Computers and Concrete, *Vol. 25, No. 6 (2020) 565-574*
DOI: https://doi.org/10.12989/cac.2020.25.6.565

An assessment of machine learning models for slump flow and examining redundant features

Ramazan Ünlü*

Management and Information Systems, Gumushane University, 29000 Gümüşhane, Turkey

(*Received December 17, 2018, Revised May 21, 2020, Accepted May 24, 2020*)

Abstract. Over the years, several machine learning approaches have been proposed and utilized to create a prediction model for the high-performance concrete (HPC) slump flow. Despite HPC is a highly complex material, predicting its pattern is a rather ambitious process. Hence, choosing and applying the correct method remain a crucial task. Like some other problems, prediction of HPC slump flow suffers from abnormal attributes which might both have an influence on prediction accuracy and increases variance. In recent years, different studies are proposed to optimize the prediction accuracy for HPC slump flow. However, more state-of-the-art regression algorithms can be implemented to create a better model. This study focuses on several methods with different mathematical backgrounds to get the best possible results. Four well-known algorithms Support Vector Regression, M5P Trees, Random Forest, and MLPReg are implemented with optimum parameters as base learners. Also, redundant features are examined to better understand both how ingredients influence on prediction models and whether possible to achieve acceptable results with a few components. Based on the findings, the MLPReg algorithm with optimum parameters gives better results than others in terms of commonly used statistical error evaluation metrics. Besides, chosen algorithms can give rather accurate results using just a few attributes of a slump flow dataset.

Keywords: multilayer perceptron regression; regression trees; support vector regression; redundant features, M5P trees

1. Introduction

With the change of architectural traditions, due to its high flowability and workability advantage self-compacting concrete (SCC) technology gains importance day by day because of unusual architectural designs of concrete structures. Fresh cementitious materials, as many materials in industry or nature, behave as fluids with a yield stress, which is the minimum stress for irreversible deformation and flow to occur (Roussel 2006). Mechtcherine *et al.* (2014) suggest that building concrete structures efficiently and with high quality, the consistency of the fresh concrete should comply with the requirements posed by the structure's geometry. Moreover, with SCC many structural problems disappeared like an improper filling, segregation, mechanical vibration mistakes, etc.

SCC compacts with its weight this feature provide advantages for reducing construction time, labor cost, and noise on the construction site (Khatib 2008). In the slump-cone test, the slump can be deduced by measuring the drop from the top of the slumped fresh concrete, and the slump flow by measuring the diameter of it (Yeh 2007).The slump flow test aims to investigate the filling ability of SCC. It measures flow spread and flow time T50. Equipment for the slump flow test is based on plate and abrams cone. The base plate of size at least 900×900 mm, made of an impermeable and rigid material, and clearly marked with circles of Ø200 mm and Ø500 mm at the center. Abrams cone with the internal upper/lower diameter equal to 100/200 mm and the height of 300 mm (Schutter 2005). The calculation of slump flow shown in Eq. (1).

$$S=\frac{(d_{max}+d_{perp})}{2} \quad (1)$$

where $d_{\max}$ the largest diameter of the circular spread of the concrete and $d_{perp}$is the circular spread of the concrete at an angle approximately perpendicular to $d_{\max}$. In general, the mathematical models used to describe the material behavior of concrete and empirical formulas derived from experimental results (Chandwani *et al.* 2015). When the model includes a large amount of independent variables, unknown interactions additives, and chemical admixtures traditional prediction methods could not provide satisfactory results. In addition to these high costs and long lead times of experimental is un-solicited status. In non-linear systems, it's hard to reveal a mathematical model for accurate results quite the opposite that Machine learning methods are self-learning algorithms and they can learn and produce a model from past data so algorithms can reach accurate solutions rapidly.

Machine learning algorithms are commonly used for complex real-world regression and classification problems. In many different studies, it is proved that machine learning methods outperform traditional methods in terms of revealing hidden patterns from datasets. Therefore, this paper presents the prediction of slump flow test results of concrete via machine learning algorithms, and chosen methods are neither used nor compared in any previous

*Corresponding author, Ph.D.
E-mail: ramazanunlu@gumushane.edu.tr

ISSN: 1598-8198 (Print), 1598-818X (Online)

study.

2. Literature review

With the increasing concrete technology, self-consolidating concrete brought new concepts like fluidity, mobility, and compact ability and encourages architects and engineers for designing and building extraordinary structures. Due to this change in architectural design style slump-flow test of concrete placed in an important position. Besides the design methods (Ferrara *et al.* 2007, Okamura and Ouchi 2003, Okamura *et al.* 2000, Saak *et al.* 2001, Su *et al.* 2001), additive variations (Busari *et al.* 2018a, Massana *et al.* 2018, Pelisser *et al*. 2018, Şahmaran *et al.* 2006, Sari *et al.* 1999, Uysal and Sumer 2011, Uysal and Yilmaz 2011) and statistical applications (Busari *et al.* 2018b, González-Taboada *et al.* 2018, Habibi and Ghomashi 2018, Roussel 2006, Tregger *et al.* 2012), many researchers used artificial intelligence and machine learning methods in predicting the concrete strength and slump flow values.

Domone (1998) compared the slump flow test and flow test for measuring fluidity of concrete mixes and he found a correlation between the flow and slump flow values. Yeh (1999) proved that artificial neural networks (ANN) is more accurate than regression algorithms in his study. Dias and Pooliyadda (2001) applied the backpropagation neural networks method to predict the strength and slump of ready mixed concrete. Lee (2003) developed ANN-based I-PreConS (Intelligent PREdiction system of CONcrete Strength) that provides strength information of the concrete. Öztaş *et al.* (2006) developed ANN and regression models for the estimation of concrete slump. Yeh (2007) modeled slump flow of concrete by using second-order regressions and ANN and proved that ANN algorithm is suitable for concrete flow predictions and also Jain *et al.* (2008) and Siddique *et al.* (2011) presented an ANN model for prediction of compressive strength of self-compacting concrete (SCC) containing bottom ash. Cheng *et al.* (2012) proposed an Artificial Intelligence hybrid system to predict HPC compressive strength. Cao *et al.* (2013) established the effectiveness of the support vector machines (SVM) algorithm by presenting a successful prediction model for the elastic modulus of SCC. Chandwani *et al.* (2015) presented a hybrid model, they used ANN and Genetic Algorithms (GA) for predicting slump of Ready Mix Concrete (RMC). Aydogmus *et al.* (2015) used ensemble machine learning methods for predicting slump flow of high-performance concrete (HPC) they compared simple machine learning and ensemble machine learning methods and proved that ensemble machine learning algorithms are superior to simple machine learning algorithms. Sonebi *et al.* (2016) compared two kernel functions of SVM; Polynomial and Radial basis function (RBF) for predicting slump flow rate of SCC and they confirmed the SVM RBF model is highly precise for prediction concrete properties. Mashhadban *et al.* (2016) suggested particle swarm optimization (PSO) based ANN hybrid model for modeling mechanical properties in fiber-reinforced SCC.

