

Article

Prediction of Self-Healing of Engineered Cementitious Composite Using Machine Learning Approaches

Guangwei Chen ^{1,2}, Waiching Tang ^{1,*}, Shuo Chen ³, Shanyong Wang ⁴ and Hongzhi Cui ⁵

¹ School of Architecture and Built Environment, The University of Newcastle, Callaghan, NSW 2308, Australia; guangwei.chen@newcastle.edu.au

² School of Physics and Electronics, Qiannan Normal College of Nationalities, Duyun 558000, China

³ School of Information and Physical Sciences, The University of Newcastle, Callaghan, NSW 2308, Australia; sylvia.chen@uon.edu.au

⁴ School of Engineering, The University of Newcastle, Callaghan, NSW 2308, Australia; shanyong.wang@newcastle.edu.au

⁵ College of Civil and Transportation Engineering, Shenzhen University, Shenzhen 518060, China; h.z.cui@szu.edu.cn

* Correspondence: patrick.tang@newcastle.edu.au; Tel.: +61-249-217-246

Abstract: Engineered cementitious composite (ECC) is a unique material, which can significantly contribute to self-healing based on ongoing hydration. However, it is difficult to model and predict the self-healing performance of ECC. Although different machine learning (ML) algorithms have been utilized to predict several properties of concrete, the application of ML on self-healing prediction is considerably rare. This paper aims to provide a comparative analysis on the performance of various machine learning models in predicting the self-healing capability of ECC. These models include four individual methods, linear regression (LR), back-propagation neural network (BPNN), classification and regression tree (CART), and support vector regression (SVR). To improve prediction accuracy, three ensemble methods, namely bagging, AdaBoost, and stacking, were also studied. A series of experimental works on the self-healing performance of ECC samples was conducted, and the results were used to develop and compare the accuracy among the ML models. The comparison results showed that the Stack_LR model had the best predictive performance, showing the highest coefficient of determination (R^2), the lowest root-mean-squared error (RMSE), and the smallest prediction error (MAE). Among all individual models studies, the BPNN model performed the best in terms of the RMSE and R^2 , while SVR performed the best in terms of the MAE. Furthermore, SVR had the smallest prediction error (MAE) for crack widths less than 60 μm or greater than 100 μm , while CART had the smallest prediction error (MAE) for crack widths between 60 μm and 100 μm . The study concluded that the individual and ensemble methods can be used to predict the self-healing of ECC. Ensemble models were able to improve the accuracy of prediction compared to the individual model used as their base learner, i.e., a 2.3% to 4.9% reduction in MAE. However, selecting an appropriate individual and ensemble method is critical. To improve the performance accuracy, researchers should employ different ensemble methods to compare their effectiveness with different ML models.



Citation: Chen, G.; Tang, W.; Chen, S.; Wang, S.; Cui, H. Prediction of Self-Healing of Engineered Cementitious Composite Using Machine Learning Approaches. *Appl. Sci.* **2022**, *12*, 3605. <https://doi.org/10.3390/app12073605>

Academic Editor: Young Hoon Kim

Received: 4 March 2022

Accepted: 26 March 2022

Published: 1 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Currently, the issues associated with cracking in concrete experienced by clients, design team members, and contractors were more than any other problems according to Materials for Life (M4L) [1]. Moreover, cracks are primarily responsible for the reduction of the strength and stiffness of concrete. In European countries, the annual cost spent on maintenance, refurbishment, and repair of concrete cracks in extending the service life of infrastructure is estimated to be around 50% of their annual construction budget [2]. It has been suggested by M4L that self-healing materials have great potential to address the

problems associated with concrete cracking and thereby reducing the maintenance costs over a structure's lifetime [1].

The inspiration of self-healing comes from the biomimicry concept and the healing process in living nature [3]. For example, the skin of humans or animals can biologically repair itself from simple injuries. In cement-based materials, the process of crack self-healing can be categorized into two major mechanisms, i.e., 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, whereas the self-healing mechanism of the latter is triggered by some biological agents, such as bacteria that are deliberately introduced into the cementitious matrix.

Generally, the autogenous self-healing of concrete is mainly controlled by two mechanisms including (1) further hydration of cement particles and/or swelling of calcium silicate hydrate and (2) calcium hydroxide carbonation [5,6]. It has been reported that crack widths of 10 µm [7], 100 µm [8], 200 µm [9], 205 µm [5], and 300 µm [10] of engineered cementitious composite (ECC) can be self-healed completely [11].

ECC is a high-performance fiber-reinforced cementitious composite, and its matrix design is strongly associated with the autogenous self-healing mechanism [12]. ECC features high tensile ductility with a typical fiber volume fraction of 2% [13,14] to promote the self-healing ability [4]. However, the intrinsic self-healing ability of ECC is complex and difficult to predict because of different mineral admixture types, interactivity between different composites in the cementitious matrix and its interaction with the exposed environment [15], and unpredictable crack location, orientation, and width [16]. Previous studies have explored the influence of several factors such as limestone powders (LPs) [17,18], fly ash (FA) [19,20], hydrated lime [21], the water/binder ratio [22], water permeation [23], and different curing conditions (air, carbon dioxide, wet/dry, and water) [24] on the self-healing behavior of ECC. However, these studies did not predict the self-healing efficiency of ECC by modeling their experimental data. In fact, the relationship between multiple factors is complex and non-linear, so it is difficult to predict the self-healing of ECC mathematically based on the available data. Moreover, mathematical models based on empirical data are generally in regression forms, which cannot be used when the problem (e.g., prediction of the self-healing potential of ECC) contains too many independent variables and requires more assumptions to be made [25].

To account for the drawbacks of using mathematical models, machine learning (ML) techniques have been used for solving many civil engineering problems with multiple variables [26]. They are model-free approaches that do not rely on predefined models [27]. Many research works have been conducted using ML algorithms for the prediction of various properties of concrete [28,29]. Gilan et al. [30] developed a hybrid support vector regression (SVR)-particle swarm optimization (PSO) algorithm model to predict the compressive strength and chloride ion penetrability of concretes containing metakaolin. Yan et al. [31] predicted the bond strength of a glass-fiber-reinforced polymer bar in concrete by an artificial neural network (ANN) with the genetic algorithm (GA). Yaseen et al. [32] proposed an ML method called extreme learning machine (ELM) to predict the compressive strength of lightweight foamed concrete.

In the literature, the performance of various ML algorithms in predicting concrete properties have been evaluated and compared. Yan and Shi [33] reported that SVR is better than other individual methods in predicting the elastic modulus of normal and high-strength concrete. Chou [34] compared the performance of several individual and ensemble methods for predicting the mechanical properties of high-performance concrete. The results revealed that ensemble learning strategies outperform individual learning techniques in predicting the compressive strength of high-performance concrete. Reuter et al. [27] employed three different individual approaches for modeling concrete failure surfaces. They found that all three individual approaches are able to fit the experimental data with low error. Sobhani et al. [35] suggested that their proposed fuzzy inference system and ANN are more reliable than traditional regression models in predicting no-slump concrete.

Omran et al. [36] predicted the compressive strengths of an environmentally friendly concrete by using three individual methods, two ensemble methods, and four regression tree models. Their results showed that the individual Gaussian process regression model and its ensemble models outperformed other models.

Although different ML algorithms have been utilized to predict several properties of concrete, the application of ML on self-healing prediction is considerably rare. Recently, Mauludin and Oucif [37] reviewed the common methods used for modeling autogenous self-healing of concrete and stated that the methods can be classified into two categories: (1) numerical simulation and (2) ML. However, the only ML model reviewed in their study was the GA-ANN method proposed by Ramadan et al. [3]. They predicted the self-healing ability of cement-based materials using a dataset collected from the literature. The results showed that the 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 of cement-based materials.

Chaitanya et al. [38] used an ANN model to predict the self-healing property of concrete containing ground granulated blast furnace slag in terms of compressive strength recovery based on 51 samples collected from their experimental studies. Generally, the predicted results by the ANN model were in good agreement with the experimental values. Zhuang and Zhou [39] conducted a comparative study on six ML algorithms including SVR, decision tree regression (DTR), gradient boosting regression (GBR), ANN, Bayesian ridge regression (BRR), and kernel ridge regression (KRR) for the crack-repairing ability of the bacteria-based self-healing concrete. The results showed that GBR performed much better than other models with R^2 values of 0.93 and 0.74 for the training set and testing set, respectively. However, the R^2 values of most models were less than 0.7 on both the training and testing sets. Although extensive experiments with different combinations of influencing variables were utilized to generate the empirical dataset, their study only selected three variables including the number of bacteria, the healing time, and the initial crack width to predict the crack closure percentage as the output. Huang et al. [40] used six types of machine learning algorithms to predict the healing performance of self-healing concrete. The data were taken from the open literature; however, different studies used different self-healing indicators (e.g., crack width, permeability, or mechanical properties) to assess self-healing; therefore, their study lacks a specific analysis and discussion on the efficiency of the models for predicting a specific indicator. Ahmad et al. [41] explored several ML algorithms, including support vector machine (SVM), random forest (RF), AdaBoost, and k-nearest neighbor (KNN) for predicting the shear strength of rockfill materials. The results demonstrated that the SVM achieved the best prediction performance.

To the best of our knowledge, there has been no study to date to predict the self-healing of ECC using the ML approach. It is worthwhile to understand and evaluate the prediction capability of various ML models for the self-healing of ECC. Furthermore, conducting experiments is usually expensive and time consuming; therefore, it is necessary to develop accurate and reliable prediction models for the self-healing ability of ECC. In addition, it is critical to choose an appropriate ML method as each algorithm has a significant effect on the accuracy of the results [42]. Therefore, this study aims to provide a comparative analysis on the performance of various ML models in predicting the self-healing capability of ECC. The ML model with the best performance can be used as a baseline prediction model for developing advanced models in the future.

In this paper, four ML individual methods including linear regression (LR), SVR, back-propagation neural network (BPNN), and classification and regression tree (CART) were proposed to predict the self-healing capability of ECC. To improve the prediction accuracy, three ensemble methods, namely bagging, AdaBoost, and stacking, were used to construct ensemble models using the individual models as the base learners. A series of experimental works on the self-healing performance of ECC samples was conducted, and the results were used to develop and compare the accuracy among the ML models. Experimental data collected from the experiments were first preprocessed and then divided into a 10-fold

cross-validation algorithm (for the details, refer to Section 4.1) to avoid overfitting. Figure 1 summarizes the steps that were performed when predicting the self-healing of ECC.

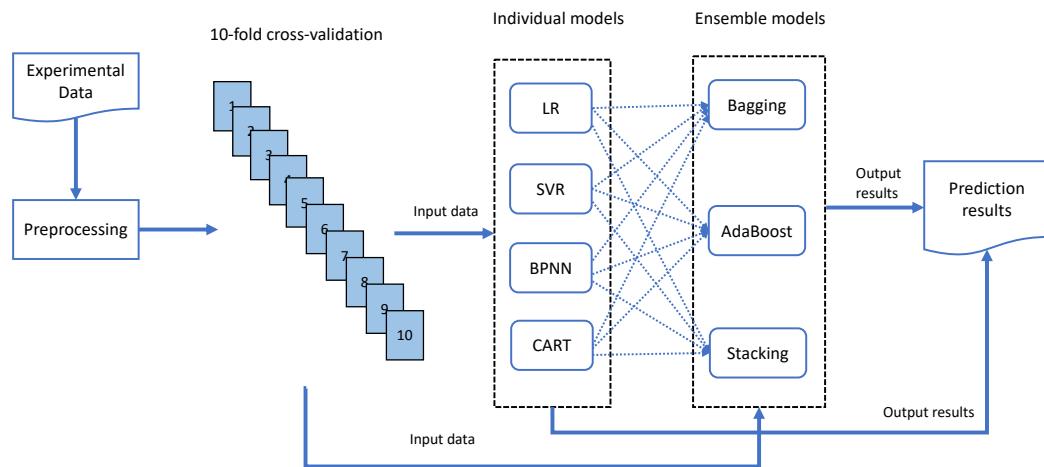


Figure 1. Flowchart of implementing prediction models for the self-healing capability of ECC.

