



# Modelling the mechanical behaviour of soils using machine learning algorithms with explicit formulations

Pin Zhang<sup>1</sup> · Zhen-Yu Yin<sup>1</sup> · Yin-Fu Jin<sup>1</sup> · Xian-Feng Liu<sup>2</sup>

Received: 21 January 2020 / Accepted: 18 February 2021 / Published online: 14 July 2021  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

## Abstract

This study systematically presents the application of machine learning (ML) algorithms for constructing a constitutive model for soils. A genetic algorithm is integrated with ML algorithms to determine the global optimum model, and the  $k$ -fold cross-validation method is used to enhance the models' robustness. Three typical ML algorithms with formulations explicitly expressed [i.e., back-propagation neural network (BPNN), extreme learning machine (ELM) and evolutionary polynomial regression (EPR)], and two modelling strategies (i.e. total or incremental stress–strain strategies) are used. A synthetic database is first generated based on a simple constitutive model to objectively evaluate the performance of three ML algorithms and two modelling strategies. Next, the optimum ML algorithm and the well evaluated modelling strategy are applied to experimental tests for examining its robustness. All results indicate that a BPNN-based constitutive model using the incremental stress–strain strategy performs best in modelling the mechanical behaviour of soils in terms of interpolation and extrapolation abilities, followed by ELM and then EPR.

**Keywords** Constitutive model · Extreme learning machine · Evolutionary computation · Neural network · Optimization · Soils

## 1 Introduction

Experimental investigations show that the mechanical behaviour of soils is very complicated, involving elements such as state-dependence [56], contraction-dilation [57], anisotropy [72], destructuration [41, 74], stress-path dependence [21], time-dependence [75], and non-coaxiality [59]. Accurate description of such soil behaviours is vitally important in engineering practice [33, 46, 66, 89]. Numerous constitutive models have been developed during the past few decades. These models can be classified as (1)

linear-elastic, (2) elastic perfectly plastic (such as the Mohr–Coulomb model), (3) nonlinear (such as the hardening soil [62] and nonlinear Mohr–Coulomb [28] models), (4) critical state-based advanced (such as the modified cam-clay model [53], Nor-Sand model [25], CSAM model [82], Severn–Trent model [11], UH models [68–70], SANISAND model [58], SIMSAND model [26–28] and ANICREEP model [80]), hypoplasticity [36, 42, 64, 65] and (5) micromechanical models [4, 67, 76–79]. The last two categories are usually called advanced soil models [28, 80]. However, traditional soil models have three main disadvantages in modelling soil behaviours: (1) most constitutive models were developed based on certain assumptions [71, 72, 75] (e.g., the associated or non-associated flow rule, non-coaxiality), (2) each model was suitable only for a specific type of soil or specific stress-paths and (3) although the mathematical formulas in a constitutive model are developed based on some theories (e.g., elastoplasticity theory) or derived from finite experimental data (e.g., the critical state line from triaxial tests), the formula's form gives good accuracy for selected tests, but at the same time limits the model's simulation ability

✉ Zhen-Yu Yin  
zhenyu.yin@polyu.edu.hk; zhenyu.yin@gmail.com

✉ Yin-Fu Jin  
yinfu.jin@polyu.edu.hk

<sup>1</sup> Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong, Kowloon, China

<sup>2</sup> Key Laboratory of High-Speed Railway Engineering of Ministry of Education, School of Civil Engineering, Southwest Jiaotong University, Chengdu 610031, China

**Table 1** Previous research works for identifying constitutive models of geomaterials\*\*

Strategy	ML algorithm	Validation method	Methods to determine architecture	References
Total stress–strain	BPNN	No	Trial & error	He and Li [19]
	BPNN	No	Trial & error	Rashidian and Hassanlourad [51]
	BPNN	No	Trial & error	Kohestani [38]
	SVM	No	Trial & error	Kohestani [38]
	GP	No	Optimization	Cabalar and Cevik [3]
Incremental stress–strain	BPNN	No	Trial & error	Ellis et al. [7]
	BPNN	No	Trial & error	Penumadu and Zhao [49]
	BPNN	No	Optimization	Basheer [2]
	BPNN	No	Trial & error	Habibagahi and Bamdad [16]
	BPNN	No	Trial & error	Turk et al. [61]
	ENN	No	Trial & error	Johari et al. [32]
	RNN	No	Trial & error	Zhu et al. [90]
	RNN	No	Optimization	Romo et al. [52]
	EPR	No	Optimization	Javadi and Rezania [24]
	EPR	No	Optimization	Faramarzi et al. [8]
EPR	No	Optimization	Nassr et al. [45]	

for other stress paths. For example, the Modified Cam-Clay (MCC) was derived from the triaxial tests of saturated remoulded clay, and thus the MCC model is difficult to predict other kind of tests or other soils. In addition, the mathematical formulas become increasingly complicated when involving many parameters, resulting in difficulties of parameter identification and further limiting their engineering applications.

Soil normally exhibits highly nonlinear characteristics. To simulate such characteristics, machine learning (ML) algorithms are very powerful and can thus be employed as an alternative way to construct data-driven constitutive models [88]. ML algorithms have three following advantages in developing soil models [86]: (1) ML algorithms can directly extract the stress–strain relationship from the experimental data without making any assumptions [9, 10, 12]. More stable and accurate results can be obtained by ML-based models if the physical mechanism is implied in training data and/or incorporated into the training process; (2) ML algorithms have a strong ability to capture complicated non-linear relationships [1, 5, 6, 17] and (3) the prediction accuracy of ML-based models can rise with the increasing datasets [83, 87]. Numerous ML-based soil models have already been developed, and they can be categorized according to the model's training strategy, whether (1) training models using the total values of stress and strain or (2) training models in incremental form [38]. However, up to now there is no comparative study to discuss which one is more suitable to develop ML

based model for describing soil behaviours. Accordingly, the performance of two stress–strain strategies in developing ML-based constitutive models deserves investigation.

To construct a ML-based soil model, myriad ML algorithms can be adopted, such as a back-propagation neural network (BPNN) [2, 16, 19, 49, 51, 61], evolutionary neural network (ENN) [32], recurrent neural network (RNN) [52, 90], support vector machines (SVMs) [35], evolutionary polynomial regression (EPR) [8, 24, 45] and genetic programming (GP) [3]. To find an ML algorithm that efficiently models soils' stress–strain relationship, a comparison of performance of different ML algorithms is demanding. Furthermore, the performance of an ML-based constitutive model is usually evaluated by the testing data within the range of the training data (interpolation ability), but this strategy neglects its performance on the unseen data (extrapolation ability).

This study aims to comprehensively demonstrate the process of constructing a ML-based constitutive model. To this end, three representative ML algorithms that can give explicit expression—BPNN, extreme learning machine (ELM) and EPR—were selected. The *k*-fold cross-validation method was employed in the validation phase to enhance the robustness of ML-based constitutive models. A genetic algorithm (GA) was used to optimize parameters for developing the global optimum model. A synthetic database based on a simple shear soil constitutive model was first built, which focuses on revealing the real

capabilities of BPNN, ELM and EPR to model soil behaviours, including interpolation and extrapolation abilities and the effects of the total and incremental stress–strain strategies. Thereafter, the optimum ML algorithm and modelling strategy were further applied to the experimental tests for examining its robustness.

## 2 Methodology of machine learning

### 2.1 Back-propagation neural network

In this study, the BPNN denotes a feedforward neural network characterized by propagation of errors from the output layer to find a set of weights and biases able to ensure that the output of the network is identical to the actual value [54]. A BPNN includes an input layer, any number of hidden layers and an output layer, which also determine its performance. Based on a given framework, the purpose of other hyper-parameters such as activation function is to further improve the training efficiency or optimize the model. Considering that this study focuses on simulating mechanical behaviours of soils, the deep investigation regarding the effect of each hyper-parameter on the model performance is not conducted. Herein, the optimum framework of the BPNN-based model is carefully investigated, whereas remaining hyper-parameters are set as the default value in Matlab toolbox. Once the hyper-parameters are determined, weighting and bias values can be calculated by gradient descent or optimization algorithms. Figure 1a illustrates a typical BPNN with one hidden layer. Taking the numbers of inputs and hidden and output neurons to be  $r$ ,  $p$  and  $q$ , respectively, and assuming that there are  $n$  datasets in the training set, the output of the hidden and output layers can be expressed as

$$\mathbf{H} = f(\mathbf{W}\mathbf{X} + \boldsymbol{\theta}) \quad (1)$$

$$\mathbf{O} = g(\mathbf{V}\mathbf{H} + \boldsymbol{\theta}_o) \quad (2)$$

where  $\mathbf{X}$  = matrix of input variables ( $r \times n$ );  $\mathbf{H}$  = matrix of the hidden layer output ( $p \times n$ );  $\mathbf{O}$  = matrix of output variables ( $q \times n$ );  $\mathbf{W}$ ,  $\mathbf{V}$  = weights matrix on the connections between input and hidden neurons ( $p \times r$ ) and between hidden and output neurons ( $q \times p$ ), respectively;  $\boldsymbol{\theta}$ ,  $\boldsymbol{\theta}_o$  = bias vectors on the connections between input and hidden neurons ( $p \times 1$ ) and between hidden and output neurons ( $q \times 1$ ), respectively; and  $f$ ,  $g$  = activation functions in hidden and output layers, respectively.

### 2.2 Extreme learning machine

The ELM is a type of feedforward neural network characterized by a single hidden layer (see Fig. 1b). The hyper-

parameters in the ELM are equal to the number of hidden neurons. The weights of the input layer and the biases of the hidden layer are assigned randomly, and the weights of the hidden layer ( $\boldsymbol{\beta}$ ) are determined analytically through a simple generalized inverse operation of the hidden layer output matrix [22], as shown in Eqs. (3)–(4), making the ELM's learning speed thousands of times faster than seen in traditional feedforward networks:

$$\mathbf{H} = f(\mathbf{W}\mathbf{X} + \boldsymbol{\theta}) \quad (3)$$

$$\min_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{O}\| \quad (4)$$

where  $\mathbf{X}$  = matrix of input variables ( $r \times n$ ),  $\mathbf{H}$  = matrix of the hidden layer output ( $p \times n$ ),  $\mathbf{O}$  = matrix of output variables ( $q \times n$ ),  $\mathbf{W}$  = weights matrix connecting input and hidden neurons ( $p \times r$ ),  $\boldsymbol{\theta}$  = the bias vector connecting input and hidden neurons ( $p \times 1$ ),  $\boldsymbol{\beta}$  = the weight matrix connecting the hidden and the output layers ( $q \times p$ ) and  $f$  = the activation function in the hidden layer.

### 2.3 Evolutionary polynomial regression

EPR is a genetic programming method characterized by the modelling of a system using a mathematical expression in the form of polynomial structures. Constructing an EPR-based model consists of two phases: (1) structure identification and (2) parameter estimation [14]. During the first phase, optimization algorithms are used to search for symbolic structures—that is, to determine the exponent matrix. At the second phase, the parameters' values are estimated by solving a least squares (LS) linear problem. Compared with BPNN and ELM, the training set in the EPR does not require normalization. A typical EPR expression can be formulated as

$$\mathbf{y} = \sum_{j=1}^m F[\mathbf{X}, f_j(\mathbf{X}), a_j] + a_0 \quad (5)$$

where  $\mathbf{y}$  = predicted output,  $\mathbf{X}$  = matrix of input variables,  $F$  = a function constructed by the process,  $f_j(\mathbf{X}) = j$ th transformed variable,  $a_j$  = an adjustable parameter for the  $j$ th term and  $a_0$  = an optional bias.  $f_j(\mathbf{X})$  is determined by

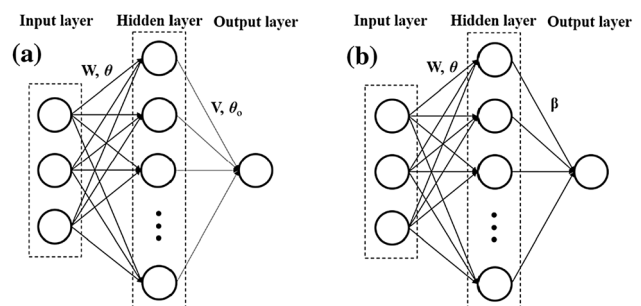


Fig. 1 Schematic view of ML algorithms: (a) BPNN; (b) ELM

the optimization algorithm, and  $a_j$  and  $a_0$  are determined by the LS.

The EPR's key objective is to identify the number of transformed variables and a combination of vectors of independent input variables. Herein, the transformed variable is obtained via

$$f_j(\mathbf{X}) = x_1^{\mathbf{ES}(j,1)} \cdot \mathbf{K} \cdot x_i^{\mathbf{ES}(j,i)} \cdot \mathbf{K} \cdot x_k^{\mathbf{ES}(j,k)} \quad (6)$$

where  $x_i$  =  $i$ th input variable,  $k$  = a total number of input variables and  $\mathbf{ES}_{m \times k}$  = exponent matrix.

