

## Article

# Machine Learning Models for Ecofriendly Optimum Design of Reinforced Concrete Columns

Yaren Aydın <sup>1</sup>, Gebrail Bekdaş <sup>1,\*</sup>, Sinan Melih Nigdeli <sup>1</sup>, Ümit Isıkdağ <sup>2</sup>, Sanghun Kim <sup>3</sup>  
and Zong Woo Geem <sup>4,\*</sup>

<sup>1</sup> Department of Civil Engineering, Istanbul University-Cerrahpaşa, 34320 Istanbul, Turkey

<sup>2</sup> Department of Informatics, Mimar Sinan Fine Arts University, 34427 Istanbul, Turkey

<sup>3</sup> Department of Civil and Environmental Engineering, Temple University, Philadelphia, PA 19122, USA

<sup>4</sup> Department of Smart City & Energy, Gachon University, Seongnam 13120, Republic of Korea

\* Correspondence: bekdas@iuc.edu.tr (G.B.); geem@gachon.ac.kr (Z.W.G.)

**Abstract:** CO<sub>2</sub> emission is one of the biggest environmental problems and contributes to global warming. The climatic changes due to the damage to nature is triggering a climate crisis globally. To prevent a possible climate crisis, this research proposes an engineering design solution to reduce CO<sub>2</sub> emissions. This research proposes an optimization-machine learning pipeline and a set of models trained for the prediction of the design variables of an ecofriendly concrete column. In this research, the harmony search algorithm was used as the optimization algorithm, and different regression models were used as predictive models. Multioutput regression is applied to predict the design variables such as section width, height, and reinforcement area. The results indicated that the random forest algorithm performed better than all other machine learning algorithms that have also achieved high accuracy.

**Keywords:** reinforced concrete; optimization; predictive modeling; carbon emission; harmony search



**Citation:** Aydın, Y.; Bekdaş, G.; Nigdeli, S.M.; Isıkdağ, Ü.; Kim, S.; Geem, Z.W. Machine Learning Models for Ecofriendly Optimum Design of Reinforced Concrete Columns. *Appl. Sci.* **2023**, *13*, 4117. <https://doi.org/10.3390/app13074117>

Academic Editor: Muhammad Junaid Munir

Received: 23 February 2023

Revised: 21 March 2023

Accepted: 22 March 2023

Published: 23 March 2023



**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

Air pollution, population growth, and urban development become increasingly important with industrialization. Global warming and climate change are the most important factors affecting all human beings. The warming of the earth's atmosphere due to the release of greenhouse gases (GHGs) (CO<sub>2</sub>, CH<sub>4</sub>, NO<sub>x</sub>) is one of the main causes of climate change. Carbon dioxide (CO<sub>2</sub>) increases the greenhouse effect much faster than other GHGs due to its longevity. Due to inefficient CO<sub>2</sub> absorption, an imbalance in the carbon cycle is further increased [1]. Therefore, carbon dioxide emissions need to be controlled.

The past eight years (2015–2022) were the warmest on record globally, fueled by ever-rising greenhouse gas concentrations and accumulated heat [2]. Extreme heatwaves, drought, and devastating flooding have affected millions of people [3]. The past nine years have been the warmest since modern recordkeeping began in 1880 [4].

People have built structures for various purposes throughout history. Concrete has been preferred as an important structural material for many years because it can be shaped easily, is resistant to physical and chemical external effects, is economical, and is practical to use and produce [5]. In addition to providing safety with engineering design, it is an advantage to have minimum cost or the least damage to the environment [6].

Reinforced concrete (RC) is a material obtained as a result of placing reinforcements in concrete. Concrete and reinforcement are used together to eliminate each other's weaknesses, provided that adherence is ensured. In this building material, reinforcement is effective in counteracting tensile effects and makes a very important contribution to the ductile behavior of the material. Concrete that is effective in supporting compressive stresses, prevents the reinforcement from buckling by wrapping it around and increases the resistance of the material against the external environment and fire resistance [7].

Table 1 shows the strengths of concrete grades between C16 and C50. However, according to TBDY 2018 [8], concrete with lower strength than C25 grade cannot be used in new buildings that will be built in Turkey.

**Table 1.** Concrete grades and strength [9].

Concrete Grades	Characteristic Compressive Strength, $f_{ck}$ MPa	Equivalent Cube (200 mm) Compressive Strength MPa	Characteristic Axial Tension Strength, $f_{ctk}$ MPa	28-Day Elastic Modulus, $E_c$ MPa
C16	16	20	1.4	27,000
C18	18	22	1.5	27,500
C20	20	25	1.6	28,000
C25	25	30	1.8	30,000
C30	30	37	1.9	32,000
C35	35	45	2.1	33,000
C40	40	50	2.2	34,000
C45	45	55	2.3	36,000
C50	50	60	2.5	37,000

Concrete consumes more water than any other material. The concrete industry is a major consumer of freshwater [10,11]. Concrete is formed by homogeneously mixing cement, aggregate (sand + gravel), water, and, when necessary, chemical and mineral additives in appropriate proportions [7]. The major carbon emission from the concrete industry is from the production of its main binder, which is Portland cement [10].

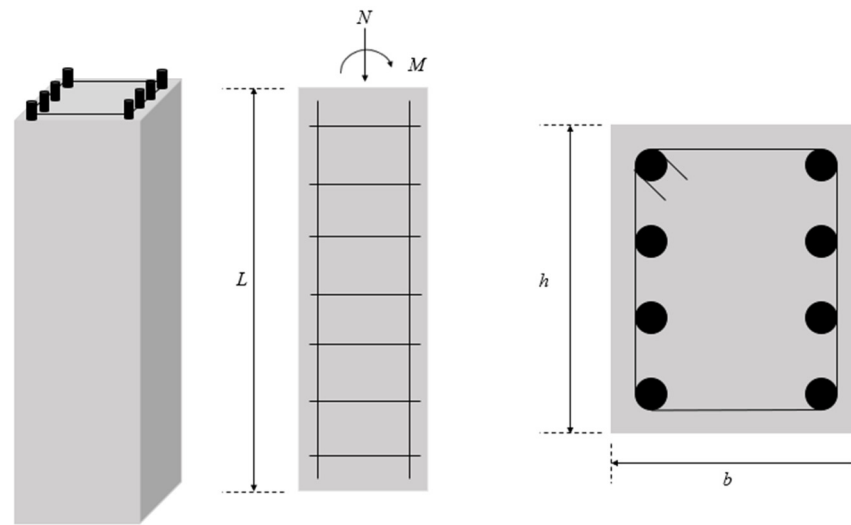
As shown in Table 2, China is the leading cement producer in the world. Cement production in China was approximately 2.35 billion tons in 2016. After China, India follows. Approximately 8% of the carbon dioxide emitted to the world originates from cement [12]. The cement industry, if considered a country, would be the third-largest emitter in the world after China and the US [13].

**Table 2.** Cement production by countries (million tons) [14].

Country	2015	2016	2017	2018	2019	2020
China	2350	2403	2320	2370	2300	2200
India	270	290	290	330	320	290
USA	83.4	84.7	86.1	87.8	88.6	89
Brazil	72	57.8	54	53.5	53.4	60
Turkey	71.4	75.4	80.6	72.5	57	72.3
Russia	69	56	54.7	53.7	54.1	56
Indonesia	65	61.3	68	70.8	64.2	64.8
South Korea	63	56.7	57.9	55	56.4	50
Japan	55	3.4	55.5	55.3	55.2	52.1
Saudi Arabia	55	55.9	47.1	42.2	52.2	53.4
Germany	31.1	32.7	34	33.7	34.2	35.5
Italy	20.8	19.3	19.3	19.3	19.2	18.1

In a structure, a column carries axial loading as compression. The most ideal columns to be used in the design are circular and rectangular columns because when earthquake force acts on them, their behavior can be predicted by engineers. Rectangular columns are commonly used in the construction of buildings and heavy structures. It is easier to construct and cast this type of column. Square and rectangular columns gain less strength and ductility than circular columns because the lateral confinement for pressure distribution in the circular section is uniform, but the stress distribution of square and rectangular columns varies from maximum to minimum at the corner [15].

Figure 1 shows a general description of an RC rectangular column where the section width and height, and the lateral and longitudinal reinforcements can be seen. The section width ( $b$ ), the section height ( $h$ ), and the total length of the column ( $L$ ) describe the geometry of the column. In the study, columns under uniaxial bending and axial load were examined, and the bending axis has been taken as similar in all cases for comparison purposes.



**Figure 1.** Geometry of the RC column.

Recently, machine learning (ML) has been used in the field of civil engineering. Rajakarunakaran et al. [16] intended to create machine learning-based regression models to predict self-compacting concrete compressive strength. The results showed that the random forest model forecasts concrete's compressive strength accurately. Deifalla and Salem [17] introduced an ML model for calculating the ultimate torsion strength of concrete beams strengthened by using externally bonded fiber-reinforced polymer. The model showed improved agreement and consistency with experimental results compared to existing models in the literature. Dissanayake et al. [18] presented the application of popular ML algorithms in the prediction of the shear resistance of the steel channel sections. The results indicated that the implemented ML models exceed the prediction accuracy of the available design equations in estimating the shear capacity of the steel channel section. Amjad et al. [19] developed a new model for predicting bearing capacity by using an extreme gradient boosting (XGBoost) algorithm. The results showed that the XGBoost algorithm has the most accurate predictions for all models developed. Aydın et al. [20] aimed to explore new ML methods to automatically classify soil for minimizing the time and cost of the classification process. The results indicated that tree-based foundational methods/classifiers, such as the decision tree classifier, and gradient boosting-based ensemble methods provide very good performance.

