

# Prediction of self-healing in engineered cementitious composite: Machine learning comparative analysis

Guangwei Chen<sup>1,2,\*</sup>, Patrick Tang<sup>1</sup>, Shuo Chen<sup>1</sup>, Shangyong Wang<sup>1</sup>, and Hongzhi Cui<sup>3<sup>1</sup></sup>

<sup>1</sup>The University of Newcastle, Callaghan, 2308, NSW, Australia

<sup>2</sup>Qiannan Normal College of Nationalities, Guizhou, China

<sup>3</sup>College of Civil and Transportation Engineering, Shenzhen University, Shenzhen 518060, China

## Abstract

Engineered cementitious composite (ECC) is a unique material which can significantly contribute to self-heal cracks based on ongoing hydration. As model-free approaches eliminating the need of new models for different prediction problems, machine learning algorithms have been widely used to model the properties of concrete. This paper aims to provide a comparative analysis on the performance of machine learning models, the individual and ensemble methods, in predicting self-healing capability of ECC. These models include four individual methods (linear regression (LR), back-propagation neural network (BPNN), classification and regression tree (CRAT), and support vector regression(SVR)), three ensemble methods (bagging, AdaBoost, and stacking) with each of the four individual models used as the based learner. The results shows that the stacking model performs the best with the lowest error and highest accuracy on the self-healing capability prediction of ECC. On average, all ensemble models improve the prediction performance of the individual model which is used as the base learner in the ensemble model.

**Keywords** ECC, self-healing, machine learning, ensemble method

## 1 Introduction

A market research project was commissioned by Materials for Life (M4L), which shows cracking damage in concrete was experienced more than any other problems [1]. Moreover, cracks are primarily responsible for the reduction of the strength and stiffness of the concrete structure. Annual cost spent on maintenance, refurbishment and repair such cracks to prolong the service life of infrastructure is estimated around 50% of the annual construction budget in European [2]. Based on such circumstances, self-healing cementitious materials are proposed by M4L research team. They state self-healing cementitious materials benefit water retaining infrastructure to reduce repair costs over a premium in the capital material cost [1].

The inspiration of self-healing comes from biomimicry concept and the healing process in living nature [3]. For example, the skin of humans or animals can biologically repair itself from scars injured area. According to the healing mechanism, self-healing can be categorised into two major classes, autogenous healing and autonomous healing [4]. The former indicates the self-healing ability results from the physical and/or chemical composition of the cementitious matrix. And the latter states the self-healing mechanism is triggered by some biological agents, such as bacteria which are artificially presented in the cementitious matrix.

Generally, autogenous self-healing in concrete is attributed to two major mechanisms including (1) further hydration of cement particles and/or swelling of calcium silicate hydrate; (2) calcium hydroxide carbonation [5]. Researchers have proposed the maximum crack widths as 10  $\mu\text{m}$  [6], 100  $\mu\text{m}$  [7], 200  $\mu\text{m}$  [8], 205  $\mu\text{m}$  [5] and 300  $\mu\text{m}$  [9] are self-healed completely [10].

Engineered cementitious composite (ECC), as an autogenous self-healing of concrete, is a high performance fiber-reinforced cementitious composite designed with micro-mechanical principles [11]. It features high tensile ductility with a moderate fiber-volume fraction [12, 13] - typically 2% by volume which is designed to promote the self-healing ability [4].

The intrinsic self-healing ability of ECC is complex and uncertain due to material nature, interactivity between different composites in the cementitious matrix and its interaction with the exposed environment [14], and unpredictable crack location, orientation and width [15]. Several studies explored the influence of different factors, such as limestone powders (LP) [16, 17], fly ash (FA) [18, 19], hydrate lime [20], water/binder ratio [21], water permeation [22] and different curing condition (air, carbon dioxide, wet/dry and water) [23] on self-healing behavior. However, the relationship between multiple factors is questionable and non-linear, which makes it difficult to predict self-healing mathematically based on available data. **Moreover, mathematical models based on empirical data are generally in regression forms which cannot be used for problem containing independent variables, such as prediction of self-healing ability of ECC, because of less accuracy and more assumption in regression form (linear, non-linear, etc.) [24].**

To compensate for the drawbacks of mathematical models with multiple interaction variables, machine learning algorithms are applied to many civil engineering problems. Many research works have been conducted using machine learning algorithms for the prediction of various properties of concrete. Gilan et al.[25] established a hybrid Support Vector Regression (SVR) - Particle Swarm Optimization (PSO) algorithm model to predict the compressive strength and Rapid Chloride Penetration Test (RCPT) results of concretes containing metakaolin. Yan et al. [26] predicted bond strength of glass fiber-reinforced polymer bar concrete by Artificial Neural Network (ANN) with Genetic Algorithm (GA). Yaseen et al.[27] proposed a machine learning method called Extreme Learning Machine (ELM) to predict the compressive strength of lightweight foamed concrete.

Compared with single method, several scholars compares the performance of various machine learning algorithms on predicting concrete properties to indicate the best modeling method. Yan and Shi [28] reported that SVR is better than other individual methods to predict elastic modulus of normal and high strength concrete. Chou [29] compared performance of individual and ensemble methods for predicting the mechanical properties of high performance concrete. Reuter et al. [30] employed three individual approaches to model concrete failure surfaces. Sobhani et al. [31] indicates ANN and a proposed fuzzy inference system are more feasible than traditional regression models on predicting no-slump concrete. Omran et al. [32] compared prediction results of environmentally friendly concrete compressive strength by using three individual methods, two ensemble methods, and four regression tree models.

**red letters represent comments that will be delete later., blue letters represent new added materials.**

Although researchers have been utilized different machine learning algorithms to anticipate and evaluate various properties of concrete, the application of machine learning focusing on self-healing is considerably rare.

**Mauludin and Oucif [33]** reviewed the methods used for modeling autogenous self-healing in concrete, which mainly fell into two categories, numerical simulation and machine learning. And the only machine learning model reviewed in their study was proposed by Ramadan et al. [3]. A GA-ANN model was applied in [3] to predict the self-healing ability of cement-based materials using dataset collected from literatures, and stated that the proposed GA-ANN model was capable of capturing the complex effects of various self-healing agents (e.g., biochemical material, silica-based additive, expansive and crystalline components) on the self-healing performance in cement-based materials. However, they didn't compare their prediction performance with linear regression or neural network model to present the advantage of proposed GA-ANN model eliminating the data dependence of models.

**Comments only shown in this draft:** In [3], the prediction accuracy  $R^2$  is over 99.7%, which is doubtable, especially on training dataset. Generally, over 99.7% prediction accuracy means the relationship between input variables and the output variable fit into a mathematical rule or expression. However, there is measurement errors in all engineering problems, and this measurement error is stochastic. That means we almost

impossible to find out a mathematical model to present the relationship between input and output variables explicitly.

Wang et al. [35] used ANN and ensemble classification models to implement binary prediction (healed or not) of self-healing materials based on 30 samples. However, the regression models were conducted by numerical simulations which assumed the relationship between the input and output variables following some logistic regression models. Later, [36] used an ANN model to predict the self-healing property, compressive strength, of Ground Granulated Blast Furnace slag using 51 samples collected from lab experiment. However, these two papers lacks prediction performance comparison on validation and testing dataset, and benchmark models, such as linear regression model. Moreover, limited number of samples (data) may not be enough for training a reliable model.

A comparative study was conducted by [34] using SVR, Decision Tree Regression (DTR), Gradient Boosting Regression (GBR), ANN, Bayesian Ridge Regression (BRR) and Kernel Ridge Regression (KRR) to predict the self-healing capability of bacteria-based concrete. The results showed GBR (with  $R^2$  0.736 on testing dataset and 0.93 on training dataset) was superior to other models (with  $R^2$  less than 0.7 on both training and testing dataset on most models). In their study, extensive experiments with different combinations of influencing variables were implemented to generate empirical dataset. However, only three variables, the number of bacteria, the healing time and the initial crack width, were selected to predict the crack close percentage as the output.

To the best of our knowledge, no studies have applied a ML model to predict the self-healing of ECC, let alone presenting a comparative analysis by comparing prediction performance of individual and ensemble ML models. Thus, the objective of this paper is to compare multiple machine learning techniques to identify the most reliable model for predicting the self-healing behavior of ECC, which is also considered as a baseline prediction model for other advanced models in our future works.

In this paper, four individual methods including linear regression (LR), SVR, back-propagation neural network (BPNN), and classification and regression tree (CART) were used to predict self-healing capability of ECC, which were then constructed as base learners in three ensemble methods naming bagging, AdaBoost and stacking for prediction. Variable controlled experiments were implemented to collect empirical data used in ML models, where only the input variables of ML models are variables in experiments, and other impact factors were controlled constantly in experiments. Empirical data collected from variable controlled experiments were preprocessed and then split by a 10-fold cross-validation algorithm (details refer to Section 4.1) to avoid overfitting and minimize bias associated with various range of data. Figure 1 summarizes the steps that were performed for predicting self-healing capability of ECC.

The remainder of this paper is organized as follows. Section 2 present the experimental program that devotes to materials used in preparing ECC mixtures and preprocessing of collected samples from experiments. It is followed by methodology Section 3 which describes the concepts and formulations of individual and ensemble models used for predicting self-healing capability of ECC. Validation and Evaluation methods are introduced in Section 4. After that, computational results of all individual and ensemble models are presented and compared in Section 5, which demonstrate the superiority of machine learning algorithms for prediction of the self-healing capability of ECC. Finally, Section 6 draws some conclusions from this work and suggests directions for future research.

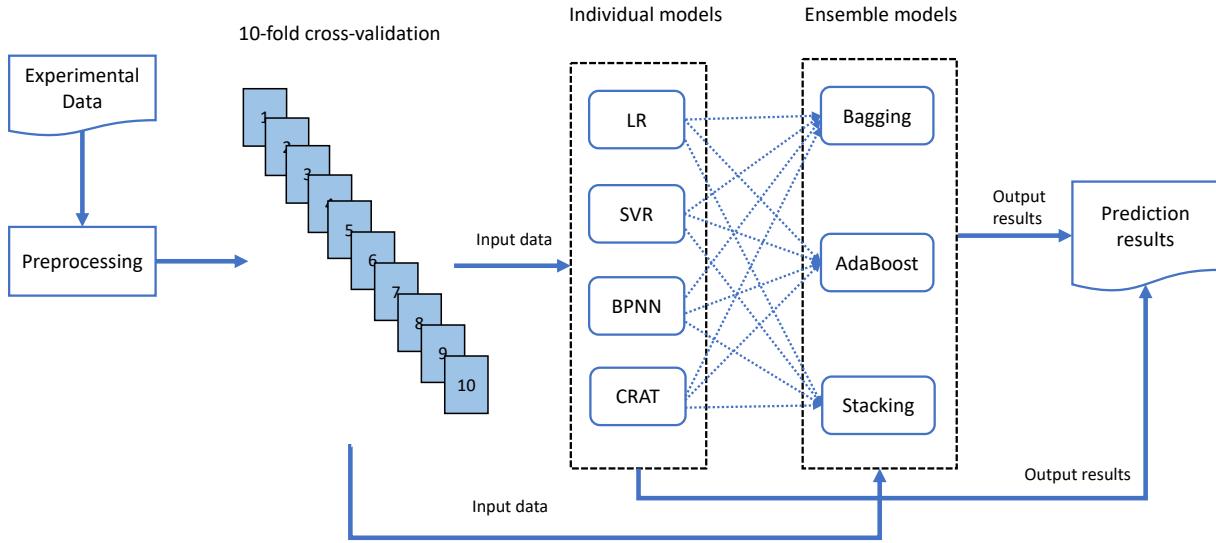


Figure 1: Flow chart of implementing prediction models for self-healing capability of ECC

## 2 Experimental Program

### 2.1 Materials and Mixture Proportion

In order to investigate the influence of mix composition on the self-healing capability of ECC, nine ECC mixes were prepared for experimental analysis. The binder composition of ECC mixes were general purpose cement (GPC), fly ash (FA), silica fume (SF), hydrated lime powder(LP), fine sand, polyvinyl alcohol fibers (PVA), as well as water and high range water reducing admixture (HRWR). GPC and FA were supplied by Boral in accordance with Australian Standard AS 3972-2010, while LP was the Adelaide Brighton Hydrated Lime with a specific gravity of 2.2-2.3, and a typical fineness of 0.1% retained on a  $75\mu\text{m}$  sieve and less than 0.05% on a  $250\mu\text{m}$  sieve. The physical and chemical properties of cementitious materials are shown in Table 1. Fine sand was employed with an average grain size of  $150\mu\text{m}$  and fineness modulus of 2.01. PVA were produced by Domocrete and their mechanical and geometrical properties are described in Table 2.