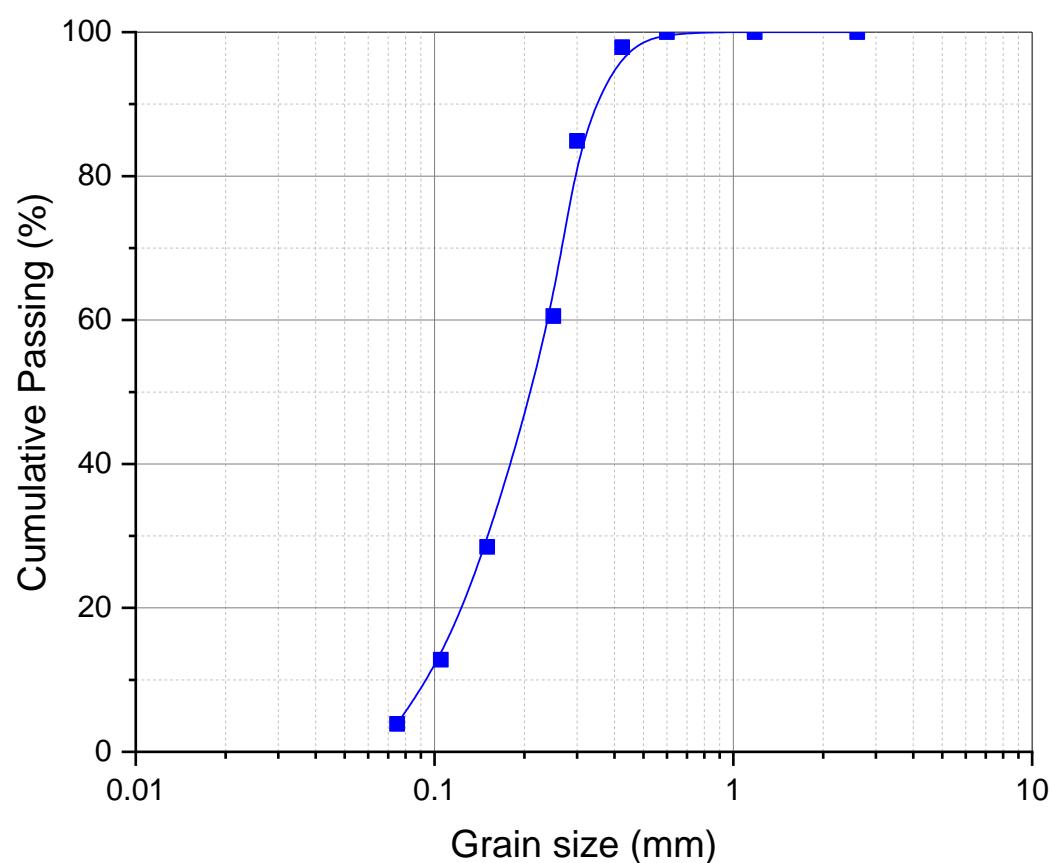
This paper is organized as follows. Section 2 presents the experimental program detailing the materials used for ECC specimen preparation and the test setup for crack data measurement. The concepts and formulations of individual and ensemble models used for predicting the self-healing of ECC are presented in Section 3, whereas the validation and evaluation methods are described in Section 4. In Section 5, the computational results are presented and compared, and the model with the best prediction performance is identified. Finally, Section 6 draws the major conclusions from this work and suggests some directions for future research.

2. Experimental Program

2.1. Materials and Mixture Proportion

In the experimental part, samples of ECC with different mineral admixtures were prepared. The materials including general-purpose cement (GPC), fly ash (FA), silica fume (SF), hydrated lime powder (LP), fine sand, polyvinyl alcohol (PVA) fibers, as well as water and high-range water-reducing admixture (HRWR) were used. The HRWR complies with AS1478.1-2000 and was added at the recommended dosages to achieve 80 ± 20 mm slump (medium degree of workability). GPC and FA were supplied by Boral in accordance with Australian Standard AS 3972-2010 [43], 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 μm sieve and less than 0.05% on a 250 μm sieve. The physical and chemical properties of the cementitious materials are shown in Table 1. Fine sand with an average grain size of 150 μm and a fineness modulus of 2.01 was used. Figure 2 shows the particle distribution of the fine sand. The PVA fibers were supplied by Domocrete, and their mechanical and geometrical properties are described in Table 2.

All ECC mixtures were prepared with a constant water-to-cementitious-material (W/CM) ratio of 0.29 and a constant sand to CM (PC + FA + LP+SF) ratio of 0.36. All fine aggregates were in saturated surface-dried condition prior to mixing. The abbreviations for labeling specimens were adopted in such a way that the letters FA, SF, and LP stand for samples with fly ash, silica fume, and limestone as the binder materials, respectively. The number after the letters shows the percentage of materials in the binder system. For example, the FA70 mixture is related to an ECC sample with binder containing 70% FA by weight, whereas FA60-SF10 is the mixture with 60% FA and 10% SF. A total of nine ECC mixtures were prepared, and the details of the mix proportion are shown in Table 3.

**Figure 2.** Particle size distribution of fine sand.**Table 1.** Physical and chemical properties of cementitious materials.

Chemical Composition (%)	GPC	FA	LP	SF
Silica (SiO_2)	19.8	65.90	1.8	95.10
Alumina (Al_2O_3)	5.3	24.0	0.5	0.21
Iron oxide (Fe_2O_3)	3.0	2.87	0.6	0.29
Calcium oxide (CaO)	64.2	1.59	72.0	-
Magnesia (MgO)	1.3	0.42	1.0	-
R_2O	0.6	1.93	-	-
Sulfur trioxide (SO_3)	2.7	-	-	-
Titanium oxide (TiO_2)	0.28	0.91	-	-
Manganic oxide (Mn_2O_3)	0.22	-	-	-
Zirconia (ZrO_2) + Hafnium (HfO_2)	-	-	-	3.46
Loss on ignition (%)	2.8	1.53	24.0	1.4
Density (g/cm^3)	3.08	2.43	2.25	2.26
Specific surface area (m^2/kg)	-	655	460	1.5×10^4

Table 2. Properties of PVA.

Length (mm)	Length/Diameter Ratio	Young's Modulus (MPa)	Elongation (%)	Tensile Strength (MPa)	Density (g/cm^3)
8	200	42,000	7	1600	1.3

Table 3. Mix proportion of all ECC mixtures.

Mix	Water/cm	Sand	Water	Fiber (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

2.2. Sample Preparation and Crack Measurement

A planetary-type mixer of 50 L capacity was used to produce the ECC specimens. During the mixing process, the solid ingredients including cement, mineral admixtures, and sand were initially placed into the mixer and dry mixed for 30 s. Then, the water with HRWR was added, and the mixture was mixed for 2 min. After that, the PVA fibers were slowly added, and mixing was continued until a uniform distribution of fibers in the mix. After mixing, ECC pastes were cast into standard molds with dimension of 100 mm × 200 mm. The specimens were demolded 24 h after casting and stored in a curing room with a temperature of 23 ± 2 °C and a relative humidity (RH) of $90 \pm 5\%$ for 28 dd. To prepare the self-healing test samples, the cylinder specimens were cut into 50 mm-thick slice samples using a diamond blade saw.

A newly developed splitting tensile test apparatus was used to generate micro-cracks, as shown in Figure 3a. It consisted of a steel frame, top and bottom members, and pre-stressed loading steel plates (5 mm thick) on both sides with loading nuts and wire springs, as shown in Figure 3b. Both steel plates were connected to the steel frame by nuts and wire springs. The specimen was placed inside the steel frame and then pre-stressed by the steel plates from both sides, limiting the propagation and size of cracks and preventing excessive crack growth.

Micro-cracks less than 150 µm were produced by pre-loading the ECC samples up to 70% of their maximum splitting strength. A digital microscope was used to measure the crack width on the surface of the specimens, as shown in Figure 3c. After the pre-loading, the cracked specimens were subjected to wet-dry (W/D) cycles to promote self-healing. Each W/D cycle included 24 h of wetting followed by 24 h of drying in laboratory conditions at 23 ± 2 °C and an RH of $50 \pm 5\%$. After 10 W/D cycles, the cracks were measured again by the digital microscope to examine the extent of crack recovery. Figure 4 illustrates the self-healing of cracks of an ECC specimen before and after 10 W/D cycles.

2.3. Data Collection

Experimental data for prediction were gathered with four features, including the crack width before self-healing (representing the influencing factor of self-healing) and the mineral contents of FA, SF, and LP. It is noteworthy that the factors such as GPC, sand, W/CM, and healing time were kept constant, and hence, they were excluded in the prediction modeling. For each ECC mixture, there were 6 identical test specimens. After pre-loading, the crack widths of the specimens were measured using the digital microscope before and after the 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, as shown in Figure 5. With the newly developed splitting tensile test, the crack propagation for all samples was generally consistent and visually straight, as shown in Figure 4. The cracks observed in the observation zone were used to manifest the self-healing capability of individual samples in this study. The schematic diagram of the measurement is shown in Figure 5. In each observation area, only one crack datum was recorded if the crack width showed little or no change along the vertical force; otherwise, multiple crack data would be collected. For example, more crack data were collected for

FA65-LP5 as more cracks were recorded in the observation zone compared to the other samples, as indicated in Table 4. In total, 617 crack data samples were collected from 9 mixtures to construct the ML training–testing dataset [44]. In previous ML studies on predicting the mechanical [35,45–48] and permeability [49] properties of concrete through machine learning methods, the number of samples is usually less than 600, so the number of samples in the study was considered as sufficient. To assist researchers and engineers who are interested in re-implementing or improving the algorithms and exploring the application of ML in ECC self-healing prediction, the codes developed in this study are published as open-source codes. The raw crack data and full code [44] are available from our open-source project GitHub repository [50]. Table 4 shows the number of collected samples and range of cracks on each sample before and after self-healing.

