**RESEARCH ARTICLE**

# Prediction of shield tunneling-induced ground settlement using machine learning techniques

**Renpeng CHEN[a,b,c], Pin ZHANG[c*], Huaina WU[a,b,c*], Zhiteng WANG[c], Zhiquan ZHONG[d]**

[a] *Key Laboratory of Building Safety and Energy Efficiency, Hunan University, Changsha 410082, China*

[b] *National Joint Research Center for Building Safety and Environment, Hunan University, Changsha 410082, China*

[c] *College of Civil Engineering, Hunan University, Changsha 410082, China*

[d] *China Construction Fifth Engineering Division Co., Ltd, Changsha 410082, China*

[*] *Corresponding authors. E-mails: zhangp@hnu.edu.cn; wuhn@hnu.edu.cn*

**ABSTRACT**   Predicting the tunneling-induced maximum ground surface settlement is a complex problem since the settlement depends on plenty of intrinsic and extrinsic factors. This study investigates the efficiency and feasibility of six machine learning (ML) algorithms, namely, back-propagation neural network, wavelet neural network, general regression neural network (GRNN), extreme learning machine, support vector machine and random forest (RF), to predict tunneling-induced settlement. Field data sets including geological conditions, shield operational parameters, and tunnel geometry collected from four sections of tunnel with a total of 3.93 km are used to build models. Three indicators, mean absolute error, root mean absolute error, and coefficient of determination the ($R^2$) are used to demonstrate the performance of each computational model. The results indicated that ML algorithms have great potential to predict tunneling-induced settlement, compared with the traditional multivariate linear regression method. GRNN and RF algorithms show the best performance among six ML algorithms, which accurately recognize the evolution of tunneling-induced settlement. The correlation between the input variables and settlement is also investigated by Pearson correlation coefficient.

## 1   Introduction

The increasing traffic pressure in cities of China has led to a large number of metros being built [1–5]. The metro tunnels are generally constructed by shield method due to its advantages of high speeds, minor disturbance to the surface traffic, etc. [6,7]. During construction of the shield tunnels, ground settlement will be inevitably caused due to disturbance to the soil layers around tunnels and the volume loss of the tail void, which may pose a risk to the surrounding infrastructures [8–12]. To predict tunneling-induced ground settlement, many endeavors have been made, e.g., to propose empirical formulae [13–15], analytical solutions [16,17], or to conduct elaborate numerical simulations [18–20]. However, these methods still have many limitations, e.g., inapplicable to complex ground conditions and construction techniques [21],

difficult to identify the parameters of the sophisticated soil constitutive models, hard to model the tunnel construction process [22,23], etc. In the past few decades, ML has become strong tool to solve nonlinear problems with high dimension [24–26], since these algorithms can effectively capture the nonlinear relationship among the influential factors with less exertion. Some of these algorithms are able to achieve the transformation of parameters space, which prompts to find out the nature of the problems [27,28].

ML algorithms adopted on predicting tunneling-induced ground settlement in the previous literature can be predominantly divided into the following two categories: artificial neural network (ANN) and support vector machine (SVM). Back-propagation neural network (BPNN) is the first type of ANN used for predicting settlement and its robustness is demonstrated to be acceptable [29–32]. Wavelet neural network (WNN) developed upon the integration between wavelet theory

and ANN, enhances the function approximation capability [20]. Then more hybrid ANNs (e.g., particle swarm optimization (PSO)-ANN, etc.) were employed to estimate the tunneling-induced settlement [33,34]. The objective of these hybrid algorithms is to determine the optimum parameters used in ANNs, and then the global optimum results can be gained. In recent years, SVM as a new powerful ML algorithm for classification and regression, has been extensively utilized in geotechnical areas [35–38]. The main characteristic of SVM is that it is developed upon the structural risk minimization (SRM), unlike the empirical risk minimization (ERM) used in ANNs [39]. Accordingly, SVM possesses good estimation capabilities in dealing with problems with a small sample size. Similar to ANNs, the integration of SVM and other optimization techniques can also improve the accuracy of models. Zhang et al. [23] proposed a hybrid algorithm which integrates the PSO and least-squares (LSSVM) to predict tunneling-induced settlement. More recent work by Kohestani et al. [40] introduced an ensemble learning algorithm, known as random forest (RF), to predict the tunneling-induced settlement since the RF algorithm accommodates big data with a relatively short learning time [41].

More ML algorithms such as radial basis function neural network (RBF), general regression neural network (GRNN), extreme learning machine (ELM), have successfully applied in other domain [42–44]. As an alternative approach, data-driven models using different ML algorithms are promising techniques for prediction and classification. However, there are still several problems need to be addressed: 1) There is no direction or analytical solution for identifying which one is the most suitable algorithm for a specific problem. In general, a methodology is to compare the performance of different ML algorithms in a given setting [45–47]. 2) A systematic, quantitative comparison of the available ML algorithms still lacks, although their performance in predicting tunneling-induced settlement may be different dramatically. Accordingly, it is urgent to conduct a benchmark study in the application of ML algorithms in predicting tunneling-induced settlement, which is of great significance during tunnel construction.

To address above problem, the paper compares the feasibility and applicability of six ML algorithms BPNN, WNN, GRNN, ELM, SVM, RF in estimating tunneling-induced settlements. In addition, multivariate linear regression method is also used for predicting as a comparison. Field data sets including geological, tunnel geometric, shield operational parameters and maximum ground surface settlement were collected from the project of Changsha Metro Line 4, China. MATLAB software is employed to run the ML algorithms in this paper.

The reminder of the paper is organized as follows: Section 2 introduces the overview of the Changsha Metro Line 4 project, tunnel construction method and data

sources. In Section 3, six ML algorithms have been presented as well as the performance evaluation indicators. The results of six ML algorithms including the determination of parameters and architectures of each ML algorithm and corresponding predictions have been presented in Section 4. The comparison of the performance of each ML algorithm in predicting tunneling-induced settlement is shown in Section 5.

## 2  Project background

### 2.1  Project description and ground condition

In this study, construction of four tunnel sections on Metro Line 4 of Changsha, China, was investigated. The four tunnel sections with the total length of 3.93 km are located on the west bank of Xiangjiang River. The four sections from north to south are as follows: 1) Liugoulong-Wangyuehu (LW section); 2) Wangyuehu-Yingwanzhen (WY section); 3) Yingwanzhen-Hunan Normal University (YH section); 4) Hunan Normal University-Hunan University (HH section).

The geological profile along the tunnel axis is shown in Fig. 1. The cover depth of tunnel varies from 10 to 28 m. The water table is approximately 5 m below the ground surface. The ground encountered by the shield tunnel can be divided into two sections: rock zones and soil zones. Rock zones is underlain by rocks with different weathering grades. The rocks in the LW section are the slightly weathered slate and moderately weathered slate. The weathered limestone and slate spread in the WY section. YH and HH sections mainly consist of weathered sandstone and mudstone. Soil zones means that the soil around the tunnel consists of clay and gravel. The top layer is backfills, which consists of clay, sand, and discarded concrete blocks. Underlying the backfill layer is clay with an average 3 m thick. The gravel layer scatters in the WY section.

### 2.2  Construction method

The metro tunnel was constructed by earth pressure balanced (EPB) shield-driven method. The cutterhead diameter and length of EPB shields used in this project are 6.28 and 8.735 m, respectively. The open ratio of cutterhead is 35%. The tunnel lining is composed of six precast concrete segments, which form rings. The outer and inner diameter of the segmental lining are 6 and 5.4 m, respectively. The ring width is 1.5 m.

The advancement of the EPB shield inevitably leads to soil disturbance. Figure 2 schematically illustrates the shield tunneling process and the key operational parameters. Face pressure that sustains the face stability is supplied by the cutterhead and the soil in the chamber. The volume of excavated soils is control by the trade-off
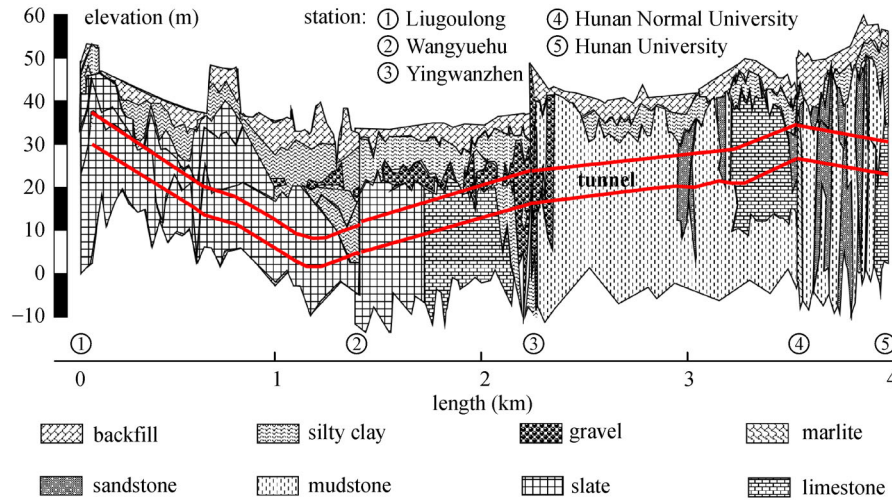
**Fig. 1**  Geological profile at the Changsha Metro Line 4 construction site.