The damage to nature due to global warming has forced researchers to think of an environmentally friendly design. As indicated in the literature, various studies have been carried out for reducing CO<sub>2</sub> emissions of reinforced concrete system design by using different datasets and various optimization and ML methods. Kayabekir et al. [21] investigated the sustainable design (minimum cost and CO<sub>2</sub> emissions) of the RC retaining walls by using the harmony search (HS) algorithm. Their proposed approach performs well with regard to economic and ecological results. Zhu et al. [22] showed that variations in the span and load of RC slabs can change their environmental sustainability. The research results indicated that composite slabs were widely recommended in engineering applications from the view of environmental sustainability. Wang et al. [23] investigated the concrete manufacturing process's potential effect on global warming. According to the results, while under different functional units, the environmental performance of composite and cast-in-situ floors varies. Paik and Seunguk [24] investigated the effect of using a void deck slab (VDS) system instead of an ordinary reinforced concrete slab on CO<sub>2</sub> emissions. The results revealed the total CO<sub>2</sub> emissions of an ordinary RC slab are more than those of the voided slab system. In large-scale construction, Purnell [25] found that RC beams designed with optimized strength concrete show significantly lower embodied carbon of a structural component expressed in terms of its structural performance values than

comparable steel or timber composite beams over the entire range of permissible concrete section sizes. Destrée and Pease [26] compared PrimeComposite, a steel fiber reinforced concrete (SFRC) with proprietary additives, to conventional graded slab systems. Results showed that CO<sub>2</sub> emissions are reduced by no less than 40% by replacing traditional concrete slab systems with PrimeComposite. Yepes et al. [27] applied optimization designs based on the CO<sub>2</sub> efficiency and cost design for RC retaining walls. The analysis showed that reducing costs by 1 euro could save up to 2.28% kg in CO<sub>2</sub> emissions. Bekdaş et al. [6] proposed a modified HS methodology for the optimization of RC beams with minimum CO<sub>2</sub> emissions, and the results showed that the optimum design based on CO<sub>2</sub> emission minimization and optimum cost design results are different. Arama et al. [28] presented the parametric modeling process of soldier pile walls based on CO<sub>2</sub> and cost optimization with the HS algorithm. Optimization analyses showed the attainment of both cost and CO<sub>2</sub> emission minimization.

In the present study, a hybrid model was proposed for optimization and prediction. RC rectangular column section dimensions are optimized for CO<sub>2</sub> emission minimization with the HS algorithm. Then, the generated data were used in the development of different ML models that predict optimum results without a rerun of the optimization process.

## 2. Materials and Methods

The overall process of ecofriendly structural design is handled in two stages. The first stage is the dataset generation through the optimization process, and the second stage is the machine learning process implemented to predict  $b$ ,  $h$ , and  $A_s$  values from  $M$  and  $N$ , where the bending moment is denoted as ( $M$ ) and axial force is denoted as ( $N$ ), the column section width is denoted as ( $b$ ), the column section height is denoted as ( $h$ ), and the reinforcement area of the section is denoted as ( $A_s$ ).

### 2.1. Optimization Process and Dataset Generation

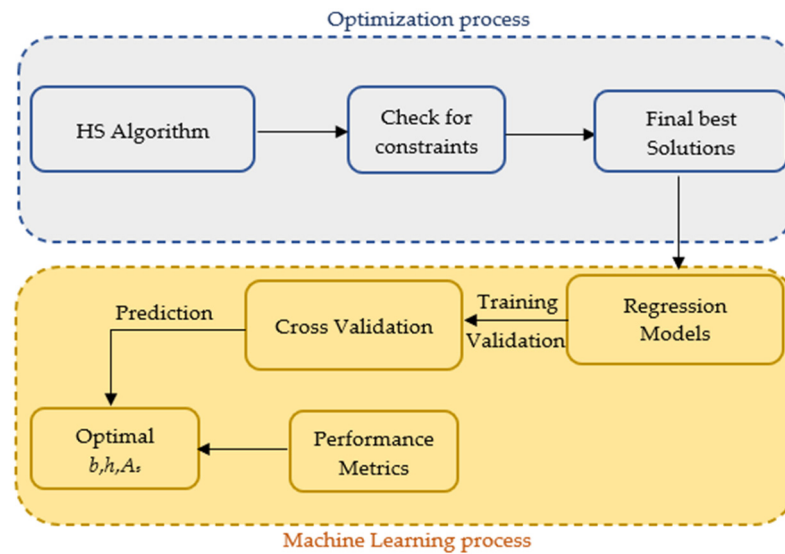
The main objective of this study is to minimize CO<sub>2</sub> emissions through the efficient structural design of RC columns. For this purpose, the HS algorithm [29] was implemented to determine the optimal RC rectangular column dimensions ( $b$  and  $h$ ) and total area of the longitudinal reinforcement ( $A_s$ ) for a given  $M$  and  $N$ . A large dataset was generated, including the optimum results of several loading cases.

The HS algorithm is employed for the optimization process via MATLAB [30] code and is a population-based metaheuristic algorithm developed by Geem et al. [29], derived from an artificial phenomenon found in musical performance. The HS algorithm, which imitates the musical best-fit search, has been applied to many civil engineering optimization problems including posttensioned RC walls [31], earthquake analysis [32], structural vibration control [33–37], retaining walls [38–41], truss structures [42,43], RC columns and beams [44,45].

Figure 2 shows a flowchart of the proposed method for optimization and prediction. In the optimization, the design variables were optimized for the minimization of the objective function. Since the problem is nonlinear because of design constraints, constraints are checked in the optimization process. As considered in previous studies [45,46], these constraints are related to stress limitations and min–max requirements defined in design codes.

The concrete compressive strength, steel yield strength, steel unit weight, elastic modulus of steel, unit shortening of concrete at fracture, and unit CO<sub>2</sub> emissions of the materials and HS parameters are defined. Moreover, the solution ranges for RC rectangular dimensions and reinforcement area were determined. The objective function of this optimization problem is given in Equation (1):

$$\min f(\text{CO}_2) = C_{C,\text{CO}_2} \times V_s + C_{S,\text{CO}_2} \times W_s. \quad (1)$$



**Figure 2.** Flowchart of the proposed method for optimization and prediction.

In Equation (1),  $C_{C,CO_2}$ ,  $C_{S,CO_2}$ ,  $V_C$ , and  $W_s$  are the concrete carbon dioxide emission per unit volume, steel carbon dioxide emission per unit weight, total volume of concrete, and the total weight of steel, respectively. The search space for this function will be between the minimum and maximum values of the variables that can take. In the solution of an optimization problem with HS, the initial harmony memory ( $HM$ ) matrix is randomly generated within a predetermined search space, and the size of the solution candidate population is harmony memory size ( $HMS$ ). This matrix contains harmony vectors ( $HV$ ), and each design variable is randomized within the defined solution ranges. Accordingly, if the  $HMS$  is  $m$ ,  $m$  random solution candidates are created between the specified values as follows and stored in the  $HM$  matrix as shown in Equation (2). Here, each row represents a design and each column represents a design variable:

$$HM = \begin{bmatrix} b^1 & h^1 & A_s^1 & L^1 & M^1 & N^1 & C_{C,CO_2}^1 & C_{S,CO_2}^1 & f(x^1) \\ b^2 & h^2 & A_s^2 & L^2 & M^2 & N^2 & C_{C,CO_2}^2 & C_{S,CO_2}^2 & f(x^2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b^m & h^m & A_s^m & L^m & M^m & N^m & C_{C,CO_2}^m & C_{S,CO_2}^m & f(x^m) \end{bmatrix}. \quad (2)$$

The function value is calculated by replacing each row of the  $HM$  matrix in the function to be minimized. In Equation (2),  $L$ ,  $M$ ,  $N$ , and  $A_s$  are the length of the column, bending moment, axial force, and the total area of the longitudinal reinforcement in the  $i$ -th solution candidate, respectively.

Then, a new solution candidate is created by using  $HM$ . The harmony vectors of the new solution candidate are determined according to harmony memory considering rate ( $HMCR$ ) shown in Equation (3). Pitch adjustment rate ( $PAR$ ) is used to modify the solution range to search for a new solution around existing ones.  $PAR$  is used similarly to fret width in the classical form of  $HS$ . With the following probabilities, each new harmony vector of the solution candidate is either randomly generated from the search space or created with random assignments between 0 and 1 values ( $rand()$ ), as shown in Equation (3).  $k$  is a randomly chosen existing solution defined by Equation (4). The equation for  $HMCR$  and  $PAR$  used in this paper can be found in Appendix A. We have

$$X_{i,new} = \begin{cases} X_{i,min} + rand()(X_{i,max} - X_{i,min}) & \text{if } HMCR > r_1 \\ X_{i,k} + rand()PAR(X_{i,max} - X_{i,min}) & \text{if } HMCR \leq r_1 \end{cases} \quad (3)$$

$$k = ceil(rand \times HMS). \quad (4)$$

Newly generated  $HV$  is replaced with the vector, which has the worst solution (maximum  $CO_2$  emission). The iterations are repeated for the maximum number of iterations. Thus, the solution matrix takes its final form.

2.2. Exploratory Data Analysis

The input data for the ML model were created by using the HS algorithm. By using the HS algorithm,  $b$ ,  $h$ , and  $A_s$  values that minimize the total  $CO_2$  emission were obtained for each combination of  $M$  and  $N$ . Thus, a dataset of 4429 configurations has been generated through HS. Descriptive statistics of features and outputs of the dataset are illustrated in Table 3.

