

## The Prediction of Self-Healing Capacity of Bacteria-Based Concrete Using Machine Learning Approaches

Xiaoying Zhuang<sup>1,2,\*</sup> and Shuai Zhou<sup>3,4</sup>

**Abstract:** Advances in machine learning (ML) methods are important in industrial engineering and attract great attention in recent years. However, a comprehensive comparative study of the most advanced ML algorithms is lacking. Six integrated ML approaches for the crack repairing capacity of the bacteria-based self-healing concrete are proposed and compared. Six ML algorithms, including the Support Vector Regression (SVR), Decision Tree Regression (DTR), Gradient Boosting Regression (GBR), Artificial Neural Network (ANN), Bayesian Ridge Regression (BRR) and Kernel Ridge Regression (KRR), are adopted for the relationship modeling to predict crack closure percentage (CCP). Particle Swarm Optimization (PSO) is used for the hyper-parameters tuning. The importance of parameters is analyzed. It is demonstrated that integrated ML approaches have great potential to predict the CCP, and PSO is efficient in the hyper-parameter tuning. This research provides useful information for the design of the bacteria-based self-healing concrete and can contribute to the design in the rest of industrial engineering.

**Keywords:** Bacteria, self-healing concrete, crack closure percentage, machine learning, prediction.

### 1 Introduction

Concrete is the most widely used construction material, which embodies energy of about 0.95 MJ/kg [Chilana, Bhatt, Najafi et al. (2016)]. Due to the high rate of consumption, high energy is required. One disadvantage of concrete is its sensitivity to cracks because of its limited tensile strength. Hence, an efficient method which can heal the cracks of cement and concrete will save the total energy and reduce environmental impact greatly. Over recent decades, the notion of designing concrete with a self-healing behavior to heal the cracks has attracted great attention [Zhu, Chen, Yan et al. (2014); Zhu, Zhou, Yan et al. (2015a); Zhu, Zhou, Yan et al. (2015b); Zhu, Zhou, Yan et al. (2016); Zhou, Zhu, Yan et al. (2016); Zhou, Zhu, Ju et al. (2017); Zhuang and Zhou (2018); Quayum, Zhuang and Rabczuk (2015); Zhang and Zhuang (2018)]. Many self-healing strategies are developed

---

<sup>1</sup> Division of Computational Mechanics, Ton Duc Thang University, Ho Chi Minh City, Vietnam.

<sup>2</sup> Faculty of Civil Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam.

<sup>3</sup> College of Materials Science and Engineering, Chongqing University, Chongqing, 400045, China.

<sup>4</sup> Institute of Continuum Mechanics, Leibniz University Hannover, Hannover, 30167, Germany.

\* Corresponding Author: Xiaoying Zhuang. Email: xiaoying.zhuang@tdtu.edu.vn.

in the concrete and microbiologically-induced calcium carbonate precipitation (MICP) has been considered as a promising way.

The bacteria-based self-healing concrete dates back to the mid-1990s [Gollapudi, Knutson, Bang et al. (1995)]. The idea is to introduce ureolytic bacteria in the enzymatic hydrolysis of urea and precipitation of calcium carbonate in the damaged regions. Then, non-ureolytic bacteria combined with organic calcium source is proposed in the bacteria-based concrete [Jonkers (2007); Jonkers, Thijssen, Muyzer et al. (2010)]. It does not need bacteria added externally, which means the real self-healing concrete is developed. All the components are mixed into the concrete matrix, and organic calcium compounds are converted to calcium carbonate. The mechanism of MICP applied in the healing of concrete can be classified into two categories. They are mainly based on urea hydrolysis by ureolytic bacteria and respiration by non-ureolytic bacteria. MICP through ureolysis has many shortages, like the secondary pollution introduced by the ammonia generated and the limited ureolysis efficiency. Using non-ureolytic bacteria can increase the healing ability and avoid the pollution. Hence, only the non-ureolytic bacteria-based self-healing concrete is considered here due to its promising future in self-healing concretes. Even though a large amount of experiments have been conducted, there is no suggestion of the material design in practical use to control the crack width, which is important in civil engineering. The cracking phenomena of materials [Hamdia, Silani, Zhuang et al. (2017); Nanthakumar, Lahmer, Zhuang et al. (2016); Vu-Bac, Lahmer, Zhuang et al. (2016); Zhuang, Huang, Rabczuk et al. (2014)] have been investigated widely including efficient remeshing techniques [Areias, Reinoso, Camanho et al. (2018); Areias and Rabczuk (2017); Areias, Rabczuk and Msekh (2016); Areias, Msekh and Rabczuk (2016); Areias, Rabczuk and Camanho (2014); Areias, Rabczuk and Dias-da-Costa (2013); Areias and Rabczuk (2013); Anitescu, Hossain and Rabczuk (2018)], phase field methods [Badnava, Msekh, Etemadi et al. (2018); Msekh, Cuong, Zi et al. (2017)], multiscale methods for fracture [Budrapu, Gracie, Bordas et al. (2014); Budrapu, Gracie, Yang et al. (2014); Talebi, Silani and Rabczuk (2015); Talebi, Silani, Bordas et al. (2014)], peridynamics [Ren, Zhuang and Rabczuk (2017); Ren, Zhuang, Cai et al. (2016)], DEM [Zhou, Zhu, Ju et al. (2017)], meshfree methods [Amiri, Milan, Shen et al. (2014); Amiri, Anitescu, Arroyo et al. (2014); Rabczuk, Gracie, Song et al. (2010); Rabczuk, Areias and Belytschko (2007); Zhuang, Cai and Augarde (2014); Zhuang, Zhu and Augarde (2014)], the phantom node method [Chau-Dinh, Zi, Lee et al. (2012)], the smooth extended finite element method [Chen, Rabczuk, Bordas et al. (2012)] as well as other partition of unity based methods like the extended isogeometric analysis [Ghorashi, Valizadeh, Mohammadi et al. (2015); Nguyen-Thanh, Valizadeh, Nguyen et al. (2015); Nguyen-Xuan, Liu, Bordas et al. (2013)]. However, the healing effects of such materials are not well simulated. The optimized parameters of the bacteria-based self-healing concrete with the given crack width have not been studied previously. After artificial intelligence system is put forth, it has been widely used to substitute for the traditional regression analysis method, and begins to be used in forecasting works [Das, Biswal, Sivakugan et al. (2011); Gordan, Jahed Armaghani, Hajihassani et al. (2016); Hoang and Pham (2016)]. Only a few researches use ML in the optimization of the self-healing concrete [Ramadan Suleiman and Nehdi (2017)]. However, a systematic comparison of the available ML

algorithms still lacks since the performance differences may be substantial in their application to the optimization of the bacteria-based self-healing concrete.

To address above problems, this paper proposes and compares six integrated ML approaches for the bacteria-based self-healing concrete optimization. ML algorithms are adopted for the relationship modeling and Particle Swarm Optimization (PSO) is utilized for the hyper-parameters tuning. Six ML algorithms, including the Support Vector Regression (SVR), Decision Tree Regression (DTR), Gradient Boosting Regression (GBR), Artificial Neural Network (ANN), Bayesian Ridge Regression (BRR) and Kernel Ridge Regression (KRR), are used. This research is a benchmark study in the application of ML approaches in biomaterials optimization. The objective of this work is to optimize the self-healing materials for the required healing properties (i.e. the required CCP). ML and PSO algorithms are detailed in Section 2. Section 3 describes the dataset for the ML models. In Section 4, the methodology of the proposed integrated approaches is illustrated. Section 5 provides the results and the discussion while Section 6 summarizes findings.

## **2 Machine learning and PSO algorithms**

A brief introduction to ML and PSO is presented in this section. To predict the self-healing capacity and provide useful information for the design of the bacteria-based self-healing concrete, the six ML methods are selected to predict CCP since they have been widely used in industrial engineering and have a good predictive performance in nonlinear prediction. Meanwhile, PSO is adopted for the hyper-parameters tuning. Firstly, the PSO is used to obtain the optimum hyper-parameters in the six ML models. Then, a part of the dataset (i.e., the training set) is adopted to train the six ML models. Finally, the rest of the dataset (i.e., the testing set) is applied to test the results of the six ML models.