Note: $D$ = diameter of cutterhead; $d_{in}$, $d_{out}$ = inner and outer diameters of segmental lining.
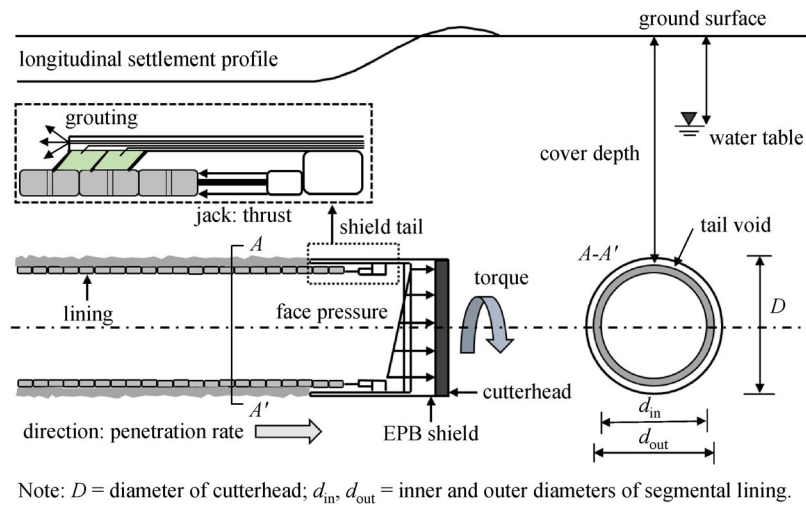
**Fig. 2**  Factors of tunneling-induced surface settlement.

between the rate of extraction of spoil through the screw and the shield penetration rate. If the penetration rate is higher than the extraction rate, that is, the volume of extraction is less than the volume replaced by the shield body, the shield may generate a much higher face pressure and the ground surface heave will occur. Otherwise, if the penetration rate is less than the extraction rate, a surface settlement will occur, which may even cause tunnel face instability. Moreover, the thrust device of a EPB shield consists of several parallel-arranged hydro-cylinders, which are jointed to the cutterhead and the segments with spherical hinges [48]. The thrust is mainly used to overcome the friction force between the shield body and the surrounding soil layers as well as the force at the tunnel face. The torque acting on the cutter head is produced during the process of cutting soils. It always changes with

the variation of face pressure and thrust. Grouting filling is an important factor in contributing to the surface settlement. As the shield is jacked forward, a tail void around the outside of lining is created as shown in Fig. 2. Tail void grouting is necessary in order to prevent the soil around tunnel moving toward the void. The five operational parameters presented above thrust ($Th$), torque ($To$), face pressure ($Fp$), penetration rate ($Pr$), grout filling ($Gf$) are the key factors affecting tunneling-induced settlement, and these parameters will be considered as input variables in the computational models.

### 2.3 Data sources

The factors relevant to tunneling-induced settlement can be classified into three categories [32]: tunnel geometry,

geological condition, and shield operational parameters. In this paper, ten input variables are considered including five shield operational parameters discussed in Section 2.2, four geological parameters blow counts of standard penetration test (MSPT) and dynamic penetration test (MDPT) of soil layers, uniaxial compressive strength of weathered rocks (MUCS) and groundwater table (W). One geometry parameter is the cover depth of tunnels (C). Due to the consistent tunnel specification in this project, the overburden of the tunnel is the unique geometry factor. Herein, geological parameters were obtained from the geological investigated reports and operational parameters were collected per minute from the automatic collecting equipment embedded in the EPB shield system. Settlement monitoring points were installed at about 7.5 m intervals along the tunnel alignment and the ground settlement were obtained from manual records. Finally, a database including 200 monitoring points installed along the tunnel

centerline is established. Tables 1 and 2 show the range of input and output variables for training and test sets, respectively. In this research, 80% of data sets arbitrarily sampled out of database are considered for the training set, and the remaining 20% are used to test model.

The database applied to the training model is mapped to the interval (−1, 1) using a data normalization algorithm since it can reduce the computational cost in this way. For a parameter $x$, the normalized value $x$ is obtained from

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}(\bar{x}_{\max} - \bar{x}_{\min}) + \bar{x}_{\min}, \tag{1}$$

where $x_{\max}$ and $x_{\min}$ = the maximum and minimum value of the variable $x$, $\bar{x}_{\max}$ and $\bar{x}_{\min}$ = the maximum and minimum values of the variable $x$ after normalization. The final outputs need to be transformed into the original vector space.

**Table 1**  Ranges of all variables for training set

| type | variable (unit) | data (200) | | | unit |
|------|-----------------|------|------|------|------|
| | | Min. | Max. | Ave. | |
| geometry | cover depth (C) | 13.40 | 31.70 | 19.14 | m |
| operation | torque (To) | 0.74 | 3.95 | 2.22 | MN·m |
| | penetration rate (Pr) | 3.46 | 51.00 | 23.91 | mm/rev |
| | thrust (Tr) | 3.80 | 24.20 | 13.04 | MN |
| | face pressure (Fp) | 0.00 | 2.10 | 1.16 | bar |
| | grout filling (Gf) | 4.00 | 11.00 | 5.63 | m³ |
| geology | tunnel depth below the water table (W) | 4.05 | 25.38 | 11.99 | m |
| | modified standard penetration test (MSPT) | 0.00 | 38.72 | 7.78 | – |
| | modified dynamic penetration test (MDPT) | 0.00 | 12.44 | 0.55 | – |
| | modified uniaxial compressive strength (MUCS) | 0.00 | 36.30 | 8.28 | MPa |
| output | maximum settlement (S) | −19.13 | 3.05 | −2.50 | mm |

**Table 2**  Ranges of all variables for test set

| type | variable (unit) | data (200) | | | unit |
|------|-----------------|------|------|------|------|
| | | Min. | Max. | Ave. | |
| geometry | cover depth (C) | 14.10 | 29.70 | 19.39 | m |
| operation | torque (To) | 1.11 | 3.20 | 2.29 | MN·m |
| | penetration rate (Pr) | 5.33 | 54.60 | 26.54 | mm/rev |
| | thrust (Tr) | 4.38 | 20.70 | 12.43 | MN |
| | face pressure (Fp) | 0.00 | 2.20 | 1.18 | bar |
| | grout filling (Gf) | 4.00 | 6.90 | 5.54 | m³ |
| geology | tunnel depth below the water table (W) | 5.05 | 23.48 | 12.22 | m |
| | modified standard penetration test (MSPT) | 0.01 | 32.29 | 6.51 | – |
| | modified dynamic penetration test (MDPT) | 0.00 | 5.66 | 0.29 | – |
| | modified uniaxial compressive strength (MUCS) | 0.00 | 31.62 | 8.99 | MPa |
| output | maximum settlement (S) | −12.13 | 2.52 | −1.90 | mm |

# 3 Machine learning methodology

## 3.1 Description of the algorithms

### 3.1.1 Multivariate linear regression

The multivariate linear regression (MLR) is the simplest regression method. For a given input vector $\boldsymbol{X} = (x_1, x_2, x_3, ..., x_n)$ and an output $y$, the linear regression can be obtained based on the following equation:

$$y = b + \sum_{i=1}^{n} a_i x_i, \tag{2}$$

where $a_i$, $b$ = fitting parameters of the model; $n$ = number of features.

### 3.1.2 BPNN

The BPNN is one of the most popular feedforward neural networks. BPNN algorithm develops on the basis of the architecture of the multilayer perceptron neural network which consists of an input layer, one or multiple hidden layers and an output layer. By adjusting the number of hidden layers and neurons, an optimal BPNN can be obtained. Each neuron in the hidden layer is associated with a given weight and bias term to transmit inputs from input layer to outputs. The output of the hidden layer in BPNN is expressed as:

$$y_j = f\left(\sum_{i=1}^{m} \omega_{ji} x_i + \theta_j\right), \tag{3}$$

where $x_i$ = input value, $y_j$ = result from hidden neuron $j$, $\omega_{ji}$ = weights on the connection between the input and the hidden neuron $j$, $\theta_j$ = bias term, $f$ = activation function, $m$ = the number of neurons in the input layer. The $\omega_{j,i}$ and $\theta_j$ values are determined randomly for the initialization in computer software Matlab. To account for the nonlinearity in this problem, the activation function used herein is a tansig function defined as:

$$f(I) = \frac{-2}{1 + \exp(-2I)} - 1, \tag{4}$$

where $I = \sum_{i=1}^{m} \omega_{ji} x_i + \theta_j$. Similar to the process in hidden layer, the results of hidden layer are finally mapped into output layer by a linear or nonlinear function.

### 3.1.3 WNN

The WNN is an algorithm developed on the basis of the topology of neural network. This algorithm combines the theory of wavelet transform and neural network. WNN model generally has a feedforward structure with a fixed hidden layer. The activation functions employed in the hidden layer are drawn from an orthonormal wavelet family. The theoretical features of wavelet transform helps determine the neural network parameters during the training process [22]. It has been proved that WNN models provide an approach to avoid local minima, accelerate convergence and identify optimum network structures more efficiently. In this paper, the performance of WNN models are dependent on wavelet functions. Pourtaghi and Lotfollahi-Yaghin [22] proposed that the network with Morlet wavelet show the best performance in predicting tunneling-induced settlement by comparing the performance of different wavelet functions. So Morlet mother wavelet function is also chosen as the activation function of hidden layer in this study, which is expressed as:

$$\varphi(t) = \cos(1.75t)\mathrm{e}^{-t^2/2}, \tag{5}$$

where $\varphi$ = Morlet mother wavelet function, $t$ = input variable.

