Descriptive and predictive analytics of the cement and concrete production process

Juan Pablo Madrid Peláez[1]
Salomón Cardeño Luján[2]

Advisors:
Juan Carlos Rivera Agudelo[3]
Henry Laniado Rodas[4]

Research practice 2
Research proposal
Mathematical Engineering
Department of Mathematical Sciences
School of Sciences
Universidad EAFIT

June 2022

[1]email: jpmadridp@eafit.edu.co
[2]email: scardenol@eafit.edu.co
[3]CV: CvLAC, Research Group: Mathematical Modeling (Leader), email: jrivera6@eafit.edu.co
[4]CV: CvLAC, Research Group: Mathematical Modeling, email: hlaniado@eafit.edu.co

Abstract

The paper uses both a Machine Learning and a combined Taylor Modeling with Heuristics approach to predict the Compressive Strength, the main characteristic to determine the physicochemical behaviors of concrete and cement. Implementing a data-based approach, can result in a reduction of the variability of the process by improving the overall reliability. The AdaBoost, ANN and SVM algorithms were implemented, where the AdaBoost performed the best in terms of fitness, error, and predictions, which was expected due to its k-folds nature. To counter the black-box nature of Machine Learning, a Taylor Modeling algorithm was implemented to build the mathematical model of the data with the use of an evolutionary algorithm to find the best parameters for the model.

Keywords: Machine, Learning, Evolutionary, Algorithm, Predictions.

1 Introduction

The use of concrete and cement is of high relevance to the construction and building materials industry, presenting various advantages and properties over other materials which makes them widely used. Concrete is one of the most used building materials (Naseri *et al.*, 2020), is a mixture composed of cement, gravel, sand, admixtures, and water, with properties like malleability in its liquid form and high resistance when solid (ARGOS, 2021b). Cement, on the other hand, is both a raw material used in the concrete mixture and a building material that acts as a binder when mixed with water, due to the resulting properties of being adhesive and cohesive as a reaction with water (ARGOS, 2021a). It is of high interest to consider and identify variables (mechanical properties of the material, local properties of the environment, etc.) involved in the productive process relevant to the end-product of both building materials.

The construction and building materials industry is always evolving and facing new challenges, giving relevance to innovations and modern technologies or techniques such as the use of relevant data to build descriptive and predictive analytics towards particular goals. In the case of concrete and cement, it's of high interest to make estimates and/or predictions of mechanicals properties of the material (e.g. compressive strength) and mixture designs (sustainable purposes, performance improvement, estimates and/or predictions of mechanical properties), both variables relevant to the productive process and end-product. To name a few examples, consider the use of Machine Learning in concrete strength simulations (Chou *et al.*, 2014), the prediction of the compressive strength of concrete as a function of the mixture proportions by using statistical analysis and machine learning methods (Young *et al.*, 2019), the prediction of setting behavior and strength evolution of hydrating cement systems (Oey *et al.*, 2020), sustainable concrete mixture designs by a new machine learning technique (Naseri *et al.*, 2020), and the prediction of compressive strength for concrete with an adaptative boosting algorithm (Feng *et al.*, 2020).

Concrete and cement are building materials highly sensitive to changes in their components leading to economical, designing, and controlling efforts to handle variability in the production process which impacts the end-product. This presents the opportunity of a data-based approach where descriptive and predictive analytics are implemented in the cement and concrete production process to help reduce said variability, thus improving reliability in the mechanical properties, and sustainability in the mixture design. The success of the approach implies a positive impact on the economical and design efforts, control and reliability of mechanical properties (therefore the end-product), and the improvement of mixtures (eco-friendly and sustainable).

The contribution of this project is substantiated by the descriptive and predictive analysis constructed to the identified variables influential to variability in the concrete and cement production process. This involves preprocessing the available data, exploring the resulting data for variable dependances, representative variables, descriptive analysis, data selection for prediction, prediction analysis. These analytics will result in predicted behaviors for mechanical properties of the building materials and different mixture combinations. In particular, as the research takes part in a joint collaboration between the company ARGOS, UPB and EAFIT, it is not only beneficial for all parts, but it is of particular interest for ARGOS production process and decision-making.

2 Problem Statement

Concrete and cement are highly used materials in the construction and building materials industry around the world, this is partly because they have certain physicochemical properties and characteristics that contribute to this fact. When it comes to the production process, there's high sensitivity in the components of both materials which leads to end-products with high variability. This implies that any expectations or predictions of the desired properties (that made the materials so appealing and popular in the first place) become unreliable, which then generates negative impacts on quality control and assurance, reproducibility, economy, mixture design, order requirements, etc. On top of that, local conditions such as weather need to be taken into consideration, because they also have an impact in the production process and end-products (Ortiz *et al.*, 2005; Nasir *et al.*, 2020; Larsson & Rudberg, 2019; Schaefer *et al.*, 2006; Al-Negheimish & Alhozaimy, 2008). In the context of this paper, the idea is to mine or preprocess a database provided by Argos (a local cement manufacturing company) if available or to use public access databases, in order to apply statistical and machine learning techniques to construct descriptive and predictive analytics of the cement and concrete production process.

Because the problem can adopt both a general and a locally focused context, different approaches with different goals in mind can be found when doing a literature review. Nonetheless, the desired physicochemical properties which make concrete and cement so appealing and popular materials are always needed and/or considered. To name a few, properties such as compressive strength, CO_2 emissions, performance, mixture proportions, are always the subject of study even if the goals differ. This is because measures, standards and the behavior of the materials are often determined as a function of said properties. Another similarity lies in the fact that data-based approaches with statistical techniques have become somewhat common and combining them with the use of Machine Learning (ML) algorithms is a popular choice. The similarities and differences stated before can be found in works like: Machine learning prediction of mechanical properties of concrete: Critical review (Chaabene *et al.*, 2020), Machine Learning Techniques in Concrete Mix Design (Ziolkowski & Niedostatkiewicz, 2019), Neural Network, Machine Learning, and Evolutionary Approaches for Concrete Material Characterization (Rafiei *et al.*, 2016) and Comparison of machine learning techniques to predict compressive strength of concrete (Dutta *et al.*, 2018).

Another implication of the global and local nature of the problem is that it has repercussions in both scales. On a global scale, the problem impacts the worldwide construction and building materials industry by introducing a trade-off between appealing physicochemical properties that make concrete and cement competitive and popular choices amongst other materials and the fact that their components in the production process are so sensitive that end up producing high variability in what was expected of the resulting properties in the first place. On a local scale, the problem

retains the global scale impact but also presents a new one when local circumstances like weather also impact the production process by affecting concrete and cement properties and therefore the resulting variability. Generally speaking, the problem isn't new; but new approaches have been present with the implementation of new machine learning algorithms and methods, even heuristics and metaheuristics have been used in order to improve much further the solutions.

