Optimizing the Prediction Accuracy of Concrete Compressive Strength Based on a Comparison of Data-Mining Techniques

Jui-Sheng Chou, P.E., Ph.D.[1]; Chien-Kuo Chiu, P.E., Ph.D.[2]; Mahmoud Farfoura[3]; and Ismail Al-Taharwa[4]

Abstract: This study attempts to optimize the prediction accuracy of the compressive strength of high-performance concrete (HPC) by comparing data-mining methods. Modeling the dynamics of HPC, which is a highly complex composite material, is extremely challenging. Concrete compressive strength is also a highly nonlinear function of ingredients. Several studies have independently shown that concrete strength is determined not only by the water-to-cement ratio but also by additive materials. The compressive strength of HPC is a function of all concrete content, including cement, fly ash, blast-furnace slag, water, superplasticizer, age, and coarse and fine aggregate. The quantitative analyses in this study were performed by using five different data-mining methods: two machine learning models (artificial neural networks and support vector machines), one statistical model (multiple regression), and two metaclassifier models (multiple additive regression trees and bagging regression trees). The methods were developed and tested against a data set derived from 17 concrete strength test laboratories. The cross-validation of unbiased estimates of the prediction models for performance comparison purposes indicated that multiple additive regression tree (MART) was superior in prediction accuracy, training time, and aversion to overfitting. Analytical results suggested that MART-based modeling is effective for predicting the compressive strength of varying HPC age. **DOI: 10.1061/(ASCE)CP.1943-5487.0000088.** *© 2011 American Society of Civil Engineers.*

CE Database subject headings: Concrete; Compressive strength; Data collection; Artificial intelligence; Predictions.

Author keywords: High-performance concrete; Compressive strength; Data mining; Computing intelligence; Predictive techniques.

Introduction

High-performance concrete (HPC) has been used in the concrete construction industry in recent years because of its demonstrated high strength. In addition to the four basic ingredients of conventional concrete, (i.e., portland cement, fine and coarse aggregates, and water), HPC includes supplementary cementitious materials (e.g., fly ash, blast-furnace slag, and chemical admixture such as superplasticizer) (Chang et al. 1996).

HPC is used for structures subject to extreme environmental conditions, for example, marine structures, highway bridges and pavements, nuclear structures, tunnels, and precast units. Chemical admixtures reduce water content and simultaneously reduce porosity within the hydrated cement paste (Hover 1998). However, an overly low ratio of water to chemical admixture is undesirable because the effectiveness of chemical admixtures (e.g., superplasticizer) primarily depends on ambient temperature, cement chemistry, and fineness. Mineral admixtures, also called cement replacement materials, act as pozzolanic materials and fine fillers; they, therefore, increase microstructure strength and density in the hardened cement matrix (CEB-FIP 1994).

The price differential between cement and the supplementary cementitious material also yields economic benefits because portland cement is the most expensive component of a concrete mixture. Partially replacing cement with combined pozzolanic industrial by-products may also increase the efficiency of cement use and decrease the consumption of energy and the release of hazardous airborne emissions. Including concrete additives to improve technical properties of concrete (i.e., workability, durability, and strength) adds further complexity. Such additives introduce new dimensions to the modeling of HPC compressive strength. Thus, traditional modeling techniques to predict concrete behavior, particularly for compressive strength, are difficult, time-consuming, and unreliable.

Concrete mixes must comply with American Concrete Institute recommendations. Strength tests are usually performed 7–28 days after pouring the concrete. The 28-day waiting period required to perform such a test may delay the construction process, but neglecting the test would limit quality control in large construction sites. Therefore, the rapid and reliable prediction of concrete strength is essential for predesign or quality control. It would enable the adjustment of the mix proportion if the concrete does not meet the required design level, which would save time and construction costs. The early prediction of concrete strength is essential for

[1]Associate Professor of Project Management, Dept. of Construction Engineering, National Taiwan Univ. of Science and Technology, 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan (corresponding author). E-mail: jschou@mail.ntust.edu.tw

[2]Assistant Professor of Structural Engineering, Dept. of Construction Engineering, National Taiwan Univ. of Science and Technology, 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan. E-mail: ckchiu@mail.ntust.edu.tw

[3]Ph.D. Candidate, Dept. of Computer Science and Information Engineering, National Taiwan Univ. of Science and Technology, 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan. E-mail: mfarfora@rss.gov.jo

[4]Doctoral Student, Dept. of Computer Science and Information Engineering, National Taiwan Univ. of Science and Technology, 43, Sec. 4, Keelung Rd., Taipei 106, Taiwan. E-mail: tahrawee@yahoo.com

Note. This manuscript was submitted on April 15, 2010; approved on July 22, 2010; published online on August 26, 2010. Discussion period open until October 1, 2011; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Computing in Civil Engineering***, Vol. 25, No. 3, May 1, 2011. ©ASCE, ISSN 0887-3801/2011/3-242–253/$25.00.

estimating the time needed for concrete form removal, project scheduling, and quality control.

Because the relationships among components and concrete properties are highly nonlinear, establishing a comprehensive mathematical model is problematic. Therefore, traditional models of concrete properties are inadequate for meeting the design requirements of HPC compressive strength. The primary goal of a modeling system that uses large experimental datalike concrete mixes is to properly reflect the very nature of physical phenomena such as concrete compressive strength. In such cases, most modeling techniques use experimental data to derive mathematical models in the form of analytical linear or nonlinear functions.

Several studies independently have shown that concrete strength is determined not only by the water to cement ratio, but also by other content (Oluokun 1994). The empirical equations presented in current codes and standards for estimating compressive strength are based on tests of concrete without supplementary cementitious materials. The validity of these relationships for concrete with supplementary cementitious materials (e.g., fly ash and blast-furnace slag) is, therefore, questionable. Understanding the relationship between concrete composition and strength is essential for optimizing concrete mixtures (Popovics 1990).

Because accurately predicting concrete strength is a critical issue in concrete construction, this study developed novel intelligent techniques for accurately predicting the concrete compressive strength of varying age (i.e., 3, 7, 14, 28, 56, 91, 180, 270, and 365 days). Similar work in the literature employed artificial neural networks (ANNs) and its variations (Fazel Zarandi et al. 2008; Kasperkiewicz and Dubrawski 1995; Topçu and SarIdemir 2008; Yeh 1998; Yeh and Lien 2009) and the traditional linear regression (LR) technique (Yeh 1998) to predict compressive strength in HPC. None exploit other predictive techniques, namely support vector machine (SVM) or metaclassifiers such as multiple additive Regression trees (MART) and bagging regression trees (BRT). This study, therefore, fills this gap.

This paper is organized as follows. The "Related Works" section reviews related literature. The "Methods and Experimental Settings" section describes the results of a case study that uses prediction techniques and a cross-validation algorithm with a concrete strength test data set. The "Results and Discussions" section reports and discusses the prediction results for the five models, and the "Conclusions" section concludes this study with a summary and suggestions.

Related Works

Numerous studies have proposed approaches for modeling concrete compressive strength. Oh et al. (1999) applied ANNs to optimize the proportion of four concrete ingredients (water, cement, fine aggregate, and coarse aggregate). They introduced a tool for minimizing uncertainty and errors in proportioning concrete mixes, which is a complicated, time-consuming, and uncertain task (Oh et al. 1999). Mostofi and Samaee (1995) employed multilayer perceptron ANNs to estimate HPC compressive strength (Mostofi and Samaee 1995). Yeh (1998) modeled HPC strength by using artificial neural networks as a function of cement, fly ash, blast-furnace slag, water, superplasticizer, coarse aggregate, fine aggregate, and age of testing and obtained promising results (Yeh 1998).

