State-of-the-Art Review of Machine Learning Applications in Constitutive Modeling of Soils

Pin ZHANG[1], Zhen-Yu YIN[1,*] and Yin-Fu JIN[1]

1 Department of Civil and Environmental Engineering, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

* Corresponding author: Dr Zhen-Yu YIN, Tel: +852 3400 8470; Fax: +852 2334 6389; E-mail: zhenyu.yin@polyu.edu.hk; zhenyu.yin@gmail.com

Abstract: Machine learning (ML) may provide a new methodology to directly learn from raw data to develop constitutive models for soils by using pure mathematic skills, and it has presented success and versatility in cases of simple stress paths due to its strong non-linear mapping capacity without limitations of constitutive formulations. However, current studies on the ML based constitutive modeling of soils is still very limited. This study comprehensively reviews the application of ML algorithms in the development of constitutive models of soils and compares the performance of different ML algorithms. First, the basic principles of typical ML algorithms used in describing soil behaviors are briefly elaborated. The main characteristics and the limitations of such ML algorithms are summarized and compared. Then, the methodology of developing a ML-based soil model is reviewed from six aspects, such as adopted ML algorithms, data source, framework of the ML based model, training strategy, generalization ability and application scope. Finally, five new ML based models are developed using five typical ML algorithms (i.e. BPNN, RBF, LSTM, GRU and BiLSTM that can predict multi outputs together) based on same set of experimental results of sand, and compare each other in terms of the predictive accuracy and generalization ability. Results show the long short-term memory (LSTM) neural network and its variants are most suitable for developing constitutive models. Moreover, some useful suggestions for developing ML-based soil models are also provided for the community.

Keywords: Soils; stress-strain relationship; constitutive model; machine learning; optimization; artificial intelligence

1 Introduction

Soil is a complicated granular material that exhibits non-linear mechanical behaviors involving state-dependence [1, 2], stress dilatancy [3], anisotropy [4, 5], destructuration [6, 7], stress-path dependence [8], time-dependence [9], and non-coaxiality [10], and so on. To describe such complicated soil behaviors for prompting their application in engineering practice, researchers have preoccupied with proposing constitutive models during last decades. Such models can be mainly categorized into (1) linear elastic perfectly plastic models (e.g. the Mohr-Coulomb model, Drucker Prager model), (2) nonlinear models (e.g. the hardening soil [11] and nonlinear Mohr-Coulomb [12] models), (3) critical state–based models (e.g. the modified cam-clay model [13], Nor-Sand model [14], CSAM model [15], Severn–Trent model [16], UH models [17-19], SANISAND model [20], SIMSAND model [12, 21, 22] and ANICREEP model [23], hypoplasticity [24-27]) and (4) micromechanical models [28-33]. In general, these models have four main limitations in simulating soil behaviors: (1) all constitutive models are proposed on the basis of certain assumptions [4, 9, 34]; (2) each model is only suitable for few soil types; (3) although the mathematical formulas in a constitutive model are derived from the experimental data, and the formula's form presents excellent accuracy for the selected tests, meanwhile it limits the model's predictive ability for other tests with different stress paths; (4) the mathematical formulas become increasingly complicated with numerous parameters [35], as presented in Fig. 1, resulting in difficulties with respect to the calibration of parameters and applications in engineering practice. Furthermore, the complexity of advanced constitutive models generally increases the risk of non-convergence during the numerical analysis using such models being implemented into numerical codes.

Fig. 1 Relationship between the complexity of constitutive model and the number of parameters

Machine learning (ML) has been back on the stage of research works in all the walks in recent years [36-40] due to its excellent capacity of solving nonlinear problems with desired speed and accuracy [41]. Therefore, ML may provide a new methodology to model the complicated mechanical behaviors of soils. In general, ML has three advantages in developing constitutive models of soils: (1) ML algorithms can directly learn the stress–strain relationship from the raw data without making any assumptions [42-44]; (2) ML is able to develop a uniform model for simulating behaviors of various soils as long as the experiments of such soils are involved in a database; (3) the predictive accuracy and application scopes of ML-based models can be improved with the increasing number of datasets; (4) ML based model is a data-driven model, thereby no parameters calibration is needed once the configurations of ML are determined.

However, current studies of ML based constitutive modeling exhibit obvious limitations, which can be concluded from two aspects: model development and application. From the perspective of developing a ML based constitutive model, most of models were developed by conventional ML algorithms, but some effective ML algorithms such as LSTM or its variants that are characterized by predicting sequential data such as stress-strain relationship have been rarely used. Current ML based models were developed based on experimental or synthetic datasets generated by physics-based constitutive models. The performance of these ML based models was merely evaluated using several experimental tests of a given soil, thereby it is hard to guarantee the robustness of such models and the feasibility of applying such models to predict stress-strain relationship of other soils. Synthetic datasets are derived from theoretical formulations, thereby ML based models developed based on synthetic datasets cannot show better performance and dig deeper mechanism than physics-based constitutive models used for data creation. Such problems regarding the data source have not been fully discussed and resolved. Currently, the input parameters and framework used in ML based constitutive modeling are diverse, and there is no methodology or suggestion to guide the selection of input parameters and framework. Meanwhile currently used modules (e.g. activation functions) and learning strategies (e.g. optimizers) for constructing ML based models extremely lag the development in the ML domain. Such out-of-style algorithms and training strategies reduce the learning efficiency. The training process is also easily trapped into the local optima. Moreover, methods of preventing overfitting and enhancing model robustness are not used in most of studies of ML based models.

From the perspective of the application of ML based constitutive models, the generalization ability of ML based constitutive models in literature has not been carefully checked. It is clear that extrapolation is the more pervasive task for the prediction of soil stress-strain relationship, but only the interpolation predictability of ML based constitutive models was investigated in most of studies. Moreover, the ML based models in literature were merely used to calculate several stress-strain responses with different values of commonly used soil physical properties such as relative density, overconsolidation ratio, particle size distribution, individual fraction of mixed materials, but the range of such physical properties is very limited. In addition, almost all of these ML based models were independently developed based on drained or undrained datasets, which means that such models cannot simultaneously simulate soil behaviors under complex loading conditions, even simply for both drained and undrained conditions. Overall, most of current ML based models cannot comprehensively simulate mechanical behaviors of a soil sample, and thus the further application would be far from the reality. As a result, the application of ML based constitutive models with the integration of numerical analysis platform for practical engineering project has not been conducted up to now.

Hence, this study aims to comprehensively investigate the current application of ML algorithms in the development of constitutive modeling for soils. Six important aspects for developing a ML based constitutive model such as the adopted ML algorithms, data source, framework of the ML based model, training strategy, generalization ability and application scope are discussed. Then, the limitations of current ML based constitutive models and the potential aspects that deserve to be further improved are summarized. Finally, a real case to compare the performance of different ML algorithms in developing a soil model is presented.

2 Some typical machine learning algorithms

According to the literature investigation results, it can be seen from Fig. 2 that the number of articles regarding the application of ML for developing constitutive models has gradually increased since the end of last century, which indicates an increasing interest of the researchers to explore this new methodology for constitutive modeling of soils. Table 1 summarizes all ML based constitutive models in open literatures collected from google scholar. To construct a ML based constitutive model of soils, it can be observed that researchers have used numerous ML algorithms, such as genetic programming (GP) [45], evolutionary polynomial regression (EPR) [46-51], support vector machine (SVM) [52, 53], backpropagation neural network (BPNN) [53-74], radial basis function (RBF) neural network [74-76], recurrent neural network (RNN) [77-79], long short-term memory (LSTM) neural network [80-82] and gate recurrent unit (GRU) neural network [83], to simulate stress-strain responses of various soils including clay, sand, gravel, ballast, rockfill, frozen soil, reinforced soil and soils with various mixture such as turf and carbonate. Overall, the proportion of such eight ML algorithms used for constitutive modeling is summarized in Fig. 3.

Table 1 Summary of ML based constitutive models in literature

Fig. 2 Increasing number of papers regrading ML based constitutive models

Fig. 3 Proportion of various machine learning based model

2.1 Genetic programming

GP is a type of evolutionary algorithm, which is characterized by the symbolic optimization to search the optimum structure of formulations rather than the number optimization in general evolutionary algorithms. Binary tree is the widely used method in GP to represent the candidate structure. Fig. 4 shows the process of building a binary tree, in which the symbol and parameter values at all nodes form the equation that is assembled from the leaves to the root. Various hierarchically structured trees consist of a population. Thereafter the evolutionary process is activated that includes selection, crossing and mutation operations [84]. The symbol and parameter values at each node can be changed using crossing and mutation (see Fig. 4), and the structure with lower fitness value is selected. GP has been extensively used to investigate the relationship between independent variables and answer variables because its simple and explicit expression can provide clear explanation such as the prediction of soil physical indices [85-88]. However, GP heavily relies on stochastics procedures and operators, and the combination of the initial population is relatively numerous [89]. Such non-deterministic operations cannot ensure to find the optimum solution and develop a model with excellent generalization ability. There is no doubt that one deterministic formulation only has a unique result, which means that a GP based model cannot be used to predict multi outputs. Meanwhile the output of GP is single, which is not convenient to be further applied. Such factors lead to the few applications of GP to develop a constitutive model of soils, as presented in Fig 3 and Table 1, there is only one research work where GP is used to simulate stress-strain relationship of the sand–mica mixtures under undrained and monotonic loading condition.

Fig. 4 Framework of genetic programming

2.2 Evolutionary polynomial regression

EPR is a type of genetic programming, in which the number of transformed parameters is prescribed in advance, and it is generally used with meta-heuristic algorithms such as genetic algorithm and particle swarm optimization [90]. As shown in Fig. 5, EPR starts from generating an exponent matrix **E** using a meta-heuristic algorithm, thereafter the transformed parameters ***xt*** can be obtained by:

$$\boldsymbol{xt}_i = x_{1,i}^{\mathbf{E}} x_{2,i}^{\mathbf{E}} \ldots x_{n,i}^{\mathbf{E}} \tag{1}$$

The corresponding EPR expression can thus be formulated as:

$$\boldsymbol{y} = \sum_{i=1}^{m} c_i \boldsymbol{xt}_i + c_0 \tag{2}$$

where the constant vector $\boldsymbol{c} = [c_0, c_1, \ldots, c_m]$ can be determined by linear least square methods. The advantages and limitations of EPR are similar to GP. Most of previous ML based constitutive models of soils were developed based on EPR with the proportion of 14.29% including the investigation of soil monotonic and cyclic behaviors under drained or undrained condition. However, EPR is hardly used to predict soil stress-strain relationships in the recent research works, considering the EPR cannot perform well for complicated problems with high-dimensional data in comparison with currently proposed ML algorithms. Similar to GP, its output is also single.

Fig. 5 Framework of evolutionary polynomial regression

2.3 Support vector machine

SVM is developed based on structural risk minimization, thereby it can be used to train model with small datasets. The computational cost is related to the number of support vectors rather the number of input parameters, but the computational cost of SVM is expensive with numerous training datasets. In SVM, datasets are first mapped to a high-dimension space by a kernel function to find a linear decision surface or hyperplane to separate datasets [91]. As presented in Fig. 6, $\gamma^{(i)}$ that is orthogonal to the hyperplane is term as the geometric margin, which is used to measure the distance of a training sample to the decision boundary. The training of SVM is to find a hyperplane which can separate all datasets with a largest “gap”, that is, the minimum γ reaches the maximum value, which can be expressed by:

$$\begin{gathered} \min_{\gamma,\omega,b} \frac{1}{2}\left\|\omega\right\|^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t. } y\left(i\right)\left(\left(\omega\right)^T x^{(i)} + b\right) \geq 1-\xi_i,\ i=1,2,\ldots,m,\ \xi_i \geq 0 \end{gathered} \tag{3}$$

where m is a total of training samples. ω and b are weights and biases. ξ and C are slack and penalty parameters. SVM shows excellent performance for high-dimensional datasets, and it has been extensively used in classification problems with numerous features [92, 93]. Nevertheless, SVM based model cannot be expressed with an explicit formulation, and it also cannot be used to predict multi outputs. Therefore, the poor readability and interpretability prevent its application scopes such as the combination with numerical modeling. It can be seen from Fig. 3 that the proportion of SVM based constitutive model of soils is lowest. It was only used to model stress-strain relationship under monotonic loading and drained conditions.

Fig. 6 Framework of support vector machine

2.4 Backpropagation neural network

The most widely used ML algorithm for modeling soil stress-strain relationship is backpropagation neural network (BPNN), which is a type of feedforward neural network. It can be seen from Fig. 3 that half of ML based constitutive models are developed using BPNN. The monotonic and cyclic stress-strain relationships of clay and sand under drained or undrained condition have been sufficiently investigated using BPNN. Fig. 7 presents the architecture of BPNN with three layers, and the corresponding formulations are presented in Eqs. [4]–[5]. The input data flows from the input layer to the output layer and error is propagated from the output layer for finding a set of weights that ensure that the output value produced by the network is the same as the actual output value [94].

$$\mathbf{H} = f\left(\mathbf{W}_1\mathbf{X} + \boldsymbol{\theta}_1\right) \tag{4}$$

$$\mathbf{O} = g\left(\mathbf{W}_2\mathbf{H} + \boldsymbol{\theta}_2\right) \tag{5}$$

where $\mathbf{X}$ is the input matrix. $\mathbf{H}$ and $\mathbf{O}$ are the output of hidden and output layers, respectively. $\mathbf{W}_1$ and $\mathbf{W}_2$ are weights matrix on the connections between input and hidden layers, between hidden and output layers, respectively. $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are the bias vectors added in the hidden and output layers, respectively. f, g are activation functions in hidden and output layers, respectively. BPNN is a multilayer stack of simple modules, and a system with 5–20 nonlinear layers can implement extremely intricate functions [41, 95, 96], thereby BPNN has been extensively used for regression and classification problems in many domains [97]. However, BPNN cannot store history information, meanwhile gradients exploding or vanishing may occur as the increasing depth of network architecture, which means that it is not suitable to predict sequential data and build deep network.

Fig. 7 Framework of backpropagation neural network

2.5 Radial basis function neural network

RBF neural network is characterized by the fixed architecture (three layers including an input, a hidden and an output layers) and fast learning process. The weights connecting the input and hidden layers are randomly assigned, and the weights connecting the hidden and output layer are determined using linear least square methods, e.g. least mean square [98]. As presented in Fig. 8, given an input matrix $\mathbf{X}_{m\times n}$, the output can be obtained by:

$$\boldsymbol{y} = \mathbf{W} \times \Phi(\mathbf{X}, \boldsymbol{c}) = \sum_{i=1}^{k} \mathbf{W}\Phi\left(\left\|\mathbf{X} - c_i\right\|\right) \tag{6}$$

$$\Phi(x) = e^{-\frac{x^2}{2\sigma^2}} \tag{7}$$

where c_j is the jth center. k is the number of hidden neurons; || || denotes *Euclidean* distance. Φ is the basis function, and Gaussian formulation is commonly used as presented in Ep. [7], in which σ is the smoothing parameter. The weights and biases of RBF are obtained by using function approximation rather than the recursive iterations in BPNN, which means that the computational cost is low, and it is extremely suitable for approximation and interpolation [99, 100]. The shallow network structure of RBF means the prediction capacity of RBF neural network is poorer than BPNN, thereby it can be seen from Fig. 3 that only 7.14% of ML based constitutive models of soils were developed based on RBF. However, the exploitation of such algorithm is sufficient including the modeling of monotonic and cyclic stress-strain relationships of clay and sand under drained or undrained condition.

