XLI
CILAMCE
XLI Ibero-Latin American Congress on Computational Methods in Engineering
16-19 November, 2020 FOZ DO IGUAÇU
PARANÁ – BRAZIL

A Computational Method to Predict the Concrete Compression Strength Using Decision Trees and Random Forest

Priscila F. S. Silva, Gray F. Moita, Vanderci F. Arruda

[1] *Centro Federal de Educação Tecnológica de Minas Gerais*

Av. Amazonas, 5.253, Nova Suíça, CEP: 30.421-169, Belo Horizonte, MG, Brasil.

201422800040@aluno.cefetmg.br; gray@dppg.cefetmg.br; vanderci-engcivil@hotmail.com

Abstract. The engineering properties of concrete made structures depend on various parameters such as the properties of the mixture of concrete, including its nonhomogeneous nature. A clear understanding of such complex behavior is needed to use these materials successfully in various engineered structures. Recently, the advancement of machine learning techniques has managed to propose different optimum solutions to general engineering applications. This study aims to predict concrete compression strength by employing methods such as Decision Trees (DTR) and Random Forests (RFR) using the database available in Yeh [1]. The model used in this study considers the effect of eight contributory factors, i.e., cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate and age to predict the concrete compressive strength. Computational methods like DTR and RFR are used to develop a predictive model. A tuning method called GridsearchCV is also used to automate the process of adjusting the algorithms. The study also compared the performance of the algorithms concerning their predicting abilities. The divergence of the root means square error (RMSE) and R^2 between the output and target values of the test set was monitored and used to establish a better solution.

Keywords: Concrete Compression Strength; Machine Learning; Decision Trees; Random Forest.

1 Introduction

The compressive strength of concrete is still one of the most widely used parameters in structural engineering for the design of reinforced concrete structures. The performance of concrete, when defined empirically, can be affected by nonlinear factors when using the concrete compression test as a destructive procedure on concrete specimens. However, this activity involves time, planning and financial resources because the commonly used compressive strength factor is obtained on the 28th day. Moreover, concrete is heterogeneous and does not follow the premises idealized for homogeneous materials when it is subjected to stresses and strains, presenting different results when it comes to tensile and compressive strength [2].

Structural engineering has been a field of significant development through the implementation and testing of new computational models, predicting the different properties of concrete mixtures. In behavioral models, pattern recognition is constructive, and computational intelligence methods can be used. Bio-inspired models can also be an excellent aid for designing structures for civil engineering applications [3]–[6].

This paper focuses on using computational intelligence to analyze and develop a prediction model for concrete compressive strength using computational methods, emphasizing accuracy and efficiency, and the potential to deal with experimental data. This study aims to contribute to a new model to determine the compressive strength of concrete using models such as Decision Trees (DTR) and Random Forest (RFR).

2 Database

The chosen database was made available in the article written by Yeh [1]. The programming language used to implement the technique shown in this paper was Python and Scikit-learn library was also used in this work.

The database visualization and pre-processing are sought to obtain a better understanding of the dataset to be studied. The first one intends to visualize correlations between inputs and outputs to achieve this goal. In this project, the following methods were used for better visualization of the database:

Histograms: the purpose of using histograms is to estimate whether the database has a normal distribution or biased to the left or right. The figures obtained can improve the visualization and analyses of the resources more effectively [7].

Density plots: Density plots are variables that provide an idea of each feature distribution in the dataset. With these plots, one can see a smooth distribution curve drawn over the top of each histogram.

Box plots: Box plots are another effective way to summarize the distribution of each available resource in the dataset. These boxes are useful because they indicate the median value and the first and last quartile of the used data.

In addition to the above, the dataset needs to be pre-processed before its application to improve computational models [8]. The pre-processing method used is described as follows:

Feature scaling: This method involves transforming all characteristics on a standard scale [9], [10]. Usually, resources are transformed within a range between 0 and 1. The scale is necessary to construct a machine learning model.

To obtain better results, the database is usually split into training and testing data. Thus, the algorithm is trained with a volume of data that is validated in the test set. This is done to guarantee that the result obtained is not biased and only learns from similar data used for training. The dataset is reorganized with re-sampling. The primary re-sampling forms are presented in the following:

Cross-validation: There are several types of cross-validation. However, the most common is the k-fold method. In this method, several samples k are created, each sample being set aside while the model trains with the remainder. The process repeats until it is possible to determine the "quality" of each observation [11]–[13]. The most common values for the number of samples are between 5 and 10.