Table 1 Descriptive statistics of the dataset

Name of attributes	Max	Min	Median	# Null value
Cement	227.745	137	204.95	0
Slag	79.023	0	101	0
Fly ash	146.459	0	163.5	0
Water	197.305	160	196	0
SP	8.549	4.4	8.15	0
Coarse Aggr.	884.158	708	878.5	0
Fine Aggr.	741.234	640.6	742.85	0

3. Methodology

In this study, we use publicly available concrete slump flow dataset proposed by Yeh (2007) for our experiment. The dataset has 103 instances with seven different numeric attributes. Some descriptive statistics of the dataset is given in Table 1 below.

Through our study, we have used and compare four different algorithms which are Support Vector Machines (SVM) (Cortes and Vapnik 1995), Artificial Neural Network (ANN) (Rosenblatt 1958), M5P decision trees (Quinlan 1992), and Multilayer Perceptron Regressor (MLPR) (Hornik *et al.* 1989) to explore the better prediction model in terms of various evaluation metrics. Because the output of a regression model is a continuous value, well-known evaluation metrics in machine learning domains such as accuracy, precision, recall, etc. are not suitable ones. Instead, some statistical metrics are preferred. Thus, we have applied four different statistical performance metrics which are Correlation coefficient (R), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Synthesis Index (SI). The formulations of chosen algorithms and evaluation metrics are given in section 3.1 and 3.2 respectively.

On the other hand, all algorithms run with k-folds cross-validation method which is a commonly used technique in applied machine learning in order to have a lower biased performance. The details of the method are given in through Section 3. In the experiment, we have used Weka and Python 3.6 Scikit-learn which is a collection of machine learning algorithms library for data mining tasks. The dataset are normalized during the preprocessing step before creating an ultimate prediction model.

3.1 Algorithms

3.1.1 Support Vector Regression (SVR)

Support vector machines (SVM) is a well-known supervised learning algorithm originally proposed by Cortes and Vapnik (1995). Initially, the SVM is designed in order to solve binary classification problems that can be linearly separable. Then, it is extended as Support vector regression (SVR) by Vapnik *et al*. (1997) to handle regression problems.

Assuming we have a training dataset $\{(x_1, y_2), \ldots., (x_l, y_l)\}$, where each $x_i \in R^n$, $y_i \in R$ the decision function is given in Eq. (2)

$$f(x)=w\varphi(x)+b \quad (2)$$

with respect to $w \in R^n, b \in R$, where φ denotes a non-linear transformation from R^n to high dimensional space. The primal optimization problem is given in Eq. (3)

$$\text{f min } R_{reg}(f)=\frac{1}{2}\|W\|^2+C\sum_{i=0}^{l} S(f(x_i)-y_i) \quad (3)$$

where $S(\cdot)$ is a cost function and C is a constant moreover, vector w is given in Eq. (3).

$$w=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\varphi(x_i) \quad (4)$$

By substituting Eq. (4) into Eq. (2), the decision function can be rewritten as in Eq. (5).

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)(\varphi(x_i)\varphi(x))+b \quad (5)$$

In Eq. (5) the dot product can be replaced with kernel function $k(x_i, x)$ and Eq. (5) can be rewritten as shown in Eq. (6)

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)k(x_i,x)+b \quad (6)$$

The ε insensitive loss function is the most widely used cost function. The function is given in Eq. (7).

$$S(f(x)-y)=\begin{cases}|f(x)-y|-\varepsilon \text{ for } |f(x)-y|\geq\varepsilon \\ 0 \quad\quad , \text{otherwise}\end{cases} \quad (7)$$

where ε is the width of the regression tube, for a given value, the corresponding dual formulation is shown in Eq. (8).

$$\begin{aligned}&\text{argmax}_{\alpha_i,\alpha_i^*} \\ &-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_i-\alpha_i^*)k(x_i,x_j)-\varepsilon\sum_{j=1}^{l}(\alpha_i-\alpha_i^*) \\ &+\sum_{i,j=1}^{l}(\alpha_i-\alpha_i^*)\end{aligned} \quad (8)$$

subject to:

$$\sum_{i=1}^{l}\alpha_i-\alpha_i^*=0$$
$$C\geq\alpha_i\geq 0$$
$$C\geq\alpha_i^*\geq 0$$

where α_i and α_i^* are Lagrange multipliers.

3.1.2 Multilayer perceptron regressor (MLPReg)

Artificial Neural Networks - sometimes also called Multilayer perceptron- is emerged from neural structure of the brain by Anderson and McNeill (1992). The MLPReg consists of a set of connected nodes - called perceptron- which nonlinearly maps between input and output vector. Connections between perceptron utilized by weights and output data are a function of the sum of weighted inputs modified by an activation function. The logistic function is commonly used activation function due to its easily computed derivative. Because of the hardness of training and optimizing many layers the architecture of an MLPReg consists of several layers of neurons that are fully connected meaning each node connected to every single node in the next layer. The layers between input and output layers are called hidden layers. The simple structure of the MLP is shown in Fig. 1 below.

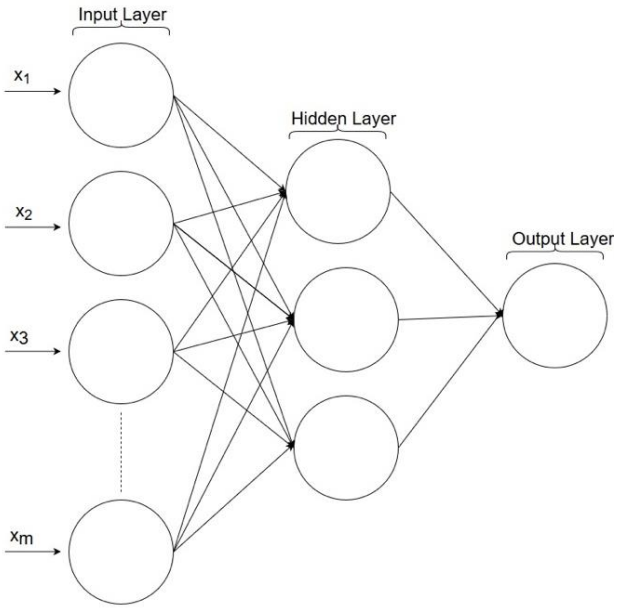

Fig. 1 Simple structure of MLP for regression problem (Ünlü 2019)

The output of a node is scaled by the connecting weight and fed forward to be an input to the nodes in the next layer of the network. The MLPReg is optimized by minimizing errors with iterated process back and forth which is called back-propagation algorithm working based on the error correction learning rule that feeds forward to transform connection weights to be as an input of the next layer (Haykin 1994).