## 2.4 Genetic algorithm

GA is a meta-heuristic optimization algorithm inspired by natural evolution [20]. It has been extensively employed in geotechnical engineering for tasks such as identification of constitutive models' parameters [26, 28, 73, 81], model selection [27], slope [39, 60], embankment [15, 44], tunnelling [37, 40], pile foundation [29, 31] and excavation [30]. In this study, the GA was selected to optimize initial weights and biases in BPNN and ELM algorithms and to search for symbolic structures in EPR. In GA, a population of individuals is first generated. A chromosome based on a coding scheme (real-coded GA) is then employed to represent each individual. After calculating the loss value of each individual, the best individual having the lowest loss value in the population is selected and then evolves through crossover and mutation operations to generate a new population. The process continues until it satisfies the termination criterion, that is, whether it reaches the maximum generation. Meanwhile, the loss value converges at a constant value.

## 2.5 K-fold cross-validation

Three phases are involved in the integrated process of constructing a ML model: training, validation and testing. The validation phase seeks to improve the robustness of the model and avoid overfitting. The  $k$ -fold cross validation can detect whether the overfitting issue exists after the training of model is completed. Moreover, the cross validation can also prevent the overfitting by integrating such method into the training process as the loss function [50] which is a commonly used method in the data-mining field. Currently, the  $k$ -fold cross-validation (CV) method is widely used to validate models [55]. In this method, the original training set is randomly divided into  $k$  sub-datasets. Herein,  $k-1$  sub-datasets, which form a new sub-training set, are employed to train models, and the performance of the trained model is validated by the remaining sub-dataset. Each sample in the training set thus has an opportunity to train and validate models.  $k$  is

generally set as 10 [34], thereby 10-fold CV method was used in this study.

At each round, the ML model with a fixed set of hyper-parameters was trained ten times based on nine sub-training sets, thereafter the performance of this ML model was evaluated by the mean squared error (MSE) for the remaining sub-dataset. Therefore, the loss function in the GA can be expressed as

$$\text{MSE} = \frac{\sum_{i=1}^m (\bar{y}_i - y_i)^2}{km} \quad (7)$$

where  $\bar{y}_i$  = predicted output,  $y_i$  = actual output,  $m$  = the number of datasets in the remaining sub-dataset and  $k$  = the number of CV sets.

## 2.6 Evaluation indicators

Two commonly used evaluation indicators—mean absolute error (MAE) and mean absolute percentage error (MAPE)—were used to evaluate the performance of ML models in this study. The combination of MAE and MAPE helps overcome the deficiencies of both, so that both are used extensively to evaluate model performance [5, 84, 85]. Low values of these two indicators indicate that a model has excellent performance. The expressions of MAE and MAPE can be obtained by

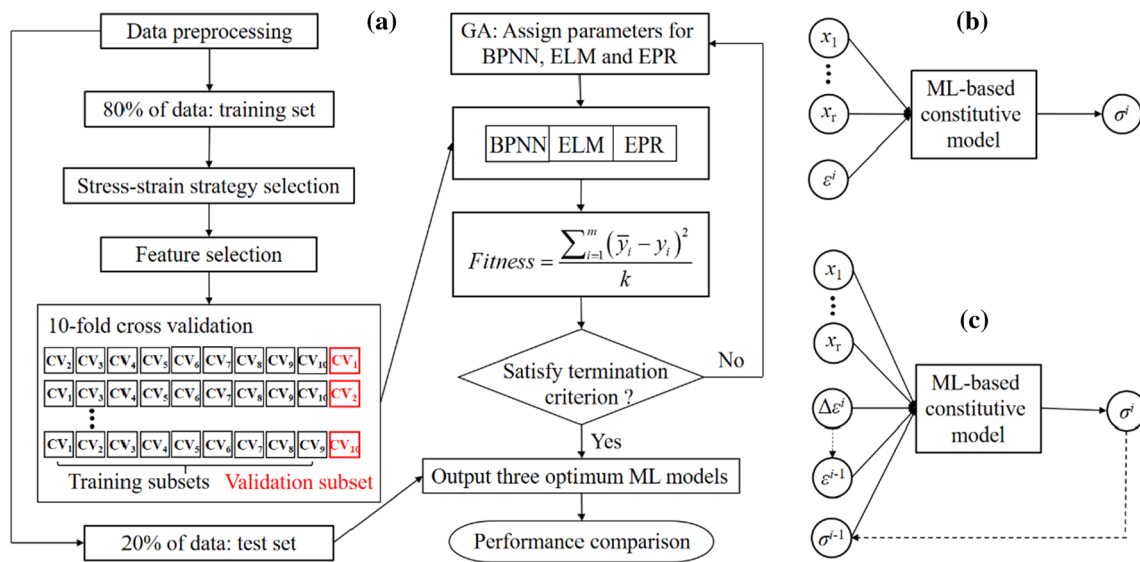
$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |r_i - p_i| \quad (8)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{r_i - p_i}{r_i} \right| \times 100\% \quad (9)$$

where  $r$  = actual output value,  $p$  = predicted output value and  $n$  = the total number of datasets.

## 2.7 Model framework

Figure 2a presents the flowchart for constructing a ML-based constitutive model. This type of data-driven model starts from the collection of datasets with which to form a database. In the ML domain, 80% (for training the model) and 20% (for testing the model) is a widely acknowledged scheme for data split ratio in the community. Such separation ratio can ensure the ML-based model being well trained and tested, which has been theoretically proved [13]. Therefore, 80% are used to train the model and 20% are used to test it in this study. The total or incremental stress–strain strategy is selected beforehand; thereafter, the corresponding features or input variables can be determined. At the next step, the 10-fold cross-validation method is used to divide the training set into ten subsets for training and validating models. At each round, GA is employed to identify the general parameters of ML



**Fig. 2** Model framework: (a) flowchart of constructing ML-based constitutive models; (b) schematic view of the total stress–strain strategy; (c) schematic view of the incremental stress–strain strategy

algorithms. The hyper-parameters are determined by the trial-and-error method. After determining three optimal constitutive models based on BPNN, ELM and EPR, their performance is compared using the test set.

Figures 2b, c illustrate the schematic view of the total and incremental stress–strain strategy, respectively. In the total stress–strain strategy, the stress in the  $i$ th step is affected by the strain at the  $i$ th step and the physical parameters (see Eq. (10)). In the incremental stress–strain strategy, the stress at the  $i$ th step is affected by the strain, the stress at the  $(i-1)$ th step, the strain increment at the  $i$ th step and the physical parameters (see Eq. (11)):

$$\sigma^i = f(\mathbf{X}, \varepsilon^i) \tag{10}$$

$$\sigma^i = f(\mathbf{X}, \sigma^{i-1}, \varepsilon^{i-1}, \Delta\varepsilon^i) \tag{11}$$

where  $\mathbf{X} = [x_1, x_2, \dots, x_r]$ , the vector of independent variables;  $\sigma^i, \sigma^{i-1}$  = stress at the  $i$ th and  $(i-1)$ th steps;  $\varepsilon^i, \varepsilon^{i-1}$  = strain at the  $i$ th and  $(i-1)$ th steps;  $\Delta\varepsilon^i$  = axial strain increment at the  $i$ th step; and  $f$  = formulation of stress–strain relationship, as determined by the ML algorithms in this study.

It should be noted that in the incremental stress–strain strategy, the predicted stress at the  $i$ th step needs to update the input stress variable in real time to predict the stress at the  $(i + 1)$ th step. In addition, the strain  $\varepsilon$  at the  $(i + 1)$ th step is updated by the following:

$$\varepsilon^i = \varepsilon^{i-1} + \Delta\varepsilon^i \tag{12}$$

To eliminate the effect of scales of parameters on the model performance and improve convergence, all datasets need to be pre-processed. Herein, the independent variables (such as  $\sigma_{n0}$  and  $p$ ) only have their initial values keeping

constant, and the strain and strain increment are manually pre-set, which comply with uniform distribution. Considering the distributions of all variables are different from each other and do not conform to Gaussian distribution, Minmax normalization method instead of standardization method is used in this study, as shown in follows:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} (\bar{x}_{max} - \bar{x}_{min}) + \bar{x}_{min} \tag{13}$$

where  $x$  = actual value of input variables,  $x_{min}$  = minimum value of input variables and  $x_{max}$  = maximum value of input variables.  $\bar{x}_{min} = -1; \bar{x}_{max} = 1$ .

### 3 ML-based constitutive models using synthetic data

#### 3.1 Synthetic data by a simple soil model

To comprehensively compare the performance of three ML algorithms and two modelling strategies on developing constitutive models, a simple sand shear constitutive model was first used to generate synthetic datasets (see Eq. (14)). The purpose of ML-based constitutive models developed based on synthetic datasets rather than directly based on the experimental data is to eliminate the interference of experimental and measurement errors on the mapping capability of ML algorithms [88]. Moreover, the experimental data tend to be limited and insufficient for comparison of ML algorithms’ performance, whereas the data can be generated infinitely by a theoretical function.



**Table 2** Hyper-parameters regarding model framework in three selected ML algorithm

Algorithm	Description	Range (T/I)
BPNN	Number of hidden layers	1/1
	Number of neurons in hidden layers	1–5/1–9
ELM	Number of neurons in hidden layers	1–11/1–11
EPR	Number of transformed variables	1–11/1–11

T = total stress–strain strategy, I = incremental stress–strain strategy

**Table 3** Methods for determining the number of hidden neuron

Methods	References	Number of neurons (T/I)
$\leq 2N_i + 1$	Nielsen [47]	5/9
$\frac{2+N_i \times N_o + 0.5N_o \times (N_o^2 + N_i) - 3}{N_o + N_i}$	Paola [48]	1/1
$2N_i/3$	Wang [63]	2/3
$\sqrt{N_i \times N_o}$	Masters [43]	2/2
$2N_i$	Iebling and Milton [23]	4/8

$N_i$  = number of input variables,  $N_o$  = number of output variables, T = total stress–strain strategy, I = incremental stress–strain strategy

$$\tau = \sigma_{n0} \frac{\mu\gamma}{1/G + \gamma} \tag{14}$$

where  $\sigma_{n0}$  = vertical stress;  $\tau$  = shear stress;  $\gamma$  = shear strain;  $G$  = shear modulus, 1000 kPa; and  $\mu$  = friction angle,  $\tan(\pi/6)$ .

A total of fourteen curves were generated to develop the ML-based constitutive model. Herein, the axial strain  $\gamma$  ranges from 0 to 10%, and a fixed set of axial strain increment  $\Delta\gamma$ , including 0.01%, 0.05%, 0.1%, 0.15% and 0.2%, was chosen for ten curves. Each curve consists of 91 data points. Nine curves ( $\sigma_{n0} = 25, 50, 100, 200, 250, 300, 400, 500$  and  $600$  kPa) with a total of 819 data points were employed to train the ML-based constitutive model, and the remaining five curves ( $\sigma_{n0} = 15, 150, 350, 650$  and  $700$  kPa) were used to test the model.

According to the stress–strain strategy, as mentioned in Eqs. (11)–(12), the vector  $X$  of independent variables in

**Table 4** Values of parameters in the GA algorithm

Algorithm	$p_{cross}$	$p_{mutation}$	Population	Generation
GA	0.7	0.1	20	500/5000

500 = maximum generation for the BPNN and EPR; 5000 = maximum generation for the ELM

this soil model is  $\sigma_{n0}$ . As a result, the total and incremental stress–strain strategies have two and four input variables, respectively. Both have an output variable. The corresponding total and incremental stress–strain strategy can be expressed by

$$\tau^i = f(\sigma_{n0}, \gamma^i) \tag{15}$$

$$\tau^i = f(\sigma_{n0}, \tau^{i-1}, \gamma^{i-1}, \Delta\gamma^i) \tag{16}$$

where the definitions of  $\tau$ ,  $\gamma$  and  $\Delta\gamma$  are similar to those of  $\sigma$ ,  $\varepsilon$  and  $\Delta\varepsilon$  in Eqs. (11)–(12).

### 3.2 Determination of parameters in ML algorithms

The parameters to be determined in the ML algorithms include hyper-parameters and general parameters. The search space of the hyper-parameters regarding the framework of ML are presented in Table 2. A single-layer BPNN was used to construct constitutive models, which is sufficient to capture the stress–strain relationship. Table 3 summarizes several methods of determining the optimal number of hidden neurons. The optimal number of hidden neurons ranged from one to five in the total stress–strain strategy and from one to ten in the incremental stress–strain strategy. Because there is no method for determining the optimal number of neurons and transformed terms in the ELM and EPR, respectively, the ranges of hidden neurons and transformed terms in these two algorithms increase continuously until the number of hidden neurons and transformed terms cannot improve the model’s performance. In this way the ranges of hyper-parameters in three ML algorithms can be determined, as shown in Table 2.

