Computers and Concrete, *Vol. 16, No.* 5 *(2015) 741-757*
DOI: http://dx.doi.org/10.12989/cac.2015.16.5.741

A comparative assessment of bagging ensemble models for modeling concrete slump flow

Hacer Yumurtacı Aydogmus[*1], Halil Ibrahim Erdal[2a], Onur Karakurt[3b], Ersin Namli[4a], Yusuf S. Turkan[4a] and Hamit Erdal[5b]

[1]*Department of Industrial Engineering, Alanya Alaaddin Keykubat University, 07450, Alanya-Antalya, Turkey*
[2]*Turkish Cooperation and Coordination Agency (TİKA), Atatürk Bulvarı No:15 Ulus-Ankara, Turkey*
[3]*Department of Civil Engineering, Gazi University, Celal Bayar Bulvarı, 06570, Maltepe-Ankara, Turkey*
[4]*Department of Industrial Engineering, Istanbul University Engineering Faculty, 34320, İstanbul, Turkey*
[5]*Institude of Social Sciences, Ataturk University, 25240, Erzurum, Turkey*

(Received November 1, 2014, Revised November 3, 2015, Accepted November 6, 2015)

Abstract. In the last decade, several modeling approaches have been proposed and applied to estimate the high-performance concrete (HPC) slump flow. While HPC is a highly complex material, modeling its behavior is a very difficult issue. Thus, the selection and application of proper modeling methods remain therefore a crucial task. Like many other applications, HPC slump flow prediction suffers from noise which negatively affects the prediction accuracy and increases the variance. In the recent years, ensemble learning methods have introduced to optimize the prediction accuracy and reduce the prediction error. This study investigates the potential usage of bagging (Bag), which is among the most popular ensemble learning methods, in building ensemble models. Four well-known artificial intelligence models (i.e., classification and regression trees CART, support vector machines SVM, multilayer perceptron MLP and radial basis function neural networks RBF) are deployed as base learner. As a result of this study, bagging ensemble models (i.e., Bag-SVM, Bag-RT, Bag-MLP and Bag-RBF) are found superior to their base learners (i.e., SVM, CART, MLP and RBF) and bagging could noticeable optimize prediction accuracy and reduce the prediction error of proposed predictive models.

Keywords: bagging (bootstrap aggregating); classification and regression trees; ensemble learning; multilayer perceptron; support vector machines

1. Introduction

Self-compacting concrete (SCC) is a highly flowable concrete which does not segregate and can spread into place, fill the formwork with heavily congested reinforcement without any mechanical vibration Nanthagopalan and Santhanam (2011). The three key characteristics of self-consolidating concrete (SCC) are flowability, segregation resistance and passing ability.

*Corresponding author, Ph. D., E-mail: hacera@akdeniz.edu.tr
[a] Ph.D., E-mail: halilerdal@yahoo.com, enamli@istanbul.edu.tr
[b] Ph.D. Student, E-mail: karakurtonur@yahoo.com, hamit_erdal@hotmail.com

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

Quality control of flowability is typically predicted by the final diameter of a slump-flow test; a larger diameter indicates higher flowability Tregger *et al*. (2012). The main property that defines SCC is high workability in attaining consolidation and specified hardened properties Ferraris *et al*. (2000).

The High-Performance Concrete (HPC) technology has been the important subject in concrete research since HPC has unique properties and numerous advantages in practical applications. Among them, the fine workability (i.e., easy placing and consolidation) is one of the most representative characteristics Yen *et al*. (1999).

In addition to four basic ingredients of the conventional concrete, i.e., Portland cement, fine and coarse aggregates and water, the making of HPC needs to incorporate the supplementary cementitious materials such as fly ash and blast furnace slag, and chemical admixture and super plasticizer. The use of fly ash and blast furnace slag plays an important role in contributing to a better workability and low slump loss rates of HPC Yeh (2007).

The diverse requirements of mixability, stability, transportability, placeability, mobility, compactability and finishability of fresh concrete mentioned above are collectively referred to as workability Gambhir (2004). According to Yeh (2007) workability of concrete is defined as the property determining the effort required to manipulate a freshly mixed quantity of concrete with minimum loss of homogeneity. Also The American Concrete Institute (ACI 116R-00, 73) describes workability as "that property of freshly mixed concrete or mortar that determines the ease with which it can be mixed, placed, consolidated, and finished to a homogenous condition." The Japanese Association of Concrete Engineers defines workability as "that property of freshly mixed concrete or mortar that determines the ease and homogeneity with which it can be mixed, placed, and compacted due to its consistency, the homogeneity with which it can be made into concrete, and the degree with which it can resist separation of materials" Ferraris (1999).

There are many methods and flow test is a widely used method for measuring the workability of concrete. Flow test gives the satisfactory performance for concretes of the consistencies for which slump test can be used. The test consists of moulding a fresh concrete cone on the top of the platform of the table. The spread of concrete, measured as the increase in diameter of cone, is taken as a measure of the flow or consistency of the concrete Gambhir (2004).

In recent years, machine learning methods have been applied to many civil engineering problems. Chou and Tsai (2012) developed a hierarchical classification and regression (HCR) approach for predicting (HPC) compressive strength for high performance concrete. Kewalramani and Gupta (2006) used multiple regression (MR) analysis and artificial neural networks (ANN) for prediction of compressive. Topcu and Sarıdemir (2008) developed ANN and fuzzy logic models for predicting the compressive strength. Chou and Pham (2013) compared ensemble models with individual numerical models in terms of their performance in predicting the compressive strength of HPC in their study. The best prediction performance results obtained via ensemble technique combining two or more models. Heshmati *et al*. (2008) proposed new formulations of compressive strength and slump flow using variant of genetic programming (GP).

The objective of this study is to investigate the potential usage of bagging (Bag), which is among the most popular ensemble learning methods, in building ensemble models.

2. Method

The following three performance measures are used to evaluate the proposed predictive

models. The correlation coefficient (R) is a common measure of how well the curve fits the actual data. A value of 1 indicates a perfect fit between actual and predicted values, meaning that the values have the same propensity. The mathematical formula for computing *R* is

$$R = \frac{n\sum y.y' - (\sum y)(\sum y')}{\sqrt{(\sum y^2) - (\sum y)^2}\sqrt{(\sum y'^2) - (\sum y')^2}} \tag{1}$$

where *y*= actual value, *y′*= predicted value, and *n*=number of data samples.

