applied sciences

Type of the Paper (Article.)

Modeling of compressive strength of self-compacting rubberized concrete using machine learning

Miljan Kovačević [1]*

[1] University of Pristina, Faculty of Technical Sciences, Knjaza Milosa 7, 28000 Kosovska Mitrovica, Serbia ; miljan.kovacevic@pr.ac.rs

* Correspondence: miljan.kovacevic@pr.ac.rs; Tel.: +381 606173801

Citation: Lastname, F.; Lastname, F.; Lastname, F. Title. *Appl. Sci.* **2021**, *11*, x. https://doi.org/10.3390/xxxxx

Academic Editor: Firstname Lastname

Received: date
Accepted: date
Published: date

Abstract: This paper gives a comprehensive overview of the state-of-the-art machine learning methods that can be used for estimating self-compacting rubberized concrete (SCRC) compressive strength, including multilayered perceptron artificial neural network (MLP-ANN), ensembles of MLP-ANNs, regression tree ensembles (random forests, boosted and bagged regression trees), support vector regression (SVR) and Gaussian process regression (GPR). As a basis for the development of the forecast model, a database was obtained from an experimental study containing a total of 166 samples of SCRC. Ensembles of MLP-ANNs have shown the best performance in forecasting with the mean absolute error (MAE) equal to 2.81% and the Pearson's linear correlation coefficient (R) equal to 0.96. The significantly simpler GPR model had almost the same accuracy criterion values as the most accurate model, furthermore feature reduction is easy to combine with GPR using automatic relevance determination (ARD), leading to models with better performance and lower complexity. GPR with ARD is not previously used for estimating SCRC compressive strength.

Keywords: self-compacting rubberized concrete; compressive strength; machine learning; artificial neural networks; regression tree ensembles, support vector regression, Gaussian process regression

1. Introduction

Since 2006, the European Union has banned all types of waste tire disposal because the lengthy process of tire degradation damages the environment and nature. There are fewer and fewer natural resources used in concrete production in the world, so the emphasis is placed on the use of waste products from other industries and the replacement of recycled aggregates such as recycled rubber (Yildirim). The waste rubber has a significant effect on the concrete's properties in the fresh state. The addition of recycled rubber reduces the penetration of aggressive substances into the material, ensuring less permeability of the concrete and thus improving its durability. In addition to its durability, it also improves impact and wear resistance and other mechanical properties. Depending on the content and size of the rubber fraction, it reduces the compressive strength, reduces the shrinkage and thermal conductivity coefficient, and increases the freezing resistance and the sound absorption coefficient.

Self-compacting concrete (SCC) is used to increase productivity, ie lower costs and faster operating speed and better working conditions. SCC is a special type of concrete that does not need to use concrete compacting devices during casting [1, 2]. Its selfbuildability and self-leveling feature has eliminated the need of expensive vibrating equipment, reduced the cost of time of construction and the number of workers on site, but also increased safety at the site.

Unlike ordinary concrete, SCC contains a higher proportion of fine aggregate particles, and the water-binding ratio is lower, which affects the required force for the flow of concrete. Such a composition results in a decrease in the viscosity that is resolved, already mentioned, by the addition of superplasticizers. Large amounts of recycled rubber particles in the material reduce the possibility of filling the formwork without additional vibration. In order to ensure the same properties or approximate properties of SCRC as in the reference mixture, it is necessary to modify and adjust the proportion of admixtures to the concrete. In this way, the concrete behavior of the rubber with the rubber without the rubber in the fresh state is ensured. Even when the content of rubber particles are limited to maximum of 20 – 30 %, mechanical properties must be improved by adding higher content of cement and lower water to binder ratio or adding supplementary cementing materials, such as slag, silica fume or/and fly ash.

Moreover, the key indicator, commonly used for assessing the strength, the compressive strength of SCRC, generally decreases with the increase in content of rubber in SCRC. There are no expressions in building codes (for example, Eurocode 2 or ACI Committee 209) for the prediction of compressive strength of rubberized concrete, and especially of SCRC. Models from the literature for predicting the compressive strength of rubberized concrete given by researchers are presented in this article. However, they are based on a reduction coefficient with respect to the referent mixture of concrete without recycled rubber particles. It means that it is always necessary to make a reference mixture of concrete without the addition of crumb rubber particles. Therefore, in this article, an effort was made to model the compressive strength of SCRC by adopting one of the machine learning methods. So far, metaheuristic methods, and especially neural networks, have been successfully applied in various fields, such as in the control and optimization of processes, economics, medicine, engineering, etc. []. They have also been used to model the properties of concrete in the fresh or solid state [], but much less in concrete with the addition of rubber []. Some of the researches are described in the following described studies.

Due to its simplicity and efficiency, artificial neural networks (ANNs) have grown in interest and application in recent years in various fields such as economics, medicine and engineering (Nikoo et al 2015, Sadowski et al 2019). However, not so much research has been done on the modelling/prediction of compressive strength using ANN (Topcu and Sarıdemir 2008, Abdollahzadeh et al. 2011, Diaconescu et al. 2013).

Modelling the compressive strength of rubberized concrete using ANN, as the main mechanical characteristic, can be found in following articles.

Based on the database containing experimental studies of 70 rubberized concrete mixtures, Gesoglu et al. 2010 proposed ANN-based explicit models with 8 inputs (cement, coarse aggregate, fine aggregate, water, silica fume, crumb rubber, tire rubber chips and super plasticizer) for the prediction of output - mechanical properties of rubberized concretes (compressive strength, static elastic modulus and splitting tensile strength).

Li et al. (2011) applied an ANN with 6 input parameters (cement, sand ratio, water/binder, blast furnace slag, fly ash and super plasticizer,) and 3 output parameters (slump flow, slump and V-test) to predict the workability of self-compacting concrete on totally 23 mixtures. The average errors they achieved for slump flow, slump and V-test were 2.2 %, 2.7 % and 7.0 %, respectively.

El-Khoja et al. (2018) developed an ANN model with 5 input parameters (w/c ratio, fine aggregate, coarse aggregate, coarse rubber and fine rubber) to predict the compressive strength (output parameter) on the database of total number of 287 different experimental data.

Hadzima-Nyarko et al. (2019) collected so far the largest database of rubberized concrete with 457 different experimental data and performed ANN with six input parameters (cement, w/c ratio, fine aggregate, coarse aggregate, coarse rubber and fine rubber) to model the compressive strength.

Except ANN models, some other methods, such as k-nearest neighbor (KNN), regression trees (RT), and random forests (RF) have also been applied to estimate the compressive strength of concrete. For example, Ahmadi-Nedushan (2012) developed a KNN model to predict the compressive strength of concrete with 104 experimental data; Chopra et al. (2018) estimated the compressive strength of concrete using the RF model with 49 data points.

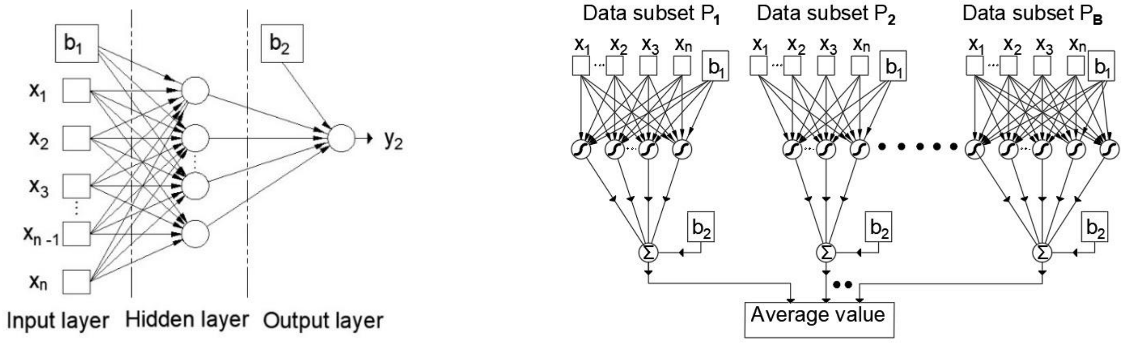
The outcomes of the above studies are certainly encouraging, especially considering the fact that applications of ML models to estimate the compressive strength of concrete are still at an early stage.

2. Methods

2.1 Multilayered perceptron artificial neural network (MLP-ANN)

Artificial neural networks are based on the parallel processing of various information similar to the human brain. They contain artificial neurons that are interconnected into a single parallel structure. A multilayer perceptron is a neural network with forward signal propagation that consists of at least three layers of neurons: input, hidden, and output layers.