Kasperkiewicz and Dubrawski (1995) applied fuzzy-ARTMAP ANNs to predict the 28-day compressive strength of HPC mixes. The composition of HPC was simplified as a mixture of six components (cement, silica, superplasticizer, water, and fine and coarse aggregate). However, the performance measure of their model was insufficient for further generalization because the correlation coefficient ($r = 0.784$) was moderate, and the data set used in training the model was only 200 instances (Kasperkiewicz and Dubrawski 1995). Fazel Zarandi et al. (2008) developed a fuzzy polynomial neural network (FPNN) that combined fuzzy neural networks (FNNs) and polynomial neural networks (PNNs). Six different FPNN architectures were constructed. Each architecture had six input parameters (concrete ingredients) and one output parameter (28-day compressive strength of the mix-design). The best correlation obtained by the FPNN was 0.8209, which was clearly an improvement in complex, high-order modeling by ANNs (Fazel Zarandi et al. 2008).

Another research direction applied by Yeh and Lien (2009) was a genetic operation tree (GOT), which combines an operation tree and a genetic algorithm to automatically produce self-organized formulas for predicting the compressive strength of HPC. Comparison results showed that GOT ($R^2 = 0.8669$) obtained formulas that were more accurate than nonlinear regression formulas but less accurate than neural network models ($R^2 = 0.9338$).

Rather than modeling historical experiment data, Trtnik et al. (2009) used an ultrasonic pulse velocity technique, which is among the most common nondestructive techniques for assessing concrete properties (Trtnik et al. 2009). This testing method employs a portable ultrasonic nondestructive digital indicating tester (PUNDIT) to generate an ultrasonic pulse, which is transmitted through the concrete and received at the opposite surface. Upon receiving the pulse, the instrument amplifies it and measures the time it required to travel through the concrete. Nevertheless, the use of this method to evaluate concrete compressive strength is problematic because ultrasonic pulse velocity is affected by factors that may not affect compressive strength similarly or to the same extent.

Gupta et al. (2006) presented a neural-fuzzy inference system for predicting the compressive strength of HPC (Gupta et al. 2006). The system parameters included concrete mix-design, specimen size and shape, curing technique and period, and the environmental conditions maximum temperature, relative humidity, and wind velocity. A data driven rule-based expert system was developed to overcome the bottlenecks in knowledge acquisition. Although this reduced predictive accuracy ($R^2 = 0.76$), the system enabled the easy update of the knowledge base at any stage to enhance the neural-expert interface.

Generally, previous studies applied similar ANN techniques with only minor modifications and some traditional regression techniques. To this end, reliable, applicable, and practicable models are still needed to predict HPC compressive strength. Models must not only meet modeling requirements, they must also be sufficiently robust to model involved uncertainties and easy to manipulate. The aim of this study was to exploit prevailing data-mining predictive techniques to model HPC compressive strength as a function of its primary ingredients and to improve strength prediction performance. The following section describes the proposed data-mining predictive techniques in detail.

Methods and Experimental Settings

Data Description and Preparation

The experimental data set was obtained from a University of California, Irvine (UCI), repository of data (Yeh 1998). A final set of 1,030 samples of ordinary portland cement containing various additives and cured under normal conditions was evaluated from numerous university research labs (Chang et al. 1996; Chang 1997; Chung 1995; Giaccio et al. 1992; Gjorv et al. 1990; Hwang

J. Comput. Civ. Eng., 2011, 25(3): 242-253

Table 1. HPC Attributes

Attribute	Unit	Minimum	Maximum	Average	Standard deviation
Cement	kg/m^3	102.0	540.0	281.168	104.506
Blast-furnace slag	kg/m^3	11.0	359.4	107.277	61.884
Fly ash	kg/m^3	24.5	200.1	83.862	39.989
Water	kg/m^3	121.8	247.0	181.567	24.354
Superplasticizer	kg/m^3	1.7	32.2	8.486	4.037
Coarse aggregate	kg/m^3	801.0	1,145.0	972.919	77.754
Fine aggregate	kg/m^3	594.0	992.6	773.580	80.176
Age of testing	Day	1.0	365.0	45.662	63.170
Concrete compressive strength	MPa	2.3	82.6	35.818	16.706

1991; Hwang 1966; Langley et al. 1989; Lee 1994; Lessard et al. 1993; Lin 1994; Mo 1995). All tests were performed on 15 cm cylindrical specimens of concrete prepared by using standard procedures. Table 1 shows the experimental data set of nine HPC attributes used in this study.

Data sets described in the literature often contain unexpected inaccuracies. For example, the class of fly ash may not be indicated. Another problem is that the superplasticizers may be produced by different manufacturers and have different chemical compositions (Kasperkiewicz and Dubrawski 1995). Concrete compressive strength not only is determined by the water to concrete ratio, but also by the other materials used in the mix. Concrete contains five ingredients other than cement and water. The multiple ingredients, in addition to the nonlinearity of concrete structures, complicate the computation of compressive strength. The following predictive techniques are proposed for applying these complex inputs when modeling the compressive strength of HPC.

Predicted Data-Mining Techniques

The Waikato Environment for Knowledge Analysis (WEKA) suite (Ian and Eibe 1999) and SPSS Clementine were used to test various training techniques and algorithms by using the experimental data set. Additionally, WEKA provided various filters for preprocessing, model evaluation, visualization, and postprocessing. Comprehensive experiments were performed to determine the best mining technique for predictive accuracy, learning time, and overall performance.

Artificial Neural Network

An artificial neural network is a computational model that attempts to simulate the structure and/or functional aspects of biological neural networks. Various ANN applications can be categorized as classification or pattern recognition or prediction and modeling. ANNs are widely used in many industrial areas, including process engineering, control and monitoring, technical diagnosis and nondestructive testing, power systems, robotics, transportation, telecommunications, remote sensing, banking, finance and insurance, forecasting, document processing, and construction engineering. The use of ANNs specifically for predicting concrete compressive strength has been studied intensively (Fazel Zarandi et al. 2008; Jang 1993; Kasperkiewicz and Dubrawski 1995; Lee et al. 2009; Mostofi and Samaee 1995; Oh et al. 1999; Seyhan et al. 2005; Topçu and SarIdemir 2008; Yeh 1998, 1999). Researchers have also explored the use of ANNs to construct concrete compressive strength models that are more accurate than regression models (Yeh 1998).

The advantages of ANNs are the unrestricted number of inputs and outputs and the clearly defined number of hidden layers and hidden neurons. The primary drawback of ANNs are the considerable time needed to determine the number of layers and hidden neurons, which requires repetitive trial and error-tuning processes. The solution proposed by Hegazy et al. (1994) is to use only one hidden layer to generate arbitrary mapping between inputs and outputs, and the number of neurons in the hidden layer is $0{:}75m$; m, or $2m+1$, where m = number of input neurons (Hegazy et al. 1994). However, the current study indicated that ANNs with three hidden layers are more accurate.