Assuming that the number of neurons in input layer, hidden layer and output layer are $m$, $n$, and $N$, respectively. The output of $j$th neuron in hidden layer is:

$$z_j = \varphi\left(\frac{\sum\limits_{i=1}^{m} \omega_{ij} x_i - b_j}{a_j}\right), \tag{6}$$

where $b_j$ = shift factor of $\varphi$, $a_j$ = the retractable factor of $\varphi$, $\omega_{ij}$ = weights on the connection between the input and the hidden neuron $j$. The final output of WNN is as follows:

$$y_k = \sum_{j=1}^{n} \omega_{jk} z_j, \tag{7}$$

where $\omega_{jk}$ = weights on the connection between the hidden neurons the output neuron.

### 3.1.4 GRNN

GRNN is a variation of RBF neural network which is designed for function approximation. A GRNN consists of four layers: input layer, hidden layer, summation layer and output layer. The summation layer includes only two neurons. The first neuron of summation layer sums all outputs of hidden layer that is the numerator of equation. The second neuron is equal to unity.

The GRNN does not need to repeat the iterative training process and initialize the network connection weights, the main objective of learning is to find out the best smoothing parameter ($\sigma$). Provided that $f(x, y)$ stands for the joint probability density function of random variables $x$ and $y$, then the observe value of $x$ is $x_0$, the regression of $y$ on $x$ is given by:

$$E[y|x_0] = y(x_0) = \frac{\int_{-\infty}^{\infty} y f(x_0, y) \mathrm{d}y}{\int_{-\infty}^{\infty} f(x, y) \mathrm{d}y}, \qquad (8)$$

where $y(x_0)$ = predictive output of $y$ with respect to $x_0$, $n$ = the number of sample of observations. The density function $f(x_0, y)$ can be calculated according to the following formulae:

$$f(x_0, y) = \frac{1}{n(2\pi)^{(p+1)/2} \sigma^{p+1}} \sum_{i=1}^{n} \mathrm{e}^{-d(x_0, x_i)} \mathrm{e}^{-d(y, y_i)}, \qquad (9)$$

$$d(x_0, x_i) = \sum_{j=1}^{p} [(x_{0j} - x_{ij})/\sigma]^2, \qquad (10)$$

$$d(y, y_i) = (y - y_i)^2, \qquad (11)$$

where $p$ = dimension of vector variable $x$, $\sigma$ = smoothing parameter. Considering with the above formulae, the $y(x_0)$ can be defined as:

$$y(x_0) = \frac{\sum_{i=1}^{n} y \mathrm{e}^{-d(y, y_0)}}{\sum_{i=1}^{n} \mathrm{e}^{-d(x_0, x_i)}}. \qquad (12)$$

Noting that numerator of Eq. (12) is the value of first neuron in summation layer and denominator is equal to the value of second neuron.

### 3.1.5  ELM

ELM is a modification of the single-hidden layer feedforward neural network, which can be obtained by removing back-propagation from a multilayer perceptron. In contrast to other ML algorithm like back-propagation and SVM, ELM provides a nonlinear model at the speed of a linear model. It randomly chooses hidden nodes and analytically determines the output weights of SLFNs algorithms. In addition, all the hidden layers in hierarchical ELM as a whole are not required to be iteratively tuned and it only needs to compute output weights. In general, ELM can be represented as:

$$y = H\beta, \qquad (13)$$

$$\text{Minimize}: \|H\beta - y\|, \qquad (14)$$

where $H$ = the hidden layer output matrix of the neural network, $\beta$ = weight vector connecting the hidden nodes and the output nodes, $y$ = the observations of outputs. The optimum ELM model can be achieved by minimizing the value of $\|H\beta - y\|$. Detailed description of the ELM algorithm can be referred to Huang et al. [49].

### 3.1.6  SVM

SVM is a ML tool that uses statistical learning theory to solve multiple-dimensional functions. The SVM regression reduces the error bound rather than the residual error on the training data set [39]. Hence, SVM aims to find a function $f_{sv}(x)$ that has at most a deviation $\varepsilon$ from each of the targets in the training set [43]. For linear function, $f_{sv}(x)$ can be written as:

$$f_{sv}(x) = \sum_{i=1}^{n} \omega_i \varphi_i(x) + b, \qquad (15)$$

where $\varphi$ = mapping function from original data to a high-dimension feature space, $\omega_i$ = weight vector, $b$ = threshold of SVM. The SVM formulates the regression as an optimization problem in the primal weight space. As given by:

$$\text{minimize}: \frac{1}{2}|\omega|^2,$$

$$\text{subject to}: \left| \sum_{i=1}^{M} \omega_i \phi_i(x) + b - y_i \right| \leqslant \varepsilon. \qquad (16)$$

In addition, the kernel function as an mapping function can handle nonlinear cases [43]. The Gaussian kernel function as follow is used in this paper.

$$\phi(x_i, x) = \exp\left( \frac{-\|x - x_i\|}{2\sigma^2} \right), \qquad (17)$$

where $\sigma$ = the width of the Gaussian kernel function, also known as smoothing parameter.

### 3.1.7  RF

RF is an ensemble learning algorithm for classification and regression tasks. Two powerful ML techniques, bootstrap aggregating [50] and random subspace [51] are integrated into the RF algorithm. In bagging, $n$ bootstrap sets are made by sampling with replacement $N$ training examples from the training set. The number of samples and features in bootstrap is arbitrary, less than the original training set. Then a decision tree is built using a bootstrap sample. A decision tree classifies a bootstrap sample by testing attributes of this bootstrap sample at each node. Each node tests a particular attribute, with the leaves of the tree representing the output labels. Moving down a particular branch of a tree tests particular attributes at each node in order to arrive to an output label. The final result aggregate the outputs from all leaves [52].

The output of the RF prediction can be expressed as:

$$y = \frac{1}{n_{\text{tree}}} \sum_{i=1}^{n_{\text{tree}}} y_i(x), \qquad (18)$$

where $y$ = average output of RF prediction from a total amount of $n_{\text{tree}}$, $y_i(\boldsymbol{x})$ = individual prediction of a tree for an input vector $\boldsymbol{x}$.

## 3.2 Performance evaluation method

Performance evaluation is generally conducted to assess the accuracy of model. An important contribution of Tseranidis et al. [53] is a list of eight measures of error to assess model performance. Three performance indicators: mean absolute error (*MAE*), root mean square error (*RMSE*), and coefficient of determination ($R^2$) are selected to demonstrate the correspondence between predictions and measurements in this research. The definition of *MAE*, *RMSE*, and $R^2$ are as follows:

$$MAE = \frac{1}{n}\sum_1^n |r_i - p_i|, \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_1^n (r_i - p_i)^2}, \tag{20}$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (r_i - p_i)^2}{\sum_{i=1}^n (r_i - \overline{r}_i)}, \tag{21}$$

where $r$ = actual settlement, $p$ = the predicted settlement, $n$ = total number of events considered, $\overline{r}$ = average value of measured settlements. *MAE* reflects the average magnitude of error between predicted and measured value, while *RMSE* describes the standard deviation of differences between them. $R^2$ provides a measure of how well observed outcomes are replicated by the model.