In the general case, each neuron of one layer is connected to each neuron of the next layer, as shown in Figure 1 for the example of a three-layer MLP network with n inputs and one output. The properties of the network depend on the number of neurons and the type of activation function, so if the network is used as a universal approximator, then it must use nonlinear activation functions in the hidden layer to be able to approximate nonlinear relationships between input and output variables [1]. A model with one hidden layer whose neurons have a sigmoid activation function, while output layer neurons have a linear activation function, can approximate an arbitrary function when there is a sufficient number of neurons in the hidden layer [1].



(**a**) (**b**)

Figure 1. (**a**) Multilayer perceptron artificial neural network; (**b**) An ensemble of neural networks formed by the Bootstrap Aggregating (Bagging) approach [2,3]

As the number of input neurons is determined by the dimensions of the input vector and the number of output neurons by the dimension of the output vector, determining the network structure is reduced to determining the optimal number of hidden layer neurons if MLP architecture with the property of universal approximator is used.

The method for precise and reliable determination of the minimum required number of neurons has not been determined yet. What can be determined to some extent is the upper limit, i.e., the maximum number of hidden layer neurons, which can be used to model a system represented by a specific set of data. It is proposed to take into account a smaller amount of N_H from the set of inequalities (1) and (2), where N_i denotes the number of neural network inputs and N_s denotes the number of training samples [2,3].

$$N_H \leq 2 \times N_i \tag{1}$$

$$N_H \leq \frac{N_s}{N_i + 1} \tag{2}$$

In order to improve the generalization of the model, and when there is data with noise and a smaller set of data, it is possible to train a larger number of neural networks and find the mean value of their outputs. In this way, ensemble models are created, while the individual models that make up the structure of the ensemble are called base models or submodels. The data set based on which the ensemble models were trained in each iteration is formed by the Bootstrap method [4]. The Bootstrap method formed a set of the same size as the original data set.

2.2 Regression tree ensembles

2.2.1 Bagging

Methods based on classification trees (Classification And Regression Trees - CART) use the segmentation of the space of input variables in multidimensional rectangles or so-called boxes and then apply a model where multidimensional rectangles are assigned the appropriate value [4-6]. The lines that segment space has the form $\mathrm{X_i} = c$, with the remark that binary segmentation of the space is applied. Depending on the value of the input variables, the regression model (Figure 2) assigns a constant value of c_m to each of the mentioned regions, which is equal to the mean value of the output variable for that region R_m, i.e. in this case:

$$\hat{f}(X) = \sum_{m=1}^{5} c_m I\{(X_1, X_2) \in R_m\}. \tag{3}$$

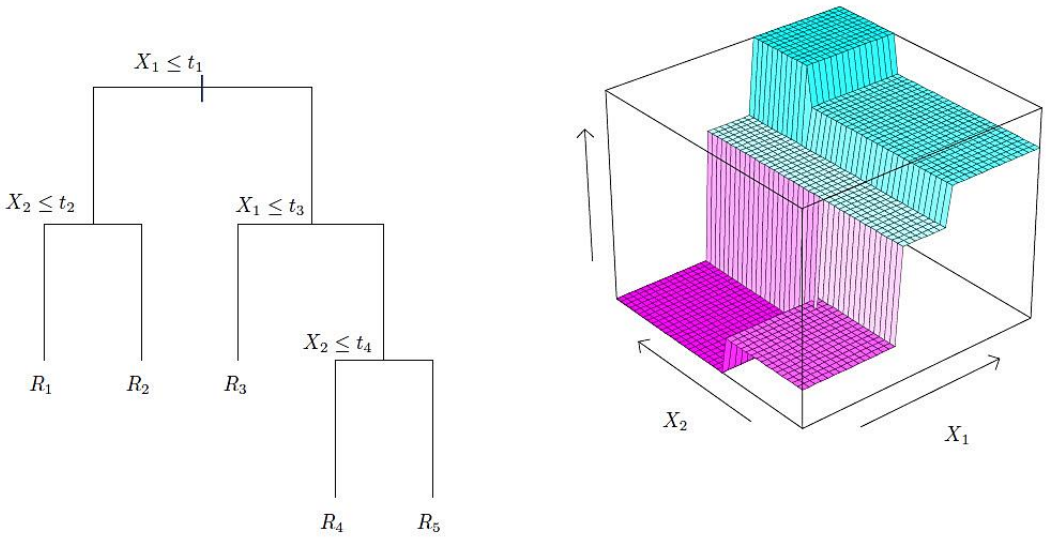


Figure 2. Space segmentation into regions and 3D regression surface in regression tree [4].

With the above procedure of forming a regression tree model, there is a possibility that the formed regression tree has good performance on the training set but poor generalization on the test data set. The Bootstrap aggregation - Bagging method allows the mentioned problem to be solved.

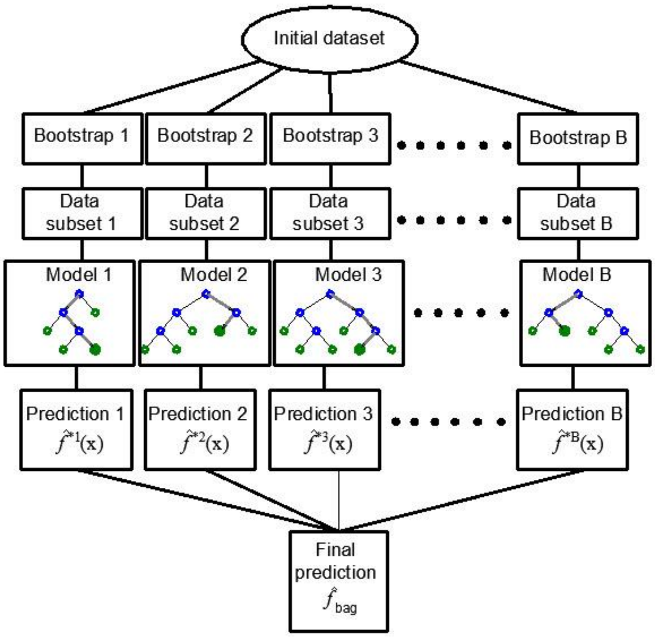

Initial dataset
Bootstrap 1
Bootstrap 2
Bootstrap 3
Bootstrap B
Data subset 1
Data subset 2
Data subset 3
Data subset B
Model 1
Model 2
Model 3
Model B
Prediction 1 $\hat{f}^{*1}(x)$
Prediction 2 $\hat{f}^{*2}(x)$
Prediction 3 $\hat{f}^{*3}(x)$
Prediction B $\hat{f}^{*B}(x)$
Final prediction $\hat{f}_{bag}$

Figure 3. Boostrap agreggation – Bagging in regression tree ensembles [3].

In order to practically carry out the mentioned procedure, it is necessary to have more training sets to reduce the variance by averaging. The mentioned problem of generating a more significant number of training sets can be overcome by the bootstrap method of sampling, i.e., by repeating sampling within the same training data set. The bagging method applies sampling with replacement [4]. If the model trained on the b-th bootstrap training set has the prediction function $\hat{f}^{*b}(x)$ at the point x, then by averaging all B models, it can be obtained a model (Figure 3.) whose predictive function will be determined by the following expression:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x). \tag{4}$$

2.2.2 Random Forests

The Random Forests method differs from the Bagging method in that it does not use all the variables in generating the model. In the process of generating the ensemble, the method tries to form regression trees that are decorrelated, which leads to reduced variance when the ensemble results are averaged for all the trees within the ensemble [4].

Suppose that training dataset D is composed of l observations and n features. First, a sample from training dataset is taken randomly with replacement and bootstrap is created. Before each split, $m \leq n$ features are randomly selected as candidates for splitting. Typical values for m are approximately $\mathrm{m}/3$ [4,7]. The Random Forest model is obtained by aggregating individual tree models obtained in this way.

2.2.3 Boosting trees

The Boosting method uses sequential model training, where each new regression tree added to the ensemble has the function of improving the performance of the previous tree collection. In this part of the paper, the application of the Gradient Boosting method in regression trees will be discussed [4,8-10].

In the case of a quadratic error function (Figure 4), a new submodel is added to the basic model in each subsequent step, which best estimates the residuals of the previous model. In this way, by adding a model through the application of an iterative procedure, a definite model is obtained, which represents an ensemble of previously obtained models.