Fig. 8 Framework of radial basis function neural network

2.6 Recurrent neural network

The ML algorithms mentioned above cannot account for soil loading history themselves, that is, the model output only depends on the current input of stress or strain increment. RNN is characterized by a cyclic connection topology, as presented in Fig. 9. In this way, the hidden layer output at the time step t is not only affected by the input, but also relates to the hidden output at the (t–1)th step, which can be expressed by:

$$\mathbf{H}^t = f\left(\mathbf{W}_1 \boldsymbol{x}^t + \mathbf{U}\mathbf{H}^{t-1} + \boldsymbol{b}_1\right) \tag{8}$$

where $\mathbf{U}$ is matrix connecting hidden layers at adjacent steps. The calculation of output is similar to the Eq. [5].

Hence, the history information is stored and it is applied to predict the next status. Such history-dependent characteristic makes RNN applicable to investigate problems with sequential datasets, such as language transformation, speech recognition [78, 101]. Soil response is undoubtedly affected by the loading history, and the stress-strain datasets have sequential characteristic. RNN has been applied to investigate soil behavior under monotonic loading and different drained conditions, and been gradually gained attention in the development of ML based constitutive model of soils (8.33% as shown in Fig. 3).

Fig. 9 Framework of recurrent neural network

2.7 Long short-term memory neural network

The training of conventional RNN has two obvious issues: *i*) the back-propagated gradients either grow or shrink at each time step, resulting in exploding or vanishing gradients [95], *ii*) the learning efficiency of the hidden layers in the front of the architecture is poorer than the later hidden layers, which means RNN only stores the short-term history. To overcome such problems, LSTM neural network is developed, in which a memory cell is added in the architecture of LSTM, as presented in Fig. 10. Such memory cell can store information over extended time intervals and handle long-time-lag tasks [102] by using a novel entity termed as “gate”, that is, forget, input and output gates. The outputs of such gates at the *t*th step can be obtained by:

$$\boldsymbol{f}^t = \sigma\left(\mathbf{W}_f \boldsymbol{x}^t + \mathbf{U}_f \boldsymbol{h}^{t-1} + \boldsymbol{b}_f\right) \tag{9}$$

$$\boldsymbol{i}^t = \sigma\left(\mathbf{W}_i \boldsymbol{x}^t + \mathbf{U}_i \boldsymbol{h}^{t-1} + \boldsymbol{b}_i\right) \tag{10}$$

$$\boldsymbol{o}^t = \sigma\left(\mathbf{W}_o \boldsymbol{x}^t + \mathbf{U}_o \boldsymbol{h}^{t-1} + \boldsymbol{b}_o\right) \tag{11}$$

where σ is the *sigmoid* function. In the forget gate, σ = 1 and 0 represent all information is maintained or discarded, respectively. In the input gate, σ = 1 and 0 represent all information is selected or discarded, respectively. The memory cell and hidden layer states at the *t*th current step are updated using:

$$\tilde{\boldsymbol{c}}^t = tanh\left(\mathbf{W}_c \boldsymbol{x}^t + \mathbf{U}_c \boldsymbol{h}^{t-1} + \boldsymbol{b}_c\right) \tag{12}$$

$$\boldsymbol{c}^t = \boldsymbol{f}^t \odot \boldsymbol{c}^{t-1} + \boldsymbol{i}^t \odot \tilde{\boldsymbol{c}}^t \tag{13}$$

$$\boldsymbol{h}^t = \boldsymbol{o}^t \odot tanh\left(\boldsymbol{c}^t\right) \tag{14}$$

where *tanh* is the activation function; $\odot$ denotes elementwise product; $\boldsymbol{c}^t$ stores the long-term memory. $\boldsymbol{f}^t \odot \boldsymbol{c}^{t-1}$ represents the discarded information; $\boldsymbol{i}^t \odot \tilde{\boldsymbol{c}}^t$ represents the newly selected information. The update of memory cell status with an addition format can avoid the gradients vanishing and exploding. Because of the effectiveness of LSTM, increasing researchers have used it to model soil behavior in the recent years [80, 81, 101]. Zhang et al. [80] have successfully used it to simulate cyclic behaviors of granular materials under both drained and undrained conditions.

Fig. 10 Framework of memory cell of LSTM

2.8 Gated recurrent unit neural network

GRU is a variant of LSTM and it has fewer weights and biases than LSTM as presented in Fig. 11, because the memory cell of GRU only has two gates, that is, update (***z***) and reset (***r***) gates, respectively [103]. The output of such two gates can be obtained by:

$$\boldsymbol{r}^t = \sigma\left(\mathbf{W}_r \boldsymbol{x}^t + \mathbf{U}_r \boldsymbol{h}^{t-1} + \boldsymbol{b}_r\right) \tag{15}$$

$$\boldsymbol{z}^t = \sigma\left(\mathbf{W}_z \boldsymbol{x}^t + \mathbf{U}_z \boldsymbol{h}^{t-1} + \boldsymbol{b}_z\right) \tag{16}$$

Herein, reset gate decides which part of previous hidden information $\boldsymbol{h}_{t-1}$ can be discarded, thereby the current candidate hidden state $\boldsymbol{c}_t$ can be expressed by:

$$\tilde{\boldsymbol{h}}^t = tanh\left[\mathbf{W}\boldsymbol{x}^t + \mathbf{U}\left(\boldsymbol{r}^t \odot \boldsymbol{h}^{t-1}\right)\right] \tag{17}$$

The update gate decides which part of the current hidden state $\boldsymbol{h}_t$ need to be updated through the candidate hidden state $\boldsymbol{c}_t$, thereby $\boldsymbol{h}_t$ can be obtained by:

$$\boldsymbol{h}^t = \boldsymbol{z}^t \boldsymbol{h}^t + \left(1 - \boldsymbol{z}^t\right)\tilde{\boldsymbol{h}}^t \tag{18}$$

where $(1–\boldsymbol{z}_t)$ indicates the information inherits from the previous hidden state.

Similar to the LSTM, GRU has also been extensively used in sequential issues [104]. Both LSTM and

GRU have presented successful application in many domains, and there is no deterministic statement to explain which algorithm is suitable to certain specific problems. Recently, [83] have also used GRU to investigate traction-separation relationship of granular material.

Fig. 11 Framework of memory cell of GRU

2.9 Summary and suggestions

There are typically eight ML algorithms used to develop constitutive models of soils, i.e. GP, EPR, SVM, BPNN, RBF, RNN, LSTM and GRU. The main advantages and limitations of such ML algorithms as mentioned above are summarized in Table 2. It can be seen that neural networks have been becoming the mainstream to develop constitutive models of soils, because such algorithms have excellent generalization ability and can predict multi outputs simultaneously. The multi outputs prediction is important, because it provides a basis to integrate with numerical analysis codes to ensure the application of ML based model in engineering practice. RNNs, particularly the LSTM and its variants such as GRU that can eliminate the problems existing in conventional RNNs, have increasingly been introduced to develop constitutive models of soils, because such algorithms based model can account for the loading history. Overall, currently adopted ML algorithms in developing soil models tend to lag behind the development of the ML domain. Special attention should be paid on timely introduction of advanced and efficient algorithms.

Table 2 Main characteristics of typically adopted ML algorithms for developing constitutive models of soils

3 Procedure of proposing a ML based constitutive model

3.1 Selection of data source

3.1.1 Experimental data

Sufficient data are the basis to develop a ML based constitutive model. It can be seen from Table 1 that most of ML based models are constructed using experimental data. The studied soils involve clay, sand, gravel, ballast, rockfill, frozen soil, reinforced soil and soils with various mixture such as turf and carbonate. It should be noted that current research works merely focus on the modeling of soil shearing behavior under triaxial [54, 55], direct shearing [68], simple shearing [80, 101], tension-shear [83] and unconfined compression shearing test [59], and the stress history prior to shearing has not been considered. It can be obtained from Table 1 that shearing behavior of soils under drained or undrained triaxial tests has been largely investigated using the ML based constitutive models comparing to others.

Learning from raw experimental data ensures ML algorithms capture the essential stress-strain relationship, because the mechanical responses of soils are included in such data. Nevertheless, for ML based model as data-driven model, in previous studies the type of tests and number of experimental data adopted for training are still limited. Moreover, the performance of current ML based models has been merely evaluated on several experimental tests of a given soil. Such factors lead to the robustness of current ML models hard to be guaranteed. To this end, the reliability of such models developed based on a given soil for modeling the stress-strain relationship of other soils has not been investigated.

3.1.2 Synthetic data

The number of synthetic datasets is infinite, and they can eliminate the interference of experimental and measurement errors. As presented in Table 1, most of synthetic datasets were derived from conventional physics-driven constitutive models, such as using the simple monotonic Konder's expression [105], the Modified Cam Clay (MCC) [106, 107], the hardening soil (HS) model [108], the two-surface model in multilaminate framework (TDH) [109] and the endochronic model [110]. Furthermore, various numerical modeling methods such as discrete element method were also used to generate synthetic datasets [73].

The performance of ML based model developed using sufficient datasets is stable and robust, thereby synthetic datasets are suitable to explore the training strategy and framework of the ML based constitutive models. For instance, Sidarta and Ghaboussi [56] utilized synthetic datasets to propose an auto-progressive method to describe stress-strain relationship of sand under monotonic loading. Basheer [59] presented and cross-compared several methodologies for effectiveness in approximating a theoretical hysteresis model resembling stress-strain behavior, and a true sequential dynamic mapping method was recommended to simulate cyclic behavior of soils. Considering the synthetic datasets are generated from theoretical formulations, thereby such ML based models cannot show better performance and dig deeper mechanism than the physics-based constitutive models used for data creation.

3.2 Training framework of ML based soil model

The framework of a machine learning based model involves two important factors. The first is to determine the composition of input and output parameters that are known as feature selection, and the second is to determine its topology.

3.2.1 Feature selection

Feature selection refers to determine the best set of features with maximum information to maximize the model accuracy [111, 112]. If the selected features include sufficient information regarding the soil behavior, then the trained ML based model would learn the soil behavior to qualify as a constitutive model. Such trained model not only is able to reproduce the experiments trained on, but also has capacity of approximating the results of other unexposed experiments. The selected features can be categorized into three groups: *i*) physical properties, such as relative density D_r and initial void ratio e_0; *ii*) state parameters, such as stresses and/or strains, and *iii*) history parameters, such as stresses and/or strains and/or other variables like breakage index if the grain crushing is accounted for example. Physical property parameters (*pp*) are used to describe the intrinsic characteristics of studied soils. State parameters (*s*) focus on controlling the evolution of stress-strain development. In detail, state parameters are divided into static (*ss*) and dynamic (*sd*) parameters. *ss* parameters represent the unchanged or known attributes of stress-strain responses, e.g. amplitude of the applied shear stress and encoding for representing drained or undrained condition. *sd* parameters directly affect the stress-strain evolution, thereby they are updated in real time, e.g. p' and q. History parameters (*o*) are the model outputs at the previous step. The application of such parameters aims to account for the effects of stress-strain history to the current stress-strain development. The selected features in current ML based constitutive models of soils involve four combinations of such three types of parameters, as follows:

(1) state parameters [74-76], such as p' and q as inputs [75];

(2) physical property with state parameters [45, 49, 53, 66, 69, 71, 81], such as percentage of mica p_m and

axial strain ε_a as inputs [45];

(3) state parameters with history information [47, 48, 63, 64, 113], such as p', ε_a, q, deviatoric strain increment $\Delta\varepsilon_q$, and volumetric strain ε_v as inputs (q and ε_v are the model outputs) [48];

(4) physical property with state parameters and history information [50-52, 54, 55, 57-62, 67, 68, 70, 72, 77-80], such as D_r, ε_a, effective confining stress σ'_3, q and pore-water pressure u as inputs (q and u are also the model outputs).

Herein, regarding the first and third combinations, such research works focus on the investigation of stress-strain relationships of a given soil with fixed physical properties. The start points of such research works are the validation of the performance of ML based models and the exploration of a reasonable training strategy. The most representative research was implemented by [55], in which a nested modularity of the history stress-strain information was added to guide how to increase the number of neurons in the input and hidden layers of BPNN. Such framework has been applied to constitutive modeling of sands [63-65], structures [114, 115] and composite materials [116]. However, the application scopes of such models are still limited, because they are not useful once the studied material is changed. Therefore, the research works in the second and fourth combinations have preoccupied with overcoming such limitations by adding the physical properties as additional information to the input parameters. The application scopes of such ML based constitutive models can thus be enlarged, but the corresponding model complexity and datasets size also increase, and the requirement for the performance of ML algorithms is also high. In general, the fourth combination including three types of features should be recommended during the selection of input parameters, because such combination can ensure the application scope and accuracy of ML based constitutive models.

3.2.2 Topology of ML based model

The topology of current ML based constitutive models of soils can be categorized into two groups: forward and feedback topologies. The forward topology means the data flow from the input layer to the output layer, as presented in Fig. 12, thereby it cannot account for the stress-strain history information. The feedback topology means the model outputs as the feedback to be used as inputs of model, as shown in Fig. 13. The use of history information as inputs is beneficial to enhance predictability. It should be noted that an internal connection may exist between dynamic state parameters. For example, if the selected features include strain and strain increments, the value of strain at each step needs to be updated beforehand using the strain increment. The outputs of such two topologies at the *t*th step can be expressed by:

$$\left(o_1^t,...,o_l^t\right)=f\left(pp_1,pp_i,...,ss_1,...,ss_j,sd_1^t,sd_2^t,...,sd_k^t\right) \tag{19}$$

$$\left(o_1^t,...,o_l^t\right)=f\left(pp_1,pp_i,...,ss_1,...,ss_j,sd_1^{t-1},sd_2^t,...,sd_k^t,o_1^{t-h},...,o_l^{t-h},o_1^{t-1},...,o_l^{t-1}\right) \tag{20}$$

where *i*, *j*, *k*, *l*, *h* are the number of parameters regarding physical property, static state, dynamic state, output and recursive steps, respectively.

In general, it can be stated that the model with feedback topology and three types of input parameters can present excellent performance. In particular, Ghaboussi and Sidarta [55] pointed out the model can be represented more accurately as more history information are included. However, the integration of more history points would definitely lead to the increasing complexity of the ML based model. The tradeoff between complexity and accuracy has to be well treated in the development of ML based constitutive models of soils. The establishment of a ML based model should include three types of input parameters, in which the history information stepwise increases until the optimum number of recursive steps is found.

Fig. 12 Forward topology for training constitutive model of soil

Fig. 13 Feedback topology for training constitutive model of soil

3.3 Training strategy

3.3.1 Determination of hyper-parameters

Training of ML based model starts from selecting hyper-parameters, and the performance of ML based model is primarily affected by the hyper-parameters. ANNs have more hyper-parameters in comparison with other ML algorithms. To obtain a well-performed ANN based model, current research works focused on the optimization of architecture, that is, the number of hidden layers and the number of hidden neurons. Table 3 summarizes the ultimate architecture used in ANN based constitutive models. The commonly used method is trial and error, in which the number of hidden layers and neurons is adjusted by using domain knowledge and it actually relies on the user's subjective experience heavily. Ghaboussi et al. [116] proposed an auto-progressive method to guide the adjustment of hidden neurons for developing an ANN based constitutive model. It can be observed that the deep ANN is not used in current ANN based models, in which the maximum number of hidden layers is only three and the number of hidden neurons ranges from 4 to 90. The hyper-parameter learning rate used in current ANN based constitutive models tended to be set as the default value 0.01 or 0.001, and some modified adaptive learning rate strategies [117] have not been used. It can be observed from Table 4 that the hyper-parameters of SVM (e.g. slack and penalty parameters), EPR (e.g. number of transformed terms) and GP (e.g. mutation rate) based constitutive models are also determined by trial and error method. Other algorithms for determining the hyper-parameters such as meta-heuristic algorithm [118-120] have rarely been applied.