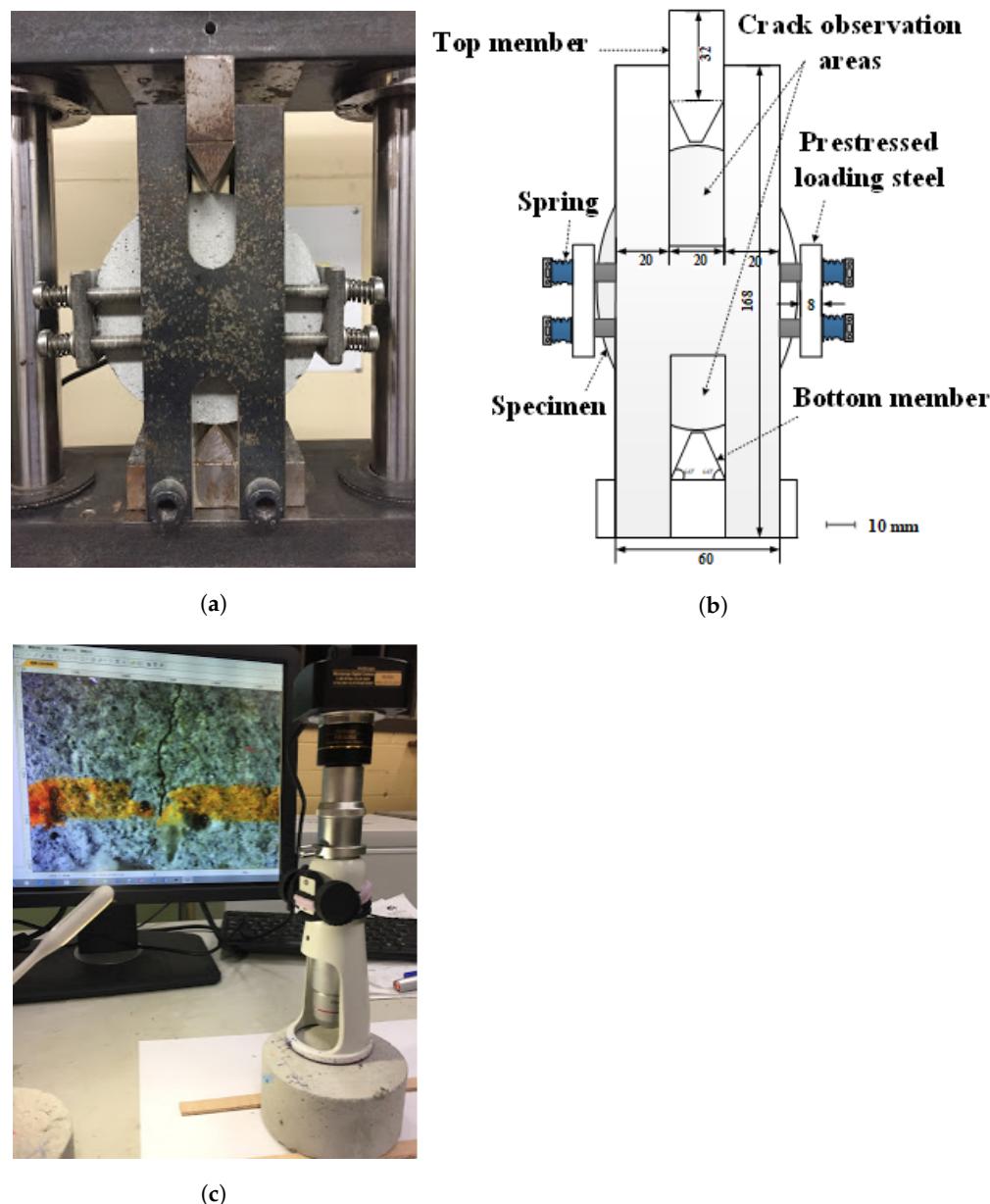
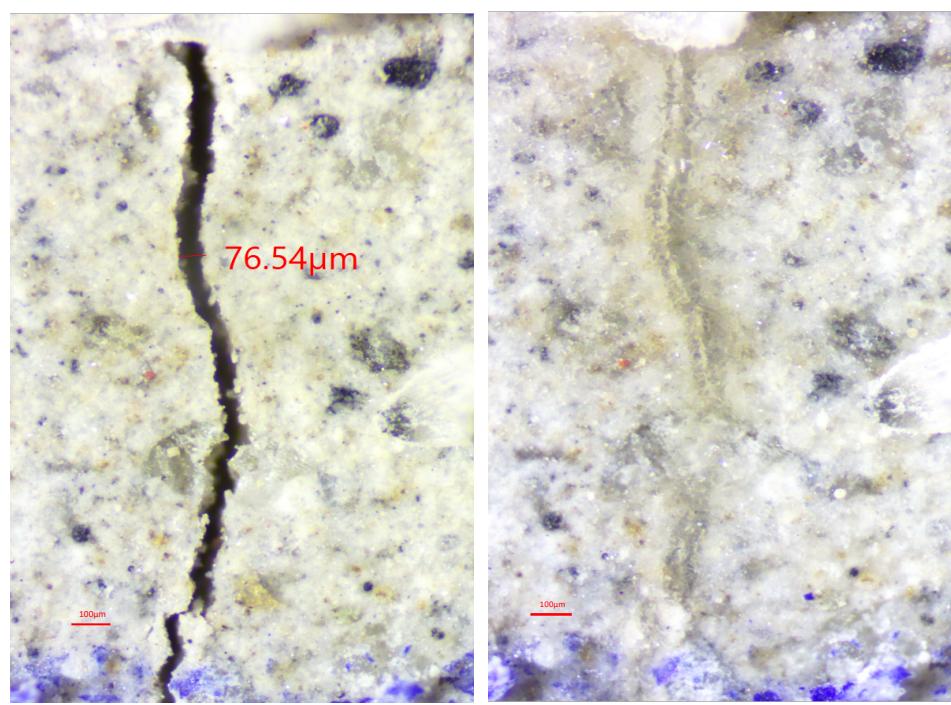


Figure 3. Splitting tensile test apparatus and microscope used in the experiment for creating and measuring ECC cracks. (a) Splitting tensile test apparatus; (b) schematic diagram of the apparatus; (c) crack width measurement.



(a) Crack before self-healing

(b) Crack after self-healing

Figure 4. Surface images of cracked ECC specimens: (a) before and (b) after self-healing.

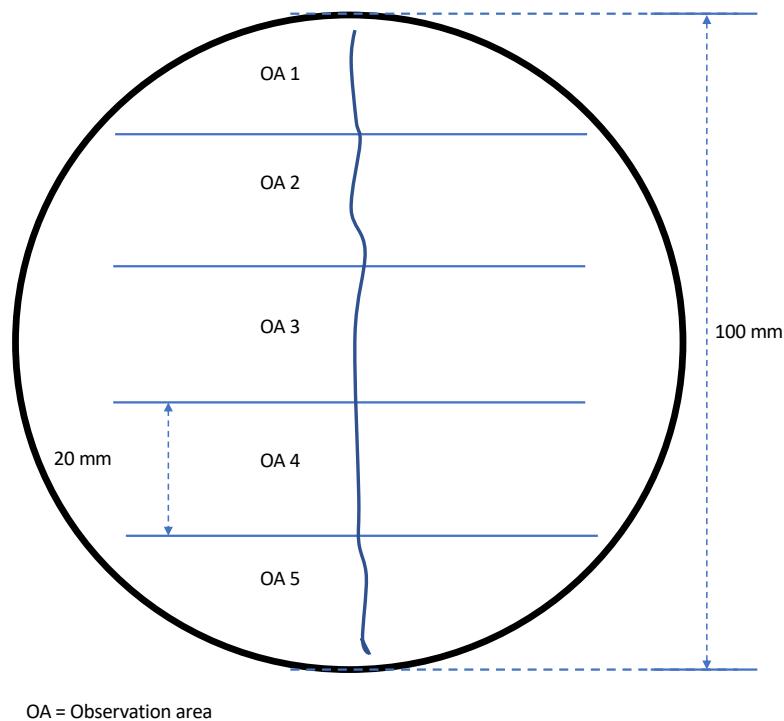


Figure 5. Schematic diagram of measuring observation areas on the surface of ECC mixture specimens.

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

Mix	Number of Cracks	Crack Width before Self-Healing		Crack Width after Self-Healing	
		Min (μm)	Max (μm)	Min (μm)	Max (μm)
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

2.4. Preprocessing of Data

Since the input and output data of different features vary in range and units, the features with a bigger number would steer the model performance. As shown in Table 3, the range of FA varied from 641.16 kg to 816.03 kg, but the range of SF varied from 0 kg to 174.86 kg. Similarly, the range of the crack width varied from 0 μm to 135.47 μm , as shown in Table 4. To eliminate this potential bias, the experimental data were preprocessed through min–max normalization in order to scale the range of all features into [0, 1] with the following equation:

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

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

3. Proposed Machine Learning Models

To predict the self-healing capability of ECC, four individual ML models including LR, SVR, BPNN, and CART and three ensemble methods including bagging, AdaBoost, and stacking were proposed. Ensemble models were constructed using individual models as the base estimators. To establish a baseline for comparison, the modeling parameters were set to be the same in both individual models and ensemble models. The main reason for choosing these techniques was due to their popularity, and some of them are even recognized as the top data mining algorithms in related fields of concrete [34]. 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 [51]. Given our dataset, $T = \{(x_i, y_i), i = 1, 2, \dots, n\}$, where $n = 617$ is the size of the sample dataset, $x_i \in R^n$ are the independent variables representing a sample of selected features from the FA, SF, LP, and crack width before self-healing, R^n is n -dimensional space, and $y_i \in R^1$ is the target output (crack width after self-healing) that corresponds to x_i . Let $d = 4$ denote the number of an independent variable of a random vector $x = \{x_1; x_2; \dots; x_d\}$ and y be the corresponding output (dependent variable). The general formula of LR for predicting the self-healing capability of ECC can be expressed as follows [51]:

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

where $w_i, (i = 1, 2, \dots, d)$ is the regression coefficient, while b is an error term. The prediction performance of LR was used as a benchmark to compare the performance of other individual and ensemble models in this study.

3.2. Support Vector Regression

Support vector machine (SVM) is a supervised machine learning method first introduced by Vapnik [52,53] based on the statistical learning theory [54]. Since then, it has gained popularity due to its attractive features and promising empirical performance [55]. SVM includes two main categories: support vector classification (SVC) and SVR. For classification purposes, SVMs often use a *kernel* function to map the input data as vectors to a high-dimensional feature space so that an optimal separating hyperplane can be constructed [56].

For regression purposes, the basic idea is to provide a nonlinear function by mapping input data into a high-dimensional feature space, where a special type of hyperplane is constructed. After that, a regression model is established in the hyperplane [57].

Given our dataset, $T = \{(x_i, y_i), i = 1, 2, \dots, n\}$, where $n = 617$ is the size of the sample dataset, $x_i \in R^n$ is the input vector representing the selected features of a sample, including the FA, SF, LP, and crack width before self-healing, R^n is the n -dimensional vector space, and $y_i \in R^1$ is the target output indicating the crack width after self-healing that corresponds to x_i . The SVR aims to seek an optimum regression function $f(x)$ with minimal empirical risk, which can be expressed as follows [53]:

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

where $\langle \cdot, \cdot \rangle$ is denoted as the dot product in T and 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 ϵ -insensitive loss function, penalizing predictions that have a distance between the predicted crack width and the target crack width when the self-healing is greater than ϵ . Therefore, the problem of finding w and b to reduce the empirical risk with respect to an ϵ -insensitive loss function is equivalent to the convex optimization problem that minimizes the margin (w) with the full prediction error within the range of ϵ . Then, this problem can be expressed as [53]:

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

By introducing slack variables ξ and ξ_i^* to allow some errors to cope with infeasible solutions of the optimization problem, the formulation can be generated as [53]:

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

The constant C is the penalty value imposed on predictions that lies outside the ϵ margin. Lagrange multipliers are included to solve this problem. By constructing the objective function and all constraints, a dual set of variables is introduced as follows [58]:

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

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 expressed as [58]:

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

As for the nonlinear regression, the input data have to be mapped into a high-dimensional feature space, in which the 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 [59]:

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

Different SVM algorithms use different kinds of kernel functions such as linear, polynomial, radial basis function, and sigmoid kernel. In this work, the Gaussian radial basis function (RBF) was chosen, which is defined as [59]:

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

3.3. Artificial Neural Network

The artificial neural network (ANN), also called a neural network, 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 [60]. The neurons are fully interconnected between the neighboring layers by the weight, and typically, there are no inter-connections between neurons within the same layer [61].

There are many possible network structures available. BPNN was utilized in this study because the back-propagation (BP) algorithms are the most widely used and effective learning algorithms for training an ANN. A preliminary architecture of the BPNN was 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, with 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 capable of approximating most continuous functions, of which the complex nonlinear relationship could be approximated in accuracy [31]. The architecture of the BPNN model for predicting self-healing is demonstrated in Figure 6.

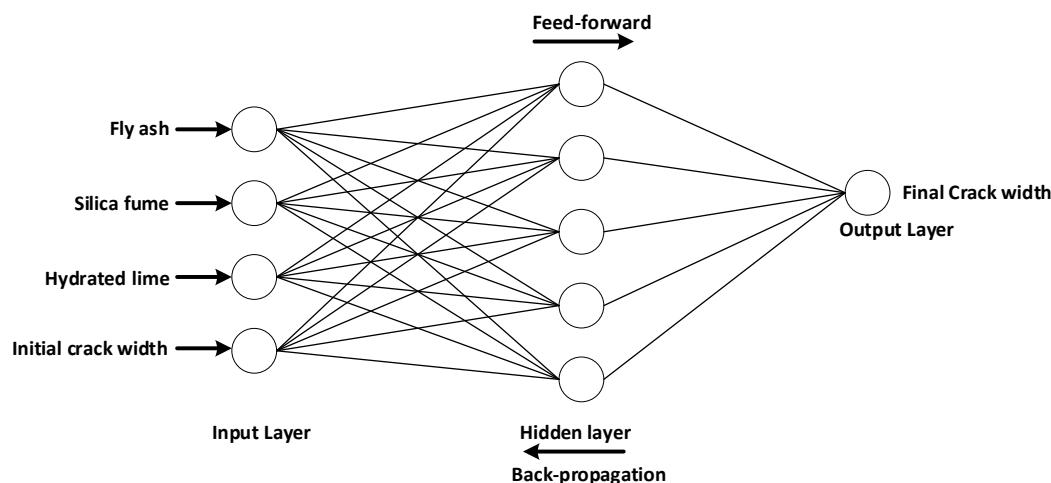


Figure 6. Schematic diagram of the BPNN model for predicting the self-healing of ECC.