The root mean squared error (RMSE) is the square root of the mean square error. The RMSE is thus the average distance of a data point from the fitted line measured along a vertical line. The RMSE is given by the following equation

$$RMSE = \sqrt{\frac{\sum (y' - y)^2}{n}} \tag{2}$$

The mean absolute error (MAE) is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by

$$MAE = \frac{1}{n}\sum_{i=1,n} |y - y'| \tag{3}$$

According to Refaeilzadeh *et al*. (2009) cross-validation (CV) is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must cross-over in successive rounds such that each data point has a chance of being validated against. *K*-fold CV is the basic form of cross-validation and the other forms of cross-validation are special cases of *K*-fold CV or involve repeated rounds of *K*-fold CV.

In *K*-fold CV, the sample is divided into K different subsets $t_h (h = 1,2,\dots,K)$ of approximately equal size. The model is trained K times, each time leaving out one of the subsets from the training, but using only the omitted subset to compute the prediction error. The mean of these *K* values is the CV-estimate of the extra-sample error. Denote by $c^h (h = 1,2,\dots,K)$ the training set obtained by removing the h-th subset t_h and let $m = n/K$ be the number of units in each subset (assuming that *n* is a multiple of K) Borra and Di Ciaccio (2010). The CV-estimator is defined as the average error on the K analyses

$$err^{cv} = \frac{1}{K}\sum_{h=1}^{K}\frac{1}{m}\sum_{j\in t_h} L(y_j, \hat{f}_{c^h}(x_j)) \tag{4}$$

In this study, we applied a 10-fold CV to assess model predictive performance. First, the dataset is randomly divided into ten (*k*) subsets of equal size in which the class is represented in approximately the same proportions as in the full dataset. Next, each subset is held out in turn and the learning scheme trained on the remaining nine-tenths (*k-1*); then its error rate is calculated on the holdout set. Thus, the learning procedure is executed a total of ten times on different training sets Erdal (2013).

2.1 Bagging

Breiman's Bagging (bootstrap aggregating) is one of the first cases of an ensemble of decision trees Abellan and Masegosa (2012), Breiman (1996). It is noteworthy that most of the ensemble learning algorithms, such as Bagging and Boosting are designed for supervised learning Jia *et al*. (2011). A replica dataset of size *n* is randomly drawn with replacement from the original dataset of the n patterns. A bootstrap sample D_l may contain some in *D* multiple times, whereas others are not included. The idea of the bootstrap is that sampling from the actual dataset *D* is the best possible approximation for sampling from the unknown distribution *P*. Then a model is built by using this so-called bootstrap dataset. This procedure is repeated *T* times and thus results in *T* models. Then *T* models are aggregated by using the mean for regression problems Erdal *et al*. (2013). The bagging ensemble model structure developed in the present study is shown in Fig. 1.

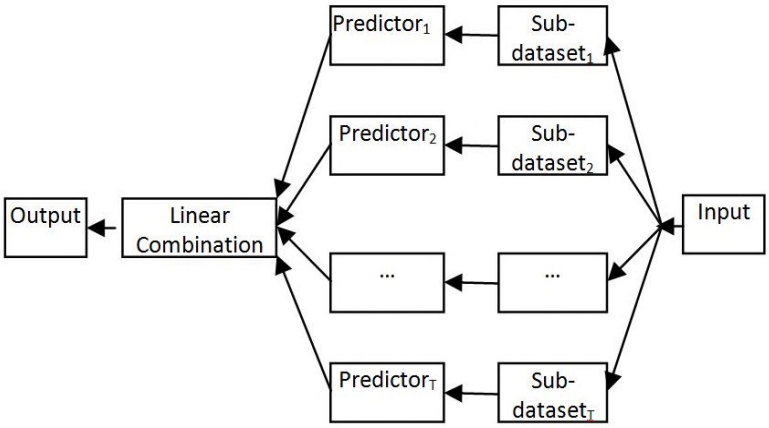

Fig. 1 Bagging ensemble model structure

As shown in Fig. 1 Bagging and Boosting have final prediction as a linear combination of classifiers. After several regression models are constructed, the average value of the predictions of each regression model gives the final prediction Karakurt *et al*. (2013).

For a regression problem, it works as follows Bühlmann and Yu (2002)

A training set of D consists of data $\{(X_i, Y_i), i=1,2,\ldots,n\}$ where X_i is a realization of a multi-dimensional predictor variable and Y_i is a realization of a real-valued variable. A predictor $(Y|X = x)=f(x)$ is denoted by

$$C_n(x) = h_n(D_1,...,D_n)(x) \tag{4}$$

Theoretically, bagging is defined as follows: First, construct a bootstrapped sample

$$D_i^* = (Y_i^*, X_i^*) \tag{5}$$

according to the empirical distribution of the pairs $D_i=(X_i, Y_i)$, where $(i=1,2,\ldots, n)$.

$$C_n^*(x) = h_n(D_i^*,...,D_n^*)(x) \tag{6}$$

Secondly, estimate the bootstrapped predictor by the plug-in principle,
where $C_n(x)=h_n(D_1,\ldots, D_n)(x)$

Finally, the bagged predictor is

$$C_{n;B}(x) = E\left|D_n^*(x)\right| \tag{7}$$

To sum up, bagging is one of the simplest to implement technique which can reduce variance when combined with the base learner generation, with a satisfactory performance Wang *et al.* (2011). A more detailed version of bagging is described in Breiman (1999).

2.2 Support vector regression

Support vector machine's soft margin classifier was introduced by Cortes and Vapnik (1995) for binary classification. SVM is based on the structural risk minimization method Vapnik (2000). Basak *et al*. (2007) stated that support vector classification (SVC) and support vector regression (SVR) are the two main categories for support vector machines. SVR, most common application form of SVM, has been proposed in 1997 by Vapnik, Steven Golowich, and Alex Smola Basak *et al*. (2007).