Through our study, we use a conventional back-propagation MLPReg. The output of n^{th} neuron in l^{th} layer is calculated as shown in Eq (9).

$$y_l^n=\varphi\left[\sum_{j=1}^{p}w_{lj}^n(t)y_j^{n-1}(t)+\gamma_l^n\right] \quad (9)$$

here $\varphi(\cdot)$ is the activation function, w_{lj}^n is the connection weight, t is the time index, and $\gamma_l^n=w_{lj}^n(t)$ is weighted. For the n-layer network, the synaptic weight is calculated as in Eq. (10).

$$w_{ji}^n(t+1)=w_{ji}^n(t)+\Delta w_{ji}^n(t) \quad (10)$$

subject to $l\leq n\leq N$ and can be revised as given in Eq. (11).

$$\Delta w_{ji}^n(t)=\varepsilon\lambda_j^n(t)y_i^{n-1}(t) \quad (11)$$

Subject to $0\leq\varepsilon\leq 1$

where ε is the learning rate, and $\lambda_j^n(t)\equiv\partial E_t/\partial u_j^n$ is the local error gradient. To leverage the back-propagation

algorithm, a momentum term α is added as shown in Eq. (12).

$$\Delta w_{ji}^{n}(t)=\varepsilon\lambda_{j}^{n}(t)y_{i}^{n-1}(t)+\alpha\Delta w_{ji}^{n}(t-1) \quad (12)$$

Subject to $0 \leq \alpha \leq 1$

For the output layer, the local error gradient is given in Eq. (13).

$$\lambda_{j}^{N}(t)=\left[d_{j}(t)-y_{j}^{n}(t)\right]\varphi\left[u_{j}^{N}(t)\right]\equiv e_{j}(t)\varphi\left[u_{j}^{N}(t)\right] \quad (13)$$

where $d_j(t)$ is the goal output signal, and $\varphi(\cdot)$ is the activation function.

One needs to note that, we must choose appropriate activation function for the regression problem (i.e., the sigmoid or similar functions are not the suitable ones because the output values will be forced to be either zero or one). Thus, the neural network for the regression problem will be reconstructed as having a single neuron in the output layer which will produce a continuous value being linearly combination of the weights and inputs values.

3.1.3 M5P decision trees

Decision trees are widely used in machine learning and data analysis domains to visually and explicitly represent decision making for both the classification and regression problems. A tree model in general consists of branches and leaves named as conjunctions and nodes respectively in a data science context. It is a kind of a set of decision rules (i.e., taking true or false values if the value of nodes is greater than a threshold value or vice versa). M5P is a reconstructed model of Quinlan's M5 algorithm (Quinlan 1992) for creating trees of regression models. The main logic of the M5P is running a decision tree in conjunction with the possibility of linear regression functions at the nodes. Thus, M5P can be named as is a binary regression tree model where the last nodes utilized a linear regression function that can yield a continuous numerical output.

We can generalize the tree model as two stages. At first step, it involves the use of a divergence metric to create a decision tree. The branching benchmark for the M5P algorithm is the class values that reach a node as the amount of the error and the expected reduction in error as a result of testing each attribute at that node is calculated (Quinlan, 1992). The calculation of the Standard Deviation Reduction (SDR) is represented as in Eq. (14).

$$DR=sd(T)-\frac{\sum_{i=1}^{n}|T_i|}{|T|}\times sd(T) \quad (14)$$

where T is a collection of training cases, T_i is a subset of training cases, and sd is the standard deviation.

3.1.3 Random Forest trees (RF)

RF is an ensemble learning algorithm proposed by Breiman (2001). The main goal of the algorithm is to enhance the classification and regression trees (CART). It works based on the aggregation of a large number of decision trees with a bagging mechanism. It consists of a set of n trees $\{T_1(X), T_2(X) \dots T_n(X)\}$, where $X = \{x_1, x_2 \dots x_m\}$, is an m-dimensional input vector. The ensemble process creates n outputs corresponding each tree $\widehat{Y_1} = T_1(X), \widehat{Y_2} = T_2(X) \dots . \widehat{Y_t} = T_i(X)$, where $\widehat{Y_t}$, $i = 1,2, \dots ., n$, is the ith tree output. To obtain the final output, an average of all tree predictions is calculated. For each tree construction, it is used as a deterministic algorithm that creates a new training set (bootstrap samples) is selected with replacement. So, through the selection process while some data samples might be reselected and others might be left out of the sample which constitutes out-of-bag samples. Also, each node of trees only selects a small subset of attributes for the split, which yields the advantage of quick classification of high dimensional data. Besides the advantages, the main limitation of the RF algorithm is a large number of trees might make the algorithm slow and inefficient for real-time predictions.

3.2 Evaluation metrics

3.2.1 Correlation coefficient (R)

The Pearson product-moment correlation coefficient also known as *R* is a commonly used statistical measure proposed by Lawrence and Lin (1989). The *R* score aims to show how well the predicted values fit the actual output. The formulation of *R* is as in Eq. (15).

$$\left(\frac{n\sum y.y'-(\sum y)(\sum y')}{\sqrt{n(\sum y^2)-(\sum y)^2}-\sqrt{n(\sum y'^2)-(\sum y')^2}}\right)^2 \quad (15)$$

where *y*=actual values, *y*'=predicted value, and n= number of instances.

The value of R must lie between -1 and +1. So that 1 indicates a perfect fit between actual and predicted output with the highest propensity and -1 indicates vice versa.

3.2.2 Root mean squared error (RMSE)

RMSE is another frequently used statistical metric which is a measure of the differences between actual values and observed values by an algorithm. The calculation of RMSE can be calculated as shown in Eq. (16).