In addition to the hyper-parameters, the initial weights and biases in BPNN and ELM as well as the exponent matrix in the EPR were determined using the GA. Note that the values of exponents must be non-negative in the EPR algorithm, because the datasets include the initial stress–strain stage (0, 0); indeed, negative exponents are wrong under this condition. The values of exponents were thus limited to [0, 1, 2, 3]. Table 4 presents the parameter values in GA. Note that BPNN and EPR are set to a maximum of 500 generations, whereas for the ELM, because of its different convergence rate, the figure is 5000.

### 3.3 Results of the validation set

Figure 3 presents the evolution of loss value generated by three types of ML-based constitutive models using the total stress–strain strategy. It can be clearly observed that the convergence rates of BPNN and EPR are much faster than that of ELM. The loss value roughly holds steady when the generation exceeds 350 and 200 in BPNN and EPR,

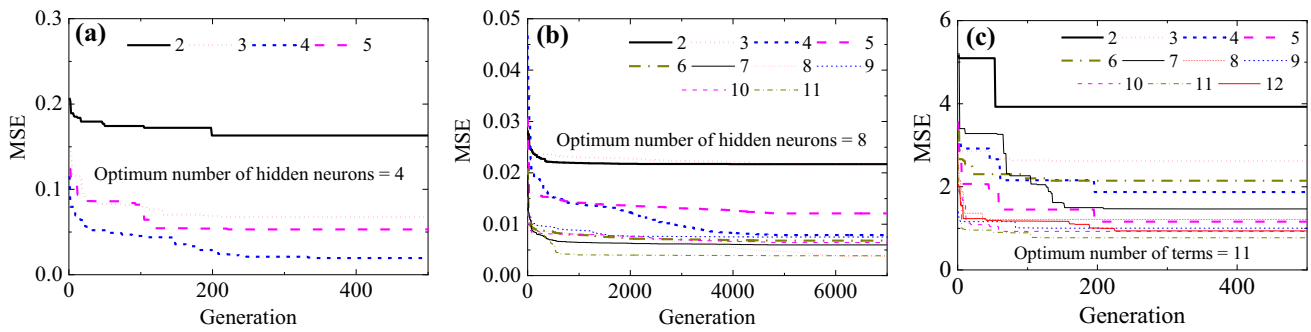


Fig. 3 Evolution of loss value using the total stress–strain strategy for: (a) BPNN; (b) ELM; (c) EPR

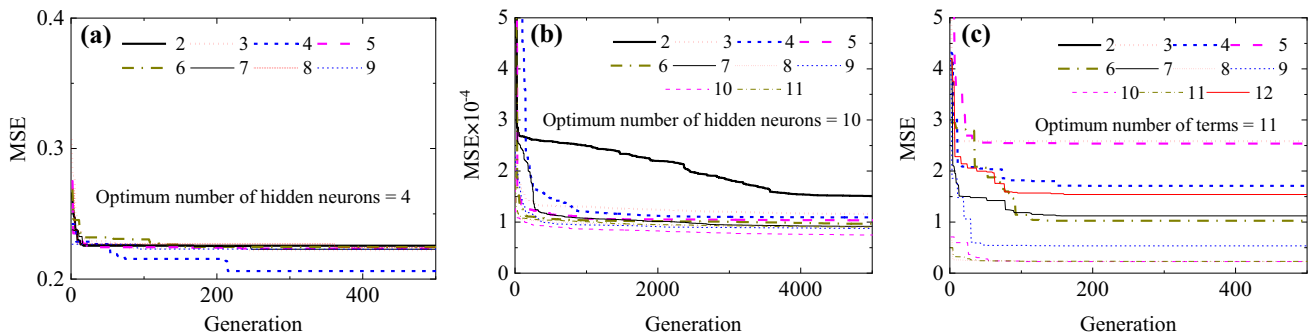


Fig. 4 Evolution of loss value using the incremental stress–strain strategy for: (a) BPNN; (b) ELM; (c) EPR

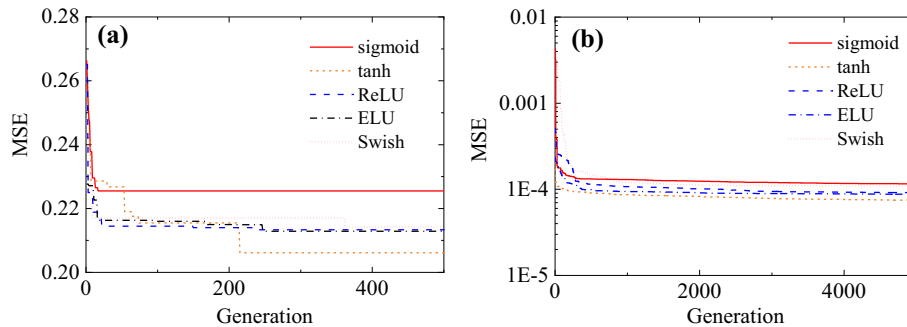


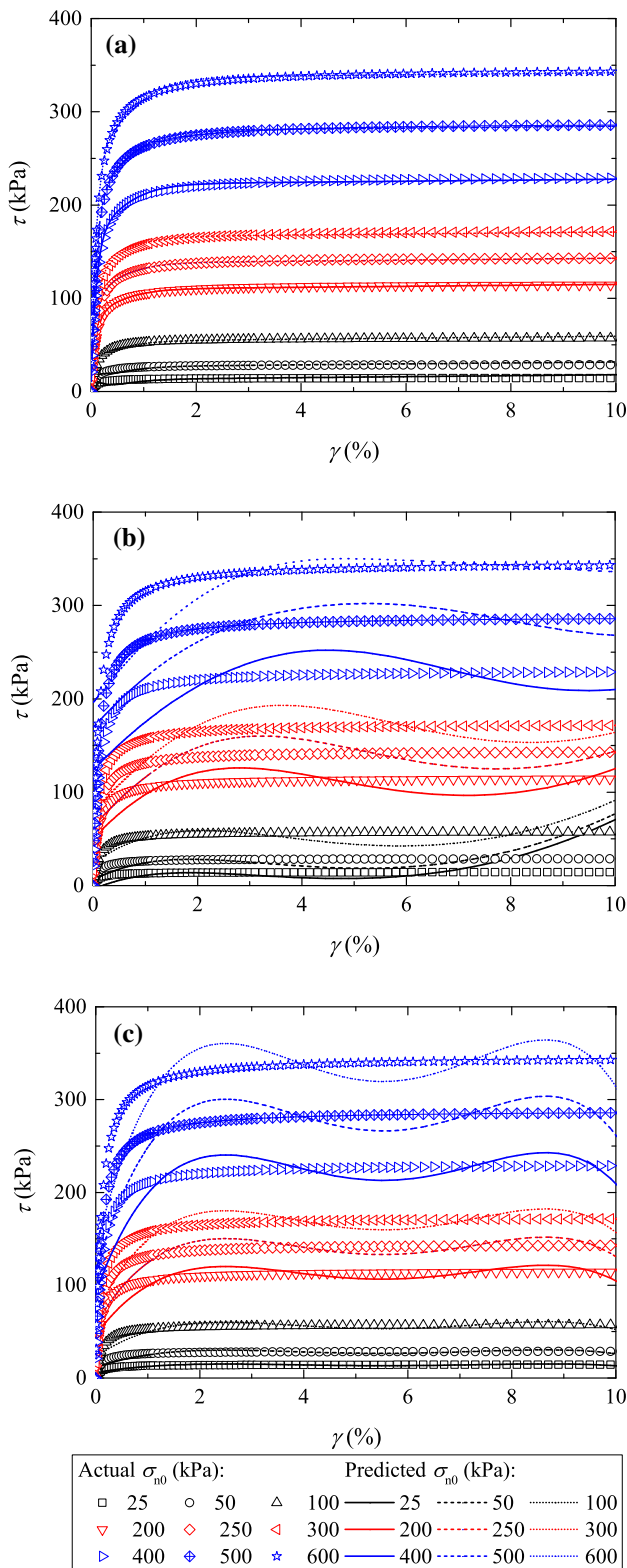
Fig. 5 Evolution of loss value generated by five activation functions in: (a) BPNN; (b) ELM

respectively, whereas the loss value remains roughly constant when the generation reaches 4000 in ELM. From the perspective of the convergent loss value, as shown in Fig. 3, the optimal number of hidden neurons in BPNN and ELM are four and eight, respectively, and the optimal number of transformed terms in EPR is eleven.

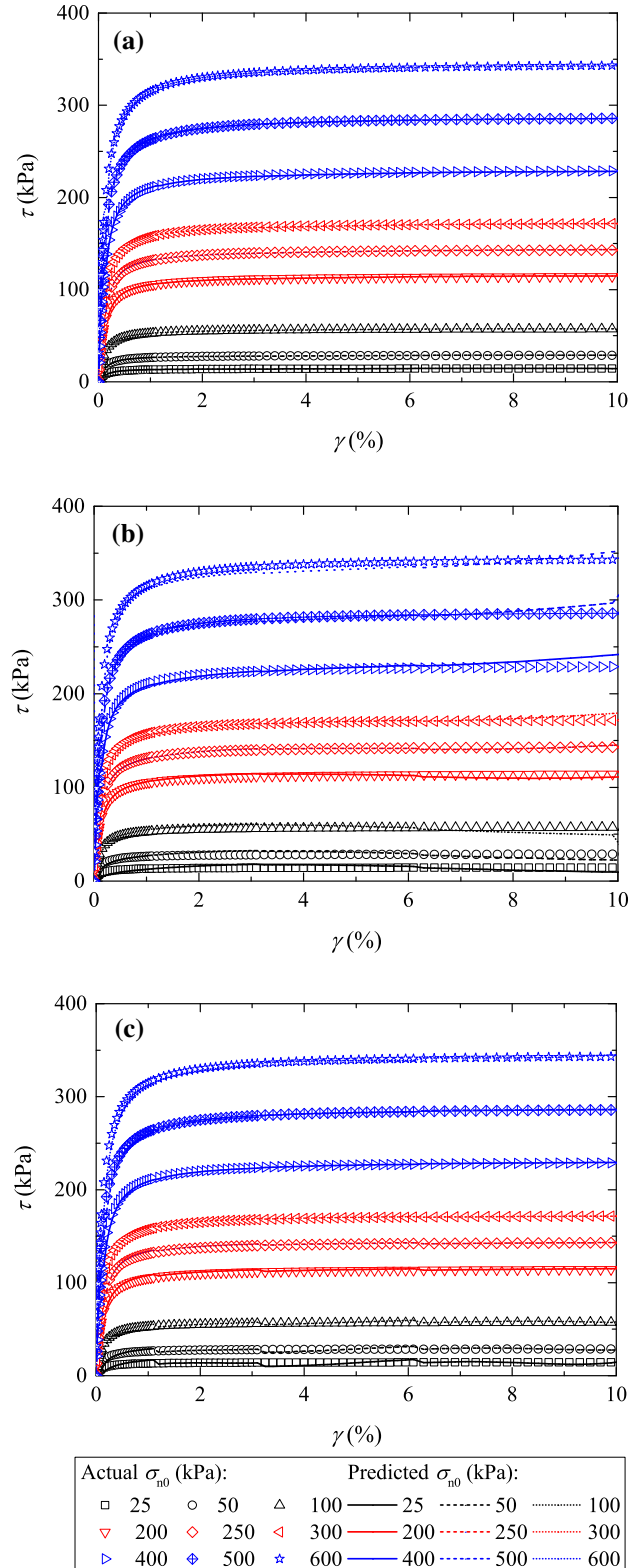
The evolution of loss value generated by three ML algorithms using the incremental stress–strain strategy is shown in Fig. 4. Overall, the convergence rate in three types of ML-based constitutive models using the incremental stress–strain strategy is faster than that using the total stress–strain strategy. The loss value roughly holds steady when the generation reaches 250, 1000 and 100 in the BPNN, ELM and EPR, respectively. From the perspective of the convergent loss value, as shown in Fig. 4,

the optimal numbers of hidden neurons in BPNN and ELM are four and ten, respectively, and the optimal number of transformed terms in EPR is eleven. Note that the optimal loss values are much less than those yielded using the total stress–strain strategy.