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 is multiplied by the respective weights added by a bias and summed together. After that, an activation function f is applied to form the output z , which can be expressed in the following equation [25]:

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

where w_{ij} is the connection weights between the i th neuron of the input and the j th neuron of the hidden layer and b_j is the bias of the j th neuron. The sigmoid function was applied as the 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 an iterated method to adjust the weights from the output layer to the input layer. At first, the outputs were calculated in a feed-forward from the input layer via the hidden layer to the output layer. Then, an error was generated by comparing the output with the target output. After that, the error was back-propagated to the hidden layer and input layer. By adjusting the connection weights and biases, the error was further reduced. The process was repeated until the error was minimized or reaching the termination to avoid over-fitting [62].

3.4. Classification and Regression Tree

CART [63] is a tree decision algorithm that splits data into mutually exclusive subgroups based on a recursive binary partitioning procedure. It develops the relationship between the target variables (the crack width after the self-healing of ECC) and the independent variables (the input features of FA, SF, LP, and crack width before the self-healing of ECC) to create decision rules to form subgroups as branches and leaves, as shown in Figure 7. The process of CART starts from the root node, which contains the entire dataset 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. CART's tree can be either a classification tree [64] or a regression tree [65] depending on the type of target and independent variables, which may be categorical or numerical.

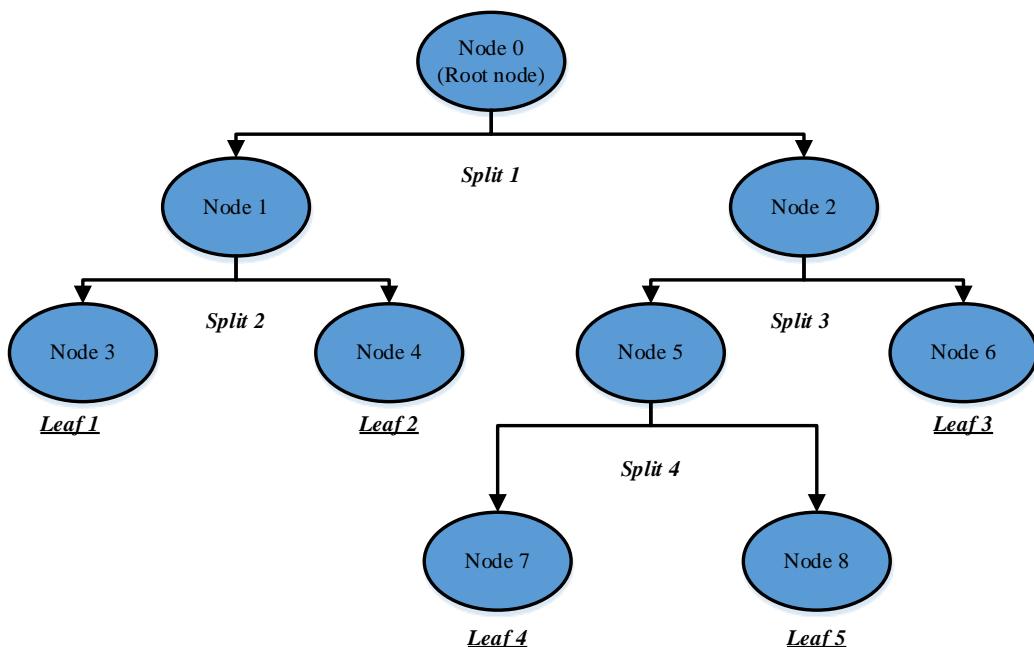


Figure 7. Structure of the classification and regression tree [65].

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

Given a dataset has K classes and the probability of a record in the dataset that belongs to class i is p_i , $i \in \{1, 2, 3, \dots, K\}$, the *Gini* impurity can be expressed as [63]:

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

3.5. Ensemble Methods

In contrast to many ML approaches such as SVM and CART (which develop a single learner from training data), ensemble methods train multiple base learners and combine them [34] to improve the generalizability over a single estimator. Therefore, weak learners (base learners) can be boosted to become strong learners [66] in an ensemble method. The base learners in an ensemble are developed from an individual learning algorithm such as decision tree, SVM, or other kinds of learning algorithms. Breiman [67] showed that ensemble methods are usually more accurate than individual learning methods.

The input features of FA, SF, LP, and crack width before the self-healing of ECC were considered as the d -dimensional predictor variable X , whereas the crack widths after the self-healing of ECC were the one-dimensional output Y . Each estimator used an individual algorithm to provide one estimated function $g(\cdot)$. The output presented by ensemble-based function $g_{en}(\cdot)$ was obtained by a linear combination of individual functions. This ensemble approach can be expressed mathematically as [34]:

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

where c_j expresses the combination coefficients, dependent on the ensemble models used.

3.5.1. Bagging

Bagging method (bootstrap aggregating) can generate multiple versions of a predictor to obtain an aggregated predictor [68]. It generates multiple models independently of different versions of the datasets via random bootstrapping of the original training set. In other words, several training examples could repeatedly appear in different bootstrap replicates. Then, the individual predictions are aggregated through a combination method (either voting or averaging) to form the final prediction. The bagging method can be used to reduce the variance of a base estimator (e.g., a regression tree), by introducing randomization into its construction procedure and making an ensemble out of it. This study used four individual models to build bagging ensemble models including an LR bagging ensemble model (abbreviated as Bag_LR), an SVR bagging ensemble model (abbreviated as Bag_SVR), a BPNN bagging ensemble model (abbreviated as Bag_BPNN), and a CART bagging ensemble model (abbreviated as Bag_CART).

3.5.2. AdaBoost

Similar to bagging, the AdaBoost method [69] manipulates the training examples to generate multiple predictions to form the final prediction. The main difference with bagging is that AdaBoost applies a weight to each of the training examples. In each iteration, the weights are individually updated to minimize the weighted error on the training set. For example, the weights on those training examples incorrectly predicted in previous iteration increase, whereas the weights of the correctly predicted training examples decrease. Therefore, AdaBoost tends to construct progressively more difficult learning problems in subsequent iterations. Once the training process has finished, the predictions are combined through a weighted majority vote (or sum) to produce the final prediction. Therefore, the final classifier usually can achieve a high degree of accuracy in the test set.

By combining four individual models as base estimators in AdaBoost, this study obtained four AdaBoost ensemble models. They were an LR AdaBoost ensemble model (abbreviated as Ada_LR), an SVR AdaBoost ensemble model (abbreviated as Ada_SVR), a BPNN AdaBoost ensemble model (abbreviated as Ada_BPNN), and a CART AdaBoost ensemble model (abbreviated as Ada_CART).

3.5.3. Stacking

Stacking regression combines multiple regression models via a meta-regressor, using the out-of-fold prediction concept [70]. The stacking method used in this work splits the dataset into k folds, in which the 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 are used and stacked as the input data to the second-level regressors to form a final set of predictions [71]. The schematic diagram of the stacking model is shown in Figure 8. In this study, one stacking-based ensemble model (abbreviated as Stack_LR) was proposed based on the two-level scheme. SVR, BPNN, and CART were used as regression models in the first level to obtain the prediction results, and LR was used as the meta-regressor in the second level to combine and generate the final prediction results.

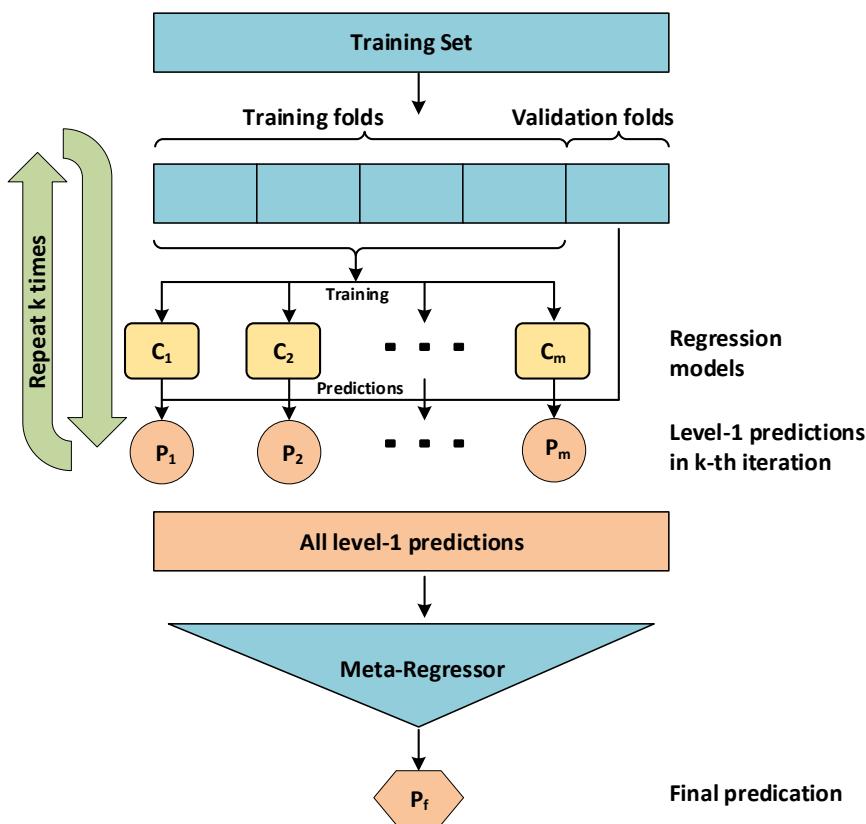


Figure 8. Schematic diagram of the stacking model [72].

4. Validation and Evaluation

4.1. Cross-Validation Method

Generally, a 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 the bias of random data splitting, the K-fold cross-validation is commonly used as it can yield optimal computational time and reliable variance [34,73]. In this study, a ten-fold cross-validation approach was applied to assess the model performance, as shown in Figure 9. The dataset was split randomly into 10 equal-sized subsets with a similar distribution. In each validation process, nine of the subsets were used for training and the rest for testing. The process was repeated 10 times [74]. The average accuracy after 10-times validation is reported as the model accuracy.

4.2. Performance Evaluation

To show and validate the accuracy of the proposed ML models, three statistical indices, namely mean absolute error (MAE), root-mean-squared error (RMSE), and the coefficient of determination R^2 , were used and expressed in Equations (14)–(16), respectively. The average deviation of the performance of an individual model or an ensemble model from a benchmark model in terms of the three statistical measures (MAE, RMSE, and R^2) was calculated using Equation (17).

- Mean absolute error (MAE).

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

- Root-mean-squared error (RMSE)

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

- 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, and \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 an individual method used in the ensemble method as the base learner.

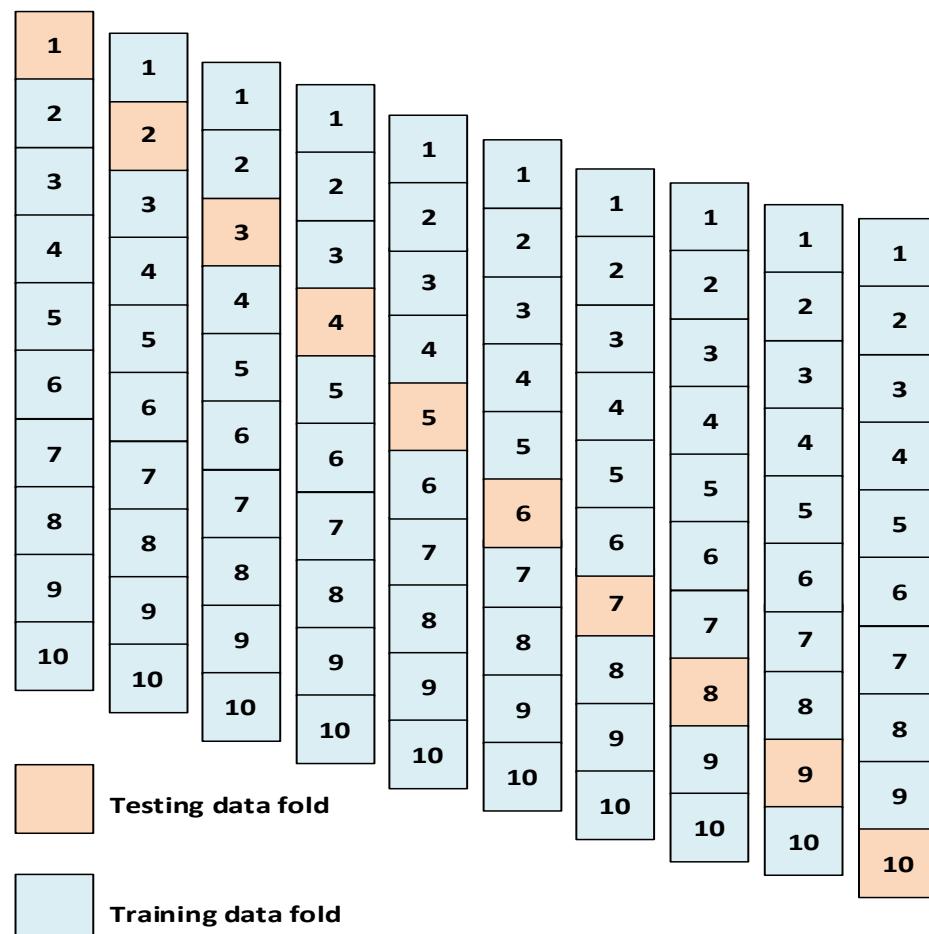


Figure 9. Ten-fold cross-validation approach.