Table 3 Summary of architectures and learning strategies used in ANN based models

3.3.2 Selection of activation functions

It is obvious that the currently used methods to develop ANN based constitutive model is out of style. Some advanced and effective modules and optimization algorithms in the ML domain have not been applied to develop an ANN based constitutive model, which may be attributable to the rapid development of the neural network in the field of artificial intelligence. For instance, the activation functions in most of ANN based models are *sigmoid* and *tanh* (see Eq. [20], respectively), as presented in Fig. 14(a). The derivative of the activation function is used during the error back propagation. The derivative functions of *sigmoid* and *tanh* are presented in Eq. [21], and the corresponding graph is presented in Fig. 14(b). It can be observed that the derivative value is close to zero that is gradient vanishing when the absolute values of inputs are larger than 4, which means that the weights and biases of ANN cannot be updated effectively when the values of input parameters are away from zero. Therefore, *sigmoid* and *tanh* activation functions suffer from saturation and limited sensitivity. Rectified linear unit (*ReLU*) activation function is thereafter proposed to overcome such issues (see Eq. [20]). In Fig. 14(b), it can be observed that the derivative is one when the values of input parameters are larger than zero, thereby the gradients vanishing problem is resolved. However, the derivative is zero when the values of input parameters are less than zero, which means some information will be missed. *Leaky ReLU* [121] and *ELU* as variants of *ReLU* were thereafter proposed (see Eq. [20]), and it can be seen from Fig. 14(b) that the value of the gradient is not stuck at zero. However, such effective activation functions *ReLU* and its variants have rarely been used to build ML based constitutive model, and it was only used in the LSTM based constitutive model proposed by Zhang et al. [80]. Furthermore, some newly developed activation functions, such as Swish [122] and Mish [123], also perform well in other fields (such as natural language processing (NLP) and image recognition (IR)), which can be tentatively adopted in the constitutive modeling of soils by ML algorithms.

$$\begin{cases} sigmoid\left(x\right)=\dfrac{1}{1+e^{-x}} \\ tanh\left(x\right)=\dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \\ ReLU\left(x\right)=\begin{cases} x, \ x>0 \\ 0, \ x\leq 0 \end{cases} \\ \text{Leaky } ReLU\left(x\right)=\begin{cases} x, \ x>0 \\ \alpha x, \ x\leq 0 \end{cases} \\ ELU\left(x\right)=\begin{cases} x, \ x>0 \\ \alpha\left(e^{x}-1\right), \ x\leq 0 \end{cases} \end{cases} \tag{21}$$

$$\begin{cases} sigmoid'\left(x\right)=\dfrac{e^{-x}}{\left(1+e^{-x}\right)^{2}} \\ tanh'\left(x\right)=1-\dfrac{\left(e^{x}-e^{-x}\right)^{2}}{\left(e^{x}+e^{-x}\right)^{2}} \\ ReLU'\left(x\right)=\begin{cases} 1, \ x>0 \\ 0, \ x\leq 0 \end{cases} \\ \text{Leaky } ReLU'\left(x\right)=\begin{cases} 1, \ \ x>0 \\ \alpha, \ x\leq 0 \end{cases} \\ ELU'\left(x\right)=\begin{cases} 0, \ x>0 \\ \alpha e^{x}, \ x\leq 0 \end{cases} \end{cases} \tag{22}$$

Fig. 14 Activation functions: (a) original formulation; (b) derivative

3.3.3 Selection of optimization algorithm

Regarding the optimization algorithm, it can be seen from Table 3 that Stochastic Gradient Descend (SGD), Backpropagation training [94] (BProp), Quick Propagation training (QProp) [124], Resilient Propagation training (RProp) [125], Levenberg–Marquardt (LM) [126, 127], Scaled Conjugate Gradient (SCG) [128], Generalized Delta Rule (GDR) [129] and adaptive learning rate Delta Bar method (DB) [117] have been used to update the weights and biases of ANN. Such learning strategies were proposed in the end of last century, which are known empirically to find poor solutions for networks [130]. They are easily trapped into local optima and the computational cost is expensive. However, recently proposed effective learning strategies have rarely been applied to train ANN based constitutive models such as AdaGrad [131] which works well with sparse gradients, RMSProp [132] which works well in on-line and non-stationary settings, *Adam* [133] that integrates the advantages of AdaGrad and RMSProp, and AdaMax [133] that is a variant of *Adam*. Zhang et al. [80] have noticed the advancement of such learning strategies, and first introduced *Adam* to train LSTM based constitutive model to simulate soil cyclic behavior. Table 4 presents the learning strategy used in SVM, EPR and GP based constitutive models. The learning strategy used in SVM and GP has not been clearly explained in current research works. In EPR, it can be observed that genetic algorithm (GA), as a meta-heuristic algorithm, was the primary method to optimize the exponent matrix of EPR. In reality, meta-heuristic algorithms such as evolutionary algorithms and reinforcement learning based optimizer [134], have also been used to optimize the weights and biases of ANNs in other domains [93, 135], but they have not been adopted in ANN based soil modeling.

Table 4 Summary of learning strategies used in SVM, EPR and GP based models

3.3.4 Selection of loss function

Regarding the loss functions, it can be seen from Tables 2 and 3 that absolute and relative error indicators are the two main groups used in the ML based constitutive models. The commonly used absolute error indicators cover mean absolute error (MAE), mean square error (MSE), sum of square error (SSE) and mean sum of square error (MSSE). Such absolute error based loss functions put emphasis on shrinking the difference of large output value and sacrificing the accuracy in the prediction of small output value. The performance of some relative error based loss functions such as relative mean squared error (REMSE) and mean absolute percentage error (MAPE) may perform better in the prediction of initial stress-strain relationship. However, the training process with such relative error based loss functions are hard to converge, because relative error is sensitive to the denominator value. The loss value can be easily perturbed if the denominator value is low and be useless with the denominator of zero [136]. The expressions of such loss functions are presented as followings:

$$\begin{cases} \mathrm{MAE} = \dfrac{1}{n}\sum_{i=1}^{n}\left|y_i^p - y_i^m\right| \\ \mathrm{MSE} = \dfrac{1}{n}\sum_{i=1}^{n}\left(y_i^p - y_i^m\right) \\ \mathrm{SSE} = \sum_{i=1}^{n}\left(y_i^p - y_i^m\right)^2 \\ \mathrm{MSSE} = \dfrac{1}{n}\sum_{i=1}^{n}\left(y_i^p - y_i^m\right)^2 \\ \mathrm{REMSE} = \dfrac{1}{n}\sum_{i=1}^{n}\left(\dfrac{y_i^p - y_i^m}{y_i^m}\right)^2 \\ \mathrm{MAPE} = \dfrac{1}{n}\sum_{i=1}^{n}\left|\dfrac{y_i^p - y_i^m}{y_i^m}\right| \times 100\% \end{cases} \tag{23}$$

where y_i^m and y_i^p are the actual and predicted values of selected outputs, respectively; n is total number of rows of datasets.

3.3.5 Methods for preventing overfitting

From the perspective of Tables 2 and 3, currently ML based constitutive models hardly use methods to prevent overfitting problem and thus enhance the model robustness. The commonly used methods for preventing overfitting problem in ML based modeling involve: (1) weight decay and (2) dropout. Weight decay, such as L_1 and L_2, constrains the capacity of model by adding penalty terms to the original objective function. The key idea of dropout is to randomly drop neurons (along with their connections) from the neural network during training, which can prevent neurons from co-adapting too much [137]. Weight decay can be integrated with ANN, GP and EPR algorithms [138, 139], whereas dropout is tailored to ANNs [140]. In recent years, Lin et al. [74] applied weight decay method to prevent overfitting of ML based constitutive model, in addition, Wang et al. [82] and Zhang et al. [80] introduced dropout method [137] to further prevent overfitting, and the *k*-fold cross-validation was applied to improve the model robustness. Overall, currently used modules (e.g. activation functions) and learning strategies (e.g. optimizer, overfitting prevention and robustness improvement) for constructing ML based constitutive models extremely lag the development in the ML domain. Therefore, introduction of effective training methods in the ML domain is necessary to develop a more robust ML based constitutive model.

3.4 Summary and suggestions

Data source is the basis to develop ML based constitutive models. The performance of current ML based models developed based on experimental datasets were always evaluated on several experimental tests with limited stress paths of a given soil, thereby it is hard to guarantee the robustness of such models and the feasibility of applying such models to simulate stress-strain relationship of other soils. Synthetic datasets are derived from theoretical formulations, thereby ML based models developed based on synthetic datasets cannot show better performance and dig deeper mechanism than the physical models. To make use of the advantages of experimental and synthetic datasets, a tradeoff is to use synthetic datasets to explore and develop a general framework of establishing a ML based model, thereafter such framework is applied to the experimental datasets to assist to discover the potential mechanism of soil behavior and enhance model robustness.

The framework of ML based constitutive model depends on the selected features and the topology, which reveals the operation mechanism of such ML based models. The applied features in current ML based constitutive models of soils involve four combinations of three types of parameters, that is, 1) state parameters; 2) physical properties with state parameters; 3) state parameters with history information; 4) physical properties with state parameters and history information. In general, the fourth combination including three types of features is recommended during the selection of input parameters, because such combination can ensure the application scope and accuracy of ML based constitutive models.

The topology of current ML based constitutive models of soils can be categorized into two groups: forward and feedback topology. The feedback topology has been gradually acknowledged, because it can generate more accurate model than the forward topology as more history information is included. However, the integration of more history points would definitely lead to increasing complexity of the ML based model. The tradeoff between complexity and accuracy has to be well maintained in the development of ML based constitutive modeling. Therefore, the establishment of a ML based model is recommended to stepwise increase the history information until the optimum number of recursive steps is found.

Training strategy is the key factor to affect the training process and the performance of ML based constitutive models. Currently adopted modules (e.g. activation functions) and learning strategies (e.g. optimizers) for constructing a ML based constitutive model extremely lag the development in the ML domain. Such out-of-style methods and learning strategies reduce the learning efficiency and the training process is easily trapped into the local optima. Moreover, current ML based models rarely use methods to avoid overfitting problem and enhance model robustness. The introduction of effective training methods in the ML domain is recommended to develop a more robust ML based constitutive model.

4 Estimation of ML based model performance

From the perspective of modeling results presented in the current research works, the stress-strain responses can be accurately captured by ML based models. However, the reliability and robustness of such models have not been comprehensively discussed. The generalization ability and application scopes of such models deserve to be deeply investigated.

4.1 Generalization ability

Generalization ability refers to models' ability to produce sensible answers on previously unexposed data [95]. It is important to distinguish two cases as follows during the evaluation of generalization ability [141]: (1) Interpolation: the training set is expected to be fully representative of input parameters during application, and the ranges of input parameters in the testing set totally fall into the training set; (2) Extrapolation: the training set can only represent certain features of datasets, and the values of input parameters in the testing set outside the ranges of training set.

In other words, interpolation predictability is inherent to the environment in which the system is used, whereas extrapolation means the exploration regions where no samples have reached. It can be seen from Fig. 15 that approximately half of research works (44.68%) merely examine the interpolation ability of ML based constitutive models, and only 2.13% of models examine their extrapolation ability. Moreover, 4.26% of research works only exhibit the predicted results on the training set, which actually cannot be used to test the performance of ML based models, and 38.3% of research works do not clearly explain which predictability is tested. Both interpolation and extrapolation predictability have been investigated in 10.64% of research works. It is clear that extrapolation is the more pervasive task, and in the prediction of soil stress-strain response it would be preferable if extrapolation could be performed. Therefore, in addition to examine the interpolation predictability of ML based constitutive models, it is reasonable and highly recommended to enhance the examination of extrapolation predictability.

Fig. 15 Proportion of testing set type used in the training of constitutive model of soil

4.2 Application scope

The advanced physics-based constitutive models of soils have been able to simulate various soil mechanical behaviors such as compression [142], shear [143, 144], influence of intermediate principal stress [145], inherent and induced anisotropy [23, 146], non-coaxiality[147], small strain stiffness [148, 149], cyclic effect [30, 150], time-dependency [151-153], temperature effects [154], soil structure and destructuration [155]. From the perspective of input parameters used in current research works, such ML based models were merely used to simulate some of these features with different values of some commonly used soil physical properties such as relative density, overconsolidation ratio, particle size distribution, individual fraction of mixed materials, but the range of such physical properties is small. Meanwhile such research works rarely use the ML based constitutive models to analysis the effects of such physical properties on the mechanical behaviors of soils. The reliability of such ML based models to investigate mechanical behaviors of soils has not been sufficiently discussed. Furthermore, almost all of these ML based models were independently developed based on drained or undrained datasets except the model proposed by Ghaboussi and Sidarta [55] and Zhang et al. [80] (see Table 1), which means that such models cannot simultaneously describe soil behaviors under both drained and undrained conditions. Such models are thus limited to simulate the stress-strain behavior of a given soil with single stress path (e.g. drained or undrained stress path). The most recent research work conducted by Zhang et al. [80] started to focus on modeling the cyclic behaviors of sand under both drained and undrained conditions, i.e., the cyclic mobility mechanism, the degradation of effective stress and large deformation under the undrained condition, and shear strain accumulation and densification under the drained condition. Overall, most of current ML based models cannot comprehensively simulate mechanical behaviors of a soil sample. As a result, the further application by the models combined with numerical platform is not realistic. This is also the reason that up to now, it lacks the application of ML based constitutive models integrated in numerical platforms to directly analyze practical engineering projects.

4.3 Example of sand model using different ML algorithms

To compare the performance of different ML algorithms and apply novel training strategies, the widely adopted four ML algorithms including BPNN, RBF, LSTM, and GRU are used to simulate soil behaviors. The reason for the selection of such four algorithms is that they can predict multi outputs together, which ensures the deep application of such algorithms based constitutive models to simulate complex soil behaviors and to be integrated in numerical codes. In contrast, however, GP, EPR and SVM only predict one variable at one time. In addition, a variant of LSTM termed as bidirectional LSTM (BiLSTM) which has not been used to investigate soil behaviors is also introduced in study. Such algorithm utilizes both the positive and reverse sequential information by concatenating the hidden-layer outputs of each model. The detailed information regarding BiLSTM can refer to Graves et al. [156], which is not introduced in this study for brevity.

The datasets used in this study are collected from 27 triaxial consolidation drained tests (three initial void ratio e_0 with nine consolidation confining pressure σ_3) on the Baskarp sand conducted by Ibsen and Bødker [157]. Herein, 23 experiments (about 80% of total 27 tests) with the confining stresses of 10, 20, 40, 80, 160, 320, 640 kPa are used to train ML based models. The remaining 4 experiments with confining stresses of 5 and 800 kPa (for extrapolation), 20 and 160 kPa (for interpolation) are used to examine the developed ML based models. The feedback framework is used as shown:

$$\left[p'^{t}, q^{t}, e^{t}\right] = f\left(p'^{t-1}, q^{t-1}, e^{t-1}, \varepsilon_1^{t}, d\varepsilon_1^{t}, \sigma_3', e_0\right) \tag{24}$$

where p'^{t}, q^{t}, e^{t}, ε_1^{t}, $d\varepsilon_1^{t}$ are the mean effective stress, deviatoric stress, void ratio, axial strain, axial strain increment at the tth step, respectively; p'^{t-1}, q^{t-1} and e^{t-1} are the mean effective stress, deviatoric stress and void ratio at the (t–1)th step, thereby the outputs of the LSTM based model at the (t–1)th step are used as the inputs at the tth step; σ'_3 is the confining stress, and e_0 is the initial void ratio.