# 4 Results

## 4.1 Parameters analysis

To provide a descriptive overview of the data distribution, the correlation matrix between ten input variables and one output variable is presented in Fig. 3. The scatter plot matrix in the upper panel illustrates the relationship between pairwise parameters. Each Pearson correlation coefficient $R$ filled in the low panel means the correlation between the variables in the column and corresponding row. The expression of Pearson correlation coefficient is shown in Eq. (22). It can be observed that all parameters have a relatively poor correlation ($R < 0.5$) with one another, and the data set is quiet widely distributed. Hence, scaling all input variables into [–1, 1] range based on their minimum and maximum values will greatly save computational cost [37]. The results also indicate that the ground
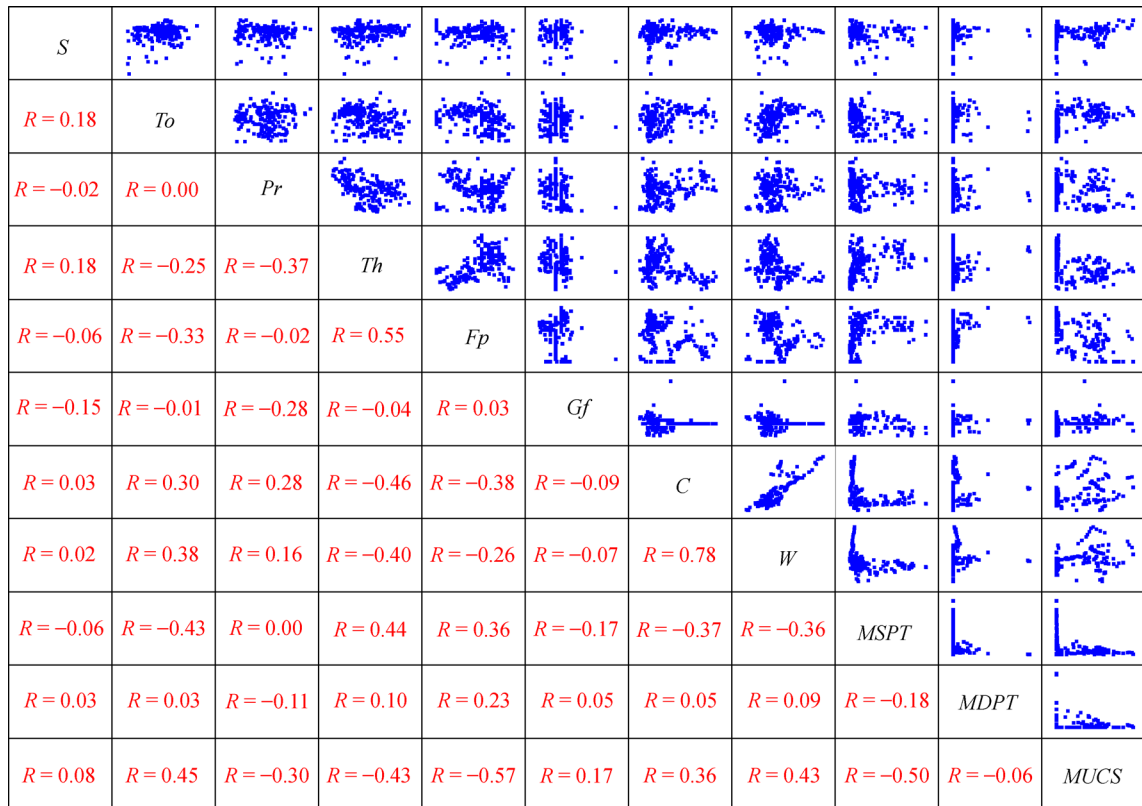


**Fig. 3** Correlation matrix and scatter plot of all variables.

settlement shows positive correlation with torque, thrust, tunnel depth, water table and the strength of gravel and rock, which is consistency with common recognition. However, some results violate the practical knowledge, like $R = -0.06$ and $-0.15$ for face pressure and grout filling with respect to settlement, respectively, which means that the increase in grout filling and face pressure leads to an decrease of the magnitude of ground settlement but the ground settlement increases actually. Overall, the ground settlement weakly correlates to the ten input variables, and a simple linear relationship between variables does not exist.

$$R = \frac{n\sum_{i=1}^{n}x_iy_i - \sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i}{\sqrt{n\sum_{i=1}^{n}x_i^2 - \left(\sum_{i=1}^{n}x_i\right)^2}\sqrt{n\sum_{i=1}^{n}y_i^2 - \left(\sum_{i=1}^{n}y_i\right)^2}}, \quad (22)$$

where $x_i$ = the value of input variable, $y_i$ = the value of output variable, $n$ = total number of events considered.

### 4.2 Multivariate linear regression

Figure 4 shows the relationship between the predicted settlement obtained from the multivariate linear regression (MLR) and the measure one. The maximum settlement is obtained by the following equation.

$$S = 1.065x_1 + 0.106x_2 + 0.399x_3 - 0.427x_4 - 0.530x_5$$

$$+ 0.137x_6 - 0.133x_7 - 0.008x_8 + 0.07x_9$$

$$+ 0.174x_{10} - 11.108, \quad (23)$$

where $[x_1, x_2,..., x_n]$ are the ten input parameters, as shown in Table 1. The values of $MAE$, $RMSE$ (calculated based on both training and test set) and the coefficient of determination of $R^2$ (calculated based on test set) are presented in the figure. The marginal histogram shows that the predicted value ranges from $-5.625$ to $-0.125$ mm, but the measured settlement ranges from $-12.5$ to $0.125$ mm. Further, the coefficient of determination $R^2$ is significantly low (0.09), and the absence of agreement between the measured and predicted values is consistently observed. It suggests linear model is not able to properly predict the tunneling-induced ground settlement. The settlement exhibits a nonlinear variation with respect to the shield operation factors, geological conditions and tunnel characteristics.

### 4.3 Model development and parameter optimization

A framework is developed in MATLAB to train all six different types of models examined in this paper, as shown in Fig. 5. In general, the completed process of establishing a ML model involves three phases: training, validation, and test. Several validation methods has been conducted including bootstrap, substitution method, and holdout method, etc. [54,55]. The most popular one is probably k-fold cross-validation (CV) method [56]. Based on this, K-CV method is used as validation method in this paper. K-CV is a technology of dividing the original data set randomly into K sub-data sets. K-1 sub-data sets are used as training set and a remaining data set is the test set. That is, each sample has opportunity to become the test set. The process repeats $K$ times, and the ultimate result is originated from the averaged value. The training set is divided into four subsets in this research. Herein, each ML
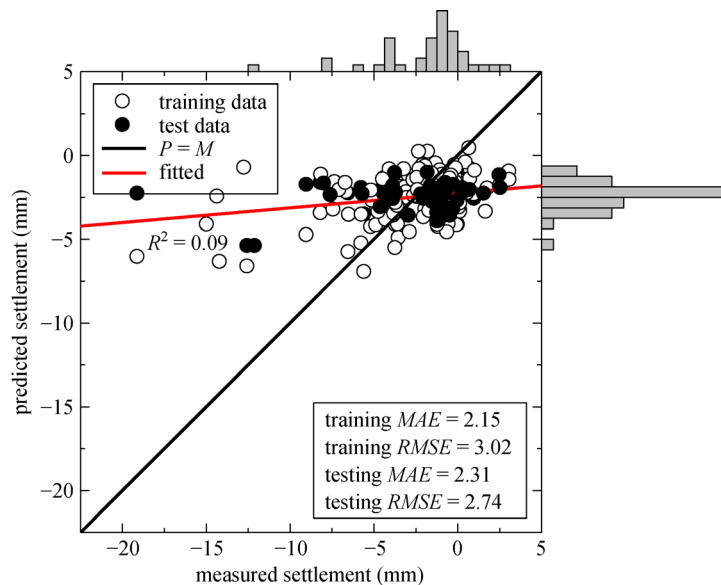


**Fig. 4**  Marginal histogram of settlements predicted by multivariate regression method, compared to measured values.
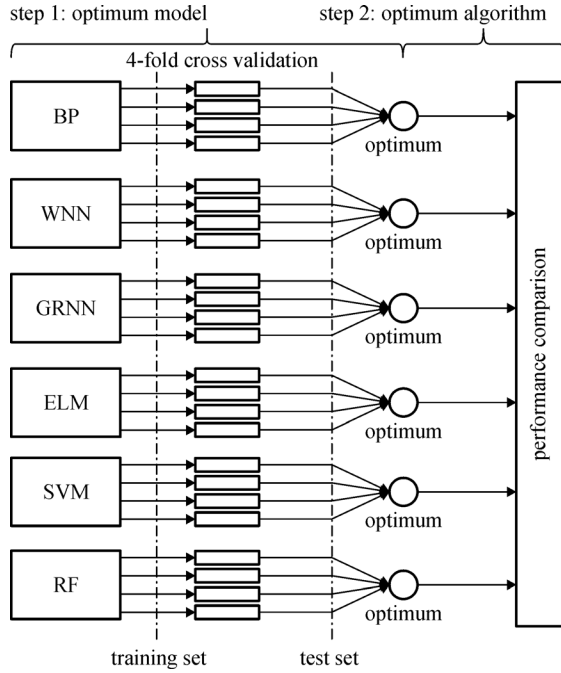
**Fig. 5**  Workflow of determining optimum ML algorithm.

algorithm with 100 sets of parameters or architectures are calculated in parallel. Three subsets are used to train model while the remaining subset is used to test model. After computing 100 times, four sets of *MAE* and *RMSE* values are obtained. The optimum parameters or architectures of ML algorithms are determined by the average *MAE* and *RMSE* values of four validation sets. The parameters or architecture with the lowest error are defined as the optimum model. Then the test set is used to test each model, thereby the most appropriate algorithm for this project can be determined by comparing the *MAE* and *RMSE* values of six algorithms.

To optimize the network, the number of neurons in hidden layer is tuned. The result presents the BPNN cannot be converged if the number of neurons in the hidden layer is less than 10 in this case. As shown in Fig. 6(a), the average *MAE* and *RMSE* values of 4-folds CV sets vary dramatically with the increase in the hidden neurons. The number of neurons of 53 yields the lowest *MAE* and *RMSE* values.

The variation of *MAE* with the increasing neurons in WNN is presented in Fig. 6(b). The *MAE* and *RMSE* values increase monotonically with the increase in the number of neurons. The magnitude of *MAE* and *RMSE* values is large, similar to the BPNN algorithm. The optimum number of neurons is here identified to 10.

Figure 6(c) illustrates the variation of *MAE* and *RMSE* values of GRNN models with different smoothing parameter $\sigma$. It can be observed that *MAE* and *RMSE* values hold steady when the $\sigma$ value is larger than 4. If $\sigma$ sets large, the generalization ability of network will increase and also degrade the error of prediction. However,

excessively high $\sigma$ can result in overfitting. Low smooth factor can reduce the network's generalization ability and may even prevent it from doing any prediction at all [57]. Therefore, the model with $\sigma = 2$ is regarded as the best option based on the trade-off between *MAE* and *RMSE*.