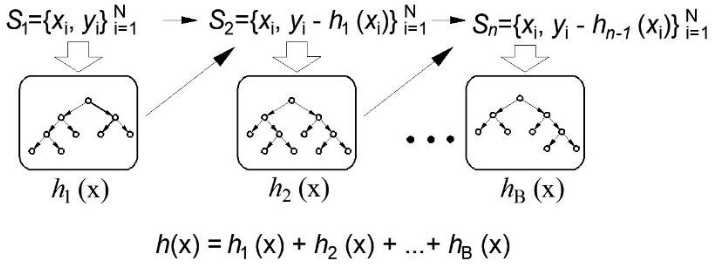

$S_1=\{x_i, y_i\}_{i=1}^{N}$
$S_2=\{x_i, y_i - h_1(x_i)\}_{i=1}^{N}$
$S_n=\{x_i, y_i - h_{n-1}(x_i)\}_{i=1}^{N}$
h_1 (x)
h_2 (x)
h_B (x)
$h(x) = h_1(x) + h_2(x) + \ldots + h_B(x)$

Figure 4. Gradient boosting in regression tre ensembles [3].

Estimate the relative influence of the predictor variable in this method is based on the number of times a variable is selected for splitting, weighted by the squared improvement to the model as a result of each split, and averaged over all trees [11].

2.3 Support vector regression (SVR)

Suppose a training dataset $\{(x_1,y_1),(x_2,y_2),\ldots,(x_l,y_l)\} \in \mathbb{R}^n \times \mathbb{R}$ is given, where $x_i \in \mathbb{R}^n$ is the n-dimenzional vector denoting model's inputs and y_i are the observed responses to these inputs.

The approximation function has the following form:

$$f(x) = \sum_{i=1}^{l} ({\alpha_i}^* - \alpha_i)\,\mathbf{K}(\mathbf{x_i},\mathbf{x}) + b. \tag{5}$$

In equation (5) $\mathbf{K}$ denotes the kernel function, α_i, ${\alpha_i}^*$ and b are the parameters obtained by minimizing the error function.

In order for SVR regression to be applied, the empirical risk function is introduced:

$$R_{emp}^{\varepsilon}(\mathbf{w},b) = \frac{1}{l}\sum_{i=1}^{l} |y_i - f(\mathbf{x_i},\mathbf{w})|_{\varepsilon.} \tag{6}$$

With the SVR algorithm, the goal is to minimize the empirical risk R_{emp}^{ε} as well as the $\|w\|^2$ value simultaneously. The so-called Vapnik's linear loss function (Figure 5) with ε-insensitivity zone is introduced, defined by the following expression [12,13]:

$$|y - f(\mathbf{x},\mathbf{w})|_{\varepsilon} = \begin{cases} 0 \quad if \quad |y - f(\mathbf{x},\mathbf{w})| \le \varepsilon \\ |y - f(\mathbf{x},\mathbf{w})| - \varepsilon \quad otherwise. \end{cases} \tag{7}$$

Taking into account the above, the problem can be reduced to minimizing the following function:

$$R = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} |y_i - f(\mathbf{x_i},\mathbf{w})|_{\varepsilon.} \tag{8}$$

The constant C has the role of balancing between the approximation error and the norm of the weight vector $\|w\|$. Minimizing R is equivalent to minimizing:

$$R_{w,\xi,\xi^*} = \frac{1}{2}\left[\|\mathbf{w}\|^2 + C\left(\sum_{i=1}^{l}\xi + \sum_{i=1}^{l}\xi_*\right)\right], \tag{9}$$

where ξ and $\xi^*$are the slack variables, which are shown in Figure 5.

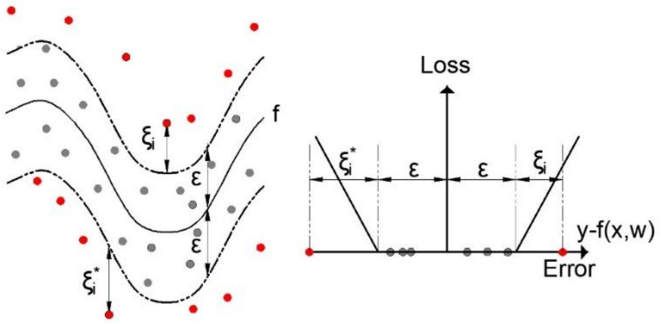


Figure 5. Nonlinear SVR with ε-insensitivity zone

Linear, RBF and sigmoid kernels used in this paper are defined as [14]:

$$\mathbf{K}\langle x_i, x\rangle = \langle x_i, x\rangle \tag{10}$$

$$\mathbf{K}\langle x_i, x\rangle = \exp(-\gamma\|x_i - x\|^2)\,, \gamma > 0 \tag{11}$$

$$\mathbf{K}\langle x_i, x\rangle = \tanh(\gamma\langle x_i, x\rangle + r)\,, \gamma > 0 \tag{12}$$

In this paper, LIBSVM software with SMO optimization algorithm was used [15,16]. The LIBSVM software was used within the MATLAB program [16].

2.4 Gaussian proces regression

A Gaussian processes model is a probability distribution over possible functions that fit a set of points. Consider a problem of nonlinear regression:

$$y = f(x) + \varepsilon, \quad \varepsilon \sim N(0, \sigma^2). \tag{13}$$

Where the function $f(\cdot) : R^n \to R$ is unknown and needs to be estimated, y_i is target variable, x are input variables and ε is normaly distributed additive noise. Gaussian process regression [17] assumed that $f(\cdot)$ follow Gaussian process with mean function $\mu(\cdot)$ and covariance function $k(\cdot\,,\,\cdot)$. The n observations in an arbitrary data set $\mathbf{y} = \{y_1, \ldots, y_n\}$, can always be imaginated as a sample from some multivariate (n variate) Gaussian distribution:

$$(y_1, \ldots, y_n)^T \sim N(\boldsymbol{\mu}, K), \tag{14}$$

where $\boldsymbol{\mu} = (\mu(x_1), \ldots, \mu(x_n))^T$ is the mean vector, and K is $n \times n$ covariance matrix of which the (i,j)th element $K_{ij} = k(x_i, x_j) + \sigma^2\delta_{ij}$. Here δ_{ij} is Kronecker delta function. Let x^* be any test point and y^* be corresponding response value. The joint distribution of $(y_1, \ldots, y_n, y^*)$ is an (n+1) variate normal distribution $(y_1, \ldots, y_n, y^*) \sim N(\boldsymbol{\mu}^*, \sum)$, where $\boldsymbol{\mu}^* = (\mu(x_1), \ldots, \mu(x_n),\ \mu(x^*))^T$ and covariance matrix:

$$\sum = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1n} & K_{1*} \\ K_{21} & K_{22} & \cdots & K_{2n} & K_{2*} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ K_{n1} & K_{n2} & \cdots & K_{nn} & K_{n*} \\ K_{*1} & K_{*2} & \cdots & K_{*n} & K_{**} \end{bmatrix} = \begin{bmatrix} K & K^* \\ K^{*T} & K^{**} \end{bmatrix} \tag{15}$$

where $K^* = (K(x^*, x_1), \ldots, K(x^*, x_n))^T$ and $K^{**} = K(x^*, x^*)$.
The conditional distribution of y^*, given $\mathbf{y} = (y_1, \ldots, y_n)^T$ is then $N(\hat{\boldsymbol{y}}^*, \hat{\boldsymbol{\sigma}}^{*\mathbf{2}})$ with

$$\hat{\boldsymbol{y}}^* = \mu(x^*) + K^{*T} K^{-1}(y - \mu), \tag{16}$$

$$\hat{\boldsymbol{\sigma}}^{*\mathbf{2}} = K^{**} + \sigma^2 - K^{*T} K^{-1} K^*. \tag{17}$$

For some covariance functions hyperparameters can be used to determine which inputs (variables) are more relevant than the others, using the automatic relevance determination (ARD). For example, consider squared exponential covariance function with different length scale parameters for each input (ARD SE):

$$k\left(x_p, x_q\right) = v^2 exp\left[-\frac{1}{2}\sum_{i=1}^{n}\left(\frac{x_p^i - x_q^i}{r_i}\right)^2\right]. \quad (18)$$

where r_i denotes the length scale of the covariance function along the input dimension i. If r_i is very large relative importance of the i-the input is smaller [17]. The hyperparameters $\{v, r_1, ..., r_n\}$ and the noise variance σ^2 can be estimated by the maximum likelihood method. The log-likelihood of the training data is given by:

$$L(v, r_1, ..., r_n, \sigma^2) = -\frac{1}{2}\log\det K - \frac{1}{2}\,\mathrm{y}^T K^{-1}\mathrm{y} - \frac{n}{2}\log 2\pi. \quad (19)$$

3. Evaluation and performance measures

The root mean square error (RMSE), mean absolute error (MAE), Pearson's Linear Correlation Coefficient (R) and mean absolute percentage error (MAPE) were used to assess the quality of the model.