The optimum configurations of five ML based models are determined by trial and error method, and the detailed process for determining such configurations are not presented for brevity. It should be noted that the configurations of each model presented in Table 5 are optimum for comprehensively comparing the performance of different ML algorithms on the modeling of soil behaviors. Therefore, it can be seen from Table 5 that the configurations of each optimum ML based model are different. To quantitively evaluate the performance of different ML based models, absolute and relative error evaluation indicators MAE and MAPE (see Eq. [23]) are used. The values of indicators are summarized in Table 6. Regarding the training set, it can be observed that the training performance of LSTM, GRU and BiLSTM outperforms BPNN and RBF. Such three time series prediction algorithms show better performance on the learning of soil behaviors. Regarding the testing set, it can be observed that the values of MAE and MAPE generated by BiLSTM are lowest at all cases. BiLSTM reduces the MAE and MAPE values in comparison with LSTM, whereas GRU increases the values of such two indicators. BiLSTM can learn the forward and backward data information, thereby the enhancement in the interaction of data is indeed beneficial to learn the complex soil behaviors from the raw data. BPNN and RBF produce much larger MAE and MAPE values on the testing sets, which indicates the generalization ability of such algorithms is much poorer than LSTM and its variants.

Table 5 Configurations of different ML based models

Table 6 Values of indicators generated by different ML based models

To clearly reveal the reliability of such ML based models, Fig. 16 presents the predicted evolution of stress-strain relationship using five ML based constitutive models. The results shown in Figs. 16(a) and (b) can reflect the interpolation prediction capacity of ML based model, while the results shown in Figs. 16(c) and (d) are used to evaluate their extrapolation prediction capacity. It is clear that BPNN and RBF cannot accurately capture soil stress-strain relationships, showing much poorer performance than time series prediction algorithm LSTM and its variants GRU and BiLSTM. The results in Figs. 16 (c) and (d) indicate that BPNN exhibits poor extrapolation prediction capacity on extrapolated data, resulting in the prediction error accumulating as the increasing strain. The predicted p', q and e at all four cases using RBF severely deviate from the measured results, which indicates the function approximation method used in RBF limits its capacity of learning mechanical behaviors of soils. The predicted p', q and e using LSTM for both interpolation and extrapolation experiments show excellent agreement with measured results. GRU has less weights and biases than LSTM, which more or less reduces the time series prediction capacity and further leads to the slightly poorer performance particularly on the extrapolation experiments in capturing soil behavior. It can be observed from Fig. 16 that the performance of BiLSTM is slightly better than LSTM on both extrapolation and extrapolation testing sets. The results indicate BiLSTM based model outperforms the remaining models. It can be seen from Figs. 16 (c) and (d) that BiLSTM based model can accurately capture shearing-induced volumetric contraction and dilation of experiments, which sufficiently indicates the reliability of such model.

Fig. 16 Predicted stress-strain responses using four ML algorithms: (a) e_0 = 0.696, σ'_3 = 19.9 kPa; (b) e_0 = 0.695 σ'_3 = 160 kPa; (c) e_0 = 0.852, σ'_3 = 5 kPa; (d) e_0 = 0.852, σ'_3 = 800 kPa

4.4 Summary and suggestions

Generalization ability which represents the predictability on the unseen data needs to be carefully examined before the further application of ML based constitutive model. It is clear that extrapolation is the more pervasive task for the prediction of soil stress-strain relationship, but most of current research works merely investigated the interpolation predictability of the ML based constitutive model. Therefore, in addition to examine the interpolation predictability, it is reasonable and recommended to enhance the examination of extrapolation predictability of the ML based constitutive model.

The application scopes of ML based constitutive models deserve to be deeply explored to guarantee its research significance. The reliability of current ML based models to investigate mechanical behaviors has not been sufficiently discussed. Furthermore, almost all of these ML based models were independently developed based on drained or undrained datasets, which means that such models are limited to modeling simple stress-strain behavior for a given soil with a fixed drained condition. Overall, most of current ML based model cannot comprehensively simulate mechanical behaviors of a soil sample, and the further application with the combination of numerical modeling is not realistic. To prompt the application of the ML based constitutive model and proves its significance in engineering practice, the integration with numerical modeling deserves to be conducted.

5 Conclusions

This study comprehensively reviewed the application of ML algorithms in the development of constitutive modeling and compared the performance of different ML algorithms. First, the main characteristics and the limitations of eight typically adopted ML algorithms were summarized and compared. Thereafter the methodology of developing ML based soil models was reviewed from six aspects: applied ML algorithms, data source, framework of the ML based model, training strategy, generalization ability and application scope. Finally, a comparison of five typical ML algorithms that can predict multi outputs together on the development of soil model from a series of experiments on sand was presented in terms of the prediction accuracy and generalization ability. The main conclusions are made as follows:

(1) Long short-term memory (LSTM) neural network and its variants such as gate recurrent unit (GRU)

and bidirectional LSTM (BiLSTM) are suitable to be adopted in constitutive modeling, but they have rarely been used to develop constitutive models of soils. Special attention should be paid on the timely introduction of advanced and efficient algorithms.

(2) The ML based constitutive modeling is recommended to be first developed based on synthetic datasets to explore a general framework, thereafter such framework is applied to the experimental datasets to assist to discover the potential mechanism of soil behavior and enhance model robustness.

(3) Input parameters including physical properties, state and stress-strain history information parameters can ensure the application scope and accuracy of ML based constitutive models, thereby the feedback topology (model outputs at previous step are used as the input parameters at current step) is suitable. The history information can stepwisely increase until the optimum number of recursive steps is found.

(4) Currently used modules (e.g. activation functions) and learning strategies (e.g. optimizers) for constructing ML based constitutive models extremely lag the development in the ML domain. Meanwhile such models hardly use methods to avoid overfitting problem and enhance model robustness. The introduction of effective training methods in the ML domain is necessary to develop a more robust ML based constitutive model.

(5) The extrapolation predictability is the more pervasive task for the prediction of soil stress-strain relationship. It is reasonable and recommended to enhance the examination of extrapolation predictability for ML based constitutive models.

(6) Current ML based models cannot comprehensively simulate mechanical behaviors of a soil with

complex stress paths, thereby the further application using numerical codes with model integration has not been conducted. To prompt the application of ML based constitutive models and prove its significance in engineering practice, the development of ML based models valid for complex stress paths and the implementation of such developed ML based models in numerical modeling codes is deserved.

Acknowledgements

This research was financially supported by the RIF project of Research Grants Council (RGC) of Hong Kong Special Administrative Region Government (HKSARG) of China (Grant No.: PolyU R5037-18F).

Credit author statement

Pin Zhang: Conceptualization, Methodology, Formal analysis, Writing-Review and Original Draft.

Zhen-Yu Yin: Supervision, Investigation, Methodology, Visualization, Writing-Review and Editing.

Yin-Fu Jin: Validation, Visualization, Writing-Review and Editing.

Conflicts of Interest

The authors declare that the work described has not been published before; that it is not under consideration for publication anywhere else; that its publication has been approved by all co-authors; that there is no conflict of interest regarding the publication of this article.

References

[1] 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

[2] Kang X, Xia Z, Chen R, Ge L, Liu X (2019) The critical state and steady state of sand: A literature review. Mar Georesour Geotec 37(9): 1105-1118

[3] 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

[4] 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

[5] Kang X, Xia Z, Chen RP (2019) Measurement and correlations of K_0 and V_s anisotropy of granular soils. Proceedings of the Institution of Civil Engineers - Geotechnical Engineering: 1-16

[6] 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

[7] 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

[8] Hu X, Zhang Y, Guo L, Wang J, Cai Y, Fu H, Cai Y (2018) Cyclic behavior of saturated soft clay under stress path with bidirectional shear stresses. Soil Dyn Earthq Eng 104: 319-328

[9] 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

[10] Tian Y, Yao YP (2017) Modelling the non-coaxiality of soils from the view of cross-anisotropy. Comput Geotech 86: 219-229

[11] Vermeer P (1978) A double hardening model for sand. Geotechnique 28(4): 413-433

[12] 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

[13] Roscoe KH, Burland J (1968) On the generalized stress-strain behaviour of wet clay. Engineering Plasticity Cambridge, UK: 535-609

[14] Jefferies M (1993) Nor-Sand: a simle critical state model for sand. Geotechnique 43(1): 91-103

[15] Yu H (1998) CASM: A unified state parameter model for clay and sand. Int J Numer Anal Methods Geomech 22(8): 621-653

[16] Gajo A, Wood M (1999) Severn–Trent sand: a kinematic-hardening constitutive model: the q–p formulation. Geotechnique 49(5): 595-614

[17] Yao Y, Hou W, Zhou A (2009) UH model: three-dimensional unified hardening model for overconsolidated clays. Geotechnique 59(5): 451-469

[18] 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

[19] Yao Y, Sun D, Matsuoka H (2008) A unified constitutive model for both clay and sand with hardening parameter independent on stress path. Computers and Geotechnics 35(2): 210-222

[20] Taiebat M, Dafalias YF (2008) SANISAND: Simple anisotropic sand plasticity model. Int J Numer Anal Met 32(8): 915-948

[21] 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

[22] 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

[23] 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

[24] Mašín D (2005) A hypoplastic constitutive model for clays. Int J Numer Anal Methods Geomech 29(4): 311-336

[25] Wu W, Bauer E, Kolymbas D (1996) Hypoplastic constitutive model with critical state for granular materials. Mech Mater 23(1): 45-69

[26] 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

[27] Kolymbas D (1985) A generalized hypoelastic constitutive law. Proc XI Int Conf Soil Mechanics and Foundation Engineering San Francisco: 2626

[28] Chang CS, Hicher PY (2005) An elasto-plastic model for granular materials with microstructural consideration. Int J Solids Struct 42(14): 4258-4277

[29] Yin ZY, Chang CS, Hicher PY, Karstunen M (2009) Micromechanical analysis of kinematic hardening in natural clay. Int J Plast 25(8): 1413-1435

[30] 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

[31] 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

[32] Yin ZY, Chang CS (2009) Microstructural modelling of stress-dependent behaviour of clay. Int J Solids Struct 46(6): 1373-1388

[33] Xiong H, Nicot F, Yin Z (2017) A three - dimensional micromechanically based model. Int J Numer Anal Methods Geomech 41(17): 1669-1686

[34] Yao YP, Hou W, Zhou AN (2009) UH model: three-dimensional unified hardening model for overconsolidated clays. Géotechnique 59(5): 451-469

[35] Yin Z-Y, Jin Y-F. Practice of Optimisation Theory in Geotechnical Engineering. Springer2019.

[36] Amroune M (2019) Machine learning techniques applied to on-line voltage stability assessment: a review. Archives of Computational Methods in Engineering:

[37] Wang Z, Liu K, Li J, Zhu Y, Zhang Y (2019) Various frameworks and libraries of machine learning and deep learning: a survey. Archives of Computational Methods in Engineering:

[38] Li H, Yu H, Cao N, Tian H, Cheng S (2020) Applications of artificial intelligence in oil and gas development. Archives of Computational Methods in Engineering:

[39] Zhang P, Chen RP, Wu HN (2019) Real-time analysis and regulation of EPB shield steering using Random Forest. Automat Constr 106: 102860

[40] Lee S, Ha J, Zokhirova M, Moon H, Lee J (2017) Background information of deep learning for structural engineering. Archives of Computational Methods in Engineering 25(1): 121-129

[41] 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

[42] 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

[43] 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

[44] 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

[45] Cabalar AF, Cevik A (2011) Triaxial behavior of sand–mica mixtures using genetic programming. Expert Syst Appl 38(8): 10358-10367

[46] Javadi AA, Rezania M (2009) Applications of artificial intelligence and data mining techniques in soil modeling. Geomech Eng 1(1): 53-74

[47] Faramarzi A, Javadi AA, Alani AM (2012) EPR-based material modelling of soils considering volume changes. Comput Geosci-UK 48: 73-85

[48] Javadi AA, Faramarzi A, Ahangar-Asr A (2012) Analysis of behaviour of soils under cyclic loading using EPR-based finite element method. Finite Elem Anal Des 58: 53-65

[49] Cuisinier O, Javadi AA, Ahangar-Asr A, Masrouri F (2013) Identification of coupling parameters between shear strength behaviour of compacted soils and chemical's effects with an evolutionary-based data mining technique. Comput Geotech 48: 107-116

[50] 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

[51] Ahangar Asr A, Faramarzi A, Javadi AA (2018) An evolutionary modelling approach to predicting stress-strain behaviour of saturated granular soils. Engineering Computations 35(8): 2931-2952

[52] Zhao H, Huang Z, Zou Z (2014) Simulating the stress-strain relationship of geomaterials by support vector machine. Mathematical Problems in Engineering 2014: 1-7

[53] 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

[54] Ellis GW, Yao C, Zhao R, Penumadu D (1995) Stress-strain modeling of sands using artificial neural networks Journal of Geotechnical Engineering 121(5): 429-435

[55] Ghaboussi J, Sidarta DE (1998) New nested adaptive neural networks (NANN) for constitutive modeling. Comput Geotech 22(1): 29-52

[56] Sidarta DE, Ghaboussi J (1998) Constitutive modeling of geomaterials from non-uniform material tests. Comput Geotech 22(1): 53-71

[57] Penumadu D, Zhao RD (1999) Triaxial compression behavior of sand and gravel using artificial neural networks (ANN). Comput Geotech 24: 207-230

[58] Basheer IA (2002) Stress-strain behavior of geomaterials in loading reversal simulated by time-delay neural networks. Journal of Materials in Civil Engineering 14: 270-273

[59] Basheer IA (2000) Selection of methodology for neural network modeling of constitutive hystereses behavior of soils. Comput-Aided Civ Inf 15: 440–458

[60] Habibagahi G, Bamdad A (2003) A neural network framework for mechanical behavior of unsaturated soils. Can Geotech J 40(3): 684-693

[61] Banimahd M, Yasrobi SS, P.K.Woodward (2005) Artificial neural network for stress–strain behavior of sandy soils: Knowledge based verification. Comput Geotech 32(5): 377-386

[62] Shahin MA, Indraratna B (2006) Modeling the mechanical behavior of railway ballast using artificial neural networks. Can Geotech J 43(11): 1144-1152

[63] Fu Q, Hashash YMA, Jung S, Ghaboussi J (2007) Integration of laboratory testing and constitutive modeling of soils. Comput Geotech 34(5): 330-345

[64] Hashash YMA, Song H (2008) The integration of numerical modeling and physical measurements through inverse analysis in geotechnical engineering. KSCE Journal of Civil Engineering 12(3): 165-176

[65] Hashash YMA, Fu Q, Ghaboussi J, Lade PV, Saucier C (2009) Inverse analysis–based interpretation of sand behavior from triaxial compression tests subjected to full end restraint. Can Geotech J 46(7): 768-791

[66] He S, Li J (2009) Modeling nonlinear elastic behavior of reinforced soil using artificial neural networks. Appl Soft Comput 9(3): 954-961

[67] 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

[68] Sezer A (2011) Prediction of shear development in clean sands by use of particle shape information and artificial neural networks. Expert Syst Appl 38(5): 5603-5613