Another important hyper-parameter used in ELM and BPNN is the activation function. For comprehensively comparing the performance of BPNN and ELM, the optimum activation functions used in each algorithm should be determined. The commonly used five activation functions are applied, as shown in Eq. [18]. Figure 5 presents the evolution of loss values generated by BPNN and ELM based models (incremental strategy) with five activation functions. It can be observed that the *tanh* is the optimum activation functions in the hidden layers for both BPNN-



**Fig. 6** Predicted results on the training set using the total stress–strain strategy for: (a) BPNN; (b) ELM; (c) EPR



**Fig. 7** Predicted results on the training set using the incremental stress–strain strategy for: (a) BPNN; (b) ELM; (c) EPR



and ELM-based models, thereby the *tanh* is used as the activation function in the following study.

$$\left\{ \begin{array}{l} \text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \\ \text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \\ \text{ReLU}(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \\ \text{ELU}(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & x \leq 0 \end{cases} \\ \text{Swish}(x) = \frac{x}{1 + e^{-x}} \end{array} \right. \quad (17)$$

### 3.4 Results of the training set

The optimal hyper-parameters of the three ML algorithms using two stress–strain strategies are determined as heretofore mentioned. Accordingly, three optimal ML-based constitutive models of each stress–strain strategy are constructed based on the training set. Figure 6 presents the predicted stress–strain curves using three optimally trained models on the basis of the total stress–strain strategy, compared with the measured curves. It is clear that the predicted results of BPNN show perfect agreement with the measured curves, whereas the results predicted by ELM and EPR deviate from the measured curves. In particular, the prediction error for the ELM- and EPR-based constitutive models is much larger at the initial stage (0, 0), and these models also yield a large error at the early stage of stress–strain curves, attributable to the principles of these two algorithms as described in Sect. 3.3.

Figure 7 presents the predicted stress–strain curves using three optimal models based on the incremental stress–strain strategy, compared with the measured curves. It can be seen that all three models can accurately capture the stress–strain curves, which indicates that the

incremental stress–strain strategy for simulating stress–strain relationship shows a significant improvement.

### 3.5 Results of the test set

During the last phase, the performance of the ML-based constitutive model is evaluated against the test set, with  $\sigma_{n0}$  in the training set ranging from 50 to 600 kPa. Generally, test datasets are taken within the range of training datasets, so that test sets with  $\sigma_{n0} = 150$  and 350 kPa are taken into consideration. To investigate the ability of the ML-based constitutive model to extrapolate beyond the range of training datasets, test sets for which  $\sigma_{n0} = 15, 650$  and 700 kPa are also conducted. Table 5 summarizes the values of indicators for these five test sets. For the interpolated test sets, Fig. 8 presents the results of simulation using three optimal ML-based constitutive models based on the total stress–strain strategy. The predicted stress–strain curve using BPNN largely agrees with the measured curve, and the corresponding MAE and MAPE values are also lower than those produced by ELM and EPR. Notably, ELM and EPR cannot accurately predict initial stress when strain equals zero, and ELM- and EPR-based constitutive models cannot accurately predict the evolution of stress. Figure 9 presents the results of simulation using three optimal ML models based on the incremental stress–strain strategy for the test set. The predicted stress–strain curves using the BPNN-based constitutive model still agree perfectly with the measured curves and outperform the ELM- and EPR-based constitutive models. The performance of the ELM-based constitutive model is better than that of the total stress–strain strategy, and it also accurately captures the evolution of stress. Nevertheless, the change in the performance of the EPR-based constitutive model is different from others, perfectly predicting the stress–strain relationship for  $\sigma_{n0} = 350$  but exhibiting worse performance at predicting the stress–strain relationship for  $\sigma_{n0} = 150$ . Note that prediction performance at the initial stage

**Table 5** Values of indicators for the test set

Strategy	Algorithm	Interpolation ( $\sigma_{n0}$ , kPa)				Extrapolation ( $\sigma_{n0}$ , kPa)					
		MAE		MAPE		MAE			MAPE		
		150	350	150	350	15	650	700	15	650	700
T	BPNN	0.58	1.36	1.22	1.02	1.95	2.84	3.35	28.81	1.13	1.21
	ELM	9.42	22.58	18.88	20.63	8.88	34.16	41.34	142.46	19.60	19.64
	EPR	6.28	14.66	14.06	14.06	0.63	27.23	29.32	14.06	14.06	14.06
I	BPNN	2.07	3.69	4.17	2.97	0.88	5.02	8.03	12.79	2.45	3.13
	ELM	4.44	5.46	9.05	4.09	6.96	15.36	18.88	151.64	7.92	9.31
	EPR	26.95	10.25	33.25	6.33	23.90	56.08	49.59	290.27	17.98	18.06

T = total stress–strain strategy, I = incremental stress–strain strategy

of ELM- and EPR-based constitutive models is clearly improved from that seen with the total stress–strain strategy. Overall, ML-based constitutive models that use the incremental stress–strain strategy offer reliable performance for interpolated test sets, and a BPNN-based constitutive model exhibits the best performance.

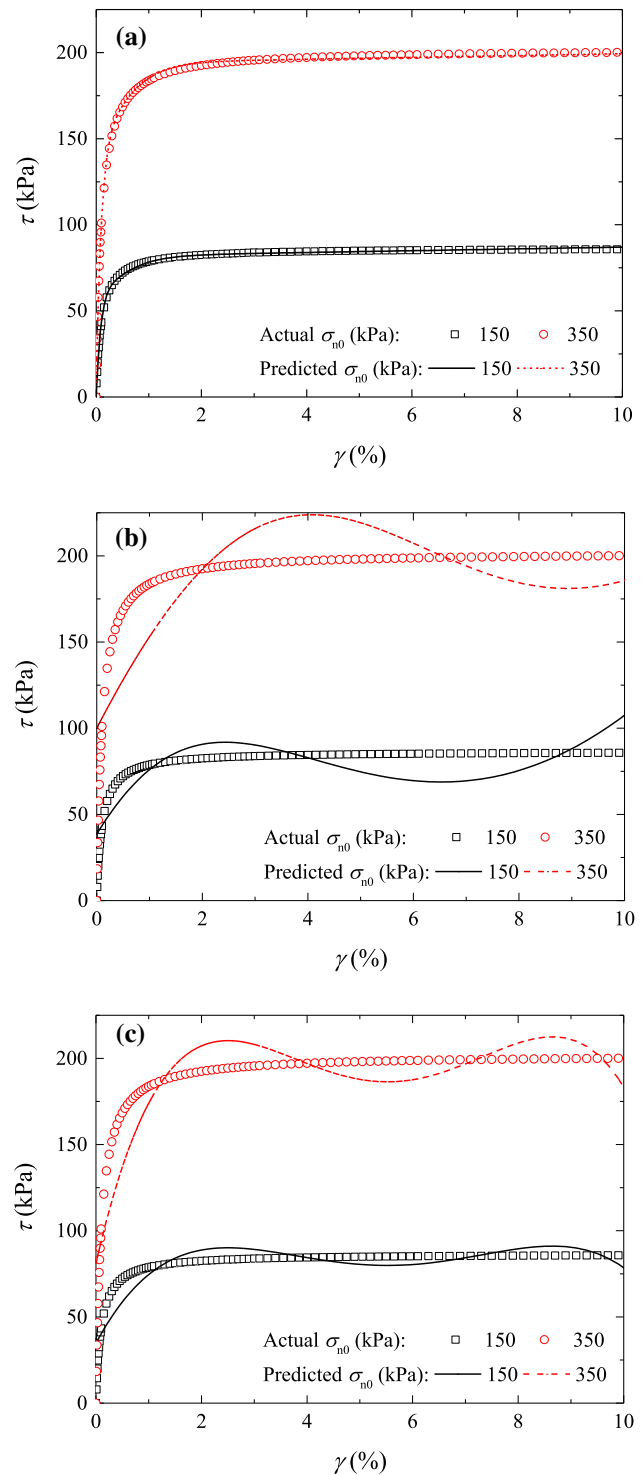
The extrapolated test sets are used to further examine the generalization ability of ML-based models. Figure 10 presents the results of simulation using three optimal ML-based constitutive models based on the total stress–strain strategy. For  $\sigma_{n0} = 650$  and 700 kPa, three ML-based constitutive models can still capture the stress–strain relationship. The BPNN-based constitutive model performs perfectly, followed by the EPR and ELM-based constitutive models. However, for  $\sigma_{n0} = 15$  kPa, the predicted stress–strain curve by the BPNN-based constitutive model deviates from the actual stress–strain curve. The results of simulation using three optimal ML-based constitutive models based on the incremental stress–strain strategy are shown in Fig. 11. In addition to stress–strain curves for  $\sigma_{n0} = 650$  and 700 kPa, it can be observed that the BPNN-based constitutive model's ability to predict the stress–strain relationship for  $\sigma_{n0} = 15$  kPa improves significantly. Meanwhile, the performance of the ELM-based constitutive model improves dramatically with lower MAE and MAPE values, whereas the prediction performance of EPR-based constitutive model decreases.

Overall, ML-based constitutive models are better at predicting stress–strain relationships within the range of the training datasets than at extrapolating beyond the range of the training datasets. ML-based constitutive models developed using the incremental stress–strain strategy outperform those developed using the total stress–strain strategy. A BPNN-based constitutive model developed using the incremental stress–strain strategy is thus recommended for describing the stress–strain relationship, because this model makes highly accurate predictions of the stress–strain relationship for the interpolated and extrapolated test sets.

## 4 ML-based constitutive models using real data

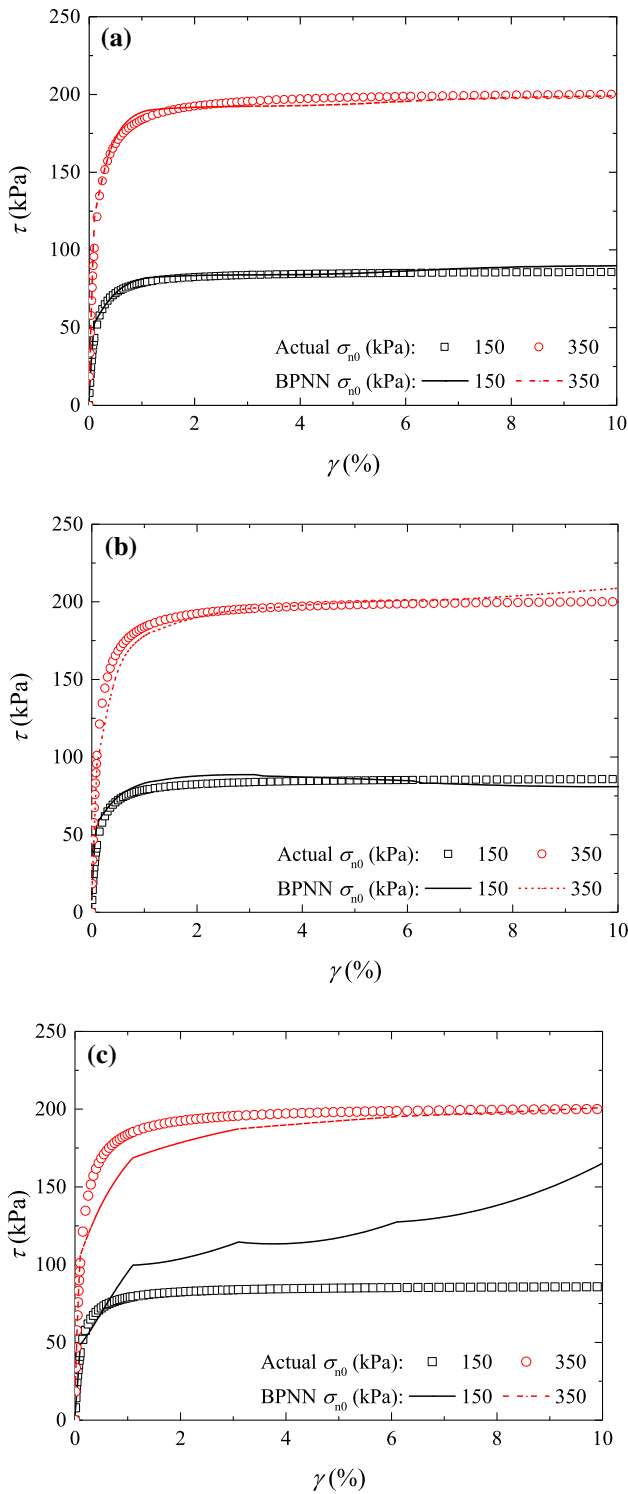
### 4.1 Database

To investigate ML-based constitutive models' ability to predict soil behaviour in engineering practice, this study uses datasets from twelve sets of triaxial compression shear tests conducted by [18] on Kaolinite clays having various over-consolidation ratios (OCRs). The results of shear and void ratio behaviour are collected as shown in Fig. 12. Herein, datasets from nine tests having OCRs of 1, 2, 2.25,



**Fig. 8** Predicted results on the test set (interpolation) using the total stress–strain strategy: (a) BPNN; (b) ELM; (c) EPR