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-y_i')^2} \quad (16)$$

3.2.3 Mean Absolute Error (MAE)

Similar to other metrics described above MAE aims to summarize and assess the quality of the prediction model by converting all errors to positive. As the usual error is described the value between actual and predicted output. MAE is formulated as in Eq. (17).

$$\frac{1}{n}\sum_{i=1}^{n}|y-y'| \quad (17)$$

3.2.4 Relative absolute error (RAE)

The absolute error is the magnitude of the difference between the ground true output and the predicted one. The formulation of the RAE can be calculated based on Eq. (18).

$$RAE=\frac{|y'_1-y_1|+\cdots+|y'-y_n|}{|y_1-\bar{y}|+\cdots|y_n-\bar{y}|} \quad (18)$$

3.2.5 Root relative squared error (RRSE)

The RRSE takes the total squared error and normalizes it by dividing by the total squared error of the predictor. Due to taking the square root of the relative squared error, we can reduce the error without changing the dimensions of the quantity being predicted. The formulation of RRSE is shown in Eq. (19).

$$RRSE = \sqrt{\frac{(y'_1 - y_1)^2 + \cdots + (y'_n - y_n)^2}{(y_1 - \bar{y})^2 + \cdots + (y_n - \bar{y})^2}} \quad (19)$$

3.2.6 Synthesis Index (SI)

Finally, we can use SI to obtain a comprehensive performance measure which is calculated via four statistical measures in this study, MAE, RMSE, RAE, and RRSE. The calculation of SI is shown in Eq. (20).

$$SI = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{P_i - P_{min,i}}{P_{max,i} - P_{min,i}}\right) \quad (20)$$

3.3 K-folds cross-validation

K-folds cross-validation is a commonly used method in applied machine learning in order to have less bias performance. To perform a cross-validation method in conjunction with a predictor, the dataset D is randomly split k different subsets $(D_1, D_2, \ldots . D_k)$ as given in Eq. (21).

$$\bigcup_{i=1}^{k}\{D_1, D_2, \ldots ., D_k\} = D \quad (21)$$

Through the prediction process, one of each D_i is used for test and remaining is used for the training until each subset of the data is used as test set (i.e., for the *10-folds* cross-validation this process repeated as 10 times). In the end, the performance of the predictor is calculated as the average of *k* runs as shown in Eq. (22).

$$p^* = \frac{\sum_{i=1}^{k} p_i}{k} \quad (22)$$

where $(p_i, i = 1,2, \ldots . k)$ is the performance of predictor in i^{th} iteration.

K-folds cross-validation scheme is illustrated is shown in Fig. 2 in which *k* is chosen as 10 for illustration purpose.

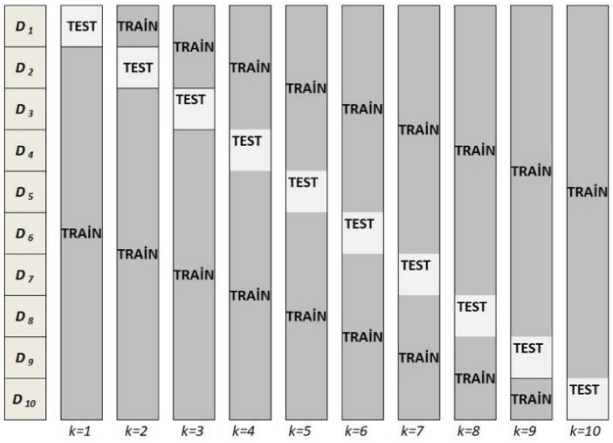

Fig. 2 *K*-folds cross-validation scheme (*k*=10 for illustration purpose)

Table 2 Pearson correlation between features and output

Attributes	Cement	Slag	Fly ash	Water	SP	Coarse	Fine	Flow
Cement	1.0000	0.2143	0.7179	0.5211	0.2079	0.5042	0.3537	0.4398
Slag	0.2143	1.0000	0.3492	0.0853	0.5996	0.1849	0.1744	0.4509
Fly ash	0.7179	0.3492	1.0000	0.4178	0.2039	0.5148	0.4220	0.1927
Water	0.5211	0.0853	0.4178	1.0000	0.3138	0.7987	0.3991	0.8598
SP	0.2079	0.5996	0.2039	0.3183	1.0000	0.0405	0.0196	0.4818
Coarse	0.5042	0.1849	0.5148	0.7987	0.0405	1.0000	0.6816	0.5674
Fine Aggr.	0.3537	0.1744	0.4220	0.3991	0.0196	0.6816	1.0000	0.4128
Flow	0.4398	0.4509	0.1927	0.8598	0.4818	0.5674	0.4128	1.0000

4. Results and discussions

In this chapter, we provide the detailed comparison of chosen methods regarding regression analysis for the given dataset. Before running the experiment, as mentioned in Section 3 all data features are normalized by z-score and we have used the 10-folds cross-validation method to ensure the reliability of the results of algorithms. Hyperparameters are crucial to create optimum efficiency. Even a small change in one of them can dramatically increase or decrease the performance of the method. Thus, finding optimum parameters is a crucial process. It can yield not only the best performance but also can save from computational time (i.e. very high training time might not increase the accuracy of the method while increasing computational time.). For this, we have utilized parameter optimization to get the best possible performance for each method by using grid search methodology which can be found in the Python Scikit-learn machine learning library. Basically, the value of each hyperparameter is selected from a range of predefined values and the combination of all hyperparameters giving the best performance is chosen.

In addition to parameter optimization, we have also deployed feature selection. Instead of finding the most relevant features to correct output, we look for redundant features to avoid both unnecessary information given by them and increasing computational time if the feature is not really important for an overall experiment in Lab. The following Table 2 illustrates the Pearson correlation between features and features and output.Based on Table 2 we can say that the most redundant features are Cement and Fly Ash with a correlation rate of 0.7179 and Cement and Water with a rate of about 0.5211. However, these are the main features, and ignoring one of them might not be a good idea. On the other hand, the water feature is the most relevant one to output with the correlation rate of about 0.8598. As expected the water is a mandatory ingredient for the mixture and having the most contribution while creating a prediction model. Hence, ignoring one feature to create a prediction model for the lump flow problem does not make sense based on given attributes. However, this method might be very helpful in the case of adding an unusual attribute in some Lab experiment. Hence, we can move

Table 3 Pearson correlation between features and output

Method	Bag size percent	Iterations
RF	100	1500

Table 4 The results of RF method in terms of evaluation metrics

Evaluation Metrics	R^2	MAE	RMSE	RAE	RRSE	SI
Results	0.68	9.78	12.55	68.28	72.70	0.71

Table 5 The results of M5P-Trees method in terms of evaluation metrics

Evaluation Metrics	R^2	MAE	RMSE	RAE	RRSE	SI
Results	0.68	10.38	12.7	72.4	73.6	0.85

Table 6 Optimum parameters for the SVR methods.