Table 3. Descriptive and statistical features of the inputs and outputs.

Variable	Description	Data Type	Min	Max	Mean	Standard Deviation
Inputs:						
$M$ [kN·m]	Bending moment	float64 *	100.003	399.806	237.550	81.593
$N$ [kN]	Axial force	float64	1000.298	399.929	2387.248	831.938
Outputs:						
$b$ [mm]	Cross-section width	float64	250	266.841	250.551	2.372
$h$ [mm]	Cross-section height	float64	310.392	1000	646.801	202.609
$A_s$ [mm <sup>2</sup> ]	Total reinforcement area	float64	2127.295	7016.328	3615.663	1010.983

\* float64: 64-bit double precision values [46].

The histogram and scatter plots of the dataset are shown in Figure 3. In the figure,  $M$  stands for bending moment,  $N$  stands for axial force,  $b$  stands for section width,  $h$  stands for section height, and  $A_s$  stands for reinforcement area. When the dataset generated by HS algorithm in Figure 3 is examined, it is seen that the  $M$ ,  $N$ ,  $h$ , and  $A_s$  variables (normal-like) are distributed. Moreover,  $b$  takes a value of about 250 mm. The optimization aims to minimize the width and find an optimum value for the height. In that situation, the width is always smaller than the optimum height. For that reason, all sections are under bending on weak direction.

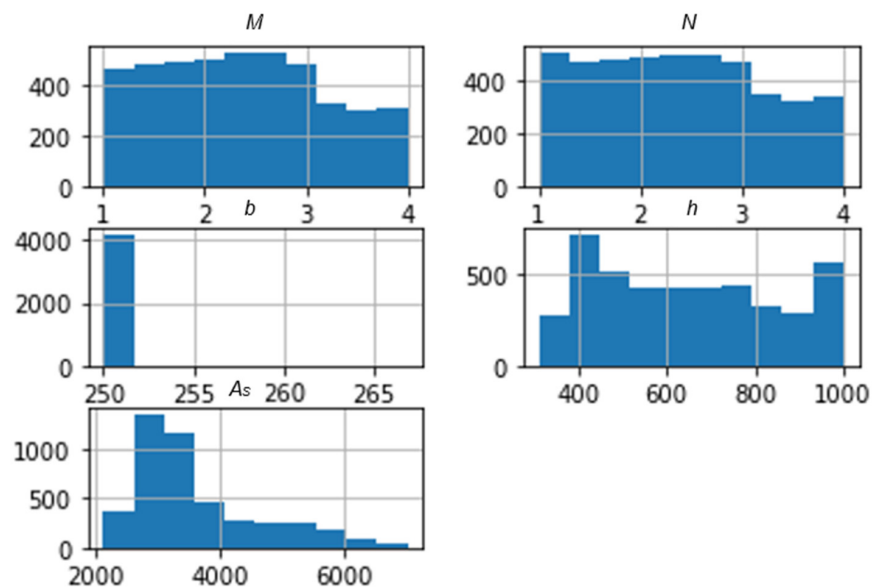
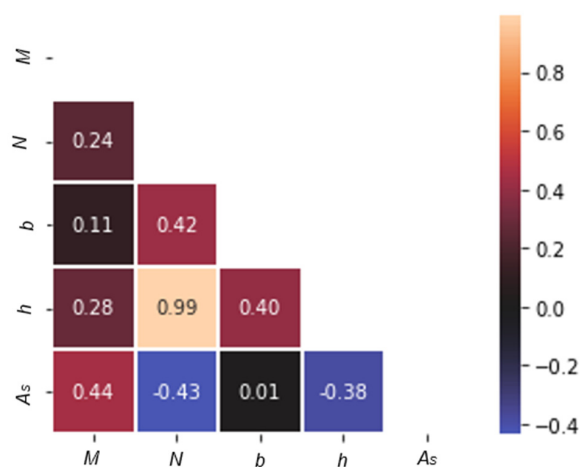


Figure 3. The histogram plots of the dataset.

By using the Seaborn [47] library of Python [48], the correlation matrix in Figure 4 was created. As the color becomes lighter, the positive correlation between the variables increases. When the correlation matrix in Figure 4 is examined, it is shown that the



correlation coefficients between bending moment ( $M$ ) and axial force ( $N$ ), column section width ( $b$ ), column section height ( $h$ ), and reinforcement area ( $A_s$ ) are positive and significant. The variables with the highest correlation between them were  $N$  and  $h$ . However, when a comparison is made between  $N$  and  $b$  and  $N$  and  $h$ , respectively, it is observed that the axial force has a higher significance level than the bending moment. This means that the axial force has a higher significance level than the bending moment in explaining  $b$  and  $h$ , as mentioned in the scatter plot. The negative value indicated that there was a negative correlation between the variables. As the value increased in the negative direction, the correlation matrix increased in the negative direction. The negative correlation between  $N$  and  $A_s$  and  $h$  and  $A_s$  indicates that while the value of one of the two variables increases, the value of the other decreases. The variables with the lowest correlation between them were  $b$  and  $A_s$ .



**Figure 4.** Visualization of the correlation matrix between input and output variables.

### 2.3. Machine Learning Models

After preparing the dataset with the optimum dimensions, several ML models were tested to predict  $b$ ,  $h$ ,  $A_s$  from  $M$  and  $N$ . The ability to learn in humans is one of their most important features, and it makes humans different from other living things and machines. ML is a result of the desire of machines to learn like humans and make the best decisions. The result of learning can be used for recognition, prediction, and classification [49]. With the help of ML, more accurate and valid results can be obtained by making forward-looking predictions and planning based on known data. In ML, the written algorithm makes an inference by looking at the existing inputs. A key advantage of ML is that ML methods can examine large amounts of data to find patterns that might otherwise be missed [50].

ML has four subfields, including supervised learning, unsupervised learning, semisupervised learning, and reinforcement learning. Unsupervised learning is based on only input data without labels [51]. Supervised learning requires learning a mapping between a set of input variables and an output variable and applying this mapping to predict outputs for data whose outputs are unknown [52]. In semisupervised learning, in addition to the unlabeled data, some supervised information is provided to the algorithm, but this is not necessary for all examples [53]. In reinforcement learning, an agent is placed in an initially unknown environment and only receives evaluative feedback called reward [54].

In this study, the regression type of supervised method was used as the dependent variables were continuous. Regression is a problem of predicting a real value. Examples of regressions include predicting stock values or variations. In regression, the penalty for an incorrect estimate typically depends on the size of the difference between the actual and predicted values, unlike the classification problem, in which there is a closeness between the various categories [55]. The dataset used in this study was a multioutput dataset (i.e., containing multiple dependent variables). For the prediction process, we have utilized Python [48] as the language, Anaconda3 [56] as the environment, and Spyder 5.2.2 as the

editor. In addition to basic Python libraries such as Numpy and Pandas, the scikit-learn library [57] developed for ML applications was used for defining ML models.

ML algorithms used in the study will be explained under the title of foundational methods are linear regression, decision tree regression, elastic net regression, K nearest neighbors regression (KNN), and support vector regression (SVR). Other ML algorithms used in the study will be explained under the title of ensemble methods including random forest regression, gradient boosting regression, histogram-based gradient boosting regression, and voting and stacking.

### 2.3.1. Foundational Methods

The linear regression algorithm is a method used to establish the relationship between one or more independent variables and another dependent variable [58]. A decision tree is an efficient tool for the solution of classification and regression problems [59]. Decision tree regression recursively splits data into smaller parts by using a fast divide-and-conquer greedy algorithm [60]. Decision tree regression uses the decision tree learning tree as the base learner for the regression process. Elastic net is a regression method that does both variable selection and regularization. Regularization serves to solve the problem of model overfitting [61]. KNN regression, a nonparametric method, uses the information derived from the observed data for prediction [62]. SVR is used to minimize the inherent risk by minimizing an upper bound of the generalization error. This ensures that SVR has more potential to generalize the input–output relationship for new input data [63].

### 2.3.2. Ensemble Methods

Random forests (RFs) are effective in prediction and random forest (RF) regression is formed by growing trees depending on a random vector, such that the tree predictor takes on numerical values [64]. In ML, “boosting” is a way to combine multiple simple models into a single composite model. The term “gradient” in “gradient boosting” indicates that the algorithm uses gradient descent to minimize loss. Gradient boosting is used for regression, while gradient boosting is used to estimate a continuous value [65]. Histogram-based gradient boosting regression is much faster than the gradient boosting regressor for big datasets and has native support for missing values [66].

There are several ways to combine multiple learners to produce the final output in a model-building scheme. An example of using output produced by all core learners given input is voting and stacking [67]. These are ensemble methods.

With regard to the voting regressor, voting provides a basic way to classify ensembles. The default scheme averages the probability or numerical estimates for classification and regression, respectively [68]. Voting is used to get a linear combination of learners (Figure 5) [67]. A voting ensemble increases the system’s performance. It can be utilized for both classification and regression issues by integrating the results of numerous methods. For regression problems, voting regressors (VRs) are the estimators of all models that are averaged to obtain the final estimate [69]. The final estimate is less prone to error [70].