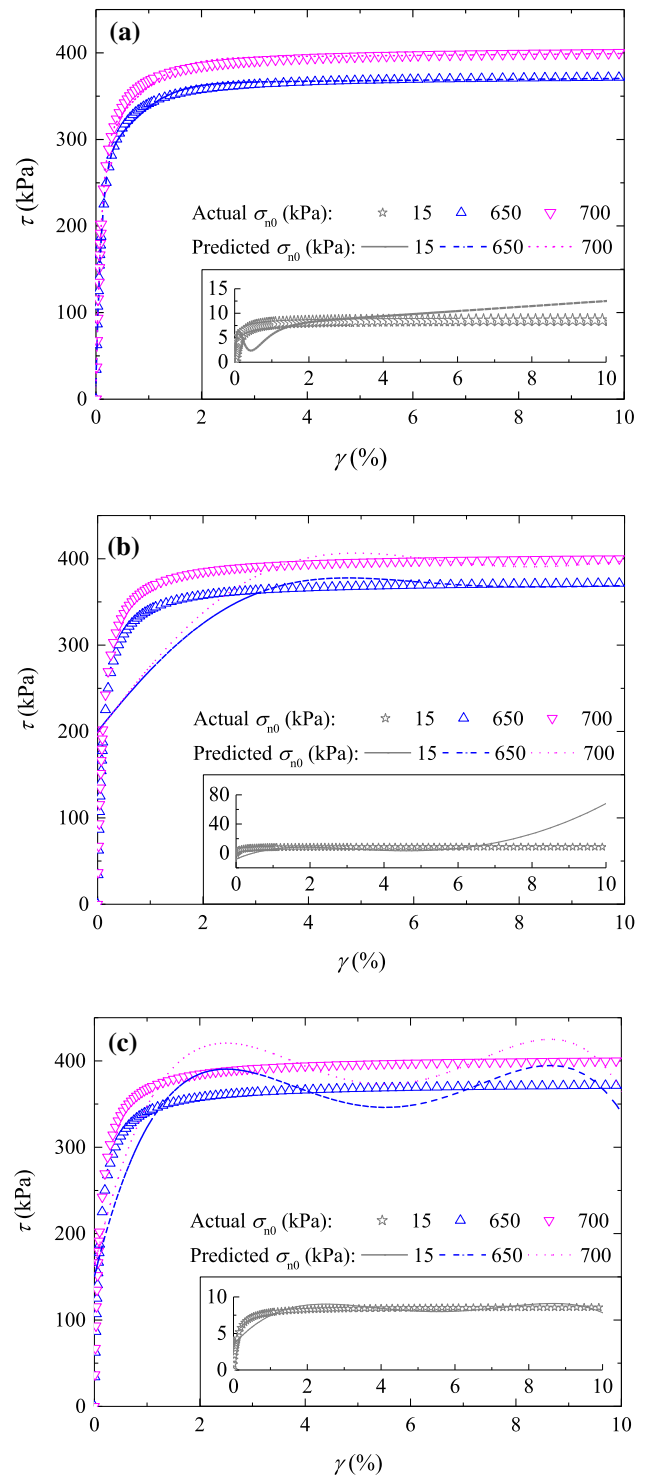
2.5, 2.7, 4, 5, 10 and 20 are used to train the model, and the remaining three, with OCRs of 3, 8 and 50 kPa, are used to test it.



**Fig. 9** Predicted results on the test set (interpolation) using the incremental stress–strain strategy: (a) BPNN; (b) ELM; (c) EPR

### 4.2 Selection of a simulation strategy

According to previous comparisons, BPNN integrating the incremental stress–strain strategy is used to model



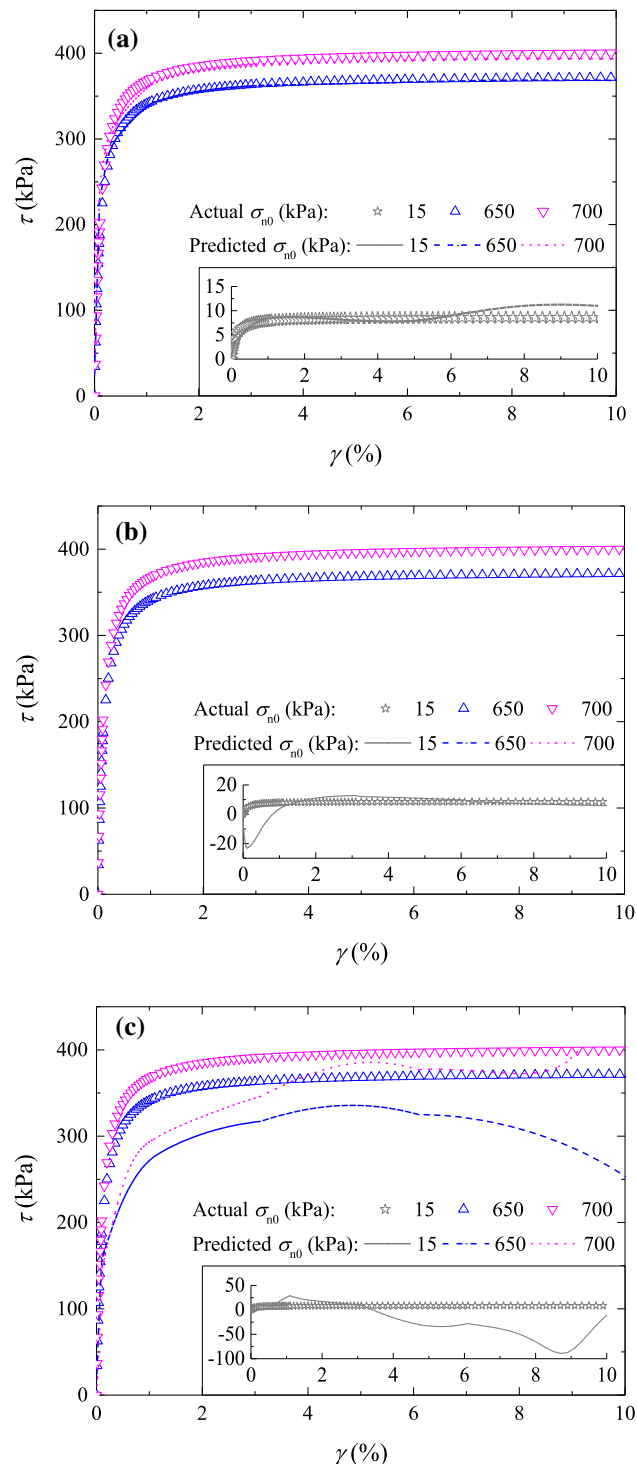
**Fig. 10** Predicted results on the test set (extrapolation) using the total stress–strain strategy: (a) BPNN; (b) ELM; (c) EPR

Kaolinite clays’ behaviour. According to the incremental stress–strain strategy seen in Eq. (16), the vector  $X$  of independent variables is the OCR, and there are two output variables: deviatoric stress  $q$  and void ratio  $e$ . Accordingly,

ML-based Kaolin clays’ constitutive models can be obtained by

$$q^i = f(p^{i-1}, q^{i-1}, e^{i-1}, e_1^{i-1}, \Delta \varepsilon_1^i) \tag{18}$$

$$e^i = g(p^{i-1}, q^{i-1}, e^{i-1}, e_1^{i-1}, \Delta \varepsilon_1^i) \tag{19}$$



**Fig. 11** Predicted results on the test set (extrapolation) using the incremental stress–strain strategy: **(a)** BPNN; **(b)** ELM; **(c)** EPR

where  $p^{i-1}, q^{i-1}, e^{i-1}, e_1^{i-1}$  = mean stress, deviatoric stress, void ratio and axial strain at the  $(i-1)$ th steps, respectively;  $q^i, e^i, \Delta \varepsilon_1^i$  = deviatoric stress, void ratio and axial strain increment at the  $i$ th step, respectively; and  $f, g$  = formulations of deviatoric stress–strain and void ratio–strain relationships.

Figure 13 presents the framework of the BPNN-based constitutive model for predicting Kaolinite clay behaviour. Note that the predicted deviatoric stress and void ratio at the  $i$ th step must update the deviatoric stress and void ratio in real time to predict deviator stress and volumetric strain at the  $(i + 1)$ th step. Updates to strain  $\varepsilon$  at the  $(i + 1)$ th step follow Eq. (12). After training, formulations of BPNN-based Kaolinite constitutive models are found and summarized in Appendix.

### 4.3 Results of simulation

Validation results of these simulations with which to determine the optimal parameters of the BPNN-based Kaolinite constitutive model are not presented, but Appendix A presents the formulation of optimal the BPNN-based Kaolinite clay constitutive model in detail. It can be observed that the optimal number of hidden neurons in BPNN is 8. Figure 14 presents the results of the training set predicted by the optimal BPNN model, compared with the measured results, showing that the BPNN-based constitutive model can accurately capture non-linear deviatoric stress–strain and void ratio–strain relationships.

Figure 15 presents the results of the predicted deviatoric stress–strain and void ratio–strain relationships for the interpolated test set. Because this study uses the recursive simulation strategy, prediction error accumulates gradually with increasing strain. The accumulated error is negligible up to strain of 20% for simulation of the deviatoric stress–strain relationship. By contrast, the predicted void ratio–strain curve gradually deviates from the accrual curve when strain exceeds 10%. Overall, the BPNN-based constitutive model better simulates the deviatoric stress–strain relationship, likely because the void ratio–strain relationship is more complicated than the deviatoric stress–strain relationship. Figure 15 shows that deviatoric stress increases monotonically with strain for all experimental tests, whereas the void ratio–strain relationship differs for different OCR values because of the dilatant behaviour associated with high OCR and the contractive behaviour associated with low OCR.

Figure 16 presents the predicted deviatoric stress–strain and void ratio–strain relationships for the extrapolated test set. Absent experimental results for OCR = 22.5, 25, 27.5 and 30, the reasonability of predicted curves is referred from the results for OCR = 20 and 50. For OCR = 50, a

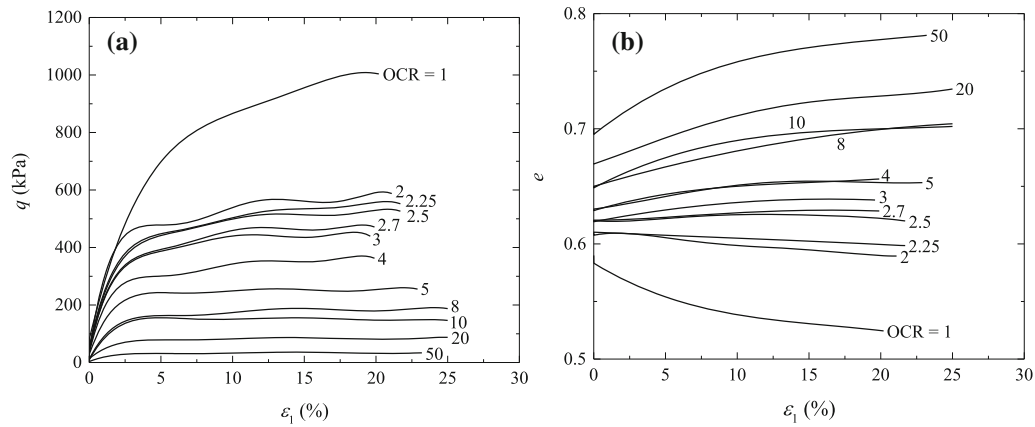


Fig. 12 Experimental data of Kaolinite clay [18]

predicted deviatoric stress–strain curve using the BPNN-based constitutive model agrees well with the actual curve, although no experimental data are available in the training set beyond OCR = 20. Predicted curves for OCR = 22.5, 25, 27.5 and 30 also suggest reasonable trends (deviatoric stress increasing monotonically with increasing strain and peak deviatoric stress increasing with decreasing OCR), with all results falling into the range OCR = 20 and OCR = 50. However, the BPNN-based constitutive model’s predicted void ratio–strain curves obviously deviate from the actual curves when strain exceeds 5%. Overall, the BPNN-based constitutive model performs well (in terms of both interpolation and extrapolation) at simulating actual soil behaviour so long as datasets are sufficient. What’s more, the ability to obtain simple, explicit function can further extend the application of the BPNN-based constitutive model.