Method	Learning Algorithm	C	Kernel	Variance Prop
SVM	RegSMOimproved	1	PolyKernel	0.0001

further our experiment by using all available attributes. The following Table 3 shows the best parameter of the Random Forest (RF) method for the given problem.

Table 4 shows the results of the RF method for the given regression problem. It can be seen that the RF slightly captures the pattern of the actual value (R^2=0.68). Because of this poor performance error rates are not sufficient to be able to ignore the Lab process by using the RF method.

For the M5P-Trees method, we have assigned 8 as the value of the minimum number of instances parameter as a result of the grid search method. Table 5 illustrates the results of M5P-Trees regarding evaluation metrics. M5P-Trees method one of the worst methods among the chosen methods based on R2 and other error metrics. Although the correlation between the predictions and actual values is positive, it is not enough to close to 1 to used M5P-Trees for future automated predictions. So, we can infer the M5P-Trees method could not capture enough the pattern of actual values.

Table 7 The results of SVR method in terms of evaluation metrics

Evaluation Metric	R^2	MAE	RMSE	RAE	RRSE	SI
Results	0.65	10.65	13.30	74.31	77.05	1

Table 8 Optimum parameters for the MLPReg method

Method	Activation Function	Loss Function	Ridge	Tolerance
MLPReg	Sigmoid	Squared Error	0.05	1e-6

For the M5P-Trees method, we have assigned 8 as the value of the minimum number of instances parameter as a result of the grid search method. Table 5 illustrates the results of M5P-Trees regarding evaluation metrics. M5P-Trees method one of the worst methods among the chosen methods based on R^2 and other error metrics. Although the correlation between the predictions and actual values is positive, it is not enough to close to 1 to used M5P-Trees for future automated predictions. So, we can infer the M5P-Trees method could not capture enough the pattern of actual values.

For the SVM method, the optimum parameters are given below in Table 6, and the results based on given parameters are provided in Table 7.

SVM is another low-performance algorithm for the given regression problem. The correlation coefficient between the actual and predicted values is very low, so causing the error values are high. It can be inferred that SVM is not a good choice among chosen methods to create a good performance prediction model for the given problem.

MLP is the last method that we have used. The advantage of the MLP is to be able to capture the nonlinear

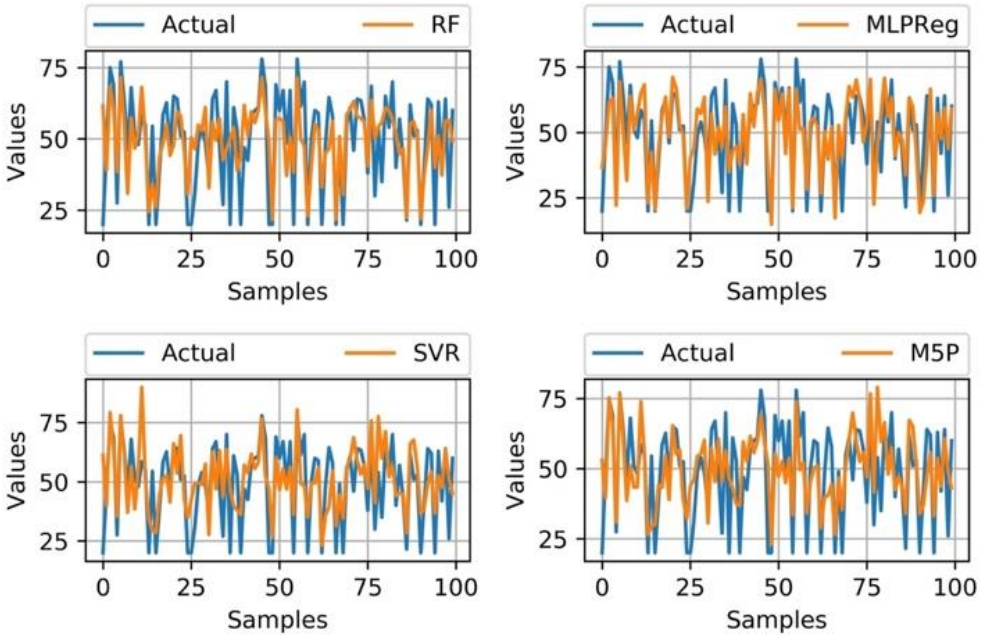

Fig. 3 Comparison of all methods

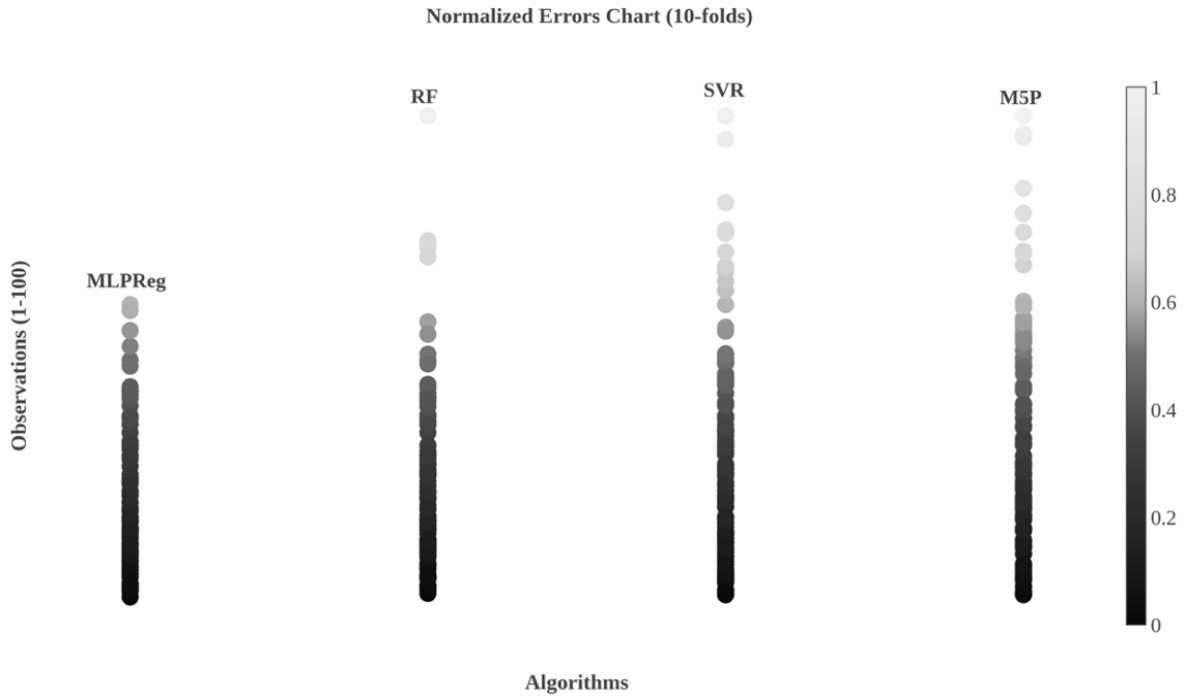

Fig. 4 Normalized error rates of chosen methods based on 10 Cross-Validation

Table 9 The results of MLPReg method in terms of evaluation metrics

Evaluation Metrics	R^2	MAE	RMSE	RAE	RRSE	SI
Results	0.80	8.13	10.19	56.75	59.00	0

relationship between the input and output values. The following Table 8 illustrates the optimum parameters for the MLP method.