GridsearchCV: It is the tuning process that uses hyper-parametrization to determine the optimal values for a given model. GridsearchCV performs an exhaustive search on the specified parameters. This method is computationally expensive but produces excellent results [14], [15].

3 Analytical methods

To predict the compressive strength of concrete, a suitable machine learning method that best suits the dataset is selected. Decision trees (DTR) and Random Forests (RFR) were chosen due to their power of decision and their regular use in linear regression tasks [16], [17].

Decision trees are easy to use without many pre-processing strategies. These trees divide the decision boundaries into rectangles parallel to the axis. The idea is to build a collection of trees with a controlled variation [18]. On the other side, the Random Forest is a technique that can perform regression and classification tasks using multiple Decision Trees and bootstrap aggregation, commonly known as bagging. This technique involves combining multiple Decision Trees to determine the final output by relying on individual Decision Trees. The biggest problem with Decision Trees is that they tend to over-fit training data. Error pruning is the most common technique for avoiding this type of problem [19].

The DTR and RFR parameters were defined with the help of GridsearchCV, where the numbers of estimators and the maximum characteristics are the parameters to be adjusted. In most situations, as the number of estimators increases, the models used tends to be optimized. The maximum characteristics parameter analyzes the maximum characteristics to be considered in each division.

To evaluate the error obtained the root mean squared error, as in Eq. (1) below

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \qquad (1)$$

is commonly used, where $\hat{y}$ is the predicted value of y and y is the average value of y.

4 Analysis and results

This work required the acquisition of reliable experimental data to determine the compressive strength of concrete through computational intelligence. The database chosen was obtained from studies by Yeh [1]. This database presents 1030 experimental stress versus compression tests. Eight input parameters and one output parameter were used. The input parameters are cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate and the age of the specimen. The output parameter is the compressive strength of concrete (f_c). The database used present the maximums and minimums of input and output components, as shown in Table 1.

Table 1: Inputs and outputs: Maximum and Minimum

Model attributes	**Values**	
	(Maximum)	(Minimum)
Cement (kg/m³)	540	102
Blast furnace slag (kg/m³)	359.4	0
Fly ash (kg/m³)	200.1	0
Water (kg/m³)	247	121.8
Superplasticizer (kg/m³)	32.2	0
Coarse aggregate (kg/m³)	1145	801
Fine aggregate (kg/m³)	992.6	594
Age (days)	365	1
Concrete compressive strength (MPa)	82.6	2.33

It can be seen that the database provided by Yeh [1] is quite consolidated and has a proper distribution for input and output variables.

The visualization of histograms, density boxes and box plots obtained from the database used is provided. As stated earlier, this visualization aims to give a better idea of which method is more appropriate to obtain the result. Figure 1shows the histograms and density plots of the data used in the model. The variables have an almost normal distribution. Thus, it is possible to see that the efficiency of learning algorithms can be facilitated.

To build up the used predictive models, the collected experimental data was split into two parts: the training set and the testing set. The results presented in this paper used 85% of the data (875 samples) for training the computer models; the remaining 15% of the data (155 samples) were used for testing. It is worth emphasizing that cross-validation with 10-fold was used to select the best parameters for each method, so there was not necessary to use the data validation set. Table 2 present the range of parameters used in the GridsearchCV and the best parameters used in the experiments for the Decision Trees and Random Forest algorithms.