Figure 6(d) shows the variation of *MAE* and *RMSE* values in the ELM models. Similar to WNN algorithm, The *MAE* and *RMSE* value increases monotonically with an increase in neurons of hidden layer and the prediction error is large when the number of neurons exceeds 20. The optimum number of neurons of the hidden layer is equal to 3.

The main objective of K-CV method is to identify the optimum penalty parameter $c$ and kernel parameter $g$ in the SVM. Figure 6(e) shows the contour map of *RMSE* value with the variation of $c$ and $g$ values. It indicates the least value of *RMSE* is equal to 0.02 when the $c$ and $g$ values are equal to 5.6569 and 0.0625, respectively.

As shown in Fig. 6(f), for lower number of trees in RF algorithm, the *MAE* value decreases with an increase in the number of trees. However, the accuracy declines steadily as the number of trees exceeds 13 trees, which means that any further increase in the number of trees will result in overfitting. Then beyond 40 trees, the *MAE* and *RMSE* values saturate. Herein, the optimum number of trees is identified to be 13.

Table 3 presents the optimum average *MAE* value of six machine learning algorithms and the corresponding optimum parameters or architectures. The table also presents the computational cost of each method after complete 100 times. It can be seen the calculation speed of ELM and SVM methods is the fastest. Three ML algorithms WNN, GRNN, and RF have the same magnitude of calculation time. Undoubtedly, the computational costs of BP is highest with 165 s. The result is consistent with the descriptions of six ML methods in the Section 3.1. The six ML methods with these optimum parameters or architectures will be used to establish the final models and their performance will be evaluated by the test set.

### 4.4  Performance analysis

The predicted results based on six best models obtained from the Section 4.3 are plotted together, compared with measured settlement, as shown in Fig. 7. The *RMSE* and *MAE* values of training and test set are also calculated and the predictions of test set are fitted using linear regression. Three indicators can be employed to analyze the performance of each model.

The settlements predicted using the BPNN are plotted in Fig. 7(a) compared to the measured values. The *MAE* and *RMSE* values of the training set are relatively low, 0.21 and 0.33, respectively. However, the *RMSE* and *MAE* values of test set increase to 3.35 and 4.28, respectively, compared with the MLR method. The value of $R^2$ increases slightly to
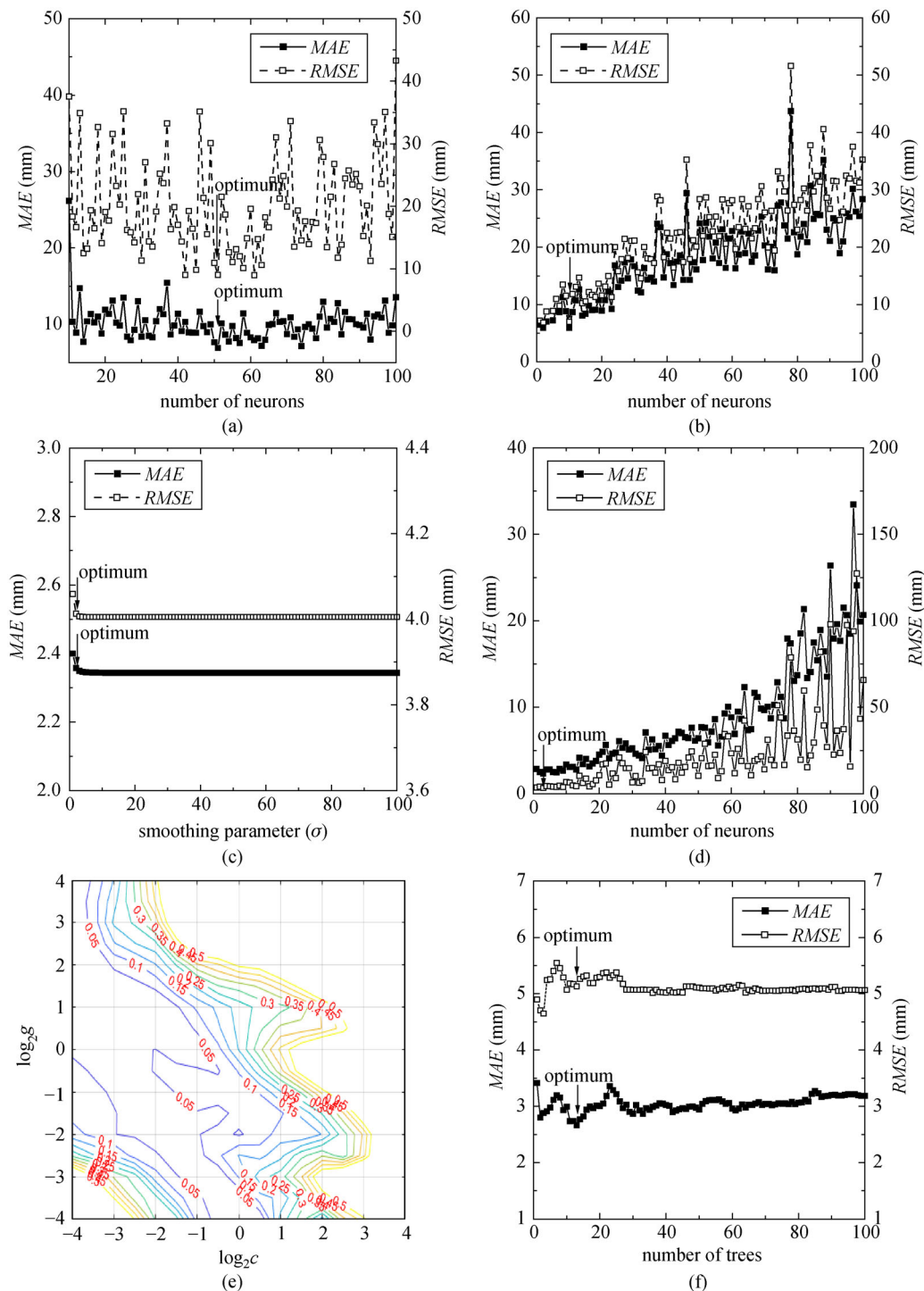
**Fig. 6**  Relation between the model performance and the parameter of test set. (a) BPNN; (b) WNN; (c) GRNN; (d) ELM; (e) SVM; (f) RF.

0.12 but it is still as low as the case of MLR method.

The predicted results of training and test set using the WNN are plotted in Fig. 7(b). The WNN shows better performance than the BPNN with *MAE* and *RMSE* values of test set decreasing to 2.18 and 3.53, respectively. The ranges of predicted settlement are nearly equal to measured settlements, but the $R^2$ is still relatively low, similar to that

of the BPNN.

Figure 7(c) illustrates the settlements predicted using the GRNN. The predicted settlements show an excellent match with settlements over a wide range of measured values. Results of training set fit the data perfectly and yield low error (*MAE* = 0.34, *RMSE* = 0.75). Moreover, the predictions of test set also yield relatively low error and

**Table 3**  Values of optimum architectures or parameters in six ML algorithms

| algorithm | optimum parameters | *MAE* (mm) | time (s) |
|---|---|---|---|
| BPNN | hidden_layer_number = 1<br>hidden_layer_neuron_num = 53 | 6.33 | 165 |
| WNN | hidden_layer_neuron_num = 10 | 5.90 | 32 |
| GRNN | $\sigma$_width index = 2 | 2.35 | 41 |
| ELM | hidden_layer_neuron_num = 3 | 2.33 | 5 |
| SVM | $c$_penalty = 5.6569<br>$g$_width index = 0.0625 | 1.49 | 9 |
| RF | ntree = 13 | 2.67 | 28 |

the majority of points closely lie in a straight line with slope of 1 ($P$ = M line). It is worth noting that the predicted settlement of test set deviates from the measured settlement at the maximum settlement point, which is attributed to a lack of training data in that range. Overall, the GRNN has better generalization ability, which makes it possible for the GRNN to capture the complicated nonlinear relationships between input and output variables.

The measured settlements and predictions using the ELM are plotted in Fig. 7(d). The main feature of the ELM is computational efficiency. Interestingly, the ELM yields a poor performance, similar to the case of the MLR method. Although the values of *MAE* and *RMSE* are acceptable, but the $R^2$ value is significantly small. The predicted settlement values only range from –6 to –4 mm in contrast to –12.5 to 0.125 mm in the practical project. So the predicted settlements using this ELM are discredited.

Settlements predicted using the SVM are plotted in Fig. 7(e), compared with measured settlements. The SVM gives a reasonably good prediction of the settlement over a wide range of measured values. The higher value of $R^2$ demonstrates that the predictions exhibit a linear correlation with respect to measured settlements. The predictions of SVM can be mapped to another space using this fitted linear regression function, which further improves the accuracy of the SVM model. However, the range of predicted values (−5.5 to 0 mm) deviates from the measured settlement values. The discrepancy is notably high for large settlement data under −4 mm.

The settlements predicted using RF are plotted in Fig. 7(f). Similar to the GRNN, the results of training set fit the data perfectly and yield low error with *MAE* = 0.05, *RMSE* = 0.59. The fitted line of the test set is also close to the $P$ = M line and the *MAE* and *RMSE* values of RF are 1.85 and 2.66, respectively.