During the production process, a planetary-type mixer of 50 L capacity was used to produce specimens with water to cementitious materials (W/CM) ratio of 0.29, and sand to CM (PC + FA + LP+SF) ratio of 0.36. All the fine aggregates were in saturated surface dried condition prior to concrete mixing. Among nine ECC mixes, a mixture was prepared with GPC and FA while other mixtures were obtained by replacing part of the FA by SF or LP. For example, ECC mixture represented by FA70 stated that 70% of the cementitious materials weight was FA, and by FA60-SF10 indicated that 10% of the FA was replaced by SF. The details of mix proportion for all nine mixtures are shown in Table 3.

### 2.2 Experiment

To prepare cement pastes, solid ingredients including cement, mineral admixtures and sand were first mixed for 30 seconds. Water pre-mixed with HRWR was then added into dry solid ingredients and mixed for 2 minutes. After that, PVA were slowly added into the matrix which were blended continually until PVA distributed evenly. For creating ECC composite cylinders, cement pastes were cast into standard moulds with dimension of  $\varnothing 100\text{mm} \times 200\text{mm}$  which were demolded after 24 hours. The demolded cylinders were cured in a curing room with a temperature of  $23 \pm 2^\circ\text{C}$  and the relative humidity (RH) of  $90 \pm 5\%$  for 28

Table 1: Physical and chemical properties of cementitious materials

<i>Chemical composition (%)</i>	GPC	FA	LP	SF
Silica ( $\text{SiO}_2$ )	19.8	65.90	1.8	95.10
Alumina ( $\text{Al}_2\text{O}_3$ )	5.3	24.0	0.5	0.21
Iron oxide ( $\text{Fe}_2\text{O}_3$ )	3.0	2.87	0.6	0.29
Calcium oxide ( $\text{CaO}$ )	64.2	1.59	72.0	-
Magnesia ( $\text{MgO}$ )	1.3	0.42	1.0	-
$\text{R}_2\text{O}$	0.6	1.93	-	-
Sulfur trioxide ( $\text{SO}_3$ )	2.7	-	-	-
Titanium oxide ( $\text{TiO}_2$ )	0.28	0.91	-	-
Manganic oxide ( $\text{Mn}_2\text{O}_3$ )	0.22	-	-	-
Zirconia ( $\text{ZrO}_2$ ) + Hafnium ( $\text{HfO}_2$ )	-	-	-	3.46
Loss on ignition (%)	2.8	1.53	24.0	1.4
Density ( $\text{g/cm}^3$ )	3.08	2.43	2.25	2.26
Specific surface area ( $\text{m}^2/\text{kg}$ )	-	655	460	$1.5 \times 10^4$

Table 2: Properties of PVA

Length (mm)	Length/ diameter ratio	Young's modulus (MPa)	Elongation (%)	Tensile strength (MPa)	Density ( $\text{g/cm}^3$ )
8	200	42000	7	1600	1.3

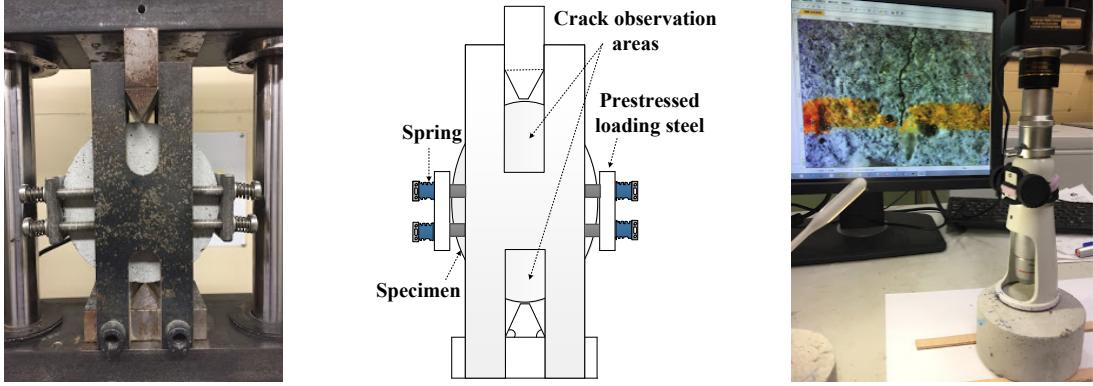
Table 3: Mix proportion of all ECC mixtures

Mix	Water/CM	Sand	Water	fibre (V)	GPC	Fly ash	SF	LP	HRWR
FA70	0.29	419.67	338.07	26	349.73	816.03	0.00	-	5.13
FA65-SF5	0.29	419.67	338.07	26	349.73	757.74	58.29	-	5.13
FA60-SF10	0.29	419.67	338.07	26	349.73	699.45	116.58	-	5.13
FA55-SF15	0.29	419.67	338.07	26	349.73	641.16	174.86	-	5.13
FA65-LP5	0.29	419.67	338.07	26	349.73	757.74	-	58.29	5.13
FA60-LP10	0.29	419.67	338.07	26	349.73	699.45	-	116.58	5.13
FA55-LP15	0.29	419.67	338.07	26	349.73	641.16	-	174.86	5.13
FA55-SF5-LP10	0.29	419.67	338.07	26	349.73	641.16	58.29	116.58	5.13
FA55-SF10-LP5	0.29	419.67	338.07	26	349.73	641.16	116.58	58.29	5.13

days. Finally, cylinders were cut into specimens with dimension of  $\text{Ø}50\text{mm} \times 100\text{mm}$  using a diamond blade saw.

A newly developed splitting tensile test apparatus was used to generate micro-cracks and control the influence on crack differences, which was shown in Figure 2 (a). It consisted of a steel frame, top member, bottom member, prestressed loading steel plates (5mm thick) on both sides with loading nuts and wire springs. Both steel plates were connected to the steel frame by nuts and wire springs. While a specimen was placed inside the steel frame, it was pre-stressed by the steel plates from both sides limiting the propagation and size of crack and preventing excessive crack growth.

Micro-cracks less than  $150\mu\text{m}$  were first produced by pre-loading up to 70% of their maximum splitting strength. We used a digital microscope to observe the crack width on the surface of specimens shown



(a) Splitting tensile test apparatus    (b) Schematic diagram of apparatus    (c) Crack width measurement

Figure 2: Splitting tensile test apparatus and microscope used in experiment for creating and measuring ECC cracks

in Figure 2 (c). After that, specimens were subjected to self-heal through wet-dry (W/D) cycles which contained submersion in water at  $23 \pm 2^\circ\text{C}$  for 24 hours and drying in a laboratory medium at  $50 \pm 5\%$  RH and  $23 \pm 2^\circ\text{C}$  for 24 hours. After 10 W/D cycles, cracks are measured again by the digital microscope to indicate the partial or full closure of crack. Figure 3 illustrated the comparison of crack width changes for two ECC mixture specimens before and after W/D cycles due to self-healing.

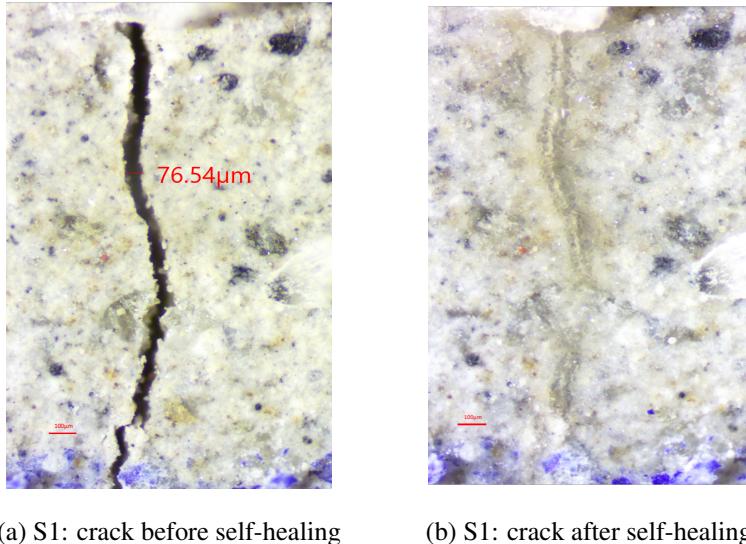


Figure 3: Comparison of crack width changes in two ECC specimens, S1 and S2, before and after self-healing

### 2.3 Data Collection

Empirical data for prediction were gathered with four features, crack width before self-healing representing the influencing factor of self-healing, and the weight of FA, SF, and LP illustrating the variable composites of ECC. It is noteworthy that some impact factors in the test, such as GPC, sand, W/CM, and healing time are controlled as constants which are excluded for the prediction modeling due to no effect. There are 6 specimens with dimension of  $\varnothing 50\text{mm} \times 100\text{mm}$  for each mixtures which were observed using digital

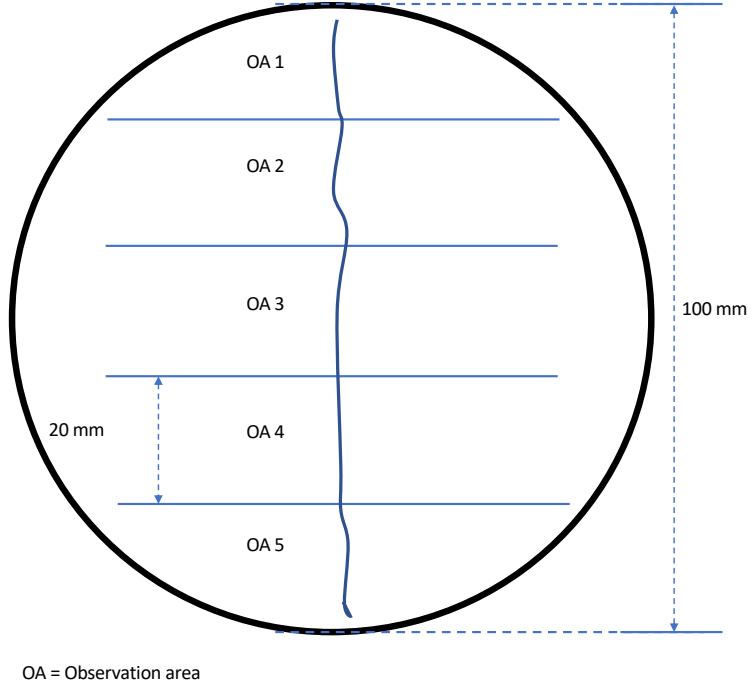


Figure 4: Schematic diagram of measuring observation areas on the surface of ECC mixture specimen

microscope to collect crack width samples before self-healing and after self-healing. Four horizontal lines were drawn on the surface of each specimen along the direction of vertical force, which divided the specimen into five observation areas. The schematic diagram of observation measurement is shown in Figure 4. In each observation area, we recorded one data sample if the crack width showed little change along the vertical force; Otherwise, we recorded multiple data samples with different crack widths separately before and after self-healing, while the crack width is inconsistent along the vertical force. For example, the front end of the crack is wide, and the rear part is narrow. Totally, 617 data samples were collected from nine mixtures to construct machine learning training-testing dataset. Table 4 shows the number of collected samples and range of crack width before and after self-healing in each mixture.