Considering a set of training data $\{(x_1, y_1),\ldots., (x_\ell, y_\ell)\}$, where each $x_i \in R^n$, $y_i \in R$, the decision function is given by

$$f(x)=(w\cdot\Phi(x))+b \tag{8}$$

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 by

$$\text{minimize} \quad R_{reg}(f) = \frac{1}{2}\|w\|^2 + C\sum_{i=0}^{\ell} S(f(xi) - yi) \tag{9}$$

where $S(\cdot)$ is a cost function and C is a constant moreover, vector w is given by

$$w = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)\Phi(x_i) \tag{10}$$

By substituting Eq. (8) into Eq. (10), the decision function is

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)(\Phi(x_i)\cdot\Phi(x)) + b \tag{11}$$

In Eq. (11) the dot product can be replaced with kernel function $k(x_i, x)$, Eq. (12) can be rewritten as

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)k(x_i, x) + b \tag{12}$$

The ε -insensitive loss function is the most widely used cost function. The function is given by

$$S(f(x)-y)=\begin{cases}|f(x)-y|-\varepsilon, & for \quad |f(x)-y|\geq\varepsilon\\ 0 & otherwise\end{cases} \tag{13}$$

where ε is the width of the regression tube, for given value, the corresponding dual formulation is by

$$\text{maximize } -\frac{1}{2}\sum_{i,j=1}^{\ell}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)k(x_i,x_j)-\varepsilon\sum_{j=1}^{\ell}(\alpha_i+\alpha_i^*)+\sum_{i,j=1}^{\ell}(\alpha_i-\alpha_i^*) \tag{14}$$

with respect to $\alpha_i,\alpha_i^* \in R^n$ and, subject to $\sum_{i=1}^{\ell}\alpha_i-\alpha_i^*=0 \quad C\geq\alpha_i^*\geq 0 \quad \& \quad C\geq\alpha_i\geq 0$

where $\alpha_i \text{ and } \alpha_i^*$ are Lagrange multipliers.

2.3 Multilayer perceptron

Artificial neural network, as a branch of artificial intelligence, is a simplified model based on the neural structure of the brain Anderson and McNeill (1992). Osman and Laporte (1996) noted that ANN has very powerful applications in scientific and engineering when used to predict, classify or recognize patterns due to the inherent data classification capabilities and massively parallel processing power.

ANN consists of a number of nodes representing neurons and these neurons are arranged into layers. The neurons from one layer are connected to the neurons in the two layers on either side of it. The perceptron is a simple neuron model that is used to classify its inputs into one of two categories Coppin (2004). Multilayer perceptrons as one of the neural network approaches are universal approximators, which have better generalization capabilities to capture complex relationship between inputs and outputs Thomas and Thomas (2011).

MLP is frequently trained with error back-propagation algorithm based on the error correction learning rule Haykin (1999).

This study uses a conventional back-propagation artificial neural network. The output signal for the l^{th} neuron in the n^{th} layer is given by

$$y_l^n(t)=\varphi[\sum_{j=1}^{p}w_{lj}^n(t)y_j^{n-1}(t)+\Psi_l^n] \tag{15}$$

where $\varphi(\cdot)$ is the activation function, w_{lj}^n is the connection weight, t is the time index and $\Psi_l^n=w_{l_0}^n(t)$ is the weighted. For an n-layer network, the synaptic weight $w_{ji}^n(t)$ is given by

$$w_{ji}^n(t+1)=w_{ji}^n(t)+\Delta w_{ji}^n(t) \tag{16}$$

subject to $l\leq n\leq N$ and it can be revised as given by

$$\Delta w_{ji}^n(t)=\eta\lambda_j^n(t)y_i^{n-1}(t) \tag{17}$$

subject to $0<\eta<1$

where η is the learning rate, and $\lambda_j^n(t) \equiv -\partial E_t / \partial u_j^n$ is the local error gradient. To improve the back-propagation algorithm, a momentum term α is added

$$\Delta w_{ji}^n(t) = \eta \lambda_j^n(t) y_i^{n-1}(t) + \alpha \Delta w_{ji}^n(t-1) \tag{18}$$

subject to $0< \alpha<1$

For the output layer, the local error gradient is given by

$$\lambda_j^N(t) = [d_j(t) - y_j^N(t)]\varphi[u_j^N(t)] \equiv e_j(t)\varphi[u_j^N(t)] \tag{19}$$

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

2.4 Radial basis function neural network

Radial basis function neural network (RBFNN) is the mostly adopted network topology because of numerous advantages such as better prediction capabilities, simpler network structures, and faster learning process Li *et al*. (2008). RBFNN is a feed forward neural network and involves three layers. These layers are listed as *input layer* which is consists of source nodes that connect the network to its environment, *hidden layer* applies a nonlinear transformation from the input space to the hidden space and *output layer* is linear, supplying the response of the network to the activation pattern applied to the input layer Haykin (1999).

The most general formula for any radial basis function neural network (RBF-NN) is

$$h(x) = \phi((x-c)^T R^{-1}(x-c)) \tag{20}$$

where c is the center, R is the metric and ϕ is the function Orr (1996). The metric is often Euclidean so that $R = r^2 I$ for some scalar radius r and the Eq. (13) simplifies to

$$h(x) = \phi\left(\frac{(x-c)^T (x-c)}{r^2}\right) \tag{21}$$

The simplification is a one-dimensional input space in which case $h(x) = \phi\left(\frac{-(x-c)^2}{r^2}\right)$

The Gaussian function $\phi(z) = e^{-z}$ is used. Therefore, a typical radial function is the Gaussian which, in the case of a scalar input, is

$$h(x) = \exp\left(\frac{-(x-c)^2}{r^2}\right) \tag{22}$$

2.5 Classification and regression trees

As mentioned in study of Erdal and Karakurt (2013), classification and regression trees (CART) has gained popularity in the recent years. The CART modeling which was proposed by

Breiman *et al*. (1984) is a non-parametric statistical methodology that can incorporate both numerical and categorical variables into the analysis Li (2006).