It should be noted that the MAE and RMSE are commonly used indicators in ML to measure the error. Small values of the MAE and RMSE indicate less error, meaning better predictive models have been achieved. In this study [75], the MAE statistic is a measure of errors between the predicted values (the estimated value of crack width of ECC

after self-healing) with the target values (the observed value of the crack width of ECC after self-healing in empirical data). The RMSE statistic computes the square root of the average residual error between the predicted values and the target values. R^2 measures the strength of association between the predicted values and the target values, based on the proportion of the total variation of the outcomes. A greater value close to 1 represents a better prediction performance that commendably replicates the observed crack width of ECC after self-healing. The deviation statistic indicates the improvement of the prediction performance of an individual or an ensemble model from a benchmark model, which 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 the MAE, RMSE, and R^2 according to the ten-fold cross-validation. The abbreviations for labeling models were adopted in such a way that the letters Bag, Ada, and Stack stand for the ensemble methods of bagging, AdaBoost, and stacking, respectively. The letters LR, SVR, BPNN, and CART stand for the base estimators. However, Stack_LR refers to the model combining the base methods including SVR, BPNN, and CART in the first level and using LR as a meta-regressor in the second level.

5.1. Prediction Performance of the Proposed Models

Table 5 shows the ten-fold cross-validation results (MAE, RMSE, and R^2) for both individual and ensemble models and their deviation with respect to the results of the LR model.

Table 5. Average performances of machine learning models for the 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
	CART	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_CART	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_CART	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

Generally, most of the proposed models were able to learn and predict the empirical data with an acceptable degree of precision. Based on the results, the Stack_LR model showed the best prediction performance as it had the highest R^2 value and the lowest MAE and RMSE values. Among the individual models, SVR performed the best in terms of the MAE (4.296), whereas BPNN had the lowest RMSE value (6.515) and the highest R^2 of 0.899. Among the individual models boosted by either AdaBoost or bagging, Bag_CART gave the best performance in terms of the MAE (4.093), while Bag_BPNN performed better on the RMSE value (6.341). In terms of R^2 , the Bag_CART and Bag_BPNN models showed the same performance (0.901) and were better than other ensemble methods, except Stack_LR. The performances of all ML models described in Table 5 are depicted in Figure 10a–c in terms of the MAE, RMSE, and R^2 , respectively.

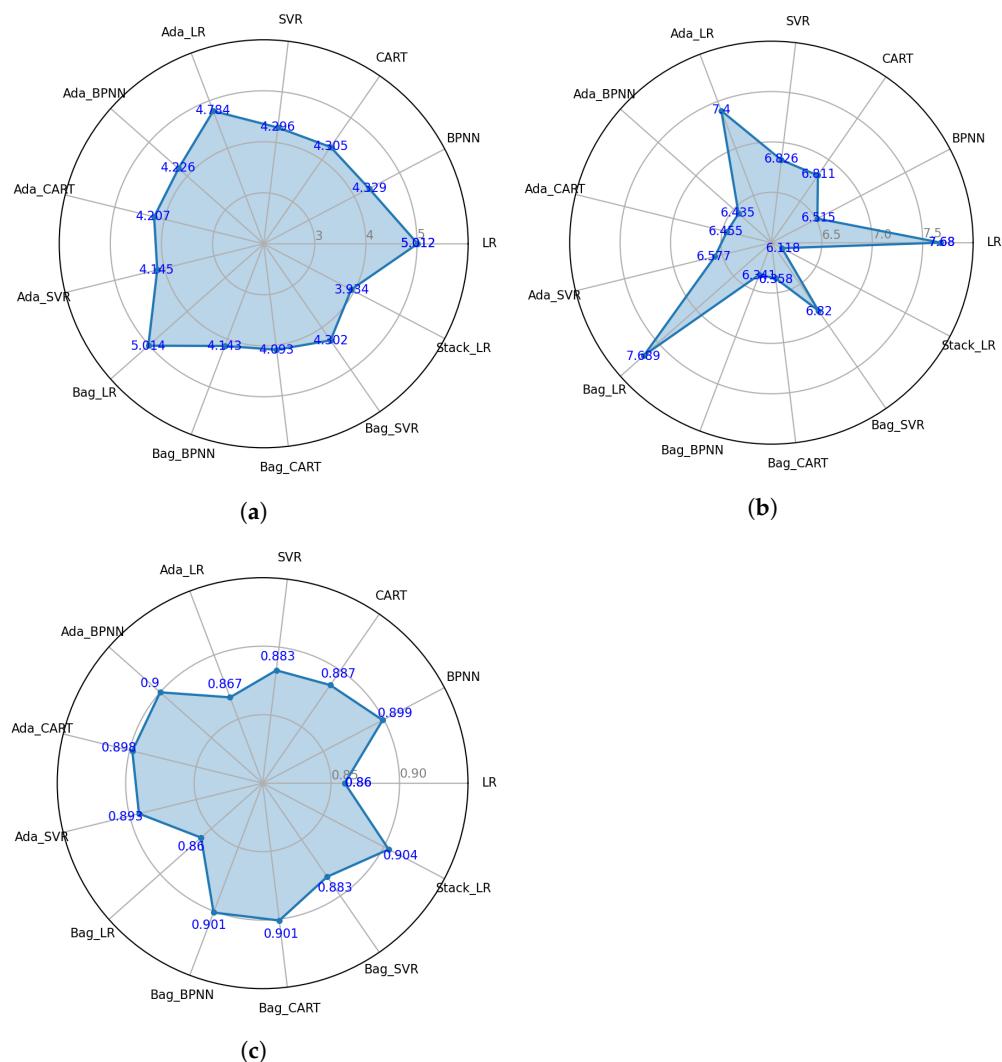


Figure 10. Average prediction performance of 10-fold cross-validation on all ML models for predicting the self-healing ability of ECC. (a) MAE; (b) RMSE; (c) R^2 .

Overall, all models could reduce the error values and increase the prediction accuracy compared with LR, except Bag_LR. Among the models boosted by AdaBoost, Ada_SVR performed the best with the lowest MAE value, whereas Ada_BPNN performed the best on the RMSE value, showing the highest R^2 value. In the case of bagging, both Bag_CART and Bag_BPNN performed better in terms of the MAE, RMSE and R^2 than those of the corresponding models boosted by AdaBoost. However, Bag_LR showed a poor performance compared to LR on the MAE and RMSE values. For a better comparison among the ensemble methods used, the performance results between the ensemble models and their corresponding individual (or benchmark) models are indicated in Table 6. The results indicate that most ensemble methods improved the performance of individual models. For example, the MAE and RMSE values of the BPNN after bagging reduced by 4.3% and 2.7%, respectively, and its R^2 was much higher than that of the individual BPNN model. Among all the ensemble methods studied, stacking showed the best improvement on all performance measures.

However, the results showed that the effectiveness of the ensemble methods on the individual models varied. For instance, the bagging method enhanced the performance of the BPNN and CART substantially, but not for both LR and SVR models. On the other hand, the AdaBoost method brought a considerable improvement for the LR and SVR models. To

improve the performance accuracy, researchers should employ different ensemble methods and compare their effectiveness on different ML models.

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

Benchmark	Model	MAE	RMSE	R ²	Benchmark	Model	MAE	RMSE	R ²
		Dev (%)	Dev (%)	Dev (%)			Dev (%)	Dev (%)	Dev (%)
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
CART	Ada_CART	−2.3	−5.2	1.2	CART	Bag_CART	−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

5.2. Prediction Performance Comparison

5.2.1. Comparison of the MAE

To reveal the accuracy of the proposed ML models in self-healing prediction, the comparison of the observed crack widths of ECC after self-healing with predicted crack widths are shown in Figures 11–14. Figure 11a shows the observed crack widths compared with the crack widths predicted by different individual ML models. Figure 11b–e shows the variations between the observed and the crack widths predicted by each individual ML model corresponding to their initial crack widths before self-healing. In other words, the prediction performance of the models in a particular range of crack widths can be revealed. It should be noted that the horizontal line located at the vertical coordinate of zero ($y = 0$) is considered as the target line [25,31]. Generally, the smaller the variation (i.e., closer to the target line), the better the self-healing prediction was, which means smaller or even no variation between the observed and the predicted crack widths after self-healing.

As shown in Figure 11, the SVR model generally exhibited better prediction results than other individual models, while the LR model was the worst, showing substantial deviation from the target line (denoting relatively large differences between the observed and the predicted crack widths). For the initial crack widths less than 20 μm and over 100 μm before self-healing, the variations between the observed and the ones predicted by the SVR model were smaller than other individual ML models. The corresponding MAE values were 1.358 and 2.724. However, for the crack widths between 20 μm and 60 μm , the CART model performed the best with the lowest MAE of 5.045, while the BPNN model had the lowest MAE of 9.565 for the crack widths between 60 μm and 100 μm . It seems that the choice of ML models may depend on the initial crack widths. However, in terms of overall accuracy among the individual models, SVR performed the best, followed by CART, the BPNN, and LR. This is consistent with the results shown in Table 5.

The performance of ensemble methods using AdaBoost and bagging is shown in Figures 12 and 13. In general, the ensemble models Ada_CART and bag_CART exhibited lower variations in the self-healing results compared to the other ensemble models. In particular, the MAE values of Ada_CART and bag_CART for crack widths between 20 self heal and 60 μm were 5.037 and 5.000, respectively. These values were smaller than that of CART (5.045), as shown in Figure 11e. However, the variations among the BPNN, Ada_BPNN, and bag_BPNN were not significant. Similar variations can be found when comparing SVR with Ada_SVR and bag_SVR.

After stacking, the error variations shown in Figure 14 were much reduced when compared to those shown in Figures 11–13. More specifically, the MAE of stack_LR for crack widths less than 20 μm , between 20 μm and 60 μm , between 60 μm and 100 μm , and over 100 μm were 1.361, 4.931, 9.789, and 3.177, respectively. These MAE values were the lowest among all the ML models studied. Based on the results, it can be concluded that the stack_LR model performed the best.