[69] Lv Y, Nie L, Xu K (2011) Study of the neural network constitutive models for turfy soil with different decomposition degree. 2011 Second International Conference on Mechanic Automation and Control Engineering Hohhot, China: 6111-6114

[70] Araei AA (2014) Artificial neural networks for modeling drained monotonic behavior of rockfill materials. Int J Geomech 14(3):

[71] 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

[72] Stefanos D, Gyan P (2015) On neural network constitutive models for geomaterials. Journal of Civil Engineering Research 5(5): 106-113

[73] Li Z, Chow JK, Wang YH (2017) Applying the artificial neural network to predict the soil responses in the DEM simulation. IOP Conference Series: Materials Science and Engineering 216:

[74] Lin P, Ratnam R, Sankari H, Garg A (2019) Mechanism of microstructural variation under cyclic shearing of Shantou marine clay: experimental investigation and model development. Geotechnical and Geological Engineering 37(5): 4163-4210

[75] Peng X-h, Wang Z-c, Luo T, Yu M, Luo Y-s (2008) An elasto-plastic constitutive model of moderate sandy clay based on BC-RBFNN. Journal of Central South University 15(s1):

[76] Li XD, Zhang GY, Xiang Ping Fang, Wang Jing Tao, Hui X (2008) Normalization characteristic of sands under triaxial compression and numerical modeling method (in Chinese). Chinese Journal of Rock Mechanics and Engineering 27(S1): 3082-3087

[77] 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

[78] Zhu J-H, Zaman MM, Anderson SA (1998) Modeling of soil behavior with a recurrent neural network. Can Geotech J 35: 858–872

[79] Romo MP, García SR, Mendoza MJ, Taboada‐Urtuzuástegui V (2001) Recurrent and constructive‐algorithm networks for sand behavior modeling. Int J Geomech 1(4): 371-387

[80] 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: 1-21

[81] Zhang N, Shen S-L, Zhou A, Xu Y-S (2019) Investigation on performance of neural networks using quadratic relative error cost function. IEEE Access 7: 106642-106652

[82] Wang L, Cai Y, Liu D (2018) Multiscale reliability-based topology optimization methodology for truss-like microstructures with unknown-but-bounded uncertainties. Comput Method Appl M 339: 358-388

[83] Wang K, Sun W (2019) Meta-modeling game for deriving theory-consistent, microstructure-based traction–separation laws via deep reinforcement learning. Comput Method Appl M 346: 216-241

[84] Koza JR. Genetic programming: on the programming of computers by natural selection. Cambridge, MA.: MIT Press; 1992.

[85] Gomes FM, Pereira FM, Silva AF, Silva MB (2019) Multiple response optimization: Analysis of genetic programming for symbolic regression and assessment of desirability functions. Knowl-Based Syst 179: 21-33

[86] Jin YF, Yin ZY, Zhou WH, Yin JH, Shao JF (2019) A single-objective EPR based model for creep index of soft clays considering L2 regularization. Eng Geol 248: 242-255

[87] Yin ZY, Jin YF, Huang HW, Shen SL (2016) Evolutionary polynomial regression based modelling of clay compressibility using an enhanced hybrid real-coded genetic algorithm. Eng Geol 210: 158–167

[88] Hein D, Udluft S, Runkler TA (2018) Interpretable policies for reinforcement learning by genetic programming. Eng Appl Artif Intel 76: 158-169

[89] Beg AH, Islam MZ (2016) Advantages and limitations of genetic algorithms for clustering records. IEEE 11th Conf Industrial Electronics and Applications (ICIEA) Hefei, China:

[90] Yin Z-Y, Jin Y-F, Shen JS, Hicher P-Y (2018) Optimization techniques for identifying soil parameters in geotechnical engineering: Comparative study and enhancement. Int J Numer Anal Met 42(1): 70-94

[91] Cortes C, Vapnik V (1995) Support-Vector networks. Mach Learn 20: 273-297

[92] Okwuashi O, Ndehedehe CE (2020) Deep support vector machine for hyperspectral image classification. Pattern Recognition 103:

[93] Qi CC, Tang XL (2018) A hybrid ensemble method for improved prediction of slope stability. Int J Numer Anal Met 42(15): 1823-1839

[94] Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(9): 533–536

[95] LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553): 436-444

[96] Armaghani DJ, Mohamad ET, Narayanasamy MS, Narita N, Yagiz S (2017) Development of hybrid intelligent models for predicting TBM penetration rate in hard rock condition. Tunnell Undergr Space Technol 63: 29-43

[97] Atangana Njock PG, Shen S-L, Zhou A, Lyu H-M (2020) Evaluation of soil liquefaction using AI technology incorporating a coupled ENN / t-SNE model. Soil Dyn Earthq Eng 130: 105988

[98] Lu Y, Sundararajan N, Saratchandran P (1998) Performance evaluation of a sequential minimal radial basis function (RBF) neural network learning algorithm. IEEE Transactions on Neural Networks 9(2): 308-318

[99] Kang F, Li J, Xu Q (2017) System reliability analysis of slopes using multilayer perceptron and radial basis function networks. Int J Numer Anal Met 41(18): 1962-1978

[100] Hurtado JE (2001) Neural networks in stochastic mechanics. Archives of Computational Methods in Engineering 8(3): 303-342

[101] Wang K, Sun W (2018) A multiscale multi-permeability poroplasticity model linked by recursive homogenizations and deep learning. Comput Method Appl M 334: 337-380

[102] Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9: 1735–1780

[103] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau v, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. arxiv 1406.1078:

[104] Haidong S, Junsheng C, Hongkai J, Yu Y, Zhantao W (2020) Enhanced deep gated recurrent unit and complex wavelet packet energy moment entropy for early fault prognosis of bearing. Knowl-Based Syst 188:

[105] Kondner RL (1963) Hyperbolic stress-strain response: cohesive soils. Journal of Soil Mechanics and

Foundations Engineering Division 89(1): 115-144

[106] Roscoe KH, Schofield AN, Thurairajah A (1963) Yielding of clays in states wetter than critical. Géotechnique 13(3): 211-240

[107] Roscoe KH, Schofield AN, Wroth CP (1958) On the yielding of soils. Géotechnique 8(1): 22-53

[108] Schanz T, Vermeer PA, Bonnier PG. The hardening soil model: Formulation and verification. Beyond 2000 in Computational Geotechnics. Amsterdam, Balkema, Rotterdam1999. p. 281–296.

[109] Pande GN, Sharma KG (1983) Multi‐laminate model of clays—a numerical evaluation of the influence of rotation of the principal stress axes. Int J Numer Anal Met 7(4): 397-418

[110] Yin Z-Y, Wu Z-X, Hicher P-Y (2018) Modeling monotonic and cyclic behavior of granular materials by exponential constitutive function. J Eng Mech-ASCE 144(4):

[111] Zhang P (2019) A novel feature selection method based on global sensitivity analysis with application in machine learning-based prediction model. Appl Soft Comput 85: 105859

[112] Cai J, Luo J, Wang S, Yang S (2018) Feature selection in machine learning: A new perspective. Neurocomputing 300: 70-79

[113] Javadi AA, Rezania M (2009) Intelligent finite element method: An evolutionary approach to constitutive modeling. Adv Eng Inform 23(4): 442-451

[114] Yun GJ, Ghaboussi J, Elnashai AS (2008) Self-learning simulation method for inverse nonlinear modeling of cyclic behavior of connections. Comput Method Appl M 197(33-40): 2836-2857

[115] Shin HS, Pande GN (2003) Identification of elastic constants for orthotropic materials from a structural test. Comput Geotech 30(7): 571-577

[116] Ghaboussi J, Pecknold AD, Zhang M, Haj‐Ali MR (1998) Autoprogressive training of neural network constitutive models. International Journal for Numerical Methods in Engineering 42: 105-126

[117] Jacobs RA (1988) Increased rates of convergence through learning rate adaptation. Neural Networks 1: 295-307

[118] Holland J (1992) Genetic Algorithms. Scientific American 267: 66–72

[119] Mirjalili S, Lewis A (2016) The Whale Optimization Algorithm. Adv Eng Softw 95: 51-67

[120] Kennedy J, Eberhart R (1995) Particle swarm optimization. IEEE International Conference on Neural Networks Perth, Australia: 1942-1948

[121] Maas AL, Hannun AY, Ng AY. Rectifier nonlinearities improve neural network acoustic models. 30th International Conference on Machine Learning. Atlanta, Georgia, USA2013.

[122] Ramachandran P, Zoph B, Le. QV (2017) Swish: a self-gated activation function. arXiv:171005941:

[123] Misra D (2019) Mish: a self regularized non-monotonic neural activation function. arXiv:190808681:

[124] Fahlman SE (1988) An empirical study of learning speed in back-propagation networks. Technical report CMU-CS-88-162 Carnegie-Mellon University:

[125] Riedmiller M, Braun H (1993) A direct adaptive method for faster backpropagation learning. IEEE

International Conference on Neural Networks San Francisco, CA, USA: 586-591

[126] Levenberg K (1944) A method for the solution of certain non-linear problems in least squares. Q Appl Math 2(2): 164-168

[127] Marquardt DW (1963) An algorithm for least-squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics 11(2): 431-441

[128] Møller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6: 525-533

[129] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representation by error propagation. Cambridge: Parallel distributed processing, MIT Press; 1986.

[130] Larochelle H, Bengio Y, Louradour J, Lamblin P (2009) Exploring strategies for training deep neural networks. Journal of Machine Learning Research 1: 1-40

[131] Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12: 2121-2159

[132] Tieleman T, Hinton G (2012) Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning. Technical report:

[133] Kingma DP (2015) Adam: a method for stochastic optimization. International Conference on Learning Representations San Diego, CA:

[134] Zhang P, Li H, Ha QP, Yin Z-Y, Chen R-P (2020) Reinforcement learning based optimizer for improvement of predicting tunneling-induced ground responses. Advanced Engineering Informatics 45:

[135] Zhang P, Yin ZY, Jin YF, Chan T, Gao FP (2020) Intelligent modelling of clay compressibility using hybrid meta-heuristic and machine learning algorithms. Geoscience Frontiers: in press

[136] Fujii M, Takahashi A, Takahashi M (2019) Asymptotic expansion as prior knowledge in deep learning method for high dimensional BSDEs. Asia-Pacific Financial Markets 26(3): 391-408

[137] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15: 1929–1958

[138] 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

[139] Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. NIPS:

[140] Moradi R, Berangi R, Minaei B (2019) A survey of regularization strategies for deep models. Artificial Intelligence Review:

[141] Barnard E, Wessels L (1992) Extrapolation and interpolation in neural network classifiers. IEEE Control Systems 12(5): 50–53

[142] Yin ZY, Wang JH (2012) A one-dimensional strain-rate based model for soft structured clays. Science China-Technological Sciences 55(1): 90-100

[143] Yin ZY, Chang CS (2009) Non-uniqueness of critical state line in compression and extension

conditions. Int J Numer Anal Methods Geomech 33(10): 1315-1338

[144] Yin ZY, Chang CS (2013) Stress–dilatancy behavior for sand under loading and unloading conditions. Int J Numer Anal Methods Geomech 37(8): 855-870

[145] Xiao Y, Liu H, Chen Y, Chu J (2014) Influence of intermediate principal stress on the strength and dilatancy behavior of rockfill material. Journal of Geotechnical and Geoenvironmental Engineering 140(11): 04014064

[146] Yin ZY, Karstunen M, Chang CS, Koskinen M, Lojander M (2011) Modeling Time-Dependent Behavior of Soft Sensitive Clay. Journal of Geotechnical and Geoenvironmental Engineering 137(11): 1103-1113

[147] Yu H, Yuan X (2006) On a class of non-coaxial plasticity models for granular soils. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 462(2067): 725-748

[148] 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 Intell 78: 102-123

[149] Jin Y-F, Yin Z-Y, Zhou W-H, Liu X (2020) Intelligent model selection with updating parameters during staged excavation using optimization method. Acta Geotech:

[150] Yin ZY, Xu Q, Hicher PY (2013) A simple critical-state-based double-yield-surface model for clay behavior under complex loading. Acta Geotech 8(5): 509-523

[151] Yin ZY, Yin JH, Huang HW (2015) Rate-Dependent and Long-Term Yield Stress and Strength of Soft Wenzhou Marine Clay: Experiments and Modeling. Marine Georesources & Geotechnology 33(1): 79-91

[152] Yin ZY, Zhu QY, Yin JH, Ni Q (2014) Stress relaxation coefficient and formulation for soft soils. Géotechnique Letters 4: 45-51

[153] Yin Z-Y, Zhu Q-Y, Zhang D-M (2017) Comparison of two creep degradation modeling approaches for soft structured soils. Acta Geotech 12(6): 1395-1413

[154] Zhu Q-Y, Jin Y-F, Shang X-Y, Chen T (2019) A 1D model considering the combined effect of strain-rate and temperature for soft soil. Geomechanics and Engineering 18(2): 133-140

[155] Yin Z-Y, Jin Y-F, Shen S-L, Huang H-W (2017) 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

[156] Graves A, Mohamed A-r, Hinton G (2013) Speech recognition with deep recurrent neural networks ICASSP: 6645–6649

[157] Ibsen LB, Bødker LB. Baskarp Sand No. 15: data report 9301. Aalborg: Geotechnical Engineering Group. Data Report, No. 9401. 1994.