The idea in voting is to vote on models with high variance and low variance, ensuring that the variance remains small after combination and the variance is reduced by averaging. Even if the individual models are biased, decreasing the variance can balance this deviation and reduce the error [67]. We have

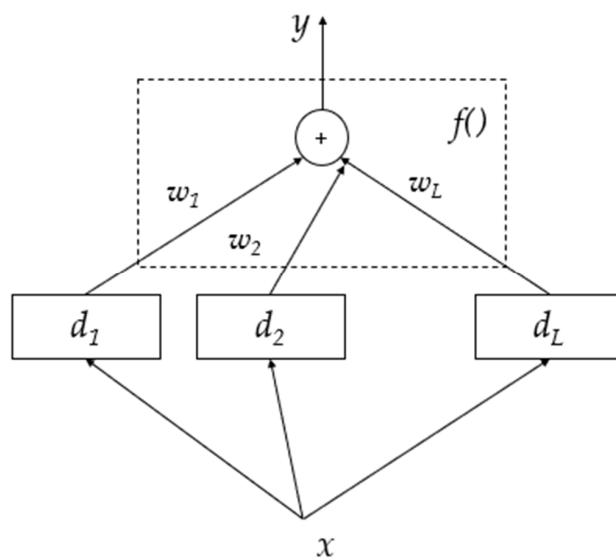
$$y_i = \sum_j w_j d_{ji} \text{ where } w_j \geq 0, \sum_j w_j, \quad (5)$$

where L is the number of independent voters.

In weighted sum,  $d_{ji}$  is the vote of model  $j$  for class  $C_i$ , and  $w_j$  is the weight of its vote. Simple voting is a special case that all voters have equal weight, namely,  $w_j = 1/L$ . For regression, the outputs of the baseline regressors can be combined by using simple or weighted averages or medians. The median is more tolerant of noise than the average. Another possible way might be to assess the accuracies of the learners on a separate



validation set and utilize that information to compute the weights so that more weights can be given to more accurate learners. These weights can also be learned from data [67].



**Figure 5.** Combining base learners and their outputs [67].

Each estimator follows the true regression function. If the discrepancy function is treated as a random noise function added to the true regression function and these noise functions are not related to the mean zero, averaging the individual estimates reduces the mean noise to be mean-like. In this sense, voting is smoothing in the function space and can be seen as a modifier of the real regression function, assuming smoothing [71].

With regard to the stacking regressor, in data mining the outputs of several different models are combined to make decisions more reliable. Some ML techniques do this by learning a collection of models and using them together. Stacking is one of these schemes. It can often improve predictive performance over a single model. It is a general technique applicable to classification and numerical estimation problems [64].

Stacking is a way of combining multiple models. Stacking tries to find which classifiers are reliable by using another learning algorithm like metalearner to find the best way to combine baseline learner results [68].

Stacking employs a metalearning method to learn how to best consolidate the multiple classifiers from different ML techniques [72]. Stacking has the advantage of combining the capabilities of several well-implemented methods for a regression or classification and make more successful predictions than a single method in the ensemble [73].

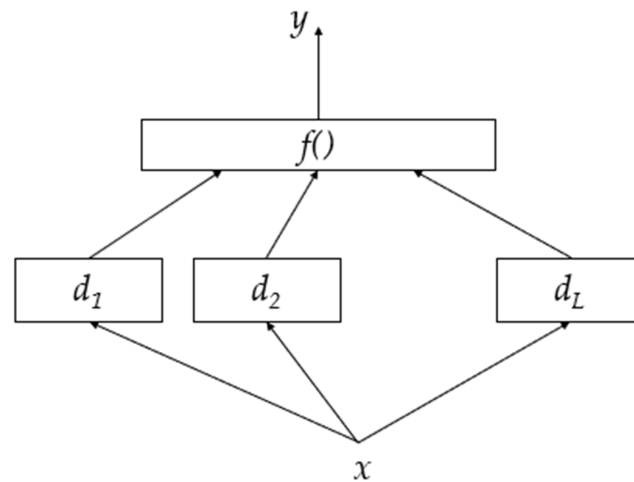
Vote combines predictions while stacking learns how to combine predictions [68]. Voting's problem is that it is not clear which classifier is reliable. Stacking combines classifiers by using stacking for both classification and regression problems. The base classifier, metalearner, and cross-validation fold number are determined by the user [68]. Unlike stacking, metalevel learning does not occur when classifiers are combined with voting schemes (such as plurality, probabilistic, or weighted voting). The voting scheme remains the same for all different training sets and learning algorithms or base classifiers [72].

Stacked generalization is a technique by which the way in which the outputs of the basic learners are combined is not necessarily linear but is learned through an associative system  $f(\cdot|\Phi)$ , another learner whose  $\Phi$  parameters are also trained (Figure 6) [67]. We have

$$y = f(d_1, d_2, \dots, d_L | \Phi). \quad (6)$$

The combiner learns what the output is from a given combination of outputs. The training data cannot be used to train the combiner function, as the primary learner may memorize the training set. A relational system must really learn how base learners make mistakes.

Stacking is a method of predicting and correcting for key base learners. Therefore, the relational basis should be trained on unused data to train students [67].



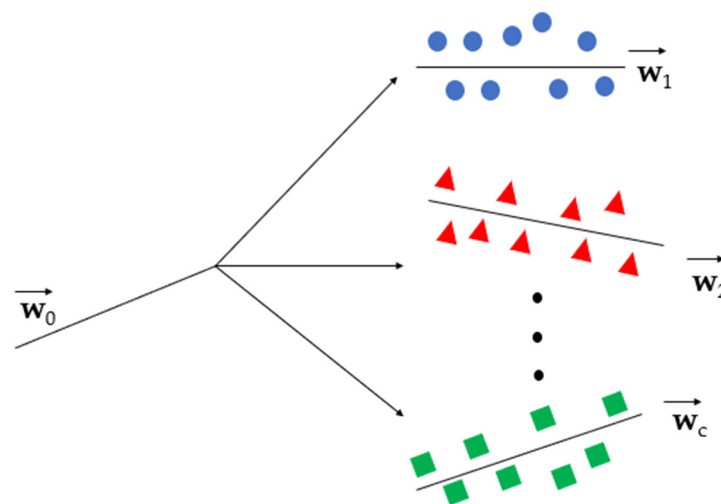
**Figure 6.** In stacking, the combiner is another learner. It is not limited to linear combinations such as voting [67].

Stacking has no constraints on the combination function, and unlike voting,  $f()$  can be nonlinear [67]. The output of the basic learning  $d_j$  defines a new L-dimensional space where the output regression function is learned by the combined function [67].

In stacking, basic learners can complement each other through the different learning algorithms [66]. When comparing voting and stacking, the trained rule in stacking is more flexible but requires extra parameters and creates variance [67].

#### 2.4. Multioutput Regressor (MOR)

A multioutput regression is also known as multitarget regression or multiresponse regression [74]. Multioutput regressor uses multiple single-target regression models for each output [75]. Multitarget model trees achieve comparable (sometimes better) accuracy, despite being much smaller than single-target model trees [76]. MOR refers to the simultaneous prediction of multiple output variables [77]. Multioutput regression accordingly delivers models that are smaller and learn faster [78] with equally good predictive power. Figure 7 represents an illustration of the underlying multiinput multioutput as a scheme [79].



**Figure 7.** Illustration of the underlying multiinput multioutput scheme [79].

For the data linear regression, decision tree regression, elastic net regression, K neighbors regression, support vector regression, random forest regression, gradient boosting regression, histogram-based gradient boosting regression, and voting and stacking are selected. Since a model is required to predict all three values for each section, the multiple output regressor of the scikit-learn library is used.

### 2.5. Performance Criterion

Model evaluation criteria are extremely important for comparing studies by using different models or algorithms. It has been determined that various evaluation criteria are used in studies (Bekdaş et al. [80]; Cakiroglu et al. [81]) in which the criteria of different works of literature using ML methods are compared. As a result of the examination of the aforementioned studies, square correlation coefficient ( $R^2$ ), root mean squared error (RMSE), mean absolute error (MAE), and mean squared error (MSE) performance criteria were used to evaluate the performance of the models in our study.

The performance of an ML model is evaluated through metrics. Frequently used regression metrics include RMSE and MAE. An  $R^2$  score is also calculated to ascertain how well the regression models approximate real data points.

The main evaluation criteria in this paper are  $R^2$  as Equation (5), RMSE as Equation (6), MAE as Equation (7), and MSE as Equation (8). Among them,  $R^2$  is the variance in the dependent variable accounted for by the specified model [82]. MSE is a measure of change in evaluation data that represents the expectation between the predicted value and the target value [83]. By taking the square root of the MSE value, the root mean squared error (RMSE) is calculated. MAE gives equal weight to all errors, while RMSE penalizes variance. This is because errors with a large absolute value are weighted more than errors with a small absolute value [84]. RMSE is the average squared root error between actual observation and model output [70]. The lower the measures, the more accurate the prediction results [85]. We have

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - x_i)^2}{\sum_{i=1}^n (y_i - \bar{x}_i)^2} \quad (7)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (8)$$

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (9)$$

$$MSE = \sum_{i=1}^n \frac{(y_i - x_i)^2}{n}, \quad (10)$$

where  $x_i$  represents the predicted value for the  $i$ th observation,  $y_i$  represents the actual value for the  $i$ th observation,  $\bar{x}_i$  represents the average of predicted values, and  $n$  represents the number of observations [70].

Since it is important for the validity of the model to test a model created in ML with new data that is not in the dataset, it is necessary to split the data set into training and test sets. The purpose of using training data is to determine the best values of the control parameters of the various ML algorithms used. The generalization power of the model developed after the training phase, which expresses the real-life success, is measured with the help of test data. For the ML methods used in the analysis, the training phase with the data corresponding to 70% of the total 4429 data; the test phase was carried out with the data corresponding to the remaining 30%. Test data must show high performance and at the same time, the performance values of the ML model in the training and test data are expected to be close to each other. Irregularities that may occur in the distribution of the dataset during the splitting of the dataset as training and test sets may adversely affect the performance of the ML model. This problem can be solved with the k-fold cross-validation method developed by Stone [86] in 1974. If more data is needed, the data is split into

training data and test data. The higher the classification accuracy of the training data, the more efficiently the test data create tests. Figure 8 shows the flowchart for the creation and evaluation of ML models.