## 2.4 Preprocessing of Data

The input and output data of different features (referring to Table 3 and 4) vary in range and units which weigh all features unequally for prediction models and might end up creating bias. To eliminate this effect, we preprocessed empirical data to the range [0,1] by the min-max scaling presented in the following function.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Where  $x'$  is the scaled value of the variable  $x$ ,  $x_{max}$ ,  $x_{min}$  are the maximum and minimum values of variable  $x$  respectively.

Table 4: Number of crack samples and range of crack width before and after self-healing collected from nine ECC mixes

Mix	Number of crack samples	Crack width before self-healing		Crack width after self-healing	
		Min ( $\mu m$ )	Max ( $\mu m$ )	Min ( $\mu m$ )	Max ( $\mu m$ )
FA70	87	3.28	134.69	0	121.37
FA65-SF5	77	4.37	135.47	0	124.01
FA60-SF10	88	5.18	121.78	0	113.11
FA55-SF15	88	3.45	115.8	0	109.53
FA65-LP5	112	7.65	119.45	0	105.65
FA60-LP10	37	5.62	126.82	0	110.97
FA55-LP15	61	6.42	132.65	0	115.95
FA55-SF5-LP10	34	8.74	123.09	0	110.78
FA55-SF10-LP5	33	4.64	131.57	0	119.79

### 3 Methodology

Machine learning techniques used to predict the self-healing capability of ECC in this study are four individual methods including LR, SVR, BPNN and CART, as well as three ensemble methods including bagging, AdaBoost and stacking. Ensemble methods are constructed using individual methods as base estimators to predict the self-healing capability of ECC. To establish a baseline for comparison, modeling parameters of individual methods are set to the same in both individual models and ensemble models. The reason of choosing these techniques are because they are the most popular, and some of are even recognized as the top data mining algorithms in related field of concrete [29]. The proposed individual and ensemble techniques are described in the following subsections.

#### 3.1 Linear Regression

LR attempts to determine the relationship between a dependent variable (response variable) and one or more independent variables (explanatory variables) by fitting a linear regression equation [37]. Given our dataset  $T = \{(x_i, y_i), i = 1, 2, \dots, n\}$ , where  $n = 617$  is the size of sample dataset.  $x_i \in R^n$  is independent variables representing a sample of selected features from FA, SF, LP and crack width before self-healing,  $R^n$  is  $n$ -dimensional space,  $y_i \in R^1$  is the target output (crack width after self-healing) that corresponds to  $x_i$ . Let  $d = 4$  denote the number of a independent variable of an random vector  $x = \{x_1; x_2; \dots; x_d\}$ , and  $y$  is the corresponding output ( dependent variable). The general formula of LR for predicting self-healing capability of ECC can be expressed as follows:

$$y = w_1x_1 + w_2x_2 + \dots + w_dx_d + b \quad (2)$$

where  $w_i$ , ( $i = 1, 2, \dots, d$ ) denotes a regression coefficient,  $b$  is an error term.

#### 3.2 Support Vector Regression

The support vector machine (SVM) is a supervised machine learning method and first introduced by Vapnik [38, 39] based on statistical learning theory [40]. Since then, it gains popularity due to attractive features, and promising empirical performance. SVM includes two main categories: support vector classification (SVC) and SVR. For classification problem, input data as vectors are mapped to a high-dimension feature

space where an optimal separating hyperplane is constructed by using a *kernel* function to assign new data into different categories[41]. By implementing the structural risk minimization (SRM), SVC obtains good generalization.

For the regression, the basic idea is to provide a nonlinear function by mapping input data into the high-dimensional feature space where a special type of hyperplane is constructed. After that, a regression model is build in the hyperplane [42]. It implements approximately the SRM to set up an upper bound of the generalization error so as to achieve generalized performance [43].

Given our dataset  $T = \{(x_i, y_i), i = 1, 2, \dots, n\}$ , where  $n = 617$  is the size of sample dataset,  $x_i \in R^n$  is the input vector representing a sample of selected features from FA, SF, LP and crack width before self-healing,  $R^n$  is  $n$ -dimensional vector space,  $y_i \in R^1$  is the target output indicating crack width after self-healing that corresponds to  $x_i$ . The SVR aims to seek a optimum regression function  $f(x)$  with minimized empirical risk, which can be expressed as follow:

$$f(x) = \langle w, x \rangle + b \quad \text{with} \quad w \in T, b \in R \quad (3)$$

where  $\langle \cdot, \cdot \rangle$  denotes the dot product in  $T$ ,  $w$  and  $b$  are the weight vector and bias value which are estimated by minimizing the empirical risk, that is, the distance between the predicted crack width and the target crack width after self-healing.

SVR adopts an  $\epsilon$ -insensitive loss function penalizing predictions that has a distance between the predicted crack width and the target crack width after self-healing greater than  $\epsilon$ . Therefore, the problem of finding  $w$  and  $b$  to reduce the empirical risk with respect to an  $\epsilon$ -insensitive loss function is equivalent to the convex optimization problem that minimizes the margin ( $w$ ) with the full prediction error within the range of  $\epsilon$ . Then this problem can be expressed as:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|w\|^2 \\ & \text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon \\ \langle w, x_i \rangle + b - y_i \leq \epsilon \end{cases} \end{aligned} \quad (4)$$

By introducing slack variables  $\xi, \xi_i^*$  to allow some errors to cope with infeasible solution of the optimization problem, the formulation can be generated as [39]:

$$\begin{aligned} & \text{minimize} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (5)$$

The constant  $C$  is the penalty value imposed on predictions that lie outside the  $\epsilon$  margin. After calculating a Lagrange constructed from objective function and all constraints, introducing a dual set of variables as follows [44]:

$$\begin{aligned} L_P = & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n (\eta_i \xi_i + \eta_i^* \xi_i^*) \\ & - \sum_{i=1}^n \alpha_i (\epsilon + \xi_i - y_i + \langle w, x_i \rangle + b) \\ & - \sum_{i=1}^n \alpha_i^* (\epsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b) \\ s.t. \quad & \alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0 \end{aligned} \quad (6)$$

Where  $L_P$  is the Lagrangian and  $\alpha_i, \alpha_i^*, \eta_i, \eta_i^*$  are Lagrange multipliers.

The optimality can be achieved by the partial derivatives of  $L_P$  with respect to the primal variables following the saddle point condition. Then the function of SVR is obtained as:

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \quad (7)$$

As for the nonlinear regression, the input data have to be mapped into a high-dimensional feature space, in which dot product can be replaced by a kernel function  $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ , and the function (7) can be written as:

$$f(x) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(x_i, x) + b \quad (8)$$

There are various kernel functions, such as linear, polynomial, radial basis function and sigmoid kernel. In this work, the Gaussian radial basis function (RBF) is chosen, which is defined as [45]:

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad (9)$$

### 3.3 Artificial Neural Network

Artificial neural network (ANN), also called neural network which is originated from simulating biological neural networks. Generally, it consists of many neurons in layers including one input layer, one or several hidden layers and an output layer [46]. The neurons are fully interconnected between the neighboring layers by weight, and typically no inter-connections between neurons within the same layer [47].

There are many possible network structures, in this study the type utilized is the BPNN. A preliminary architecture of the BPNN is determined to be 4 - n - 1, where 4 input neurons represent the input features standing for FA, LP, SF and crack width before self-healing,  $n = 5$  indicates the number of neurons in the hidden layer, and 1 target neuron in the output layer for the predicted crack width after self-healing. This is a three-layer network with one hidden layer which is already able to approximate most continuous function, of which the complex nonlinear relationship could be approximated in accuracy [26]. The architecture of the BPNN model for predicting self-healing is demonstrated in Figure 5.

Given a set of inputs  $\{x_1, x_2, x_3, \dots, x_n\}$ , while information is passed through the input layer to the hidden layer, each neuron in the input layer are multiplied by respective weights added by a bias and are summed together. After that, a activation function  $f$  is applied to form the output  $z$ . This can be expressed in the following equation [24]:

$$z = f\left(\sum_{i=1}^n w_{ij} x_i + b_j\right) \quad (10)$$

where  $w_{ij}$  is the connection weights between the  $i$ th neuron of input and the  $j$ th neuron in the hidden layer, and  $b_j$  is the bias of the  $j$ th neuron. The sigmoid function is applied as activation function between the input, hidden, and output neurons to form the output.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (11)$$

The goal of training a neural network is to determine the values of the connection weights and the biases of the neurons. The back propagation indicates a iterated method which adjusts the weights from output layer to input layer. At first, a output are calculated feed-forward from the input layer via the hidden layer to the output layer. Then a error is generated by comparing the output with the target output. After that, the

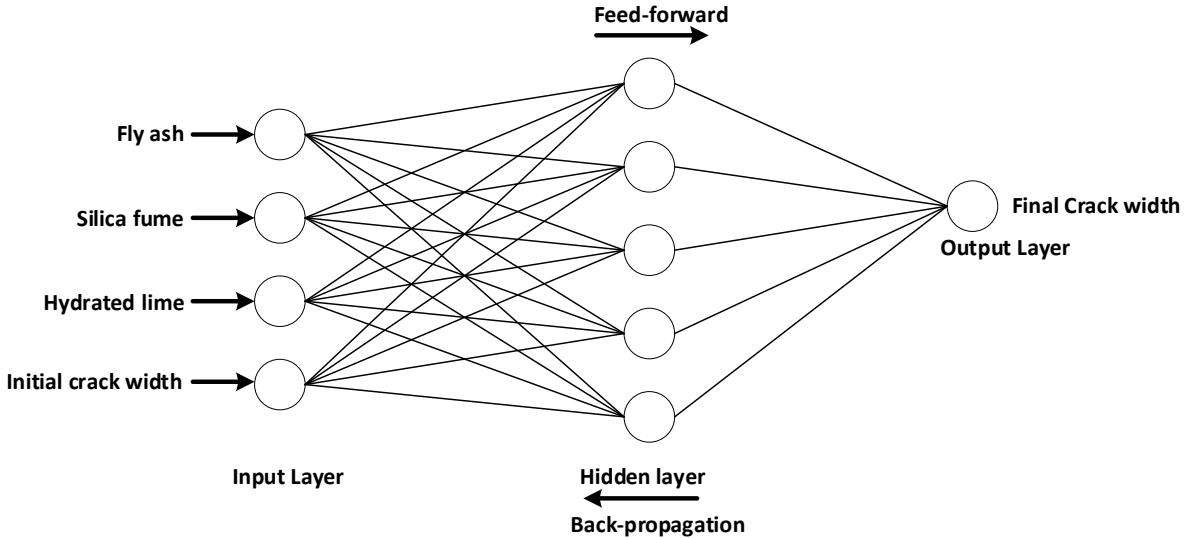


Figure 5: Schematic diagram of BPNN model for predicting self-healing capability of ECC

error is back propagated to the hidden layer and input layer with adjusting the connection weights and biases to reduce the error. Such a process will be repeated until the error is minimized or the termination is reached to avoid over-fitting.