It works as follows Hancock *et al*. (2005): Each node within the tree has a partitioning rule and the partitioning rule is defined through minimization of the relative error (RE) which is the minimization of the sums-of-squares of a split for regression problems

$$\mathrm{RE(d)} = \sum_{l=0}^{L}[(y_1 - \overline{y_L}])^2 + \sum_{r=0}^{R}[(y_r - \overline{y_R}])^2 \tag{23}$$

Where y_1 and y_r are the left and right partitions with L and R observations of y in each, with respective means $\overline{y_L}$ and $\overline{y_R}$.The decision rule d is a point in some estimator variable x that is used to determine the left and right branches. The partitioning rule that minimizes the RE is then used to construct a node in the tree. The primary parameters for the CART are the following: the number of folds; the minimum total weight; and the number of seeds and the values for these parameters are 3, 2 and 1, respectively. A CART structure is depicted in Fig. 2.

3. Application and discussion of results

3.1 System data sets

One hundred - three various mix proportions collecting data were used to build the workability model. The data sets, based on Yeh's (2007) slump flow modeling study, obtained from HPC

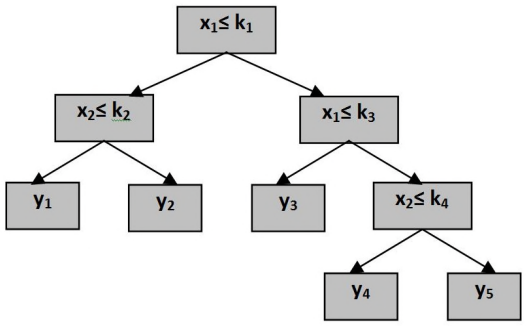

Fig. 2 A CART structure

Table 1 Ratio of data used in single and ensemble predictive models

Ratio of Data	Maximum	Minimum
Water / Cement	1.69	0.48
Fine Aggragate/Coarse Aggragate	1.18	0.66
W/(Cement + Fly ash+Slag)	0.66	0.29
SP/(Cement + Fly ash+Slag)	0.04	0.01
Fly ash/(Cement + Fly ash+Slag)	0.61	0.00
Slag/(Cement + Fly ash+Slag)	0.55	0.00
(Fly ash+slag)/(Cement + Fly ash+Slag)	0.74	0.00
Aggregate/(Cement + Fly ash+Slag)	5.56	2.36

produced with seven admixtures as cement (C), fly ash, blast furnace slag, water (W), superplasticizer (SP), coarse aggregate and fine aggregate. The general details of these data sets were given in Table 1. One hundred - three various data were evaluated by predictive single models (SVM, RBF, CART, MLP) and ensemble models (Bag-SVM, Bag-RBF, Bag-RT, Bag-MLP). The predicted values of these single and ensemble models compared with each other and the actual values.

3.2 Results and discussion

In this section, the results of analyses are presented. Tables 2-4 summarizes the performance measures of base learners (i.e., multilayer perceptron MLP, support vector machines SVM, classification and regression trees CART, radial basis function neural networks RBF) and bagging ensemble models (i.e., bagged-multilayer perceptron Bag-MLP, bagged-support vector machines Bag-SVM, bagged-classification and regression trees Bag-RT, bagged- radial basis function neural networks Bag-RBF). The mean absolute error (MAE), root mean squared error (RMSE) and correlation coefficient (R) performance statistics are used to evaluate the performance of the proposed predictive models.

Two empirical studies are conducted for this study according to training-testing approach i.e., (i) stratified 5-fold cross-validation and (ii) stratified 3-fold cross-validation. Split sample method could be used in the study which is a common technique to evaluate the predictive performance on a test set. In split sample method, a subsample of the dataset is kept back from training and

Table 2 Performance statistics of proposed predictive models for 5-fold cross validation

	Single models				Ensemble models			
	SVM	RBF	CART	MLP	Bag-SVM	Bag-RBF	Bag-RT	Bag-MLP
R	0.6299	0.4092	0.5015	0.7112	0.6652	0.592	0.6219	0.7741
MAE (cm)	11.15	13.49	12.26	10.59	10.87	11.97	10.78	8.27
RMSE (cm)	13.87	16.42	15.79	14.68	13.24	14.25	13.70	11.24

Table 3 Relative improvements of predictive models for 5-fold cross validation

Method A	Method B	Improvement		
		R	**MAE (cm)**	**RMSE (cm)**
SVM	Bag-SVM	5.60%	2.51%	4.52%
RBF	Bag-RBF	44.67%	11.29%	13.18%
CART	Bag-RT	24.01%	12.02%	13.21%
MLP	Bag-MLP	8.84%	21.86%	23.48%

Table 4 Performance statistics of proposed predictive models for 3-fold cross validation

	Single models				Ensemble models			
	SVM	RBF	CART	MLP	Bag-SVM	Bag-RBF	Bag-RT	Bag-MLP
R	0.6394	0.5484	0.4984	0.6146	0.6533	0.5983	0.6722	0.7584
MAE (cm)	11.29	12.02	11.74	12.66	11.12	11.67	10.14	8.56
RMSE (cm)	14.01	14.65	15.48	16.39	13.64	14.13	12.99	11.84

employed to measure the accuracy of forecasting. Nevertheless, especially in small data sets, this method can be quite misleading and very dependent on the validation set, and cross-validation is accepted to be superior to the ordinary split-sample methods Aertsen *et al*. (2010).

The optimum parameters are chosen by monitoring the R of each of the models. The SVM's parameters are as follows: The kernel is, radial basis function kernel and Poly kernel; the complexity parameter is 1, 2 and 3; the epsilon is 1.0E-11 and 1.0E-12 the exponent is 1, 2 and 3 the best parameter configuration for this technique is: Poly kernel is chosen to be the kernel function for SVM moreover, the complexity parameter is 2, epsilon is 1.0E-11 and the exponent is 1. The primary parameters for the CART are the following: the number of folds; the minimum total weight and the number of seeds; the number of folds is 5, 10 and 15; the minimum total weight is 1, 2 and 3; the number of seeds is 1, 2 and 3. The best values for these parameters are 5, 2 and 1, respectively. The data is used in several experiments to obtain best parameters for MLP. The parameters for the network are: the number of hidden layers was 1, 2, and 3; the learning rate is 0.2, 0.3 and 0.4; the momentum factor is 0.2, 0.3, and 0.4; and the training epochs is 500 and 1,000. The experiments indicated that the best network parameters were as follows: the number of hidden layers is 2; the number of the learning rate is 0.3; the momentum factor is 0.2, and the training time is 500. The RBF's parameters tested in the proposed model included the following: the minimum standard deviation for is 0.1 and 0.5 the Ridge value is 1.0 E-7 and 1.0 E-8; the number of clusters are 1, 2 and 3, finally the maximum number of iterations is 1 and 5. The best values for RBF-NN are: the minimum standard deviation is 0.1, the Ridge value is 1.0 E-8, the number of clusters is 2, and finally the maximum number of iterations is 1. Bagging has three parameters of particular importance: the size of each bag (as a percentage); the number of iterations (number of trees); and the number of seeds; the size of each bag is 80, 90 and 100; the number of iterations is 30, 40 and 50; and the number of seeds is 1, 2 and 3. In this case, the values for these parameters are 100, 40, and 1, respectively. The base model (i.e., MLP, RBF, CART, and SVM) parameters are identical to the case in which are they are separately applied.