Finally, there are multiple solutions to this problem but not sufficiently satisfactory in the sense that new algorithms, methods, or modifications can always be presented in order to improve existing results. Also, if local conditions are to be considered, the solutions become more of a framework or a reference to be reproduced and adapted, nonetheless, even if the solutions are not sufficient, they are still useful and relevant to find particular ones.

2.1 Mathematical formulation of the problem

Dataset
Let $X = (X_1, X_2, X_3, ..., X_n)$ denote the dataset composed of $n \in \mathbb{N}$ input variables.

Preprocessing the dataset
Let Y denote the transformation of the dataset X, i.e., the resulting dataset after cleansing, scaling, transformation and/or reduction has been applied to X. The resulting Y is expressed as $Y = (Y_1, Y_2, Y_3, ..., Y_m)$ where $m \in \mathbb{N}$ and $m \leq n$. This implies that preprocessing the dataset X results in a dataset Y at most as big as X.

Training and testing data selection (dataset split)
Let A and B denote the training and testing data, where $Y = (A, B)$. This implies that the preprocessed dataset Y has been split in such a way that $A = (Y_1, Y_2, Y_3..., Y_i)$ and $B = (Y_{i+1}, Y_{i+2}, Y_{i+3}..., Y_m)$ where $i \in \mathbb{N}$ and $i < m$. Usually, the split isn't denoted this formally and only the resulting percentage of the data split is given, e.g. let the training data be 90% and the testing the remaining 10% of the dataset Y.

Machine Learning Method
Let M denote a Machine Learning Method, i.e., a mathematical method (usually programmed in a computer) composed of (has an input of) the tuple (A, R, B) where A is the training data, R are the training parameters and B is the testing data used for validation after the training process is done. The training parameters R depend on the specifics of the ML algorithm and thus expressed in a general manner.

Taylor Model
Let F denote a mathematical model that takes the tuple (A, d, v) where A is the training data, d is the degree of truncation of a Taylor serie resulting in a Taylor polynomial, and v corresponds to the number of variables or dimension of the Taylor polynomial. The model results in a mathematical expression with unknown coefficients.

Parameter estimation with evolutionary algorithms
Let E denote an evolutionary algorithm that takes the tuple (A, F, O), where A is the training data, F is the previously constructed Taylor Model, and O is an objective function. The evolutionary algorithm optimizes the objective function O in order to find the best parameters of the model F that best fit the data A.

3 Objectives

3.1 General objective

Develop descriptive and predictive analytics for the relevant variables (mechanical properties and environmental) in the productive process of concrete and cement that impact on variability with statistical techniques and machine learning to counter said variability thus resulting in improved reliability and mixture selection for the end-product of both building materials.

3.2 Specific objectives

- Preprocess or mine the available data by data cleansing, transformation and/or reduction.
- Perform an exploratory and descriptive analysis of the available data using statistical techniques.
- Predict the compressive strength of concrete and cement by the implementation of machine learning techniques.
- Build a mathematical model by the use of Taylor Modeling and estimate the best parameters for the model with evolutionary algorithms.

4 Scope

A partial limitation of the project is that the only database provided by ARGOS was the cement quality data base, which reduces the whole focus of this research to merely cement.

The implementation of statistical analysis, ML algorithms, Taylor modeling and the evolutionary algorithm were carried out in Python. The execution of the machine learning algorithms will be performed with the help of the Scikit-learn library.

One of the main expected results of this work is a computational tool based on Machine Learning that allows predicting with a certain degree of accuracy the compressive strength, CO_2 emission, and cost of concrete, given the physicochemical specifications of its components. On the other hand, it is also expected to obtain a mathematical model for the data, successfully constructed by the Taylor model algorithm and with effective estimations of its parameters by the use of the evolutionary algorithm.

5 State of the art

Computer modeling tools to study the properties of construction materials are booming (Boukhatem *et al.*, 2011), in the specific case of concrete, one of the most studied techniques for its analysis is Artificial Neural Networks (ANNs). Altun *et al.* (2008) showed that ANNs are superior to regression methods in estimating the strength and other behaviors of concrete. Uysal & Tanyildizi (2012) predicted compressive strength after exposing certain concrete to high temperatures using ANNs. Dantas *et al.* (2013) analyzed the behavior of concrete containing recycled aggregate concrete.

Support Vector Machines (SVM) is another Machine Learning technique quite used in the literature, for example Yu *et al.* (2018) used the SVM technique to predict the strength of high-performance concretes. Omran *et al.* (2016) compared the accuracy of different techniques to predict the behavior of various environmentally friendly concretes, including SVM.

Yeh (1998) published a well-recognized database, with which some authors train their ML algorithms, for example Feng *et al.* (2020) used the mentioned database to train an ensemble learning algorithm called Adaptive Boosting (AdaBoost), on the other hand, Chou *et al.* (2014) only used it to compare the results of his methods applied to several datasets.

One thing to notice is that, despite ML popularity in the topic, the need for explicit mathematical models is still present, which can be seen in the works of Naseri *et al.* (2020) to design sustainable concrete mixtures, Kheder *et al.* (2003) to predict cement compressive strength at ages of 7 and 28 days within 24 hours, or even in the works of Souto-Martinez *et al.* (2017) in the field of carbon sequestration prediction, to name a few. The use of evolutionary algorithms with the sole purpose of parameter estimation for the mathematical models (curve fitting) can be reviewed in the works of Yuan *et al.* (2014) with the use of hybrid genetic-based models (a class of evolutionary algorithms), or the works of (Asteris *et al.*, 2019) that uses an evolutionary algorithm for precise curve fitting. Despite this, the use of evolutionary algorithms with the purpose of curve fitting of a model in this context, is not that common in comparison to more general purpose approaches. An implementation in a wider focus with different objectives can be reviewed in the works of Chou *et al.* (2011) that implements a genetic algorithm to automatically produce self-organizing formulas (to already implemented methods) in order to make compressive strength predictions, or even Nguyen *et al.* (2020) that uses evolutionary algorithms for AI systems optimizations to predict a different property of concrete, to name a few. Lastly, because evolutionary algorithms are modular in nature, i.e., they use different operators in different problem contexts, they can be highly reusable, which implies that older works in curve fitting (in the general context) such as Gulsen *et al.* (1995) or Yoshimoto *et al.* (2003), can still be of relevance.