### **2.1 ML algorithms**

#### **2.1.1 SVR**

The basic idea of SVR is to map the data  $x$  into a high dimensional feature space via a nonlinear mapping and to perform a linear regression in this feature space by

$$f(x) = \sum_{i=1}^D w_i \phi_i(x) + b \quad (1)$$

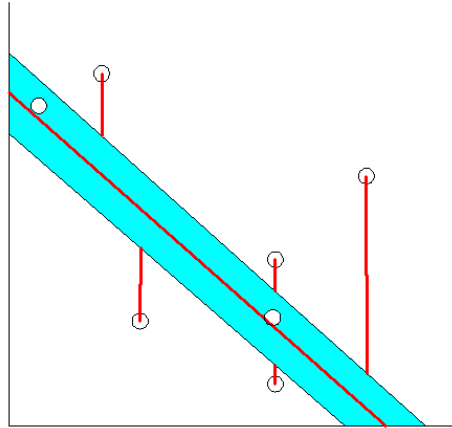
where  $\{\phi_i(x)\}_{i=1}^D$  are called features,  $b$  and  $\{w_i\}_{i=1}^D$  are related coefficients. In this way, a nonlinear regression is transferred to a linear regression. The coefficients  $\{w_i\}_{i=1}^D$  can be determined from the data by minimizing the function

$$R[w] = \frac{1}{N} \sum_{i=1}^N |f(x_i) - y_i|_{\varepsilon} + \lambda \|w\|^2 \quad (2)$$

where  $\lambda$  is a regularization constant and the cost function is defined by

$$|f(x_i) - y_i|_{\varepsilon} = \begin{cases} |f(x_i) - y_i| - \varepsilon & \text{for } |f(x_i) - y_i| \geq \varepsilon \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Eq. (3) can be illustrated in Fig. 1. If the point lies in the blue area (i.e. the distance between the point and the line is less than  $\varepsilon$ ), the cost is zero. If the point lies outside the blue area, the cost is  $|f(x_i) - y_i| - \varepsilon$ . More details of SVR can be found in previous research [Cristianini and Shaw-Taylor (2000); Vapnik (1999)].



**Figure 1:** The illustration of the cost function in SVR

### 2.1.2 DTR

DTR is a non-parametric procedure for predicting continuous dependent variable where the data is partitioned into nodes on the basis of conditional binary responses [Breiman, Friedman, Olshen et al. (1984)]. Models use a binary tree to recursively partition the predictor space into subsets in which the distribution of  $y$  is successively more homogenous [Chipman, George and McCulloch (1998)]. A decision tree  $P$  with  $t$  terminal nodes is used for communicating the decision. A parameter  $\phi = (\phi_1, \phi_2, \phi_3, \dots, \phi_t)$  associates the parameter value  $\phi_i$  ( $i = 1, 2, 3, \dots, t$ ) with the  $i$ th terminal node. The partitioning procedure searches through all values of predictor variables to find the variable  $x$  that provides best partition into child nodes, which minimizes the weighted variance. A detailed discussion of DTR model can be found in previous research [Breiman, Friedman, Olshen et al. (1984); Chipman, George and McCulloch (1998)].

### 2.1.3 GBR

Gradient boosting regression is a machine learning technique for regression problems, typically on the basis of decision trees. It builds the model in a stage-wise fashion and allows optimization of an arbitrary differentiable loss function. The goal in GBR is to find a loss function  $F^*(x)$  and minimize the expected value of it over the joint distribution of all  $(y, x)$  values. Boosting evaluates  $F^*(x)$  by an additive expansion. The gradient boosting algorithm improves  $F^*(x)$  by adding an estimator  $h$  to provide a better model. A generalization of this idea to loss functions other than squared error is that residuals for a given model are the negative gradients of the squared error loss function. Hence, gradient

boosting is a gradient descent algorithm by adding a different loss. More details can be found in previous research [Friedman (2002)].

#### *2.1.4 BRR*

BRR is composed of Bayesian Regression and the Ridge Regression (linear least squares with  $l_2$ -norm regularization). The  $l_2$  regularization used in BRR is equivalent to finding a maximum a posteriori estimation under a Gaussian prior over the parameters  $w$  with precision  $\lambda^{-1}$ . Instead of setting lambda manually, it is possible to treat it as a random variable to be estimated from the data. The output  $y$  is assumed to be Gaussian distributed to obtain a probabilistic model. The prior for the parameter  $w$  is given by a spherical Gaussian. The priors over  $\lambda$  are chosen to be gamma distributions. More details can be found in previous research [MacKay (1992)].

#### *2.1.5 ANN*

ANN is a mathematical technique using an analogy to biological neurons to generate a general solution to a problem [Rumelhart, Hinton and Williams (1986)]. All neural functions are stored in the neurons and the connections between them. After learning historical data, ANN can be used effectively to predict new data. The training of ANNs is considered as the establishment of new connections between neurons. ANN architecture may have one or more hidden layers between the input and output layers. Each layer constitutes neurons, which are connected with other neurons by the weights passing signals to others. When the amount of signals received by one neuron overtakes its threshold, the activation function is awoken and the outcome is treated as the input of next neuron. It can approximate an arbitrary nonlinear function with satisfactory accuracy [Zhang, Wu, Zhong et al. (2008)]. They learn from examples by building an input–output mapping without explicit derivation of the model equation. They have been widely used in pattern classification, function approximation, optimization, prediction and automatic control and in many different domains, such as load forecasting and strength forecast [Hadavandi, Shavandi and Ghanbari (2010); Khosravi, Nahavandi and Creighton (2013); Khotanzad, Elragal and Lu (2000); Bashir and El-Hawary (2009); Qi, Fourie, Chen et al. (2018)].

#### *2.1.6 KRR*

Kernel ridge regression (KRR) combines Ridge Regression (linear least squares with  $l_2$ -norm regularization) with the kernel trick. For non-linear kernels, this corresponds to a non-linear function in the original space. The form of the model learned by KRR is identical to SVR. However, different loss functions are used. KRR uses squared error loss while SVR uses  $\epsilon$ -insensitive loss. Both adopt  $l_2$  regularization. In contrast to SVR, fitting KRR can be done in closed-form and is typically faster for medium-sized datasets. The learned functions are very similar. For larger training sets SVR scales better. SVR is faster than KRR for all sizes of the training set. More details about KRR can be found in previous research [Murphy (2012)].

## 2.2 PSO algorithm

There are many hyper-parameters in ML models, which need to be chosen properly. Traditional tuning methods, such as trial and error, are time-consuming. PSO can be utilized for the hyper-parameters tuning of the ML algorithm [Kennedy and Eberhart (1995)]. Compared with other metaheuristic algorithms [Montazeri-Gh, Poursamad and Ghalichi (2006); Ma, Xu, Wang et al. (2015); Long and Nhan (2012); Castaings, Lhomme, Trigui et al. (2016)], PSO requires fewer parameters to be tuned and less computational efforts for multi-objective optimization [Yang (2014); Geng, Mills and Sun (2014)]. Inspired by the social behavior of bird flocking, PSO is an evolutionary optimization algorithm in which a population of individuals changes with time. The major concepts of PSO come from the observations of the feeding behaviors of bird swarms, where flocks are formed through grouping of simple individuals and individuals interact with each other. Scientific simulation of the unpredictable collective behaviors generated from local information is performed. It is high-efficient and does not use gradient. The objective function value is then calculated for each particle. The PSO algorithm is described as below:

$$v_{id} = wv_{id} + c_1 \text{rand}() (p_{id} - x_{id}) + c_2 \text{Rand}() (p_{gd} - x_{id}) \quad (4)$$

$$x_{id} = x_{id} + v_{id} \quad (5)$$

where  $w$  is the constant inertia weight,  $c_1$  and  $c_2$  are positive constants,  $\text{rand}()$  and  $\text{Rand}()$  are two random functions in the range  $[0,1]$ ;  $x_i=(x_{i1}, x_{i2}, \dots, x_{iD})$  represents the  $i$ th particle;  $p_i=(p_{i1}, p_{i2}, \dots, p_{iD})$  represents the best previous position of the  $i$ th particle; the symbol  $g$  represents the index of the best particle among all the particles in the population;  $v_i=(v_{i1}, v_{i2}, \dots, v_{iD})$  represents the velocity for particle  $i$ . The swarm topology defines how particles are connected to one another to exchange information with the global best. The particles in the swarm make up a cloud that covers the entire search space in the initial iteration and gradually contracts as the iterations advance. It has been successfully applied to many problems such as artificial neural network training, function optimization, fuzzy control, and pattern classification, etc. Because of its ease of implementation and fast convergence to acceptable solutions, PSO has received broad attention in recent years [Poli (2008)]. More details can be found in previous research [Kennedy and Eberhart (1995)].