The five training algorithms commonly used in ANNs are Levenberg-Marquardt, gradient descent, gradient descent with

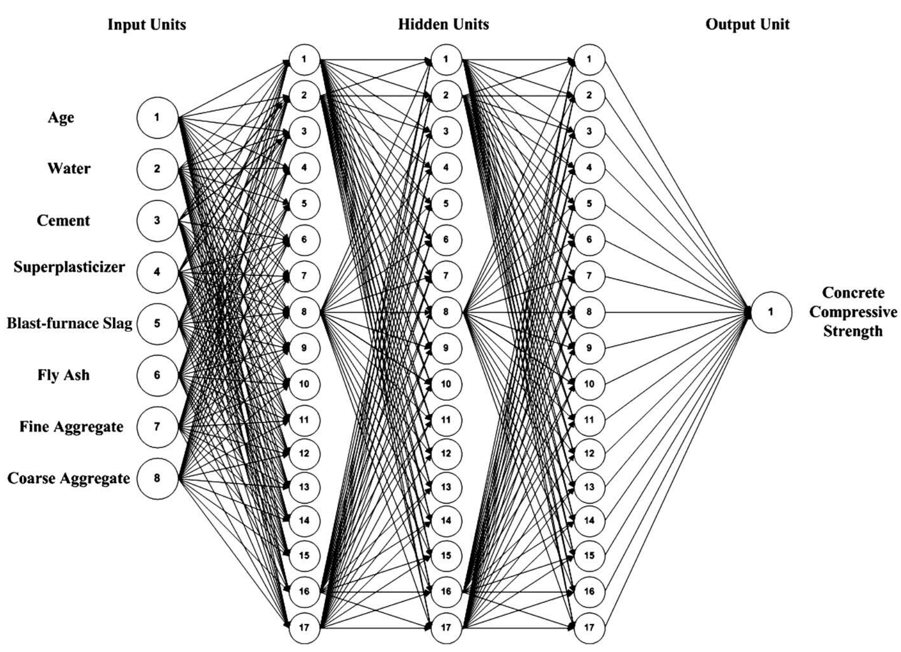

Fig. 1. Graphical representation of the proposed ANNs model

momentum, gradient descent momentum and adaptive learning rate, and gradient descent with adaptive learning rate. The back-propagation algorithm is widely used to adjust connection weights and bias values training. Fig. 1 shows the typical network architecture.

The network parameters tested in the proposed model included the following: the number of hidden layers was 1, 2, and 3; the number of hidden neurons was 8, 12, and 17; the learning rate was 0.1, 0.3, 1.0, and 3.0; the momentum factor was 0.0, 0.25, 0.5, and 0.75; and the training time was 3,000; 5,000; and 10,000 (each cycle covered the entire data set available for training). Integrated performance testing indicated that the best network parameters were as follows: the number of input neurons was 8; the number of hidden layers was 3; the number of hidden neurons was 17; the number of output neurons was 1; the learning rate was 1.0; the momentum factor was 0.3; and the training time was 10,000.

Multiple Regression

Multiple regression (MR) models depict the relationship among two or more variables. The computational problem addressed by multiple regression is fitting a plane to an n-dimensional space where n = number of independent variables to a number of points. For a system with n inputs (independent variables) and one output (dependent variable), Y, the general least square (or linear regression) problem is to determine unknown parameters, b_i, of the linear model, as shown in Eq. (1):

$$Y = C + b_1 * X_1 + b_2 * X_2 + b_3 * X_3 + \ldots + b_{n-1} * X_{n-1} + b_n * X_n \tag{1}$$

In the proposed regression model, Y represents concrete compressive strength, and $b_1, b_2, \ldots, b_8$ are regression coefficients. The X_i values represent cement, blast-furnace slag, fly ash, water, superplasticizer, coarse aggregate, fine aggregate, and age; and C is the estimated constant. Regression analysis estimates the unbiased values of the regression coefficients $b_1, b_2, \ldots, b_8$ against the training data set.

In the literature, MR is commonly used for modeling HPC compressive strength (Yeh 1998). However, its primary drawback is its inability to cope with highly nonlinear HPC components. Therefore, its modeling performance is reportedly poor. However, because of its simplicity, this study investigated the applicability of MR by using SPSS Clementine for comparison purposes.

Support Vector Machine

The support vector machine (SVM) was first introduced by Vapnik (1995) and his colleagues at AT&T Bell Laboratories (Vapnik 1995, 1998). The SVM has been used in many civil engineering applications (Dibike et al. 2001; Pal and Mather 2003) but rarely for predicting HPC compressive strength. In this study, epsilon support vector regression (ε-SVR) (Smola and Schölkopf 2004), a variation of SVM for function estimation, was used to construct the HPC input-output model.

In SVM regression, the input is first mapped onto an m-dimensional feature space by using fixed (i.e., nonlinear) mapping. A linear model is then constructed in this feature space (Fig. 2). The linear model in the feature space, $f(\mathrm{x}, \omega)$, can be expressed in mathematical notation as

$$f(\mathbf{x}, \omega) = \sum_{j=1}^{m} \omega_j g_j(\mathbf{x}) + b \tag{2}$$

where $g_j(\mathbf{x})$, $j = 1, \ldots, m$ = set of nonlinear transformations; and b = "bias" term. Data are often assumed to be zero mean and

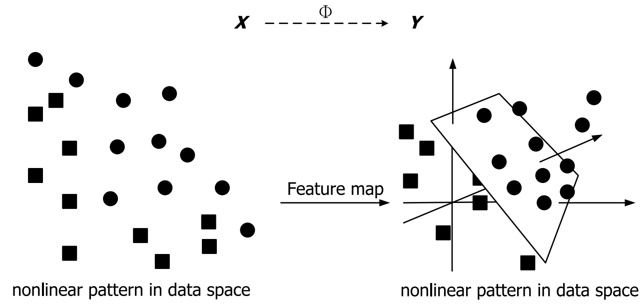

Fig. 2. Nonlinear-to-linear model mapping

can, therefore, be obtained in the preprocessing stage, so the bias term is dropped. Estimation quality is measured by the loss function, $L[y, f(\mathrm{x}, \omega)]$. The SVM regression employs the following ε-insensitive loss function, which was proposed by Vapnik (1998):

$$L_\varepsilon[y, f(\mathbf{x}, \omega)] = \begin{cases} 0 & \text{if } |y - f(\mathbf{x}, \omega)| \le \varepsilon \\ |y - f(x, \omega)| & \text{otherwise} \end{cases} \tag{3}$$

Linear regression is performed in SVM regression in the high-dimension feature space by using ε-insensitive loss and reduces model complexity by minimizing $||\omega||^2$. This procedure can be demonstrated by introducing the nonnegative slack variables ξ_i, ξ_i^*, where $i = 1, \ldots, n$ to identify training samples that deviate from the ε-insensitive zone. Thus, SVM regression is formulated as a minimization of the following function:

$$\min \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$
$$\text{subject to} \begin{cases} y_i - f(\mathbf{x}_i, \omega) \le \varepsilon + \xi_i^* \\ f(\mathbf{x}_i, \omega) - y_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, i = 1, \ldots, n \end{cases} \tag{4}$$

This optimization problem can be transformed into the dual problem, which is solved by

$$f(\mathbf{x}) = \sum_{i=1}^{n_{SV}} (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) \quad \text{subject to}$$
$$0 \le \alpha_i^* \le C, 0 \le \alpha_i \le C \tag{5}$$

where n_{SV} = number of support vectors (SVs) and the kernel function

$$K(\mathbf{x}, \mathbf{x}_i) = \sum_{j=1}^{m} g_j(\mathbf{x}) g_j(\mathbf{x}_i) \tag{6}$$

Clearly, SVM generalization performance (i.e., estimation accuracy) requires correct settings for the metaparameters C and ε and the kernel parameters, which are related to the kernel type [e.g., the σ-parameter in a radial basis function (RBF) kernel and the couple (σ, γ) in the kernel with moderate decreasing (KMOD)]; default values of kernel parameters clearly depend on their type and the implementing software. Table 2 shows the most common kernel types and their corresponding equations. Existing software implementations of SVM regression usually treat SVM metaparameters as user-defined inputs (Smola and Schölkopf 2004; Vapnik 1995). Selecting specific kernel types and kernel function parameters usually requires application-domain knowledge and should reflect the distribution of training data inputs.