Table

Table 1 Summary of ML based constitutive models in literature

Soil type	References	Experiment type	Loading type	Drained type	Data source	Test scope	Algorithm	History applied	Input	Output
Sand	Ellis et al. (1995)	Triaxial test	M	U	Experiment	I	BPNN	Yes	C_u, D_r, OCR, σ_3, σ_1, u, ε_a	σ_1, u
Sacramento river sand	Sidarta and Ghaboussi (1998)	Triaxial test	M	D	Experiment	/	BPNN	Yes	σ_{rr}, σ_{zz}, σ_{rz}, $\sigma_{\theta\theta}$, ε_{rr}, ε_{zz}, ε_{rz}, $\varepsilon_{\theta\theta}$, e_0	σ_{rr}, σ_{zz}, σ_{rz}, $\sigma_{\theta\theta}$
Sacramento river sand	Ghaboussi and Sidarta (1998)	Triaxial test	M	D + U	Experiment	I	BPNN	Yes	p, q, u, ε_d, ε_v, $\Delta\varepsilon_d$, $\Delta\varepsilon_v$, e_0	Δp, Δq, Δu
Residual soil	Zhu et al. (1998a)	Triaxial test	M	D	Experiment	I&E	RNN	Yes	σ_1, $\Delta\sigma_1$, σ_3, $\Delta\sigma_3$, u, e, ε_a, ε_v	ε_a, ε_v
Residual soil	Zhu et al. (1998a)	Triaxial test	M	U	Experiment	I	RNN	Yes	ε_a, $\Delta\varepsilon_a$, σ_3, e, q, u	q, u
Sand	Zhu et al. (1998b)	Triaxial test	M	D	Experiment	I	RNN	Yes	D_r, Δq, σ_1, $\Delta\sigma_1$, ε_a, ε_v	ε_a, ε_v
Sand	Penumadu and Zhao (1999)	Triaxial test	M	D	Experiment	/	BPNN	Yes	D_{50}, C_u, C_c, h, n_s, e, σ_3, ε_a, $\Delta\varepsilon_a$, q, ε_v	q, ε_v
/	Basheer (2000)	/	C	/	Synthetic data using KM	I	BPNN	Yes	b, ε_a, σ_n	σ_n
Fat clay	Basheer (2000)	Unconfined compression	C	/	Experiment	/	BPNN	Yes	λ_1, λ_2, λ_3, ρ_d, w, ε_a, σ_1	σ_1
Coarse sand	Romo et al. (2001)	Triaxial test	M	U	Experiment	I&E	RNN	Yes	D_r, σ_3, ε_a, q, u	q, u
/	Basheer (2002)	/	C	/	Synthetic data	I	BPNN	Yes	b, ε_a, σ_1	σ_1
Lateritic gravel	Habibagahi and Bamdad (2003)	Triaxial test	M	D	Experiment	I&E	BPNN	Yes	ρ_d, w; Sr; p–u_a; ε_a, q, ε_v, u_a–u_w	q, ε_v u_a–u_w
Toyoura sand	Banimahd et al. (2005)	Triaxial test	M	U	Experiment	/	BPNN	Yes	D_r, C_u, C_c, I_s, P_f, σ_3, ε_a, $\Delta\varepsilon_a$, q	q
Toyoura sand	Banimahd et al. (2005)	Triaxial test	M	U	Experiment	/	BPNN	Yes	D_r, C_u, C_c, I_s, P_f, σ_3, ε_a, $\Delta\varepsilon_a$, u	u
Ballast	Shahin and Indraratna (2006)	Triaxial test	M	D	Experiment	I	BPNN	Yes	D_{50}, C_u, C_c, σ_3, e, ε_a, $\Delta\varepsilon_a$, γ', q, ε_v	q, ε_v

Boston blue clay	Fu et al. (2007)	Triaxial test	M	U	Synthetic data using MCC	/	BPNN	Yes	$\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}, \varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23}$	$\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}$
Ricci sand	Hashash and Song (2008)	Triaxial test	M	D	Experiment	/	BPNN	Yes	$\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}, \varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23}$	$\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}$
Moderate sandy clay	Peng et al. (2008)	Triaxial test	M	D	Experiment	I	RBF	No	p, q	$\varepsilon_d, \varepsilon_v$
Sand	Li et al. (2008)	Triaxial test	M	D	Experiment	No	RBF	No	q	ε_d
Reinforced soil	He and Li (2009)	Triaxial test	M	U	Experiment	I	BPNN	No	$\sigma_3, \varepsilon_a, \beta_F, \beta_L, t_c$	σ_1
Ricci sand	Hashash et al. (2009)	Triaxial test	M	D	Experiment	/	BPNN	Yes	$\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}, \varepsilon_{11}, \varepsilon_{22}, \varepsilon_{33}, \varepsilon_{12}, \varepsilon_{13}, \varepsilon_{23}$	$\sigma_{11}, \sigma_{22}, \sigma_{33}, \sigma_{12}, \sigma_{13}, \sigma_{23}$
/	Javadi and Rezania (2009)	Triaxial test	M	D	Experiment	I	EPR	Yes	$q, \sigma_3, \varepsilon_a, \Delta\varepsilon_a$	q
Lateritic gravel	Johari et al. (2011)	Triaxial test	M	D	Experiment	I&E	BPNN	Yes	$\rho_d, w; Sr; p–u_a; \varepsilon_a, q, \varepsilon_v, u_a–u_w$	$q, \varepsilon_v, u_a–u_w$
Western Anatolian sand	Sezer (2011)	Direct shear	M	/	Experiment	/	BPNN	Yes	$D_r, D_R, D_{10}, C_u, C_c, r, s, \gamma, \sigma_3, e, \tau$	τ
Turfy soil	Lv et al. (2011)	Triaxial test	M	U	Experiment	/	BPNN	No	$d, \sigma_3, \varepsilon_a$	σ_1
Sand–mica mixtures	Cabalar and Cevik (2011)	Triaxial test	M	U	Experiment	/	GP	No	p_m, ε_a	q, u
Kaolin clay	Faramarzi et al. (2012)	Triaxial test	M	D	Experiment	I	EPR	Yes	$p, q, \varepsilon_v, \varepsilon_q, \Delta\varepsilon_q$	q, ε_v
/	Javadi et al. (2012)	Triaxial test	C	D	Synthetic data using MCC	I	EPR	Yes	$p, q, \varepsilon_v, \varepsilon_q, \Delta\varepsilon_q$	q, ε_v
Mixed Manois argillite	Cuisinier et al. (2013)	Triaxial test	M	U	Experiment	/	EPR	No	$\rho_d, t, \varepsilon_a, u, \sigma_3, e_M, e_m, SS$	q
Rockfill	Araei (2014)	Triaxial test	M	D	Experiment	I	BPNN	Yes	$\rho_d, n_s, \sigma_3, \varepsilon_a, \Delta\varepsilon_a, q, LA, w_o, p_{p1}, p_{p2}, p_{p3}, p_{p4}$	q
Rockfill	Araei (2014)	Triaxial test	M	D	Experiment	/	BPNN	Yes	$\rho_d, n_s, \sigma_3, \varepsilon_a, \Delta\varepsilon_a, \varepsilon_v, LA, w_o, p_{p1}, p_{p2}, p_{p3}, p_{p4}$	ε_v

Carbonate sand	Rashidian and Hassanlourad (2014)	Triaxial test	M	D	Experiment	I	BPNN	No	D_r, σ_3, ε_a, p_c, e_{max}	q, ε_v
Clay	Zhao et al. (2014)	Triaxial test	M	D	Synthetic data using MCC	/	SVM	Yes	p, q, ε_a, ε_v, e	ε_a, ε_v
Sand	Stefanos and Gyan (2015)	Triaxial test	M	D	Synthetic data using HS	I	BPNN	Yes	σ_{11}, σ_{22}, σ_{33}, τ, $\Delta\varepsilon_{11}$, $\Delta\varepsilon_{22}$, $\Delta\varepsilon_{33}$, $\Delta\gamma$, ξ	σ_{11}, σ_{22}, σ_{33}, τ
Loose sand	Stefanos and Gyan (2015)	Triaxial test	C	U	Synthetic data using TDHM	I	BPNN	Yes	σ_{11}, σ_{22}, σ_{33}, τ, $\Delta\varepsilon_{11}$, $\Delta\varepsilon_{22}$, $\Delta\varepsilon_{33}$, $\Delta\gamma$, ξ	σ_{11}, σ_{22}, σ_{33}, τ
Carbonate sand	Kohestani and Hassanlourad (2016)	Triaxial test	M	D	Experiment	I	BPNN	No	D_r, σ_3, ε_a, p_c, e_{max}, e_{min}	q, ε_v
Carbonate sand	Kohestani and Hassanlourad (2016)	Triaxial test	M	D	Experiment	I	SVM	No	D_r, σ_3, ε_a, p_c, e_{max}, e_{min}	q, ε_v
/	Li et al. (2017)	/	M	/	Synthetic data using DEM	E	BPNN	No	/	q
Frozen soil	Nassr et al. (2018)	Triaxial test	M	U	Experiment	/	EPR	Yes	T, ε_a, $\dot{\varepsilon}$, $\Delta\varepsilon_a$, σ_3, q	q
Granular soil	Ahangar Asr et al. (2018)	Triaxial test	M	D	Experiment	/	EPR	Yes	D_{50}, C_u, C_c, h, n_s, e, σ_3, ε_a, $\Delta\varepsilon_a$, q, ε_v	q, ε_v
Granular	Wang et al (2018)	Simple shear	C	/	Synthetic data using DEM	I	LSTM	Yes	σ_3, γ, τ	τ
Clay	Lin et al. (2019)	Triaxial test	C	U	Experiment	/	RBF	No	Nc, q	ε_a
Clay	Lin et al. (2019)	Triaxial test	C	U	Experiment	/	BPNN	No	Nc, q	ε_a
Granular	Wang et al (2019)	Tension-shear test	C	/	Synthetic data using DEM	/	GRU	Yes	$\delta_{n,m}$, φ, CN, A_{sf}, d_a, c_t, l_{sp}, ρ_g	$t_{n,m}$
/	Zhang et al. (2019)	/	M	/	Synthetic data using MCC	No	LSTM	No	e, p, λ, ε_a	q
Sand	Zhang et al. (2020)	Triaxial test	C	D + U	Synthetic data using EM	I&E	LSTM	Yes	$L1$, $L2$, $L3$, m, e_0, q, p, ε_v, ε_a	q, p, ε_v, ε_a
Fontainebleau sand	Zhang et al. (2020)	Simple shear	C	D + U	Experiment	I	LSTM	Yes	Nc, m, τ_{ave}, τ_{cyc}, e_0, τ; σ_n, γ, e	σ_n, γ, e

Toyoura sand	Zhang et al. (2020)	Triaxial test	C	U	Experiment	I	LSTM	Yes	D_r, q_{ave}, q_{cyc}, q, p, ε_a, Nc	p, ε_a

Remarks: M = monotonic; C= cyclic; D = drained; U = undrained; I = interpolation; E = extrapolation; KM = monotonic Konder's expression; MCC = modified Cam Clay; HS = hardening soil; TDH = Two-Surface model in multilaminate framework; EM = endochronic model; DEM = discrete element method; e_0 = void ratio; $\varepsilon_{\mathrm{v}} = \varepsilon_1 + 2\varepsilon_3$; $\varepsilon_{\mathrm{d}} = \varepsilon_1 - \varepsilon_3$; σ = normal stress; τ = shear stress; γ = shear strain; / = corresponding information is not recorded. Nc = number of cycles; p = mean effective stress; q = deviatoric stress; σ_3 = effective confining stress; σ_1 = effective major principal stress; $L1$, $L2$, $L3$ = three labels for marking the cyclic loading process as mentioned earlier; e_0 = initial void ratio; m = code for controlling experiment types, 1 represents drained condition and 0 represents undrained condition; e_0 = initial void ratio; τ_{ave} = average shear stress; τ_{cyc} = cyclic shear stress amplitude; q_{ave} = average deviatoric stress; q_{cyc} = amplitude of the cyclic deviatoric stress; ρ_d = dry density; t = alkaline water circulation time; ε_{a} = axial strain; u = pore-water pressure; e_M = macro-pores; e_m = porosity of micro-pores; SS = specific surface; λ= slope of the virgin consolidation line; T = temperature; $\dot{\varepsilon}$ = strain rate; D_{50} = average grain size; C_u =coefficient of uniformity; C_c =coefficient of curvature; h = hardness of the mineral; n_s =shape factor; p_m = percentage of mica; p_c = carbonate calcium content; β_{F} = contents of fiber; β_{L} = contents of lime; t_c = curing period of soil; e_{max} = maximum void ratio; e_{min} = minimum void ratio; w = water content; Sr = degree of saturation; mean stress with respect to pore-air pressure; u_{a}–u_{w} = suction; b = empirical constant; λ_1, λ_2, λ_3 = encoding for loading; P_f = fine percentage; I_s = fine shape index; σ_{rr}, σ_{zz}, σ_{rz}, $\sigma_{\theta\theta}$, σ_{11}, σ_{22}, σ_{33}, σ_{12}, σ_{13}, σ_{23} = stress components; ε_{rr}, ε_{zz}, ε_{rz}, $\varepsilon_{\theta\theta}$, ε_{11}, ε_{22}, ε_{33}, ε_{12}, ε_{13}, ε_{23} = strain components; LA = Los Angeles abrasion; w_{o} = optimum moisture content; p_{p1}, p_{p2}, p_{p3}, p_{p4} = passing percentages for grain size 39.2, 25.4, 4.75, and 0.2 mm; γ' = bulk unit weight; D_{R} = area–perimeter fractal dimension; D_{10} = effective diameter; r = roundness; s = sphericity; ξ = current length of strain trajectory; d = decomposition degree; $\delta_{n,m}$ = normal displacement jump; $t_{n,m}$ = normal traction; φ = porosity; CN = coordination number; A_{sf} = strong fabric tensor; d_a, c_t, l_{sp}, ρ_g = measures of grain connectivities.

Table 2 Main characteristics of typically adopted ML algorithms for developing constitutive models of soils

ML algorithms	Advantages	Limitations
GP	Simple and explicit expression	Numerous structure; No sequential prediction ability; single output prediction
EPR	Simple and explicit expression	Poor non-linear mapping ability; No sequential prediction ability; single output prediction
SVM	Structural risk minimization; Adaptability for high-dimensional data	Poor readability and interpretability; No sequential prediction ability; single output prediction
BPNN	Strong non-linear mapping ability; multi output prediction	No sequential prediction ability; Gradients exploding or vanishing
RBF	Low computational cost; multi output prediction	No sequential prediction ability; Poor generalization ability
RNN	Strong non-linear mapping ability; Sequential prediction ability; multi outputs prediction	Gradients exploding or vanishing
LSTM	Strong non-linear mapping ability; Sequential prediction ability; multi outputs prediction	Numerous weights and biases
GRU	Strong non-linear mapping ability; Sequential prediction ability; multi outputs prediction	Numerous weights and biases

Table 3 Summary of architectures and learning strategies used in ANN based models

Algorithm	References	Hidden layers	Hidden neurons	Hyper-parameter determination	Activation function	Learning strategy	Loss function	Note
BPNN	Ellis et al. (1995)	1	10	/	/	SGD	/	/
	Ghaboussi and Sidarta (1998)	2	20-20	Auto-progressive	*sigmoid*	DB	/	/
	Sidarta and Ghaboussi (1998)	2	9-9/20-20	Auto-progressive	/	/	/	/
	Penumadu and Zhao (1999)	1	15	Trial and error	/	BProp	/	/
	Basheer (2000)	1	10/20	Trial and error	/	/	/	/
	Basheer (2002)	1	15	/	/	/	/	/
	Habibagahi and Bamdad (2003)	1	4	Trial and error	*sigmoid*	GDR	MSSE	/
	Banimahd et al. (2005)	1	10/15	Trial and error	*sigmoid*	LM	/	/
	Shahin and Indraratna (2006)	1	10	Trial and error	*tanh*, *sigmoid*	/	/	/
	Fu et al. (2007)	2	14-14/18-18	Auto-progressive	/	/	/	/
	Hashash and Song (2008)	2	14-14	Auto-progressive	/	/	/	/
	He and Li (2009)	1	4	Trial and error	*tanh*	LM	SSE	/
	Hashash et al. (2009)	2	18-18	Auto-progressive	/	/	/	/
	Sezer (2011)	2	15-30	/	/	LM, GD, SCG	MSE	/
	Johari et al. (2011)	1	5	Trial and error	*tanh*	GA	SSE	/
	Lv et al. (2011)	1	50	/	*tanh*	LM	MSE	/
	Araei (2014)	1	10	Trial and error	*tanh*	LM	/	/
	Rashidian and Hassanlourad (2014)	1	10	Trial and error	*tanh*	LM	MSE	/
	Stefanos and Gyan (2015)	2	18-8	/	/	RProp	/	/
	Kohestani and Hassanlourad (2016)	2	20-20	Trial and error	*sigmoid*	LM	SSE	/
	Li et al. (2017)	1	41	/	/	SCG	MSE	/
	Lin et al. (2019)	1	39/41	/	/	/	/	Weight decay
RBF	Lin et al. (2019)	1	27/16	/	/	/	/	Weight decay

RNN	Zhu et al. (1998a), Zhu et al. (1998b)	1	20	Trial and error	*tanh*	SGD	SSE	/
	Romo et al. (2001)	1	20	Trial and error	*bi-sigmoid*	SCG, QProp	/	/
LSTM	Wang et al. (2018)	2	80-80	Trial and error	*sigmoid*	/	MSE	Dropout
	Wang et al. (2019)	2	32-32	Trial and error	*/*	*Adam*	MSE	/
	Zhang et al. (2019)	2	12-12	/	*sigmoid*	CG	REMSE	/
	Zhang et al. (2020)	2	90-40/90-30	Trial and error	*tanh*, *ReLU*	*Adam*	MSE	*k*-fold cross-validation, Dropout, Weight decay
	Zhang et al. (2020)	3	80-80-80	Trial and error	*tanh*, *ReLU*	*Adam*	MSE	*k*-fold cross-validation, Dropout, Weight decay
	Zhang et al. (2020)	2	60-50	Trial and error	*tanh*	*Adam*	MSE	*k*-fold cross-validation, Dropout, Weight decay
GRU	Wang et al. (2019)	2	32-32	Trial and error	*/*	*Adam*	MSE	/

Remarks: BProp = Backpropagation training; QProp = Quick propagation training; RProp = Resilient propagation training; SCG = Scaled conjugate gradient; LM = Levenberg–Marquardt; SGD = Stochastic gradient descend; DB = Delta Bar; GDR = Generalized delta rule.