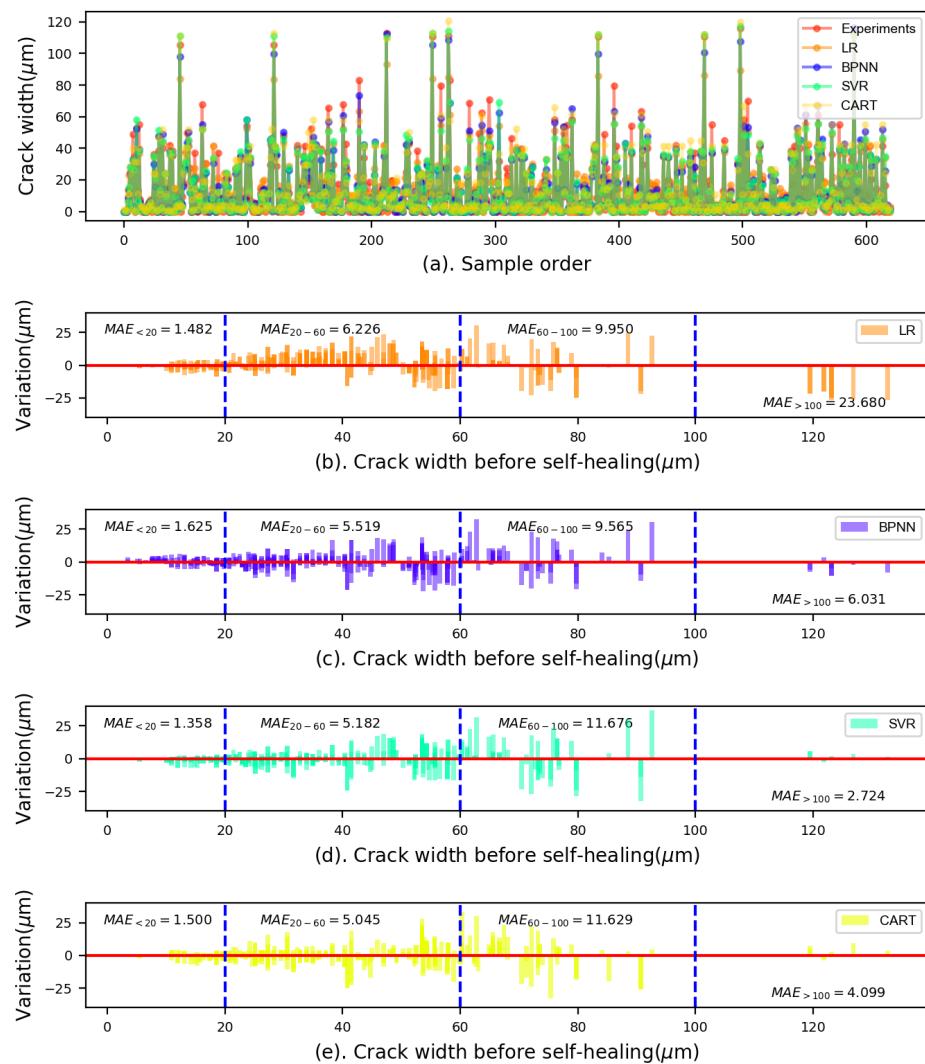


Figure 11. Comparison of the observed crack widths of ECC after self-healing with the crack widths predicted by the individual models.

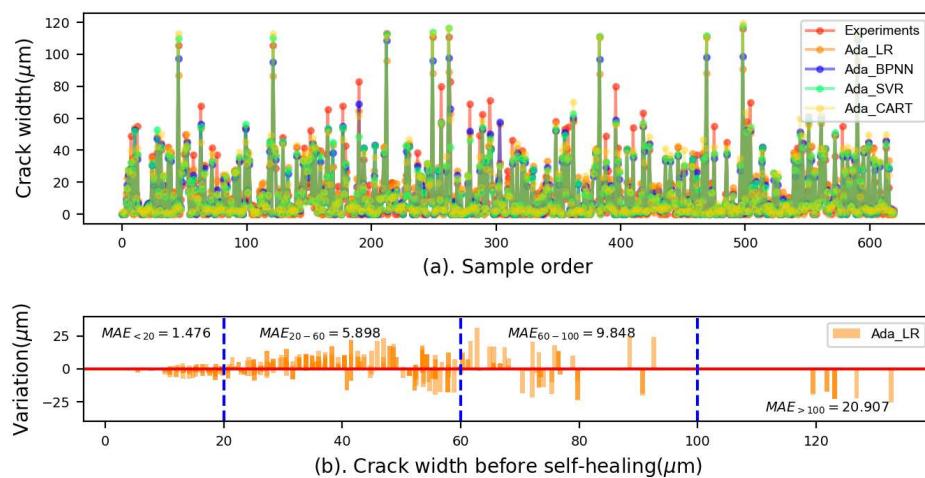


Figure 12. Cont.

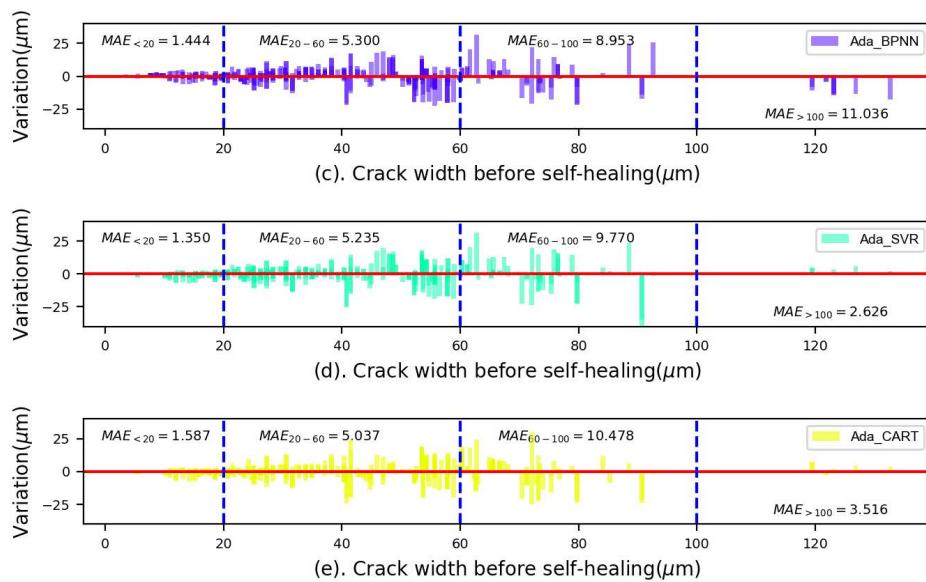


Figure 12. Comparison of the observed crack widths of ECC after self-healing with the crack widths predicted by the AdaBoost ensemble models.

It is known that a smaller crack width is favorable for autogenous healing in concrete [76,77] as small cracks consume less repair products to complete self-healing [78]. However, a larger crack width will not heal completely or just heal partially. As shown in Figures 11b, 12b, 13b and 14b, the variations between the observed and predicted results for the LR, bag_LR, and Ada_LR models increased with the increase of the crack width. For a crack width below 20 μm , the MAE values were less than 1.5, which was much lower than those for crack widths between 20 μm and 60 μm (i.e., 6.23) and between 60 μm and 100 μm (around 10). Similar trends were observed in other models, but with smaller variations. Specifically, for a crack width over 100 μm , the LR, bag_LR, and Ada_LR models showed much higher variations. Their MAE values were over 20 and higher than other ML models with the MAE less than 10.

5.2.2. Comparison of the RMSE

A box plot, as shown in Figure 15, was created to show the distribution of the RMSE results of each ML model based on the ten-fold cross-validation. The RMSE values were calculated based on the differences between the predicted and observed crack widths. The box plot is a statistical tool that is used to depict numerical data through their quartiles including the maximum, minimum, and median values of a dataset [79,80]. The medium value is shown as the red line within the box. The interquartile range (IQR) in each box covers 50% (the lower 25% to the upper 75% quartiles) of the RMSE data point, while the whiskers drawn up and down to the maximum and minimum values represent 1.5-times the IQR from the RMSE data. All other points out of the whiskers range are outliers and shown as red dots. A mean value of the RMSE equal to zero would indicate that the predictions perfectly fit the observed data. However, this is almost never achieved in practice [81]. In general, the lower the RMSE value, the better the prediction performance of a model is.

Assessment of the box plot revealed that the stack_LR model outperformed all other models because of its shortest IQR length and smallest RMSE values, as shown in Figure 5. In contrast, the LR and bag_LR models had the longest IQR length and largest RMSE values, thereby suggesting that the LR model and its ensemble methods have low accuracy. Among the individual models, the BPNN had the lowest RMSE, while SVR had the shortest IQR length, but with three outliers (out of ten data points). In general, the BPNN gave the most stable performance, showing reasonable low RMSE values with a short IQR length.

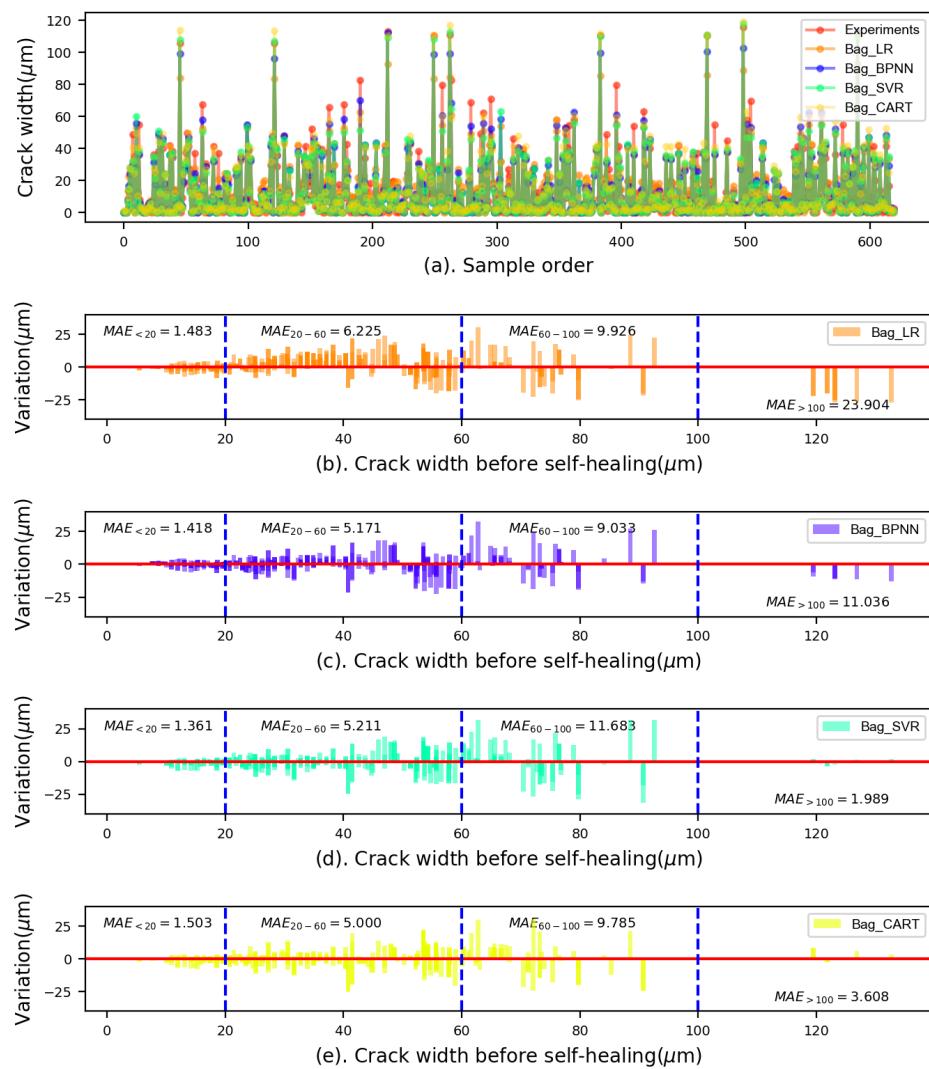


Figure 13. Comparison of the observed crack widths of ECC after self-healing with the crack widths predicted by the bagging ensemble models.

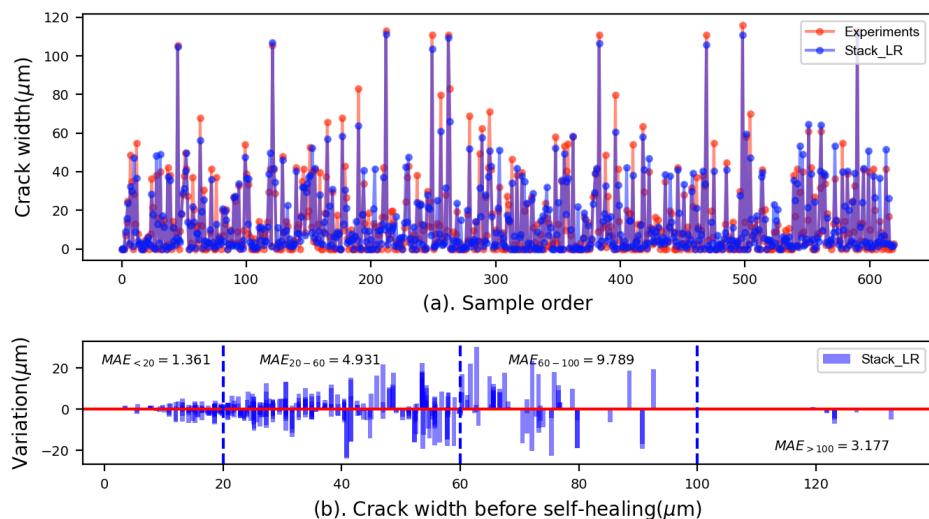


Figure 14. Comparison of the observed crack widths of ECC after self-healing with the crack widths predicted by the stacking ensemble models.