For measuring the prediction accuracy, only the test sample is considered because good learning (training) sample measures of the prediction accuracy give no guarantee for good test sample measures of the prediction accuracy. Table 2 indicates the results of the first empirical study (5-fold cross validation case). It is evident from the table that the Bag-MLP model has the best performance in term of R (0.7741), and single MLP model has the second best performance (R=0.7112). Bag-SVM model yields the third best performance (R=0.6652). Single CART (0.5015) and RBF (0.4092) models yields worst performances among all eight predictive models. Moreover, MAE and RMSE statics are inconsistent with the correlation coefficient statics. For minimizing RMSE statics, Bag-MLP (11.24 cm) is the best, Bag-SVM (13.24 cm) is the second, Bag-RT (13.70 cm) is the third and RBF (16.42 cm) is the worst model and Bag-MLP (8.27 cm) is the best, MLP (10.59 cm) is the second Bag-RT (10.78 cm) is the third and RBF (13.49 cm) is the worst model for minimizing MAE statics. The application of bagging has brought a substantial improvement for base learners: Bag-MLP (R=0.7741, MAE=8.27, RMSE=11.24), Bag-SVM (R= 0.6652, MAE=10.87, RMSE=13.24), Bag-RT (R=0.6219 , MAE=10.78, RMSE=13.70) and Bag-RBF (R=0.592 , MAE=11.97, RMSE=14.25) outperform their base learners MLP (R=0.7112, MAE=10.59, RMSE=14.68), SVM (R=0.6299, MAE=11.15, RMSE=13.87), CART (R=0.5015, MAE=12.26, RMSE=15.79) and RBF (R=0.4092, MAE=13.49, RMSE=16.42) in term of all three performance indicators.

Table 4 shows the results of second empirical study (3-fold cross validation case). It can be noted from the table that, for determining R, the Bag-MLP model is superior to other predictive

models (R=0.7584). Bag-RT model has the second best performance (0.6722) and Bag-SVM slightly performs worse (R=0.6533). Single CART (0.4984) model performs worst among the all predictive models. In addition, for reducing RMSE statics, Bag-MLP (11.84 cm) is the best, Bag-RT (12.99 cm) is the second, Bag-SVM (13.64 cm) is the third and MLP (16.39 cm) is the worst model. For minimizing MAE statics, Bag-MLP (8.56 cm) is the best, Bag-RT (10.14 cm) is the second Bag-SVM (11.12 cm) is the third and MLP (12.66 cm) is the worst model. The implementation of bagging has also brought a substantial enhancement for single models: Bag-MLP (R=0.7584, MAE=8.56, RMSE=11.84), Bag-RT (R=0.6722, MAE=10.14, RMSE=12.99) Bag-SVM (R=0.6533, MAE=11.12, RMSE=13.64), and Bag-RBF (R=0.5983, MAE=11.67, RMSE=14.13) outperform their base learners MLP (R=0.6146, MAE=12.66, RMSE=16.39), CART (R=0.4984, MAE=11.74, RMSE=15.48) SVM (R=0.6394, MAE=11.29, RMSE=14.01), and RBF (R=0.5484, MAE=12.02, RMSE=14.65) in term of all three performance indicators.

As shown Tables 3-5, bagging ensemble learning method can noticeably increase the accuracy and reduce the error statics of the single models (i.e., MLP, RBF, CART, and SVM). Bagging ensemble learning method enhances the accuracy of the MLP RBF, CART, and SVM by 23.40% – 8.84%, 44.67% – 9.10%, 23.40% – 8.84% and 5.60% – 2.17%, respectively. Moreover, bagging reduces the MAE of the MLP RBF, CART, and SVM by 32.41% – 21.86%, 11.29% – 2.96%, 13.67%– 12.02% and 2.51% – 1.49%, respectively, and it decreases RMSE of the MLP RBF, CART, and SVM by 27.77% – 23.48%, 13.18% – 3.61%, 16.09%– 13.21% and 24.52%– 2.61%, respectively.

The predicted slump flow values of these four ensemble models (BSVM, BRBF,BRT and BMLP) compared with the actual test result for 5-fold cross validation and 3-fold cross validation are shown in Figs. 3-4.

The predicted slump flow distributions of single and ensemble models and the distribution of

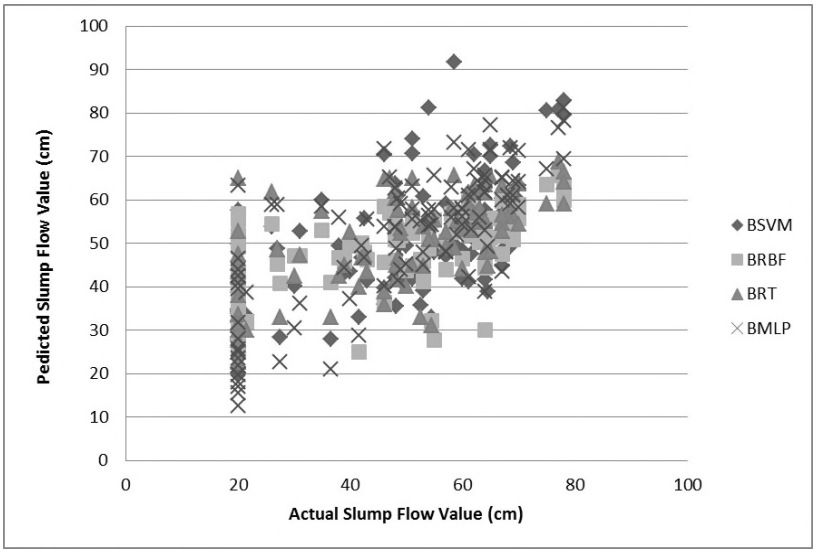

Fig. 3 Predicted slump flow values using ensemble learning models for 3-fold cross validation vs. actual test values