6 Methodology

6.1 Preprocessing

The first phase of the work consists of preprocessing the data delivered by ARGOS, this is of vital importance to eliminate errors and to make the data compatible with the machine learning (ML) algorithms used, such preprocessing possibly involves eliminating variables, applying certain scaling to others, eliminating outliers, etc.

6.2 Machine Learning

Although (Zhou, 2019) shows the superiority of ensemble methods over individual learning methods and (Feng *et al.*, 2020) shows that this is particularly true for a problem similar to ours, individual methods are not ruled out in this work, since local conditions such as ambient temperature and humidity are factors that can have a great influence on the behavior of cement and concrete, so local comparison of the behavior of multiple methods is of interest.

The ensemble learning algorithms applied in this work are Adaptive Boosting, Stacking, and Bagging. The individual learning methods to be compared for this work will be Support Vector Machines (SVM), Linear Regression (LR), Classification and Regression Tree (CART), and Artificial Neural Network (ANN). To evaluate the quality of the predictions the (T_i, P_i) tuples are constructed, where T_i is the value of the output variable of the ith observation of test set and P_i is the output value predicted by the model for that same observation. Obviously, the desired behavior of the plot

of all the (T_i, P_i) pairs is linear, therefore, the following 4 metrics of the error of fit of these pairs to their linear regression can be used:

- Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m}(P_i - T_i)^2}{m}} \tag{1}$$

- Coefficent of deterination R-Squared (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(P_i - T_i)^2}{\sum_{i=1}^{m}(T_i - \bar{T})^2} \tag{2}$$

- Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{100\%}{m}\sum_{i=1}^{m}\left|\frac{P_i - T_i}{T_i}\right| \tag{3}$$

- Mean Absolute Error (MAE)

$$MAE = \frac{\sum_{i=1}^{m}|P_i - T_i|}{m} \tag{4}$$

The four behavior estimators shown above, depending on the random selection of the test data, so they will be calculated several times with different selections of the test set, this process is known as k-folds cross-validation, where k is the number of different partitions to the data set, therefore, k also represents the times the estimator is calculated, figure 1 illustrates this process.

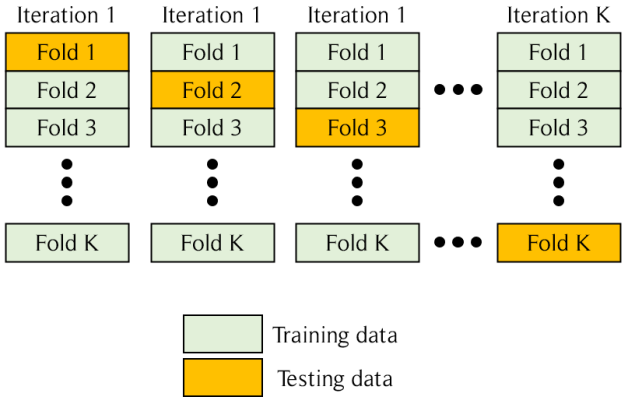

Figure 1: k-folds cross-validation

6.3 Taylor Modeling of the data

The fact that Machine Learning methods do not result in a concrete mathematical model due to their so called "Black Box" nature, in order to build a mathematical model that describes the data a metaheuristic-based programming approach is implemented by the use of a Taylor Modeling (TM) algorithm to build a general multivariate function for the data, and an Evolutionary Algorithm

(EA) that takes the model, the data and minimizes either the RMSE, R^2, MAPE and/or MAE, by calculating the optimal coefficients. Let $\mathbf{X} = (X_1, X_2, X_3, ..., X_n)$ denote the dataset composed of $n \in \mathbb{N}$ input variables, in order to model the output variable Y we use the following multivariate function

$$F(\mathbf{X}) = F(X_1, X_2, X_3, ..., X_n) = c_1 f(X_1) + c_2 f(X_2) + c_3 f(X_3) + ... + c_n f(X_n)$$

In this case, we focus on the use of an n-dimensional truncated centered Taylor Series (or MacLaurin), which results in the expression

$$F(\mathbf{X}) = \mathbf{a} \sum_{n=0}^{N} \frac{f^{(n)}(0)}{n!} \mathbf{X} + \mathcal{O}(N)$$

where $\mathbf{a} = (a_1, a_2, ..., a_N)$, $f^{(n)}(0)$ is the n-th derivative of the multivariate function f evaluated at vectorial 0, and $\mathcal{O}(N)$ represent the truncation error by using the finite Serie and stopping at term N instead of the infinite Serie. The idea behind this approach is to generate the model of the data $F(\mathbf{X})$ by using the Taylor Series up to a certain term (because all of the terms are not necessary) and the use of coefficients $\mathbf{a}$ to be minimized in the process afterwards.

6.4 Evolutionary Algorithm

An evolution algorithm is inspired by and/or based on nature, in particular, on the theory of evolution. To emulate this, it is usually put in a population dynamics or so-called population-based context with terms such as:

- **Population:** a group of individuals which corresponds to all of the solutions to the optimization problem of an specific generation.
- **Individual:** a single member of the population which corresponds to a single solution to the optimization problem. In genetic algorithms, it is usually denoted as a chromosome, and each parameter value is denoted as a gene.
- **Generation:** an iteration of the population in which many actions take place: new individuals arise, old individuals leave, etc. It corresponds to a single loop of the evolutionary algorithm main loop or cycle.

The population dynamics are composed of certain characteristics and so-called operators such as:

- **Fitness:** a characteristic of each individual of the population, that determines how likely are the chances of survival for each generation of the population.
- **Initial population:** who are to be the first individuals and how are they generated.
- **Selection operator:** which individuals of the population are to be selected for mating and to generate offspring. This is related to the probability of survival, and they are usually chosen based on their fitness.
- **Crossover operator:** how do the individuals mate to generate offspring. They are usually referred to as parents.

- **Mutation operator:** which individuals are to be affected by a change in their characteristics and how. Usually this is implemented with a certain probability and with random changes in the information of the mutated individual. The idea behind this is to allow new individuals with new information to enter the population.
- **Population update:** how to update the population on each generation, which individuals are to remain and to leave.
- **Population size:** should the population be allowed to grow on each generation (dynamic), or should it maintain a fixed size (non-dynamic).

To clarify, these are the most common characteristics and operators of Evolutionary Algorithms but depending on the combination of these or even the removal or addition of new operators, the algorithm gets called by its class such as a genetic algorithm, a differential evolution algorithm, and so on. In our case, it was of interest to have a reasonable amount of freedom to evaluate and implement different characteristics and operators, which resulted in a hybrid evolutionary algorithm.

