	Plain	0.350	0.500	-	0.300	0.450	28.39	14.19	35.5	520.0	440.0	138.00
	N-14-10	Plain	0.350	0.500	-	0.300	0.450	17.03	16.13	33.5	500.0	360.0	125.00
N-20-20	Plain	0.350	0.500	-	0.300	0.450	34.06	23.46	35.5	560.0	440.0	158.00	
[61]	HAS-51-50	Plain	0.420	0.420	-	0.370	0.370	9.03	5.94	76.0	396.0	385.0	84.90
	NAS-61-35	Plain	0.420	0.420	-	0.370	0.370	10.80	4.19	48.0	394.0	385.0	74.70
	HAS-90-50	Plain	0.420	0.420	-	0.370	0.370	15.89	5.94	78.0	400.0	385.0	104.23
	NBS-43-44	Plain	0.350	0.500	-	0.300	0.450	7.60	5.09	35.0	400.0	385.0	60.60
	HBS-74-17	Plain	0.350	0.500	-	0.300	0.450	12.89	2.02	67.0	505.0	600.0	62.20
	HBS-82-13	Plain	0.350	0.500	-	0.300	0.450	14.31	1.49	67.0	493.0	600.0	56.30
	NBS-82-13	Plain	0.350	0.500	-	0.300	0.450	14.31	1.49	35.0	493.0	600.0	52.90
	HBS-60-61	Plain	0.350	0.500	-	0.300	0.450	10.48	7.13	67.0	402.0	385.0	93.70
	HCS-52-50	Plain	0.250	0.700	-	0.200	0.650	9.03	5.09	76.0	396.0	385.0	73.54
	HCS-91-50	Plain	0.250	0.700	-	0.200	0.650	15.89	5.09	78.0	400.0	385.0	95.86
[62]	T1-1	Plain	0.300	0.350	-	0.260	0.310	5.07	5.48	43.2	410.0	370.0	32.90
	T1-2	Plain	0.300	0.350	-	0.260	0.310	7.60	8.39	44.0	410.0	370.0	45.90
	T1-3	Plain	0.300	0.350	-	0.260	0.310	10.14	10.97	41.7	410.0	370.0	54.10
	T1-4	Plain	0.300	0.350	-	0.260	0.310	11.92	16.89	42.6	510.0	355.0	62.40
	T2-2	Plain	0.300	0.350	-	0.260	0.310	7.94	5.48	41.7	510.0	370.0	38.10

Ref.	Beam	Section Type	<i>x</i>	<i>y</i>	<i>t</i>	<i>x</i> ₁	<i>y</i> ₁	<i>A</i> _I	<i>A</i> _{t/s}	<i>f</i> _c	<i>f</i> _{ly}	<i>f</i> _{ty}	<i>T</i> _{R_Exp}
			(m)	(m)	(m)	(m)	(m)	(cm ²)	(cm ² /m)	(MPa)	(MPa)	(MPa)	(kNm)
[62]	T2-3	Plain	0.300	0.350	-	0.260	0.310	11.92	8.10	42.7	510.0	370.0	50.20
	T2-4	Plain	0.300	0.350	-	0.260	0.310	14.00	9.51	42.6	512.4	370.0	56.40
	SW12-1	Plain	0.150	1.200	-	0.100	1.150	22.62	3.93	44.2	480.0	459.0	74.00
	SW10-1	Plain	0.150	1.000	-	0.100	0.950	18.10	3.93	29.5	499.0	459.0	50.70
	SW10-2	Plain	0.150	1.000	-	0.098	0.948	18.10	7.54	44.2	480.0	480.0	68.00
[56]	SW10-3	Plain	0.150	1.000	-	0.098	0.948	18.10	11.31	29.5	499.0	499.0	74.30
	SW10-4	Plain	0.150	1.000	-	0.094	0.944	32.16	16.08	33.8	497.0	497.0	80.50
	SW8-1	Plain	0.150	0.800	-	0.102	0.752	14.14	4.02	29.5	459.0	433.0	40.40
	SW8-2	Plain	0.150	0.800	-	0.098	0.748	14.14	11.31	29.5	459.0	499.0	60.10
	RA-SD4-3.2-0.3-3.28	Plain	0.300	0.400	-	0.270	0.370	30.97	7.13	73.7	452.0	484.0	86.80
RA-SD5-3.2-0.3-3.21	Plain	0.300	0.400	-	0.270	0.370	30.97	6.48	73.7	499.0	538.0	88.00	