### 3.4 Classification and Regression Tree

The CART [48] is a tree decision algorithm that split data into mutually exclusive subgroups based on recursive binary partitioning procedure. It develops the relationship between the target variables (the crack width after self-healing of ECC) and the independent variables (the input features of FA, SF, LP and crack width before self-healing of ECC) to create decision rules to form subgroups as branches and leaves in the Figure 6 that illustrates the schematic diagram of a decision tree. The process of CART starts from the root node which contains the entire data set to construct two sub-nodes representing two categories. Then this recursion process is applied to each sub-node until all divided sub-nodes are leaf nodes. The CART tree can be either a classification tree [49] or regression tree [50] depending on the type of target and independent variables which may be categorical or numerical.

The key idea of constructing a CRAT tree is achieved by selecting a variable at each node that best splits the empirical data. To locate splits, *Gini* index is used to measure the impurity of the two child nodes which contains subsets of data are as homogeneous as possible with respect to the target variable.

Given a dataset has  $K$  classes and probability of a sample which belongs to class  $i$  is  $p_i$ ,  $i \in \{1, 2, 3, \dots, K\}$ , the *Gini* impurity can be expressed as,

$$G(p) = \sum_{i=1}^K p_i(1 - p_i) = 1 - \sum_{i=1}^K p_i^2 \quad (12)$$

### 3.5 Ensemble Methods

In contrast to various learning approaches such as SVM, CART which develop a single learner from training data, ensemble methods train multiple base learners and combine them [29] to improve generalizability over a single estimator. Therefore, weak learners (base learners) can be boosted to strong learner [51] in a

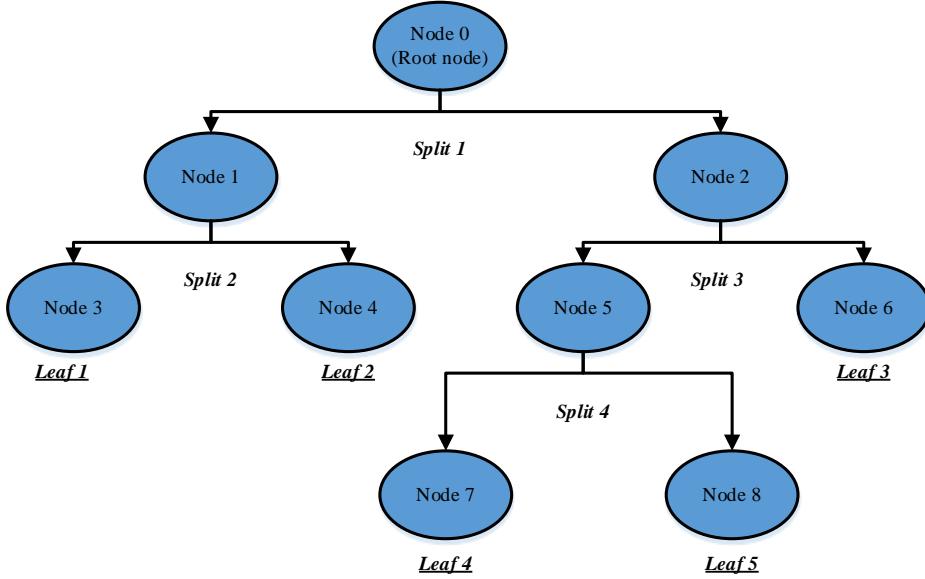


Figure 6: Structure of a classification and regression tree [50]

ensemble method. The set of base learners in an ensemble are developed from a individual learning algorithm which can be decision tree, SVM, or other kinds of learning algorithms. Researches [52] show that ensemble methods are usually significantly more accurate than individual learning methods

Suppose with a  $d$ -dimensional predictor variable  $X$  (input features of FA, SF, LP, and crack width before self-healing of ECC) and one dimensional output  $Y$  (The crack width after self-healing of ECC). Each estimator uses a individual algorithm to provide one estimated function  $g(\cdot)$ . The output presented by ensemble-based function  $g_{en}(\cdot)$  is obtained by a linear combination of individual functions. This ensemble approach can be expressed mathematically as:

$$g_{en}(\cdot) = \sum_{j=1}^N c_j * g(\cdot) \quad (13)$$

Where  $c_j$  expresses the combination coefficients which is dependent on the used ensemble models.

### 3.5.1 Bagging

Bagging (bootstrap aggregating) generate multiple versions of a predictor to obtain an aggregated predictor [53]. It generates multiple models independently on different versions of data set which are random bootstrap replications of original training set. That is, several training examples may repeatedly appear in different bootstrap replicated data set. Then those individual predictions are aggregated through an combination method (either voting or averaging) to form the final prediction. By introducing randomization into bagging method's construction procedure and making an ensemble out of it, bagging method is used as a approach to reduce the variance of a base estimator, such as a regression tree.

### 3.5.2 AdaBoost

Liking bagging, AdaBoost [54] manipulates modified versions of training examples repeatedly to generate multiple predictions which are then combined to form the final prediction. The difference is that AdaBoost applies a weight to each of the training examples. In each iteration, the weights are individually updated

based on minimizing the weighted error on the training set. It increases weights on those training examples were incorrectly predicted in previous iteration by the boosted model, whereas the weights are decreased for correctly predicted training examples. In subsequent iterations, therefore, AdaBoost constructs progressively more difficult learning problems. Once the training process has finished, the predictions are combined through a weighted majority vote (or sum) to produce the final prediction. The final classifier therefore usually achieves a high degree of accuracy in the test set.

### 3.5.3 Stacking

Stacking regression combines multiple regression models via a meta-regression, which use out-of-fold predicts concept [55] (Shown in Figure 7). It split the data set into K folds, k-1 folds are used to train the first level regressors in K successive rounds. In each round, the first level regressors are used to predict based on the remaining 1 subset. After that, the prediction results is used and stacked as input data to the second level regressor to form a final set of predictions [56]. In this study, SVR, BPNN and CRAT are used as regression models in the first level to get prediction results, and LR is used as met-regressor in the second level to combine and generate final prediction.

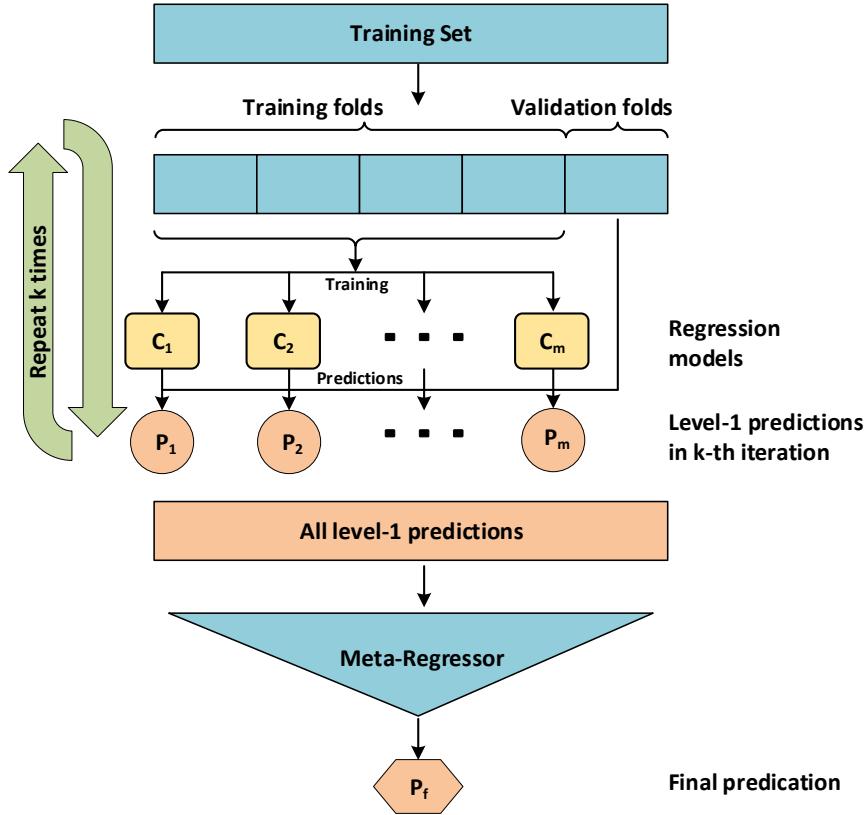


Figure 7: Schematic diagram of Stacking model [56]

## 4 Validation and Evaluation

### 4.1 Cross-validation Method

Generally, dataset is split to generate a training subset and a validation subset keeping the properties of the original dataset as much as possible to avoid misleading estimates. To minimize bias of random data splitting, the K-fold cross-validation is used widely [29]. In this study, a ten-fold cross-validation approach is applied to assess model performance in this study (shown in Figure 8), which yields the optimal computational time and reliable variance confirmed by Kohavi [57]. The empirical dataset is split into 10 equal-size subsets with a similar distribution. After that, training-tests perform repeatedly for 10 times using a different subset as the test set and the remaining subsets as the training set in each time [58]. The average accuracy in 10 times validation is expressed as the model accuracy.

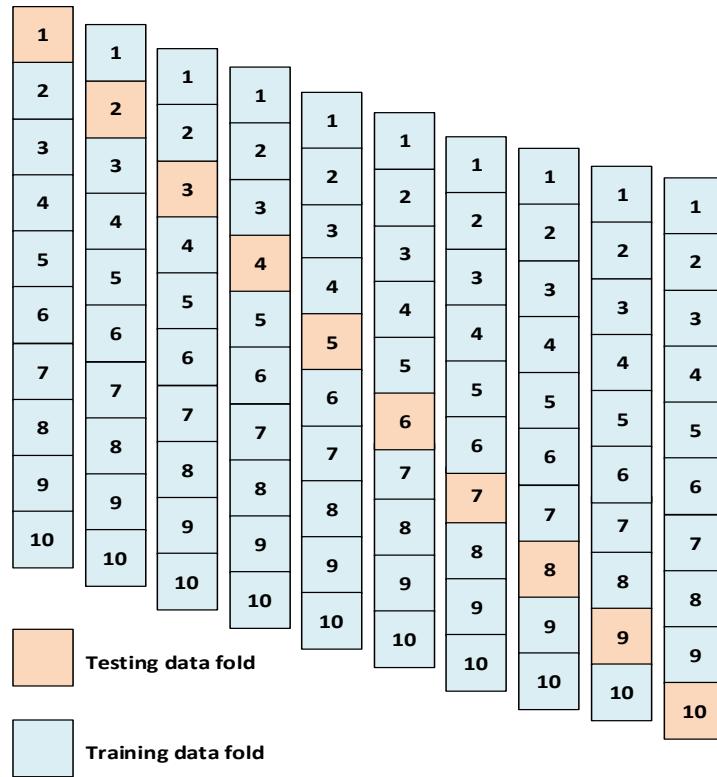


Figure 8: Ten-fold cross-validation approach

### 4.2 Performance Evaluation

To indicate and validate the accuracy of proposed machine learning models, three statistical indices, mean absolute Error (MAE), root mean square error (RMSE), and the coefficient of determination  $R^2$  are used as the following equations. The average deviation of the performance of a individual model or an ensemble model from a benchmark model in terms of three statistical measures (MAE, RMSE and  $R^2$ ) is calculated following equation (17).