The RMSE criterion for evaluating the accuracy of a model is a measure of the general accuracy of the model, and is expressed in the same units as the quantity to be modeled:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(d_k - o_k)^2}, \quad (20)$$

where is:
d_k- actual value (target value),
o_k - output or forecast given by the model,
N - number of training samples.

The MAE criterion is a measure of the absolute accuracy of the model and is used to represent the mean absolute error of the model:

$$MAE = \frac{1}{N}\sum_{k=1}^{N}|d_k - o_k|. \quad (21)$$

Pearson's linear correlation coefficient R represents a relative criterion for evaluating the accuracy of the model:

$$R = \sqrt{\left[\sum_{k=1}^{N}(d_k - \bar{d})(o_k - \bar{o})\right]^2 \cdot \left[\sum_{k=1}^{N}\left(d_k - \bar{d}\right)^2 (o_k - \bar{o})^2\right]^{-1}}. \quad (22)$$

where $\bar{o}$ represents the mean value of the prediction obtained by the corresponding model, and $\bar{d}$ represents the mean target value. Correlation coefficient values above 0.75 indicate a good correlation between the variables [18].

The mean absolute percentage error (MAPE) defined by equation (23), represents a relative criterion for evaluating the accuracy of the model,

$$MAPE = \frac{100}{N}\sum_{k=1}^{N}\left|\frac{d_k - o_k}{d_k}\right|. \quad (23)$$

The paper uses the procedure with ten-fold cross-validation of the model.

3. Dataset

A systematic search for papers examining the properties of SCRC in fresh and solid state was performed by April 2020.

Based on the published papers regarding the modelling of compressive strength of self-compacting rubberized concrete, input parameters were selected and all data that were incomplete, or that excluded any of the selected input parameters, were removed from the database.

With the aim of improving the mechanical properties of SCRC, various supplementary cementing material, such as slag, silica fume or/and fly ash, that are used in concrete mixtures, have also been reported in the literature. Therefore, the SCRC mixtures containing supplementary cementing material, such as slag, silica fume and fly ash are include in the database.The data collected through the search of papers contains the results of 166 SCRC samples (Table 2).

Emiroglu et al. (2012) provided experiments with the aim of investigation of bonding performances of crumb rubber and reinforced bars in SCRC. The authors prepared 4 different R-SCC mixtures with the replacement of crumb rubber in percentages of 15 %, 30 %, 45 % and 60 %. In the database only mixture with 15 % of crumb rubber replacement was included.

In order to investigate durability properties, Yung et al. (2013) replaced part of the fine aggregate with waste tire rubber powder in volume ratios of 5, 10, 15 and 20 %. They concluded that the best level of replacement achieved was with the addition of 5 % waste tire rubber powder (that had been passed through a #50 sieve)

The results obtained by Li et al. (2014) indicated that an SCC with adequate workability can be successfully created with partial replacement of sand or coarse aggregate with rubber particles of the same volume. When the replacement rate of sand with rubber particles was 30 %, the value of loss of compressive strength of SCC was about 30 %. Therefore, this mixture with the rubber replacing sand of 30 % wasn't taken into account.

Khalil et al. (2015) prepared SCC specimens with different ratios of crumb rubber (10, 20, 30 and 40 % of volume replacement of sand), but the last two mixtures (with 30 and 40 % of sand replacement) weren't added in the database. In order to research the effect of the added rubber particles on concrete behaviour, standard compressive, splitting and flexural strength tests were provided.

In another paper, Ismail and Hassan (2016b) investigated the mechanical properties and impact resistance of SCRC mixtures in which steel fibers were added in order to reinforce SCRC. However, all mixtures with steel fibers were removed at the base. The results showed that the addition of crumb rubber to concrete improved impact energy absorption and ductility, while the mechanical properties decreased with the increasing of content of crumb rubber.

Yu (2016) conducted a study on the effect of changing regularity of waste rubber on deformation performance of SCRC. The results showed that the rubber particles in a more uniform distribution reduced the maximum compressive strength.

Zaoiai et al. (2016) compared the rheological and mechanical performance between different mixtures formulations in order to obtain the optimum dose for rubber particles. The results of experimental testing showed that the compressive strength of SCC slightly decreased by replacing natural aggregate with rubber granulates, while the decrease in free shrinkage was proportional to the percentage of rubber granulates incorporated in the concrete composition.

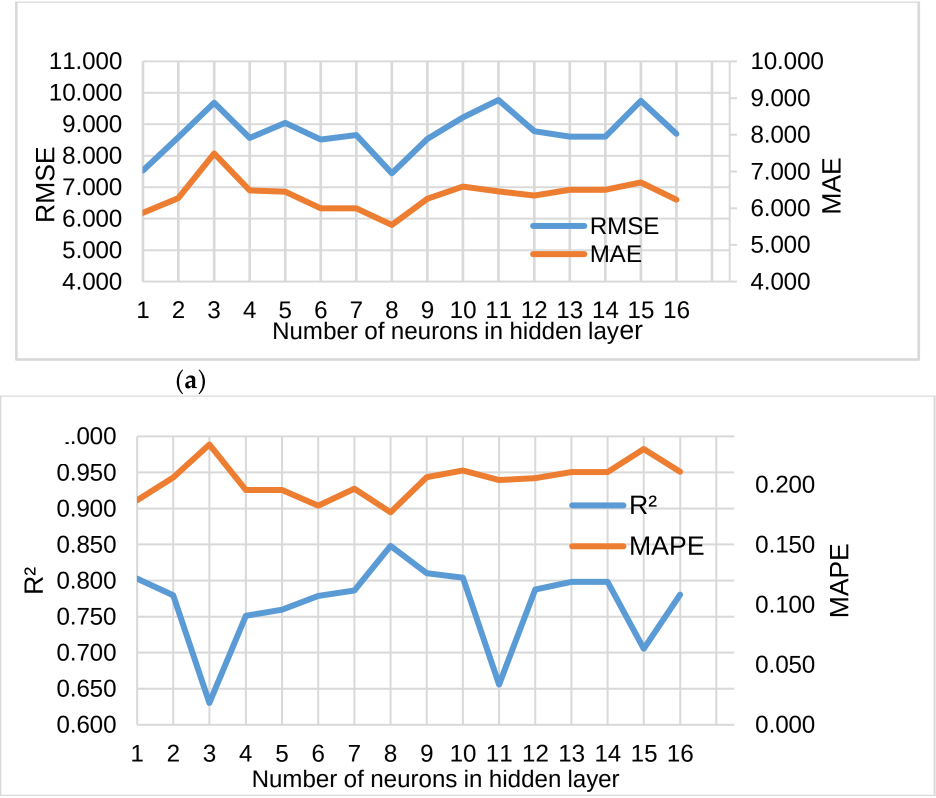
Aslani et al. (2018) designed 12 different mixtures of SCRC with 3 different rubber aggregate sizes and the replacement of rubber granules of 10, 20, 30 and 40 %. In order to optimise the fresh and hardened properties of SCRC, the optimal mix design, optimum content of crumb rubber particles and optimum size of crumb rubber particles was investigated and presented in the paper.

Hamza et al. (2018) examined the effect of 0-2 mm fine rubber as a partial substitute of sand volume in different percentages (5, 10, 15, 20 and 30 %) on the fresh and hardened properties of SCRC. Experimental results showed that the addition of fine rubber to the SCC reduced the workability, reduced its passing capacity and increased the possibility of blocking. Also, with increasing of crumb particle content, a decrease in compressive strength was observed.

4. Discussion

MLP-ANN with one hidden layer was trained using Levenberg-Marquardt algorithm [19]. The criterion to stop the training was either the maximum number of epochs (set to 1000), the minimum gradient magnitude (set to 10^{-5}) or the network performance (measured as the mean square error and set to 0). All input data are normalized in the range [-1,1] prior to training. The number of neurons in the input layer is determined by the number of input variables, i.e. it consists of 10 neurons, while there is only one neuron in the output layer. The maximum number of neurons in the hidden layer was determined experimentally using Eq. 1 and 2 and equals 16.