[63]	RA-SD6-3.2-0.2-3.21	Plain	0.300	0.400	-	0.270	0.370	30.97	6.48	73.7	630.0	538.0	89.40
	RA-SD4-3.2-0.5-2.13	Plain	0.300	0.400	-	0.270	0.370	17.19	7.13	84.7	456.0	484.0	76.30
	RA-SD5-3.2-0.7-1.63	Plain	0.300	0.400	-	0.270	0.370	11.92	6.48	84.7	529.0	538.0	74.50
	RA-SD6-3.2-0.6-1.63	Plain	0.300	0.400	-	0.270	0.370	11.92	6.48	84.7	627.0	538.0	70.00
	RA-SD4-3.2-1.1-1.33	Plain	0.300	0.400	-	0.270	0.370	7.60	7.13	83.1	474.0	484.0	74.20
RA-SD5-3.2-1.0-1.26	Plain	0.300	0.400	-	0.270	0.370	7.60	6.48	83.1	522.0	538.0	67.70	
RA-SD6-3.2-0.8-1.26	Plain	0.300	0.400	-	0.270	0.370	7.94	6.48	83.1	627.0	538.0	69.90	
[64]	MR30-0.77	Plain	0.350	0.500	-	0.300	0.450	7.60	3.96	29.3	489.8	467.5	55.00
	MT30-1.32	Plain	0.350	0.500	-	0.300	0.450	17.19	3.96	29.3	500.4	467.5	57.00
	MT40-1.32	Plain	0.350	0.500	-	0.300	0.450	17.19	3.96	40.3	500.4	467.5	58.30
	MT40-1.89	Plain	0.350	0.500	-	0.300	0.450	17.19	10.57	40.3	489.8	489.8	98.40
	NSC-S1-C30	Plain	0.200	0.300	-	0.168	0.268	4.52	6.28	42.1	689.7	534.1	19.70
[65]	NSC-S1-C45	Plain	0.200	0.300	-	0.138	0.238	4.52	6.28	39.4	689.7	534.1	12.50
	HSC-C30	Plain	0.200	0.300	-	0.168	0.268	4.52	6.28	60.8	689.7	534.1	19.90
	HSC-C45	Plain	0.200	0.300	-	0.138	0.238	4.52	6.28	60.8	689.7	534.1	13.80
	S08-3-65	Plain	0.400	0.600	-	0.310	0.510	18.08	10.97	35.4	313.3	334.9	123.00
	S08-4-90	Plain	0.400	0.600	-	0.310	0.510	13.30	7.92	35.4	474.6	485.8	124.00
S08-5-122.5	Plain	0.400	0.600	-	0.310	0.510	10.45	5.82	35.4	569.6	595.9	89.00	
[66]	S10-3-52.5	Plain	0.400	0.600	-	0.310	0.510	22.39	13.58	35.4	320.5	334.0	126.00
	S10-4-72.5	Plain	0.400	0.600	-	0.310	0.510	16.63	9.83	35.4	467.6	485.1	109.00
	S10-5-100	Plain	0.400	0.600	-	0.310	0.510	12.98	7.13	35.4	567.6	595.2	108.00
	S06-3-90	Plain	0.400	0.600	-	0.310	0.510	13.30	7.92	35.4	319.4	334.3	101.00
	S10-5-90	Plain	0.400	0.600	-	0.310	0.510	14.41	7.92	35.4	569.6	594.8	106.00
S08-3-72.5	Plain	0.400	0.600	-	0.310	0.510	16.63	9.83	35.4	308.8	334.8	106.00	
S12-5-72.5	Plain	0.400	0.600	-	0.310	0.510	18.05	9.83	35.4	565.1	595.2	120.00	
[5]	D3	Hollow	0.254	0.381	0.064	0.216	0.343	11.36	10.16	28.4	341.4	333.1	39.11
	D4	Hollow	0.254	0.381	0.064	0.216	0.343	15.48	14.01	30.6	330.3	333.1	47.93
[15,57]	T0	Hollow	0.500	0.500	0.080	0.430	0.430	32.16	10.28	45.1	345.2	357.0	185.50
	T1	Hollow	0.500	0.500	0.080	0.454	0.454	18.10	10.28	35.3	356.7	356.7	140.01
	T2	Hollow	0.500	0.500	0.080	0.430	0.430	18.10	10.28	35.3	357.0	357.0	143.10