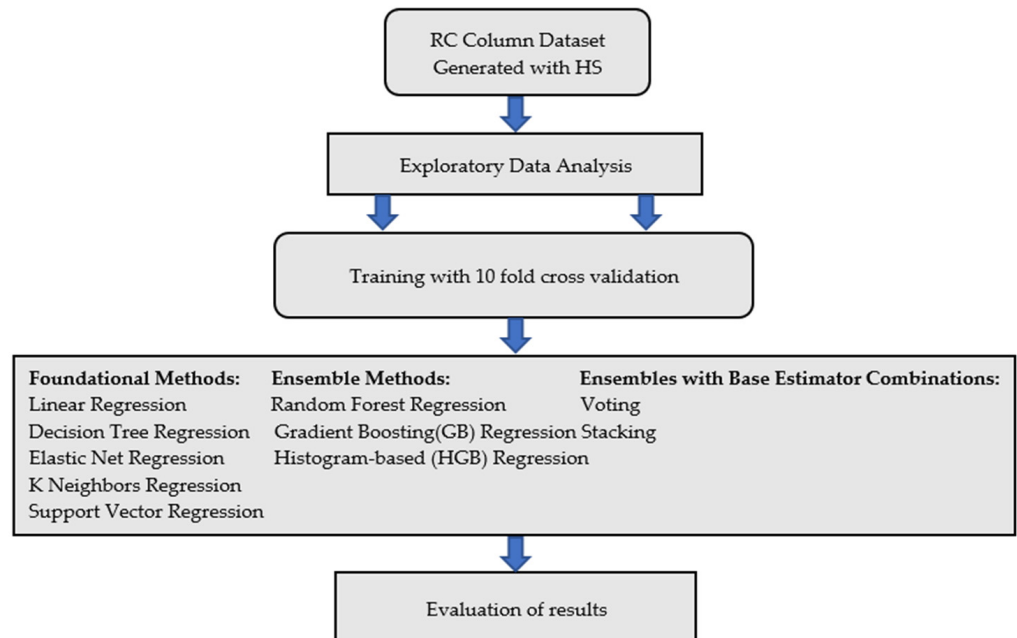


Figure 8. Flowchart for the creation and evaluation of machine learning models.

### 3. Results

In this study, predictions were made by using different ML algorithms for achieving the ecofriendly optimum design of a rectangular RC column. Bending moment and axial force features were used as inputs while reinforced rectangular column dimensions and the total area of the longitudinal reinforcement features were taken as outputs to be predicted with the MOR algorithm (Figure 9). The data used in the prediction was generated with HS algorithm as explained before. The HS algorithm was implemented to determine the optimal reinforced rectangular column dimensions and reinforcement area.

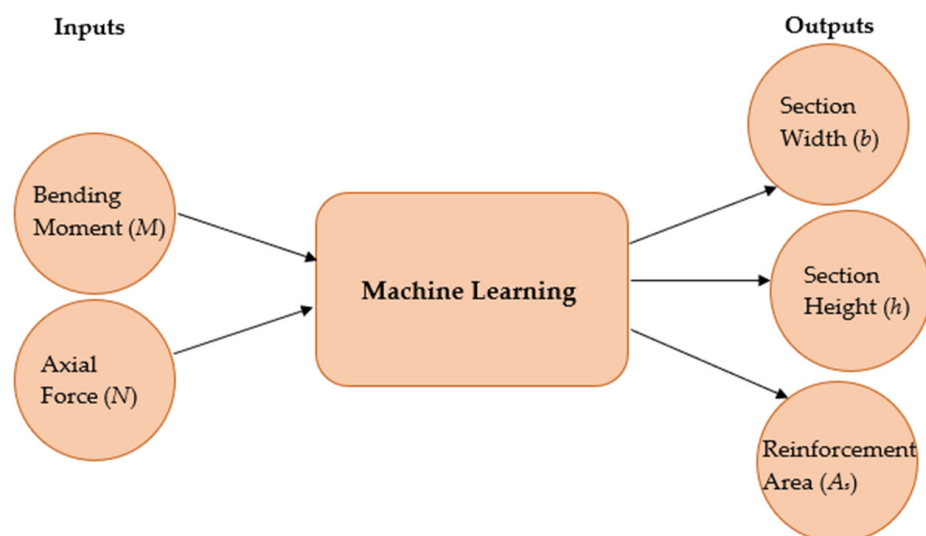


Figure 9. Inputs and outputs of the ML process.

The input dataset consisted of 4429 rows. Regression algorithms were used to predict optimal RC rectangular column dimensions and the total area of the longitudinal reinforcement based on the model inputs.

Linear regression, decision tree regression, elastic net regression, K neighbors regression, support vector regression, random forest regression, gradient boosting regression, histogram-based gradient boosting regression, voting, and stacking models were tested during the study.

To ensure the validity of the models, the 10-fold cross-validation method was applied in the model training process. In the 10-fold cross-validation, the dataset is divided into 10 parts. One of these parts is used as the test and the remaining nine parts are used as the training dataset. This process is repeated 10 times. To ensure each of these parts, it is used at least once as the test dataset. The accuracy of the model is then determined by averaging the accuracy values for all folds of the test datasets. The regression metrics of the models using different learning methods are shown in Table 4. The metrics used to assess models were coefficient of determination (R), negative root mean squared error (NRMSE), negative mean absolute error (NMAE) and negative mean squared error (NMSE).

**Table 4.** Performance results of the model.

Model	R <sup>2</sup>	NRMSE	NMAE	NMSE
Foundational				
Linear Regression	0.550	245.971	202.094	170,863.191
Decision Tree Regressor	0.996	29.915	16.465	2505.221
Elastic Net	0.546	243.435	204.567	166,992.364
K Neighbors Regressor	0.617	154.277	104.274	54,986.445
SVR	0.016	406.631	286.463	371,551.833
Ensemble				
Random Forest Regressor	0.998	17.777	10.162	840.449
Hist. Gradient Boosting Regressor	0.997	20.212	12.456	1090.250
Gradient Boosting Regressor	0.995	33.965	22.829	3406.932
Voting				
Combinations				
Triple combinations				
hgbr,rfr,gbr	0.998	19.921	12.868	1211.681
hgbr,gbr,rfr	0.998	20.408	13.191	1123.193
rfr,hgbr,gbr	0.998	20.565	13.153	1133.298
rfr,gbr,hgbr	0.998	19.952	12.862	1156.364
gbr,hgbr,rfr	0.998	20.170	13.120	1160.247
gbr,rfr,hgbr	0.998	19.849	13.196	1155.065
Quad combinations				
lm,dtr,knr,svr	0.738	177.226	137.481	74,641.302
lm,dtr,ent,knr	0.790	148.851	120.047	59,889.432
lm,svr,rfr,hgbr	0.852	150.574	117.956	56,440.011
lm,rfr,hgbr,gbr	0.969	66.7835	53.160	12,304.705
dtr,lm,hgbr,ent	0.886	123.958	102.064	43,174.734
dtr,rfr,hgbr,gbr	0.998	19.608	12.668	1155.584
ent,rfr,hgbr,gbr	0.968	62.278	53.828	12,587.107
knr,rfr,hgbr,gbr	0.975	45.830	31.108	5089.051
svr,lm,hgbr,rfr	0.852	151.568	118.307	56,129.966
svr,dtr,ent,gbr	0.851	153.118	119.770	57,176.112
lm,rfr,knr,ent	0.790	149.798	120.438	60,492.775
lm,ent,knr,svr	0.786	232.016	184.786	135,143.241
lm,knr,svr,rfr	0.788	178.366	137.091	75,001.233
knr,svr,lm,dtr	0.785	179.431	137.158	75,382.651
knr,svr,ent,hgbr	0.787	177.944	137.468	74,679.626
gbr,hgbr,rfr,ent	0.969	66.271	53.565	12,658.159

Table 4. Cont.