Table 4 Summary of learning strategies used in SVM, EPR and GP based models

Algorithm	References	Hyper-parameter determination	Learning strategy	Loss function
SVM	Zhao et al. (2014)	/	/	/
	Kohestani and Hassanlourad (2016)	Trial and error	/	ε-insensitive
EPR	Javadi and Rezania (2009)	/	GA	/
	Faramarzi et al. (2012)	Trial and error	GA	
	Javadi et al. (2012)	/	GA	
	Cuisinier et al. (2013)	/	GA	SSE
	Nassr et al. (2018)	Trial and error	GA	/
	Ahangar Asr et al. (2018)	/	GA	SSE
GP	Cabalar and Cevik (2011)	/	/	MAE

Table 5 Configurations of different ML based models

Configuration	BPNN	LSTM	GRU	BiLSTM	RBF
Architecture	130	70–70	80	70–70	70
Time step	/	3	3	3	/
Activation function	*ReLU*	*ReLU*	*ReLU*	*ELU*	Gaussian function with $\sigma = 1.5$
Loss function	MSE	MSE	MSE	MSE	MSE
Optimizer	AdaMax	AdaMax	AdaMax	AdaMax	Least square
Batch size	60	60	60	60	/
Overfitting prevention	10-fold CV	10-fold CV	10-fold CV	10-fold CV	10-fold CV

Remarks: architecture represents the number of hidden layers and neurons, e.g. 130 denotes one hidden layer with 130 neurons; CV = cross validation

Table 6 Values of indicators generated by different ML based models

ML	Training set						Testing set					
	p		*q*		*e*		*p*		*q*		*e*	
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
BPNN	5.0	0.68%	9.9	1.63%	3e-4	0.05%	47.0	7.5%	133.5	25.2%	0.018	2.4%
RBF	4.7	1.04%	9.8	2.34%	7e-4	0.11%	61.7	10.4%	107.4	21.1%	0.007	1.0%
LSTM	**1.2**	0.33%	2.8	0.89%	2e-4	0.03%	15.3	2.7%	35.5	6.9%	0.005	0.6%
GRU	2.5	0.36%	3.5	**0.53%**	2e-4	0.03%	18.3	4.6%	62.8	16.8%	0.001	0.2%
BiLSTM	1.3	**0.33%**	**1.7**	0.62%	**2e-4**	**0.02%**	**8.8**	**2.6%**	**17.7**	**7.2%**	**9e-4**	**0.1%**

Remarks: bold font denotes the optimum indicator value at each case

Figure captions

Fig. 1 Relationship between the complexity of constitutive model and the number of parameters

Fig. 2 Increasing number of papers regrading ML based constitutive models

Fig. 3 Proportion of various machine learning based model

Fig. 4 Framework of genetic programming

Fig. 5 Framework of evolutionary polynomial regression

Fig. 6 Framework of support vector machine

Fig. 7 Framework of backpropagation neural network

Fig. 8 Framework of radial basis function neural network

Fig. 9 Framework of recurrent neural network

Fig. 10 Framework of memory cell of LSTM

Fig. 11 Framework of memory cell of GRU

Fig. 12 Forward topology for training constitutive model of soil

Fig. 13 Feedback topology for training constitutive model of soil

Fig. 14 Activation functions: (a) original formulation; (b) derivative

Fig. 15 Proportion of testing set type used in the training of constitutive model of soil

Fig. 16 Predicted stress-strain responses using four ML algorithms: (a) e_0 = 0.696, σ'_3 = 19.9 kPa; (b) e_0 = 0.695 σ'_3 = 160 kPa; (c) e_0 = 0.852, σ'_3 = 5 kPa; (d) e_0 = 0.852, σ'_3 = 800 kPa

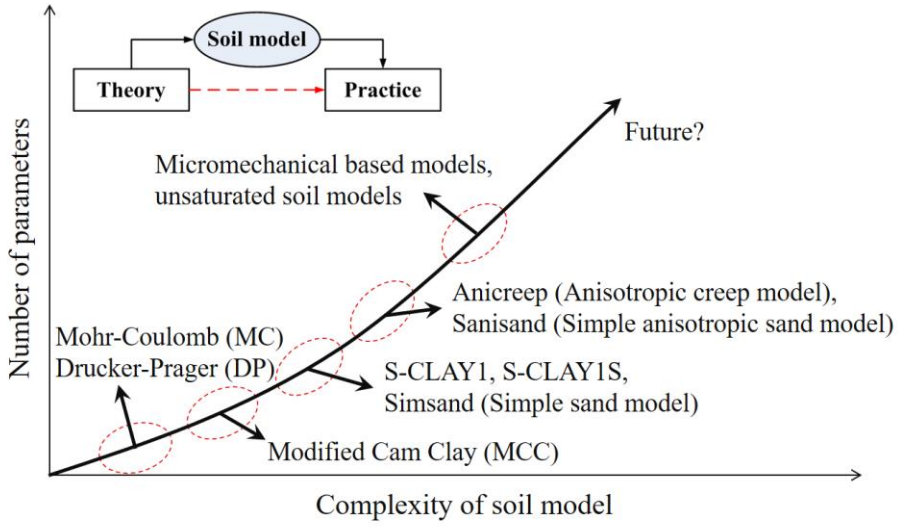

Fig. 1 Relationship between the complexity of constitutive model and the number of parameters