Ref.	Beam	Section Type	x (m)	y (m)	t (m)	x_1 (m)	y_1 (m)	A_l (cm ²)	A_t/s (cm ² /m)	f_c (MPa)	f_{ty} (MPa)	f_{ty} (MPa)	T_{R_Exp} (kNm)
[15,57]	T5	Hollow	0.800	0.400	0.080	0.730	0.330	10.00	10.28	47.1	528.6	512.9	156.88
[17]	VH1	Hollow	0.324	0.324	0.080	0.304	0.304	3.46	2.88	17.2	447.3	447.3	21.79
	VH2	Hollow	0.324	0.324	0.080	0.304	0.304	6.91	5.76	17.2	447.3	447.3	34.50
[67]	A1	Hollow	0.600	0.600	0.098	0.537	0.547	6.53	3.14	48.4	695.9	636.7	150.79
	A2	Hollow	0.600	0.600	0.107	0.538	0.531	13.95	6.28	47.3	672.4	695.9	254.08
	A3	Hollow	0.600	0.600	0.109	0.535	0.535	18.10	8.27	46.2	672.4	714.8	299.92
	A4	Hollow	0.600	0.600	0.104	0.520	0.525	23.75	11.22	54.8	723.9	714.8	368.22
	A5	Hollow	0.600	0.600	0.104	0.528	0.528	30.66	14.14	53.1	723.9	672.4	412.24
	B2	Hollow	0.600	0.600	0.108	0.533	0.534	14.58	6.70	69.8	672.4	695.9	273.28
	B3	Hollow	0.600	0.600	0.109	0.535	0.537	23.75	11.22	77.8	723.9	714.8	355.85
	B4	Hollow	0.600	0.600	0.112	0.523	0.536	32.17	15.08	79.8	723.9	672.4	437.85
	B5	Hollow	0.600	0.600	0.117	0.518	0.518	40.21	18.85	76.4	723.9	672.4	456.19
	C1	Hollow	0.600	0.600	0.097	0.540	0.549	6.53	3.14	91.7	695.9	636.7	151.76
	C2	Hollow	0.600	0.600	0.100	0.532	0.533	13.95	6.28	94.8	672.4	695.9	266.14
	C3	Hollow	0.600	0.600	0.103	0.545	0.540	23.75	10.47	91.6	723.9	714.8	351.17
	C4	Hollow	0.600	0.600	0.103	0.546	0.545	30.66	14.14	91.4	723.9	672.4	450.31
	C5	Hollow	0.600	0.600	0.104	0.540	0.543	36.69	17.40	96.7	723.9	672.4	467.26
	C6	Hollow	0.600	0.600	0.104	0.533	0.529	48.25	22.62	87.5	723.9	672.4	521.33
[61]	HAH-81-35	Hollow	0.420	0.420	0.075	0.370	0.370	14.31	4.19	78.0	493.0	385.0	94.31
	NCH-62-33	Hollow	0.250	0.700	0.075	0.200	0.650	10.80	3.40	48.0	394.0	385.0	64.14
	HCH-91-42	Hollow	0.250	0.700	0.075	0.200	0.650	15.89	4.32	78.0	400.0	385.0	87.51
	A095c	Hollow	0.497	0.711	0.145	0.437	0.651	13.16	9.93	35.1	371.0	381.0	209.98
	A120a	Hollow	0.502	0.719	0.184	0.442	0.659	20.00	7.59	27.6	464.0	380.0	215.25
[68]	B065b	Hollow	0.503	0.710	0.092	0.443	0.650	50.97	9.93	39.2	452.0	380.0	278.00
	B080a	Hollow	0.500	0.721	0.112	0.440	0.661	28.39	12.90	46.5	454.0	392.0	300.66
	B110a	Hollow	0.498	0.710	0.155	0.438	0.650	20.00	8.60	48.1	453.0	369.0	237.48
	C100a	Hollow	0.499	0.723	0.127	0.439	0.663	28.39	12.90	90.6	466.0	447.0	370.15
	D075a	Hollow	0.498	0.734	0.087	0.438	0.674	28.39	12.90	94.9	469.0	381.0	339.48
	D090a	Hollow	0.501	0.722	0.105	0.441	0.662	28.39	12.90	105.7	466.0	447.0	343.08