## 3 Crack closure of the bacteria-based self-healing concrete

The performance of six integrated ML approaches is verified and compared according to a dataset collected from the literature. Such dataset contains a considerable number of experimental cases where ML approaches can learn the relationship between CCP and its influencing variables. The whole dataset consists of 1223 cases collected from previous research for the training and the testing of prediction models [Stuckrath, Serpell, Valenzuela et al. (2014); Luo, Qian and Li (2015); Zhang, Zhou, Liu et al. (2017); Zhang, Liu, Feng et al. (2017); Khaliq and Ehsan (2016); Wiktor and Jonkers (2011)]. A wide range of variable values is covered. The successful application of proposed ML approaches for CCP is based on similar healing mechanisms and only the non-ureolytic bacteria-based self-healing concrete is considered.

For the purpose of crack prediction, three attributes are used: the number of bacteria, the

healing time and the initial crack width. These attributes are considered to be the influencing variables that govern the healing of cracks. The carrier and nutrient medium are treated as the affiliated parameter of the number of bacteria here since they are used to keep the number of bacteria. Each of these influencing variables is introduced as follows:

(1) The number of bacteria is defined as the number of bacteria in 1 g concrete. The data is in the range [0, 214993302] with a mean of 64085635.

(2) Healing time is the time between crack initiates in experiments (i.e. the start of the healing introduction) and the time of the measure of crack width after healing. More details about the healing time can be found in previous research [Zhang, Zhou, Liu et al. (2017); Zhang, Liu, Feng et al. (2017)]. The data is in the range [0, 100] with a mean of 38.

(3) The initial crack width is the crack width when the crack initiates. The data is in the range [0.06, 1] with a mean of 0.37.

The CCP is defined as

$$CCP = 1 - \frac{W_{healed}}{W_{initial}} \quad (6)$$

where  $W_{initial}$  and  $W_{healed}$  are the width of cracks before and after healing, respectively.

## **4 Methodology**

### **4.1 Dataset**

The dataset used for the implementation of the prediction is collected from 1223 specimens with different influencing variables (in Supplementary). The output of the PSO is chosen to be the CCP. The whole dataset needs to be split into the training set and the testing set. The training set is used for the model training and the hyper-parameters tuning, while the testing set is adopted for the performance evaluation of the models. The training set is chosen randomly in the beginning. The rest of the database is the testing set. The same training set is used to train the six ML models in this paper. The testing set in the six ML models is also the same. In this paper, 70% of the whole dataset is included in the training set, and the remaining 30% is included in the testing set.

### **4.2 K-fold cross validation**

k-fold cross validation is widely used during the process of hyper-parameters tuning [Stone (1974)]. By using this method, the original training set is divided into k folds. Then, k-1 folds are used to train ML models, while the remaining one fold is adopted to validate models. The training and validating process are repeated for k times with different fold as the validating fold. The performance of ML models is obtained by averaging performances from k iterations. Compared with other methods [Badawy, Msekh, Hamdia et al. (2017)], the advantage of k-fold cross validation is that all observations are used for both training and validation, and each observation is used for validation exactly once. In this study, k is set as ten [van der Gaag, Hoffman, Remijnsen et al. (2006)]. The mean squared error (MSE) and the correlation coefficient R are utilized for the hyper-parameters tuning and model evaluation. The MSE measures the



squared distances between the predicted and the experimental values. R evaluates the degree two variables' change is associated. The MSE and the R can be calculated using Eq. (7) and Eq. (8):

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i^* - y_i)^2 \quad (7)$$

$$R = \frac{\sum_{i=1}^n (y_i^* - \bar{y}^*)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (y_i^* - \bar{y}^*)^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (8)$$

where  $N$  is the number of samples,  $y_i^*$  and  $y_i$  are the predicted and experimental CCP values of the  $i$ th sample, respectively.

### 4.3 Hyper-parameters tuning

Before the construction of ML models, several important hyper-parameters need to be determined. Hyper-parameters tuning is necessary since the predictive performance of ML models varies widely with different hyper-parameters. A good predictive performance can be achieved with suitable hyper-parameters. Here, hyper-parameters to be tuned in the ML algorithm are summarized in Tab. 1. The tuning range for each hyper-parameter is determined by experience, trial tuning, and previous literature [Qi and Tang (2018); Qi, Fourie, Chen et al. (2018)]. The fitness function of PSO is chosen as the average MSE from 10-fold CV, which is minimized by PSO. The swarm size and the maximum iteration are chosen to be 300 [Armaghani, Hajihassani, Bejarbaneh et al. (2014)]. The  $w$ ,  $c_1$  and  $c_2$  are set as  $\frac{1+rand}{2}$ , 1.8 and 1.8 [Eberhart and Shi (2001)]. Details

on the hyper-parameters of six ML algorithms tuned with PSO are shown in Tab. 1.

**Table 1:** Hyper-parameters explanation and their range

| Algorithms | Parameters                  | Definition  | Scope     |
|------------|-----------------------------|---|-----------|
| SVR        | $C_{\text{penalty}}$        | Penalty parameter of the error term                                 | 0.1-10    |
|            | epsilon                     | It specifies the epsilon-tube within which no penalty is associated | 0.001-0.5 |
| DTR        | $\text{Depth}_{\text{max}}$ | The maximum depth of the tree                                       | 1-20      |
|            | $\text{Split}_{\text{min}}$ | The minimum number of samples required to split an internal node    | 2-30      |
|            | $\text{Leaf}_{\text{min}}$  | The minimum number of samples required at a leaf node               | 1-30      |
| GBR        | $\text{Depth}_{\text{max}}$ | The maximum depth of the individual regression estimators           | 3-50      |
|            | $n_{\text{estimators}}$     | The number of boosting stages                                       | 1-200     |
|            | learningrate                | It shrinks the contribution of each tree by                         | 0.0001-   |



|     |                      |   |                                     |
|-----|----------------------|---|-------------------------------------|
|     |                      | learningrate  | 0.99                                |
|     | Split <sub>min</sub> | The minimum number of samples to split a node   | 2-30                                |
|     | Leaf <sub>min</sub>  | The minimum number of samples required at a leaf node   | 1-30                                |
| BRR | alpha1               | shape parameter for the Gamma distribution prior over the alpha parameter                           | 10 <sup>-10</sup> -10 <sup>-4</sup> |
|     | alpha2               | inverse scale parameter (rate parameter) for the Gamma distribution prior over the alpha parameter  | 10 <sup>-10</sup> -10 <sup>-4</sup> |
|     | lambda1              | shape parameter for the Gamma distribution prior over the lambda parameter                          | 10 <sup>-10</sup> -10 <sup>-4</sup> |
|     | lambda2              | inverse scale parameter (rate parameter) for the Gamma distribution prior over the lambda parameter | 10 <sup>-10</sup> -10 <sup>-4</sup> |
| NN  | n <sub>layer</sub>   | The number of hidden layers   | 1-3                                 |
|     | n <sub>size</sub>    | Number of neurons in each hidden layer  | 1-20                                |
| KRR | gamma                | Interpretation of the default value is left to the kernel   | 0.01-100                            |
|     | degree               | Degree of the polynomial kernel   | 2-10                                |

## 5 Results and discussion

### 5.1 Results of hyper-parameters tuning

As mentioned above, PSO is utilized in this paper for the hyper-parameters tuning of six ML algorithms. Once MSE is stable, the minimum MSE values are found and their corresponding hyper-parameters are considered as the optimum hyper-parameters. The whole dataset is split into the training set and the testing set. Prediction models with the optimum hyper-parameters are trained using the training set. Their predictive performance is evaluated on the testing set. The optimum hyper-parameters of six ML algorithms are described as follows:

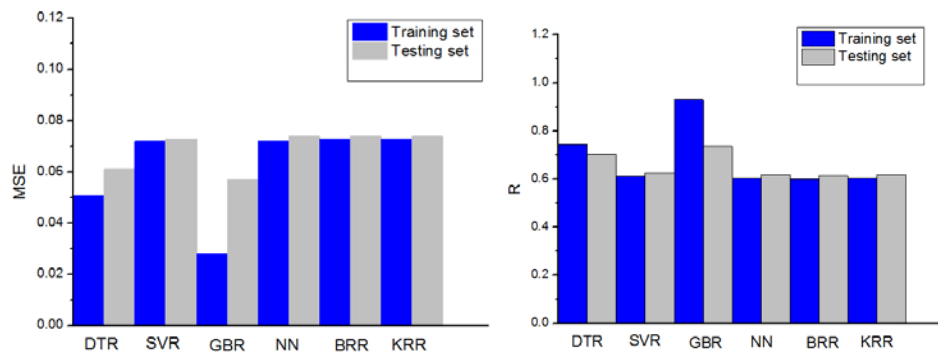
- DTR: Depth<sub>max</sub>=5, Split<sub>min</sub>=25, Leaf<sub>min</sub>=9.
- GBR: n<sub>estimators</sub>=200, Depth<sub>max</sub>=10, Split<sub>min</sub>=2, learningrate=0.0057, Leaf<sub>min</sub>=1.
- SVR: C<sub>penalty</sub>=0.144, epsilon=0.201.
- ANN: n<sub>layer</sub>=3, n<sub>size</sub>=(23, 5, 7).
- BRR: alpha1=10<sup>-10</sup>, alpha2=10<sup>-4</sup>, lambda1=10<sup>-4</sup>, lambda2=10<sup>-4</sup>.
- KRR: gamma=0.056, degree=3.

### 5.2 Comparison of integrated ML approaches

Six integrated ML approaches have been used for the prediction of CCP and their

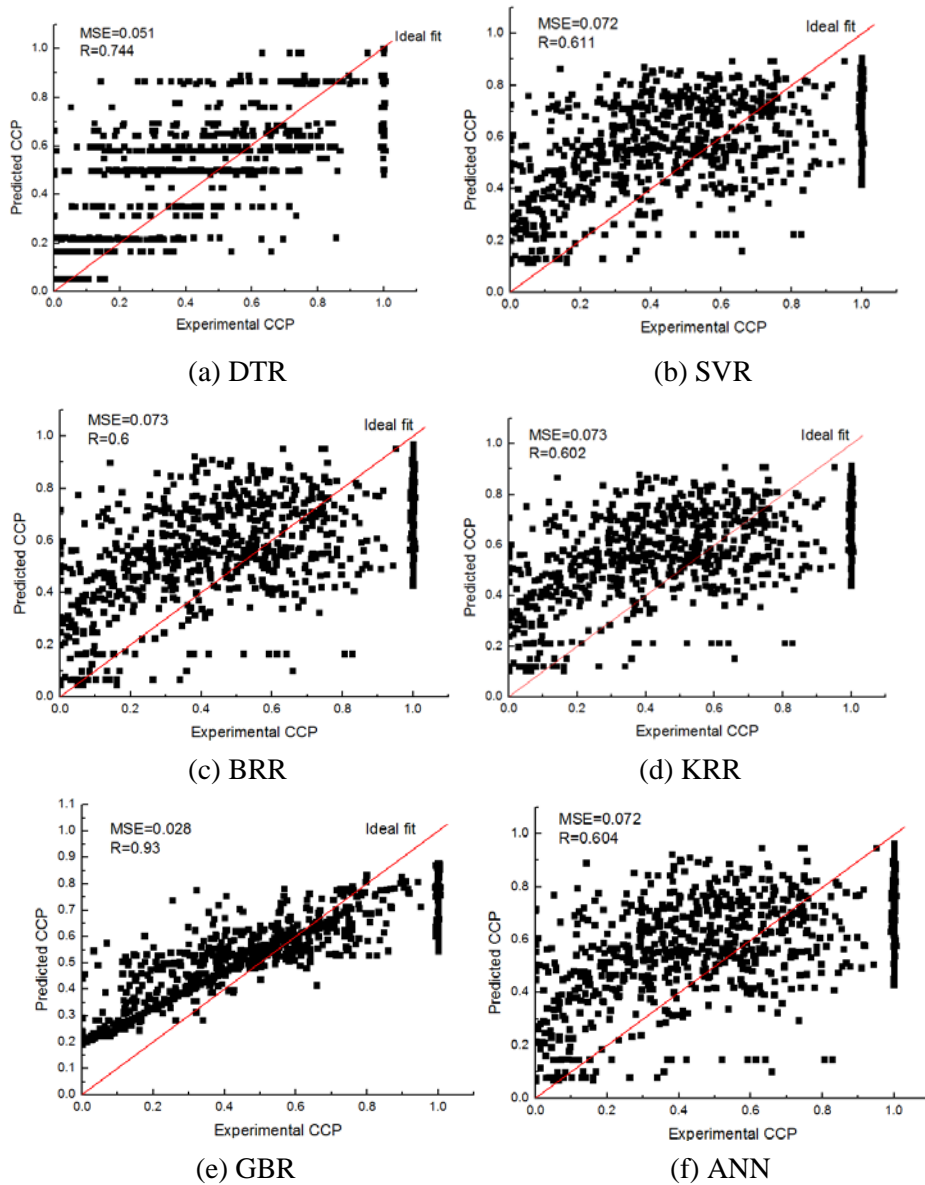
predictive performance on the testing set is compared and discussed in this section. MSE value and R value are utilized for the performance evaluation.

Fig. 2 describes MSE value and R value of six ML models on the testing set with the hyper-parameters obtained by PSO in Section 5.1. As can be seen in Fig. 2(a), the best prediction model regarding the MSE is GBR. The MSE values of the training set and the testing set are 0.028 and 0.057 respectively, when GBR is adopted. DTR also shows a good performance, with 0.051 and 0.061 as the MSE of the training set and the testing set, respectively. The other 4 models have a similar MSE about 0.7. In Fig. 2(b), GBR also shows the greatest R in all 6 models. The R values of the training set and the testing set are 0.93 and 0.74, respectively. With 0.74 and 0.7 as the R values of the training set and the testing set, DTR also proves a good predictive ability. The R values of other 4 models are similar and around 0.6. The R value of the testing set is a little greater than that of the training set in SVR, NN, BRR and KRR. Compared with DTR and GBR, the performance of BRR is relatively poor according to MSE and R values. Fig. 2 proves that DTR and GBR can achieve better predictive performance in predicting CCP of the bacteria-based self-healing concrete. The R value between the experimental and predicted CCP values is not high enough [Smith (1986)]. The main reason is due to the great variability of CCP data in experiments. With the lowest MSE and the highest R value, the optimum GBR model obtains better results than the other ML models. The optimum GBR and DTR models are recommended in predicting CCP of the bacteria-based self-healing concrete.