The idea of the behavior of any evolutionary algorithm, as described before, can be easily grasped when put as a pseudocode, such as the one below.

Algorithm 1: Basic Evolutionary Algorithm

```
P ← generate_population
for i = 1 to generations do
    (x, y) ← selection(P)
    offspring← crossover(x, y)
    if random < prob_mutation then
        offspring← update(P)
return better solution in P
```

Initially, the implemented algorithm started as a genetic algorithm, but through experimentation it evolved into an hybrid evolutionary algorithm with the following characteristics and operators:

- **Fitness function:** in this case, the mean squared error (MAE) was considered to find the optimal parameters. This implies that the problem is of minimization.
- **Initial population:** randomly generated with Uniform, Sobol and Saltelli pseudorandom number generators. The idea behind this was to initially cover as uniformly as possible the parameter space, allowing to a wider region of search with good quality. The intervals for each parameter were determined trhough previous experimentation, and thus resulting in the use of the same interval for each parameter, but it is also possible to tune the intervals per parameter.
- **Selection:** an elite uniform selection was implemented, were half of the individuals (lowest MAE) are selected by a uniform weighted sampling method.
- **Crossover:** the selected individuals mate and generate offspring through arithmetic mean, meaning that for each pair of parents a single offspring is generated.

- **Mutation:** to mutate, instead of using a usual probability-bound mutation, few individuals are selected similarly to the **Selection** operator. Then, the range of each parameter is calculated as the difference between each maximum and minimum. Afterwards, the range is multiplied by a random uniform factor in $[-2, 2]$, as implemented by Gulsen *et al.* (1995).
- **Population update:** each generation the population is updated by a combination of the elite and the best of the current generation. This means that the population is updated by selecting the best of the previous generation, the offspring of the current generation, and the mutated individuals of the current generation.
- **Population size:** a non-dynamic approach was implemented. After the population has been updated, the population cuts out the worst individuals until the population size becomes the same as the original size.
- **Local search:** as the domain of every parameter of the model is $\mathbb{R}$ this means that the problem is continuous. Because of this, a simple local search that generates individuals with sign changes in the parameters (combinations of sign chane) was implemented in order to escape local optima. Instead of simply replacing the best individual of the population with the best individual from the search, all possible better individuals replace the corresponding individuals in the population. The local search was performed if after 10 consecutive generations there was not a change in the best overall fitness.

7 Results

7.1 Machine Learning

The Machine Learning methods were implemented in the programming language *Python 3* in the OS *Windows 10 Home Single Language*. Three Machine Learning (ML) algorithms were implemented to determine the best algorithm based on the 4 metrics previously discussed: AdaBoost (Adaptative Boosting), ANN (Artificial Neural Network) and SVM (Support Vector Machine). The focus was on the AdaBoost algorithm, thus the idea behind the use of ANN's and SVM's was to make a comparison with the more traditional ML methods. The resulting performance of the methods can be observed in the table 1 below.

Algorithm	$\mathbf{R^2}$	**RMSE**	**MAPE**	**MAE**
AdaBoost	0.916	2.20	8.80%	3.54
ANN	0.838	5.14	11.87%	5.31
SVM	0.787	6.28	17.05%	6.34

Table 1: Resulting performance of the ML methods.

By analyzing the results, the AdaBoost algorithm resulted in the best performance for the data. This is because in terms of the R^2 metric its value was the highest, meaning it had the best fit for the data. On the other hand, it resulted in the lowest values for all the error metrics: RMSE, MAPE and MAE; meaning it had the least error when predicting the data. The ANN was the second-best algorithm followed by the more traditional SVM. The observed behavior was as expected, validating the well performance of the AdaBoost compared to the more traditional ML methods.

One thing that is worth noting is the fact that the resulting performance was obtained without the elimination of data outliers. The context of outlier elimination as described in Naseri *et al.* (2020) proposes the elimination of outliers by using ML methods to determine a metric called the $E30$ defined as follows

$$E30 := \frac{\text{number of data which their mean absolute percentage error is less than } 30\% \cdot 100}{n}$$

The authors stated that whatever data that does not fall inside this metric could be interpreted as an outlier and thus justified a simple deletion. The mistake of this approach is the fact it ignores the possible existence of relationship between the input variables, which justifies the need for a more suitable approach. The idea was to use a kind of "cross-validation" approach by using the AdaBoost method as shown in the figure 2 below.

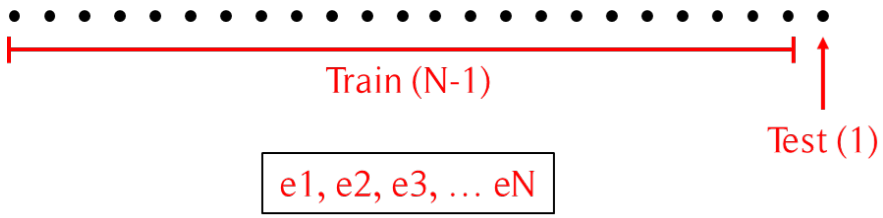

Figure 2: Proposed outlier categorization and elimination.

The idea behind this method is to train the AdaBoost algorithm on each iteration i with a $N - i$ training data set whilst having a testing data set of size i. The stop criteria are previously determined as the number of iterations based on the percentage of either the desired usable data or the desired amount of outlier data to be removed. Notice that if we wish to remove n data

$$\begin{aligned} N + (N-1) + (N-2) + ... + N - (n+1) &= \sum_{i=1}^{n} N - i \\ &= nN - \frac{n(n+1)}{2} \end{aligned}$$

and therefore

$$\therefore n = \alpha N \Rightarrow \mathcal{O}(N^2)$$

which is the resulting complexity of the process. After eliminating outliers by the use of the previous method, the AdaBoost method was trained with the data resulting in the Compressive Strength predictions (output variable) shown in the right side of figure 3 below. On the right side of figure 3 the scatter plot of the predictions compared to the true values can be observed.