- Mean absolute error (MAE).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y'_i| \quad (14)$$

- Root mean square error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2} \quad (15)$$

- Coefficient of determination ( $R^2$ )

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - y'_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (16)$$

- Deviation ( $Dev$ )

$$Dev(\%) = \frac{P_i - P_j}{P_j} * 100 \quad (17)$$

Where  $y_i$  is the target output,  $y'_i$  is the predicted output,  $n$  is the number of samples,  $\bar{y}$  is the mean of the target output.  $Dev$  indicates the statistical performance improvement compared with a benchmark model,  $P_i$  is the statistical performance (MAE, RMSE or  $R^2$ ) of an individual or ensemble method,  $P_j$  is the corresponding performance of a benchmark model, LR or a individual method used in the ensemble method as the base learner, respectively.

MAE statistics is a measure of errors between the predicted values (the estimated value of crack width of ECC after self-healing) with the estimated values (the observed value of crack width of ECC after self-healing in empirical data). RMSE statistics computes the square root of the average residual error between the predicted values and the target values. A lower value of MAE or RMSE indicates a better prediction performance of the model.  $R^2$  measures the strength of association between the predicted values and the target values, based on the proportion of total variation of outcomes. A greater value close to 1 represents a better prediction performance that commendably replicates the observed crack width of ECC after self-healing. Deviation statistics indicates the improvement of the prediction performance of a individual or an ensemble model from a benchmark model that can be the LR model or the individual model used as base learners in the corresponding ensemble model.

## 5 Results and Discussion

In this section, the prediction performance of individual and ensemble methods are examined by MAE, RMSE and  $R^2$  according to ten-fold cross-validation. Where we use the abbreviations for ease of presentation, in which Bag\_LR, Bag\_SVR, Bag\_BPNN and Bag\_CRAT express generating bagging ensemble method by using LR, SVR, BPNN and CRAT as base estimators, respectively. Ada\_LR, Ada\_SVR, Ada\_BPNN and Ada\_CRAT present generating AdaBoost ensemble method by manipulating LR, SVR, BPNN and CRAT as base estimators, respectively. Stack\_LR indicates developing stacking ensemble method based on individual methods (including SVR, BPNN, and CRAT) as learning base with using LR as a meta-regressor.

## 5.1 Prediction performance of the proposed models

Table 5 shows the average performance of ten-fold cross-validation for each model (in each row), and performance deviation from the LR model with respect of MAE, RMSE and  $R^2$ , respectively. As it can be seen, Stack\_LR is superior to all other individual or ensemble models on the basis of all three performance measures (3.934, 6.118, 0.904 for MAE, RMSE, and  $R^2$ , respectively). Furthermore, among the individual models, SVR performs the best in terms of MAE (4.296), and BPNN has the lowest error on RMSE (6.515) and highest accuracy with  $R^2$  0.899. For the single learning based ensemble methods, Bag\_CRAT has the best performance in terms of MAE (4.093), and Bag\_BPNN performs the best on RMSE (6.341). In terms of  $R^2$ , Bag\_CRAT and Bag\_BPNN have the same performance (0.901) and outperform other ensemble methods using an individual method as base estimator. Moreover, it can be concluded that most machine learning models are able to learn and predict empirical data with an acceptable degree of precision. The performance of all machine learning models described in Table 5 are depicted in Figure 9 (a), (b), and (c) in terms of MAE, RMSE and  $R^2$ , respectively.

Table 5: Average performances of machine learning models for self-healing prediction of ECC

	Models	MAE	Dev(%)	RMSE	Dev(%)	$R^2$	Dev(%)
Individual models	LR	5.012	-	7.680	-	0.860	-
	BPNN	4.329	-13.6	6.515	-15.2	0.899	4.5
	CRAT	4.305	-14.1	6.811	-11.3	0.887	3.1
	SVR	4.296	-14.3	6.826	-11.1	0.883	2.7
Ensemble models	Ada_LR	4.784	-4.6	7.400	-3.6	0.867	0.8
	Ada_BPNN	4.226	-15.7	6.435	-16.2	0.900	4.7
	Ada_CRAT	4.207	-16.1	6.455	-15.9	0.898	4.4
	Ada_SVR	4.145	-17.3	6.577	-14.4	0.893	3.8
	Bag_LR	5.014	0.0	7.689	0.1	0.860	0.0
	Bag_BPNN	4.143	-17.3	6.341	-17.4	0.901	4.8
	Bag_CRAT	4.093	-18.3	6.358	-17.2	0.901	4.8
	Bag_SVR	4.302	-14.2	6.820	-11.2	0.883	2.7
	Stack_LR	3.934	-21.5	6.118	-20.3	0.904	5.1

Overall, all models can noticeably reduce the error values and increase the prediction accuracy compared with LR, except Bag\_LR. For example, of the models based on ensemble learning by AdaBoost, Ada\_SVR performs the best on MAE, reducing by 17.3%, Ada\_BPNN performs the best on RMSE, reducing by 16.2% and on  $R^2$ , increasing by 4.7%. For the bagging ensemble methods, Bag\_CRAT performs the best on MAE, reducing by 18.3%, Bag\_BPNN performs the best on RMSE, reducing by 17.4%, and Bag\_CRAT and Bag\_BPNN have the same performance on  $R^2$ , increasing by 4.8%. However, note that Bag\_LR has a worse performance than LR in terms of MAE and RMSE. This may result from several training examples repeatedly appearing in different replicated datasets to train multiple base regressors.

Table 6 further shows the deviation of ensemble models from individual models which are used as base learners over the three statistical measures based on ten-fold cross-validation results. The results indicate that the ensemble methods using individual methods as a learning base improved the performance of the individual method on the basis of all three performance measures. For example, using BPNN as the learning base, ensemble method Bag\_BPNN has the lower error (MAE and RMSE decreasing by 4.3% and 2.7%, respectively) and higher accuracy ( $R^2$  increasing by 0.2%) than the individual method BPNN. Moreover, the comparative results showed that ensemble learning based models (Stack\_LR) distinctly outperformed the single learning based models (Ada\_LR and Bag\_LR) in terms of overall performance measures.

Specifically, bagging ensembles substantially enhance the performance of BPNN and CRAT while the AdaBoost ensemble achieve a considerable improvement for LR and SVR. It means that BPNN and CRAT

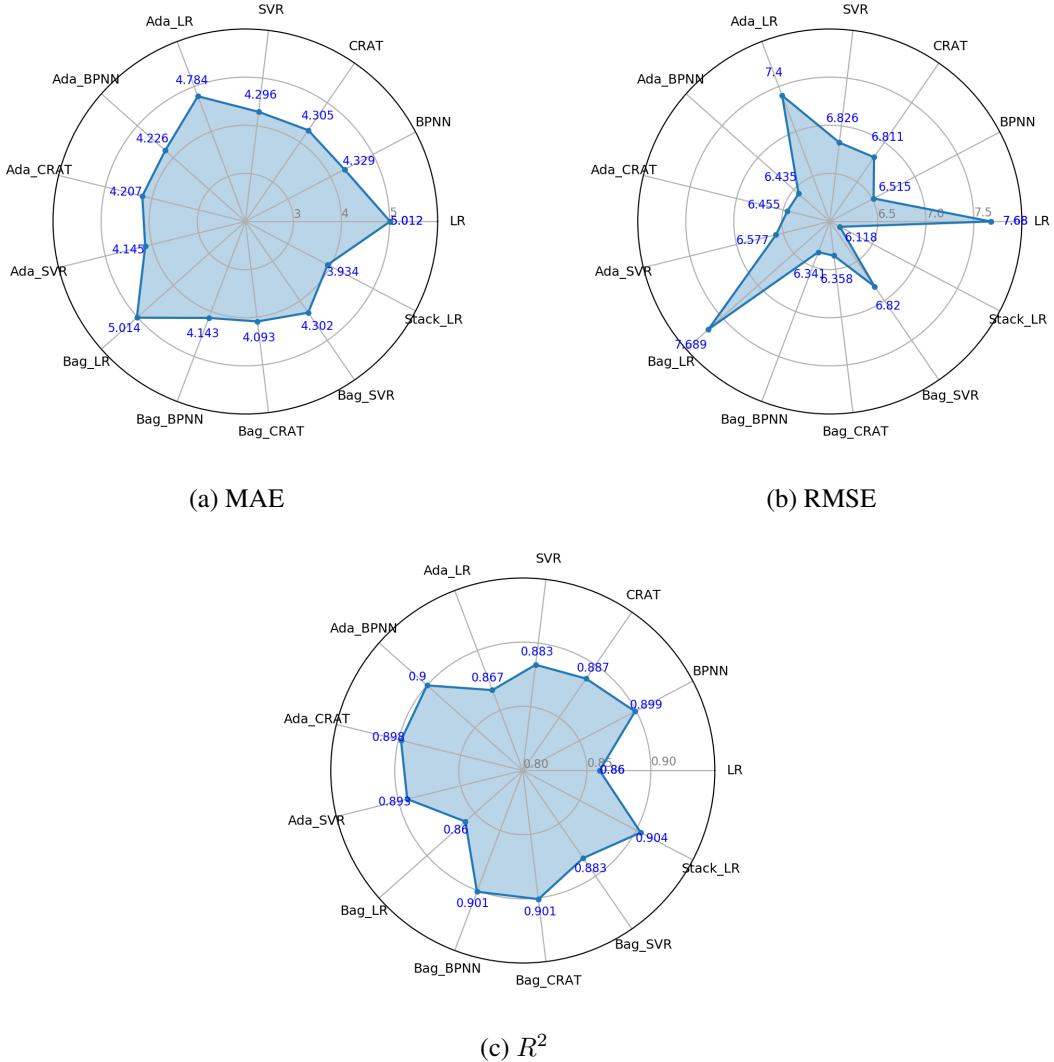


Figure 9: Average prediction performance of 10-fold cross-validation on all machine learning models for predicting self-healing ability of ECC

Table 6: Performance deviation of ensemble models from benchmark models on self-healing of ECC

Benchmark	Model	MAE	RMSE	$R^2$	Benchmark	Model	MAE	RMSE	$R^2$
		<i>Dev(%)</i>					<i>Dev(%)</i>		
LR	Ada_LR	-4.6	-3.6	0.8	LR	Bag_LR	0.0	0.1	0.0
BPNN	Ada_BPNN	-2.4	-1.2	0.1	BPNN	Bag_BPNN	-4.3	-2.7	0.2
CRAT	Ada_CRAT	-2.3	-5.2	1.2	CRAT	Bag_CRAT	-4.9	-6.6	1.6
SVR	Ada_SVR	-3.5	-3.6	1.1	SVR	Bag_SVR	0.1	-0.1	0.0
Ada_LR	Stack_LR	-17.8	-17.3	4.3	Bag_LR	Stack_LR	-21.5	-20.4	5.1

used as the base learner, bagging method achieves better results (lower error value and higher  $R^2$ ) than the AdaBoost and individual method, and the LR and SVR used as the base learner, the AdaBoost method gets better results than the bagging and individual method.