Model	R <sup>2</sup>	NRMSE	NMAE	NMSE
Quintuple combinations				
lm, dtr,ent, knr,svr	0.718	184.846	147.726	86,664.231
lm,ent, knr,dtr,svr	0.719	185.841	148.288	87,732.871
knr,lm,dtr,svr,ent	0.7196	186.463	148.442	86,791.598
Knr, dtr,svr,lm,ent	0.724	185.953	149.100	86,301.562
dtr, knr,lm,svr,ent	0.7208	187.130	146.722	85,077.065
dtr,svr, lm, ent, knr	0.720	184.505	147.026	86,665.337
ent,dtr,lm, knr,svr	0.722	186.811	148.981	86,171.159
ent,svr,dtr,lm, knr	0.719	185.300	146.752	86,947.917
svr,ent,dtr,lm, knr	0.719	184.548	147.960	87,283.171
svr,lm,ent,dtr, knr	0.718	185.733	148.987	86,629.887
Octal combinations				
gbr,hgbr,rfr,ent,lm,dtr, knr,svr	0.888	119.027	93.466	35,191.334
hgbr,rfr,lm,dtr, knr,gbr,ent,svr	0.889	118.193	93.847	35,303.200
svr, knr,ent,dtr,rfr,gbr,lm,hgbr	0.888	118.854	94.387	35,349.299
Stacking				
Final Estimator = Gradient Boosting Regressor				
gbr,hgbr,rfr,ent	0.998	18.155	11.1406	876.063
lm,dtr,ent, knr,svr	0.996	30.292	18.912	2547.112
hgbr,gbr,rfr	0.998	17.952	11.102	881.042
gbr,hgbr,rfr,ent,lm,dtr, knr,svr	0.998	18.445	11.616	895.697
Final Estimator = Hist. Gradient Boosting Regressor				
gbr,hgbr,rfr,ent	0.997	18.080	11.1449	866.437
lm,dtr,ent, knr,svr	0.996	30.388	17.929	2840.752
hgbr,gbr,rfr	0.997	19.249	11.326	1028.443
gbr,hgbr,rfr,ent,lm,dtr, knr,svr	0.997	18.023	11.443	1021.252
Final Estimator = Random Forest Regressor				
gbr,hgbr,rfr,ent	0.998	17.589	11.173	864.799
lm,dtr,ent, knr,svr	0.996	30.151	18.697	2897.098
hgbr,gbr,rfr	0.998	18.106	11.824	939.706
gbr,hgbr,rfr,ent,lm,dtr, knr,svr	0.998	17.582	10.921	912.215

When the accuracy metrics of the regression algorithms were examined, it was shown that random forest regressor provided the best results, followed by decision tree. The R<sup>2</sup> value of random forest was 0.9984, the NRMSE value was 17.7774, the NMAE value was 10.1621, and the NMSE value was 840.4499. The closest result to these results was obtained with the decision tree. The R<sup>2</sup> value was 0.9984, the NRMSE value was 29.9151, the NMAE value was 16.4650, and the NMSE value was 2505.2210. The best accuracy was achieved with the random forest algorithm.

The decision tree model follows the random forest model in terms of overall accuracy rate. Decision tree has the highest correlation coefficient determination coefficients (R<sup>2</sup>) and lowest NRMSE, NMAE, and NMSE values among foundational models.

Among the different regression techniques, the worst result is obtained with the SVR model. In the results obtained with the support vector regressor, the result is 0.0168 for R<sup>2</sup>. When the NRMSE value is examined, the highest error value also belonged to support vector regressor as 416.6314.

In the next stage, we tested different base estimator combinations in voting and stacking algorithms. The highest R<sup>2</sup> in voting models was achieved with the triple combination of gradient boosting, histogram-based gradient boosting, and random forest. The lowest R<sup>2</sup> in voting was obtained from the quintet combination of linear model, decision tree, elastic net, K neighbors, and support vector.

In stacking, R<sup>2</sup> was usually very high (more than 99%). The highest R<sup>2</sup> in stacking was achieved at octal combination of gradient boosting, histogram-based gradient boosting, random forest, elastic net, linear model, decision tree, K neighbors, and support vector has



been obtained. The lowest  $R^2$  in stacking was obtained from the quintuple combination of linear model, decision tree, elastic net, K neighbors, and support vector.

Most of the modelers did well in predicting the outcomes. The average correlation coefficients ( $R^2$ ) between actual values and estimates range from 0.71 to 0.99. In cases where the RMSE value is close to 0, it is evident that the models predict very low error rates.

#### 4. Discussion

Since RC columns are widely used, this increases CO<sub>2</sub> emissions. The optimum design of these structural elements must be done with a focus on minimizing the carbon footprint. In this study, we focus on the design optimization of RC concrete columns with the focus of minimizing the carbon footprint. A dataset of 4429 rows was produced by utilizing an HS optimization algorithm that generates optimal  $b$ ,  $h$ , and  $A_s$  values of an RC rectangular column based on  $M$  and  $N$  values to conform to the optimal design objective. Next, we trained a set of ML algorithms by utilizing this dataset and derived ML models to predict optimal  $b$ ,  $h$ , and  $A_s$  values based on input features  $M$  and  $N$ . The last stage of the study was focused on a comparison of the prediction capability of ML models. Mostly, the coefficient of determination ( $R^2$ ) is used for the evaluation/comparison of ML models. In this sense, decision tree was found to be the most successful model ( $R^2$ : 0.996) among the foundational methods. Random forest was the most successful model among the ensemble methods ( $R^2$ : 0.9984). In addition, most boosting classifiers in the methods group performed well ( $R^2$ : 0.995–0.997). The highest accuracy ( $R^2 > 0.99$ ) in different combination types tested for the voting technique was obtained in cases where ensemble methods (random forest, gradient boosting, and histogram-based gradient boosting) combination. This high accuracy is achieved as the result of the risk-reduction strategy applied by the voting approach. The highest accuracy ( $R^2$ : 0.96–0.99) in the stacking technique was obtained in cases where ensemble methods were combined.

Among similar studies, Lavercombe et al. [87] aimed to predict the compressive strength and embodied carbon of cement replacement concrete by using machine learning algorithms (deep neural network (DNN), support vector regression (SVR), gradient boosting regression (GBR), random forest (RF), k-nearest neighbors (KNN), and decision tree regression (DTR)). GBR models achieved the best prediction of the compressive strength and embodied carbon. The  $R^2$  of the GBR models for predicting the compressive strength and embodied carbon were 0.946 and 0.999, respectively. When compared with [87], our results are very similar in terms of performance metrics. Obtaining lower  $R^2$  values in analyses with foundational methods confirmed that ensemble methods provide better results for this prediction process.

In another similar effort, to estimate the environmental impacts of emissions from multiple construction activities, Fang et al. [88] applied a random forest-based estimation method. The  $R^2$  of the RF model used to estimate the construction phase carbon emissions during the early design phase was 0.605. Our results demonstrate a significant improvement in the accuracy of prediction ( $R^2 = 0.99$ ), as a study with a similar focus.

Since the main carbon emission from the concrete industry is due to the production of Portland cement, which is its main binder, finally in another similar study, Mansouri et al. [89] predicted the compressive strength of environmentally friendly concrete by using hybrid machine learning based on 147 datasets. The coefficient of determination ( $R^2$ ) of their study's gradient boosting regressor model was 0.9528 while our study's coefficient of determination of the gradient boosting regressor model was 0.995, which indicates a significant improvement for a study with a similar focus.

#### 5. Conclusions

One of the most important factors regarding environmental pollution is the greenhouse effect caused by the release of toxic gases such as CO<sub>2</sub> into the atmosphere. These types of gases eventually deplete the ozone layer, increase the air temperature, and thus pose a threat to living things. CO<sub>2</sub> causes increasing damage to the world and all living things

day by day. The control of CO<sub>2</sub> in concrete production is important for environmental protection. Environmentally friendly design has become even more important, especially in recent years. Based on the efficient use of ML, this research aimed to contribute to the minimization of the CO<sub>2</sub> emissions related to concrete production. In the study, columns under uniaxial bending and axial load were examined. Cross-section and reinforcement optimization were made according to the minimum CO<sub>2</sub> emission objective function depending on the bending moment direction and axial load. The bending axis has been taken as similar in all cases for comparison purposes.

First, a data-generation process is performed through HS, with a focus on finding optimal values of design variables (such as dimensions and reinforcement area in the cross-section) based on bending moment ( $M$ ) and axial force ( $N$ ). Following this, several machine learning (foundational and ensemble) models were trained with the dataset generated by HS, with a focus on the development of a model to predict the optimal design variables based on  $M$  and  $N$  values. The ML techniques are data-driven and the relatively large size of the dataset generated by using the HS optimization algorithm (i.e., 4429 rows) contributed to the accuracy of the prediction models. Thus, a more efficient machine learning process was carried out with this large amount of data.

Foundational methods, ensemble methods and additional voting and stacking models with various combinations (of base learners such as linear regression, decision tree regression, elastic net regression, K neighbors regression, support vector regression, random forest regression, gradient boosting regression, and histogram-based regression) were tested during the study. Random forest regression, gradient boosting regression, histogram-based regression, and voting and stacking provided a significantly good performance on this dataset with an  $R^2$  score greater than 0.99. The models with the worst performance were the ones trained by the SVR method.

The coefficient of determination ( $R^2$ ) in the random forest model was found as 0.9984. This finding indicates a perfect relationship between the ground truth and the estimated values. High prediction accuracies of ML models confirmed that ML can be utilized as an efficient method and tool for structural engineering for ecofriendly design.

The experiments demonstrated that optimization and ML models can be combined through a manual pipeline, and this pipeline can be used as an efficient medium by which to develop predictive models in support of ecofriendly RC column design. Our future studies will focus on automation of this optimization and ML modeling pipeline.

**Author Contributions:** Y.A. and Ü.I. generated the analysis codes. Y.A., Ü.I. and G.B. developed the theory, background, and formulations of the problem; Verification of the results was performed by Y.A. and Ü.I. The text of the paper was written by Y.A., Ü.I., G.B. and S.M.N. The text of the paper was edited by G.B., S.K. and Z.W.G. The figures were drawn by Y.A., G.B. and Z.W.G. supervised the research direction. 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:** Data is available on request to authors.

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

## Appendix A

In Table A1,  $i$  is the iteration number and  $\max(i)$  is the maximum number of iterations.