**Figure 2:** Predictive performance of six models (a) MSE; (b) R

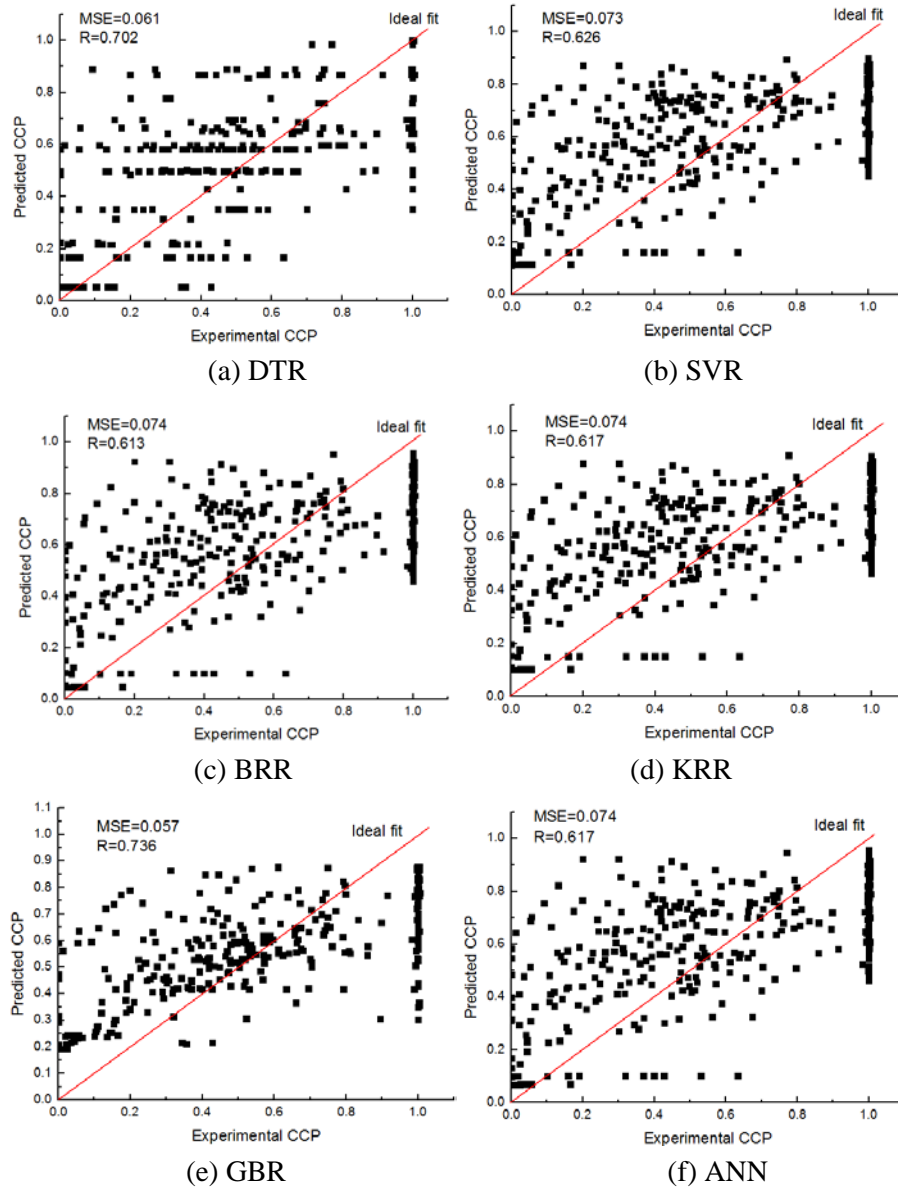
Detailed results about the performance of the six ML models on the training set are shown in Fig. 3. In Fig. 3, the data of the training set are close to the ideal fit line when GBR is adopted, which means GBR is more likely to predict the correct CCP. In contrast, the data in other 5 models are very scattered on the training set. It is caused by avoiding the overfitting when K-fold cross validation is adopted.



**Figure 3:** Performance of the 6 ML models on the training set: (a) DTR; (b) SVR; (c) BRR; (d) KRR; (e) GBR; (f) ANN

The comparison of experimental and predicted CCP values of the 6 ML models with the optimum hyper-parameters on the testing set is displayed in Fig. 4. It can be seen that the optimum ML models, which are adjusted according to the training set, can be generalized to the testing set. GBR is better than the other 5 models since the points are in the vicinity of the ideal fit line. The predicted CCP values of most samples match the experimental CCP values by GBR. This performance can improve when more experimental data are obtained. The data of other 5 models are away from the ideal fit. The GBR model with

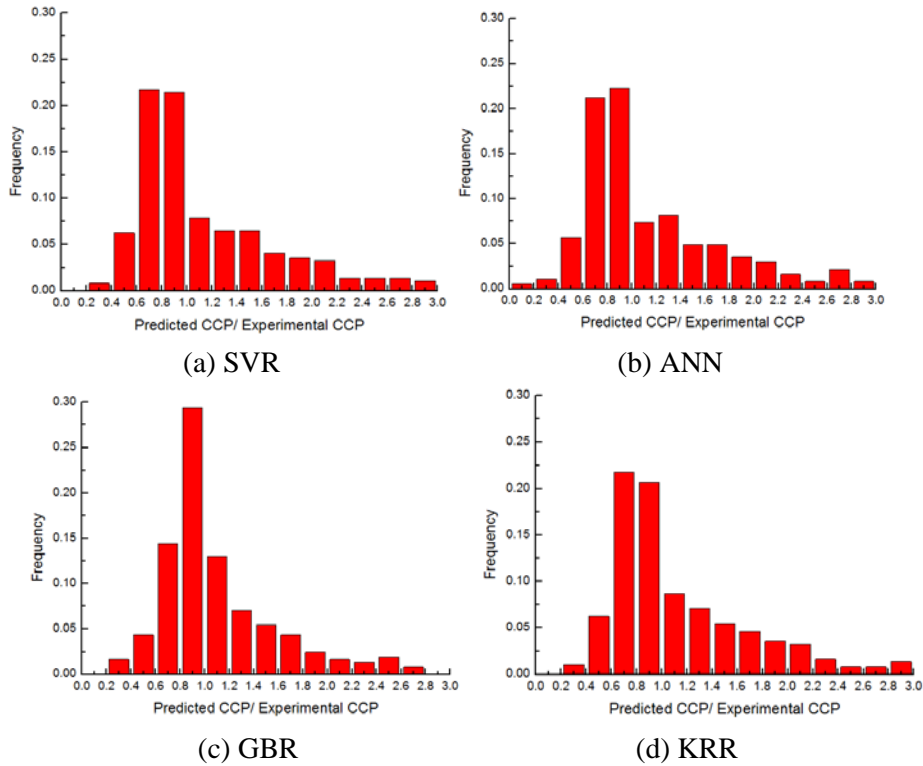
the optimum hyper-parameters can simulate the non-linear relationship between CCP values and influencing variables from Fig. 4.

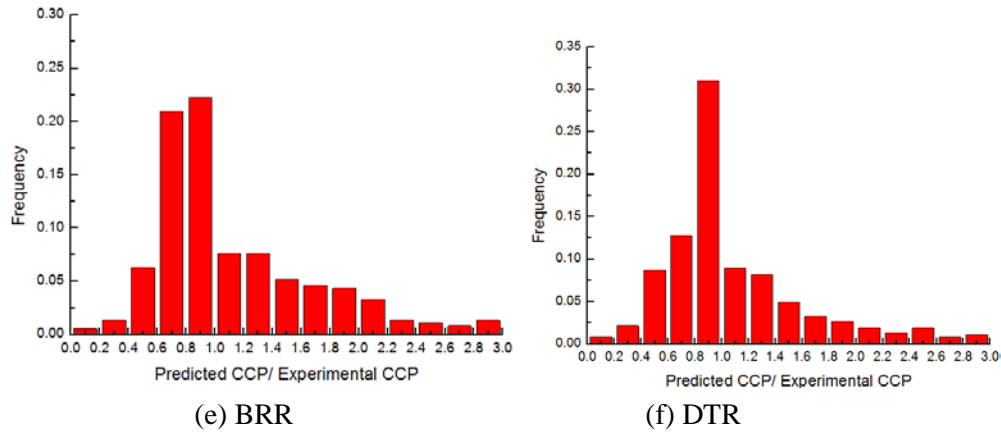


**Figure 4:** Performance of the 6 ML models on the testing set: (a) DTR; (b) SVR; (c) BRR; (d) KRR; (e) GBR; (f) ANN

The performance of the six ML models with the optimum hyper-parameters is presented on the testing set in Fig. 5 by the histogram. The predicted/experimental CCP values are analyzed. It visualizes the histogram plots of the predicted/experimental CCP values ratio by the optimum ML models on the testing set. It displays that the mean of density curves

are around one, which illustrates that the optimum ML models have a good predictive performance on testing sets. The GBR has the peak at 1.0, denoting that the optimum GBR model is accurate at predicting the CCP. The histogram for SVR, KRR, ANN and BRR models is slightly right-skewed of the peak ratio, indicating that the optimum ML model tends to predict a slightly greater CCP values than experimental values on the testing set. From this point, the performance of the optimum GBR model and DTR model on the testing set can be considered better as the peak ratio is approximate 1.0. It is worth mentioning that some predicted/experimental ratios are greater than 3 on the testing set, which is not shown in Fig. 5. The main reason of the discreteness in histograms is that the self-healing effect varies greatly in different cases in experiments. The randomness of the self-healing effect widely exists in experiments, which makes it difficult to determine the relationship between the influencing parameters and the healing result [Zhu, Zhou, Yan et al. (2015b)]. Hence, the experimental CCP data is scattered, which decreases the accuracy of ML models.