## 5.2 Comparison of predicted and observed crack width

The comparison of the proposed ML models for the prediction outputs is illustrated in Figure 10 to 13, in which the predicted crack width after self-healing by ML models and the experimental observed crack width after self-healing are first presented in sub-figure (a). In the rest of sub-figures, the error of predicted crack width over experimental observed crack width after self-healing is displayed as the measurement index based on the group of crack width before self-healing. Mostly, the smaller crack width before self-healing crack width after self-healing

Furthermore, a horizontal line located at the vertical coordinate of zero was taken as the target line in the rest of sub-figures. Generally, the error bar is shorter (indicates a closer distance from the target line) demonstrates the predicted crack width is more accurate.

It is can be seen from the Figure 10 that individual models, BPNN, CRAT and SVR

The comparison of observed and predicted crack width after self-healing of ECC as the self-healing capability of ECC by all machine learning techniques is presented in Figures ???. For example, plot (a) presents the predicted crack width after self-healing of ECC by LR model, which has  $R^2$  coefficient for 0.860 and the slope of the straight line for 0.799. As it can be seen, the Stack\_LR has the best performance and obtained a better fit to a straight line with the slope of 0.899 and  $R^2$  coefficient of 0.904. In Stack\_LR, the predicted value of crack width after self-healing of ECC adheres to the line ( $Y = X$ ) better than other models, which indicates more accuracy for predicting self-healing of ECC. In addition, Figures ?? reveal the empirical data of crack width after self-healing of ECC generated from laboratory experiments is mainly distributed between 0 to  $80\mu m$ . Whereas a few samples have a crack width over  $100\mu m$ . This insufficient data may result in machine learning models being trained inadequately and reducing the prediction accuracy.

The comparative analysis of observed and predicted crack width after self-healing of ECC as the self-healing capability of ECC by all machine learning techniques is presented in Figures ?? to ???. The coefficient of determination  $R^2$  and equation of the linear least square fit line are shown in these Figures (of the form  $y = ax + b$  relating predicted crack width “ $y$ ” after self-healing which is the dependent variable to experimental observed crack width “ $x$ ” after self-healing which is an independent variable) indicates that the prediction output is very close to the experimental observed results. Basically, the more closely the  $R^2$  approaches to one, and with the slope “ $a$ ” of the fit line is closer to 1 and the “ $y$ ” intercept “ $b$ ” is more small, close to 0 (the fit line is more close to the target line  $Y = X$ ), the better the prediction performance is achieved. It can be seen from Figure ??, for the LR model, the predicted crack width after self-healing has a reasonable performance with the coefficient  $R^2$  for 0.860, and the slope “ $a$ ” of the fit line for 0.799 and the “ $y$ ” intercept “ $b$ ” is 3.804. Similarly, as it can be seen in Figure ??, the BPNN model has a better performance than the LR model, which has the coefficient  $R^2$  for 0.899, and the slope “ $a$ ” of the fit line for 0.866 and the “ $y$ ” intercept “ $b$ ” is 1.817. And as shown in Figure ?? and ??, the predicted value of crack width after self-healing of ECC in the BPNN model adheres to the target line  $Y = X$  better than the LR model, especially for the crack width over  $100\mu m$ , which further indicates the BPNN model is more accuracy and outperform the LR model. Specifically, the Stack\_LR model has the best performance with the coefficient  $R^2$  for 0.904, and has the slope “ $a$ ” of the fit line for 0.899 (close to 1) and the “ $y$ ” intercept “ $b$ ” is 1.579 (close to 0) as shown in Figure ???. In addition, Figures ?? to ?? reveal the experimental data of crack width after self-healing of ECC generated from laboratory experiments is mainly distributed between 0 to  $60\mu m$ . Whereas a few samples have a crack width over  $100\mu m$ . This insufficient data may result in machine learning models being trained inadequately and reducing the prediction accuracy.

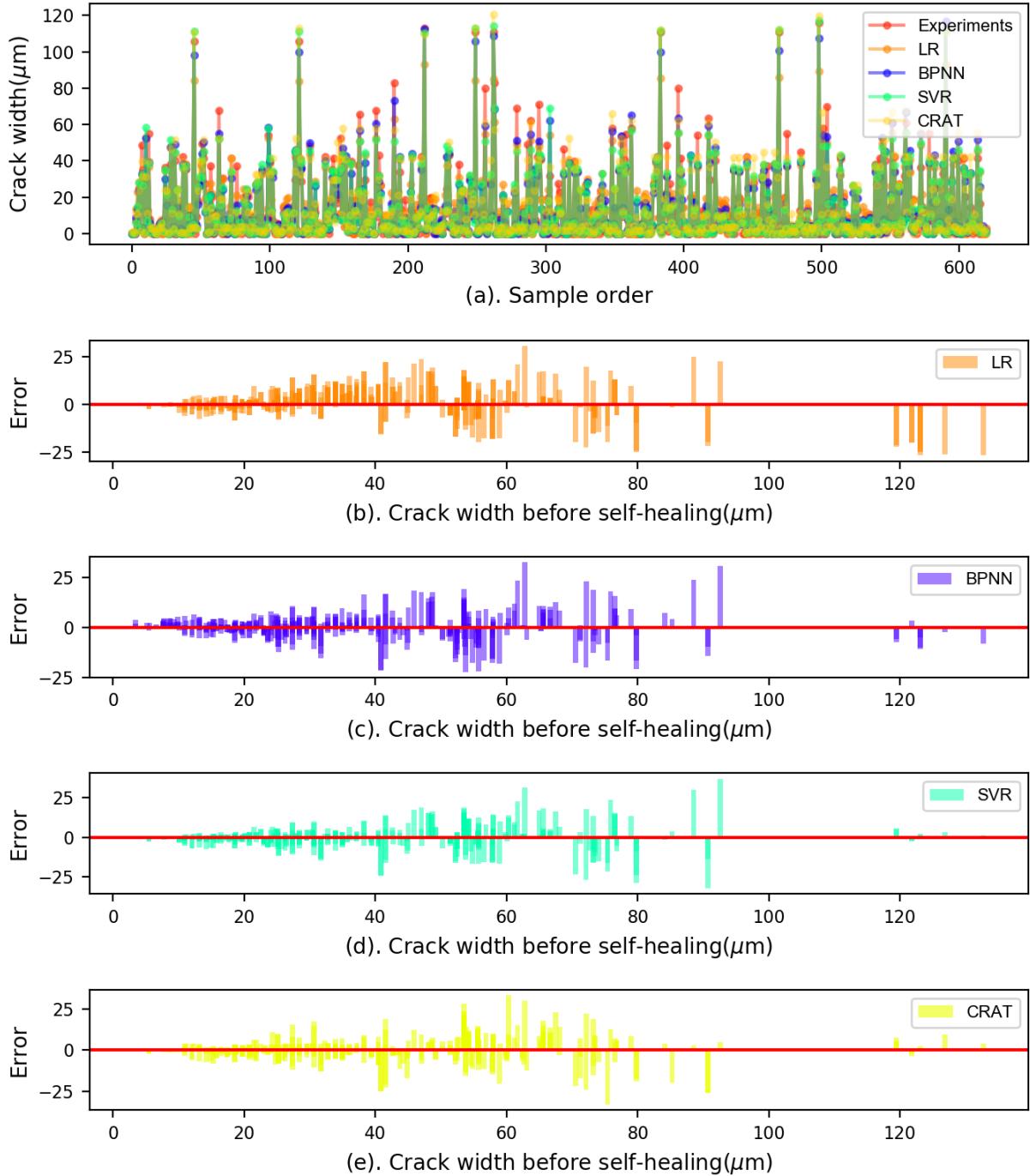


Figure 10: Comparison of predicted and observed crack width after self-healing of ECC for individual models

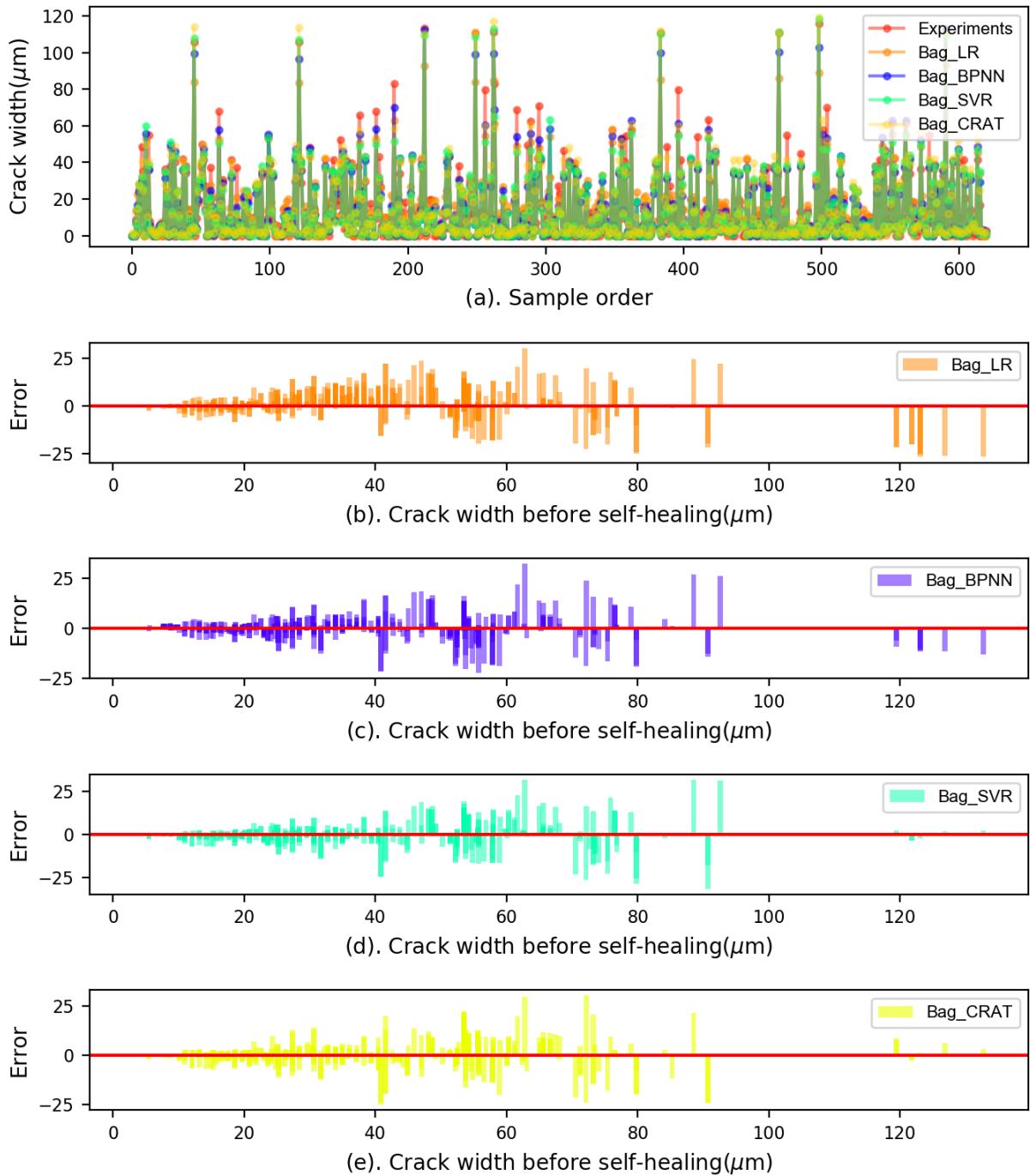


Figure 11: Comparison of predicted and observed crack width after self-healing of ECC for bagging ensemble models