Figure 6(a) shows obtained performance using RMSE and MAE as absolute measures, while Figure 6(b) presents results using R and MAPE as relative measures. By analyzing models with different numbers of neurons in the hidden layer, it was obtained that the configuration that has 8 neurons in the hidden layer is optimal in terms of all four criteria.



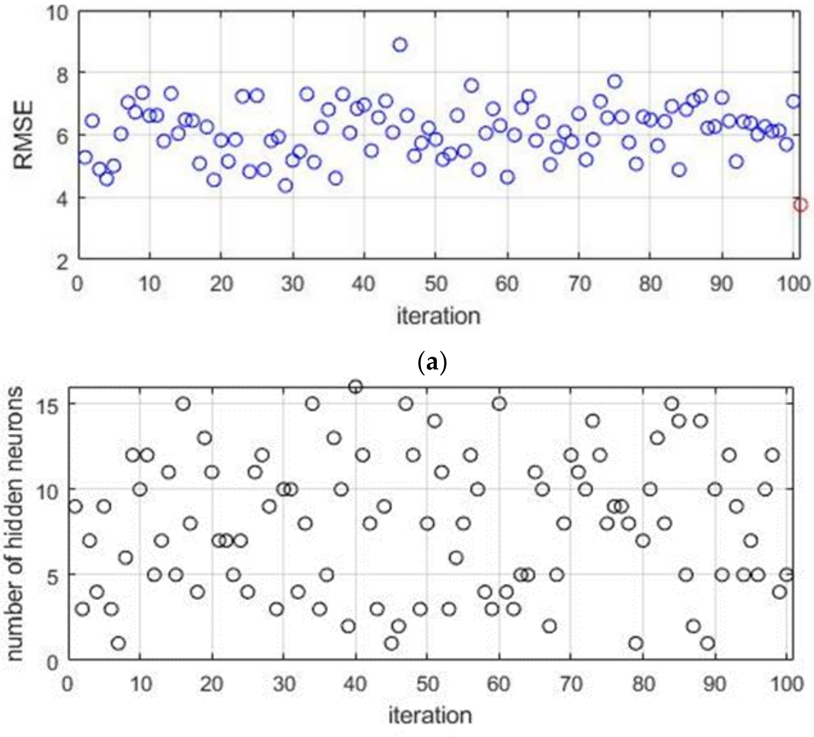
(**a**)

(**b**)

Figure 1. Comparison of performance measures using MLP-ANNs with different configurations: (**a**) RMSE and MAE, (**b**) R and MAPE

In order to improve the generalization of the model ensemble models are created. The use of base models of neural networks that have up to 16 neurons in their hidden layer was analyzed, where each of the base models in the ensemble can have a different number of neurons in the hidden layer. The optimal base model in the current iteration is defined based on the minimum RMSE value of the 16 generated models in the current iteration. After that, the procedure continues until the total number of 100 base models of the ensemble are generated.

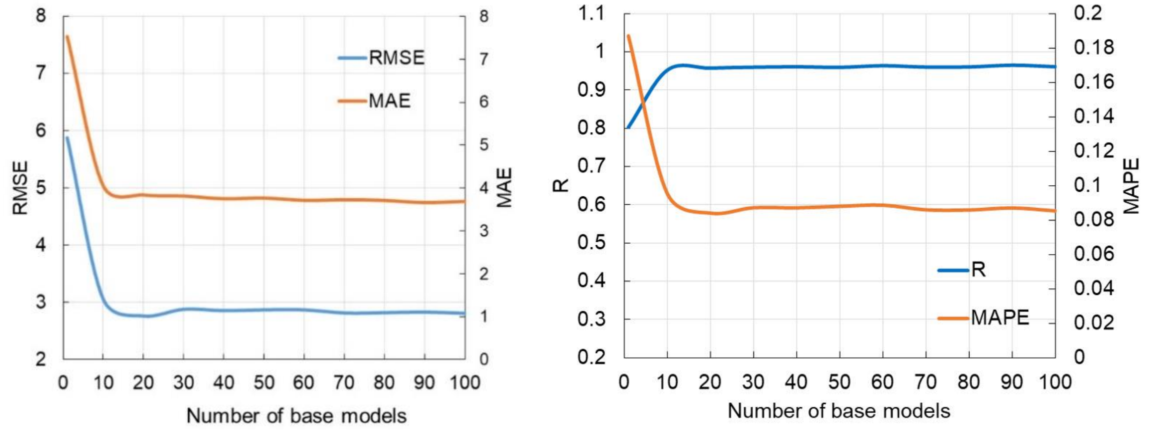
The bootstrap method formed a sample of the same size as the original sample. Since the evaluation used a ten-fold cross-validation procedure, sampling was performed within nine folds. The remaining fold was used to test the ensemble. The procedure was repeated ten times so that the whole set of data was used for testing the ensemble, and the evaluation of the prediction of the ensemble was represented by the mean value of all base models in terms of the considered model performance.



(**a**)

(b)

Figure 7. (**a**) RMSE value for each of the iterations and the corresponding architecture (**b**) The optimal number of neurons in the hidden layer in each iteration



(**a**) (**b**)

Figure 8. Comparison of performance measures using ensembles of MLP-ANNs with different number of base models: (**a**) RMSE and MAE, (**b**) R and MAPE

In Figure 7 (a), the RMSE value for the base models is marked with a blue circle, while the RMSE for the ensemble is marked with a red circle. It can be seen in Figure 8 and Figure 9 that the model of an ensemble composed of individual neural networks contributes to a significant improvement in generalization. Comparative values in terms of defined criteria for model evaluation for the optimal individual neural network model and ensemble model are shown in Table 2, and the regression plot of the modelled and target values for the ensemble model is shown in Figure 10.

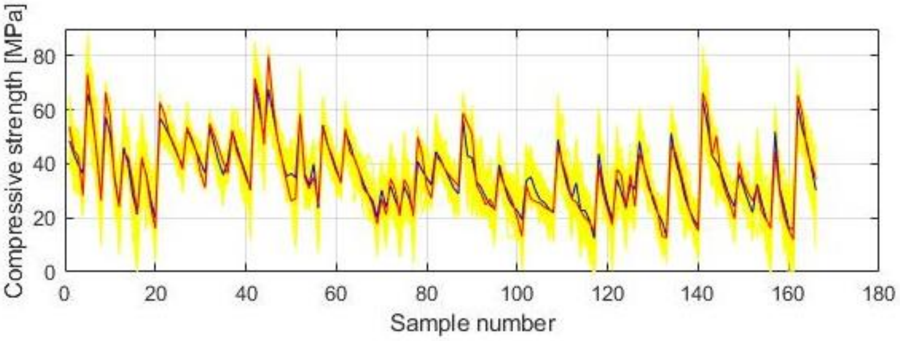


Figure 9. Prediction of individual neural networks (yellow color), ensemble model prediction (dark blue color), target values (red color) of compressive strength.

Table 2. Comparison of the optimal individual neural network and ensemble model

Model	RMSE	MAE	MAPE/100	R
NN-10-8-1	7.4424	5.5434	0.1768	0.8481
Ensemble	3.6888	2.8099	0.0854	0.9610

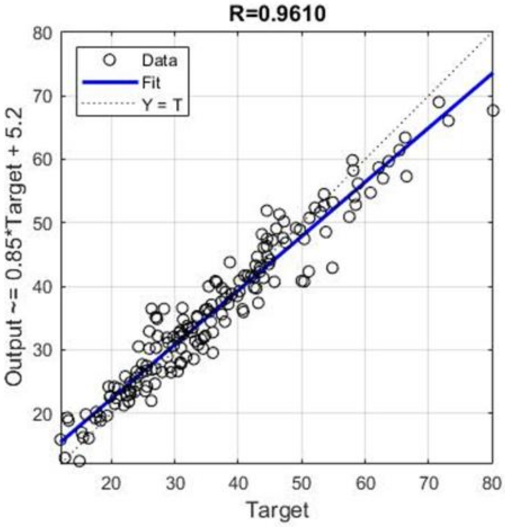


Figure 10. Regression plot for modelled and targeted values for optimal ensemble model

The application of models based on decision trees was also analyzed. MSE values were used as a criterion in model training, and numerical values of MSE of the base models of the generated ensemble were presented cumulatively. The parameters of the optimal models were determined by the grid-search method.