Table 2. SVM Kernel Function Types

Kernel	Equation
Linear	$K(x,y) = x \cdot y$
Sigmoid	$K(x,y) = \tanh(ax.y + b)$
Polynomial	$K(x,y) = (1 + x \cdot y)^d$
KMOD	$K(x,y) = a\left[\exp\left(\frac{\gamma}{\lVert x-y\rVert^2+\sigma^2}\right) - 1\right]$
RBF	$K(x,y) = \exp(-a\lVert x-y\rVert^2)$
Exponential RBF	$K(x,y) = \exp(-a\lVert x-y\rVert)$

The LIBSVM software application was used to apply SVR (Chang and Lin 2001). The LIBSVM, an integrated software tool for support vector classification, regression, and distribution estimation, was run in the WEKA application. The data set was used in several experiments to obtain suitable user-defined parameters. The best values for the user-defined parameters were then used to predict concrete strength. The experiments indicated that the best parameter configuration for this technique was: cost, $C = 50$; $\varepsilon = 5$; gamma $= 4$; the kernel function is RBF—its equation is $K(x,y) = \exp(-a\lVert x-y\rVert^2)$; and loss $= 2$.

Multiple Additive Regression Trees

Metalearning is the supervised learning from the information generated by the initial or base classifiers (Fan et al. 1996). The idea is to generate a system that provides base classifier functionality and increases precision by improving the form in which they correlate among themselves. This efficiently reduces incorrect predictions (Chan and Stolfo 1997). Predictions are obtained by adding the predictions of each classifier. Reducing the shrinkage (i.e., the learning rate) parameter helps prevent overfitting and provides a smoothing effect but increases the learning time (Fig. 3).

One of the most powerful metaclassifiers is multiple additive regression trees (MART), which is an important advance in data mining because it extends and improves the conventional classification and regression trees model by using stochastic gradient boosting (Friedman 2001). This work applied MART modeling to predict HPC compressive strength. However, practical applications of regression trees are limited by their inaccuracy. Although they sometimes perform adequately, tree-based models rarely obtain the best possible solution in any given application. This vital defect of regression trees can be corrected by incorporating so-called "boosting."

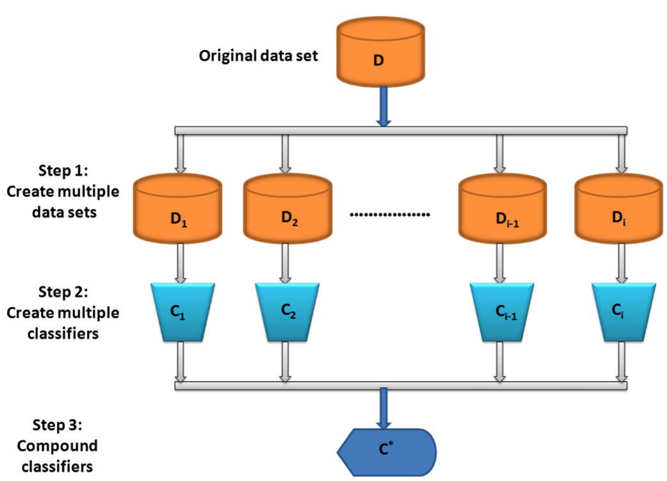

Fig. 3. Metaclassification

Boosting tree-based models can dramatically increase their accuracy by fitting a series of models, each of which has a lower error rate, and then combining them into an ensemble with better performance (Duffy and Helmbold 2002). This innovation is the primary advantage of MART because MART models combine boosting with regression trees as their primary component. Essentially, MART inherits nearly all advantages of regression tree-based modeling while overcoming their primary defect, which is inaccuracy.

The present study employed MART by using decision stump (DS) as a base learner. A DS is a one-level regression tree that classifies instances by sorting them based on feature values (Iba and Langley 1992). Each node in a decision stump represents a feature in an instance to be classified, and each branch represents a potential node value. Instances are classified starting at the root node and are sorted based on their feature values. At worst, a decision stump obtains the most likely solution at the baseline and may obtain a better solution if the selected feature is particularly informative. On the basis of several experiments in this study, the best configuration parameters for the MART model were the number of iterations were 600 and the shrinking was 0.09.

Bagging Regression Trees

Bootstrap aggregating, or "bagging" (Breiman 1996), is a "bootstrap" ensemble method that improves prediction by using different classification methods and regression methods to minimize variance in the prediction process. The underlying concept of bagging is to create individual regression or classification models by using a randomly redistributed training set to train a single learning algorithm. The training set for each regression model is generated by randomly drawing, with replacement, N instances where N = size of the original training set. 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. A more sophisticated version of bagging is described in Breiman (2001).

The regression setting in this study had pairs (X_i, Y_i) $(i = 1, \ldots, n)$, where $X_i \epsilon R^d$ denoted the d-dimensional predictor variable and the response $Y_i \epsilon R$ (regression). The target function of interest was usually $E[Y|X = x]$. The function estimator, which was obtained by a given base procedure, was

$$\hat{g}(.) = h_n[(X_1, Y_1), \ldots, (X_i, Y_i)](.)\colon R^d \to R \tag{7}$$

where $h_n(.)$ defines the estimator as a function of the data.

In this study, BRT was applied by using the base fast decision tree learner (i.e., building and using a fast decision tree learner). The primary parameters for the fast decision tree learner in this study were the following: number of folds (i.e., the amount of data used for pruning. One fold is used for pruning, the rest for growing the rules); minNum (i.e., the minimum total weight of the instances in a leaf); and seed. The bagging parameters were bagSizePercent (i.e., the size of each bag as a percentage of the training set size); numIterations (i.e., the number of iterations to be performed); and seed (i.e., the random number of seeds to be used). In this case, the values for these parameters were 5, 2, 1, 100, 50, and 1, respectively. The fast decision tree learner only sorted values for numeric attributes once. Missing values were addressed by splitting the corresponding instances into pieces.

Performance Measures

The following four performance measures were used to evaluate the proposed artificial intelligence techniques.

J. Comput. Civ. Eng., 2011, 25(3): 242-253

Linear Correlation Coefficient

The linear 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

$$R = \frac{n\sum y.y' - (\sum y)(\sum y')}{\sqrt{n(\sum y^2) - (\sum y)^2}\sqrt{n(\sum y'^2) - (\sum y')^2}} \tag{8}$$

where y = actual value; y' = predicted value; and n = number of data samples.

Coefficient of Determination, R-Square

The R-square coefficient, R^2 is a measure of how well the independent variables considered account for the measured dependent variable. The higher the R-square value, the better the predictive power.

Root Mean Squared Error

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:

$$\mathrm{RMSE} = \sqrt{\frac{\Sigma(y' - y)^2}{n}} \tag{9}$$

Mean Absolute Percentage Error

The mean absolute percentage error (MAPE) is a statistical measure of predictive accuracy. It usually expresses accuracy as a percentage. The MAPE is commonly used in quantitative forecasting methods because it indicates the relative overall fit (i.e., the goodness-of-fit). The MAPE is given by the following equation:

$$\mathrm{MAPE} = \frac{1}{n}\Sigma_{i=1}^{n}\left|\frac{y - y'}{y}\right| \tag{10}$$

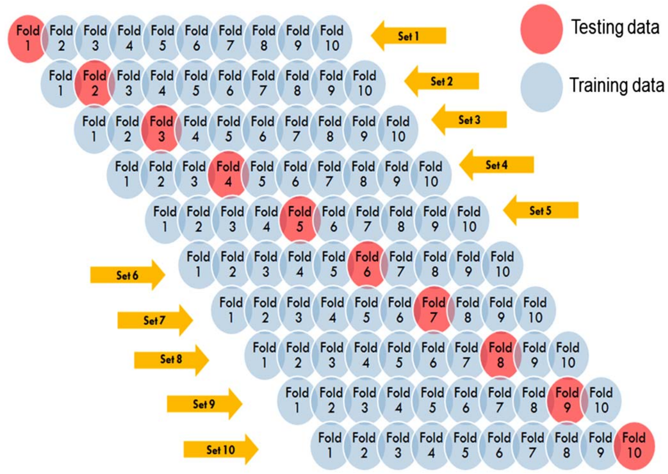

Fig. 4. 10-fold cross-validation procedure

Experimental Setting

To minimize bias associated with the random sampling of the training and holdout data samples when comparing the predictive accuracy of two or more methods, researchers often use k-fold cross-validation. Because cross-validation requires the random assignment of individual cases into distinct folds, a common practice is stratifying the folds themselves. In stratified k-fold cross-validation, the folds are created so that they contain approximately similar proportions of predictor labels (i.e., responses) as in the original data set. Empirical studies show that, compared to regular k-fold cross-validation, stratified cross-validation tends to generate results with lower bias.