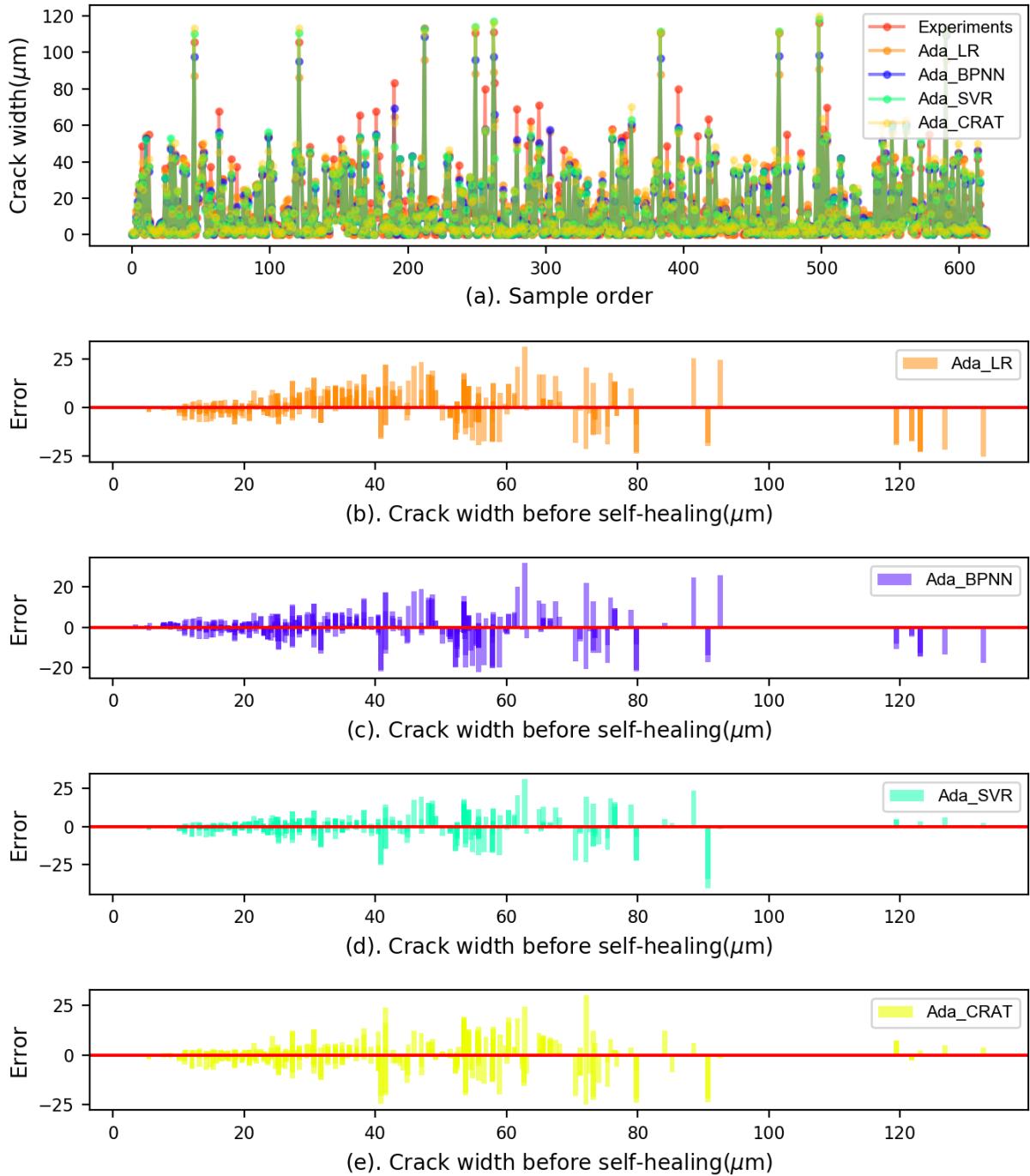


Figure 12: Comparison of predicted and observed crack width after self-healing of ECC for adaBoost ensemble models

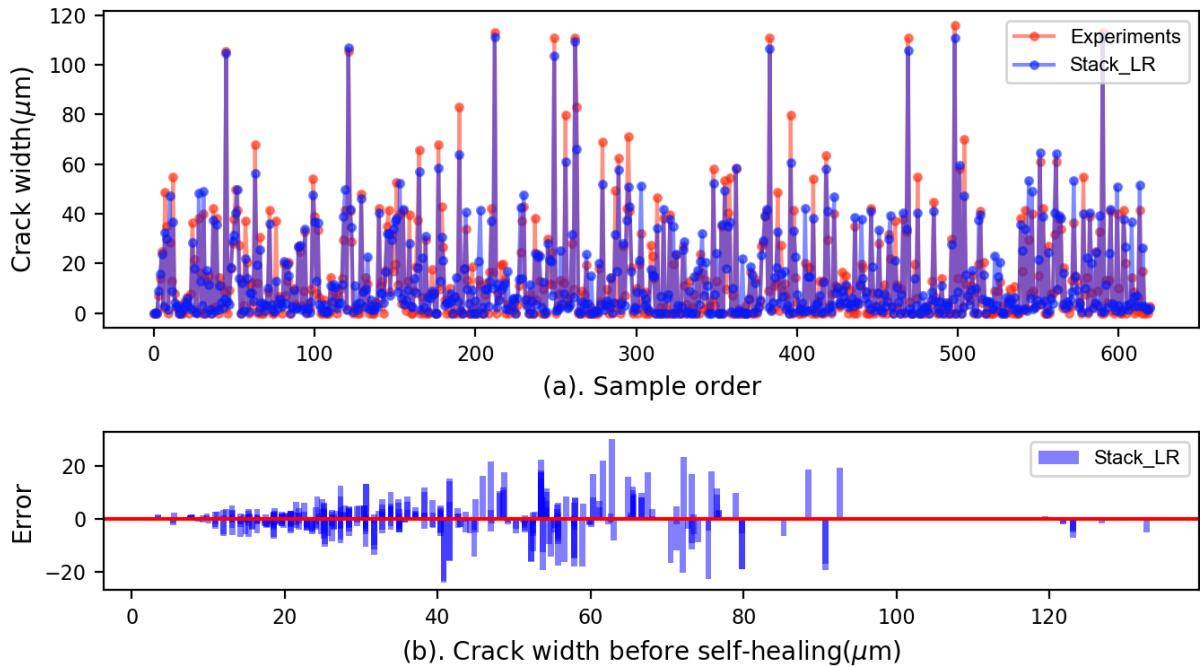


Figure 13: Comparison of predicted and observed crack width after self-healing of ECC for stacking ensemble models

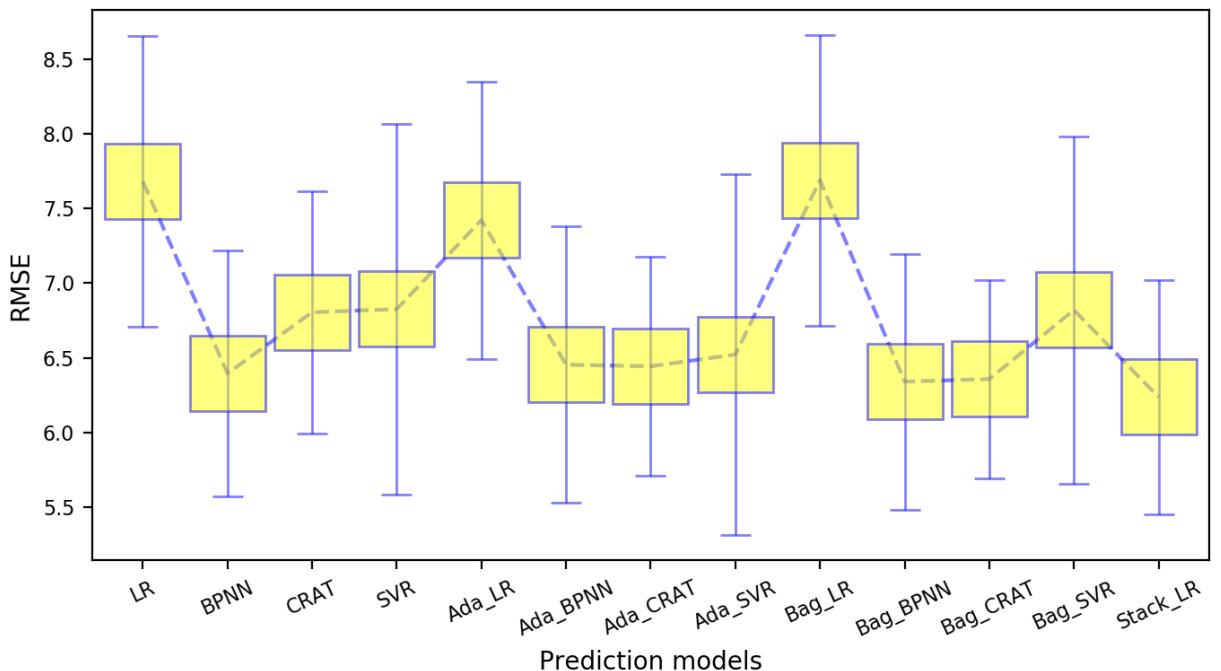


Figure 14: Comparison of predicted and observed crack width after self-healing of ECC for stacking ensemble models

## 6 Conclusions

In this chapter, a comprehensive comparison is proposed to analyse the performance of prediction of the self-healing ability of ECC using individual and ensemble methods. All machine learning models are trained and tested based on the experimental results from nine ECC mixtures. The comparison results provide valuable insights into proposing and validating the machine learning models for predicting self-healing on ECC.

A total of four individual methods (LR, BPNN, CRAT, and SVR) and three ensemble methods (bagging, AdaBoost, and stacking) are used to perform the prediction of the self-healing ability of ECC. In these, the performance of LR is used as a benchmark baseline to compare the performance of other individual and ensemble models. The proposed ensemble methods, bagging and AdaBoost, use one of the four individual methods as a base estimator to aggregate multiple single learning based prediction. The stacking method is used to combine multiple predictions generated by multiple regression models (SVR, BPNN, and CRAT).

The Stack\_LR model demonstrated great predictive ability and the best performance among all individual and ensemble methods, showing the lowest error values on MAE 3.934 and RMSE 6.118 and the highest on  $R^2$  0.904. Moreover, all ensemble methods present the improvement of the predictive ability of individual methods. In particular, the bagging method obviously improves the prediction performance of CRAT on MAE 4.9%, RMSE 6.6%, and  $R^2$  1.6%. On the other hand, the BPNN model performs the best in terms of RMSE and  $R^2$  among all individual models.

The computational results indicate the individual and ensemble methods can be used to predict the self-healing ability of ECC. However, how to choose a fit base learner for different ensemble methods is critical. Engineers engaged in the design of durable and sustainable ECC mixtures will benefit from the use of machine learning models. Also all individual and ensemble models can be used as a tool for the prediction of the properties of ECC and can be optimized to develop hybrid models for further improving predictive ability.