	H08-3-65	Hollow	0.400	0.600	0.100	0.310	0.510	18.08	10.97	36.5	361.1	352.2	128.00
	H08-4-90	Hollow	0.400	0.600	0.100	0.310	0.510	13.30	7.92	36.5	445.7	448.9	130.00
	H08-5-100	Hollow	0.400	0.600	0.100	0.310	0.510	11.88	7.13	36.5	545.5	539.8	117.00
	H10-3-52.5	Hollow	0.400	0.600	0.100	0.310	0.510	22.39	13.58	36.5	356.9	352.4	143.00
[66]	H10-4-72.5	Hollow	0.400	0.600	0.100	0.310	0.510	16.63	9.83	36.5	444.5	447.9	127.00
	H10-5-80	Hollow	0.400	0.600	0.100	0.310	0.510	14.41	8.91	36.5	546.3	538.6	125.00
	H06-3-90	Hollow	0.400	0.600	0.100	0.310	0.510	13.30	7.92	36.5	359.1	351.0	102.00
	H10-5-135	Hollow	0.400	0.600	0.100	0.310	0.510	9.02	5.28	36.5	548.1	540.3	95.00
	H08-3-72.5	Hollow	0.400	0.600	0.100	0.310	0.510	16.63	9.83	36.5	359.4	352.7	114.00
H12-5-72.5	Hollow	0.400	0.600	0.100	0.310	0.510	16.63	9.83	36.5	404.1	538.7	129.00	

References

- Rahal, K.N. Torsional strength of normal and high strength reinforced concrete beams. *Eng. Struct.* **2013**, *56*, 2206–2216. [\[CrossRef\]](#)
- Teixeira, M.M.; Bernardo, L.F.A. Torsional strength of reinforced concrete beams—Evaluation of some codes of practice. Concrete structures: New trends for eco-efficiency and performance. In Proceedings of the Fib Symposium, Lisbon, Portugal, 14–16 June 2021; pp. 2035–2046.
- Bernardo, L.F.A.; Teixeira, M.M.; De Domenico, D.; Gama, J.M.R. Improved Equations for the Torsional Strength of Reinforced Concrete Beams for Codes of Practice Based on the Space Truss Analogy. *Materials* **2022**, *15*, 3827. [\[CrossRef\]](#)
- Whittle, R. *Failures in Concrete Structures—Case Studies in Reinforced and Prestressed Concrete*; FL 33487-2742; CRC Press: Boca Raton, FL, USA; Taylor & Francis Group: Abingdon, UK, 2013; p. 140.
- Hsu, T.T.C. *Torsion of Structural Concrete—Behavior of Reinforced Concrete Rectangular Members; Torsion of Structural Concrete, SP-18*; American Concrete Institute: Detroit, MI, USA, 1968; Volume 18, pp. 261–306.
- McMullen, A.E.; Rangan, B.V. Pure Torsion in Rectangular Sections—A Re-examination. *ACI J. Proc.* **1978**, *75*, 511–519.
- ACI Committee 318. *Building Code Requirements for Structural Concrete (ACI 318-89) and Commentary (ACI 318R-89)*; American Concrete Institute: Detroit, MI, USA, 1989.
- SP 63.13330.2018; Concrete and Reinforced Concrete Structures. Ministry of Regional Development of the Russian Federation: Moscow, Russia, 2018.