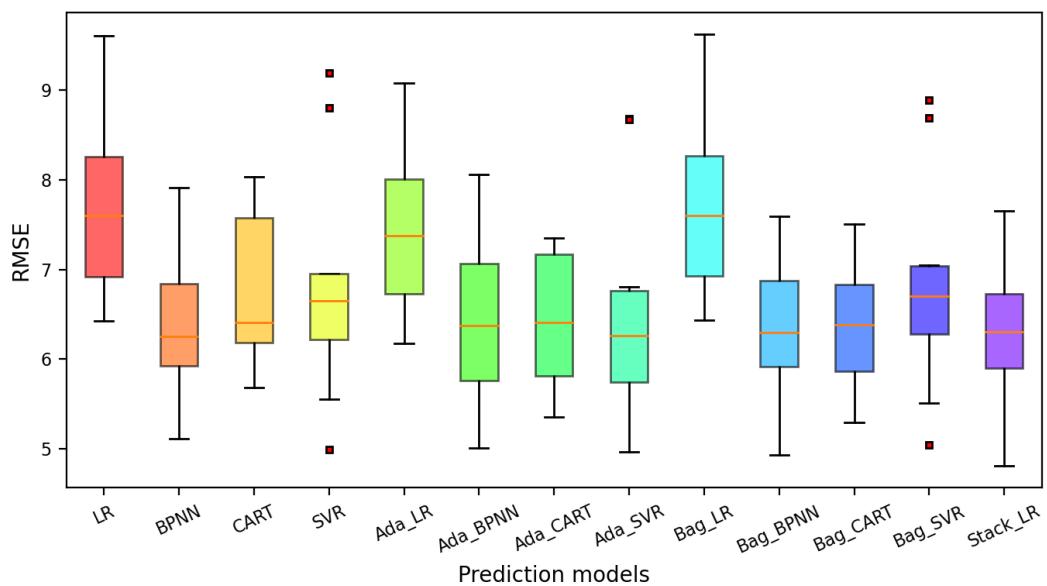


Figure 15. Ten-fold cross-validation of the RMSE by the proposed ML models in the prediction of the self-healing ability of ECC.

5.3. Limitations of Application

Although the present study provided evidence indicating that ML models can be used to predict the self-healing of ECC, there are some challenges that need to be considered. For example, the prolonged time required to optimize and tune the parameters and the high dependence on engineering datasets are the major concerns. The former can be addressed by using an optimization algorithm or developing a hybrid model to automatically adjust the parameters. In addition, high-performance computing (HPC) [82] can also be used to achieve parallel data processing so as to improve the computing performance and save time. In terms of the engineering dataset, it can be improved by experimental design and experimental process control [75].

Besides, it is worth noting that the ML models in this study mainly focused on the internal factors (features) such as materials and mix composition; other external environmental factors such as W/D cycles and healing time should also be considered. More research is required to explore the potential benefits and challenges of using ML models to predict the self-healing of cement-based composites.

6. Conclusions

In this study, four individual (LR, SVR, BPNN, and CART) and three ensemble (bagging, AdaBoost, and stacking) ML models were proposed to predict the self-healing of ECC. All the models were trained, validated, and tested based on the experimental results from nine ECC mixtures. Their prediction performances were analyzed and compared in terms of the MAE, RMSE, and R^2 . Based on the comparison of the results, the following conclusions can be drawn:

1. Among all individual ML models, the BPNN model performed the best in terms of the RMSE and R^2 , while the SVR model had the best performance in terms of the MAE;
2. All ensemble methods can generally improve the prediction accuracy of individual methods; however, the improvement varies. It was found that the bagging method mainly enhanced the performance of the BPNN and CART, whereas the AdaBoost method brought a considerable improvement for the LR and SVR models;
3. Among all the ML models studied, the Stack_LR model demonstrated great prediction on the self-healing of ECC and performed the best on the MAE, RMSE, and R^2 . The

- assessment of the box plot also revealed that the stackLR model outperformed all other models because of its shortest IQR length and smallest RMSE values;
4. For the initial crack widths less than 60 μm , the variations shown in the SVR model were smaller than those presented in other models. However, the CART model showed smaller variations for the crack widths between 60 μm and 100 μm compared to the SVR and BPNN models. For crack widths larger than 100 μm , the SVR model performed the best, showing the smallest variations;
 5. The computational results indicated that the individual and ensemble methods could be used to predict the self-healing ability of ECC. However, how to choose an appropriate base learner and ensemble method is critical. To improve the performance accuracy, researchers should employ different ensemble methods to compare their effectiveness with different ML models.

Although the proposed ML models can be used to predict the self-healing of ECC, there are some challenges such as time consumption and the quality of the dataset that need to be addressed. Future investigation and experimentation should be carried out to extend the training dataset to include various crack width distributions and diverse influencing factors such as W/D cycles, healing time, etc. In addition, more research should be undertaken to optimize the parameters in ML models and develop hybrid ML models to improve the prediction accuracy.

Author Contributions: Conceptualization, G.C. and W.T.; methodology, G.C., S.C., W.T. and S.W.; software, G.C. and S.C.; validation, G.C., W.T., S.C. and H.C.; formal analysis, G.C., S.C. and W.T.; investigation, G.C. and S.C.; resources, W.T., S.W. and H.C.; data curation, G.C. and S.C.; writing—original draft preparation, G.C., S.C. and W.T.; writing—review and editing, G.C., W.T., S.C., S.W. and H.C.; visualization, G.C., S.C. and W.T.; supervision, W.T. and S.W.; project administration, W.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in Github at <https://github.com/davidnsw/Self-healing-of-ECC> accessed on 1 March 2022, reference number [50].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gardner, D.; Lark, R.; Jefferson, T.; Davies, R. A survey on problems encountered in current concrete construction and the potential benefits of self-healing cementitious materials. *Case Stud. Constr. Mater.* **2018**, *8*, 238–247. [[CrossRef](#)]
2. Cailleux, E.; Pollet, V. 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, 28 June–1 July 2009; Volume 28.
3. Ramadan Suleiman, A.; Nehdi, M.L. Modeling Self-Healing of Concrete Using Hybrid Genetic Algorithm–Artificial Neural Network. *Materials* **2017**, *10*, 135. [[CrossRef](#)] [[PubMed](#)]
4. Tang, W.; Kardani, O.; Cui, H. Robust evaluation of self-healing efficiency in cementitious materials—A review. *Constr. Build. Mater.* **2015**, *81*, 233–247. [[CrossRef](#)]
5. Edvardsen, C. Water permeability and autogenous healing of cracks in concrete. *Mater. J.* **1999**, *96*, 448–454.
6. Tang, W.; Chen, G.; Wang, S. Self-healing capability of ECC incorporating with different mineral additives—A review. In Proceedings of the 3rd International RILEM Conference on Microstructre Related Durability of Cementitious Composites, Nanjing, China, 24–26 October 2016; Miao, C., Ed.; RILEM, RILEM Publications SARL: Nanjing, China, 2016; pp. 668–679.
7. Jacobsen, S.; Marchand, J.; Hornain, H. SEM observations of the microstructure of frost deteriorated and self-healed concretes. *Cem. Concr. Res.* **1995**, *25*, 1781–1790. [[CrossRef](#)]
8. Reinhardt, H.W.; Jooss, M. Permeability and self-healing of cracked concrete as a function of temperature and crack width. *Cem. Concr. Res.* **2003**, *33*, 981–985. [[CrossRef](#)]
9. Şahmaran, M.; Yaman, İ.Ö. Influence of transverse crack width on reinforcement corrosion initiation and propagation in mortar beams. *Can. J. Civ. Eng.* **2008**, *35*, 236–245. [[CrossRef](#)]

10. Clear, C. *The Effects of Autogenous Healing Upon the Leakage of Water Through Cracks in Concrete*; Technical Report; Wexham Spring, Buckinghamshire, UK, 1985.
11. Sahmaran, M.; Yildirim, G.; Erdem, T.K. Self-healing capability of cementitious composites incorporating different supplementary cementitious materials. *Cem. Concr. Compos.* **2013**, *35*, 89–101. [[CrossRef](#)]
12. Kamada, T.; Li, V.C. The effects of surface preparation on the fracture behavior of ECC/concrete repair system. *Cem. Concr. Compos.* **2000**, *22*, 423–431. [[CrossRef](#)]
13. Li, V.C.; Kanda, T. Innovations forum: Engineered cementitious composites for structural applications. *J. Mater. Civ. Eng.* **1998**, *10*, 66–69. [[CrossRef](#)]
14. Özbay, E.; Şahmaran, M.; Lachemi, M.; Yücel, H.E. Self-Healing of Microcracks in High-Volume Fly-Ash-Incorporated Engineered Cementitious Composites. *ACI Mater. J.* **2013**, *110*, 33.
15. Wu, M.; Johannesson, B.; Geiker, M. A review: Self-healing in cementitious materials and engineered cementitious composite as a self-healing material. *Constr. Build. Mater.* **2012**, *28*, 571–583. [[CrossRef](#)]
16. Huang, H.; Ye, G.; Damidot, D. Characterization and quantification of self-healing behaviors of microcracks due to further hydration in cement paste. *Cem. Concr. Res.* **2013**, *52*, 71–81. [[CrossRef](#)]
17. Suleiman, A.R.; Nelson, A.J.; Nehdi, M.L. Visualization and quantification of crack self-healing in cement-based materials incorporating different minerals. *Cem. Concr. Compos.* **2019**, *103*, 49–58. [[CrossRef](#)]
18. Zhou, J.; Qian, S.; Sierra Beltran, M.; Ye, G.; Schlangen, E.; van Breugel, K. Developing engineered cementitious composite with local materials. In Proceedings of the International Conference on Microstructure Related Durability of Cementitious Composites, Nanjing, China, 13–15 October 2008.
19. Li, V.C.; Yang, E.H. Self healing in concrete materials. In *Self Healing Materials*; Springer: Dordrecht, The Netherlands, 2007; pp. 161–193.
20. Zhang, Z.; Qian, S.; Ma, H. Investigating mechanical properties and self-healing behavior of micro-cracked ECC with different volume of fly ash. *Constr. Build. Mater.* **2014**, *52*, 17–23. [[CrossRef](#)]
21. Yildirim, G.; Sahmaran, M.; Ahmed, H.U. Influence of hydrated lime addition on the self-healing capability of high-volume fly ash incorporated cementitious composites. *J. Mater. Civ. Eng.* **2014**, *27*, 04014187. [[CrossRef](#)]
22. Yang, Y.; Lepech, M.; Li, V.C. Self-Healing of ECC under Cyclic Wetting and Drying. In Proceedings of the International Workshop of Durability of Reinforced Concrete under Combined Mechanical and Climatic Loads (CMCL), Qingdao, China, 27–28 October 2005.
23. Sahmaran, M.; Li, M.; Li, V.C. Transport properties of engineered cementitious composites under chloride exposure. *ACI Mater. J.* **2007**, *104*, 604.
24. Qian, S.; Zhou, J.; Schlangen, E. Influence of curing condition and precracking time on the self-healing behavior of engineered cementitious composites. *Cem. Concr. Compos.* **2010**, *32*, 686–693. [[CrossRef](#)]
25. Alshihri, M.M.; Azmy, A.M.; El-Bisy, M.S. Neural networks for predicting compressive strength of structural light weight concrete. *Constr. Build. Mater.* **2009**, *23*, 2214–2219. [[CrossRef](#)]
26. Xu, H.; Zhou, J.; G Asteris, P.; Jahed Armaghani, D.; Tahir, M.M. Supervised machine learning techniques to the prediction of tunnel boring machine penetration rate. *Appl. Sci.* **2019**, *9*, 3715. [[CrossRef](#)]
27. Reuter, U.; Sultan, A.; Reischl, D.S. A comparative study of machine learning approaches for modeling concrete failure surfaces. *Adv. Eng. Softw.* **2018**, *116*, 67–79. [[CrossRef](#)]
28. Miani, M.; Dunnhofer, M.; Rondinella, F.; Manthos, E.; Valentini, J.; Micheloni, C.; Baldo, N. Bituminous Mixtures Experimental Data Modeling Using a Hyperparameters-Optimized Machine Learning Approach. *Appl. Sci.* **2021**, *11*, 11710. [[CrossRef](#)]
29. Moayedi, H.; Bui, D.T.; Dounis, A.; Lyu, Z.; Foong, L.K. Predicting heating load in energy-efficient buildings through machine learning techniques. *Appl. Sci.* **2019**, *9*, 4338. [[CrossRef](#)]
30. Gilan, S.S.; Jovein, H.B.; Ramezanianpour, A.A. Hybrid support vector regression–particle swarm optimization for prediction of compressive strength and RCPT of concretes containing metakaolin. *Constr. Build. Mater.* **2012**, *34*, 321–329. [[CrossRef](#)]
31. Yan, F.; Lin, Z.; Wang, X.; Azarmi, F.; Sobolev, K. Evaluation and prediction of bond strength of GFRP-bar reinforced concrete using artificial neural network optimized with genetic algorithm. *Compos. Struct.* **2017**, *161*, 441–452. [[CrossRef](#)]
32. Yaseen, Z.M.; Deo, R.C.; Hilal, A.; Abd, A.M.; Bueno, L.C.; Salcedo-Sanz, S.; Nehdi, M.L. Predicting compressive strength of lightweight foamed concrete using extreme learning machine model. *Adv. Eng. Softw.* **2018**, *115*, 112–125. [[CrossRef](#)]
33. Yan, K.; Shi, C. Prediction of elastic modulus of normal and high strength concrete by support vector machine. *Constr. Build. Mater.* **2010**, *24*, 1479–1485. [[CrossRef](#)]
34. Chou, J.S.; Tsai, C.F.; Pham, A.D.; Lu, Y.H. Machine learning in concrete strength simulations: Multi-nation data analytics. *Constr. Build. Mater.* **2014**, *73*, 771–780. [[CrossRef](#)]
35. Sobhani, J.; Najimi, M.; Pourkhorshidi, A.R.; Parhizkar, T. Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and ANFIS models. *Constr. Build. Mater.* **2010**, *24*, 709–718. [[CrossRef](#)]
36. Omran, B.A.; Chen, Q.; Jin, R. Comparison of data mining techniques for predicting compressive strength of environmentally friendly concrete. *J. Comput. Civ. Eng.* **2016**, *30*, 04016029. [[CrossRef](#)]
37. Mauludin, L.M.; Oucif, C. Modeling of self-healing concrete: A review. *J. Appl. Comput. Mech.* **2019**, *5*, 526–539.
38. Chaitanya, M.; Manikandan, P.; Kumar, V.P.; Elavenil, S.; Vasugi, V. Prediction of self-healing characteristics of GGBS admixed concrete using Artificial Neural Network. *J. Phys. Conf. Ser.* **2020**, *1716*, 012019. [[CrossRef](#)]