Future investigation and experimentation will be considered to extend the training dataset to include various crack width distributions and diverse influencing factors, such as components, W/MC rate etc., on self-healing of ECC. And more research should be undertaken to explore how to optimize parameters in machine learning models and develop hybrid modelling strategies to improve the prediction accuracy.

## References

- [1] Diane Gardner, Robert Lark, Tony Jefferson, and Robert Davies. A survey on problems encountered in current concrete construction and the potential benefits of self-healing cementitious materials. *Case studies in construction materials*, 8:238–247, 2018.
- [2] E Cailleux and V Pollet. Investigations on the development of self-healing properties in protective coatings for concrete and repair mortars. In *Proceedings of the 2nd International Conference on Self-Healing Materials, Chicago, IL, USA*, volume 28, 2009.
- [3] Ahmed Ramadan Suleiman and Moncef L Nehdi. Modeling self-healing of concrete using hybrid genetic algorithm–artificial neural network. *Materials*, 10(2):135, 2017.
- [4] Waiching Tang, Omid Kardani, and Hongzhi Cui. Robust evaluation of self-healing efficiency in cementitious materials—a review. *Construction and Building Materials*, 81:233–247, 2015.
- [5] Carola Edvardsen. Water permeability and autogenous healing of cracks in concrete. *Materials Journal*, 96(4):448–454, 1999.
- [6] Stefan Jacobsen, Jacques Marchand, and Hugues Hornain. Sem observations of the microstructure of frost deteriorated and self-healed concretes. *Cement and Concrete Research*, 25(8):1781–1790, 1995.

- [7] Hans-Wolf Reinhardt and Martin Jooss. Permeability and self-healing of cracked concrete as a function of temperature and crack width. *Cement and concrete research*, 33(7):981–985, 2003.
- [8] Mustafa Şahmaran and İ Özgür Yaman. Influence of transverse crack width on reinforcement corrosion initiation and propagation in mortar beams. *Canadian Journal of Civil Engineering*, 35(3):236–245, 2008.
- [9] CA Clear. The effects of autogenous healing upon the leakage of water through cracks in concrete. Technical report, 1985.
- [10] Mustafa Sahmaran, Gurkan Yildirim, and Tahir K Erdem. Self-healing capability of cementitious composites incorporating different supplementary cementitious materials. *Cement and Concrete Composites*, 35(1):89–101, 2013.
- [11] Toshiro Kamada and Victor C Li. The effects of surface preparation on the fracture behavior of ecc/concrete repair system. *Cement and Concrete Composites*, 22(6):423–431, 2000.
- [12] Victor C Li and Tetsushi Kanda. Innovations forum: engineered cementitious composites for structural applications. *Journal of Materials in Civil Engineering*, 10(2):66–69, 1998.
- [13] Erdogan Özbay, Mustafa Şahmaran, Mohamed Lachemi, and Hasan Erhan Yücel. Self-healing of microcracks in high-volume fly-ash-incorporated engineered cementitious composites. *ACI Materials Journal*, 110(1), 2013.
- [14] Min Wu, Björn Johannesson, and Mette Geiker. A review: Self-healing in cementitious materials and engineered cementitious composite as a self-healing material. *Construction and Building Materials*, 28(1):571–583, 2012.
- [15] Haoliang Huang, Guang Ye, and Denis Damidot. Characterization and quantification of self-healing behaviors of microcracks due to further hydration in cement paste. *Cement and Concrete Research*, 52: 71–81, 2013.
- [16] Ahmed R Suleiman, Andrew J Nelson, and Moncef L Nehdi. Visualization and quantification of crack self-healing in cement-based materials incorporating different minerals. *Cement and Concrete Composites*, 103:49–58, 2019.
- [17] J Zhou, S Qian, MG Sierra Beltran, G Ye, E Schlangen, and K van Breugel. Developing engineered cementitious composite with local materials. In *International conference on microstructure related durability of cementitious composites, Nanjing, China*, 2008.
- [18] Victor C Li and En-Hua Yang. Self healing in concrete materials. In *Self healing materials*, pages 161–193. Springer, 2007.
- [19] Zhigang Zhang, Shunzhi Qian, and Hui Ma. Investigating mechanical properties and self-healing behavior of micro-cracked ecc with different volume of fly ash. *Construction and Building Materials*, 52: 17–23, 2014.
- [20] Gürkan Yildirim, Mustafa Sahmaran, and Hemn Unis Ahmed. Influence of hydrated lime addition on the self-healing capability of high-volume fly ash incorporated cementitious composites. *Journal of Materials in Civil Engineering*, 27(6):04014187, 2014.
- [21] Y Yang, M Lepech, and Victor C Li. Self-healing of ecc under cyclic wetting and drying. 2005.
- [22] Mustafa Sahmaran, Mo Li, and Victor C Li. Transport properties of engineered cementitious composites under chloride exposure. *ACI Materials Journal*, 104(6):604, 2007.

- [23] SZ Qian, J Zhou, and E Schlangen. Influence of curing condition and precracking time on the self-healing behavior of engineered cementitious composites. *Cement and concrete composites*, 32(9):686–693, 2010.
- [24] Marai M Alshihri, Ahmed M Azmy, and Mousa S El-Bisy. Neural networks for predicting compressive strength of structural light weight concrete. *Construction and Building Materials*, 23(6):2214–2219, 2009.
- [25] Siamak Safarzadegan Gilan, Hamed Bahrami Jovein, and Ali Akbar Ramezanianpour. Hybrid support vector regression–particle swarm optimization for prediction of compressive strength and rcpt of concretes containing metakaolin. *Construction and Building Materials*, 34:321–329, 2012.
- [26] Fei Yan, Zhibin Lin, Xingyu Wang, Fardad Azarmi, and Konstantin Sobolev. Evaluation and prediction of bond strength of gfrp-bar reinforced concrete using artificial neural network optimized with genetic algorithm. *Composite Structures*, 161:441–452, 2017.
- [27] Zaher Mundher Yaseen, Ravinesh C Deo, Ameer Hilal, Abbas M Abd, Laura Cornejo Bueno, Sancho Salcedo-Sanz, and Moncef L Nehdi. Predicting compressive strength of lightweight foamed concrete using extreme learning machine model. *Advances in Engineering Software*, 115:112–125, 2018.
- [28] Kezhen Yan and Caijun Shi. Prediction of elastic modulus of normal and high strength concrete by support vector machine. *Construction and Building Materials*, 24(8):1479–1485, 2010.
- [29] Jui-Sheng Chou, Chih-Fong Tsai, Anh-Duc Pham, and Yu-Hsin Lu. Machine learning in concrete strength simulations: Multi-nation data analytics. *Construction and Building Materials*, 73:771–780, 2014.
- [30] Uwe Reuter, Ahmad Sultan, and Dirk S Reischl. A comparative study of machine learning approaches for modeling concrete failure surfaces. *Advances in Engineering Software*, 116:67–79, 2018.
- [31] Jafar Sobhani, Meysam Najimi, Ali Reza Pourkhorshidi, and Tayebeh Parhizkar. Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and anfis models. *Construction and Building Materials*, 24(5):709–718, 2010.
- [32] Behzad Abounia Omran, Qian Chen, and Ruoyu Jin. Comparison of data mining techniques for predicting compressive strength of environmentally friendly concrete. *Journal of Computing in Civil Engineering*, 30(6):04016029, 2016.
- [33] Luthfi Muhammad Mauludin and Chahmi Oucif. Modeling of self-healing concrete: A review. *Journal of Applied and Computational Mechanics*, 5(Special Issue: Computational Methods for Material Failure):526–539, 2019.
- [34] Xiaoying Zhuang and Shuai Zhou. The prediction of self-healing capacity of bacteria-based concrete using machine learning approaches. *Computers, Materials & Continua* 59 (2019), Nr. 1, 2019.
- [35] Wenjun Wang, Nicolette G Moreau, Yingfang Yuan, Paul R Race, and Wei Pang. Towards machine learning approaches for predicting the self-healing efficiency of materials. *Computational Materials Science*, 168:180–187, 2019.
- [36] M Chaitanya, P Manikandan, V Prem Kumar, S Elavenil, and V Vasugi. Prediction of self-healing characteristics of ggbs admixed concrete using artificial neural network. In *Journal of Physics: Conference Series*, volume 1716, page 012019. IOP Publishing, 2020.

- [37] John Neter, Michael H Kutner, Christopher J Nachtsheim, and William Wasserman. *Applied linear statistical models*, volume 4. Irwin Chicago, 1996.
- [38] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [39] Vladimir Naumovich Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.
- [40] Xu Juncai, Ren Qingwen, and Shen Zhenzhong. Prediction of the strength of concrete radiation shielding based on ls-svm. *Annals of nuclear energy*, 85:296–300, 2015.
- [41] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.
- [42] Yankun Li, Xueguang Shao, and Wensheng Cai. A consensus least squares support vector regression (ls-svr) for analysis of near-infrared spectra of plant samples. *Talanta*, 72(1):217–222, 2007.
- [43] SF Fang, MP Wang, WH Qi, and F Zheng. Hybrid genetic algorithms and support vector regression in forecasting atmospheric corrosion of metallic materials. *Computational Materials Science*, 44(2):647–655, 2008.
- [44] P Yuvaraj, A Ramachandra Murthy, Nagesh R Iyer, SK Sekar, and Pijush Samui. Support vector regression based models to predict fracture characteristics of high strength and ultra high strength concrete beams. *Engineering fracture mechanics*, 98:29–43, 2013.
- [45] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.
- [46] Abhijit Mukherjee and Sudip Nag Biswas. Artificial neural networks in prediction of mechanical behavior of concrete at high temperature. *Nuclear engineering and design*, 178(1):1–11, 1997.
- [47] Hosein Naderpour, Amir Hossein Rafiean, and Pouyan Fakharian. Compressive strength prediction of environmentally friendly concrete using artificial neural networks. *Journal of Building Engineering*, 16:213–219, 2018.
- [48] Leo Breiman. *Classification and regression trees*. 2017.
- [49] S Dan and P Colla. Cart: Tree-structured non-parametric data analysis. *Salford Systems, San Diego, CA*, 1995.
- [50] R Put, C Perrin, F Questier, D Coomans, DL Massart, and Y Vander Heyden. Classification and regression tree analysis for molecular descriptor selection and retention prediction in chromatographic quantitative structure–retention relationship studies. *Journal of Chromatography A*, 988(2):261–276, 2003.
- [51] Dimitrios Frosyniotis, Andreas Stafylopatis, and Aristidis Likas. A divide-and-conquer method for multi-net classifiers. *Pattern Analysis & Applications*, 6(1):32–40, 2003.
- [52] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.
- [53] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

- [54] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156. Citeseer, 1996.
- [55] Sebastian Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to python’s scientific computing stack. *The Journal of Open Source Software*, 3(24), 2018. doi: 10.21105/joss.00638. URL <http://joss.theoj.org/papers/10.21105/joss.00638>.
- [56] Joseph Sill, Gábor Takács, Lester Mackey, and David Lin. Feature-weighted linear stacking. *arXiv preprint arXiv:0911.0460*, 2009.
- [57] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.
- [58] Ofer Dor and Yaoqi Zhou. Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. *Proteins: Structure, Function, and Bioinformatics*, 66(4):838–845, 2007.