9. ACI Committee 318. *Building Code Requirements for Structural Concrete (ACI 318-19) and Commentary (ACI 318R-19)*; American Concrete Institute: Detroit, MI, USA, 2019.
10. *CAN/CSA-A23.3-14; Design of Concrete Structures*. Canadian Standards Association: Toronto, ON, Canada, 2014.
11. *CEB-FIP Model Code 2010*; Comité Euro-International du Béton. Secretariat Permanent: Lausanne, Switzerland, 2013.
12. *EN 1992-1-1; Eurocode 2: Design of Concrete Structures—Part 1–1: General Rules and Rules for Buildings*. European Committee for Standardization: Brussels, Belgium, 2005; p. 225.
13. Rausch, E. Berechnung des Eisenbetons gegen Verdrehung (Design of Reinforced Concrete in Torsion). Ph.D. Thesis, Technische Hochschule, Berlin, Germany, 1929; p. 53. (In German)
14. Cowan, H.J. Elastic Theory for Torsional Strength of Rectangular Reinforced Concrete Beams. *Mag. Concr. Res.* **1950**, *2*, 3–8. [[CrossRef](#)]
15. Lampert, P.; Thurlimann, B. *Torsionsversuche an Stahlbetonbalken (Torsion Tests of Reinforced Concrete Beams)*; Bericht, Nr. 6506-2; Institut für Baustatik, ETH: Zurich, Switzerland, 1969; p. 101. (In German)
16. Mitchell, D.; Collins, M.P. Diagonal Compression Field Theory—A Rational Model for Structural Concrete in Pure Torsion. *ACI Struct. J.* **1974**, *71*, 396–408.
17. Leonhardt, F.; Schelling, G. *Torsionsversuche an Stahl Betonbalken*; Bulletin No. 239; Deutscher Ausschuss für Stahlbeton: Berlin, Germany, 1974; p. 122. (In German)
18. Jeng, C.-H.; Hsu, T.T.C. A Softened Membrane Model for Torsion in Reinforced Concrete Members. *Eng. Struct.* **2009**, *31*, 1944–1954. [[CrossRef](#)]
19. Bernardo, L.F.A.; Andrade, J.M.A. A Unified Softened Truss Model for RC and PC Concrete Beams under Torsion. *J. Build. Eng.* **2020**, *32*, 101467. [[CrossRef](#)]
20. Bhatti, M.A.; Almughrabi, A. Refined model to estimate torsional strength of reinforced concrete beams. *J. Am. Concr. Inst.* **1996**, *93*, 614–622.
21. Wang, W.; Hsu, T.T.C. Limit analysis of reinforced concrete beams subjected to pure torsion. *J. Struct. Eng.* **1997**, *123*, 86–94. [[CrossRef](#)]
22. Silva, J.R.B.; Horowitz, B.; Bernardo, L.F.A. Efficient procedure to analyze RC beam sections using the softened truss model. *ACI Struct. J.* **2017**, *114*, 765–774. [[CrossRef](#)]
23. Greene, G.G.; Belarbi, A. Model for RC members under torsion, bending, and shear. I: Theory. *J. Eng. Mech.* **2009**, *135*, 961–969.