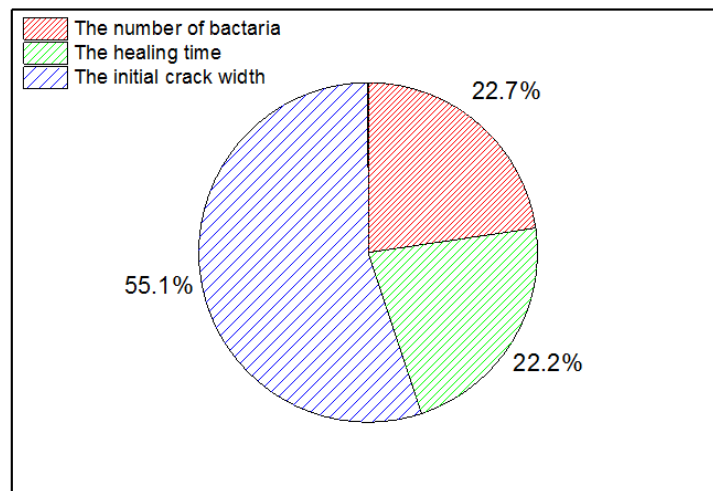




**Figure 5:** Histogram of predicted/experimental CCP values with testing sets using the ML models: (a) SVR; (b) ANN; (c) GBR; (d) KRR; (e) BRR; (f) DTR

### 5.3 Relative importance of influencing variables

The sensitivity analysis of the influencing variables is implemented in the optimum GBR models to investigate the effect of variables for the prediction of CCP. The GBR is selected according to its outstanding performance on the testing set. The importance is calculated by the Gini importance, which is a measure of variable importance based on the Gini impurity index [Breiman, Friedman, Olshen et al. (1984)]. Normalization is performed on the feature importance scores and the result is shown in Fig. 6.



**Figure 6:** Relative importance of influencing variables for crack healing

Apparently, the initial crack width is the most sensitive variable. It accounts for more than a half of importance score among all variables influencing the CCP, which agrees with previous research [Qian, Chen, Ren et al. (2015)]. Importance scores of the other two influencing variables on the crack healing are almost equal. The number of bacteria

accounts for 22.7%, which is a low value compared with the initial crack width. There are many reasons which contribute to its low value. Normal concrete samples, even without any bacteria, show some crack healing as well. The continued hydration process of cement particles, the swelling of cement matrix and the precipitation of calcium carbonate crystals can also heal the concrete to some extent [Wiktor and Jonkers (2011); Homma, Mihashi and Nishiwaki (2009); Khaliq and Ehsan (2016)]. Meanwhile, many bacteria are dead in concrete due to the pressure during the mixing stage and the formation of the dense microstructure after a period of time [Qian, Chen, Ren et al. (2015)]. Hence, the importance of the number of bacteria decreases. The healing time makes up 22.2%, which means the healing time is also a necessary variable in designing the bacteria-based self-healing concrete. Different importance scores may be evaluated with different dataset [Guyon, Gunn, Nikraves et al. (2006)]. More representative results can be obtained as more valid experimental results about CCP are available in the future.

#### **5.4 Contribution and limitations**

The primary strength of this study is to propose and compare six integrated ML approaches on predicting CCP of bacteria-based self-healing concretes. This study contributes to the concrete design and other fields of industrial engineering. On one hand, the integrated ML approaches based on ML algorithms and PSO are very promising for prediction problems. On the other hand, the model recommendation has been made for CCP and the methodology in this paper can be adopted for a wider application in the rest of industrial engineering.

The ignorance of other influencing variables for crack healing, such as the carrier of bacteria and the nutrient medium of bacteria, is a limitation of the present study since there is not enough experimental data. Meanwhile, the experimental results are scattered, which decreases the predictive accuracy of models. The performance of proposed ML approaches will improve when more experimental data is available in the future.

#### **6 Summary and conclusion**

In this study, a model for predicting CCP of the bacteria-based self-healing concrete is proposed based on ML and PSO. The ML algorithms are used for the non-linear relationship modeling between CCP and its influencing variables, and PSO is applied for the hyper-parameters tuning. The dataset is obtained through extensive experiments with different combinations of influencing variables. Inputs are selected to be the number of bacteria, the healing time and the initial crack width. The output is selected as CCP. 10-fold CV is used as the validation method. The performance of the optimum ML model is verified using the MSE and the R value.

The results show that the PSO-ML method has great potential for the prediction of CCP. PSO is efficient in the hyper-parameters tuning of ML models with minimum MSE values being achieved for all 6 ML methods. The optimum GBR model has a quite good performance on the training set and the testing set. The low MES and high R value between predicted CCP values and experiments on the training and testing sets denote that a good prediction is achieved by the optimum GBR model compared with other 5 models. The relative importance of influencing variables is studied, and the initial crack



width is found to be the most important variable. The influence of the healing time and the number of bacteria is less important compared with the initial crack width. The finding of this paper can be used for a more suitable bacteria-based self-healing concrete and in a wider application in the rest of industrial engineering.

**Acknowledgement:** This work was supported by Sofa-Kovalevskaja-Award of Alexander von Humboldt-Foundation.

## References

- Amiri, F.; Anitescu, C.; Arroyo, M.; Bordas, S.; Rabczuk, T.** (2014): XLME interpolants, a seamless bridge between XFEM and enriched meshless methods. *Computational Mechanics*, vol. 53, no. 1, pp. 45-57.
- Amiri, F.; Milan, D.; Shen, Y.; Rabczuk, T.; Arroyo, M.** (2014): Phase-field modeling of fracture in linear thin shells. *Theoretical and Applied Fracture Mechanics*, vol. 69, pp. 102-109.
- Anitescu, C.; Hossain, M. N.; Rabczuk, T.** (2018): Recovery-based error estimation and adaptivity using high-order splines over hierarchical T-meshes. *Computer Methods in Applied Mechanics and Engineering*, vol. 328, pp. 638-662.
- Areias, P.; Msekh, M. A.; Rabczuk, T.** (2016): Damage and fracture algorithm using the screened Poisson equation and local remeshing. *Engineering Fracture Mechanics*, vol. 158, pp. 116-143.
- Areias, P.; Rabczuk, T.** (2013): Finite strain fracture of plates and shells with configurational forces and edge rotation. *International Journal for Numerical Methods in Engineering*, vol. 94, no. 12, pp. 1099-1122.
- Areias, P.; Rabczuk, T.** (2017): Steiner-point free edge cutting of tetrahedral meshes with applications in fracture. *Finite Elements in Analysis & Design*, vol. 132, pp. 27-41.
- Areias, P.; Rabczuk, T.; Camanho, P. P.** (2014): Finite strain fracture of 2D problems with injected anisotropic softening elements. *Theoretical and Applied Fracture Mechanics*, vol. 72, pp. 50-63.
- Areias, P.; Rabczuk, T.; Dias-da-Costa, D.** (2013): Element-wise fracture algorithm based on rotation of edges. *Engineering Fracture Mechanics*, vol. 110, pp. 113-137.
- Areias, P.; Rabczuk, T.; Msekh, M. A.** (2016): Phase-field analysis of finite-strain plates and shells including element subdivision. *Computer Methods in Applied Mechanics and Engineering*, vol. 312, pp. 322-350.
- Areias, P.; Reinoso, J.; Camanho, P. P.; de Sa, J. C.; Rabczuk, T.** (2018): Effective 2D and 3D crack propagation with local mesh refinement and the screened Poisson equation. *Engineering Fracture Mechanics*, vol. 189, pp. 339-360.
- Armaghani, D. J.; Hajihassani, M.; Bejarbaneh, B. Y.; Marto, A.; Mohamad, E. T.** (2014): Indirect measure of shale shear strength parameters by means of rock index tests through an optimized artificial neural network. *Measurement*, vol. 55, pp. 487-498.
- Badawy, M. F.; Msekh, M. A.; Hamdia, K. M.; Steiner, M. K.; Lahmer, T. et al.**

(2017): Hybrid nonlinear surrogate models for fracture behavior of polymeric nanocomposites. *Probabilistic Engineering Mechanics*, vol. 50, pp. 64-75.