The statistics performance of predicted results of all models is presented in Fig. 8, compared with measured settlements. Figure 8(a) shows the distribution of predicted settlements based on training set. It can be seen that the mean predicted settlement is roughly identical in different models. As for the distribution of predicted settlements, the

results of BPNN, GRNN, and RF models show great agreement with distribution of actual settlements. However, the other three models cannot accurately recognize the settlement distribution characteristics. The predicted results of MLR, ELM, and SVM only range from − 6 to 0 mm while the measured settlements of training set fall into the range of − 19 to 6 mm. Meanwhile the WNN fails to predict the range of 1th and 99th percentiles as well as the outliers.

Figure 8(b) shows the distribution of predicted settlements based on the test set. In general, these models with great performance on the training set also present great performance on test set. It can be observed that GRNN and RF models still achieve the best performance among these algorithms. Note that the range of predicted results of test set using BPNN remarkably exceed the actual range. The BPNN suffers from poor generalization performance owing to the failure of finding global minimum of error function as mentioned in Section 3.1.2. Although the predicted results of training set show great performance, the reliability of the trained model cannot be guaranteed. The predicted settlements using the other three algorithms have the same deficiencies as the training set. Poor generalization ability of these models leads to the narrow range of predicted settlements.

## 5  Discussion

A comprehensive comparison of results of the different ML algorithms demonstrates that the capability of a given model to predict the tunneling-induced settlements depends on the relationship between input and output variables. Linear regression method has been regard as an effective and time-saving way to capture the input-output relationship. In this study, the failure of multivariate linear regression approach to predict the settlements, high values of *MAE* and *RMSE* and low value of $R^2$, suggests that this is a highly nonlinear problem. To identify the response mechanism, ML algorithms are employed to address this nonlinear and multivariable problem. Table 4 compares the predicted results using MLR, BPNN, WNN, GRNN, ELM, SVM, and RF algorithms.

The evolution of predicted ground settlements using different ML algorithms is shown in Fig. 9, compared with measured settlements. RF and GRNN algorithms accurately capture the evolution of actual observations. The predicted value using the BPNN algorithms vary remarkably, resulting in losing fidelity at some points. While the results of WNN, SVM, and ELM are prone to identical, and the difference between the maximum and minimum predictions is relatively small. Figure 10 presents the corresponding prediction errors of six machine learning algorithms. The prediction errors are defined as:
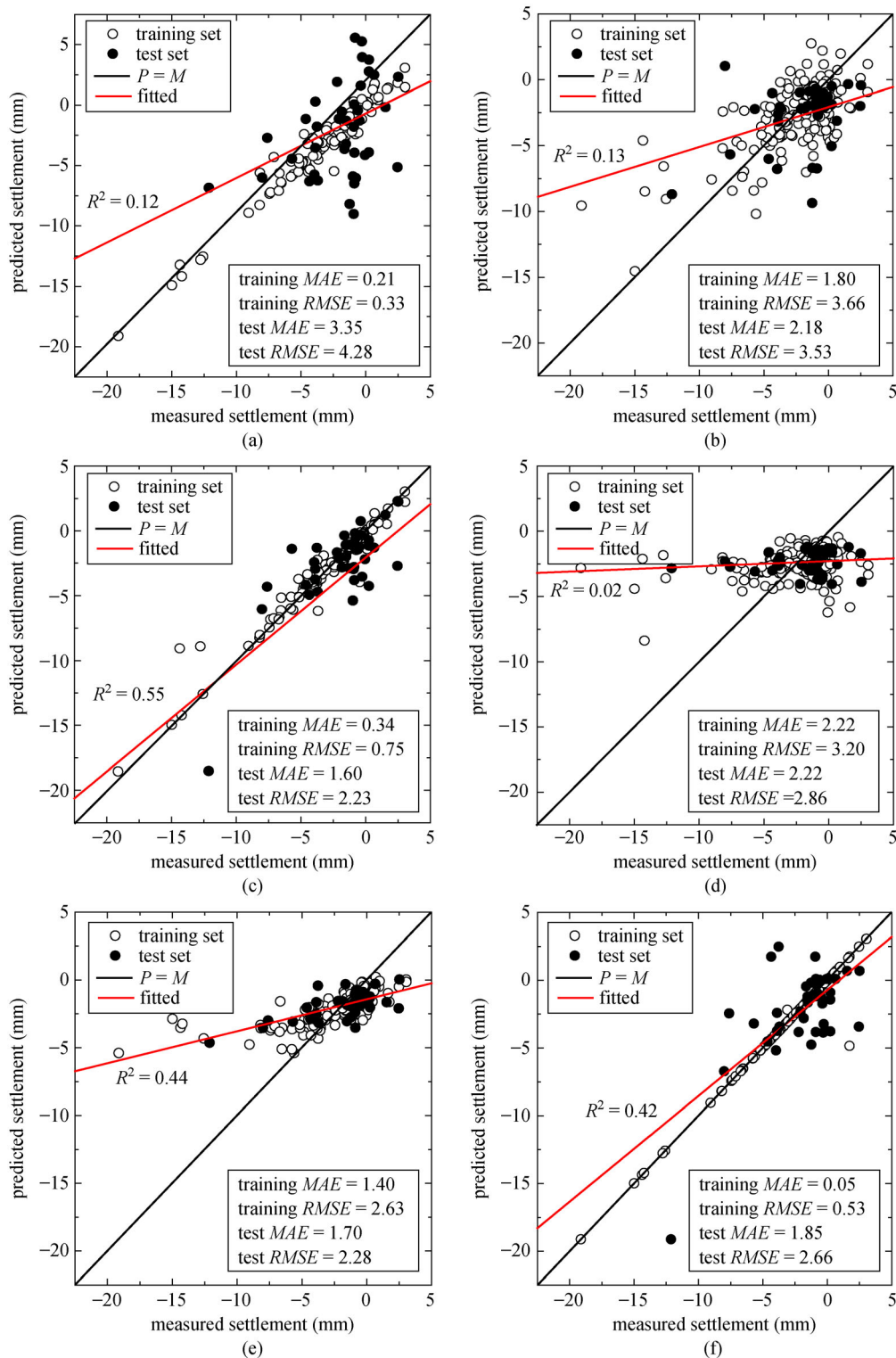
$$e = S_p - S_m, \tag{24}$$

**Fig. 7**   Predicted settlements using (a) BPNN, (b) WNN, (c) GRNN, (d) ELM, (e) SVM, (f) RF.

where $S_p$ = the predicted settlement; $S_m$ = the measured settlement. The negative value means that the predicted settlement is larger than the measured settlement while the positive settlement means that the predicted settlement is lower than the measured settlement. It can be observed that the predicted settlement is larger than measured settlement at most points, which indicates that the predicting tunneling-induced settlement using ML algorithms are safe. The prediction errors of BP, WNN, and ELM are much larger than the remaining three algorithms. The
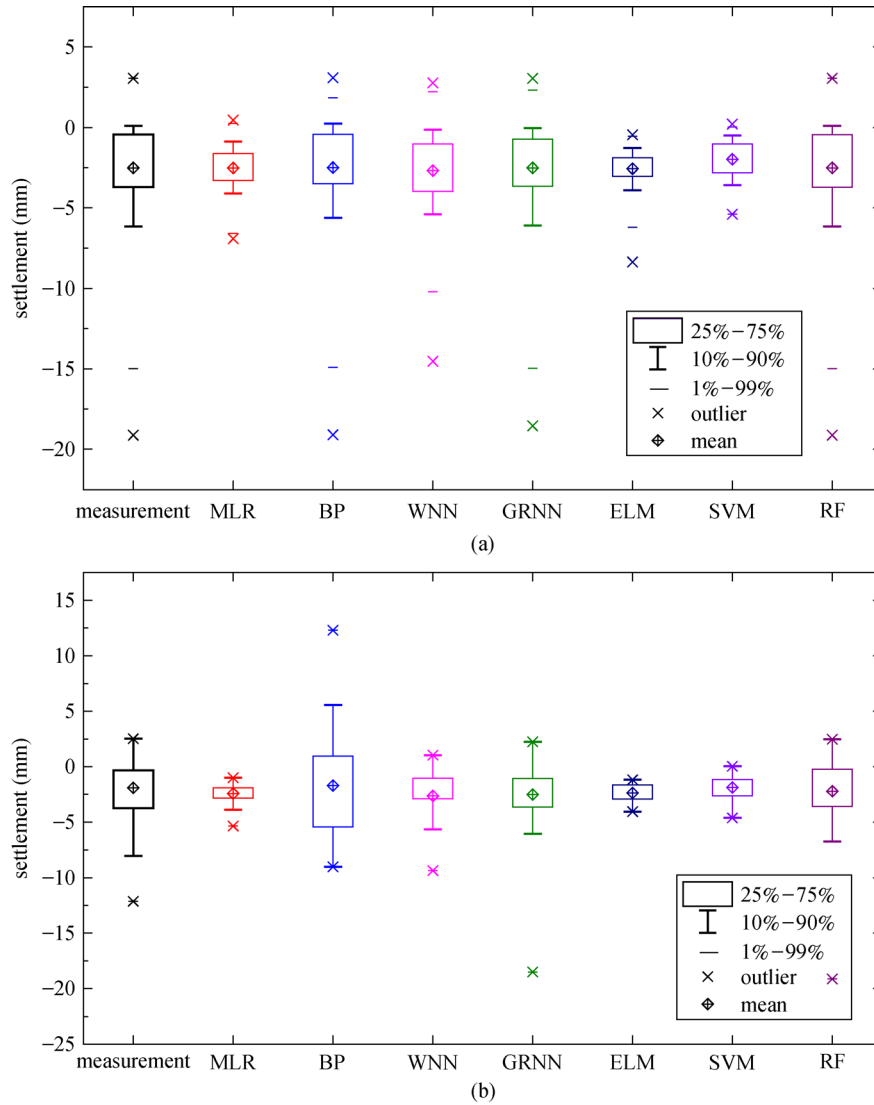
**Fig. 8**   Box plot of the settlements predicted by ML methods, compare to measured settlements (a) training set and (b) test set.