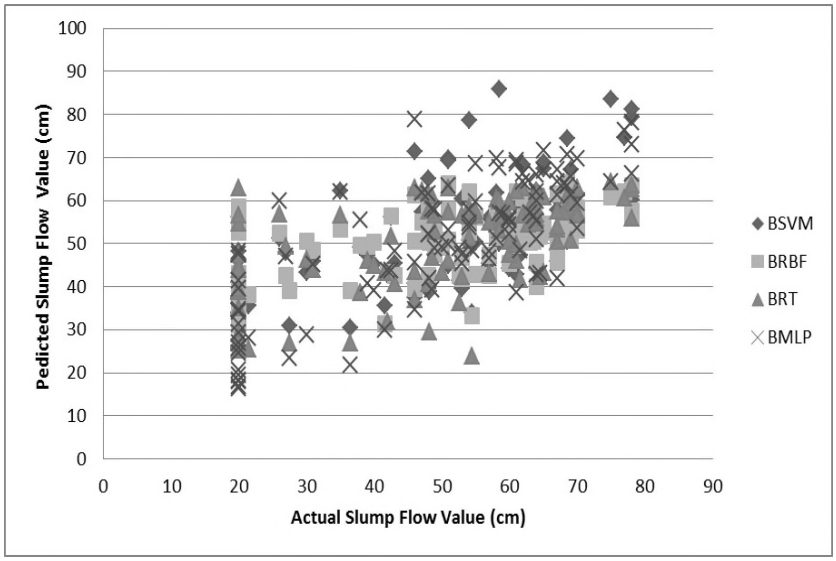

Fig. 4 Predicted slump flow values using ensemble learning models for 5-fold cross validation vs. actual test values

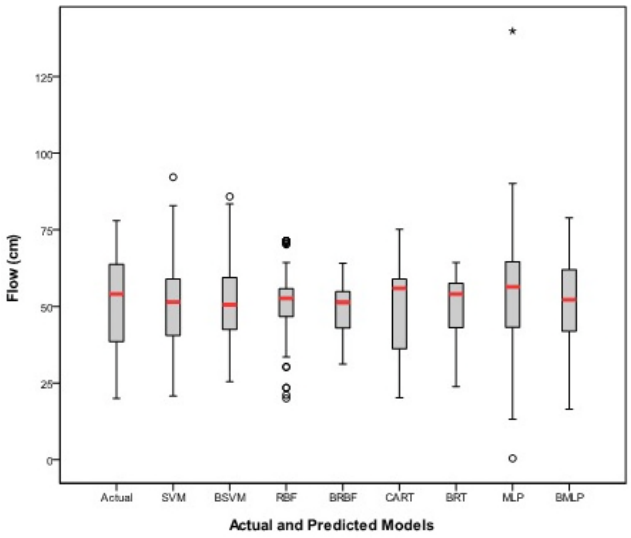

Fig. 5 Box plots of actual slump flow and predicted slump flow distributions of single and ensemble models for 5-fold cross validation

actual slump flow are shown on the boxplot graphs given in Figs. 5-6 and the statistics values of each distribution are given above Tables 6-7.

The distributions of models are nearly symmetric and the mean values close the median value and so that the distributions close to nearly zero skewness and kurtosis shown in Tables 5-6.

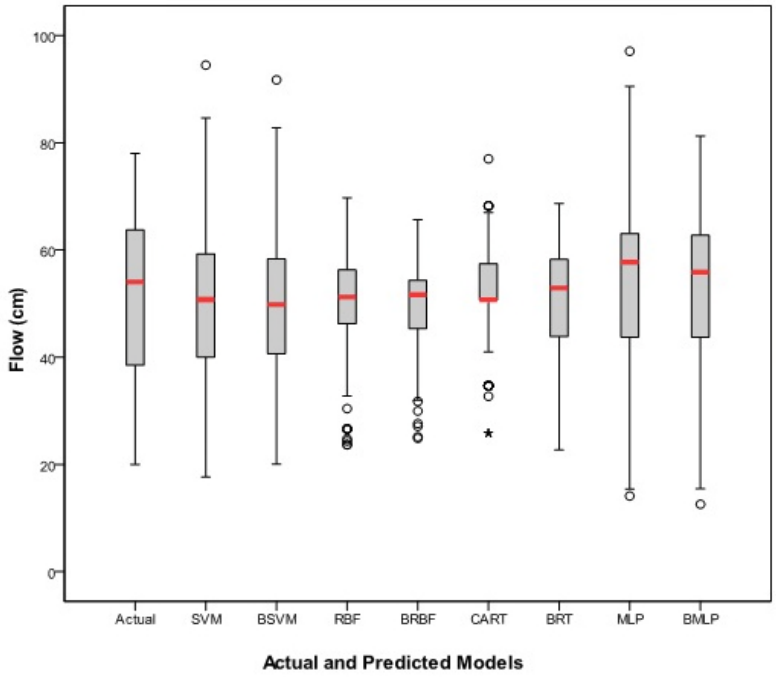

Fig. 6 Box plots of actual slump flow and predicted slump flow distributions of single and ensemble models for 3-fold cross validation

Table 5 Relative improvements of predictive models for 3-fold cross validation

Method A	Method B	Improvement		
		R	**MAE (cm)**	**RMSE (cm)**
SVM	Bag-SVM	2.17%	1.49%	2.61%
RBF	Bag-RBF	9.10%	2.96%	3.61%
CART	Bag-RT	34.87%	13.67%	16.09%
MLP	Bag-MLP	23.40%	32.41%	27.77%