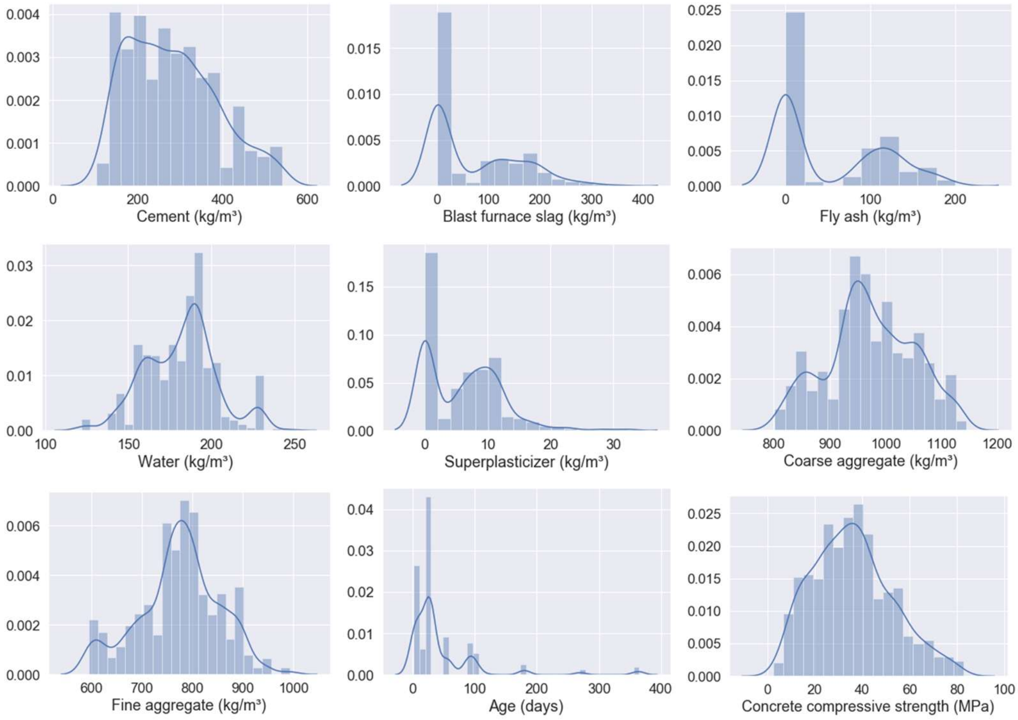

Figure 1: Histograms and density plots

Table 2: Range of parameters and best parameters used in the experiments

	Decision Trees		Randon Forest	
Parameter	**Range**	**Settings**	**Range**	**Setting**
Maximum depth	1-3	3	4-10	10
Maximum of leaf nodes	100-200	100	100-120	120
Minimum number of samples required to split	2-30	10	7-9	7
Number of trees in the forest	-	-	400-500	500

Figure 2 shows the original and predicted values for the Decision Tree method. The model used presents a good result with R² equal to 0.88 and the RMSE equal to 6.14 MPa.

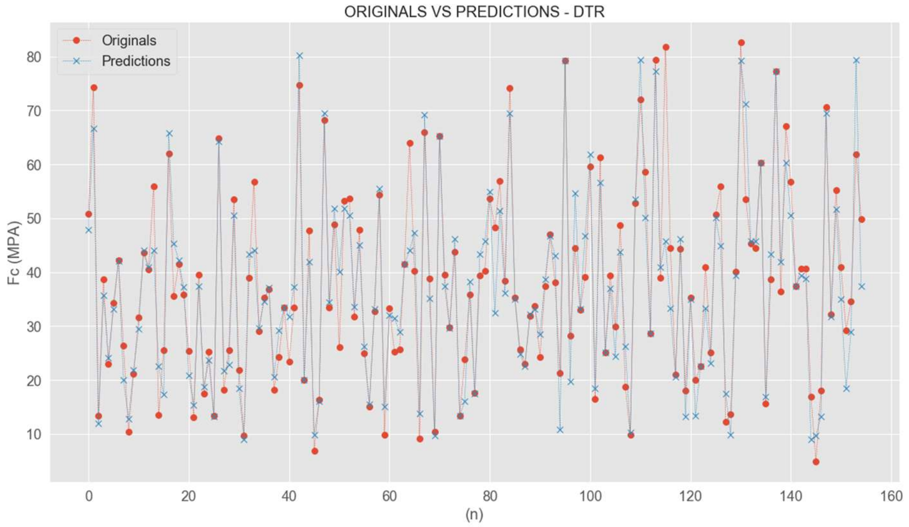

Figure 2: Original versus expected results for DTR

Figure 3 shows the original and predicted values for the Random Forest method. The model used presents an excellent result with R² equal to 0.90 and RMSE equal to 5.60 MPa.