Kohavi (1995) showed that 10 folds were optimal (i.e., obtained the minimal time needed to perform the test with acceptable bias and variance associated with the validation process) (Kohavi 1995). Thus, to assess model performance, a stratified 10-fold cross-validation approach was used. The entire data set was divided into

Table 3. Descriptive Statistics for the Proposed Five Predictive Techniques over the 10 Folds

Performance measure	Predictive technique	Minimum	Average	Maximum	Best fold
R^2	ANNs	0.8630	0.9091	0.9428	5
	MR	0.4720	0.6112	0.6906	3
	SVM	0.8420	0.8858	0.9197	5
	MART	0.8696	0.9108	0.9430	10
	BRT	0.8551	0.8904	0.9160	10
RMSE (MPa)	ANNs	3.7673	5.0303	7.0364	2
	MR	9.4194	10.4288	11.6391	2
	SVM	4.2315	5.6192	6.7248	2
	MART	3.7972	4.9489	6.1159	2
	BRT	5.1470	5.5720	6.0195	9
MAPE (%)	ANNs	9.6485	10.9032	11.6444	2
	MR	29.2059	31.6518	36.6473	9
	SVM	10.6607	12.7726	14.9052	9
	MART	9.9683	13.8855	15.6572	2
	BRT	12.6467	14.1833	16.4396	8
Training time (s)	ANNs	301.0000	394.900	480.0000	2
	MR	0.0100	0.0250	0.0900	2-4, 7, 10
	SVM	0.9800	2.1370	3.9800	3
	MART	3.6400	5.5800	9.5700	8
	BRT	0.9500	2.0230	3.5800	4

J. Comput. Civ. Eng., 2011, 25(3): 242-253

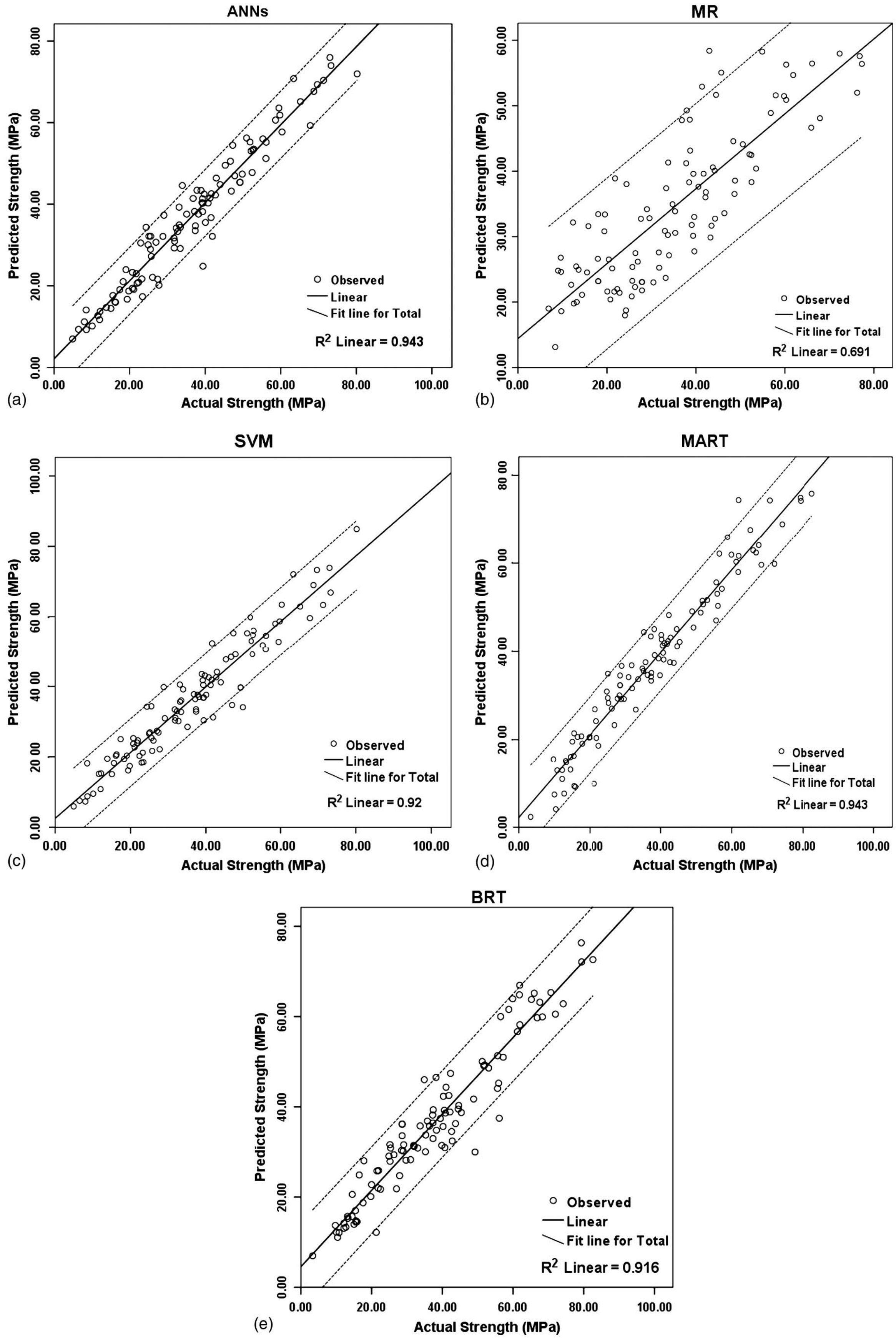

Fig. 5. Actual versus predicted HPC compressive strength

Table 4. Performance Comparison of the Predictive Techniques

Predictive technique	Average R^2	Average RMSE (MPa)	Average MAPE (%)	Average training time (s)	RI
ANNs	0.9091	5.0302	10.9032	394.9000	0.7449
MR	0.6112	10.4287	31.6517	0.02500	0.2500
SVM	0.8858	5.6192	12.7726	2.1370	0.8844
MART	0.9108	4.9489	13.8855	5.5800	0.9606
BRT	0.8904	5.5720	14.1832	2.0230	0.9138