**Table 4**   Comparison of seven models for predicting settlements

| methods | training set | | test set | | $R^2$ |
|---|---|---|---|---|---|
| | *MAE* | *RMSE* | *MAE* | *RMSE* | |
| MLR | 2.15 | 3.02 | 2.31 | 2.71 | 0.09 |
| BPNN | 0.21 | 0.33 | 3.35 | 4.28 | 0.12 |
| WNN | 1.80 | 3.66 | 2.18 | 3.53 | 0.13 |
| GRNN | 0.34 | 0.75 | 1.60 | 2.23 | 0.55 |
| ELM | 2.22 | 3.20 | 2.22 | 2.86 | 0.02 |
| SVM | 1.40 | 2.63 | 1.70 | 2.28 | 0.44 |
| RF | 0.05 | 0.53 | 1.85 | 2.66 | 0.42 |



**Fig. 9**   Results of test set of different ML methods in predicting settlements.

maximum prediction error of ELM even reaches to about 15 mm while the maximum prediction errors of BP and WNN algorithms reaches to about 10 mm. By contrast, the
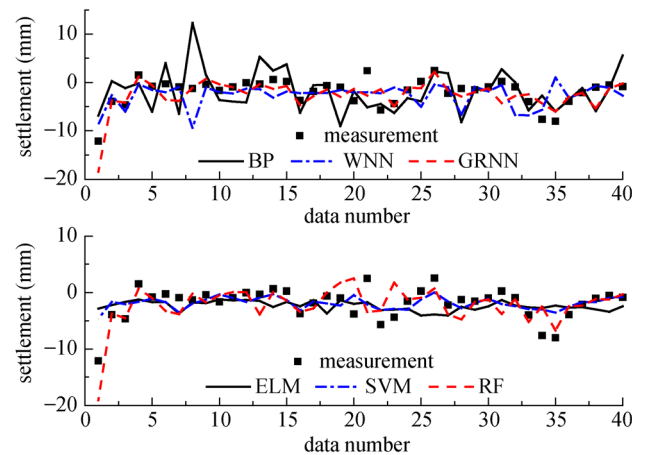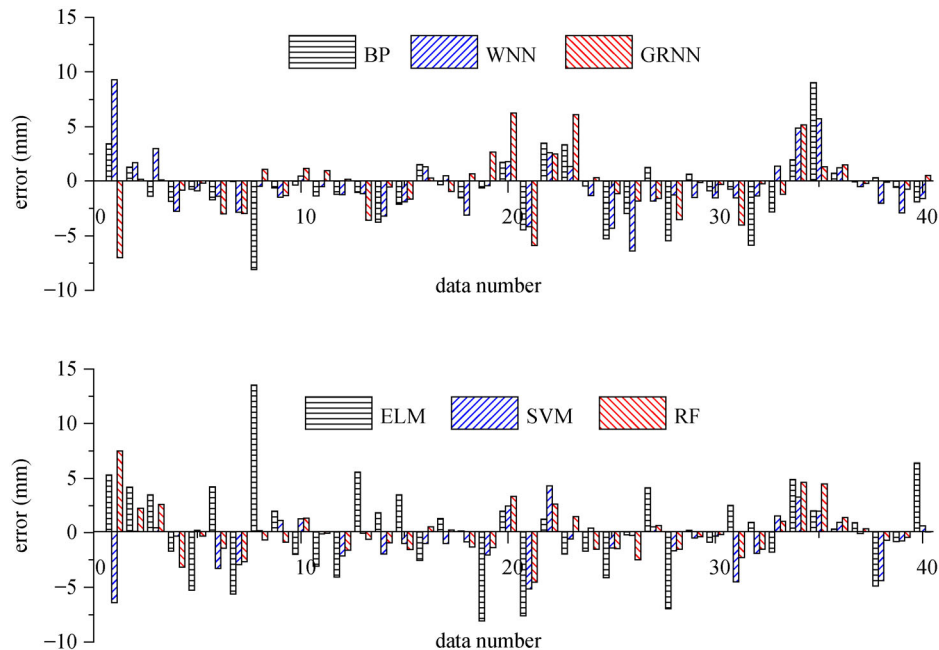
**Fig. 10** Prediction errors of test set of different ML methods in predicting settlements.

maximum prediction errors of GRNN, SVM, and RF algorithms are about 7.5 mm. Note worth that six ML algorithm are prone to yield large or small prediction errors at same monitoring points. However, the magnitude of prediction errors is different. It indicates the mechanism of six machine learning algorithms may be identical in predicting tunneling-induced settlement, but they show different performance. Overall, ML algorithms provide an efficient route toward the prediction of settlements caused by tunneling. GRNN and RF algorithms show the best performance, so these two algorithms are recommended to predict tunneling-induced settlement. The settlement prediction is a problem with complicated mechanism and multiple influencing factors, data-driven models built by ML algorithms offer a pragmatic and reliable option while waiting for the development of mechanism-based models.

## 6   Conclusions

Inspired by the significant development of ML algorithms, ANN, SVM, and RF algorithms has been used in predicting tunneling-induced ground settlement. However, a lack of a full-scale comparative studies impedes the use and popularity of these data-driven models. This paper investigated the performance of six ML algorithms in predicting the ground settlement, and the following conclusions can be drawn.

Pearson correlation coefficient is applied to investigate the correlation between the input and output variables. The ground settlement weakly correlates to the ten input variables and a simple linear relationship between

variables does not exist. Meanwhile the failure to predicting ground settlements using the multivariate linear regression also demonstrates that tunneling-induced settlement exhibits a nonlinear variation with respect to influential factors.

4-folds cross-validation method as the validation method is used to determine the optimum architecture or parameter of six machine learning algorithm. Each machine learning algorithm with 100 sets of parameters or architectures is computed. Through the analysis of the outcomes of different ML algorithms, it indicates that the variance of predicted results based on the test set using the BPNN is dramatic, losing fidelity at some monitoring points. Other three ML algorithms, WNN, ELM, and SVM fail to accurately predict the evolution of tunneling-induced settlement since the predicted settlement value is virtually identical. The predicted settlement using GRNN and RF algorithms shows great agreement with the measured settlement. Meanwhile the prediction error of BP, WNN, and ELM reaches to 15 mm while the maximum prediction errors of GRNN, SVM, and RF are 7.5 mm. Accordingly, two algorithms GRNN and RF are recommended as a useful solution for predicting tunneling-induced settlements, and the results can be as a reminder for field engineers.

It is worthy noted that different input variable will affect the results. This paper placed emphasis on the performance of six machine learning algorithms in predicting tunneling-induced settlement using ten fixed input variables. The influence of combination of different input variables on the performance of predicting settlement should be investigated in the further study.

# References

1. Chen R P, Lin X T, Kang X, Zhong Z Q, Liu Y, Zhang P, Wu H N. Deformation and stress characteristics of existing twin tunnels induced by close-distance EPBS under-crossing. Tunnelling and Underground Space Technology, 2018, 82: 468–481

2. Zhang P, Chen R P, Wu H N. Real-time analysis and regulation of EPB shield steering using random forest. Automation in Construction, 2019, 106: 102860

3. Wu H N, Shen S L, Yang J. Identification of tunnel settlement caused by land subsidence in soft deposit of Shanghai. Journal of Performance of Constructed Facilities, 2017, 31(6): 04017092

4. Wu H N, Shen S L, Liao S M, Yin Z Y. Longitudinal structural modelling of shield tunnels considering shearing dislocation between segmental rings. Tunnelling and Underground Space Technology, 2015, 50: 317–323

5. Wu H N, Shen S L, Yang J, Zhou A N. Soil-tunnel interaction modelling for shield tunnels considering shearing dislocation in longitudinal joints. Tunnelling and Underground Space Technology, 2018, 78: 168–177

6. Zhang Z, Huang M. Geotechnical influence on existing subway tunnels induced by multiline tunneling in Shanghai soft soil. Computers and Geotechnics, 2014, 56: 121–132

7. Shi H, Yang H, Gong G, Wang L. Determination of the cutterhead torque for EPB shield tunneling machine. Automation in Construction, 2011, 20(8): 1087–1095

8. Zheng G, Cui T, Cheng X, Diao Y, Zhang T, Sun J, Ge L. Study of the collapse mechanism of shield tunnels due to the failure of segments in sandy ground. Engineering Failure Analysis, 2017, 79: 464–490