24. Alnauimi, A.S.; Bhatt, P. Direct design of hollow reinforced concrete beams. Part I: Design procedure. *Struct. Concr.* **2004**, *5*, 139–146. [[CrossRef](#)]
25. Bairan Garcia, J.M.; Mari Bernat, A.R. Shear-bending-torsion interaction in structural concrete members: A non-linear coupled sectional approach. *Arch. Comput. Methods Eng.* **2007**, *14*, 249–278. [[CrossRef](#)]
26. Bernardo, L.; Taborda, C. Softened Truss Model for Reinforced Concrete Beams under Torsion Combined with Axial Force. *Appl. Mech.* **2020**, *1*, 79–96. [[CrossRef](#)]
27. Salem, N.M.; Deifalla, A. Evaluation of the Strength of Slab–Column Connections with FRPs Using Machine Learning Algorithms. *Polymers* **2022**, *14*, 1517. [[CrossRef](#)]
28. Ebid, A.; Deifalla, A. Using Artificial Intelligence Techniques to Predict Punching Shear Capacity of Lightweight Concrete Slabs. *Materials* **2022**, *15*, 2732. [[CrossRef](#)]
29. Ebid, A.; Deifalla, A. Prediction of Shear Strength of FRP Reinforced Beams with and Without Stirrups Using (GP) Technique. *Ain Shams Eng. J.* **2021**, *12*, 2493–2510. [[CrossRef](#)]
30. Shatnawi, A.; Alkassar, H.M.; Al-Abdaly, N.M.; Al-Hamdany, E.A.; Bernardo, L.F.A.; Imran, H. Shear Strength Prediction of Slender Steel Fiber Reinforced Concrete Beams Using a Gradient Boosting Regression Tree Method. *Buildings* **2022**, *12*, 550. [[CrossRef](#)]
31. Amani, J.; Moeini, R. Prediction of shear strength of reinforced concrete beams using adaptive neuro-fuzzy inference system and artificial neural network. *Sci. Iran. A* **2012**, *19*, 242–248. [[CrossRef](#)]
32. Wakjira, T.G.; Al-Hamran, A.; Ebead, U.; Alnahhal, W. Shear capacity prediction of FRP-RC beams using single and ensemble Explainable Machine learning models. *Compos. Struct.* **2022**, *287*, 115381. [[CrossRef](#)]
33. Jin, J.; Wu, T.; Zhang, Z.; Fan, J.; Lee, X. Application of optimization-based estimation analysis for predicting the compressive strength of eco-friendly modified geopolymer concrete. *Struct. Concr.* **2022**, *2022*, 1–23. [[CrossRef](#)]
34. Wakjira, T.G.; Rahmzadeh, A.; Alam, M.S.; Tremblay, R. Explainable machine learning based efficient prediction tool for lateral cyclic response of post-tensioned base rocking steel bridge piers. *Structures* **2022**, *44*, 947–964. [[CrossRef](#)]
35. Wakjira, T.G.; Ebead, U.; Alam, M.S. Machine learning-based shear capacity prediction and reliability analysis of shear-critical RC beams strengthened with inorganic composites. *Case Stud. Constr. Mater.* **2022**, *16*, e01008. [[CrossRef](#)]
36. Wakjira, T.G.; Ibrahim, M.; Ebead, U.; Alam, M.S. Explainable machine learning model and reliability analysis for flexural capacity prediction of RC beams strengthened in flexure with FRCM. *Eng. Struct.* **2022**, *255*, 113903. [[CrossRef](#)]
37. Benemaran, R.S.; Esmaeili-Falak, M.; Javadi, A. Predicting resilient modulus of flexible pavement foundation using extreme gradient boosting based optimized models. *Int. J. Pavement Eng.* **2022**. [[CrossRef](#)]
38. Zhu, Y.; Huang, L.; Zhang, Z.; Bayrami, B. Estimation of splitting tensile strength of modified recycled aggregate concrete using hybrid algorithms. *Steel Compos. Struct.* **2022**, *44*, 389–406.
39. Thai, H.-T. Machine learning for structural engineering: A state-of-the-art review. *Structures* **2022**, *38*, 448–491. [[CrossRef](#)]

40. Lyu, Z.; Yu, Y.; Samali, B.; Rashidi, M.; Mohammadi, M.; Nguyen, T.N.; Nguyen, A. Back-Propagation Neural Network Optimized by K-Fold Cross-Validation for Prediction of Torsional Strength of Reinforced Concrete Beam. *Materials* **2022**, *15*, 1477. [[CrossRef](#)]
41. Huang, H.C. Using a hybrid neural network to predict the torsional strength of reinforced concrete beams. *Adv. Mat. Res.* **2012**, *538*, 2749–2753. [[CrossRef](#)]
42. Cevik, A.; Arslan, M.H.; Köroglu, M.A. Genetic-programming-based modeling of RC beam torsional strength. *KSCE J. Civ. Eng.* **2010**, *14*, 371–384. [[CrossRef](#)]
43. Arslan, M.H. Predicting of torsional strength of RC beams by using different artificial neural network algorithms and building codes. *Adv. Eng. Softw.* **2010**, *41*, 946–955. [[CrossRef](#)]
44. Deifalla, A.; Salem, N.M. A Machine Learning Model for Torsion Strength of Externally Bonded FRP- Reinforced Concrete Beams. *Polymers* **2022**, *14*, 1824. [[CrossRef](#)] [[PubMed](#)]
45. Naderpour, H.; Fakharian, P. Predicting the torsional strength of reinforced concrete beams strengthened with FRP sheets in terms of artificial neural networks. *J. Struct. Constr. Eng.* **2018**, *5*, 20–35.