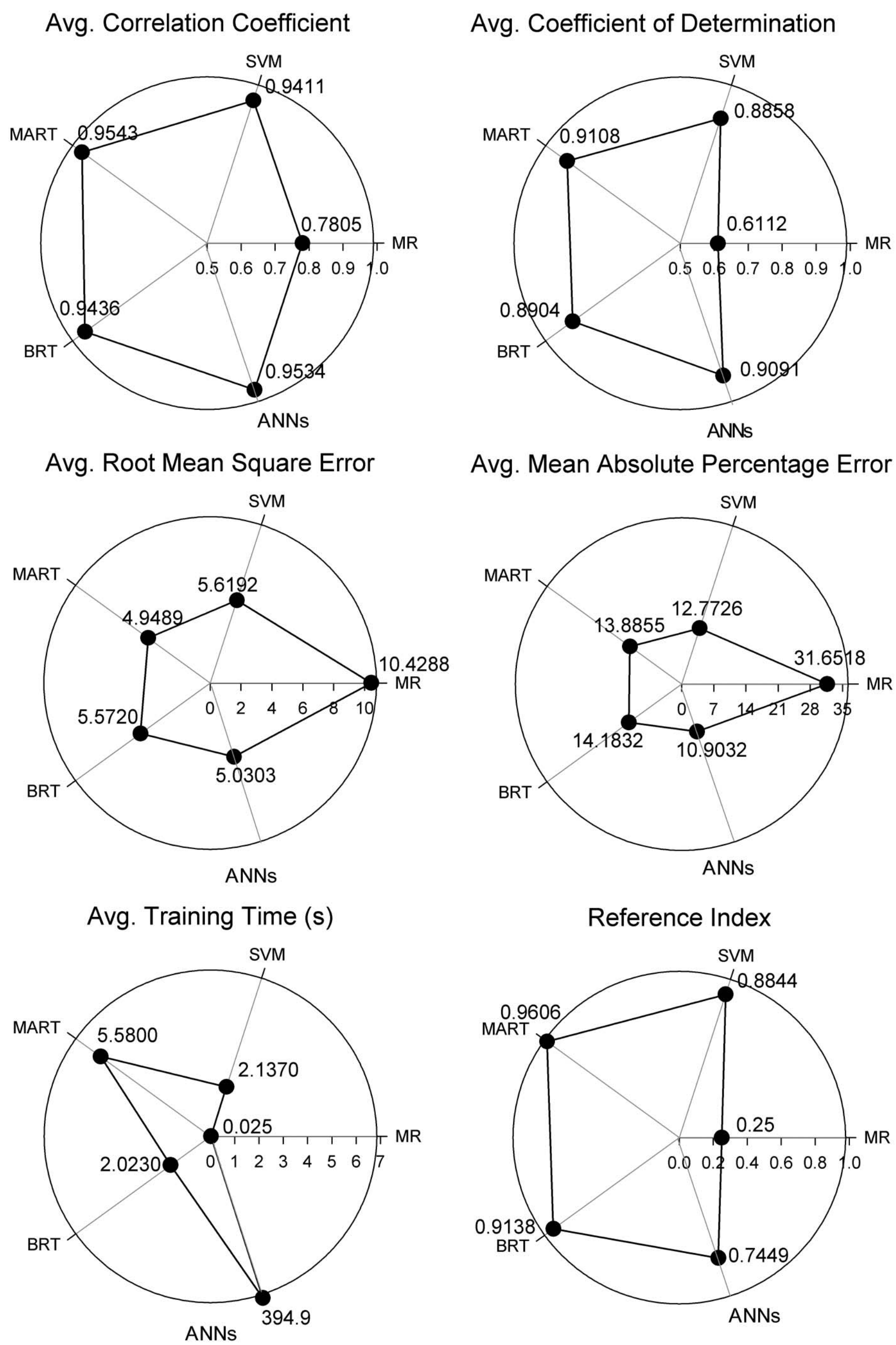

Fig. 6. Average results for performance measures of the proposed predictive techniques (ANNs, MR, SVM, MART, BRT) with 10-fold cross-validation

J. Comput. Civ. Eng., 2011, 25(3): 242-253

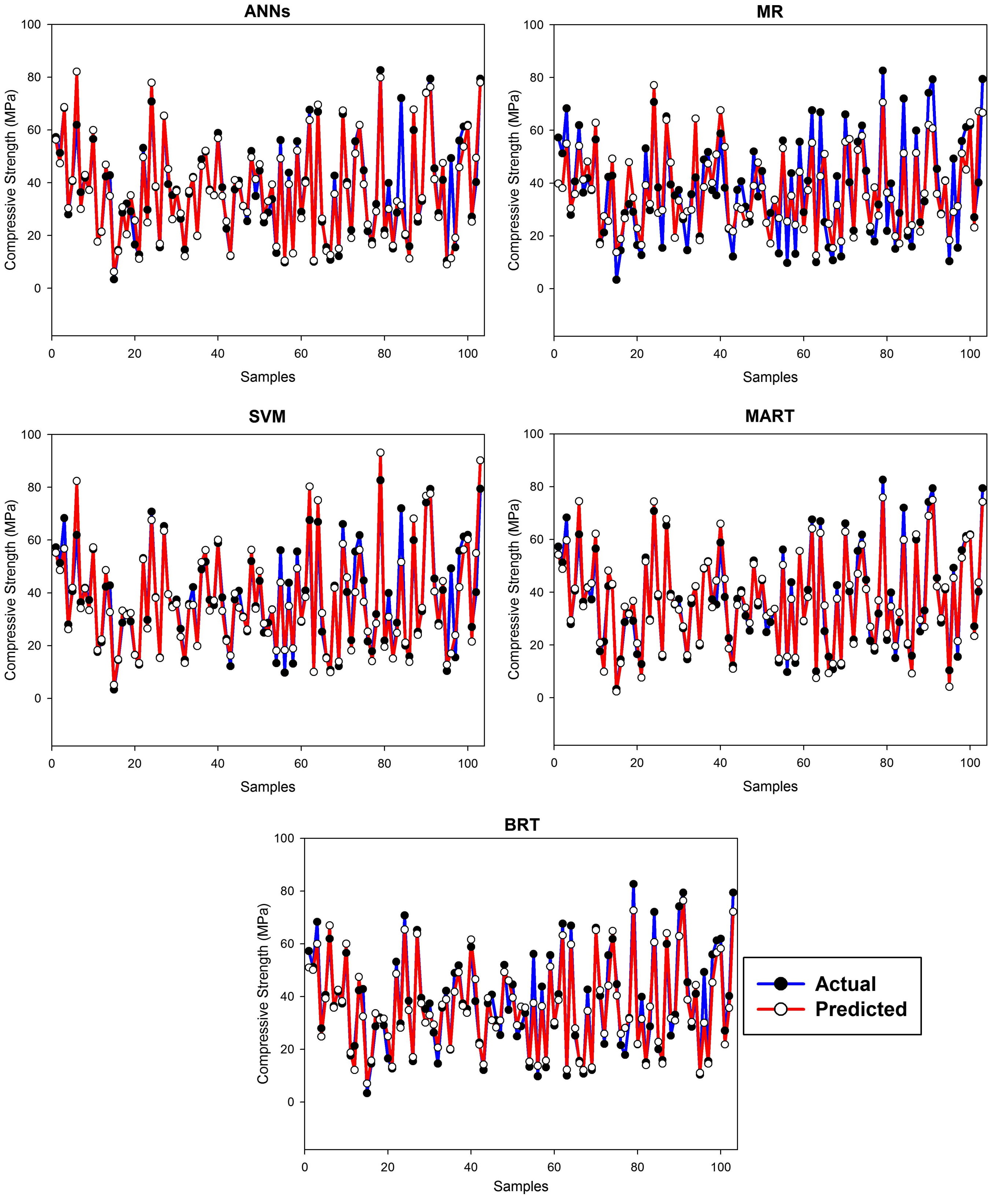

Fig. 7. Actual and predicted concrete compressive strength for best testing fold

10 mutually exclusive subsets or folds with class distributions approximating those of the original data set (i.e., stratified). The subset extraction procedure was performed in the following five steps:

1. Randomize the data set;
2. Extract one-tenth of the original data set size from the randomized data set (single fold);
3. Remove the extracted data from the original data set;
4. Repeat steps 1–3 eight times; and
5. Assign the remaining portion of the data set to the last fold (10th fold).