Table 6 Statistics for 5-fold cross validation

	Actual	SVM	BSVM	RBF	BRBF	CART	BRT	MLP	BMLP
Mean	49.6107	51.2702	51.3274	50.7474	49.5476	48.9991	49.523	52.8808	50.5762
Std. Error of Mean	1.73109	1.31696	1.29258	1.07455	0.81340	1.31235	1.12980	1.95758	1.50587
Median	54.0000	51.3770	50.5480	52.5740	51.3460	55.9630	53.9540	56.3640	52.2310
Std. Deviation	17.56861	13.36571	13.11828	10.90553	8.25510	13.31892	11.46617	19.86730	15.28291
Variance	308.656	178.642	172.089	118.931	68.147	177.394	131.473	394.710	233.567
Skewness	-.521	.349	.365	-.442	-.531	-.671	-.805	.233	-.515
Kurtosis	-.898	.144	-.072	.805	-.368	-.617	-.462	2.933	-.423
Minimum	20.00	20.76	25.48	19.97	31.17	20.16	23.83	.40	16.39
Maximum	78.00	92.18	85.86	71.43	64.06	75.13	64.34	139.87	78.88

Table 7 Statistics for 3-fold cross validation

	Actual	SVM	BSVM	RBF	BRBF	CART	BRT	MLP	BMLP
Mean	49.6107	51.1044	50.7575	49.1936	48.9624	52.0545	50.4047	54.4501	51.9640
Std. Error of Mean	1.73109	1.46913	1.43854	1.03808	.86885	1.05823	1.09373	1.79840	1.52968
Median	54.0000	50.7280	49.7840	51.2170	51.6080	50.7100	52.8820	57.7400	55.8250
Std. Deviation	17.56861	14.91000	14.59954	10.53536	8.81785	10.73989	11.10010	18.25173	15.52460
Variance	308.656	222.308	213.146	110.994	77.755	115.345	123.212	333.126	241.013
Skewness	-.521	.239	.330	-.919	-.917	-.460	-.617	-.096	-.710
Kurtosis	-.898	-.070	-.109	.363	.447	-.009	-.381	-.274	-.041
Minimum	20.00	17.67	20.04	23.68	24.88	25.82	22.69	14.12	12.57
Maximum	78.00	94.48	91.74	69.73	65.64	77.00	68.68	97.07	81.25

The boxes given in the figures indicate the interquartile ranges and the bottom and top of the box are always the 25th and 75th percentile (the lower and upper quartiles), and the horizontal band near the middle of the box is always the 50th percentile (the median) and dots indicate outside values and asterisks indicate out values.

When the statistical distributions of the predicted slump flow and the actual data were compared, the performance of RBF, BRBF and CART models were worse when compared to the actual slump flow data and the other models.

It could be understood from the empirical results that bagging ensemble learning method can improve the prediction accuracy of their base predictors. Ensemble learning could be a very effective procedure when applied to unstable learning algorithms such as decision trees and neural networks Pino-Mejias *et al*. (2008). Like many other applications, HPC behavior (workability) prediction suffers from concurrent negative effects by the noise. Wang *et al*. (2012) discussed that bagging ensemble models introduce certain mechanisms to reduce the influence of the noise.

Bagging works because as mentioned before the main goal of bagging is minimizing variance in the prediction. In bagging, different learning sub-datasets are drawn at random with replacement from the entire learning dataset. Separate individual models are generated and are used to predict the entire learning data from aforesaid sub-datasets. Many of the original instances may be repeated in the resulting training set whereas others may be omitted. After several regression models are constructed, the average value of the predictions of each regression model gives the final prediction. This means that various same type of base learners are specialized in different parts of the observation space. Ensemble models are superior to their base models because using only one model to predict the HPC behavior may not illuminate the internal mechanism of the phenomenon. However, the increase in the prediction accuracy and the decrease in the prediction error of SVM due to bagging is limited according to other base learners (i.e., MLP, RBF and CART). This may happen because SVM is more rigid (stable) method than the other single models and its stiffness may influence the bagging process.

5. Conclusions

The main conclusions drawn from the study can be summarized as followed (i) bagging ensemble method could remarkably optimize the prediction accuracy and reduce the variance of

the simple CART and ANN models (i.e., MLP and RBF) (ii) the increase in the prediction accuracy and the decrease in the prediction error of SVM due to bagging is limited (iii) Bag-MLP is found superior to other predictive models for determining the three performance indicator. Overall, as a result of this study, bagging ensemble learning method is found promising and easy to implement a technique for HPC behavior prediction. In this study, only bagging method is used in building ensemble models. The other ensemble learning methods like boosting could be used in constructing ensembles. This study focuses on reducing the concurrent negative effects by the noise; however, eliminating redundant attributes is also another crucial issue in the prediction. Attribute-base ensemble learning methods like random sub-spaces can be used for reducing redundant attributes. These may be subject of future works.

Acknowledgements

The authors wish to express their gratitude to Professor I-Cheng Yeh and UC Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Concrete+Slump+Test) for sharing the experimental dataset.

References

Aertsen, W., Kint, V., Orshoven, J., Özkan, K. and Muys, B. (2010), "Comparison and ranking of different modelling techniques for prediction of site index in Mediterranean mountain forests", *Ecol. Model.*, **221**, 1119-1130.

Abellan, J. and Masegosa, A.R. (2012), "Bagging schemes on the presence of class noise in classification", *Exp. Syst. Appl.*, **39**(8), 6827-6837.

Anderson, D. and McNeill, G. (1992), *Artificial neural networks technology*. Rome Laboratory, A DACS State of the Art Report, ELIN: A011, New York.

Basak, D., Pal, S. and Patranabis, D.C. (2007), "Support vector regression", *Neu. Information Pro. Lett. Rev.*, **11**(10), 203-224.

Borra, S. and Di Ciaccio, A. (2010), "Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods", *Comput. Stat. Data An.*, **54**(12), 2976-2989.

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984), *Classification and regression trees*, Wadsworth, Int. Group, Belmont, California.

Breiman, L. (1996), "Bagging predictors", *Mach. Learn.*, **24**(2), 123-140.

Breiman, L. (1999), *Using adaptive bagging to debias regressions.*, Technical Report No. 547 of University of CA, Berkeley.