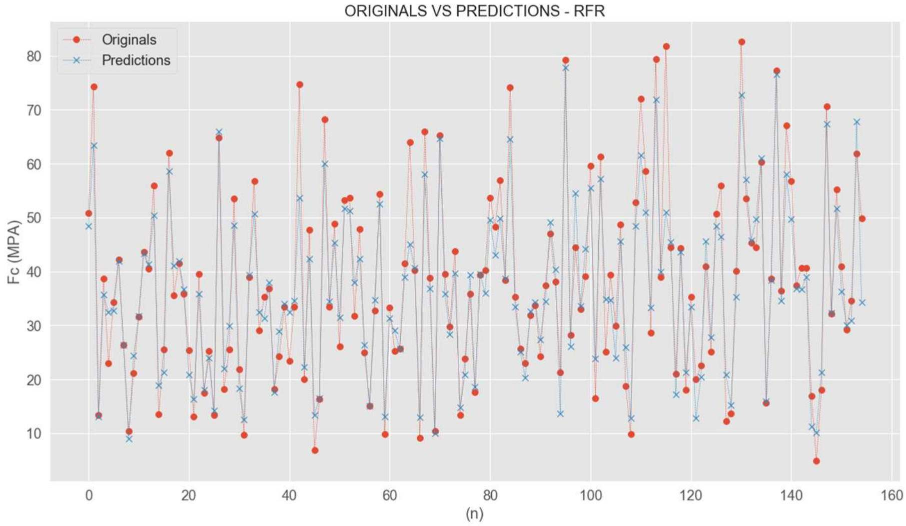

Figure 3: Original versus expected results for RFR

Figure 4 shows the scatter of predicted and original values of concrete compressive strength test data for both proposed models.

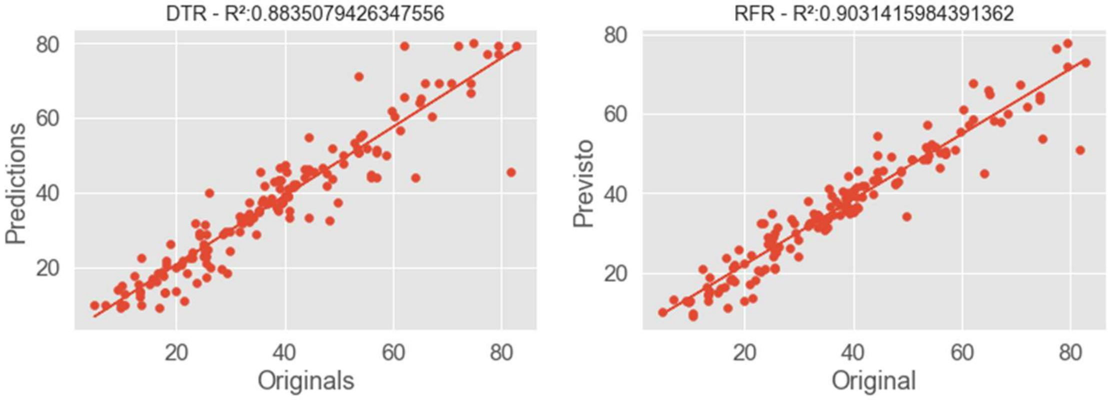

Figure 1: Scatter of predicted and experimental values of concrete compressive strength

The summaries of the performance values obtained for the training set are summarized in Table 3. The table presents the R², the RMSE and the runtime for the methods used.

Table 3: Comparison of obtained results

	R²	**RMSE (MPa)**	**Execution time (seconds)**
Decision Tree	0.89	6.14	4.504
Random forest	0.90	5.60	5.506

The results for the testing set parameters results are also compared to some previous studies using Random Forest, SVM and ANN algorithms with Yeh's dataset, as shown in Table 4.

5 Conclusions

This work aimed to present the study of computational intelligence applied to define the concrete compressive strength from a database obtained in the studies of Yeh [1]. Two computational methods of machine learning were used: Decision Tree and Random Forest. Data pre-processing and data visualization methods were also used to improve the results. The obtained results show that Random Forest was the best method with a better performance of the used methods.

The computational intelligence models used are reliable to solve different complex problems, such as prediction problems. These models can be used to solve a specific problem when a deviation in available data is expected and accepted and, also, when a defined methodology is not available. Therefore, to predict the properties of concrete with high reliability, instead of using expensive experimental investigation, conventional models can be replaced by computational intelligence models.

Computational intelligence models can predict the concrete compressive strength specimens, as shown in this study. The prediction of mean percent error values for these simulations shows a high degree of consistency with compressive strength and is experimentally evaluated from the concrete specimens used. Thus, the present study suggests an alternative approach to evaluate compressive strength against destructive testing methods.