Among all methods MLPReg is the best one by capturing the relation between the actuals and predicted values based on the result given in Table 9. It yields a positive R^2 which is rather close to 1. Also, it provides low error values in comparison to other methods. Hence, MLPReg is the best candidate among the compared methods.

To visually represent the comparison of all methods we provide the following Figs. 3 and 4. The comparison of predicted values and actual values can be seen in Fig. 3 (including training and test data). The performance of chosen methods based on the magnitude of the errors can be seen in Fig. 4. One needs to note that error is normalized for visualization convenience. We can see that RF, SVM, and M5P behaves similarly. For some data points, their error skews through the one. However, MLPReg errors have less variation and far from 1 which means it's predictions closer to actual ones than others' are. In addition to Fig. 3 given below, the complete solutions of the methods (including test and training data) are given in Table 11.

Following Table 10 shows the percentage of the improvement succeed by methods. The reference point is the SVR result. To make it more concrete, for example, M5P improve SVR 3.69% in terms of R^2 score, RF improved SVR 8.11% in terms of MAE, MLPReg improved SVR 23.43% in terms of RMSE and so on. One can see that

Table 10 Improvement rates by methods

Metrics	Increased by (%)			
	SVR	M5P	RF	MLReg
R^2	0.6506	3.69	4.95	23.96
MAE	10.6462	-2.53	-8.11	-23.63
RMSE	13.3015	-4.47	-5.66	-23.43
RAE	74.3138	-2.53	-8.11	-23.63
RRSE	77.0584	-4.47	-5.66	-23.43
SI	1	14.89	29.25	100.00

MLPReg noticeably improve with regards to all chosen evaluation metrics.

5. Conclusions

The main conclusions drawn from the study can be summarized as follows 1) the most relevant feature to output is the Water. Based on the Pearson correlation test, the amount of the water can highly affect the slump flow 2) there exist redundant features such as Fly Ash and Cement if we set threshold value as at least %50 (see Table 2). However, ignoring one of them can harm the model in terms of generalization for future predictions because the abnormal amount of Cement or Fly ash might noticeably change the slump flow according to the Lab experiments. Hence, keeping them for creating a prediction model can help us to capture abnormal behavior. 3) MLPReg is a baseline method to create a prediction model based on giving low error. It gives much more compact results compared to chosen well-known machine learning approaches. With the selection of optimum hyperparameters, it can yield state-of-the-art performance.

In this study, the feature selection process is

Table 11 Actual values and predicted values by all methods

Samples	Actual	MLPReg	SVR	M5P	RF	Samples	Actual	MLPReg	SVR	M5P	RF
1	20	36.656	61.313	53.031	61.611	51	60	54.53	46.613	47.169	57.242
2	48.2	44.955	39.832	39.581	39.134	52	67	64.685	55.523	52.647	56.197
3	75	61.441	79.302	75.276	68.314	53	48.5	41.917	46.893	44.978	37.137
4	69	63.192	63.551	65.748	60.502	54	67	66.454	56.255	53.754	50.594
5	27.5	22.198	35.407	30.905	38.479	55	20	21.218	36.227	33.53	38.296
6	77	69.453	78.036	76.772	71.512	56	78	66.651	80.506	73.911	71.318
7	64	50.788	62.446	60.356	55.547	57	61.5	51.164	49.204	49.534	50.036
8	46	31.691	37.113	38.655	30.864	58	70	51.779	48.304	52.141	47.406
9	68	64.041	56.478	51.115	57.43	59	20	28.212	35.119	34.08	22.913
10	51	50.855	38.531	43.399	46.658	60	50	54.072	49.469	45.14	40.446
11	48	56.91	55.69	43.451	52.141	61	60	55.632	49.44	50.331	55.026
12	58.5	65.007	89.917	73.951	68.078	62	59	52.295	56.166	51.092	53.198
13	55	68.244	51.28	55.2	50.373	63	20	26.622	23.081	28.831	33.059
14	20	23.078	34.015	26.446	24.456	64	48	50.346	35.724	39.741	45.079
15	54.5	42.533	29.443	29.484	34.153	65	64.5	39.699	39.367	40.443	46.485
16	20	20.603	28.54	30.023	26.143	66	58	52.189	54.721	44.424	56.221
17	39	43.725	44.535	44.244	43.715	67	20	17.387	31.349	26.469	21.859
18	58.5	56.143	51.497	43.4	48.5	68	49	52.681	44.551	46.719	50.858
19	62.7	63.598	52.798	54.719	55.148	69	20	40.512	34.582	35.381	30.471
20	46	48.305	41.448	38.97	44.187	70	55	56.157	56.11	50.759	58.475
21	65	71.058	66.204	65.407	47.879	71	61	67.561	59.609	60.422	60.398
22	64	66.785	53.66	55.177	59.53	72	46	63.464	68.707	69.895	63.174
23	51	52.986	69.545	56.747	57.99	73	64	70.003	59.154	59.952	57.877
24	52.5	49.058	42.656	38.975	45.917	74	63.5	62.431	53.153	52.507	57.276
25	20	20.752	34.77	32.196	30.491	75	57	46.35	57.297	55.769	55.932
26	20	39.996	42.331	43.792	50.473	76	38	53.442	47.143	46.917	40.469
27	31	44.048	49.266	46.144	46.496	77	68.5	70.247	75.845	76.722	63.538
28	49	58.976	49.448	54.355	54.896	78	30	22.581	41.061	41.624	50.168
29	54	57.211	48.843	55.315	53.004	79	54	44.77	77.691	78.998	54.372
30	48.5	63.482	57.524	60.177	61.036	80	35	56.704	59.981	59.655	55.871
31	36.5	23.679	27.688	30.618	32.813	81	65	70.736	71.259	66.493	60.951
32	64	57.039	62.581	57.413	56.073	82	54	56.012	52.061	48.011	59.853
33	67	41.685	46.91	45.336	49.496	83	70	63.441	56.826	57.285	56.94
34	48	52.209	63.331	60.615	57.131	84	40	41.001	44.046	42.45	51.544
35	27	44.157	43.68	45.187	42.683	85	57	50.023	45.176	47.026	44.746
36	70	59.757	63.772	56.527	46.358	86	43	47.181	44.78	43.829	46.725
37	20	34.967	47.088	46.653	49.997	87	21.5	33.777	28.358	33.884	22.334
38	61	42.853	39.937	43.114	54.005	88	62	63.25	66.037	67.275	55.443
39	53	44.766	37.948	37.39	38.838	89	51	59.797	67.667	65.143	55.986
40	20	37.564	36.187	36.882	43.643	90	53	43.002	56.468	51.664	51.34
41	47	56.832	57.081	59.597	61.827	91	20	19.43	36.417	33.906	22.013
42	42.5	37.974	53.864	52.484	51.122	92	41.5	23.609	39.244	36.886	37.528
43	58	64.815	61.803	61.666	59.636	93	64	40.93	49.325	47.406	47.161
44	60	52.231	55.578	55.635	55.686	94	62	66.607	54.314	55.765	60.11
45	61	64.954	58.211	59.761	59.986	95	20	25.9	36.719	33.182	40.043
46	78	70.336	76.647	69.304	71.498	96	63	57.297	52.609	53.759	51.119
47	68.5	59.325	52.439	59.279	59.81	97	42	47.583	47.514	43.616	37.287
48	20	35.697	47.088	51.367	49.209	98	64	59.343	63.796	60.786	56.388
49	20	14.894	26.613	23.008	21.747	99	26	43.806	51.219	49.076	56.599
50	69	66.412	60.21	55.482	56.811	100	60	59.089	45.051	43.135	49.292