The analysis was performed using the following methods:

1. Bagging method (TreeBagger),
2. Random Forests method,
3. Boosted Trees method.

The data used for training (in bag data) in the TreeBagger model are extracted from the entire data set by sampling with replacement. Data that are not extracted from the whole set (out of bag data) are test data. During the model building process, an out-of-bag error is calculated by finding the difference between the out-of-bag sample and the prediction for that same sample, and it is stored. The procedure is repeated for all trees within the ensemble.

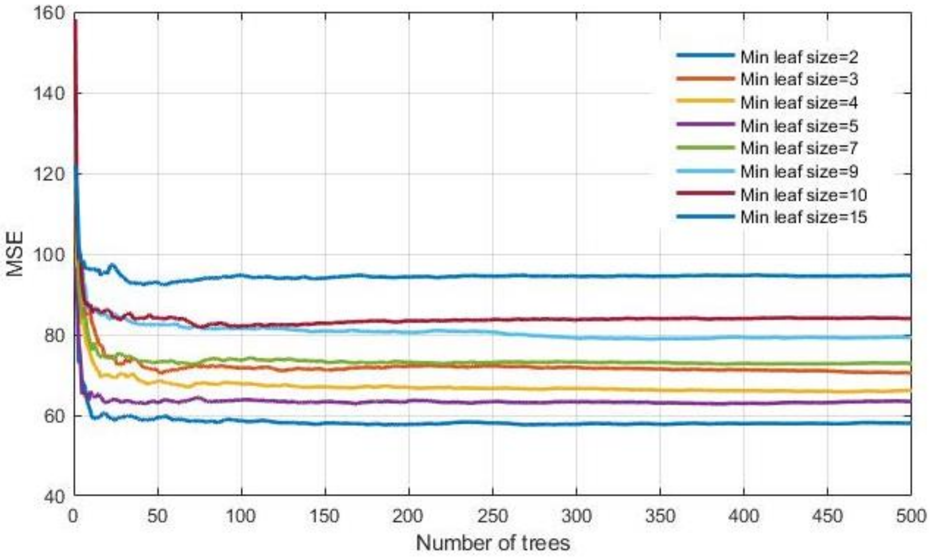


Figure 11. MSE vs. number of trees in the ensemble for different minimum leaf sizes using regression tree ensembles realized with bootstrap aggregation (bagging)

During the implementation of the Bagging method, different values of model parameters were analyzed, as follows:

1. Number of generated trees B. Within this analysis, the maximum number of generated trees is limited to 500.

2. The minimum number of data or samples assigned to the leaf (min leaf size) within the tree. Values from 2 to 15 samples with a step size of 1 per tree leaf were considered.

The lowest value of MSE (Figure 11) of the analyzed models, has a model that has a minimum number of data per tree leaf 2, marked in darker blue. The saturation of the learning curve occurs after 269 trees in the ensemble.

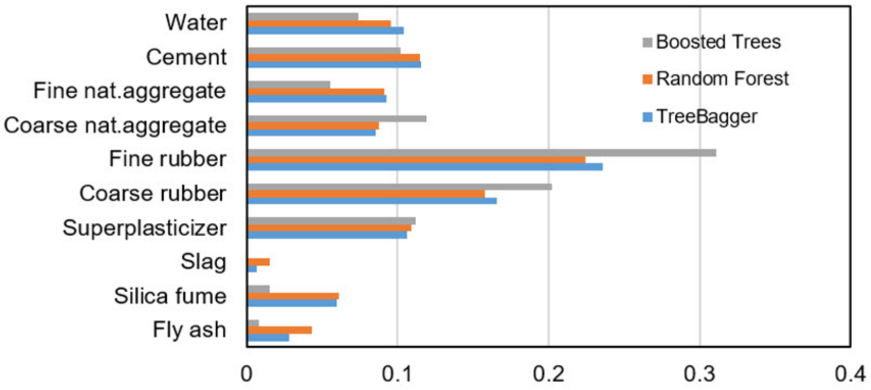


Figure 12. Significance of individual variables in the ensemble model when applying the Bagging method

In order to determine the significance of the j-th variable, it is necessary that after training the model, the values of the j-th variable be permuted within the training data and that the out of bag error for such permutated data be recalculated. The significance of the variable (Figure 12) is determined by calculating the mean value of the difference before and after permutation for all trees within the ensemble. This value is then divided by the standard deviation of these differences. The variable for which a higher value was obtained in relation to the others is ranked as more significant in relation to the variables in which smaller values were obtained [5].

During the implementation of the RF method, different values of the adaptive parameters of the model were analyzed, as follows:

1. Number of generated trees B. Within this analysis, the maximum number of generated trees is limited to 500.

2. The number of variables that are used for splitting in the tree. In the paper Random Forests by L. Bryman [5], it is recommended that the subset m of variables on which splitting is performed is p/3 of the predictor. In this paper, the values of m from 2 to 9 (Figure 13) are examined.

3. The minimum number of data or samples assigned to a leaf (min leaf size) within a tree. Values from 2 to 10 samples per tree leaf were considered.

The application of a narrowed set of splitting variables in this case did not yield results. The Figure 13 shows the values of the accuracy criteria of the ensemble of 500 basic models, from which one can see the tendency to increase the number of variables on which the splitting is performed to increase the accuracy of the model in terms of all defined criteria. The analysis showed greater accuracy in tree models in which the number of data per leaf is equal to 2. The relevance of individual variables for RF model is shown in Figure 12.

With the Boosting Trees method, there are the following model parameters:

1. Number of generated trees B. With the Gradient Boosting method, there is a possibility of overtraining the model when forming too many trees. Due to the large number of analyzed models in the research, the number of base models within the ensemble is limited to a maximum of 100.
2. Learning rate λ . This parameter determines the training speed of the model. The paper investigates a number of values, as follows: 0.001; 0.01; 0.1; 0.25; 0.5; 0.75 and 1.0.
3. Number of splits in the tree d. Models of trees with a maximum number of splits of $2^0 = 1, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7 = 128$ were generated.

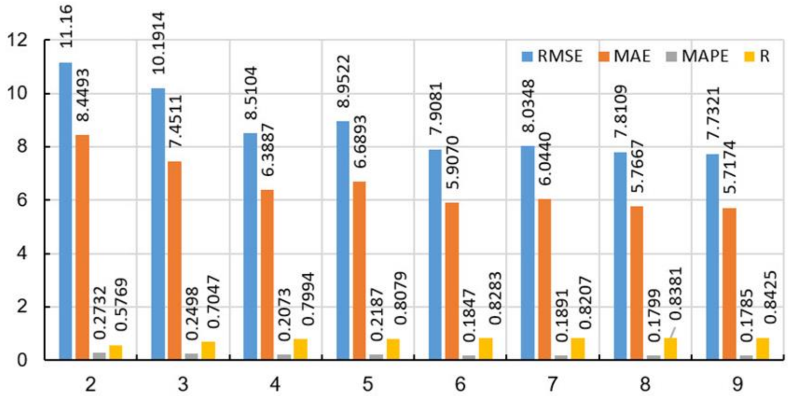


Figure 13. Influence of the number of variables on which is performed splitting on the accuracy of the model in the RF method

It was obtained that the optimal model (marked in yellow in the Figure 14) has 100 generated trees, the value of the reduction parameter is 0.10, and the maximum number of splits in the model is 64. As in other tree based models, the relevance of individual model variables is determined and shown in Figure 12.