Table 4: Comparison with the results for the same dataset in previous studies

Research	**Algorithm**	**R²**	**RMSE (MPa)**
Chou et al. [20]	ANN	0.88	-
	SVM	0.91	-
Chou et al. [21]	ANN	-	7.9
	SVM	-	5.5
Young et al.[22]	Random Forest	0.86	5.7
	ANN	0.82	6.3
	SVM	0.83	6.4
This paper	Decision trees	0.89	6.14
	Random Forest	0.90	5.6

6 Acknowledgments

CAPES and CEFET-MG supported the work described in this paper.

7 References

[1] I. C. Yeh, “Modeling of strength of high-performance concrete using artificial neural networks,” *Cem. Concr. Res.*, vol. 28, no. October, pp. 1797–1808, 1998.

[2] S. K. Babanajad, *Application of genetic programming for uniaxial and multiaxial modeling of concrete*. Switzerland: Springer, 2015.

[3] U. Reuter, A. Sultan, and D. S. Reischl, “A comparative study of machine learning approaches for modeling concrete failure surfaces,” *Adv. Eng. Softw.*, vol. 116, no. July 2017, pp. 67–79, 2018, doi: 10.1016/j.advengsoft.2017.11.006.

[4] A. A. Torky and A. A. Aburawwash, “A Deep Learning Approach to Automated Structural Engineering of Prestressed Members,” *Int. J. Struct. Civ. Eng.*, vol. 7, no. 4, pp. 347–352, 2018, doi: 10.18178/ijscer.7.4.347-352.

[5] R. Cook, J. Lapeyre, H. Ma, and A. Kumar, “Prediction of Compressive Strength of Concrete: Critical Comparison of Performance of a Hybrid Machine Learning Model with Standalone Models,” *J. Mater. Civ. Eng.*, vol. 31, no. 11, pp. 1–15, 2019, doi: 10.1061/(ASCE)MT.1943-5533.0002902.

[6] W. Ben Chaabene, M. Flah, and M. L. Nehdi, “Machine learning prediction of mechanical properties of concrete: Critical review,” *Constr. Build. Mater.*, vol. 260, p. 119889, 2020, doi: 10.1016/j.conbuildmat.2020.119889.

[7] D. W. Scott, "On optimal and data-based histograms," *Biometrika*, vol. 66, no. 3, pp. 605–610, 1979.

[8] L. Wang, G. Wang, and C. A. Alexander, "Big data and visualization: methods, challenges and technology progress," *Digit. Technol.*, vol. 1, no. 1, pp. 33–38, 2015.

[9] G. Forman, "BNS feature scaling: an improved representation over tf-idf for svm text classification," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 263–270.

[10] M.-L. Zhang and Z.-H. Zhou, "Improve multi-instance neural networks through feature selection," *Neural Process. Lett.*, vol. 19, no. 1, pp. 1–10, 2004.

[11] B. Efron and R. Tibshirani, "Improvements on cross-validation: the 632+ bootstrap method," *J. Am. Stat. Assoc.*, vol. 92, no. November, pp. 37–41, 1997.

[12] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[13] J. Shao, "Linear model selection by cross-validation," *J. Am. Stat. Assoc.*, vol. 88, no. 422, pp. 486–494, 1993, doi: 10.1080/01621459.1993.10476299.

[14] L. Buitinck *et al.*, "API design for machine learning software: experiences from the scikit-learn project," *arXiv Prepr. arXiv1309.0238*, pp. 1–15, 2013, [Online]. Available: http://arxiv.org/abs/1309.0238.

[15] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[16] I. Barandiaran and T. K. Ho, "The Random Subspace Method for Constructing Decision Forest," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 20, no. 8, pp. 832–844, 1998.

[17] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, 2005, doi: 10.1080/01431160412331269698.

[18] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, "Random forests for land cover classification," *Pattern Recognit. Lett.*, vol. 27, no. 4, pp. 294–300, 2006.

[19] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[20] J. Chou, C. Chiu, M. Farfoura, and I. Al-taharwa, "Optimizing the Prediction Accuracy of Concrete Compressive Strength Based on a Comparison of Data-Mining Techniques," *J. Comput. Civ. Eng.*, 2011, doi: 10.1061/(ASCE)CP.1943-5487.

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

[22] B. A. Young, A. Hall, L. Pilon, P. Gupta, and G. Sant, "Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: New insights from statistical analysis and machine learning methods," *Cem. Concr. Res.*, vol. 115, no. December 2017, pp. 379–388, 2019, doi: 10.1016/j.cemconres.2018.09.006.