**Badnava, H.; Msekh, M. A.; Etemadi, E.; Rabczuk, T.** (2018): An h-adaptive thermo-mechanical phase field model for fracture. *Finite Elements in Analysis and Design*, vol. 138, pp. 31-47.

**Bashir, Z. A.; El-Hawary, M. E.** (2009): Applying wavelets to short-term load forecasting using PSO-based neural networks. *IEEE Transactions on Power Systems*, vol. 24, pp. 20-27.

**Breiman, L.; Friedman, J.; Olshen, R.; Stone, C. J.** (1984): *Classification and Regression Trees*. Wadsworth, Belmont.

**Budarapu, P. R.; Gracie, R.; Bordas, S.; Rabczuk, T.** (2014): An adaptive multiscale method for quasi-static crack growth. *Computational Mechanics*, vol. 53, no. 6, pp. 1129-1148.

**Budarapu, P. R.; Gracie, R.; Yang, S. W.; Zhuang, X.; Rabczuk, T.** (2014): Efficient coarse graining in multiscale modeling of fracture. *Theoretical and Applied Fracture Mechanics*, vol. 69, pp. 126-143.

**Castaigns, A.; Lhomme, W.; Trigui, R.; Bouscayrol, A.** (2016): Comparison of energy management strategies of a battery/supercapacitors system for electric vehicle under real-time constraints. *Applied Energy*, vol. 163, pp. 190-200.

**Chau-Dinh, T.; Zi, G.; Lee, P. S.; Rabczuk, T.; Song, J. H.** (2012): Phantom-node method for shell models with arbitrary cracks. *Computers & Structures*, vol. 92, no. 1, pp. 242-256.

**Chen, L.; Rabczuk, T.; Bordas, S. P. A.; Liu, G. R.; Zeng, K. Y. et al.** (2012): Extended finite element method with edge-based strain smoothing (ESm-XFEM) for linear elastic crack growth. *Computer Methods in Applied Mechanics and Engineering*, vol. 209, no. 1, pp. 250-265.

**Chilana, L.; Bhatt, A. H.; Najafi, M.; Sattler, M.** (2016): Comparison of carbon footprints of steel versus concrete pipelines for water transmission. *Journal of the Air & Waste Management Association*, vol. 66, pp. 518-527.

**Chipman, H. A.; George, E. I.; McCulloch, R. E.** (1998): Bayesian CART model search. *Journal of the American Statistical Association*, vol. 93, pp. 935-960.

**Cristianini, N.; Shawe-Taylor, J.** (2000): *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge.

**Das, S. K.; Biswal, R. K.; Sivakugan, N.; Das, B.** (2011): Classification of slopes and prediction of factor of safety using differential evolution neural networks. *Environmental Earth Sciences*, vol. 64, pp. 201-210.

**Eberhart, R. C.; Shi, Y.** (2001): Tracking and optimizing dynamic systems with particle swarms. *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 94-100.

**Friedman, J. H.** (2002): Stochastic gradient boosting. *Computational Statistics & Data Analysis*, vol. 38, pp. 367-378.

**Geng, B.; Mills, J. K.; Sun, D.** (2014): Combined power management/design

optimization for a fuel cell/battery plug-in hybrid electric vehicle using multi-objective particle swarm optimization. *International Journal of Automotive Technology*, vol. 15, pp. 645-654.

**Ghorashi, S.; Valizadeh, N.; Mohammadi, S.; Rabczuk, T.** (2015): T-spline based XIGA for fracture analysis of orthotropic media. *Computers & Structures*, vol. 147, pp. 138-146.

**Gollapudi, U. K.; Knutson, C. L.; Bang, S. S.; Islam, M. R.** (1995): A new method for controlling leaching through permeable channels. *Chemosphere*, vol. 30, pp. 695-705.

**Gordan, B.; Jahed Armaghani, D.; Hajihassani, M.; Monjezi, M.** (2016): Prediction of seismic slope stability through combination of particle swarm optimization and neural network. *Engineering with Computers*, vol. 32, pp. 85-97.

**Guyon, I. M.; Gunn, S. R.; Nikraves, M.; Zadeh, L.** (2006): Feature extraction, foundations and applications. *Studies in Fuzziness & Soft Computing*, vol. 205, pp. 68-84.

**Hadavandi, E.; Shavandi, H.; Ghanbari, A.** (2010): Integration of genetic fuzzy systems and artificial neural networks for stock price forecasting. *Knowledge-Based Systems*, vol. 23, pp. 800-808.

**Hamdia, K.; Silani, M.; Zhuang, X.; He, P.; Rabczuk, T.** (2017): Stochastic analysis of the fracture toughness of polymeric nanoparticle composites using polynomial chaos expansions. *International Journal of Fracture*, vol. 206, no. 2, pp. 215-227.

**Hoang, N.; Pham, A.** (2016): Hybrid artificial intelligence approach based on metaheuristic and machine learning for slope stability assessment: A multinational data analysis. *Expert Systems with Applications*, vol. 46, pp. 60-68.

**Homma, D.; Mihashi, H.; Nishiwaki, T.** (2009): Self-healing capability of fibre reinforced cementitious composites. *Journal of Advanced Concrete Technology*, vol. 7, pp. 217-228.

**Jonkers, H. M.** (2007): Self healing concrete: a biological approach. *Self-Healing Materials*, vol. 100, pp. 195-204.

**Jonkers, H. M.; Thijssen, A.; Muyzer, G.; Copuroglu, O.; Schlangen, E.** (2010): Application of bacteria as self-healing agent for the development of sustainable concrete. *Ecological Engineering*, vol. 36, pp. 230-235.

**Kennedy, J.; Eberhart, R.** (1995): Particle swarm optimization. *International Conference on Neural Networks*, pp. 1942-1948.

**Khaliq, W.; Ehsan, M. B.** (2016): Crack healing in concrete using various bio influenced self-healing techniques. *Construction and Building Materials*, vol. 102, pp. 349-357.

**Khosravi, A.; Nahavandi, S.; Creighton, D.** (2013): Quantifying uncertainties of neural network-based electricity price forecasts. *Applied Energy*, vol. 112, pp. 120-129.

**Khotanzad, A.; Elragal, H.; Lu, T. L.** (2000): Combination of artificial neural-network forecasters for prediction of natural gas consumption. *IEEE Transactions on Neural Networks*, vol. 11, pp. 464-473.

**Long, V. T.; Nhan, N. V.** (2012): Bees-algorithm-based optimization of component size

and control strategy parameters for parallel hybrid electric vehicles. *International Journal of Automotive Technology*, vol. 13, pp. 1177-1183.

**Luo, M.; Qian, C. X.; Li, R. Y.** (2015): Factors affecting crack repairing capacity of bacteria-based self-healing concrete. *Construction and Building Materials*, vol. 87, pp. 1-7.

**Ma, H.; Xu, H.; Wang, J.; Schnier, T.; Neaves, B. et al.** (2015): Model-based multiobjective evolutionary algorithm optimization for HCCI engines. *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 4326-4331.

**MacKay, D. J.** (1992): Bayesian interpolation. *Neural Computation*, vol. 4, pp. 415-447.

**Montazeri-Gh, M.; Poursamad, A.; Ghalichi, B.** (2006): Application of genetic algorithm for optimization of control strategy in parallel hybrid electric vehicles. *Journal of the Franklin Institute*, vol. 343, pp. 420-435.

**Msekh, M. A.; Cuong, N. H.; Zi, G.; Areias, P.; Zhuang, X. et al.** (2017): Fracture properties prediction of clay/epoxy nanocomposites with interphase zones using a phase field model. *Engineering Fracture Mechanics*, vol. 188, pp. 287-299.

**Murphy, K. P.** (2012): Machine learning: a probabilistic perspective. *Adaptive Computation and Machine Learning*, MIT Press, Cambridge, MA.

**Nanthakumar, S.; Lahmer, T.; Zhuang, X.; Zi, G.; Rabczuk, T.** (2016): Detection of material interfaces using a regularized level set method in piezoelectric structures. *Inverse Problems in Science and Engineering*, vol. 24, no. 1, pp. 153-176.

**Nguyen-Thanh, N.; Valizadeh, N.; Nguyen, M. N.; Nguyen-Xuan, H.; Zhuang, X. et al.** (2015): An extended isogeometric thin shell analysis based on Kirchhoff-Love theory. *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 265-291.