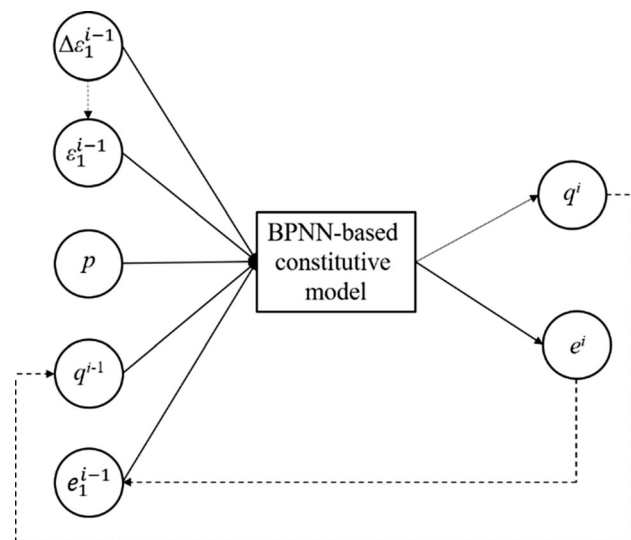


Fig. 13 Framework for predicting Kaolinite clay behaviours

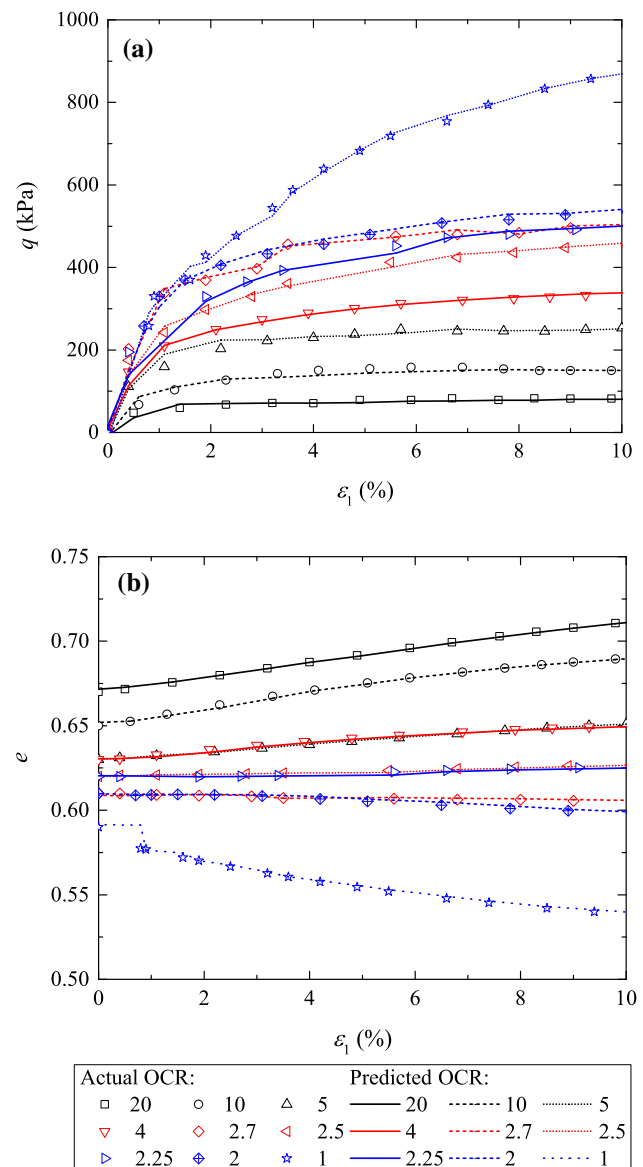
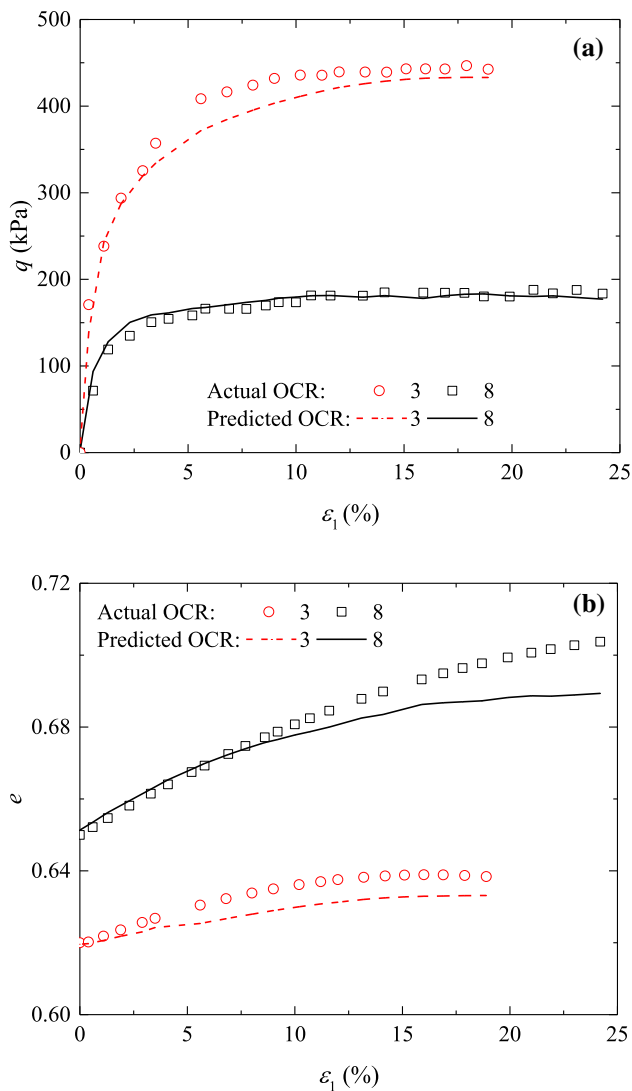


Fig. 14 Predicted results on the training set using BPNN-based constitutive model with incremental stress–strain strategy: (a)  $q-\varepsilon_1$ ; (b)  $e-\varepsilon_1$



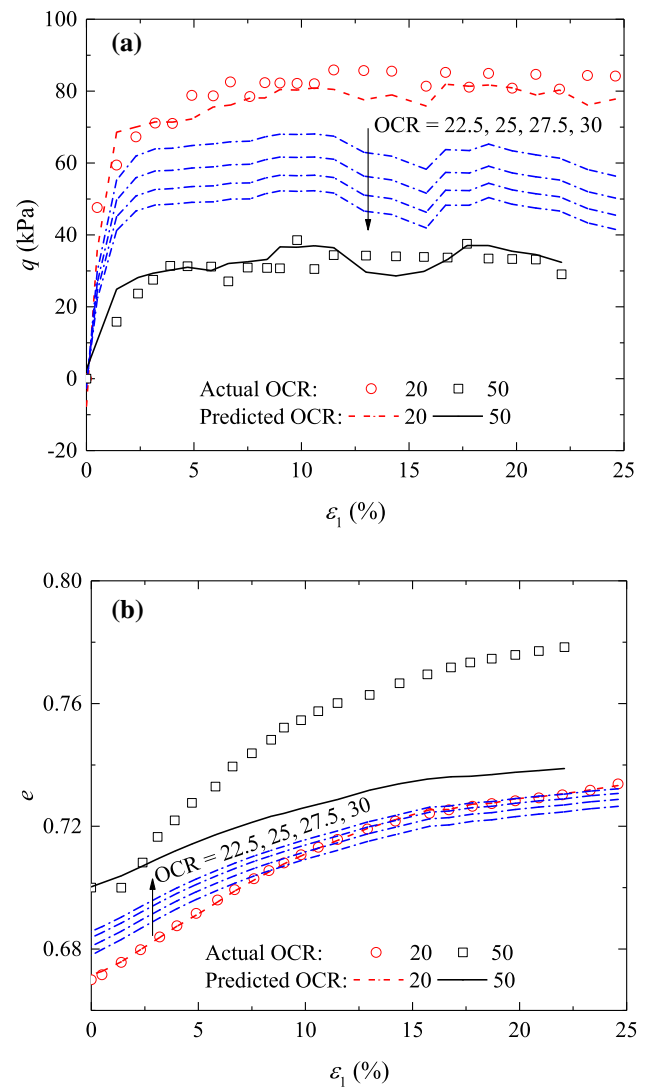


**Fig. 15** Predicted results on the test set (interpolation) using BPNN based on the incremental stress–strain strategy: (a)  $q$ – $\varepsilon_1$ ; (b)  $e$ – $\varepsilon_1$

#### 4.4 Comparison of different ML based models

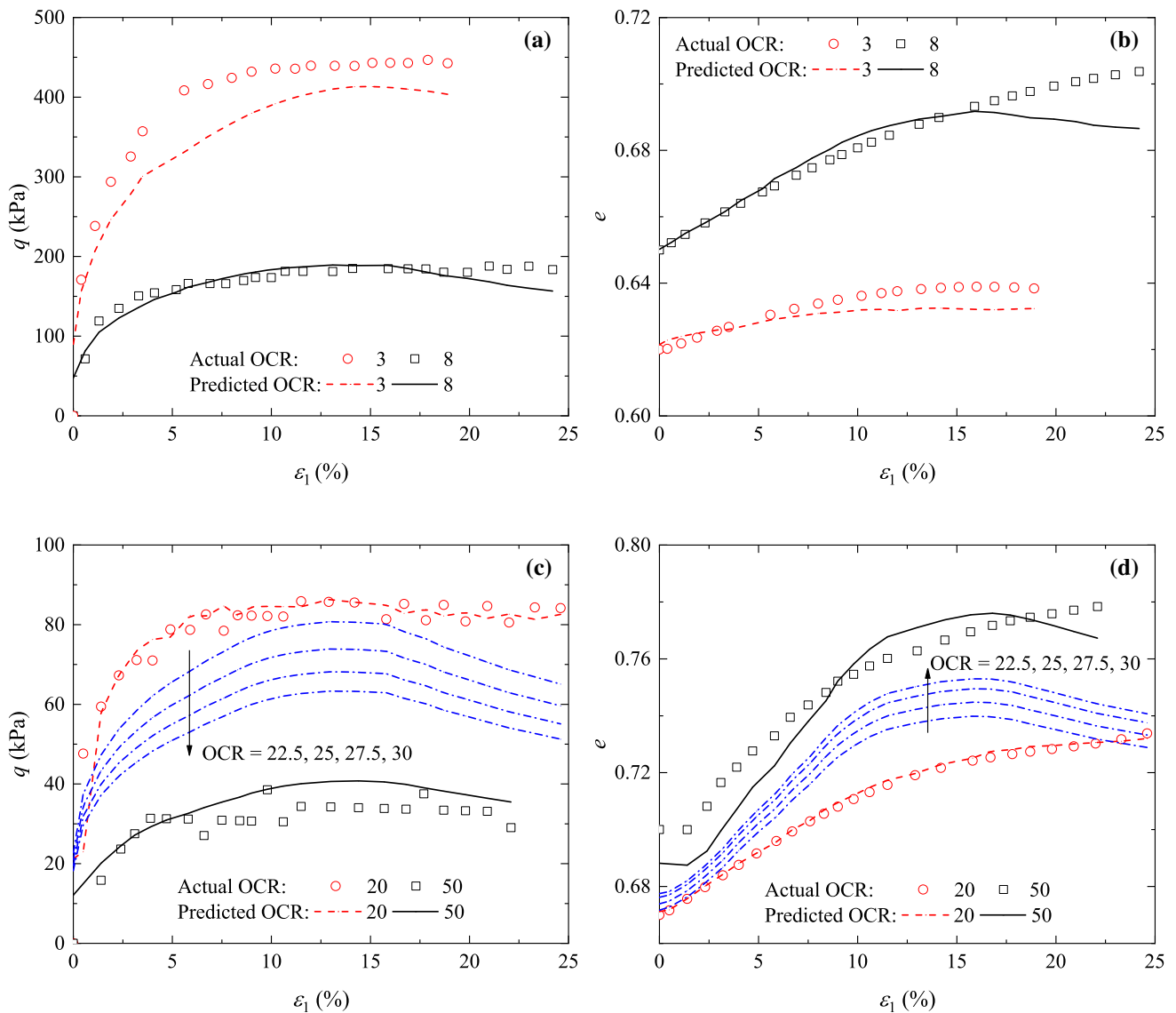
To further compare the performance of BPNN for modelling soil behaviours with ELM and EPR, the latter two ML algorithms are also used to predict the behaviours of Kaolinite clays. It should be noted that the training, test datasets and also the modelling framework for ELM and EPR are consistent with that used in BPNN. Herein, the optimum number of hidden neurons in ELM based model is identified as 7, and the optimum transformed terms in EPR based model is identified as 8. For brevity, the process for determining the hyper-parameters of ELM and EPR are not presented in detail.