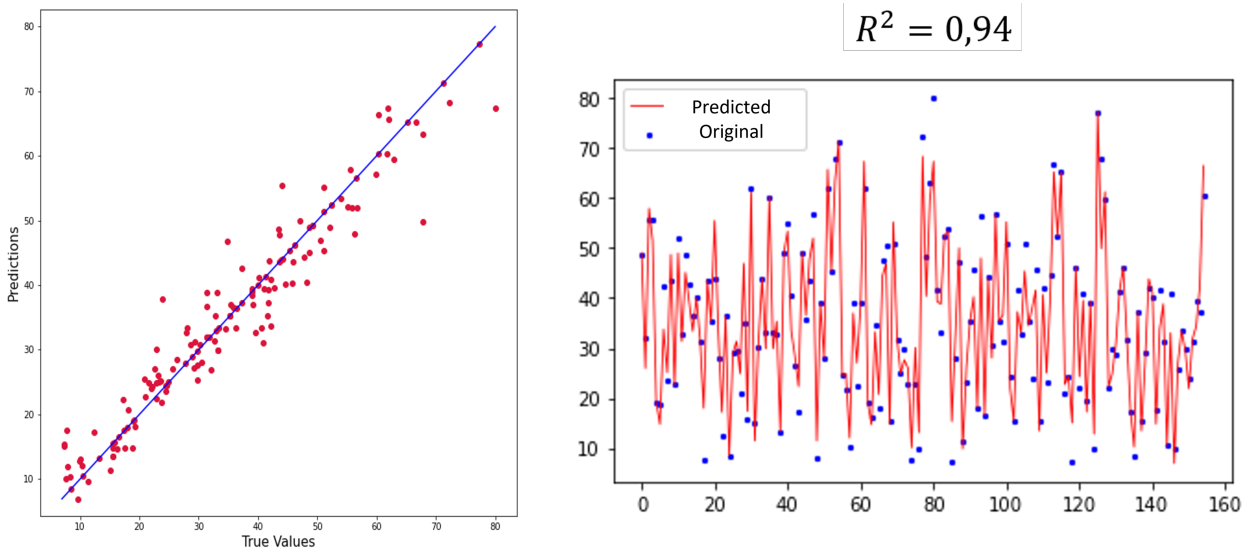

Figure 3: Scatter plot (left side) and plot of Predictions vs Data (right side) of the output variable (Compressive Strengh *MPa*).

It is observed that the AdaBoost method follows the expectations in a resulted $R^2 = 0.94$, implying a great fit for the data. By observing the general behavior of the predictions, it is clear that the AdaBoost performs well.

Finally, the most important characteristic for both concrete and cement is the Compressive Strength, as previously stated, but one of the most observed behaviors is the one obtained by considering the Compressive Strength as a function of Time. Figure 3 below shows the comparison between different behaviors for the compressive strength over time for different mixtures (left side) and the obtained behavior via the AdaBoost method (right side).

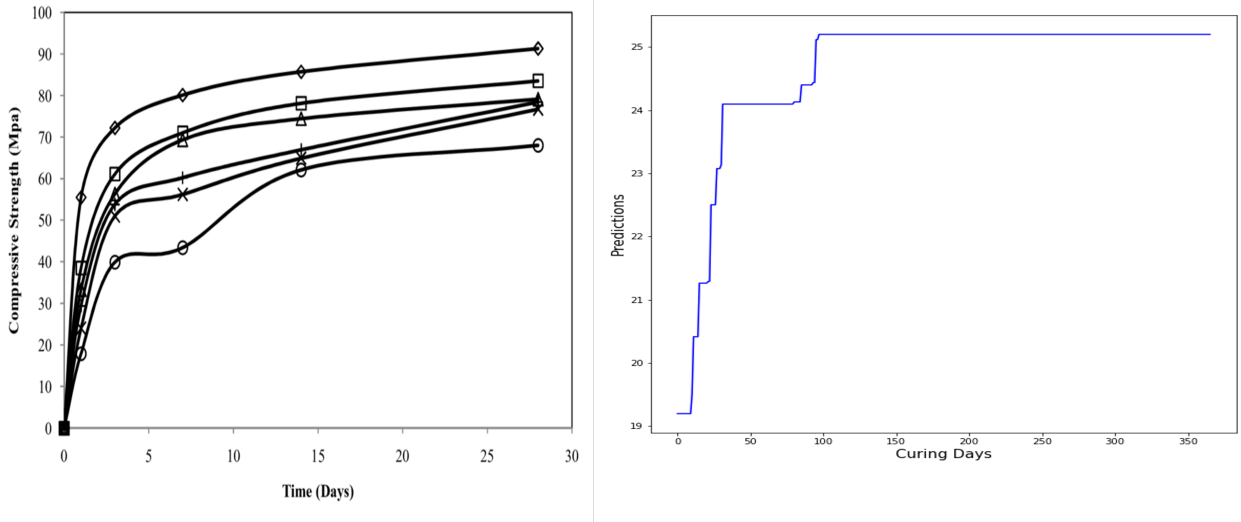

Figure 4: Concrete Compressive Strength over time for different mixtures (left side) and the resulting behavior obtained via the AdaBoost method (right side).

Again, the AdaBoost method shows a particularly good approximation of the expected behavior, predicting a reasonable behavior for the Concrete's Compressive Strength over its Curing time. One thing to notice is the fact that no smoothing was implemented, but if a smoothing method were implemented like a Spline for example, the result would be in much more proximity of the expected.

7.2 Taylor Modeling and Evolutionary Algorithm

Both the Taylor modeling and the hybrid evolutionary algorithms were implemented in the programming language *Python 3* in the OS *Windows 10 Home Single Language*. The Taylor modeling algorithm was implemented by developing a custom function based on symbolic math with the help of the package `SymPy`. Through the preprocessing an exploratory analysis was performed in order to identify the relationship between the data. The resulting Spearman correlation matrix can be observed in the figure 5 below.