46. Amini Pishro, A.; Zhang, Z.; Amini Pishro, M.; Liu, W.; Zhang, L.; Yang, Q. Structural Performance of EB-FRP-Strengthened RC T-Beams Subjected to Combined Torsion and Shear Using ANN. *Materials* **2022**, *15*, 4852. [[CrossRef](#)] [[PubMed](#)]
47. Henedy, S.N.; Naser, A.H.; Imran, H.; Bernardo, L.F.A.; Teixeira, M.M.; Al-Khafaji, Z. Development of Prediction Models for the Torsion Capacity of Reinforced Concrete Beams Using M5P and Nonlinear Regression Models. *J. Compos. Sci.* **2022**, *6*, 366. [[CrossRef](#)]
48. Yu, Y.; Liang, S.; Samali, B.; Nguyen, T.N.; Zhai, C.; Li, J.; Xie, X. Torsional capacity evaluation of RC beams using an improved bird swarm algorithm 2D convolutional neural network. *Eng. Struct.* **2022**, *273*, 115066. [[CrossRef](#)]
49. Feng, D.-C.; Wang, W.-J.; Mangalathu, S.; Hu, G.; Wu, T. Implementing ensemble learning methods to predict the shear strength of RC deep beams with/without web reinforcements. *Eng. Struct.* **2021**, *235*, 111979. [[CrossRef](#)]
50. Rahman, J.; Ahmed, K.S.; Khan, N.I.; Islam, K.; Mangalathu, S. Data-driven shear strength prediction of steel fiber reinforced concrete beams using machine learning approach. *Eng. Struct.* **2021**, *233*, 111743. [[CrossRef](#)]
51. Zhang, G.; Ali, Z.H.; Aldlemy, M.S.; Mussa, M.H.; Salih, S.Q.; Hameed, M.M.; Al-Khafaji, Z.S.; Yaseen, Z.M. Reinforced concrete deep beam shear strength capacity modelling using an integrative bio-inspired algorithm with an artificial intelligence model. *Eng. Comput.* **2022**, *38*, 15–28. [[CrossRef](#)]
52. Sanad, A.; Saka, M.P. Prediction of Ultimate Shear Strength of Reinforced-Concrete Deep Beams Using Neural Networks. *J. Struct. Eng.* **2001**, *127*, 818–828. [[CrossRef](#)]
53. Chabib, H.E.; Nehdi, M.; Said, A. Predicting the effect of stirrups on shear strength of reinforced normal-strength concrete (NSC) and high-strength concrete (HSC) slender beams using artificial intelligence. *Can. J. Civ. Eng.* **2006**, *33*, 933–945. [[CrossRef](#)]
54. Elsanadedy, H.M.; Abbas, H.; Al-Salloum, Y.A.; Almusallam, T.H. Shear strength prediction of HSC slender beams without web reinforcement. *Mater. Struct.* **2016**, *49*, 3749–3772. [[CrossRef](#)]
55. Kotu, V.; Deshpande, B. *Data Science: Concepts and Practice*; Morgan Kaufmann Publishers: Cambridge, MA, USA, 2019.
56. Peng, X.-N.; Wong, Y.-L. Behaviour of reinforced concrete walls subjected to monotonic pure torsion—An experimental study. *Eng. Struct.* **2011**, *33*, 2495–2508. [[CrossRef](#)]
57. Lampert, P.; Thürlimann, B. Essais de Poutre en Béton Armé sous Torsion Simple et Flexion Combinées (Torsionsversuche und Torsions-Biege-Versuche an Stahlbetonbalken). *Com. Eur. Béton BI* **1969**, *71*, 177–207. (In German)
58. Rasmussen, L.J.; Baker, G. Torsion in reinforced normal and high-strength concrete beams—Part 1: Experimental test series. *ACI Struct. J.* **1995**, *92*, 56–62.