**Table A1.** HS Algorithm Parameters.

Harmony Memory Considering Rate (HMCR)	$HMCR = 0.5(1 - i/(\max(i)))$
Pitch Adjustment Rate (PAR)	$PAR = 0.05(1 - i/(\max(i)))$

## References

1. Bera, S. A Linear Optimization Model for Reducing CO<sub>2</sub> Emission from Power Plants. In Proceedings of the International Conference on Industrial Engineering and Operations Management, Bangalore, India, 16–18 August 2021.
2. Past Eight Years Confirmed to be the Eight Warmest on Record. Available online: <https://public.wmo.int/en/media/press-release/past-eight-years-confirmed-be-eight-warmest-record> (accessed on 31 January 2023).
3. Provisional State of the Global Climate in 2022. Available online: <https://public.wmo.int/en/our-mandate/climate/wmo-statement-state-of-global-climate> (accessed on 31 January 2023).
4. NASA Says 2022 Fifth Warmest Year on Record, Warming Trend Continues. Available online: <https://www.nasa.gov/press-release/nasa-says-2022-fifth-warmest-year-on-record-warming-trend-continues> (accessed on 31 January 2023).
5. Şimşek, O. *Beton ve Beton Teknolojisi (Deneyler İlaveli)*, 6th ed.; Seçkin Yayıncılık: Ankara, Türkiye, 2020; pp. 21–22.
6. Bekdaş, G.; Nigdeli, S.M.; Kim, S.; Geem, Z.W. Modified Harmony Search Algorithm-Based Optimization for Eco-Friendly Reinforced Concrete Frames. *Sustainability* **2022**, *14*, 3361. [\[CrossRef\]](#)
7. Doğançün, A. *Betonarme Yapıların Hesap ve Tasarımı*, 17th ed.; Birsen Yayınevi: İstanbul, Türkiye, 2018.
8. Turkey Disaster and Emergency Management Presidency. *Turkey Building Earthquake Regulation*; Turkey Disaster and Emergency Management Presidency: Ankara, Türkiye, 2018.
9. *TS500*; Requirements for Design and Construction of Reinforced Concrete Structures. Turkish Standards Institute: Ankara, Türkiye, 2003.
10. Adesina, A. Recent advances in the concrete industry to reduce its carbon dioxide emissions. *Environ. Chall.* **2020**, *1*, 100004. [\[CrossRef\]](#)
11. Asadollahfardi, G.; Delnavaz, M.; Rashnoiee, V.; Ghonabadi, N. Use of treated domestic wastewater before chlorination to produce and cure concrete. *Constr. Build. Mater.* **2016**, *105*, 253–261. [\[CrossRef\]](#)
12. Lehne, J.; Preston, F. *Making Concrete Change: Innovation in Low-Carbon Cement and Concrete*; Chatham House: London, UK, 2018.
13. Climate Change: The Massive CO<sub>2</sub> Emitter You May Not Know about. Available online: <https://www.bbc.com/news/science-environment-46455844> (accessed on 31 January 2023).
14. CEMBUREAU. *US Geological Survey*; Global Cement Report; Global Cement Directory: Epsom, UK, 2022.
15. Maalej, M.; Tanwongsva, S.; Paramasivam, P. Modelling of rectangular RC columns strengthened with FRP. *Cem. Concr. Compos.* **2003**, *25*, 263–276. [\[CrossRef\]](#)
16. Rajakarunakaran, S.A.; Lourdu, A.R.; Muthusamy, S.; Panchal, H.; Alrubaie, A.J.; Jaber, M.M.; Ali, M.H.; Tlili, I.; Maselena, A.; Majdi, A.; et al. Prediction of strength and analysis in self-compacting concrete using machine learning based regression techniques. *Adv. Eng. Softw.* **2022**, *173*, 103267. [\[CrossRef\]](#)
17. Deifalla, A.; Salem, N.M. A Machine learning model for torsion strength of externally bonded FRP-reinforced concrete beams. *Polymers* **2022**, *14*, 1824. [\[CrossRef\]](#)
18. Dissanayake, M.; Nguyen, H.; Poologanathan, K.; Perampalam, G.; Upasiri, I.; Rajanayagam, H.; Suntharalingam, T. Prediction of shear capacity of steel channel sections using machine learning algorithms. *Thin-Walled Struct.* **2022**, *175*, 109152. [\[CrossRef\]](#)
19. Amjad, M.; Ahmad, I.; Ahmad, M.; Wróblewski, P.; Kamiński, P.; Amjad, U. Prediction of pile bearing capacity using XGBoost algorithm: Modeling and performance evaluation. *Appl. Sci.* **2022**, *12*, 2126. [\[CrossRef\]](#)
20. Aydın, Y.; Işıkdag, Ü.; Bekdaş, G.; Nigdeli, S.M.; Geem, Z.W. Use of Machine Learning Techniques in Soil Classification. *Sustainability* **2023**, *15*, 2374. [\[CrossRef\]](#)
21. Kayabekir, A.E.; Arama, Z.A.; Bekdaş, G.; Nigdeli, S.M.; Geem, Z.W. Eco-Friendly design of reinforced concrete retaining walls: Multi-objective optimization with harmony search applications. *Sustainability* **2020**, *12*, 6087. [\[CrossRef\]](#)
22. Zhu, C.; Chang, Y.; Su, S.; Li, X.; Zhang, Z. Development of qL-EIV interactive curves for comparison of the environmental performance of composite slabs and RC slabs from the perspective of mechanical features. *Sci. Total Environ.* **2019**, *683*, 508–523. [\[CrossRef\]](#)
23. Wang, J.; Tingley, D.D.; Mayfield, M.; Wang, Y. Life cycle impact comparison of different concrete floor slabs considering uncertainty and sensitivity analysis. *J. Clean. Prod.* **2018**, *189*, 374–385. [\[CrossRef\]](#)
24. Paik, I.; Na, S. Comparison of carbon dioxide emissions of the ordinary reinforced concrete slab and the voided slab system during the construction phase: A case study of a residential building in South Korea. *Sustainability* **2019**, *11*, 3571. [\[CrossRef\]](#)
25. Purnell, P. The carbon footprint of reinforced concrete. *Adv. Cem. Res.* **2013**, *25*, 362–368. [\[CrossRef\]](#)
26. Destrée, X.; Pease, B. *Reducing CO<sub>2</sub> Emissions of Concrete Slab Constructions with the Prime Composite Slab System*; Special Publication; American Concrete Institute (ACI): Indianapolis, IN, USA, 2015; Volume 299, pp. 1–12.
27. Yepes, V.; Gonzalez-Vidosa, F.; Alcalá, J.; Villalba, P. CO<sub>2</sub>-optimization design of reinforced concrete retaining walls based on a VNS-threshold acceptance strategy. *J. Comput. Civ. Eng.* **2012**, *26*, 378–386. [\[CrossRef\]](#)
28. Arama, Z.A.; Kayabekir, A.E.; Bekdaş, G.; Geem, Z.W. CO<sub>2</sub> and cost optimization of reinforced concrete cantilever soldier piles: A parametric study with harmony search algorithm. *Sustainability* **2020**, *12*, 5906. [\[CrossRef\]](#)
29. Geem, Z.W.; Kim, J.H.; Loganathan, G.V. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [\[CrossRef\]](#)
30. *The MathWorks, Matlab R2022a*; The MathWorks: Natick, MA, USA, 2022.
31. Bekdaş, G. Harmony search algorithm approach for optimum design of post-tensioned axially symmetric cylindrical reinforced concrete walls. *J. Optim. Theory Appl.* **2015**, *164*, 342–358. [\[CrossRef\]](#)