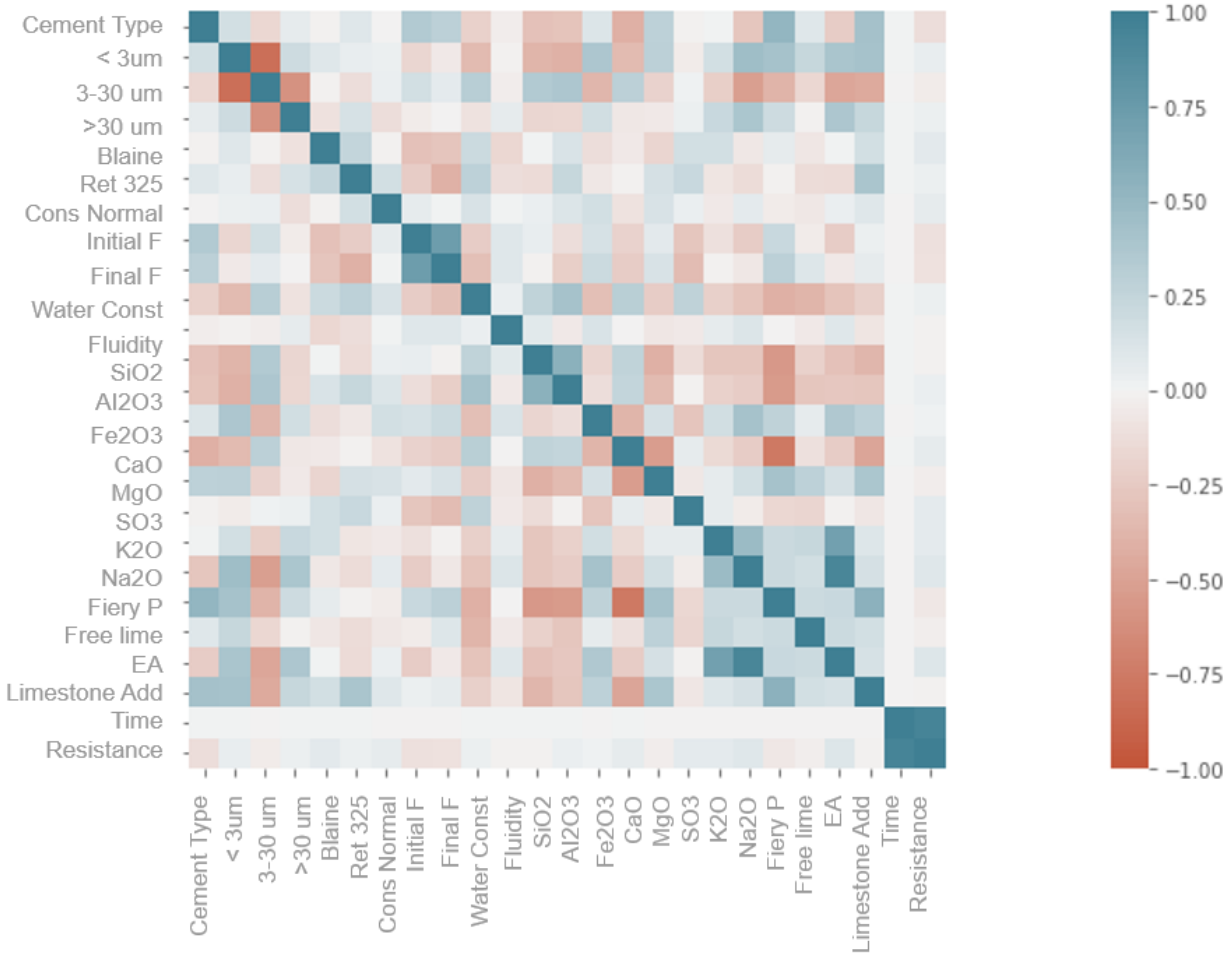

Figure 5: Spearman correlation matrix of the cement quality data.

It is easy to observe that a high correlation exits between certain variables. Three of the variables that resulted in a higher correlation with compressive strength (*Resistencia* in spanish) were time (*Tiempo*), EA and Initial F (*F Inicial*). These variables are selected as the predictors to build the

mathematical model. By continuing the exploration, it was also found that 2 types of cement were present in the data: UEM and IND. In order to see if the compressive strength for both types follow the desired log-like behavior over time (as observed in figure 4), a box-plot was implemented for the compressive strength over time for each cement type. The results can be observed in the figure 6 below.

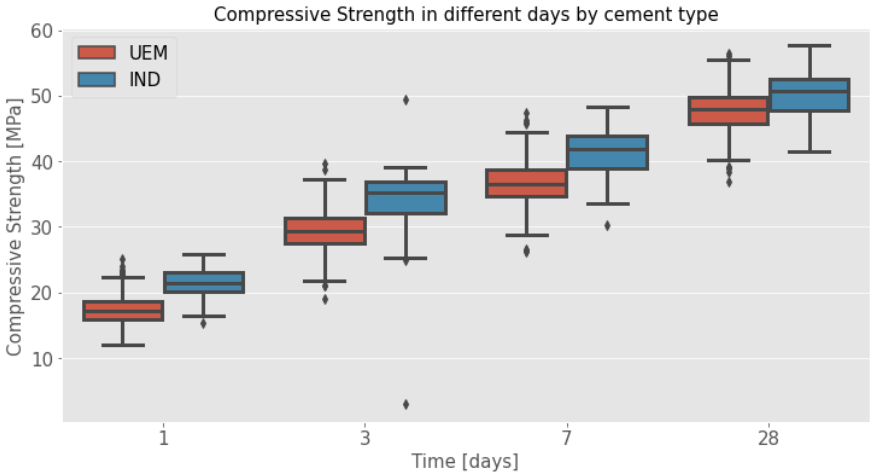

Figure 6: Box-Plot of the Compressive Strength over time for each cement type.

It can be immediately observed that the desired log-like behavior for the compressive strength over time is present for both types of cement. Notice how the time variable contains only the discrete values 1, 3, 7 and 28, this is also an expected behavior since the compressive strength is usually measured in those specific points in time. By filtering the data based on the type of the cement, it was observed that only 359 samples were of IND cement while, on the other hand, 2629 samples were of UEM cement type. Because of this, the data set was narrowed to only the UEM samples. After the exploratory analysis the Taylor model was implemented with the selected 3 variables, and a degree of 2. The output of the algorithm can be observed in the figure 7 below.

$$a_0x_0 + a_1x_1 + a_2x_2 + \frac{a_3x_0^2}{2} + \frac{a_4x_1^2}{2} + \frac{a_5x_2^2}{2} + a_6x_0x_1 + a_7x_0x_2 + a_8x_1x_2 + a_9$$

Figure 7: Code output for the Taylor Model algorithm of 3 variables and of 2nd degree.

Having obtained the Taylor polynomial, the hybrid evolutionary algorithm was implemented. A population size of 100 individuals was selected, were every individual contained each of the $a_0, ..., a_9$ parameters. The initial population was generated by using the Saltelli method with parameter bounds of $[-40, 40]$ for every parameter. As mentioned before, these parameter bounds were obtained by previous experimentation. Instead of using fitness tolerance or time as stopping criterions, the use of a number of generations parameter was implemented. In this case, a value of 100 generations was chosen after previous experimentation. The results of the hybrid evolutionary algorithm can be observed in the figure 8 below.