59. Koutchoukali, N.-E.; Belarbi, A. Torsion of high-strength reinforced concrete beams and minimum reinforcement requirement. *ACI Struct. J.* **2001**, *98*, 462–469.
60. Fang, I.-K.; Shiau, J.-K. Torsional behavior of normal- and high-strength concrete beams. *ACI Struct. J.* **2004**, *101*, 304–313.
61. Chiu, H.-J.; Fang, I.-K.; Young, W.-T.; Shiau, J.-K. Behavior of reinforced concrete beams with minimum torsional reinforcement. *Eng. Struct.* **2007**, *29*, 2193–2205. [[CrossRef](#)]
62. Lee, J.-Y.; Kim, S.-W. Torsional strength of RC beams considering tension stiffening effect. *J. Struct. Eng.* **2010**, *136*, 1367–1378. [[CrossRef](#)]
63. Joh, C.; Kwahk, I.; Lee, J.; Yang, I.-H.; Kim, B.-S. Torsional behavior of high-strength concrete beams with minimum reinforcement ratio. *Adv. Civ. Eng.* **2019**, *2019*, 1432697. [[CrossRef](#)]
64. Ju, H.; Han, S.-J.; Zhang, D.; Kim, J.; Wu, W.; Kim, K.S. Estimation of minimum torsional reinforcement of reinforced concrete and steel fiber-reinforced concrete members. *Adv. Mater. Sci. Eng.* **2019**, *2019*, 4595363. [[CrossRef](#)]
65. Ibrahim, M.S.; Gebreyouhannes, E.; Muhdin, A.; Gebre, A. Effect of concrete cover on the pure torsional behavior of reinforced concrete beams. *Eng. Struct.* **2020**, *216*, 110790. [[CrossRef](#)]
66. Kim, M.-J.; Kim, H.-G.; Lee, Y.-J.; Kim, D.-H.; Lee, J.-Y.; Kim, K.-H. Pure torsional behavior of RC beams in relation to the amount of torsional reinforcement and cross-sectional properties. *Constr. Build. Mater.* **2020**, *260*, 119801. [[CrossRef](#)]
67. Bernardo, L.F.A.; Lopes, S.M.R. Torsion in HSC hollow beams: Strength and ductility analysis. *ACI Struct. J.* **2009**, *106*, 39–48.
68. Jeng, C.-H. Unified softened membrane model for torsion in hollow and solid reinforced concrete members modeling the entire pre- and post-cracking behavior. *J. Struct. Eng.* **2015**, *141*, 1–20. [[CrossRef](#)]
69. Farrar, D.E.; Glauber, R.R. Multicollinearity in Regression Analysis: The Problem Revisited. *Rev. Econ. Stat.* **1967**, *49*, 92–107. [[CrossRef](#)]

70. Gholamy, A.; Kreinovich, V.; Kosheleva, O. *Why 70/30 or 80/20 Relation between Training and Testing Sets: A Pedagogical Explanation*; Technical Report: UTEP-CS-18-09; The University of Texas: El Paso, TX, USA, 2018; p. 6.
71. Hastie, T.; Tibshirani, R.; Friedman, J.H.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 2.
72. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
73. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [[CrossRef](#)]
74. Alhakeem, Z.M.; Jebur, Y.M.; Henedy, S.N.; Imran, H.; Bernardo, L.F.A.; Hussein, H.M. Prediction of Ecofriendly Concrete Compressive Strength Using Gradient Boosting Regression Tree Combined with GridSearchCV Hyperparameter-Optimization Techniques. *Materials* **2022**, *15*, 7432. [[CrossRef](#)]
75. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
76. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.