The predicted stress–strain relationships on the testing datasets using the optimum ELM-based model are presented in Fig. 17. It can be seen from Figs. 17a, b that the



**Fig. 16** Predicted results on the test set (extrapolation) using BPNN based on the incremental stress–strain strategy: (a)  $q$ – $\varepsilon_1$ ; (b)  $e$ – $\varepsilon_1$

predicted results on the interpolated datasets show a good agreement with experimental results. In Figs. 17c, d, the strain softening and volumetric contraction are obviously observed from the predicted results on extrapolated datasets, which severely violates the measured results. Such factors indicate the ELM-based model can well describe the known soil behaviours, but may be not suitable to predict the soil behaviours on the unseen datasets. Figure 18 presents the predicted stress–strain relationships on the testing datasets using the optimum EPR-based model. Similar to the results presented in Sect. 3, the predicted error on both interpolated and extrapolated datasets are larger than that generated by BPNN- and ELM-based models. It indicates the generalization ability of EPR is inferior to the neural networks-based algorithms; thereby it has difficulty in modelling complicated soil behaviours. Overall, the generalization ability of BPNN is excellent



**Fig. 17** Predicted results on the test set using ELM based on the incremental stress–strain strategy: **(a)**  $q$ – $\epsilon_1$  (interpolation); **(b)**  $e$ – $\epsilon_1$  (interpolation); **(c)**  $q$ – $\epsilon_1$  (extrapolation); **(d)**  $e$ – $\epsilon_1$  (extrapolation)

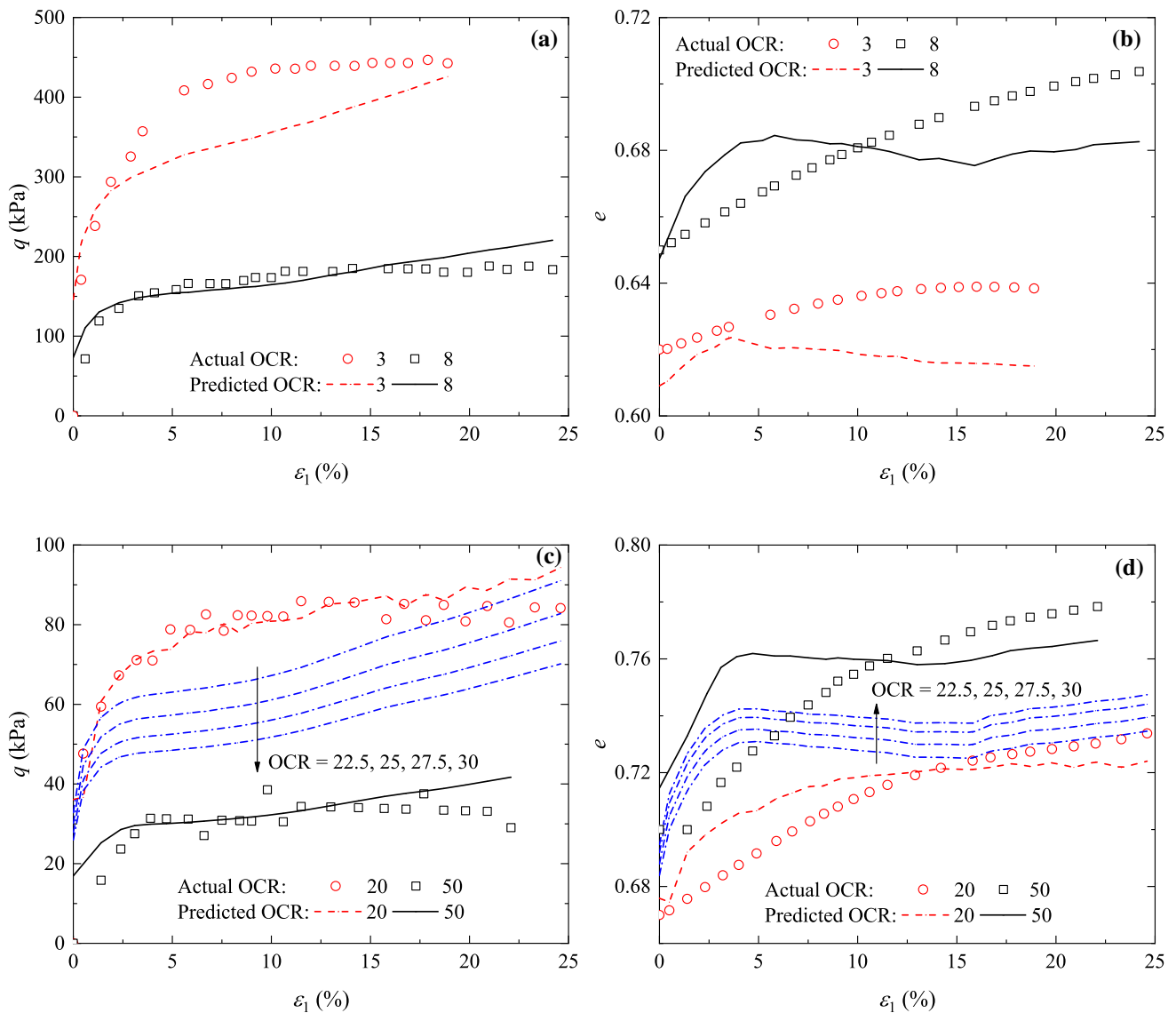
and it can be used to simulate soil behaviours on both known and unknown datasets.

It should be noted that the ML-based model is a kind of data-driven model, thereby its application scope can be expanded as the type and information of datasets increase. For example, if the database involves data under unloading and different stress paths, the ML-based model can be well trained and used to simulate soil behaviours under such conditions. Otherwise, more effective physical mechanism needs to be added to refine the ML based model. Future work will focus on such issues.

### 5 Conclusions

Determination of soil constitutive models is vitally important to engineering practice. ML algorithms have been used to model soil behaviour, because ML-based constitutive models are free of assumptions and offer strong non-linear mapping capabilities. This study systematically demonstrated the application of ML algorithms for construction of a soil constitutive model, using three commonly used ML algorithms able to present an explicit formulation—BPNN, ELM and EPR—to develop models and comprehensively comparing their modelling performance.

A database based on a simple sand shear constitutive model was first built to objectively reveal the capacity of



**Fig. 18** Predicted results on the test set using EPR based on the incremental stress–strain strategy: **(a)**  $q$ – $\varepsilon_1$  (interpolation); **(b)**  $e$ – $\varepsilon_1$  (interpolation); **(c)**  $q$ – $\varepsilon_1$  (extrapolation); **(d)**  $e$ – $\varepsilon_1$  (extrapolation)

three ML algorithms and two modelling strategies to model soil behaviour, with the intent of eliminating potential interference from noise-corrupted experimental data. Although the ML algorithm can learn directly from data, an incremental stress–strain strategy able to take loading path into consideration was more suitable than the total stress–strain strategy for constructing the ML-based constitutive model. ML-based models' hyper-parameters can be determined through the trial-and-error method, and the genetic algorithm can identify general hyperparameters for developing the global optimum model. The application of  $k$ -fold cross validation can enhance the robustness of the ML-based model, facilitating the application of the ML-based model to engineering practice.

Simulation results on theoretical and experimental data indicated that the BPNN-based constitutive model is more stable and accurate, including for interpolation and extrapolation when modelling soil behaviour, than the ELM- and EPR-based constitutive models. Notably, the BPNN-based constitutive model's predictions of deviatoric stress–strain and void ratio–strain relationships for Kaolinite clay agreed with actual experimental data. Overall, the ML-based constitutive model can directly capture non-linear soil behaviours based on limited experimental data without making any assumptions; what's more, an explicit formulation for the constitutive model can be determined that guarantees its application to numerical analysis and engineering practice.

This study focuses only on the modelling of soil behaviours under monotonic loading. Besides, this study merely explores a general modelling framework for ML algorithms, and indicates that the neural network based algorithm is superior to other types of ML algorithms in modelling soil behaviours. As the rapid development in the ML field, more advanced algorithms have been proposed. Future works will investigate the performance of various neural network -based algorithms and determine the most appropriate one for modelling soil behaviours under more complex loading paths.

## Appendix A. Formulations of BPNN-based Kaolinite constitutive models

$$\mathbf{H} = f(\mathbf{WX} + \boldsymbol{\theta}) \quad (1A)$$

$$\mathbf{O} = g(\mathbf{VH} + \boldsymbol{\theta}_o) \quad (2A)$$

where,  $\mathbf{X} = [p, q, e, \varepsilon_1, \Delta\varepsilon_1]$ , matrix of input variables;  $\mathbf{H}$  = matrix of the hidden layer output;  $\mathbf{O} = [q, e]$ , matrix of output variables;  $f = \textit{tansig}$  formulation;  $g = \textit{purlin}$  formulation. Herein,

$$\mathbf{W} = \begin{bmatrix} -1.35315 & 1.907756 & 0.915302 & -2.02378 & -2.16087 \\ 2.595596 & -0.32869 & -2.48042 & 2.352501 & -2.32131 \\ -1.36652 & 0.169634 & 3.622397 & -2.44086 & -3.21463 \\ -0.72626 & -0.01880 & -0.22374 & 0.908566 & 1.41438 \\ -0.02852 & 0.013743 & 0.634650 & -0.57525 & -0.11921 \\ -0.13418 & 0.036866 & 0.933846 & -1.36281 & -1.07577 \\ -0.19825 & -0.51040 & -6.04189 & -2.26296 & -1.74243 \\ -0.19073 & 0.055012 & 1.177738 & -0.69086 & -2.03872 \end{bmatrix}$$

$$\boldsymbol{\theta} = [2.077058; 0.827601; 0.480095; -0.95727; -0.88088; -1.21919; 0.862033; -2.24696]$$

$$\mathbf{V} = \begin{bmatrix} -0.1350 & -0.04501 & -2.23939 & -1.11558 & 1.894278 & -1.86286 & -2.12292 & 1.08516 \\ 0.065208 & -0.02499 & -0.124950 & 0.603675 & 0.853076 & -0.0923 & -0.06472 & -0.63716 \end{bmatrix}$$

$$\boldsymbol{\theta}_o = [0.31; 0.113215]$$

**Acknowledgements** This research was financially supported by the Research Grants Council (RGC) of Hong Kong Special Administrative Region Government (HKSARG) of China (Grant No.: 15209119, R5037-18F); Joint research project between SiChuan Province and National Universities funded by Science & Technology Department of Sichuan Province (No.2019YFSY0015), and Open research grant of MOE Key Laboratory of High-speed Railway Engineering.