**Nguyen-Xuan, H.; Liu, G. R.; Bordas, S.; Natarajan, S.; Rabczuk, T.** (2013): An adaptive singular ES-FEM for mechanics problems with singular field of arbitrary order. *Computer Methods in Applied Mechanics and Engineering*, vol. 253, pp. 252-273.

**Poli, R.** (2008): Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Application*, vol. 2008, pp. 1-10.

**Qi, C.; Tang, X.** (2018): Slope stability prediction using integrated metaheuristic and machine learning approaches: A comparative study. *Computers & Industrial Engineering*, vol. 118, pp. 112-122.

**Qi, C.; Fourie, A.; Chen, Q.; Zhang, Q.** (2018): A strength prediction model using artificial intelligence for recycling waste tailings as cemented paste backfill. *Journal of Cleaner Production*, vol. 183, pp. 566-578.

**Qian, C.; Chen, H.; Ren, L.; Luo, M.** (2015): Self-healing of early age cracks in cement-based materials by mineralization of carbonic anhydrase microorganism. *Frontiers in Microbiology*, vol. 6, pp. 1225.

**Quayum, M. S.; Zhuang, X.; Rabczuk, T.** (2015): Computational model generation and RVE design of self-healing concrete. *Frontiers of Structural Civil Engineering*, vol. 9, no. 4, pp. 383-396.

**Rabczuk, T.; Areias, P.; Belytschko, T.** (2007): A meshfree thin shell method for non-linear dynamic fracture. *International Journal for Numerical Methods in Engineering*,

vol. 72, no. 5, pp. 524-548.

**Rabczuk, T.; Gracie, R.; Song, J. H.; Belytschko, T.** (2010): Immersed particle method for fluid-structure interaction. *International Journal for Numerical Methods in Engineering*, vol. 81, no. 1, pp. 48-71.

**Ramadan Suleiman, A.; Nehdi, M. L.** (2017): Modeling self-healing of concrete using hybrid genetic algorithm-artificial neural network. *Materials*, vol. 10, pp. 135.

**Ren, H.; Zhuang, X.; Cai, Y.; Rabczuk, T.** (2016): Dual-Horizon peridynamics. *International Journal for Numerical Methods in Engineering*, vol. 108, no. 12, pp. 1451-1476.

**Ren, H.; Zhuang, X.; Rabczuk, T.** (2017): Dual-horizon peridynamics: a stable solution to varying horizons. *Computer Methods in Applied Mechanics and Engineering*, vol. 318, pp. 762-782.

**Rumelhart, D. E.; Hinton, G. E.; Williams, R. J.** (1986): Learning representations by back-propagating errors. *Nature*, vol. 323, pp. 533-536.

**Smith, G. N.** (1986): Probability and statistics in civil engineering. *Collins Professional and Technical Books*, pp. 244.

**Stone, M.** (1974): Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, vol. 36, pp. 111-147.

**Stuckrath, C.; Serpell, R.; Valenzuela, L. M.; Lopez, M.** (2014): Quantification of chemical and biological calcium carbonate precipitation: Performance of self-healing in reinforced mortar containing chemical admixtures. *Cement and Concrete Composites*, vol. 50, pp. 10-15.S

**Talebi, H.; Silani, M.; Bordas, S.; Kerfriden, P.; Rabczuk, T.** (2014): A computational library for multiscale modelling of material failure. *Computational Mechanics*, vol. 53, no. 5, pp. 1047-1071.

**Talebi, H.; Silani, M.; Rabczuk, T.** (2015): Concurrent multiscale modelling of three dimensional crack and dislocation propagation. *Advances in Engineering Software*, vol. 80, pp. 82-92.

**van der Gaag, M.; Hoffman, T.; Remijnsen, M.; Hijman, R.; de Haan, L. et al.** (2006): The five-factor model of the Positive and Negative Syndrome Scale II: a ten-fold cross-validation of a revised model. *Schizophrenia Research*, vol. 85, pp. 280-287.

**Vapnik, V. N.** (1999): An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, vol. 10, pp. 988-999.

**Vu-Bac, N.; Lahmer, T.; Zhuang, X.; Nguyen-Thoi, T.; Rabczuk, T.** (2016): A software framework for probabilistic sensitivity analysis for computationally expensive models. *Advances in Engineering Software*, vol. 100, pp. 19-31.

**Wiktor, V.; Jonkers, H. M.** (2011): Quantification of crack-healing in novel bacteria-based self-healing concrete. *Cement and Concrete Composites*, vol. 33, pp. 763-770.

**Yang, X. S.** (2014): *Nature-Inspired Optimization Algorithms*. Elsevier, Waltham, MA.

**Zhang, J.; Zhou, A.; Liu, Y.; Zhao, B.; Luan, Y. et al.** (2017): Microbial network of the carbonate precipitation process induced by microbial consortia and the potential



application to crack healing in concrete. *Scientific Reports*, vol. 7, pp. 14600.

**Zhang, J.; Liu, Y.; Feng, T.; Zhou, M.; Zhao, L. et al.** (2017): Immobilizing bacteria in expanded perlite for the crack self-healing in concrete. *Construction and Building Materials*, vol. 148, pp. 610-617.

**Zhang, L.; Wu, K.; Zhong, Y.; Li, P.** (2008): A new sub-pixel mapping algorithm based on a BP neural network with an observation model. *Neurocomputing*, vol. 71, pp. 2046-2054.

**Zhang, Y.; Zhuang, X. Y.** (2018): A softening-healing law for self-healing quasi-brittle materials: analyzing with strong discontinuity embedded approach. *Engineering Fracture Mechanics*, vol. 192, pp. 290-306.

**Zhou, S.; Zhu, H. H.; Ju, J. W.; Yan, Z. G.; Chen, Q.** (2017): Modeling microcapsule-enabled self-healing cementitious composite materials using discrete element method. *International Journal of Damage Mechanics*, vol. 26, pp. 340-357.

**Zhou, S.; Zhu, H. H.; Yan, Z. G.; Ju, J. W.; Zhang, L.** (2016): A micromechanical study of the breakage mechanism of microcapsules in concrete using PFC2D. *Construction and Building Materials*, vol. 115, pp. 452-463.

**Zhu, H. H.; Chen, Q.; Yan, Z. G.; Ju, J. W.; Zhou, S.** (2014): Micromechanical models for saturated concrete repaired by the electrochemical deposition method. *Materials and Structures*, vol. 47, pp. 1067-1082.

**Zhu, H. H.; Zhou, S.; Yan, Z. G.; Ju, J. W.; Chen, Q.** (2016): A two-dimensional micromechanical damage-healing model on microcrack-induced damage for microcapsule-enabled self-healing cementitious composites under compressive loading. *International Journal of Damage Mechanics*, vol. 25, pp. 727-749.

**Zhu, H. H.; Zhou, S.; Yan, Z. G.; Ju, J. W.; Chen, Q.** (2015a): A two-dimensional micromechanical damage-healing model on microcrack-induced damage for microcapsule-enabled self-healing cementitious composites under tensile loading. *International Journal of Damage Mechanics*, vol. 24, pp. 95-115.

**Zhu, H. H.; Zhou, S.; Yan, Z. G.; Ju, J. W.; Chen, Q.** (2015b): A 3D analytical model for the probabilistic characteristics of self-healing model for concrete using spherical microcapsule. *Computers and Concrete*, vol. 15, pp. 37-54.

**Zhuang, X. Y.; Cai, Y. C.; Augarde, C.** (2014): A meshless sub-region radial point interpolation method for accurate calculation of crack tip fields. *Theoretical and Applied Fracture Mechanics*, vol. 69, no. 1, pp. 118-125.

**Zhuang, X. Y.; Huang, R.; Rabczuk, T.; Liang, C.** (2014): A coupled thermo-hydro-mechanical model of jointed hard rock for compressed air energy storage. *Mathematical Problems in Engineering*, vol. 2014, pp. 1-11.

**Zhuang, X.; Zhou, S.** (2018): Molecular dynamics study of an amorphous polyethylene/silica interface with shear tests. *Materials*, vol. 11, pp. 929.

**Zhuang, X. Y.; Zhu, H. H.; Augarde, C.** (2014): An improved meshless Shepard and least square method possessing the delta property and requiring no singular weight function. *Computational Mechanics*, vol. 53, no. 2, pp. 343-357.