implemented the first time for the slump flow prediction and it is proved that even with a few numbers of components machine learning methods can give highly accurate results.

For future research, if more HPC data become available, deep learning approaches can be applied. Since deep learning approaches gives brilliant results in the different domain by based on the studies in the literature. A more complex network can be fed by more HPC data and it is possible to get much more accurate results.

References

Anderson, D. and George, M. (1992), "Artificial neural networks technology", *Kaman Sci. Corp.*, **258**(6), 1-83.

Aydogmus, H.Y., Erdal, H.İ., Karakurt, O., Namli, E., Turkan, Y.S. and Erdal, H. (2015), "A comparative assessment of bagging ensemble models for modeling concrete slump flow", *Comput. Concrete*, **16**(5), 741-757. http://dx.doi.org/10.12989/cac.2015.16.5.741.

Breiman, L. (2001), "Random forests", *Mach. Learn.*, **45**(1), 5-32.

Brouwers, H.J.H. and Radix, H.J. (2005), "Self-compacting concrete: theoretical and experimental study", *Cement Concrete Res.*, **35**(11), 2116-2136. https://doi.org/10.1016/j.cemconres.2005.06.002.

Busari, A., Joseph, A. and Bamidele, D. (2018b), "Mechanical properties of dehydroxylated kaolinitic clay in self-compacting concrete for pavement construction", *Silicon*, **11**, 2429–2437. https://doi.org/10.1007/s12633-017-9654-6.

Busari, A.A., Joseph, O.A. and Bamidele, I.D. (2018a), "Review of sustainability in self-compacting concrete: the use of waste and mineral additives as supplementary cementitious materials and aggregates", *Portugaliae Electrochimica Acta*, **36**(3), 147-62. http://dx.doi.org/10.4152/pea.201803147.

Cao, Y.F., Wei, W., Han, L.Z. and Jun, M.P. (2013), "Prediction of the elastic modulus of self-compacting concrete based on svm", *Appl. Mech. Mater.*, **357-360**, 1023-1026. https://doi.org/10.4028/www.scientific.net/AMM.357-360.1023.

Chandwani, V., Vinay, A. and Ravindra, N. (2015), "Modeling slump of ready mix concrete using genetic algorithms assisted training of artificial neural networks", *Exp. Syst. Appl.*, **42**(2), 885-893. https://doi.org/10.1016/j.eswa.2014.08.048.

Cheng, M.Y., Chou, J.S., Roy, A.F. and Wu, Y.W. (2012), "High-performance concrete compressive strength prediction using time-weighted evolutionary fuzzy support vector machines inference model", *Auto. Constr.*, **28**, 106-115. https://doi.org/10.1016/j.autcon.2012.07.004.

Cortes, C. and Vapnik, V. (1995), "Support-vector networks", *Mach. Learn.*, **20**(3), 273-297. https://doi.org/10.1007/BF00994018.

Dias, W.P.S. and Pooliyadda, S.P. (2001), "Neural networks for predicting properties of concretes with admixtures", *Constr. Build. Mater.*, **15**(7), 371-379. https://doi.org/10.1016/S0950-0618(01)00006-X.

Domone, P. (1998), "The slump flow test for high-workability concrete", *Cement Concrete Res.*, **28**(2), 177-182. https://doi.org/10.1016/S0008-8846(97)00224-X.

Ferrara, L., Park, Y.D. and Shah, S.P. (2007), "A method for mix-design of fiber-reinforced self-compacting concrete", *Cement Concrete Res.*, **37**(6), 957-971. https://doi.org/10.1016/j.cemconres.2007.03.014.

González-Taboada, I., González-Fonteboa, B., Martínez-Abella, F. and Seara-Paz, S. (2018), "Evaluation of self-compacting recycled concrete robustness by statistical approach", *Constr. Build. Mater.*, **176**, 720-736. https://doi.org/10.1016/j.conbuildmat.2018.05.059.

Habibi, A. and Ghomashi, J. (2018), "Development of an optimum mix design method for self-compacting concrete based on experimental results", *Constr. Build. Mater.*, **168**, 113-123. https://doi.org/10.1016/j.conbuildmat.2018.02.113.

Haykin, S. (1994), *Neural Networks: a Comprehensive Foundation*, Prentice Hall PTR.

Hornik, K., Stinchcombe, M. and White, H. (1989), "Multilayer feedforward networks are universal approximators", *Neur. Network.*, **2**(5), 359-366.

Jain, A., Jha, S.K. and Misra, S. (2008), "Modeling and analysis of concrete slump using artificial neural networks", *J. Mater. Civil Eng.*, **20**(9), 628-633. https://doi.org/10.1061/(ASCE)0899-1561(2008)20:9(628).