9. Chen R P, Li Z C, Chen Y M, Ou C Y, Hu Q, Rao M. Failure investigation at a collapsed deep excavation in very sensitive organic soft clay. Journal of Performance of Constructed Facilities, 2015, 29(3): 04014078

10. Huang Q, Huang H, Ye B, Zhang D, Zhang F. Evaluation of train-induced settlement for metro tunnel in saturated clay based on an elastoplastic constitutive model. Underground Space, 2018, 3(2): 109–124

11. Shen S L, Wu H N, Cui Y J, Yin Z Y. Long-term settlement behaviour of metro tunnels in the soft deposits of Shanghai. Tunnelling and Underground Space Technology, 2014, 40: 309–323

12. Wang Z F, Cheng W C, Wang Y Q. Investigation into geohazards during urbanization process of Xi'an, China. Natural Hazards, 2018, 92(3): 1937–1953

13. Vorster T E, Klar A, Soga K, Mair R J. Estimating the effects of tunneling on existing pipelines. Journal of Geotechnical and Geoenvironmental Engineering, 2005, 131(11): 1399–1410

14. Attewell P B, Hurrell M R. Settlement development caused by tunnelling in soil. Ground Engineering, 1985, 18(8): 17–20

15. Peck R B. Deep excavations and tunneling in soft ground. In: Proceedings of the 7th International Conference on Soil Mechanic and Foundation Engineering. Mexico City, 1969, 225–290

16. Sagaseta C. Analysis of undrained soil deformation due to ground loss. Geotechnique, 1987, 37(3): 301–320

17. Verruijt A, Booker J R. Surface settlements due to deformation of a tunnel in an elastic half plane. Geotechnique, 1998, 48(5): 709–713

18. Chen R P, Meng F Y, Li Z C, Ye Y H, Ye J N. Investigation of response of metro tunnels due to adjacent large excavation and protective measures in soft soils. Tunnelling and Underground Space Technology, 2016, 58: 224–235

19. Huang H, Gong W, Khoshnevisan S, Juang C H, Zhang D, Wang L. Simplified procedure for finite element analysis of the longitudinal performance of shield tunnels considering spatial soil variability in longitudinal direction. Computers and Geotechnics, 2015, 64: 132–145

20. Paternesi A, Schweiger H F, Scarpelli G. Numerical analyses of stability and deformation behavior of reinforced and unreinforced tunnel faces. Computers and Geotechnics, 2017, 88: 256–266

21. Zhang L, Wu X, Ji W, AbouRizk S M. Intelligent approach to estimation of tunnel-induced ground settlement using wavelet packet and support vector machines. Journal of Computing in Civil Engineering, 2017, 31(2): 04016053

22. Pourtaghi A, Lotfollahi-Yaghin M A. Wavenet ability assessment in comparison to ANN for predicting the maximum surface settlement caused by tunneling. Tunnelling and Underground Space Technology, 2012, 28: 257–271

23. Zhang L, Wu X, Zhu H, AbouRizk S M. Performing global uncertainty and sensitivity analysis from given data in tunnel construction. Journal of Computing in Civil Engineering, 2017, 31(6): 04017065

24. Chou J S, Lin C. Predicting disputes in public-private partnership projects: Classification and ensemble models. Journal of Computing in Civil Engineering, 2013, 27(1): 51–60

25. Zoveidavianpoor M. A comparative study of artificial neural network and adaptive neurofuzzy inference system for prediction of compressional wave velocity. Neural Computing & Applications, 2014, 25(5): 1169–1176

26. Hamdia K M, Lahmer T, Nguyen-Thoi T, Rabczuk T. Predicting the fracture toughness of PNCs: A stochastic approach based on ANN and ANFIS. Computational Materials Science, 2015, 102: 304–313

27. Dai H, Cao Z. A wavelet support vector machine-based neural network metamodel for structural reliability assessment. Computer-Aided Civil and Infrastructure Engineering, 2017, 32(4): 344–357

28. Bouayad D, Emeriault F. Modeling the relationship between ground surface settlements induced by shield tunneling and the operational and geological parameters based on the hybrid PCA/ANFIS method. Tunnelling and Underground Space Technology, 2017, 68: 142–152

29. Kim C Y, Bae G J, Hong S W, Park C H, Moon H K, Shin H S. Neural network based prediction of ground surface settlements due to tunnelling. Computers and Geotechnics, 2001, 28(6–7): 517–547

30. Santos O J Jr, Celestino T B. Artificial neural networks analysis of São Paulo subway tunnel settlement data. Tunnelling and Underground Space Technology, 2008, 23(5): 481–491

31. Shi J, Ortigao J A R, Bai J. Modular neural networks for predicting

settlements during tunneling. Journal of Geotechnical and Geoenvironmental Engineering, 1998, 124(5): 389–395

32. Suwansawat S, Einstein H H. Artificial neural networks for predicting the maximum surface settlement caused by EPB shield tunneling. Tunnelling and Underground Space Technology, 2006, 21(2): 133–150

33. Hasanipanah M, Noorian-Bidgoli M, Jahed Armaghani D, Khamesi H. Feasibility of PSO-ANN model for predicting surface settlement caused by tunneling. Engineering with Computers, 2016, 32(4): 705–715

34. Meschke G. From advance exploration to real time steering of TBMs: A review on pertinent research in the Collaborative Research Center "Interaction Modeling in Mechanized Tunneling". Underground Space, 2018, 3(1): 1–20

35. Ding L, Wang F, Luo H, Yu M, Wu X. Feedforward analysis for shield-ground system. Journal of Computing in Civil Engineering, 2013, 27(3): 231–242

36. Samui P, Sitharam T G. Least-square support vector machine applied to settlement of shallow foundations on cohesionless soils. International Journal for Numerical and Analytical Methods in Geomechanics, 2008, 32(17): 2033–2043

37. Qi C, Tang X. Slope stability prediction using integrated metaheuristic and machine learning approaches: A comparative study. Computers & Industrial Engineering, 2018, 118: 112–122

38. Liu W, Wu X, Zhang L, Wang Y, Teng J. Sensitivity analysis of structural health risk in operational tunnels. Automation in Construction, 2018, 94: 135–153

39. Behzad M, Asghari K, Coppola E A Jr. Comparative study of SVMs and ANNs in aquifer water level prediction. Journal of Computing in Civil Engineering, 2010, 24(5): 408–413

40. Kohestani V R, Bazargan-Lari M R, Asgari-marnani J. Prediction of maximum surface settlement caused by earth pressure balance shield tunneling using random forest. Journal of Artificial Intelligence and Data Mining, 2017, 5(1): 127–135

41. Sun W, Shi M, Zhang C, Zhao J, Song X. Dynamic load prediction of tunnel boring machine (TBM) based on heterogeneous *in-situ* data. Automation in Construction, 2018, 92: 23–34

42. Bendu H, Deepak B B V L, Murugan S. Application of GRNN for the prediction of performance and exhaust emissions in HCCI engine using ethanol. Energy Conversion and Management, 2016, 122: 165–173

43. Anoop Krishnan N M, Mangalathu S, Smedskjaer M M, Tandia A, Burton H, Bauchy M. Predicting the dissolution kinetics of silicate glasses using machine learning. Journal of Non-Crystalline Solids, 2018, 487: 37–45

44. Mangalathu S, Jeon J S. Classification of failure mode and prediction of shear strength for reinforced concrete beam-column joints using machine learning techniques. Engineering Structures, 2018, 160: 85–94

45. Zhou J, Li X, Mitri H S. Classification of rockburst in underground projects: Comparison of ten supervised learning methods. Journal of Computing in Civil Engineering, 2016, 30(5): 04016003

46. Dwivedi A K. Performance evaluation of different machine learning techniques for prediction of heart disease. Neural Computing & Applications, 2018, 29(10): 685–693

47. Naghibi S A, Ahmadi K, Daneshi A. Application of support vector machine, random forest, and genetic algorithm optimized random forest models in groundwater potential mapping. Water Resources Management, 2017, 31(9): 2761–2775

48. Zhao Y, Pan H, Wang H, Yu H. Dynamics research on grouping characteristics of a shield tunneling machine's thrust system. Automation in Construction, 2017, 76: 97–107

49. Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: Theory and applications. Neurocomputing, 2006, 70(1–3): 489–501

50. Breiman L. Bagging Predictors. Machine Learning, 1996, 24(2): 123–140

51. Ho T K. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 832–844

52. Liaw A, Wiener M, Liaw A. Classification and regression by random forest. R News, 2002, 23(23): 18–21

53. Tseranidis S, Brown N C, Mueller C T. Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures. Automation in Construction, 2016, 72: 279–293

54. Braga-Neto U, Hashimoto R, Dougherty E R, Nguyen D V, Carroll R J. Is cross-validation better than resubstitution for ranking genes? Bioinformatics (Oxford, England), 2004, 20(2): 253–258

55. Badawy M F, Msekh M A, Hamdia K M, Steiner M K, Lahmer T, Rabczuk T. Hybrid nonlinear surrogate models for fracture behavior of polymeric nanocomposites. Probabilistic Engineering Mechanics, 2017, 50: 64–75

56. Stone M. Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society. Series A (General), 1974, 36(2): 111–147

57. Theodosiou M. Disaggregation & aggregation of time series components: A hybrid forecasting approach using generalized regression neural networks and the theta method. Neurocomputing, 2011, 74(6): 896–905