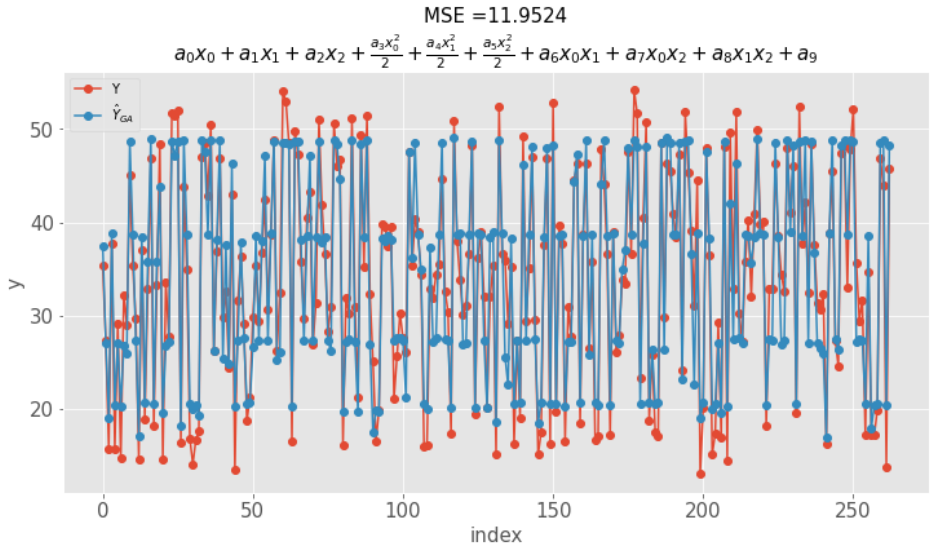

Figure 8: Comparison between the real compressive strength data (in red) and the predicted compressive strength data (in blue) from the resulting model.

The hybrid evolutionary algorithm resulted in a mean squared error of $MSE = 11.95$, which is a decent result considering the low complexity of the mathematical model and comparing it to a regular least squares fit that resulted in an $MSE = 9.88$. On the other hand, the results for the model fit can be observed in the figure 9 below.

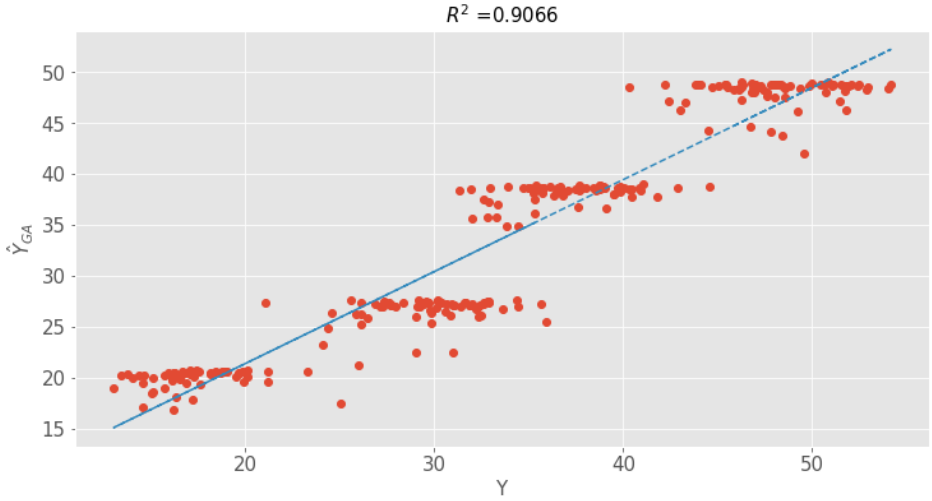

Figure 9: Resulting model fit.

The model fit resulted in an $R^2 \approx 0.91$ which is a good result, especially when compared to the $R^2 \approx 0.92$ obtained by the regular least squares fit. Overall, the resulting MSE and R^2 indicate that the model is good in comparison of a regular least squares fit, even if it's outperformed by the last one.

8 Conclusions and future research

The implementation of new machine learning methods such as the AdaBoost Algorithm shows a significant advantage over the more traditional methods by using Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs). This was observed clearly in the results of the fit metric R^2 and the error metrics RMSE, MAPE and MAE, where the AdaBoost method was the best overall, followed by the ANN and the SVM in the last place. The implementation of an outlier elimination method that considers the contribution of all the variables resulted in a more suitable approach than the $E30$ metric proposed in the papers, resulting in a more "computationally expensive" method with a complexity of $\mathcal{O}(N^2)$ but preferable for its resulting quality even with the added computational cost. The expected behavior of concrete's Compressive Strength over its curing time was able to replicate in a very close approximation with the AdaBoost method. If a smoothing method were to be implemented afterwards, it would result in an even more close approximation. The fact the AdaBoost was able to replicate the overall behavior of the data makes it a suitable candidate for predictions of the Compressive Strength and its behavior over its curing time, leading to an improve of the reliability in the production process.

In the case of the mathematical model, selecting the three most influential variables for the cement compressive strength and choosing to truncate the Taylor Serie up to the second degree, resulted in a simple model. After finding the optimal parameters that minimize the mean squared error by the use of the hybrid evolutionary algorithm, the resulting MAE and R^2 where really close in comparison to common least squares fit, and even if it was slightly outperformed by the last one, it was not by a large margin.

Regarding future research, in the case of the mathematical modeling of the data, the Evolutionary Algorithm is yet to be finish. Even if the results were good, the use of more complex models (higher dimensions and higher degree order Taylor polynomials) was not assessed. Also, there have been recent purposes for modifications of the algorithm such as implementing a weighted crossover operator that could not only improve, but also result in more offspring. A new mutation operator with random swaps, fitness-dependent local search, and rotation-based methods to widen the search area and diversity. A different approach to how the population is updated by trying methods such as a combination between best and diversity, fitness, and a similarity metric, and even the implementation of immigrants (nonparametric copula with noise). Finally, a post-optimization that adds new terms to the model as a refinement.

On the other hand, the preprocessing of the available dataset provided by ARGOS has been done only to preselected quality datasets, ignoring the data of the other process data sets. This implies the need of a proper and rigorous linking of all the data sets in order to use all of the information at our disposal. Finally, the optimization of the code has not been revised whatsoever, meaning the possibility of optimizing computational resources.

Acknowledgements

We would like to thank EAFIT University for the oportunity of the research.

We would also like to extend our gratitude to Professor Juan Carlos Duque for his guidance and feedback throughout the extension of the course.

Finally, we would like to thank ARGOS and UPB for being our partners and allowing the existence of this research in the first place.

References

Al-Negheimish, Abdulaziz I, & Alhozaimy, Abdulrahman M. 2008. Impact of extremely hot weather and mixing method on changes in properties of ready mixed concrete during delivery. *ACI Materials Journal*, **105**(5), 438.

Altun, Fatih, Kişi, Özgür, & Aydin, Kamil. 2008. Predicting the compressive strength of steel fiber added lightweight concrete using neural network. *Computational Materials Science*, **42**(2), 259–265.

ARGOS. 2021a. *Cement*.