32. Kayhan, A.H.; Korkmaz, K.A.; Irfanoglu, A. Selecting and scaling real ground motion records using harmony search algorithm. *Soil Dyn. Earthq. Eng.* **2011**, *31*, 941–953. [CrossRef]
33. Nigdeli, S.M.; Bekdaş, G. Optimum tuned mass damper design in frequency domain for structures. *KSCE J. Civ. Eng.* **2017**, *21*, 912–922. [CrossRef]
34. Kayabekir, A.E.; Nigdeli, S.M.; Bekdaş, G.; Yücel, M. Optimum design of tuned mass dampers for real-size structures via adaptive harmony search algorithm. In Proceedings of the 14th ECCOMAS Thematic Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control (EUROGEN 2021), Athens, Greece, 28–30 June 2021; pp. 20–26.
35. Ocak, A.; Nigdeli, S.M.; Bekdaş, G.; Kim, S.; Geem, Z.W. Adaptive Harmony Search for Tuned Liquid Damper Optimization under Seismic Excitation. *Appl. Sci.* **2022**, *12*, 2645. [CrossRef]
36. Syrimi, P.G.; Sapountzakis, E.J.; Tsiatas, G.C.; Antoniadis, I.A. Parameter optimization of the KDamper concept in seismic isolation of bridges using harmony search algorithm. In Proceedings of the 6th COMPDYN, Rhodes Island, Greece, 15–17 June 2017.
37. Jin, C.; Chung, W.C.; Kwon, D.S.; Kim, M. Optimization of tuned mass damper for seismic control of submerged floating tunnel. *Eng. Struct.* **2021**, *241*, 112460. [CrossRef]
38. Akin, A.; Saka, M. Optimum design of concrete cantilever retaining walls using the harmony search algorithm. In Proceedings of the 10th International Conference on Computational Structures Technology, Valencia, Spain, 14–17 September 2010.
39. Bekdaş, G.; Arama, Z.A.; Kayabekir, A.E.; Geem, Z.W. Optimal design of cantilever soldier pile retaining walls embedded in frictional soils with harmony search algorithm. *Appl. Sci.* **2020**, *10*, 3232. [CrossRef]
40. Arama, Z.A.; Kayabekir, A.E.; Bekdaş, G.; Kim, S.; Geem, Z.W. The usage of the harmony search algorithm for the optimal design problem of reinforced concrete retaining walls. *Appl. Sci.* **2021**, *11*, 1343. [CrossRef]
41. Yücel, M.; Kayabekir, A.E.; Bekdaş, G.; Nigdeli, S.M.; Kim, S.; Geem, Z.W. Adaptive-Hybrid harmony search algorithm for multi-constrained optimum eco-design of reinforced concrete retaining walls. *Sustainability* **2021**, *13*, 1639. [CrossRef]
42. Degertekin, S.O. Improved harmony search algorithms for sizing optimization of truss structures. *Comput. Struct.* **2012**, *92*, 229–241. [CrossRef]
43. Toklu, Y.C.; Bekdaş, G.; Temur, R. Analysis of trusses by total potential optimization method coupled with harmony search. *Struct. Eng. Mech.* **2013**, *45*, 183–199. [CrossRef]
44. Akin, A.; Saka, M. Optimum detailed design of reinforced concrete continuous beams using the harmony search algorithm. In Proceedings of the 10th International Conference on Computational Structures Technology, Valencia, Spain, 14–17 September 2010.
45. Kayabekir, A.E.; Bekdaş, G.; Nigdeli, S.M.; Apak, S. Cost and Environmental Friendly Multi-Objective Optimum Design of Reinforced Concrete Columns. *J. Environ. Prot. Ecol.* **2022**, *23*, 890–899.
46. Understanding Float in Python [with Examples]. Available online: <https://www.simplilearn.com/tutorials/python-tutorial/float-in-python> (accessed on 1 February 2023).
47. Seaborn Library. Available online: <https://seaborn.pydata.org/> (accessed on 31 January 2023).
48. Python (3.9) [Computer Software]. Available online: <http://python.org> (accessed on 31 January 2023).
49. Meng, T.; Jing, X.; Yan, Z.; Pedrycz, W. A survey on machine learning for data fusion. *Inf. Fusion* **2020**, *57*, 115–129. [CrossRef]
50. Camacho, D.M.; Collins, K.M.; Powers, R.K.; Costello, J.C.; Collins, J.J. Next-Generation machine learning for biological networks. *Cell* **2018**, *173*, 1581–1592. [CrossRef]
51. Kwon, D.; Kim, H.; Kim, J.; Suh, S.C.; Kim, I.; Kim, K.J. A survey of deep learning-based network anomaly detection. *Clust. Comput.* **2019**, *22*, 949–961. [CrossRef]
52. Cunningham, P.; Cord, M.; Delany, S.J. Supervised Learning. In *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 21–49.
53. Olivier, C.; Schölkopf B ve Alexander, Z. *Semi-Supervised Learning (Adaptive Computation and Machine Learning)*; MIT Press: Cambridge, MA, USA, 2006.
54. Xu, X.; He, H.G.; Hu, D. Efficient reinforcement learning using recursive least-squares methods. *J. Artif. Intell. Res.* **2002**, *16*, 259–292. [CrossRef]
55. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; MIT Press: Cambridge, MA, USA, 2018.
56. Anaconda3 [Computer Software]. Available online: <https://anaconda.org/> (accessed on 31 January 2023).
57. 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.
58. Wu, C.-S.M.; Patil, P.; Gunaseelan, S. Comparison of different machine learning algorithms for multiple regression on black friday sales data. In Proceedings of the 9th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 23–25 November 2018; pp. 16–20.
59. Xu, M.; Watanachaturaporn, P.; Varshney, P.K.; Arora, M.K. Decision tree regression for soft classification of remote sensing data. *Remote Sens. Environ.* **2005**, *97*, 322–336. [CrossRef]
60. Pekel, E. Estimation of soil moisture using decision tree regression. *Theor. Appl. Climatol.* **2020**, *139*, 1111–1119. [CrossRef]
61. Hands-On Tutorial on ElasticNet Regression. Available online: <https://analyticsindiamag.com/hands-on-tutorial-on-elasticnet-regression/> (accessed on 4 February 2023).
62. Modaresi, F.; Araghinejad, S.; Ebrahimi, K. A comparative assessment of artificial neural network, generalized regression neural network, least-square support vector regression, and K-nearest neighbor regression for monthly streamflow forecasting in linear and nonlinear conditions. *Water Resour. Manag.* **2018**, *32*, 243–258. [CrossRef]



63. Chen, K.Y.; Wang, C.H. Support vector regression with genetic algorithms in forecasting tourism demand. *Tour. Manag.* **2007**, *28*, 215–226. [[CrossRef](#)]
64. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
65. Implementing Gradient Boosting in Python. Available online: <https://blog.paperspace.com/implementing-gradient-boosting-regression-python/> (accessed on 4 February 2023).
66. Scikit-Learn Documentation. Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html> (accessed on 4 February 2023).
67. Alpaydin, E. *Introduction to Machine Learning*, 3rd ed.; MIT Press: Cambridge, MA, USA, 2014; Volume 67, pp. 69–72.
68. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. Practical machine learning tools and techniques. In *Data Mining*; Elsevier: Amsterdam, The Netherlands, 2005; Volume 2.
69. Erdebilli, B.; Devrim-İçtenbaş, B. Ensemble Voting Regression Based on Machine Learning for Predicting Medical Waste: A Case from Turkey. *Mathematics* **2022**, *10*, 2466. [[CrossRef](#)]
70. Yulisa, A.; Park, S.H.; Choi, S.; Chairattanawat, C.; Hwang, S. Enhancement of voting regressor algorithm on predicting total ammonia nitrogen concentration in fish waste anaerobiosis. *Waste Biomass Valorization* **2022**, *14*, 461–478. [[CrossRef](#)]
71. Perrone, M.P. Improving Regression Estimation: Averaging Methods for Variance Reduction with Extensions to General Convex Measure. Ph.D. Thesis, Brown University, Providence, RI, USA, 1993.
72. Džeroski, S.; Ženko, B. Is combining classifiers with stacking better than selecting the best one? *Mach. Learn.* **2004**, *54*, 255–273. [[CrossRef](#)]
73. Kilimci, Z.H. Ensemble Regression-Based Gold Price (XAU/USD) Prediction. *J. Emerg. Comput. Technol.* **2022**, *2*, 7–12.
74. Bodendorf, F.; Xie, Q.; Merkl, P.; Franke, J. A multi-perspective approach to support collaborative cost management in supplier-buyer dyads. *Int. J. Prod. Econ.* **2022**, *245*, 108380. [[CrossRef](#)]
75. Shaaban, K.; Hamdi, A.; Ghanim, M.; Shaban, K.B. Machine learning-based multi-target regression to effectively predict turning movements at signalized intersections. *Int. J. Transp. Sci. Technol.* **2022**, *12*, 245–257. [[CrossRef](#)]
76. Appice, A.; Džeroski, S. Stepwise induction of multi-target model trees. In Proceedings of the Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, 17–21 September 2007; Springer: Berlin/Heidelberg, Germany, 2007; Volume 18, pp. 502–509.
77. Borchani, H.; Varando, G.; Bielza, C.; Larranaga, P. A survey on multi-output regression. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2015**, *5*, 216–233. [[CrossRef](#)]
78. Kocev, D.; Džeroski, S.; White, M.D.; Newell, G.R.; Griffioen, P. Using single-and multi-target regression trees and ensembles to model a compound index of vegetation condition. *Ecol. Model.* **2009**, *220*, 1159–1168. [[CrossRef](#)]
79. Chou, J.S.; Truong, D.N.; Che, Y. Optimized multi-output machine learning system for engineering informatics in assessing natural hazards. *Nat. Hazards* **2020**, *101*, 727–754. [[CrossRef](#)]
80. Bekdaş, G.; Cakiroglu, C.; Islam, K.; Kim, S.; Geem, Z.W. Optimum Design of Cylindrical Walls Using Ensemble Learning Methods. *Appl. Sci.* **2022**, *12*, 2165. [[CrossRef](#)]
81. Cakiroglu, C.; Islam, K.; Bekdaş, G.; Isikdag, U.; Mangalathu, S. Explainable machine learning models for predicting the axial compression capacity of concrete filled steel tubular columns. *Constr. Build. Mater.* **2022**, *356*, 129227. [[CrossRef](#)]
82. Lüdecke, D.; Ben-Shachar, M.S.; Patil, I.; Waggoner, P.; Makowski, D. Performance: An R package for assessment, comparison and testing of statistical models. *J. Open Source Softw.* **2021**, *6*, 3139. [[CrossRef](#)]
83. Huang, J.C.; Ko, K.M.; Shu, M.H.; Hsu, B.M. Application and comparison of several machine learning algorithms and their integration models in regression problems. *Neural Comput. Appl.* **2020**, *32*, 5461–5469. [[CrossRef](#)]
84. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [[CrossRef](#)]
85. Solano, E.S.; Dehghanian, P.; Affonso, C.M. Solar Radiation Forecasting Using Machine Learning and Ensemble Feature Selection. *Energies* **2022**, *15*, 7049. [[CrossRef](#)]
86. Stone, M. Cross-Validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B Methodol.* **1974**, *36*, 111–147. [[CrossRef](#)]
87. Lavercombe, A.; Huang, X.; Kaewunruen, S. Machine learning application to eco-friendly concrete design for decarbonisation. *Sustainability* **2021**, *13*, 13663. [[CrossRef](#)]
88. Fang, Y.; Lu, X.; Li, H. A random forest-based model for the prediction of construction-stage carbon emissions at the early design stage. *J. Clean. Prod.* **2021**, *328*, 129657. [[CrossRef](#)]
89. Mansouri, E.; Manfredi, M.; Hu, J.W. Environmentally Friendly Concrete Compressive Strength Prediction Using Hybrid Machine Learning. *Sustainability* **2022**, *14*, 12990. [[CrossRef](#)]

**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.