This procedure was first used to obtain 10 distinct folds. Each fold was used once to test the performance of the five aforementioned data-mining techniques, and the remaining nine folds were used for training, which obtained 10 independent performance estimates (Fig. 4). Calculations for the cross-validation estimate of overall accuracy was derived by simply averaging the k individual accuracy measures for cross-validation accuracy (CVA), where k = number of folds used; and A_i = accuracy measure of each fold:

$$\mathrm{CVA} = \Sigma_1^k \frac{A_i}{k} \tag{11}$$

Results and Discussions

Experimental Results

The predictive data-mining techniques proposed in this study were evaluated by using the accuracy measures discussed previously (i.e., the coefficient of determination R^2, the root mean squared error RMSE, the mean absolute percentage error MAPE, and the training time). The results were achieved by using a 10-fold cross-validation for each model and were based on the average results [Eq. (11)] obtained for the test data set (10 folds) for each set. The developed models were trained and tested by the cross-validation method to ensure that all the data set instances were applied in both the training and testing phases. Because of these characteristics, the cross-validation method provided better validation capabilities than the considered models. The comparison of maximum and minimum values in Table 3 indicates that the deviation in the results obtained by different performance measures for each of the 10 folds was very limited, which is the best feature of the stratified cross-validation technique.

Fig. 5 shows actual values against the predicted values of HPC compressive strength. The best results obtained by MART, ANNs, SVM, and BRT obtained a better fit to a straight line than MR did, which indicates that these techniques were more accurate for predicting HPC compressive strength. These findings also show tolerant and reliable prediction capabilities of generalization for MART, ANNs, BRT, and SVM predictive techniques. Table 4 summarizes the results of all the performance measures for the proposed five data-mining predictive techniques. Additionally, R^2, RMSE, MAPE, and training time are normalized into one reference index (RI) to enable an overall comparison of the predictive techniques.

The best model for determining R^2 was the MART model (0.9543), which had an average training time of 5.580 s. The second best model was ANNs, which had $R^2 = 0.9091$ but a poor average training time (394.9) because of the natural inheritance from the essences of the back-propagation scheme of ANNs. The BRT and SVM obtained results very similar to those of ANNs and MART for accuracy R^2 and RMSE but had better training time (2.023 and 2.137, respectively). The MR model exhibited the worst predictive capabilities with $R^2 = 0.6112$, indicating that this model cannot reliably approximate the compressive strength of HPC. Fig. 6 depicts the performance measures described in Table 4.

Notably, Table 4 shows the direct relationship between R^2 and RMSE. The best model for minimizing RMSE was MART (4.9489), and the worst was MR (10.4287). MAPE was used to assess the average prediction ability, which is expressed in generic percentage terms. It measures the reliability of the predictive

Table 5. Comparison with Previous Works

Previous work	Description	Technique	Sample size	Performance measure (R^2)
Yeh 1998	Modeled the strength of HPC as a function of cement, fly ash, blast-furnace slag, water, superplasticizer, coarse aggregate, fine aggregate, and age of testing. The output was the predicted HPC compressive strength at several ages.	ANNs, liner regression	727	0.914 (average) 0.574 (average)
Kasperkiewicz and Dubrawski 1995	The input was a mixture of six components: cement, silica, superplasticizer, water, and fine and coarse aggregate. The output was the average of 28-day HPC compressive strength.	Fuzzy-adaptive resonance theory-MAP ANNs	340	0.615
Fazel Zarandi et al. 2008	The input parameters were the weight of coarse aggregate, fine aggregate, super plasticizer, silica fume, water, and cement. The output was the average of 28-day HPC compressive strength.	Fuzzy polynomial neural networks	458	0.8209
Yeh and Lien 2009	The model inputs were cement, fly ash, blast-furnace slag, water, superplasticizer, coarse aggregate, fine aggregate, and age of testing. The output was the average of 28-day HPC compressive strength.	Genetic operation trees, ANNs	1,196	0.8669 0.9338
Gupta et al. 2006	Input parameters were concrete mix-design, size and shape of specimen, curing technique and period, and the environmental conditions (maximum temperature, relative humidity, and wind velocity). The output was the predicted HPC compressive strength at several ages.	Neural-fuzzy inference system	864	0.760

J. Comput. Civ. Eng., 2011, 25(3): 242-253

techniques over the unseen data. MAPE was inconsistent with the other performance measures (i.e., R^2 and RMSE); the best results were obtained by the ANNs model (MAPE = 10.9032). The MAPE of SVM (12.7726) was also superior to the MART (13.8855). This finding was clearly demonstrated by the plotted relationships between the actual and the predicted HPC compressive strength obtained by the proposed predictive techniques (Fig. 5).

The normalized RI provided a general measure by combining the four performance measures (i.e., R^2, RMSE, MAPE, and training time). On the basis of the RI, the best models in order of performance quality were MART, BRT, and SVM. The ANNs were affected by the training time factor, which degraded its performance.

Fig. 7 shows the best data fit obtained by the proposed predictive techniques for each fold. This figure compares the results of the proposed model with actual laboratory data. The horizontal axis represents the index of instances in the testing data, and the vertical axis represents the compressive strength in MPa of the samples measured. The figure shows that, except for the MR model, which had the worst data fit, the other four models obtained a very good fit to the original source data. Moreover, the MART had the most agreement. The ANNs and SVM also revealed very good agreement. The ANNs clearly obtained similar results for most instances of the data except for a few predicted instances with very large deviations (i.e., the worst three predicted instances are located between the 80th and the 100th instances, as shown in the ANNs plot), and SVM obtained similar results with small deviations. These findings can be interpreted based on the MAPE results, which indicate the reliability of model prediction.

Comparison with Previous Work

Table 5 summarizes the primary previous works in HPC compressive strength prediction. Notably, R^2 varied from 0.574 to 0.9338, which is an extremely large deviation in the problem of predicting the HPC compressive strength. Our investigation achieved good results for R^2 (0.9108), which approximated those reported for ANNs. Unlike previous works, this study employed *k*-fold cross-validation, which ensured good generalization capability. This study also standardized the work in this field by discovering hidden relationships and knowledge from a complete data set.

Conclusions

This study developed a data-mining approach and performance measures to predict compressive strength and assess the prediction reliability for HPC. Specifically, five popular data-mining methods were used: two derived from machine learning (ANNs and SVM), one from statistics (MR), and two metaclassifier models (MART and BRT). The experimental data set was acquired from a UCI machine learning repository of a 1,030-instance data set. In this investigation, the proposed predictive techniques were applied to the prepared data by using nine of the 10-folds for training the models, and the 10th fold for testing. This procedure was repeated 10 times with a distinct fold as test data in each experiment.

The proposed approaches were compared for performance outcomes by using four different performance measures (R^2, RMSE, MAPE, and training time) to obtain a comprehensive comparison of the applied predictive techniques. The findings showed that MART achieved the best accuracy for R^2 and RMSE. The ANNs and SVM obtained the best prediction power for future unseen data based on the results of MAPE, and ANNs performed the worst in training time. In some cases, the ANN training time was as high as 100 times that required for other models. Overall, the MR performed best in the training time needed to learn the model but performed the worst in accuracy.

The performance of models proposed in previous works was compared for the same performance measures. Compared with other models, the proposed model has a superior generalization capability because of the use of 10-fold cross-validation. A new RI was used as an index of accuracy (R^2, RMSE, MAPE) and training time. According to the RI, the best models, in order of performance quality, are MART, BRT, SVM, ANNs, and MR.

This study establishes that a predictive model must consider not only R^2 and RMSE, but also MAPE and training time. Predicting HPC compressive strength is a nonlinear problem that cannot be solved by traditional MR techniques. Analytical results indicate that MART is the most reliable model for high-performance concrete compressive strength in predictive accuracy, speed, ease of use, and interpretability.

Acknowledgments

The writers would like to thank UC Irvine Machine Learning Repository (http://archive.ics.uci.edu/ml/) and Professor I-Cheng Yeh for sharing the experimental data set.

References

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

Breiman, L. (2001). "Using iterated bagging to debias regressions." *Mach. Learn.*, 45(3), 261–277.