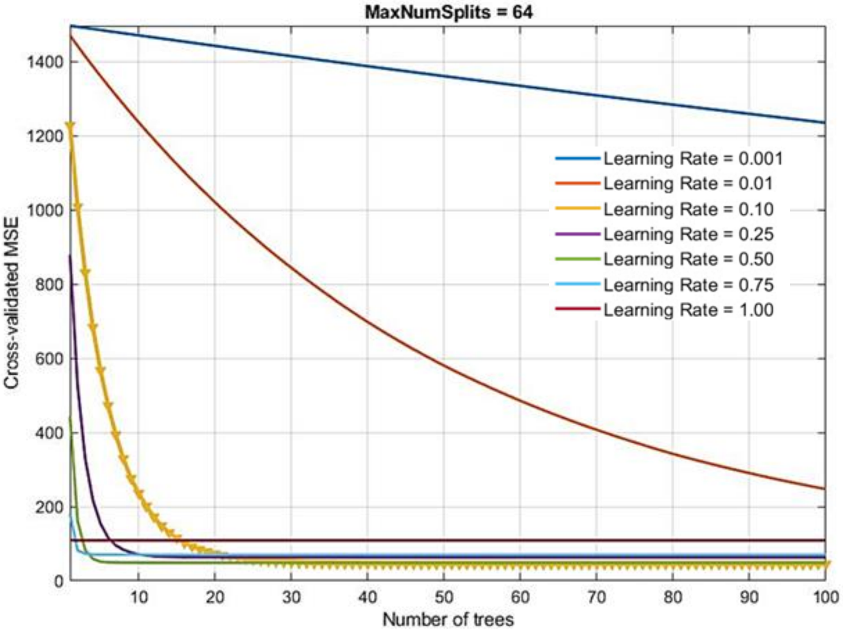


Figure 14. Dependence of MSE values on learning rate and number of base models in Boosted Trees method

Table 3. Comparative analysis of results in Bagging, R.Forests and Boosted Trees methods

Metoda	RMSE	MAE	MAPE/100	R
TreeBager	8.1890	6.0546	0.1881	0.8214
R.Forests	7.7321	5.7174	0.1785	0.8425
Boosted Trees	**7.4821**	**5.4248**	**0.1573**	**0.8432**

In order to obtain a good regression model using the support vector method, it is necessary to select the appropriate kernel function. For selected kernel functions, it is necessary to determine their parameters, as well as the value of penalty parameter C (Figure 15).

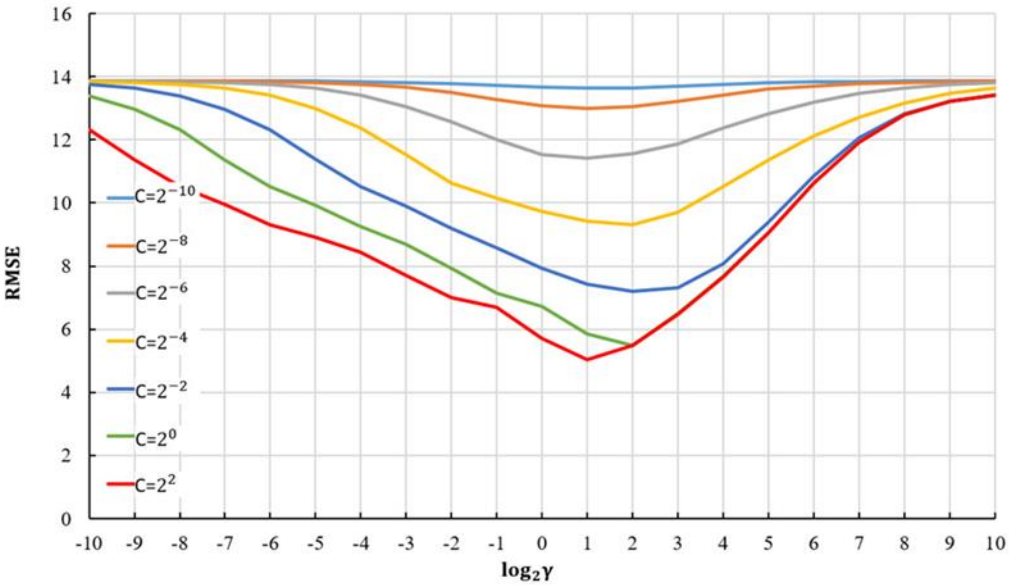


Figure 15. RMSE vs. hyperparameters C and γ for $\varepsilon = 2^{-6}$ using SVR with RBF kernel in rough search

In this paper, therefore, the use of several different kernel functions is investigated in order to find the best one. The use of SVR models with linear, RBF, and sigmoid kernel was analyzed. Normalization, by which all input data were transformed into the range [0,1], was done before training and testing the model. The optimal model was determined using the grid search algorithm for all kernels (C = 4.22195 and ε = 0.105765 for the linear kernel; C = 7.67645; ε = 0.0230915; γ = 1.89915 for the RBF kernel; C = 1113.70875; ε = 0.0920525; γ = 0.000945345 for sigmoid kernel).

Comparative analysis of different SVR models shows that the models have different accuracy depending on the adopted criteria depending on the kernel function. Models with linear and sigmoid kernels have similar accuracy according to different criteria. The model with the RBF kernel function (Table 4) has significantly higher accuracy with respect to all criterion functions.

Table 4. Comparative analysis of results using linear, RBF and sigmoid kernel in SVR method

Model	RMSE	MAE	MAPE/100	R
Lin. kernel	8.7154	6.6468	0.2105	0.7751
RBF kernel	**4.9646**	**3.5352**	**0.1171**	**0.9332**
Sig. kernel	8.7104	6.6094	0.2073	0.7718

During the development of the Gaussian process model, covariance functions that have one length scale parameter for all input variables (exponential, square-exponential, matern 3/2, matern 5/2) and rational quadratic covariance function as well as their equivalent ARD covariance functions that have a separate length scale for each input variable. Standardization procedure was performed using Z-score, i.e., the data are transformed to have a mean value of zero and a variance equal to one. Models with constant base functions were analyzed.

Table 5. Comparative analysis of results of GPR with various covariance functions

GP model covariance function	RMSE	MAE	MAPE/100	R
Exponential	5.0574	3.5064	0.1038	0.9316
ARD-Exponential	4.6120	3.1634	**0.0947**	0.9427
Squared Exponential	5.0447	3.4686	0.1133	0.9300
ARD-Squared Exponential	4.9670	3.4076	0.1101	0.9334
Matern 3/2	4.7244	3.2487	0.1021	0.9386
ARD-Matern 3/2	**4.4341**	**3.1022**	0.0958	**0.9474**
Matern 5/2	4.8275	3.3133	0.1061	0.9360
ARD-Matern 5/2	4.6527	3.2691	0.1037	0.9424
Rational Quadratic	4.6467	3.2022	0.0997	09407
ARD Rational quadratic	4.5937	3.1894	0.1006	0.9435

Parameter values are determined by maximizing the log marginal probability. By using ARD covariance functions, it is possible to see the relevance of individual variables or predictors in the model (Automatic Relevance Determination - ARD). Higher parameter values for ARD covariance functions indicate less relevance of a particular variable to which they refer.

According to the three defined criteria, the RMSE, MAE and R model with ARD matern 3/2 covariance function (Table 5.) can be considered optimal, while according to the MAPE criterion it is second in accuracy with a difference of 0.0011 compared to the first ranked model according to that criterion .

The analysis of the relevance of the variable models will be performed based on the parameters of the covariance function on the model with the ARD matern 3/2 function as the most accurate model. The values of the distance scale parameters are shown in the logarithmic scale (logarithms with base 10) in Figure 15.

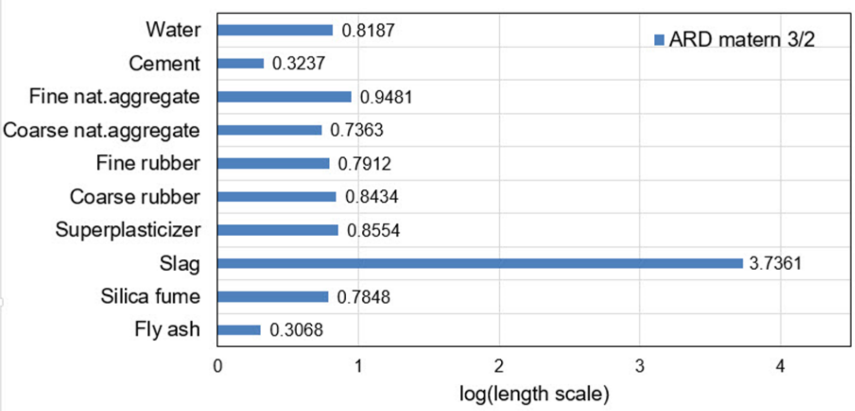


Figure 16. Variable selection using ARD matern 3/2 covariance function

It can be seen (Figure 16) that in the optimal model with ARD matern 3/2 function the input variables 2 (Cement) and 10 (Fly ash) have the greatest relevance and the greatest impact on the model's accuracy. Variables 1 (Water), 3 (Fine natural aggregate), 4 (Coarse natural aggregate), 5 (Fine rubber), 6 (Coarse rubber), 7 (Superplasicizer) and 9 (Silica fume) have similar relevance. Input variable 8 (Slag) has the least relevance and the most negligible impact on the model's accuracy.