## References

- Badawy MF, Msekh MA, Hamdia KM, Steiner MK, Lahmer T, Rabczuk T (2017) Hybrid nonlinear surrogate models for fracture behavior of polymeric nanocomposites. *Probabilist Eng Mech* 50:64–75
- Basheer IA (2000) Selection of methodology for neural network modeling of constitutive hysteresis behavior of soils. *Comput-Aided Civ Inf* 15:440–458
- Cabalar AF, Cevik A (2011) Triaxial behavior of sand–mica mixtures using genetic programming. *Expert Syst Appl* 38(8):10358–10367
- Chang CS, Hicher PY (2005) An elasto-plastic model for granular materials with microstructural consideration. *Int J Solids Struct* 42(14):4258–4277
- Chen RP, Zhang P, Kang X, Zhong ZQ, Liu Y, Wu HN (2019) Prediction of maximum surface settlement caused by EPB shield tunneling with ANN methods. *Soils Found* 59(2):284–295
- Chen RP, Zhang P, Wu HN, Wang ZT, Zhong ZQ (2019) Prediction of shield tunneling-induced ground settlement using machine learning techniques. *Front Struct Civ Eng* 13(6):1363–1378
- Ellis GW, Yao C, Zhao R, Penumadu D (1995) Stress-strain modeling of sands using artificial neural networks. *J Geotech Eng* 121(5):429–435
- Faramarzi A, Javadi AA, Alani AM (2012) EPR-based material modelling of soils considering volume changes. *Comput Geosci-UK* 48:73–85
- Feng XT, Chen BR, Yang CX, Zhou H, Ding X (2006) Identification of visco-elastic models for rocks using genetic programming coupled with the modified particle swarm optimization algorithm. *Int J Rock Mech Min* 43(5):789–801
- Feng XT, Li SJ, Liao HJ, Yang CX (2002) Identification of non-linear stress-strain-time relationship of soils using genetic algorithm. *Int J Numer Anal Met* 26(8):815–830
- Gajo A, Wood M (1999) Severn-Trent sand: a kinematic-hardening constitutive model: the q–p formulation. *Geotechnique* 49(5):595–614
- Gao W, Ge M, Chen D, Wang X (2016) Back analysis for rock model surrounding underground roadways in coal mine based on black hole algorithm. *Eng Comput-Germany* 32(4):675–689
- Gholamy A, Kreinovich V, Kosheleva O (2018) Why 70/30 or 80/20 relation between training and testing sets: a pedagogical explanation. *Int J Intell Technol Appl Stat* 11(2):105–111
- Giustolisi O, Savic DA (2006) A symbolic data-driven technique based on evolutionary polynomial regression. *J Hydroinform* 8(4):235–237
- Guo X, Dias D, Carvajal C, Peyras L, Breul P (2018) Reliability analysis of embankment dam sliding stability using the sparse polynomial chaos expansion. *Eng Struct* 174:295–307
- Habibagahi G, Bamdad A (2003) A neural network framework for mechanical behavior of unsaturated soils. *Can Geotech J* 40(3):684–693
- Hasanipanah M, Noorian-Bidgoli M, Jahed Armaghani D, Khamesi H (2016) Feasibility of PSO-ANN model for predicting surface settlement caused by tunneling. *Eng Comput-Germany* 32(4):705–715
- Hattab M, Hicher PY (2004) Dilating behaviour of overconsolidated clay. *Soils Found* 44(2):27–40
- He S, Li J (2009) Modeling nonlinear elastic behavior of reinforced soil using artificial neural networks. *Appl Soft Comput* 9(3):954–961
- Holland JH. *Adaptation in natural and artificial system*: University of Michigan Press, 1975.
- Hu X, Zhang Y, Guo L, Wang J, Cai Y, Fu H et al (2018) Cyclic behavior of saturated soft clay under stress path with bidirectional shear stresses. *Soil Dyn Earthq Eng* 104:319–328
- Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. *Neurocomputing* 70:489–501
- Itebeling K, Milton B (1996) Designing a neural network for forecasting financial and economic time series. *Neurocomputing* 10(3):215–236
- Javadi AA, Rezaia M (2009) Applications of artificial intelligence and data mining techniques in soil modeling. *Geomech Eng* 1(1):53–74
- Jefferies M (1993) Nor-Sand: a simple critical state model for sand. *Geotechnique* 43(1):91–103
- Jin Y-F, Wu Z-X, Yin Z-Y, Shen JS (2017) Estimation of critical state-related formula in advanced constitutive modeling of granular material. *Acta Geotech* 12(6):1329–1351
- Jin Y-F, Yin Z-Y, Shen S-L, Hicher P-Y (2016) Investigation into MOGA for identifying parameters of a critical-state-based sand model and parameters correlation by factor analysis. *Acta Geotech* 11(5):1131–1145
- Jin Y-F, Yin Z-Y, Shen S-L, Hicher P-Y (2016) Selection of sand models and identification of parameters using an enhanced genetic algorithm. *Int J Numer Anal Met* 40(8):1219–1240
- Jin Y-F, Yin Z-Y, Wu Z-X, Daouadji A (2018) Numerical modeling of pile penetration in silica sands considering the effect of grain breakage. *Finite Elem Anal Des* 144:15–29
- Jin Y-F, Yin Z-Y, Zhou W-H, Huang H-W (2019) Multi-objective optimization-based updating of predictions during excavation. *Eng Appl Artif Intel* 78:102–123
- Jin YF, Yin ZY, Wu ZX, Zhou WH (2018) Identifying parameters of easily crushable sand and application to offshore pile driving. *Ocean Eng* 154:416–429
- Johari A, Javadi AA, Habibagahi G (2011) Modelling the mechanical behaviour of unsaturated soils using a genetic algorithm-based neural network. *Comput Geotech* 38(1):2–13
- Kim J, Hwang W, Kim Y (2018) Effects of hysteresis on hydro-mechanical behavior of unsaturated soil. *Eng Geol* 245:1–9
- Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International joint conference on artificial intelligence*: Morgan Kaufmann Publishers Inc., 1995. p. 1137–1143
- Kohestani VR, Hassanlourad M (2016) Modeling the mechanical behavior of carbonate sands using artificial neural networks and support vector machines. *Int J Geomech* 16(1):04015038
- Kolymbas D. A generalized hypoelastic constitutive law. *Proc XI Int Conf Soil Mechanics and Foundation Engineering*. San Francisco: Balkema, Rotterdam, 1985. p. 2626.
- Koopialipoor M, Jahed Armaghani D, Haghghi M, Ghaleini EN (2017) A neuro-genetic predictive model to approximate over-break induced by drilling and blasting operation in tunnels. *B Eng Geol Environ*:1–10
- Lefik M, Schrefler B (2003) Artificial neural network as an incremental non-linear constitutive model for a finite element code. *Comput Methods Appl Mech Eng* 192(28–30):3265–3283
- Liu C, Jiang Z, Han X, Zhou W (2019) Slope displacement prediction using sequential intelligent computing algorithms. *Measurement* 134:634–648



40. Liu K, Liu B (2019) Intelligent information-based construction in tunnel engineering based on the GA and CCGPR coupled algorithm. *Tunnell Undergr Space Technol* 88:113–128
41. Liu WZ, Shi ML, Miao LC, Xu LR, Zhang DW (2013) Constitutive modeling of the destructuration and anisotropy of natural soft clay. *Comput Geotech* 51:24–41
42. Mašin D (2005) A hypoplastic constitutive model for clays. *Int J Numer Anal Methods Geomech* 29(4):311–336
43. Masters T (1994) *Practical neural network recipes in C++*. Academic Press
44. Müthing N, Zhao C, Hölter R, Schanz T (2018) Settlement prediction for an embankment on soft clay. *Comput Geotech* 93:87–103
45. Nassr A, Esmaeili-Falak M, Katebi H, Javadi A (2018) A new approach to modeling the behavior of frozen soils. *Eng Geol* 246:82–90
46. Ng CWW, Akinniyi DB, Zhou C, Chiu CF (2019) Comparisons of weathered lateritic, granitic and volcanic soils: Compressibility and shear strength. *Eng Geol* 249:235–240
47. Nielsen RH (1987) Kolmogorov's mapping neural network existence theorem. In: *Proceedings of the IEEE first international conference on neural networks*, San Diego, CA, USA, pp 11–13
48. Paola JD (1994) *Neural network classification of multispectral imagery*. The University of Arizona
49. Penumadu D, Zhao RD (1999) Triaxial compression behavior of sand and gravel using artificial neural networks (ANN). *Comput Geotech* 24:207–230
50. Prechelt L (1998) Automatic early stopping using cross validation quantifying the criteria. *Neural Netw* 11:761–767
51. Rashidian V, Hassanlourad M (2014) Application of an artificial neural network for modeling the mechanical behavior of carbonate soils. *Int J Geomech* 14(1):142–150
52. Romo MP, Garcia SR, Mendoza MJ, Taboada-Urtuzuástegui V (2001) Recurrent and constructive-algorithm networks for sand behavior modeling. *Int J Geomech* 1(4):371–387
53. Roscoe KH, Burland J. *On the generalized stress-strain behaviour of wet clay*. *Engineering Plasticity*. Cambridge, UK: Cambridge University Press, 1968. p. 535–609.
54. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323(9):533–536
55. Stone M (1974) Cross-validatory choice and assessment of statistical predictions. *J R Stat Soc C-appl* 36(2):111–147
56. Su D, Yang ZX (2019) Drained analyses of cylindrical cavity expansion in sand incorporating a bounding-surface model with state-dependent dilatancy. *Appl Math Model* 68:1–20
57. Su L-J, Yin J-H, Zhou W-H (2010) Influences of overburden pressure and soil dilation on soil nail pull-out resistance. *Comput Geotech* 37(4):555–564
58. Taiebat M, Dafalias YF (2008) SANISAND: Simple anisotropic sand plasticity model. *Int J Numer Anal Met* 32(8):915–948
59. Tian Y, Yao YP (2017) Modelling the non-coaxiality of soils from the view of cross-anisotropy. *Comput Geotech* 86:219–229
60. Tran C, Srokosz P (2010) The idea of PGA stream computations for soil slope stability evaluation. *Cr Mecanique* 338(9):499–509
61. Turk G, Logar J, Majes B (2001) Modelling soil behaviour in uniaxial strain conditions by neural networks. *Adv Eng Softw* 32:805–812
62. Vermeer P (1978) A double hardening model for sand. *Geotechnique* 28(4):413–433
63. Wang C (1994) *A theory of generalization in learning machines with neural application*. The University of Pennsylvania
64. Wang S, Wu W, Yin Z-Y, Peng C, He X-Z (2018) Modelling time-dependent behaviour of granular material with hypoplasticity. *Int J Numer Anal Methods Geomech* 42(12):1331–1345
65. Wu W, Bauer E, Kolymbas D (1996) Hypoplastic constitutive model with critical state for granular materials. *Mech Mater* 23(1):45–69
66. Xie X, Qi S, Zhao F, Wang D (2018) Creep behavior and the microstructural evolution of loess-like soil from Xi'an area, China. *Eng Geol* 236:43–59
67. Xiong H, Nicot F, Yin Z (2017) A three-dimensional micromechanically based model. *Int J Numer Anal Methods Geomech* 41(17):1669–1686
68. Yao Y, Hou W, Zhou A (2009) UH model: three-dimensional unified hardening model for overconsolidated clays. *Geotechnique* 59(5):451–469
69. Yao Y, Sun D, Luo T (2004) A critical state model for sands dependent on stress and density. *Int J Numer Anal Methods Geomech* 28(4):323–337
70. Yao Y, Sun D, Matsuoka H (2008) A unified constitutive model for both clay and sand with hardening parameter independent on stress path. *Comput Geotech* 35(2):210–222
71. Yao YP, Hou W, Zhou AN (2009) UH model: three-dimensional unified hardening model for overconsolidated clays. *Géotechnique* 59(5):451–469
72. Yin Z-Y, Chang CS, Karstunen M, Hicher P-Y (2010) An anisotropic elastic-viscoplastic model for soft clays. *Int J Solids Struct* 47(5):665–677
73. Yin Z-Y, Jin Y-F, Shen S-L, Huang H-W (2016) An efficient optimization method for identifying parameters of soft structured clay by an enhanced genetic algorithm and elastic-viscoplastic model. *Acta Geotech* 12(4):849–867
74. Yin Z-Y, Karstunen M (2011) Modelling strain-rate-dependency of natural soft clays combined with anisotropy and destructuration. *Acta Mech Solida Sin* 24(3):216–230
75. Yin Z-Y, Karstunen M, Chang CS, Koskinen M, Lojander M (2011) Modeling time-dependent behavior of soft sensitive clay. *J Geotech Geoenviron Eng* 137(11):1103–1113
76. Yin Z-Y, Zhao J, Hicher P-Y (2014) A micromechanics-based model for sand-silt mixtures. *Int J Solids Struct* 51(6):1350–1363
77. Yin ZY, Chang CS (2009) Microstructural modelling of stress-dependent behaviour of clay. *Int J Solids Struct* 46(6):1373–1388
78. Yin ZY, Chang CS, Hicher PY (2010) Micromechanical modelling for effect of inherent anisotropy on cyclic behaviour of sand. *Int J Solids Struct* 47(14–15):1933–1951
79. Yin ZY, Chang CS, Hicher PY, Karstunen M (2009) Micromechanical analysis of kinematic hardening in natural clay. *Int J Plast* 25(8):1413–1435
80. Yin ZY, Chang CS, Karstunen M, Hicher PY (2010) An anisotropic elastic-viscoplastic model for soft clays. *Int J Solids Struct* 47(5):665–677
81. Yin ZY, Jin YF, S SJ, Hicher PY (2017) Optimization techniques for identifying soil parameters in geotechnical engineering: comparative study and enhancement. *Int J Numer Anal Met* 42(1):1–25
82. Yu H (1998) CASM: a unified state parameter model for clay and sand. *Int J Numer Anal Methods Geomech* 22(8):621–653
83. Zhang P, Wu H-N, Chen R-P, Chan THT (2020) Hybrid meta-heuristic and machine learning algorithms for tunneling-induced settlement prediction: a comparative study. *Tunnell Undergr Space Technol* 99:103383
84. Zhang P, Yin Z-Y, Jin Y-F, Chan THT (2020) A novel hybrid surrogate intelligent model for creep index prediction based on particle swarm optimization and random forest. *Eng Geol* 265:105328
85. Zhang P, Yin Z-Y, Zheng Y, Gao F-P (2020) A LSTM surrogate modelling approach for caisson foundations. *Ocean Eng* 204:107263
86. Zhang P, Yin ZY, Jin YF (2021) State-of-the-art review of machine learning applications in constitutive modeling of soils.

- Arch Comput Method Eng. <https://doi.org/10.1007/s11831-020-09524-z>
87. Zhang P, Yin ZY, Jin YF, Chan T, Gao FP (2021) Intelligent modelling of clay compressibility using hybrid meta-heuristic and machine learning algorithms. *Geosci Front* 12(1):441–452
88. Zhang P, Yin ZY, Jin YF, Ye GL (2020) An AI-based model for describing cyclic characteristics of granular materials. *Int J Numer Anal Met* 44(9):1315–1335
89. Zhou YF, Tham LG, Yan WM, Dai FC, Xu L (2014) Laboratory study on soil behavior in loess slope subjected to infiltration. *Eng Geol* 183:31–38
90. Zhu JH, Zaman MM, Anderson SA (1998) Modelling of shearing behaviour of a residual soil with Recurrent Neural Network. *Int J Numer Anal Met* 22(8):671–687

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.