ARGOS. 2021b. *Concrete*.

Asteris, Panagiotis G, Ashrafian, Ali, & Rezaie-Balf, Mohammad. 2019. Prediction of the compressive strength of self-compacting concrete using surrogate models. *Comput. Concr*, **24**(2), 137–150.

Boukhatem, Bakhta, Kenai, Said, Tagnit-Hamou, Arezki, & Ghrici, Mohamed. 2011. Application of new information technology on concrete: an overview. *Journal of Civil Engineering and Management*, **17**(2), 248–258.

Chaabene, Wassim Ben, Flah, Majdi, & Nehdi, Moncef L. 2020. Machine learning prediction of mechanical properties of concrete: Critical review. *Construction and Building Materials*, **260**, 119889.

Chou, Jui-Sheng, Chiu, Chien-Kuo, Farfoura, Mahmoud, & Al-Taharwa, Ismail. 2011. Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques. *Journal of Computing in Civil Engineering*, **25**(3), 242–253.

Chou, Jui-Sheng, Tsai, Chih-Fong, Pham, Anh-Duc, & Lu, Yu-Hsin. 2014. Machine learning in concrete strength simulations: Multi-nation data analytics. *Construction and Building Materials*, **73**, 771–780.

Dantas, Adriana Trocoli Abdon, Leite, Monica Batista, & de Jesus Nagahama, Koji. 2013. Prediction of compressive strength of concrete containing construction and demolition waste using artificial neural networks. *Construction and Building Materials*, **38**, 717–722.

Dutta, Susom, Samui, Pijush, & Kim, Dookie. 2018. Comparison of machine learning techniques to predict compressive strength of concrete. *Computers and Concrete*, **21**(4), 463–470.

Feng, De-Cheng, Liu, Zhen-Tao, Wang, Xiao-Dan, Chen, Yin, Chang, Jia-Qi, Wei, Dong-Fang, & Jiang, Zhong-Ming. 2020. Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Construction and Building Materials*, **230**, 117000.

Gulsen, M, Smith, AE, & Tate, DM. 1995. A genetic algorithm approach to curve fitting. *International Journal of Production Research*, **33**(7), 1911–1923.

Kheder, GF, Al Gabban, AM, & Abid, SM. 2003. Mathematical model for the prediction of cement compressive strength at the ages of 7 and 28 days within 24 hours. *Materials and structures*, **36**(10), 693–701.

Larsson, Robert, & Rudberg, Martin. 2019. Impact of Weather Conditions on In Situ Concrete Wall Operations Using a Simulation-Based Approach. *Journal of Construction Engineering and Management*, **145**(7), 05019009.

Naseri, Hamed, Jahanbakhsh, Hamid, Hosseini, Payam, & Nejad, Fereidoon Moghadas. 2020. Designing sustainable concrete mixture by developing a new machine learning technique. *Journal of Cleaner Production*, **258**, 120578.

Nasir, Muhammad, Gazder, Uneb, Maslehuddin, Mohammed, Al-Amoudi, Omar S Baghabra, & Syed, Imran Ali. 2020. Prediction of properties of concrete cured under hot weather using multivariate regression and ANN Models. *Arabian Journal for Science and Engineering*, **45**(5), 4111–4123.

Nguyen, Hung Quang, Ly, Hai-Bang, Tran, Van Quan, Nguyen, Thuy-Anh, Le, Tien-Thinh, & Pham, Binh Thai. 2020. Optimization of artificial intelligence system by evolutionary algorithm for prediction of axial capacity of rectangular concrete filled steel tubes under compression. *Materials*, **13**(5), 1205.

Oey, Tandre, Jones, Scott, Bullard, Jeffrey W, & Sant, Gaurav. 2020. Machine learning can predict setting behavior and strength evolution of hydrating cement systems. *Journal of the American Ceramic Society*, **103**(1), 480–490.

Omran, Behzad Abounia, Chen, Qian, & Jin, Ruoyu. 2016. Comparison of data mining techniques for predicting compressive strength of environmentally friendly concrete. *Journal of Computing in Civil Engineering*, **30**(6), 04016029.

Ortiz, J, Aguado, A, Agulló, L, & García, T. 2005. Influence of environmental temperatures on the concrete compressive strength: Simulation of hot and cold weather conditions. *Cement and concrete research*, **35**(10), 1970–1979.

Rafiei, Mohammad H, Khushefati, Waleed H, Demirboga, Ramazan, & Adeli, Hojjat. 2016. Neural Network, Machine Learning, and Evolutionary Approaches for Concrete Material Characterization. *ACI Materials Journal*, **113**(6).

Schaefer, Vernon R, Wang, Keijin, *et al.* 2006. *Mix design development for pervious concrete in cold weather climates*. Tech. rept. Iowa. Dept. of Transportation. Highway Division.

Souto-Martinez, Adriana, Delesky, Elizabeth A, Foster, Kyle EO, & Srubar III, Wil V. 2017. A mathematical model for predicting the carbon sequestration potential of ordinary portland cement (OPC) concrete. *Construction and building materials*, **147**, 417–427.

Uysal, Mucteba, & Tanyildizi, Harun. 2012. Estimation of compressive strength of self compacting concrete containing polypropylene fiber and mineral additives exposed to high temperature using artificial neural network. *Construction and Building Materials*, **27**(1), 404–414.

Yeh, I-C. 1998. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research*, **28**(12), 1797–1808.

Yoshimoto, Fujiichi, Harada, Toshinobu, & Yoshimoto, Yoshihide. 2003. Data fitting with a spline using a real-coded genetic algorithm. *Computer-Aided Design*, **35**(8), 751–760.

Young, Benjamin A, Hall, Alex, Pilon, Laurent, Gupta, Puneet, & Sant, Gaurav. 2019. Can the compressive strength of concrete be estimated from knowledge of the mixture proportions?: New insights from statistical analysis and machine learning methods. *Cement and Concrete Research*, **115**, 379–388.

Yu, Yang, Li, Wengui, Li, Jianchun, & Nguyen, Thuc N. 2018. A novel optimised self-learning method for compressive strength prediction of high performance concrete. *Construction and Building Materials*, **184**, 229–247.

Yuan, Zhe, Wang, Lin-Na, & Ji, Xu. 2014. Prediction of concrete compressive strength: Research on hybrid models genetic based algorithms and ANFIS. *Advances in Engineering Software*, **67**, 156–163.

Zhou, Zhi-Hua. 2019. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC.

Ziolkowski, Patryk, & Niedostatkiewicz, Maciej. 2019. Machine learning techniques in concrete mix design. *Materials*, **12**(8), 1256.