Chan, P., and Stolfo, S. (1997). "On the accuracy of meta-learning for scalable data mining." *J. Intell. Inf. Sys.*, 8(1), 5–28.

Chang, C. Z. (1997). *Research on the mix proportion of high flowing eugenic concrete*, Chung Hua Univ., Hsin Chu, Taiwan.

Chang, C.-C., and Lin, C.-J. (2001). *LIBSVM*: A library for support vector machines [Computer software]. ⟨http://www.csie.ntu.edu.tw/~cjlin/libsvm⟩

Chang, T.-P., Chung, F.-C. , and Lin, H.-C. (1996). "A mix proportioning methodology for high-performance concrete." *J. Chin. Inst. Eng.*, 19(6), 645–655.

Chung, F. C. (1995). *Study on characteristic of coarse aggregate in high-performance concrete*, National Taiwan Univ. of Science and Technology, Taipei, Taiwan.

Dibike, Y. B., Velickov, S., Solomatine, D. P., and Abbott, M. B. (2001). "Model induction with support vector machines: Introduction and applications." *J. Comput. Civ. Eng.*, 15(3), 208–216.

Duffy, N., and Helmbold, D. (2002). "Boosting methods for regression." *Mach. Learn.*, 47(2–3), 153–200.

Euro-International Concrete Committee and International Federation for Prestressing (CEB-FIP). (1994). "Application of high performance concrete." *Rep. of CEB-FIP working group on HS/HPC*, Int. Federation for Structural Concrete, Lausanne, Switzerland, 69.

Fan, D., Chan, P., and Stolfo, S. (1996). "A comparative evaluation of combiner and stacked generalization." *AAAI 1996 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, Association for the Advancement of Artificial Intelligence, Menlo Park, CA, 40–46.

Fazel Zarandi, M. H., Türksen, I. B., Sobhani, J., and Ramezanianpour, A. A. (2008). "Fuzzy polynomial neural networks for approximation of the compressive strength of concrete." *Appl. Soft Comput.*, 8(1), 488–498.

Friedman, J. H. (2001). "Greedy function approximation: A gradient boosting machine." *Ann. Stat.*, 29(5), 1189–1232.

Giaccio, G., Rocco, C., Violini, D., Zappitelli, J., and Zerbino, R. (1992). "High-strength concretes incorporating different coarse aggregates." *ACI Mater. J.*, 89(3), 242–246.

Gjorv, O. E., Monteiro, P. J. M., and Mehta, P. K. (1990). "Effect of condensed silica fume on the steel-concrete bond." *ACI Mater. J.*, 87(6), 573–580.

J. Comput. Civ. Eng., 2011, 25(3): 242-253

Gupta, R., Kewalramani, M. A., and Goel, A. (2006). "Prediction of concrete strength using neural-expert system." *J. Mater. Civ. Eng.*, 18(3), 462–466.

Hegazy, T., Fazio, P., and Moselhi, O. (1994). "Developing practical neural network applications using back-propagation." *Microcomput. Civ. Eng.*, 9(2), 145–159.

Hover, K. C. (1998). "Concrete mixture proportioning with water-reducing admixtures to enhance durability: A quantitative model." *Cem. Concr. Compos.*, 20(2–3), 113–119.

Hwang, G. E. (1991). *A study on blast furnace slag concrete*, National Chiao Tung Univ., Hsin Chu, Taiwan.

Hwang, T. S. (1966). *Compressive strength of blast furnace slag concrete*, National Chiao Tung Univ., Hsin Chu, Taiwan.

Ian, W. H., and Eibe, F. (1999). *Data mining: Practical machine learning tools and techniques with Java implementations*, Morgan Kaufmann, New Zealand.

Iba, W., and Langley, P. (1992). "Induction of one-level decision trees." *Proc., Ninth Int. Machine Learning Conf.*, Int. Machine Learning Society, Princeton, NJ.

Jang, J. S. R. (1993). "ANFIS: Adaptive-network-based fuzzy inference system." *IEEE Trans. Syst., Man Cybern.*, 23(3), 665–685.

Kasperkiewicz, J., and Dubrawski, A. (1995). "HPC strength prediction using artificial neural network." *J. Comput. Civ. Eng.*, 9(4), 279–284.

Kohavi, R. (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection." *Proc., 14th Int. Joint Conf. on Artificial Intelligence*, Morgan Kaufmann, San Francisco, 1137–1143.

Langley, W. S., Carette, G. G., and Malhotra, V. M. (1989). "Structural concrete incorporating high volumes of ASTM Class F fly ash." *ACI Mater. J.*, 86(5), 507–514.

Lee, C. F. (1994). *A study on dry shrinkage and creep property of HPC*, National Taiwan Univ. of Science and Technology, Taipei, Taiwan.

Lee, J. J., Kim, D., Chang, S. K., and Nocete, C. F. M. (2009). "An improved application technique of the adaptive probabilistic neural network for predicting concrete strength." *Comput. Mater. Sci.*, 44(3), 988–998.

Lessard, M., Chaalla, O., and Aitcin, P.-C. (1993). "Testing high-strength concrete compressive strength." *ACI Mater. J.*, 90(4), 303–308.

Lin, F. Y. (1994). *Mixture proportion and quality of HPC*, National Taiwan Univ. of Science and Technology, Taipei, Taiwan.

Mo, H. L. (1995). *A study on high performance concrete*, National Taiwan Univ. of Science and Technology, Taipei, Taiwan.

Mostofi, D., and Samaee, A. (1995). "HPC strength prediction using artificial neural network." *J. Comput. Civ. Eng.*, 9(4), 44–49.

Oh, J., Lee, I., Kim, J., and Lee, G. (1999). "Applications of neural networks for proportioning of concrete mixes." *ACI Mater. J.*, 96(1), 51–59.

Oluokun, F. A. (1994). "Fly ash concrete mix design and the water-cement ratio law." *ACI Mater. J.*, 91(4), 362–371.

Pal, M., and Mather, P. M. (2003). "Support vector classifiers for land cover classification." *Proc., 6th Annual Int. Conf. and Exhibition MapIndia*, Map India, Noida, India, 28–31.

Popovics, S. (1990). "Analysis of the concrete strength versus water-cement ratio relationship." *ACI Mater. J.*, 87(5), 517–529.

Seyhan, A. T., Tayfur, G., Karakurt, M., and Tanoglu, M. (2005). "Artificial neural network (ANN) prediction of compressive strength of VARTM processed polymer composites." *Comput. Mater. Sci.*, 34(1), 99–105.

Smola, A. J., and Schölkopf, B. (2004). "A tutorial on support vector regression." *Stat. Comput.*, 14(3), 199–222.

SPSS Clementine version 12 [Computer software]. IBM, Chicago.

Topçu, I. B., and SarIdemir, 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.

Trtnik, G., Kavcic, F., and Turk, G. (2009). "Prediction of concrete strength using ultrasonic pulse velocity and artificial neural networks." *Ultrasonics*, 49(1), 53–60.

Vapnik, V. N. (1995). *The nature of statistical learning theory*, Springer-Verlag, New York.

Vapnik, V. N. (1998). *Statistical learning theory*, Wiley, New York.

Yeh, I.-C. (1998). "Modeling of strength of high performance concrete using artificial neural networks." *Cem. Concr. Res.*, 28(12), 1797–1808.

Yeh, I.-C. (1999). "Design of high-performance concrete mixture using neural networks and nonlinear programming." *J. Comput. Civ. Eng.*, 13(1), 36–42.

Yeh, I. C., and Lien, L.-C. (2009). "Knowledge discovery of concrete material using genetic operation trees." *Expert Syst. Appl.*, 36(3), 5807–5812.

J. Comput. Civ. Eng., 2011, 25(3): 242-253