Bühlmann, P. and Yu, B. (2002), "Analyzing bagging", *Statist.*, **30**(4), 927-961.

Chou, J.S. and Tsai, C.F. (2012), "Concrete compressive strength analysis using a combined classification and regression technique", *Automat. Constr.*, **24**, 52-60.

Chou, J.S. and Pham, A.D. (2013), "Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength", *Constr. Build. Mater.*, **49**, 554-563.

Coppin, B. (2004), *Artificial intelligence illuminated*., Jones and Bartlett Publishers; Massachusetts.

Cortes, C. and Vapnik, V. (1995), "Support vector networks", *Mach. Learn.*, **20**, 273-297.

Erdal, H.I., Karakurt, O. and Namli, E. (2013), "High performance concrete compressive strength forecasting using ensemble models based on discrete wavelet transform", *Eng. Appl. Artif. Intel.*, **26**(4), 1246-1254.

Erdal, H.I. (2013), "Two-level and hybrid ensembles of decision trees for high performance concrete compressive strength prediction", *Eng. Appl. Artif. Intel.*, **26**(7), 1689-1697.

Erdal, H.İ. and Karakurt, O. (2013), "Advancing monthly streamflowprediction accuracy of CART models using ensemble learning paradigms", *J. Hydrol.*, **477**, 119-128.
Ferraris, C.F. (1999), "Measurement of the rheological properties of high performance concrete: state of the art report", *J. Res. Natl. Inst. Stan.*, **104**(5), 461-478.
Ferraris, C.F., Brower, L., Ozyildirim, C. and Daczko, J. (2000), "Workability of self-compacting concrete", *Int'l symposium on high performance concrete proceedings*, *Natl. Inst. Stan*, Florida.
Gambhir, M.L. (2004), *Concrete Technology*, Tata McGraw-Hill Publishing Company Limited.
Hancock, T., Put, R., Coomans, D., Vanderheyden, Y. and Everingham,Y. (2005), "A performance comparison of modern statistical techniques for molecular descriptor selection and retention prediction in chromatographic QSRR studies", *Chem. Intel. Lab.Syst.*, **76** (2), 185-196.
Haykin, S. (1999), *Neural Networks: A comprehensive foundation*, 2nd Editions Pearson Education, 9th Indian reprint 2005, India.
Heshmati, A.A.R., Salehzade, H., Alavi, A.H., Gandomi, A.H. and Abadi, M.M. (2008), "A multi expression programming application to high performance concrete", *World Appl. Sci. J.*, **5**(2), 215-223.
Jia, J., Xiao, X., Liu, B. and Jiao, L. (2011), "Bagging-based spectral clustering ensemble selection", *Pattern Recogn. Lett.*, **32**(10),1456-1467.
Karakurt, O., Erdal, H.I., Namlı, E., Yumurtacı Aydoğmuş, H. and Türkan, Y.S. (2013), "Comparing ensembles of decision trees and neural networks for one-day-ahead streamflow prediction", *Sci. Res. J. (SCIRJ)*, **I**(IV).
Kewalramani, M.A. and Gupta, R. (2006), "Concrete compressive strength prediction using ultrasonic pulse velocity through artificial neural networks", *Automat. Constr.*, **15**(3), 374-379.
Minqiang, Li., Tian, Jin. and Fuzan, Chen. (2008), "Improving multiclass pattern recognition with a co-evolutionary RBFNN", *Pattern Recogn. Lett.*, **29**(4), 392-406.
Li, Y. (2006), "Predicting materials properties and behavior using classification and regression trees", *Mater. Sci. Eng. A*, **433**, 261-268.
Nanthagopalan, P. and Santhanam, M. (2011), "Fresh and hardened properties of self-compacting concrete produced with manufactured sand", *Cement Concrete Comp.*, **33**(3), 353-358.
Orr, M.J.L. (1996), *Introduction to radial basis function networks: centre for cognitive science*, University of Edinburgh.
Osman, I.H. and Laporte, G. (1996), "Metaheuristics: a bibliography", *Ann. Oper. Res.*, **63**, 513-623.
Pino-Mejias, R., Jimenez-Gamero, M.D., Cubiles-de-la-Vega, M.D. and Pascual-Acosta, A. (2008), "Reduced bootstrap aggregating of learning algorithms", *Pattern Recogn. Lett.*, **29**, 265-271.
Refaeilzadeh, P., Tang, L. and Liu, H. (2009), *Cross-validation*, *Encyclopedia of database systems*. Springer.
Rodríguez, J.D., Perez, A. and Lozano, J.A. (2010), "Sensitivity analysis of k-fold cross validation in prediction error estimation", *Pattern Anal. Mach. Intel., IEEE Tran.*, ***32***(3), 569-575.
Thomas, P. and Thomas, A. (2011), "Multilayer perceptron for simulation models reduction: application to a sawmill workshop", *Eng. Appl. Artif. Intel.*, **24**, 646-657.
Topcu, I.B. and Sarıdemir, M. (2008), "Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic", *Comput. Mater. Sci.*, **41**(3), 305-311.
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**, 499-505.
Vapnik, V.N. (2000), *The Nature of Statistical Learning Theory*, 2nd Editions, Springer-verlag New York Inc.
Wang, G., Hao, J., Mab. J. and Jiang, H. (2011), "A comparative assessment of ensemble learning for credit scoring", *Exp. Syst. Appl.*, **38**, 223-230.
Wang, G., Ma, J., Huang, L. and Xu, K. (2012), "Two credit scoring models based on dual strategy ensemble trees", *Knowl. Based Syst.*, **26**, 61-68.
Yeh, I.C. (2007), "Modeling slump flow of concrete using second-order regressions and artificial neural networks", *Cement Concrete Comp.*, **29**, 474-480.
Yen, T., Tang, C.W., Chang, C.S. and Chen, K.H. (1999), "Flow behaviour of high strength

high-performance concrete", *Cement Concrete Comp.*, **21**, 413-424.
Vapnik, V., Golowich, S.E. and Smola, A. (1996), "Support vector method for function approximation, regression estimation, and signal processing", *Adv. Neu. Inform. Pr. Syst. 9*.

CC