39. Zhuang, X.; Zhou, S. The prediction of self-healing capacity of bacteria-based concrete using machine learning approaches. *Comput. Mater. Contin.* **2019**, *59*, 57–77. [CrossRef]
40. Huang, X.; Wasouf, M.; Sresakoolchai, J.; Kaewunruen, S. Prediction of healing performance of autogenous healing concrete using machine learning. *Materials* **2021**, *14*, 4068. [CrossRef] [PubMed]
41. Ahmad, M.; Kamiński, P.; Olczak, P.; Alam, M.; Iqbal, M.J.; Ahmad, F.; Sasui, S.; Khan, B.J. Development of Prediction Models for Shear Strength of Rockfill Material Using Machine Learning Techniques. *Appl. Sci.* **2021**, *11*, 6167. [CrossRef]
42. Nasiri, S.; Khosravani, M.R. Machine learning in predicting mechanical behavior of additively manufactured parts. *J. Mater. Res. Technol.* **2021**, *14*, 1137–1153. [CrossRef]
43. AS 3972-2010 General Purpose and Blended Cements. Available online: https://infostore.saiglobal.com/en-au/standards/as-3972-2010-122323_saig_as_as_268436/?gclid=Cj0KCQjw3IqSBhCoARIAsAMBkTb0Ex9GHRGK_51CWoBo1ioVfTRsjLMIFR7Gt7V7PJWLzXBtybrTbPFQaAni7EALw_wcB&gclsrc=aw.ds (accessed on 30 September 2021)
44. Chen, G. Repeatability of Self-Healing in ECC with Various Mineral Admixtures. Ph.D. Thesis, School of Architecture and Built Environment, University of Newcastle, Callaghan, Australia, 2021.
45. Nehdi, M.; Djebbar, Y.; Khan, A. Neural network model for preformed-foam cellular concrete. *Mater. J.* **2001**, *98*, 402–409.
46. Oztacs, A.; Pala, M.; Ozbay, E.; Kanca, E.; Caglar, N.; Bhatti, M.A. Predicting the compressive strength and slump of high strength concrete using neural network. *Constr. Build. Mater.* **2006**, *20*, 769–775. [CrossRef]
47. Deepa, C.; SathiyaKumari, K.; Sudha, V.P. Prediction of the compressive strength of high performance concrete mix using tree based modeling. *Int. J. Comput. Appl.* **2010**, *6*, 18–24. [CrossRef]
48. Duan, Z.H.; Kou, S.C.; Poon, C.S. Prediction of compressive strength of recycled aggregate concrete using artificial neural networks. *Constr. Build. Mater.* **2013**, *40*, 1200–1206. [CrossRef]
49. Sun, J.; Zhang, J.; Gu, Y.; Huang, Y.; Sun, Y.; Ma, G. Prediction of permeability and unconfined compressive strength of pervious concrete using evolved support vector regression. *Constr. Build. Mater.* **2019**, *207*, 440–449. [CrossRef]
50. Chen, G. Self-Healing of ECC. 2021. Available online: <https://github.com/davidnsw/Self-healing-of-ECC> (accessed on 1 March 2022).
51. Neter, J.; Kutner, M.H.; Nachtsheim, C.J.; Wasserman, W. *Applied Linear Statistical Models*; Irwin: Chicago, IL, USA, 1996; Volume 4,
52. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
53. Vapnik, V.N. An overview of statistical learning theory. *IEEE Trans. Neural Netw.* **1999**, *10*, 988–999. [CrossRef] [PubMed]
54. Juncai, X.; Qingwen, R.; Zhenzhong, S. Prediction of the strength of concrete radiation shielding based on LS-SVM. *Ann. Nucl. Energy* **2015**, *85*, 296–300. [CrossRef]
55. Rundo, F.; Trenta, F.; di Stallo, A.L.; Battiatto, S. Machine learning for quantitative finance applications: A survey. *Appl. Sci.* **2019**, *9*, 5574. [CrossRef]
56. Suykens, J.A.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300. [CrossRef]
57. Li, Y.; Shao, X.; Cai, W. A consensus least squares support vector regression (LS-SVR) for analysis of near-infrared spectra of plant samples. *Talanta* **2007**, *72*, 217–222. [CrossRef]
58. Yuvaraj, P.; Murthy, A.R.; Iyer, N.R.; Sekar, S.; Samui, P. Support vector regression based models to predict fracture characteristics of high strength and ultra high strength concrete beams. *Eng. Fract. Mech.* **2013**, *98*, 29–43. [CrossRef]
59. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
60. Mukherjee, A.; Biswas, S.N. Artificial neural networks in prediction of mechanical behavior of concrete at high temperature. *Nucl. Eng. Des.* **1997**, *178*, 1–11. [CrossRef]
61. Naderpour, H.; Rafiean, A.H.; Fakharian, P. Compressive strength prediction of environmentally friendly concrete using artificial neural networks. *J. Build. Eng.* **2018**, *16*, 213–219. [CrossRef]
62. Yi, D.; Ahn, J.; Ji, S. An effective optimization method for machine learning based on ADAM. *Appl. Sci.* **2020**, *10*, 1073. [CrossRef]
63. Breiman, L. *Classification and Regression Trees*; Routledge: London, UK, 2017.
64. Dan, S.; Colla, P. *CART: Tree-Structured Non-Parametric Data Analysis*; Salford Systems: San Diego, CA, USA, 1995.
65. Put, R.; Perrin, C.; Questier, F.; Coomans, D.; Massart, D.; Vander Heyden, Y. Classification and regression tree analysis for molecular descriptor selection and retention prediction in chromatographic quantitative structure–retention relationship studies. *J. Chromatogr. A* **2003**, *988*, 261–276. [CrossRef]
66. Frosyniotis, D.; Stafylidis, A.; Likas, A. A divide-and-conquer method for multi-net classifiers. *Pattern Anal. Appl.* **2003**, *6*, 32–40. [CrossRef]
67. Dietterich, T.G. Ensemble methods in machine learning. In Proceedings of the International Workshop on Multiple Classifier Systems, Cagliari, Italy, 21–23 June 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 1–15.
68. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]
69. Freund, Y.; Schapire, R.E. Experiments with a New Boosting Algorithm. In Proceedings of the International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; Volume 96, pp. 148–156.
70. Raschka, S. MLxtend: Providing machine learning and data science utilities and extensions to Python’s scientific computing stack. *J. Open Source Softw.* **2018**, *3*, 638. [CrossRef]
71. Sill, J.; Takács, G.; Mackey, L.; Lin, D. Feature-weighted linear stacking. *arXiv* **2009**, arXiv:0911.0460.
72. Stacking. StackingCVRegressor-mlxtend. Available online: https://rasbt.github.io/mlxtend/user_guide/regressor/StackingCVRegressor/ (accessed on 4 March 2022).

73. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 20–25 August 1995; Volume 14, pp. 1137–1145.
74. Dor, O.; Zhou, Y. Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training. *Proteins Struct. Funct. Bioinform.* **2007**, *66*, 838–845. [[CrossRef](#)]
75. Nguyen, H.; Vu, T.; Vo, T.P.; Thai, H.T. Efficient machine learning models for prediction of concrete strengths. *Constr. Build. Mater.* **2021**, *266*, 120950. [[CrossRef](#)]
76. Herbert, E.N.; Li, V.C. Self-Healing of Microcracks in Engineered Cementitious Composites (ECC) under a Natural Environment. *Materials* **2013**, *6*, 2831–2845. [[CrossRef](#)]
77. Liu, H.; Zhang, Q.; Gu, C.; Su, H.; Li, V. Influence of microcrack self-healing behavior on the permeability of Engineered Cementitious Composites. *Cem. Concr. Compos.* **2017**, *82*, 14–22. [[CrossRef](#)]
78. De Belie, N.; Gruyaert, E.; Al-Tabbaa, A.; Antonaci, P.; Baera, C.; Bajare, D.; Darquennes, A.; Davies, R.; Ferrara, L.; Jefferson, T.; et al. A review of self-healing concrete for damage management of structures. *Adv. Mater. Interfaces* **2018**, *5*, 1800074. [[CrossRef](#)]
79. Taffese, W.Z.; Sistonen, E.; Puttonen, J. CaPrM: Carbonation prediction model for reinforced concrete using machine learning methods. *Constr. Build. Mater.* **2015**, *100*, 70–82. [[CrossRef](#)]
80. Olalusi, O.B.; Spyridis, P. Machine learning-based models for the concrete breakout capacity prediction of single anchors in shear. *Adv. Eng. Softw.* **2020**, *147*, 102832. [[CrossRef](#)]
81. Contributors, W. Root-Mean-Square Deviation—Wikipedia, The Free Encyclopedia. 2021. Available online: https://en.wikipedia.org/wiki/Root-mean-square_deviation (accessed on 28 August 2021).
82. UON. Getting Started with HPC. 2021. Available online: <https://www.newcastle.edu.au/events/research-and-innovation/hpc> (accessed on 1 March 2022).