Khatib, J.M. (2008), "Performance of self-compacting concrete containing fly ash", *Constr. Build. Mater.*, **22**(9), 1963-1971. https://doi.org/10.1016/j.conbuildmat.2007.07.011.

Lawrence, I. and Lin, K. (1989), "A concordance correlation coefficient to evaluate reproducibility", *Biometrics*, 255-268. https://doi.org/10.2307/2532051.

Lee, S.C. (2003), "Prediction of concrete strength using artificial neural networks", *Eng. Struct.*, **25**(7), 849-857. https://doi.org/10.1016/S0141-0296(03)00004-X.

Mashhadban, H., Kutanaei, S.S. and Sayarinejad, M.A. (2016), "Prediction and modeling of mechanical properties in fiber reinforced self-compacting concrete using particle swarm optimization algorithm and artificial neural network", *Constr. Build. Mater.*, **119**, 277-287. https://doi.org/10.1016/j.conbuildmat.2016.05.034.

Massana, J., Reyes, E., Bernal, J., León, N. and Sánchez-Espinosa, E. (2018), "Influence of nano-and micro-silica additions on the durability of a high-performance self-compacting concrete", *Constr. Build. Mater.*, **165**, 93-103. https://doi.org/10.1016/j.conbuildmat.2017.12.100.

Mechtcherine, V., Gram, A., Krenzer, K., Schwabe, J.H., Shyshko, S. and Roussel, N. (2014), "Simulation of fresh concrete flow using discrete element method (dem): theory and applications", *Mater. Struct.*, **47**(4), 615-630. https://doi.org/10.1617/s11527-013-0084-7.

Okamura, H. and Ouchi, M. (2003), "Self-compacting concrete", *J. Adv. Concrete Technol.*, **1**(1), 5-15. https://doi.org/10.3151/jact.1.5.

Okamura, H., Ozawa, K. and Ouchi, M. (2000), "Self-compacting concrete", *Struct. Concrete*, **1**(1), 3-17. https://doi.org/10.3151/jact.1.5.

Öztaş, A., Pala, M., Özbay, E., Kanca, E., Caglar, N. and Bhatti, M.A. (2006), "Predicting the compressive strength and slump of high strength concrete using neural network", *Constr. Build. Mater.*, **20**(9), 769-75. https://doi.org/10.1016/j.conbuildmat.2005.01.054.

Pelisser, F., Vieira, A. and Bernardin, A.M. (2018), "Efficient self-compacting concrete with low cement consumption", *J. Clean. Prod.*, **175**, 324-332. https://doi.org/10.1016/j.jclepro.2017.12.084.

Quinlan, J.R. (1992), "Learning with continuous classes", *5th Australian Joint Conference on Artificial Intelligence*, World Scientific, 343-348.

Rosenblatt, F. (1958), "The perceptron: a probabilistic model for information storage and organization in the brain", *Psychol. Rev.*, **65**(6), 386. https://doi.org/10.1037/h0042519.

Roussel, N. (2006), "Correlation between yield stress and slump: comparison between numerical simulations and concrete rheometers results", *Mater. Struct.*, **39**, 501-509. ttps://doi.org/10.1617/s11527-005-9035-2.

Saak, A.W., Jennings, H.M. and Shah, S.P. (2001), "New methodology for designing self-compacting concrete", *Mater. J.*, **98**(6), 429-439.

Şahmaran, M., Christianto, H.A. and Yaman, İ.Ö. (2006), "The effect of chemical admixtures and mineral additives on the properties of self-compacting mortars", *Cement Concrete Compos.*, **28**(5), 432-440. https://doi.org/10.1016/j.cemconcomp.2005.12.003.

Sari, M., Prat, E. and Labastire, J.F. (1999), "High strength self-compacting concrete original solutions associating organic and inorganic admixtures", *Cement Concrete Res.*, **29**(6), 813-818. https://doi.org/10.1016/S0008-8846(99)00037-X.

Siddique, R., Aggarwal, P. and Aggarwal, Y. (2011), "Prediction of compressive strength of self-compacting concrete containing bottom ash using artificial neural networks", *Adv. Eng. Softw.*, **42**(10), 780-786. https://doi.org/10.1016/j.advengsoft.2011.05.016.

Sonebi, M., Cevik, A., Grünewald, S. and Walraven, J. (2016),

"Modelling the fresh properties of self-compacting concrete using support vector machine approach", *Constr. Build. Mater.*, **106**, 55-64. https://doi.org/10.1016/j.conbuildmat.2015.12.035.

Su, N., Hsu, K.C. and Chai, H.W. (2001), "A simple mix design method for self-compacting concrete", *Cement Concrete Res.*, **31**(12), 1799-1807. https://doi.org/10.1016/S0008-8846(01)00566-X.

Tregger, N., Gregori, A., Ferrara, L. and Shah, S. (2012), "Correlating dynamic segregation of self-consolidating concrete to the slump-flow test", *Constr. Build. Mater.*, **28**(1), 499-505. https://doi.org/10.1016/j.conbuildmat.2011.08.052.

Ünlü, R. (2019), "A comparative study of machine learning and deep learning for time series forecasting: a case study of choosing the best prediction model for turkey electricity production", *J. Nat. Appl. Sci.*, **23**(2), 635-646. https://doi.org/10.19113/sdufenbed.494396.

Uysal, M. and Kemalettin, Y. (2011), "Effect of mineral admixtures on properties of self-compacting concrete", *Cement Concrete Compos.*, **33**(7), 771-776. https://doi.org/10.1016/j.cemconcomp.2011.04.005.

Uysal, M. and Sumer, M. (2011), "Performance of self-compacting concrete containing different mineral admixtures", *Constr. Build. Mater.*, **25**(11), 4112-4120. https://doi.org/10.1016/j.conbuildmat.2011.04.032.

Vapnik, V., Golowich, S. and Smola, A. (1997), Advances in Neural Information Processing Systems 9, Chapter Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing.

Yeh, I.C. (1999), "Design of high-performance concrete mixture using neural networks and nonlinear programming", *J. Comput. Civil Eng.*, **13**, 36-42. https://doi.org/10.1061/(ASCE)0887-3801(1999)13:1(36).

Yeh, I.C. (2007), "Modeling slump flow of concrete using second-order regressions and artificial neural networks", *Cement Concrete Compos.*, **29**(6), 474-80. https://doi.org/10.1016/j.cemconcomp.2007.02.001.

CC