In further analysis, the models that used the most relevant variables were considered due to the possibility that the presence of irrelevant variables may reduce the accuracy of the model. By using a narrowed set of variables in certain cases, it is possible to obtain a model of the same or higher accuracy. Also, in this way, the complexity of the model is reduced, and the model training process is accelerated.

In the further analysis, a comparison was made (Table 6) of the two models, as follows:

1. Model where variable 8 is excluded as less relevant (Slag),
2. A model that includes all variables.

Table 6. Comparative analysis of GPR models with different sets of input variables

Model	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	RMSE	MAE	MAPE/100	R
1.	1	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**1**	4.3934	3.0583	0.0942	0.9482
2.	1	1	1	1	1	1	1	1	1	1	4.4341	3.1022	0.0958	0.9474

The application of individual models of neural networks gave models with unsatisfactory accuracy in terms of all criteria. For this reason, the use of neural network ensembles was considered. A limit is defined in terms of the maximum number of hidden layer neurons, while the number of inputs and outputs defines the number of input and output layer neurons. The analysis showed that the application of ensembles gives models of significantly higher accuracy with the optimal individual model of the neural network with the architecture 10-8-1. The value of the ensemble correlation coefficient was increased to the value 0.9610, which is a satisfactory value. The values of RMSE, MAE, and MAPE are approximately halved in relation to the individual optimal neural network model. With an ensemble of 40 base models, there is convergence in terms of defined criteria, and a further increase in the complexity of the model does not lead to an increase in accuracy.

A comparative analysis of all tree-based models shows that the model obtained by applying Boosted Trees has the best accuracy, while models based on the Bagging and Random Forests methods showed lower and similar values in terms of all criteria. All models based on regression trees can determine the relevance of individual input variables in the model. As the least relevant input variable, all models identified a variable denoting Slag. In the event of a significant expansion of the database, the complexity of the method could be analyzed, whereby the models in the Bagging and Random Forests methods can be processed in parallel, which is not the case with the Boosted Trees method where models are trained sequentially within the ensemble. In addition, it should be pointed out that a significant number of basic models are needed to saturate the learning curve.

Comparative analysis of different SVR models shows that the models, depending on the kernel function, have different accuracy according to the adopted criteria. The use of the RBF kernel function, in this case, gave satisfactory results, while the use of linear and sigmoid kernels gave significantly worse results. Models with linear and sigmoid kernel function have almost twice worse values in terms of the three criteria RMSE, MAE, and MAPE. The value of the correlation coefficient R is significantly lower than the RBF model. Training models with the RBF function are relatively simple, but these models do not provide direct insight into the relevance of individual variables in the model.

GPR models that do not use different length scale parameters for individual input variables in the considered problem in most cases have worse criterion values than models that use different length scale parameters (ARD models). The best model is a model with ARD matern 3/2 covariance function. The values of the length scale parameters for the optimal model can be used to assess the relevance of the predictor or variable in the model. This model singles out Fly ash and Cement as the most significant variables. Slightly smaller but similar relevance are the variables representing coarse natural aggregate, Silica Fume, and Fine rubber. The variables Coarse rubber and Superplasticizer represent the next group of variables that have similar and lower relevance than the previous. The least relevant are the variables Fine natural aggregates and Slag.

In further analysis, GPR models that used the most relevant variables were considered due to the possibility that the presence of irrelevant variables may reduce the accuracy of the model accuracy. It has been shown that by eliminating the variable of smaller significance, which in this case represents the variable Slag, and by using a narrowed set of variables, a model of higher accuracy is obtained. Also, in this way, the complexity of the model is reduced, and the model training process is accelerated.

5. Conclusions

In order to make a decision for the use of appropriate SCRC concrete, it is necessary to have a reliable model for predicting compressive strength, which is one of its most important mechanical characteristics.

This paper gives a comprehensive overview of the state-of-the-art machine learning methods that can be used for estimating SCRC compressive strength, including MLP-ANN, ensembles of MLP-ANNs, regression tree ensembles (random forests, boosted and bagged regression trees), SVR with RBF kernel, and GPR with exponential, squared exponential, Matern, and rational quadratic covariance functions.

As a basis for the development of the forecast model, a database was obtained from an experimental study containing a total of 166 samples of SCRC. All models were trained and tested under the same conditions using ten-fold cross-validation.

An ensemble of neural networks and a GPR model with an ARD matern 3/2 covariance function stood out as the models of the highest accuracy.

The application of an ensemble of neural networks has in this case the greatest accuracy in terms of all criteria but also the greatest complexity, where satisfactory accuracy is achieved only with the formation of an ensemble with a significant number of basic models. A significant shortcoming of this model is the lack of transparency of the model and the impossibility of directly determining the relevance of variables.

The use of the GPR method gives models of satisfactory accuracy and, at the same time, significantly less complexity. The proposed model with ARD Matern 3/2 function enables the ranking of the influence of individual variables on the accuracy of the model. It has been shown that by eliminating variables of lesser importance and using a narrowed set of variables, a model of higher accuracy can be obtained. Also, in this way, the complexity of the model is reduced, and the model training process is accelerated.

This research indicates that in the planning phases, it is possible to quickly and accurately estimate the compressive strength of SCRC concrete using methods based on machine learning.

Based on the proposed models for the neural network ensemble and GPR models, a model application was developed in the MATLAB program and deposited on the Github website.

References

1. Beale, M. H.; Hagan, M. T.; Demuth, H. B. *Neural network toolbox*. The Mathworks, Inc: 2010.
2. Kovačević, M.; Ivanišević, N.; Dašić, T.; Marković, Lj. Application of artificial neural networks for hydrological modelling in Karst. *Gradjevinar*. **2018**. 70(*1)*, 1-10. [CrossRef]
3. Kovačević, M.; Ivanišević, N.; Petronijević, P.; Despotović, V. Construction cost estimation of reinforced and prestressed concrete bridges using machine learning. *Građevinar*. **2021**. *73 (1),* 1-13. [CrossRef]
4. Hastie, T.; Tibsirani, R.; Friedman, J. *The Elements of Statistical Learning*. Springer: 2009.
5. Breiman, L.; Friedman, H.; Olsen, R.; Stone, C.J. *Classification and Regression Trees.* Chapman and Hall/CRC: Wadsworth, 1984.
6. Breiman L.: Bagging Predictors, *Mach. Learn*, .1996. *24*(2), 123-140. [CrossRef]
7. Breiman, L. Random Forests. *Mach. Learn*. **2001**. 45, 5-32. [CrossRef]
8. Freund, Y.; Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput Syst. Sci*. **1997**. 55 (1), 119-139. [CrossRef] https://doi.org/10.1006/jcss.1997.1504
9. Elith, J.; Leathwick, J. R.; Hastie, T. A working guide to boosted regression trees. *J. Anim. Ecol*. **2008**. *77*(4), 802-813. [CrossRef] **https://doi.org/10.1111/j.1365-2656.2008.01390.x**
10. Friedman, J.H. Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **2001**. *29*(5), 1189– 1232.
11. Friedman, J.H.; Meulman, J.J. (2003) Multiple additive regression trees with application in epidemiology. *Stat. Med* , *22*, 1365– 1381. [CrossRef]
12. Vapnik, V. *The nature of statistical learning theory*. Springer: New York, 1995.
13. Kecman, V. *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. MIT Press: 2001
14. Smola, A. J.; Sholkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**. *14*, 199–222.
15. Chang, C. C.; Lin, C. J. LIBSVM : a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**. 2(3). [CrossRef]
16. LIBSVM - A Library for Support Vector Machines. Available online: https://www.csie.ntu.edu.tw/~cjlin/libsvm/.(accessed 21 February, 2021.).
17. Rasmussen, C. E.; Williams, C. K. *Gaussian Processes for Machine Learning*. The MIT Press: Cambridge, Massachusetts, 2006.
18. P. Matić: Kratkoročno predviđanje hidrološkog dotoka pomoću umjetne neuronske mreže, doctoral dissertation, University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, Split, Croatia, 2014.
19. Hagan, M. T.; Demuth, H. B.; Beale, M. H.; Jesus, O. De. *Neural network design*, Martin T. Hagan: Oklahoma State University Stillwater, Oklahoma, USA, 2014.