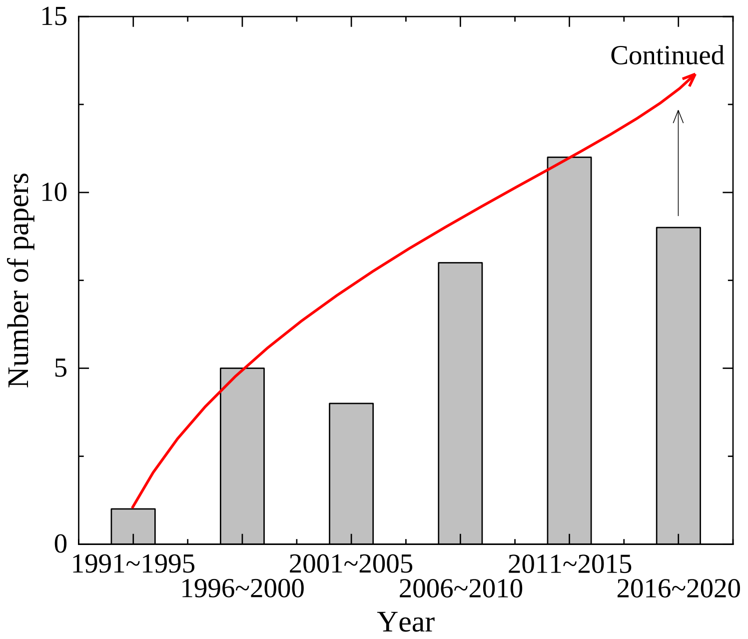

Fig. 2 Increasing number of papers regrading ML based constitutive models

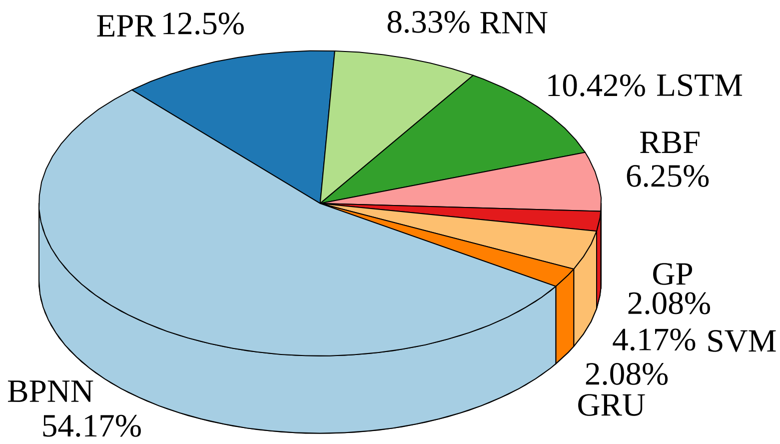

Fig. 3 Proportion of various machine learning based model

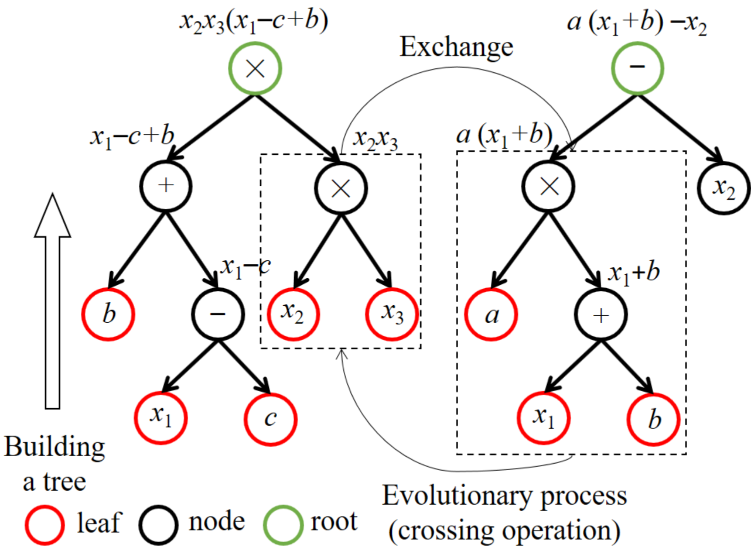

Fig. 4 Framework of genetic programming

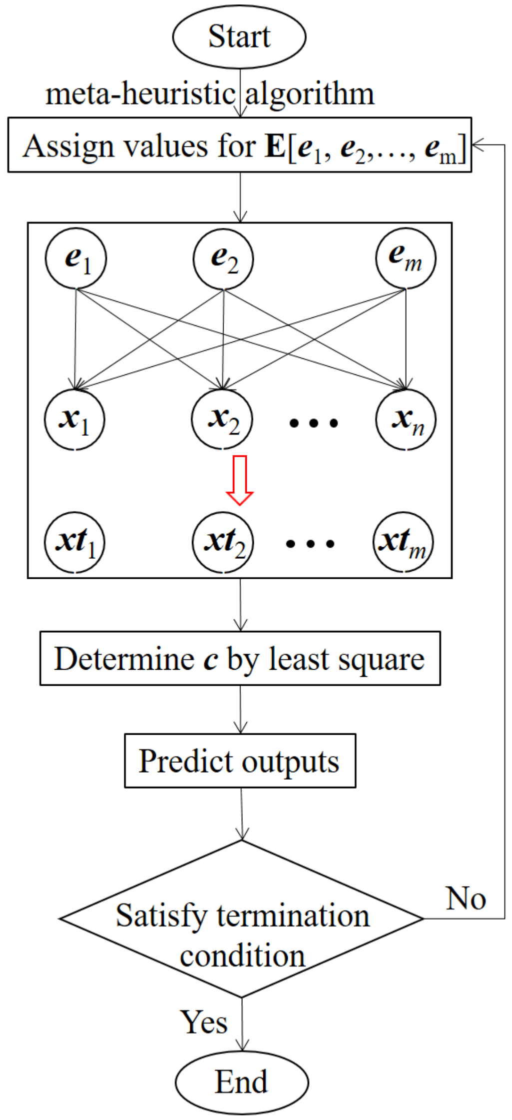

Fig. 5 Framework of evolutionary polynomial regression

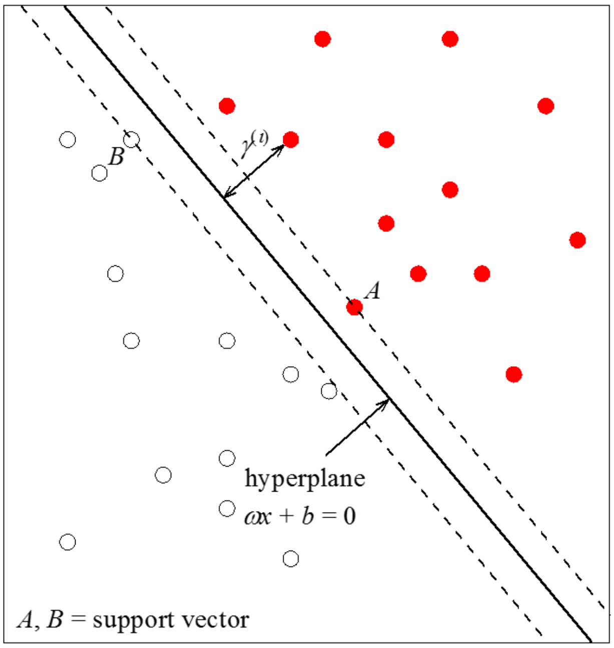

Fig. 6 Framework of support vector machine

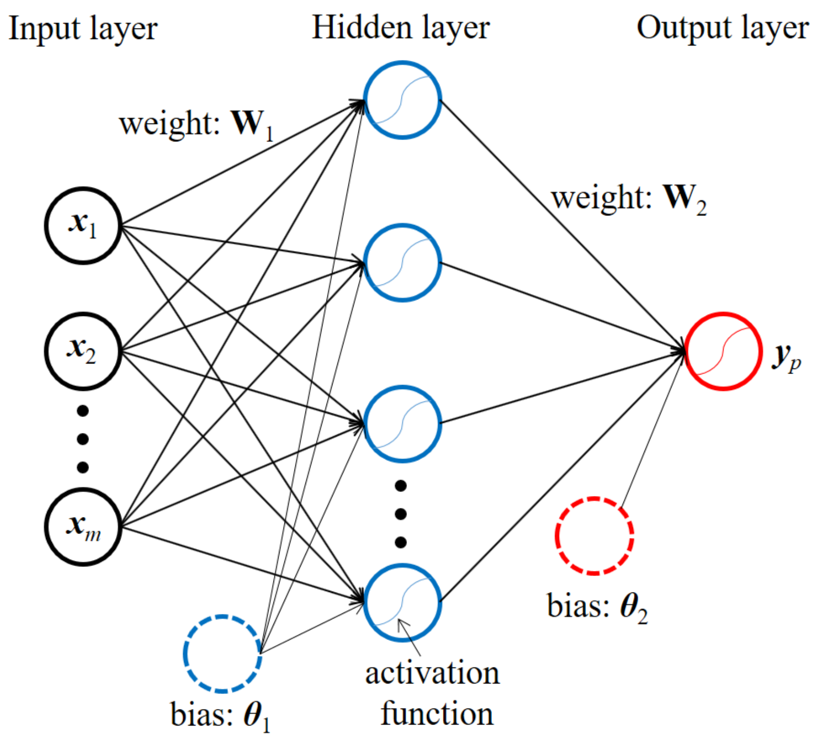

Fig. 7 Framework of backpropagation neural network

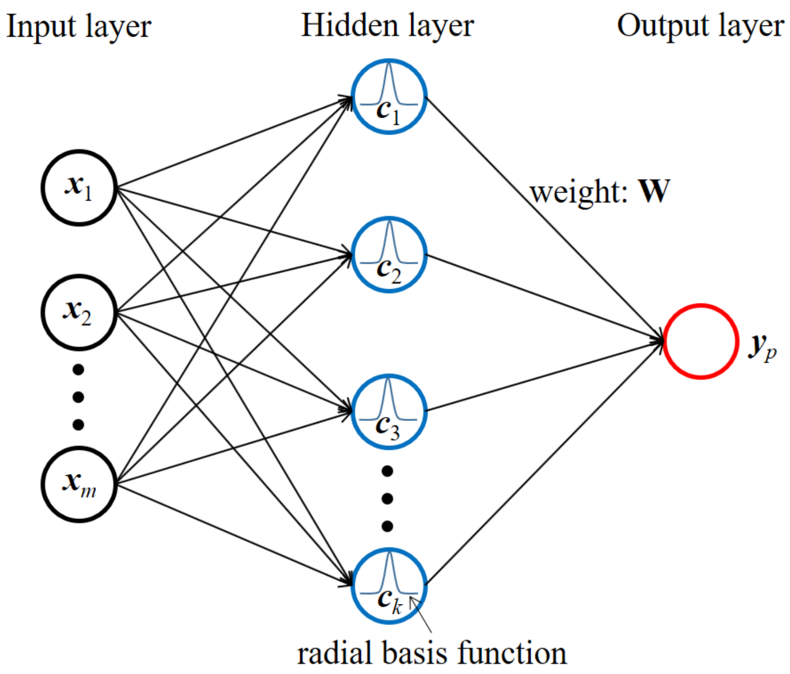

Fig. 8 Framework of radial basis function neural network

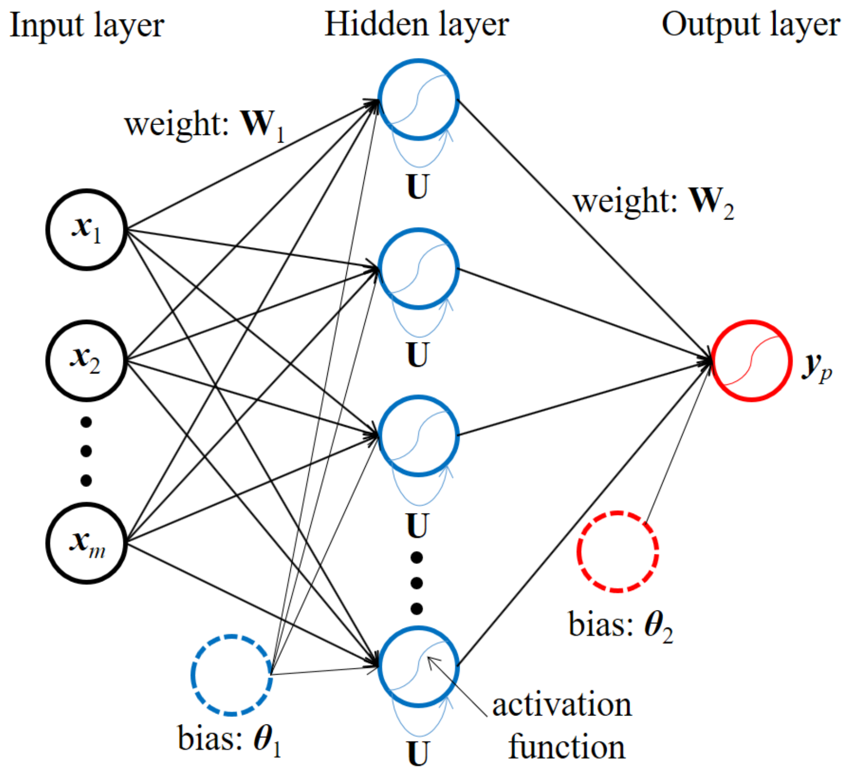

Fig. 9 Framework of recurrent neural network

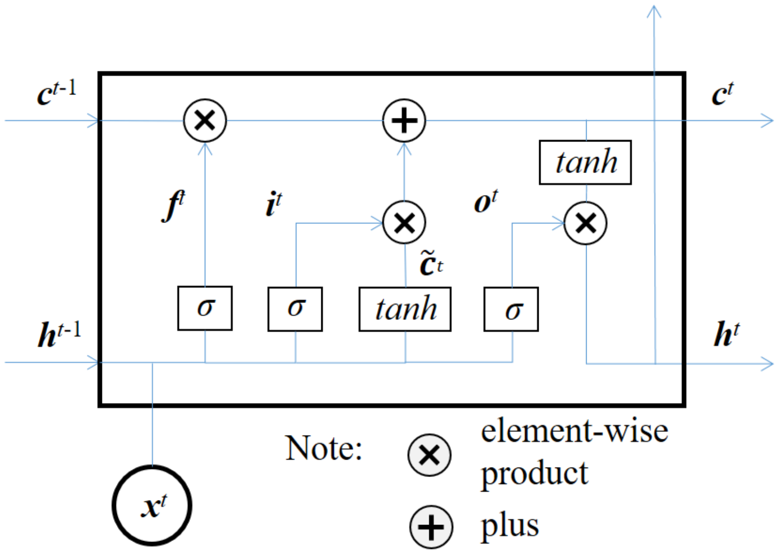

Fig. 10 Framework of memory cell of LSTM

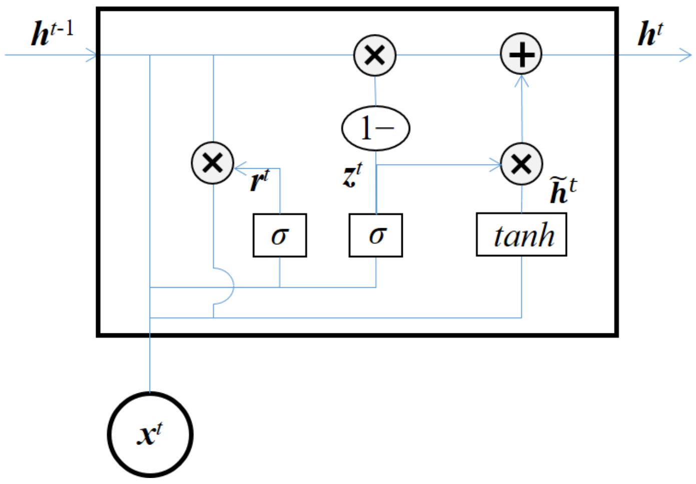

Fig. 11 Framework of memory cell of GRU

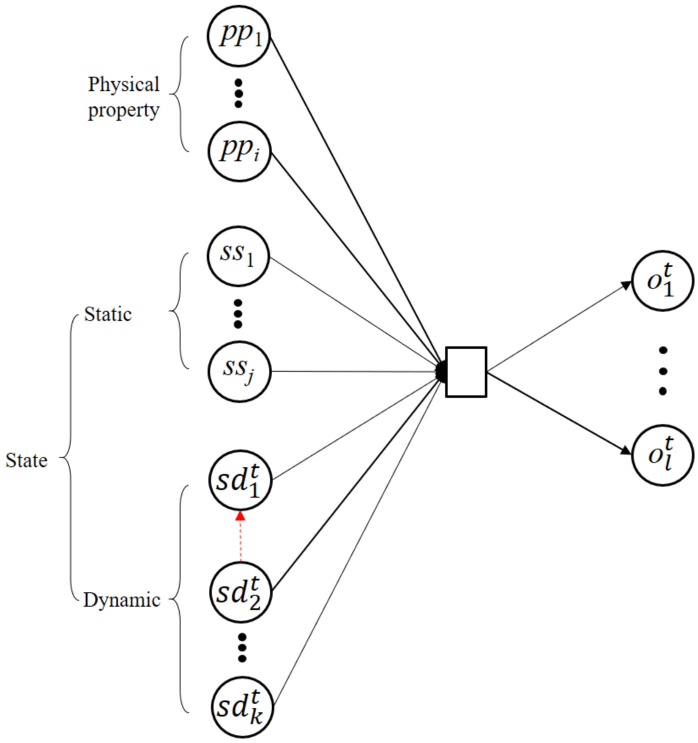

Fig. 12 Forward topology for training constitutive model of soil

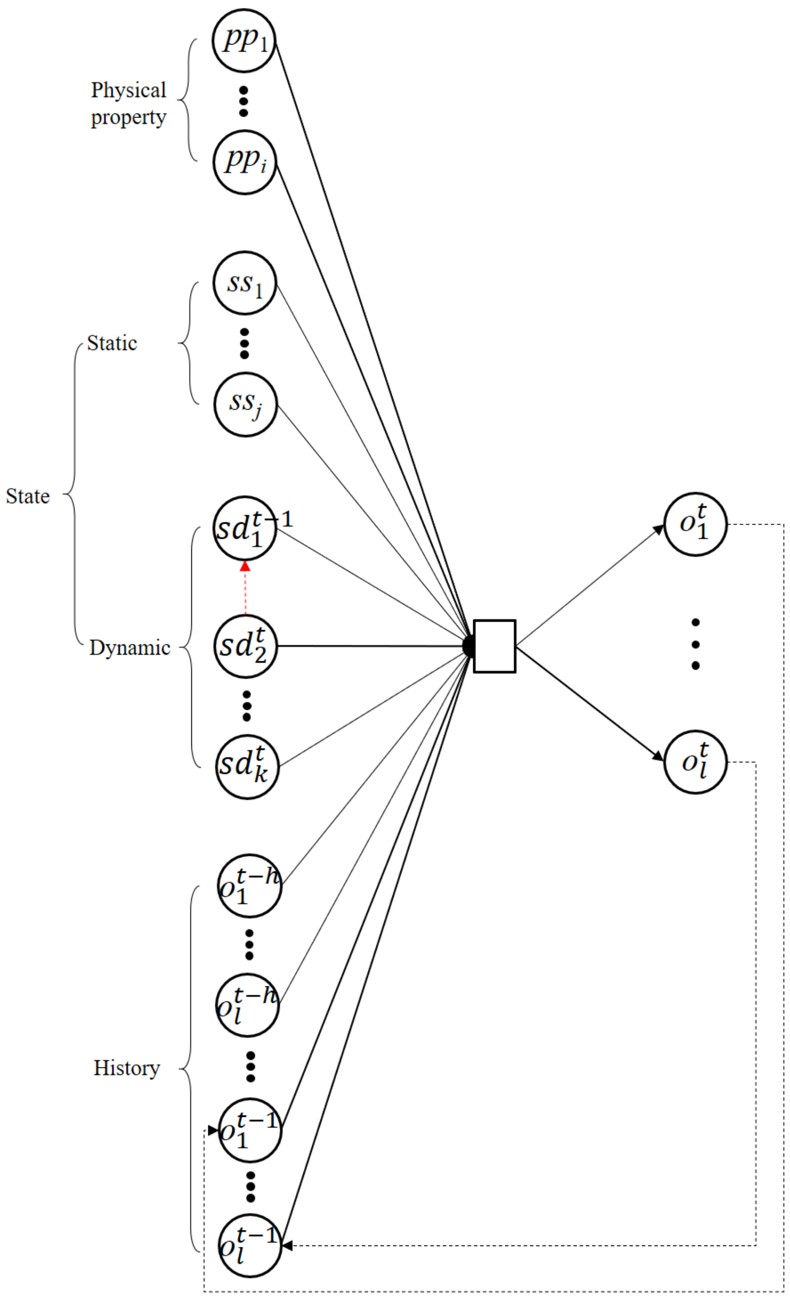

Fig. 13 Feedback topology for training constitutive model of soil

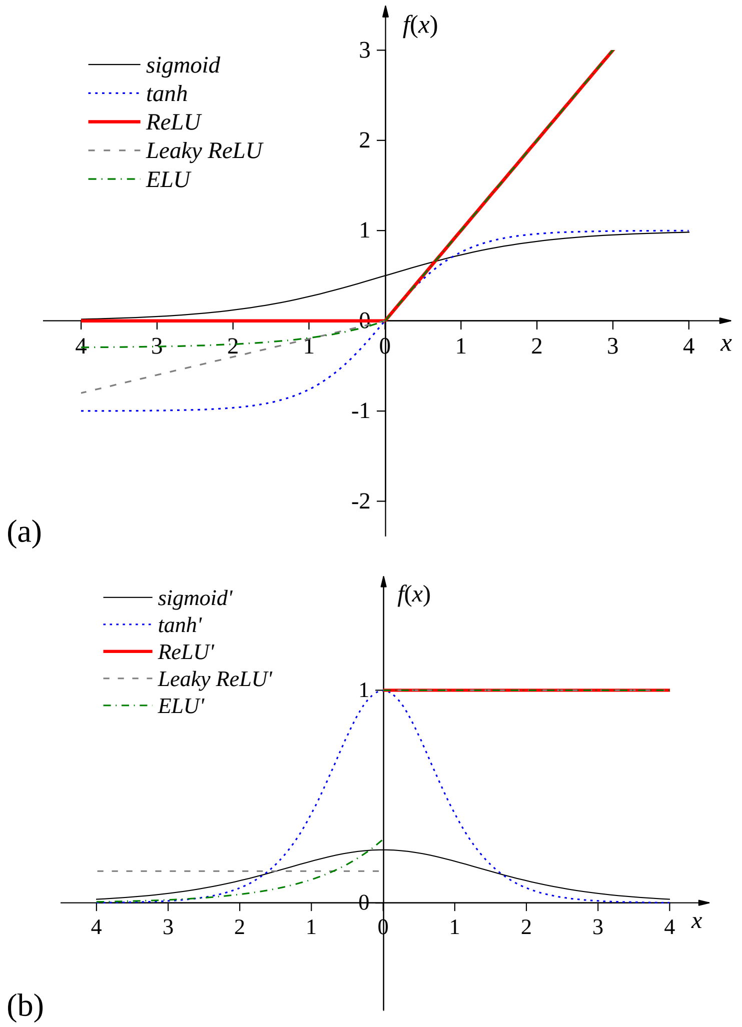

Fig. 14 Activation functions: (a) original formulation; (b) derivative

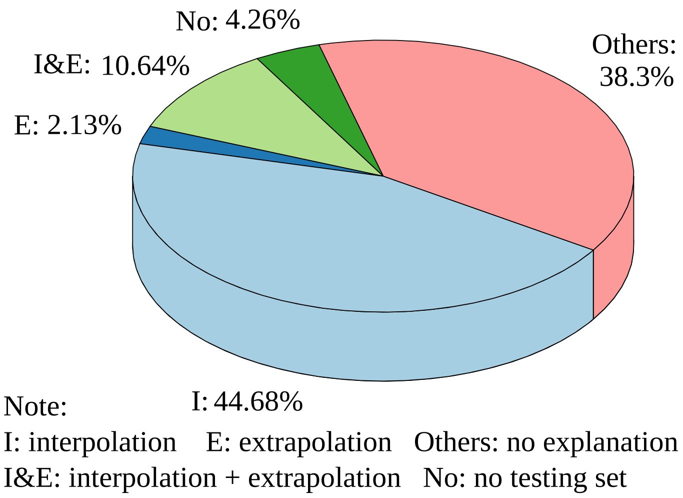

Fig. 15 Proportion of testing set type used in the training of constitutive model of soil

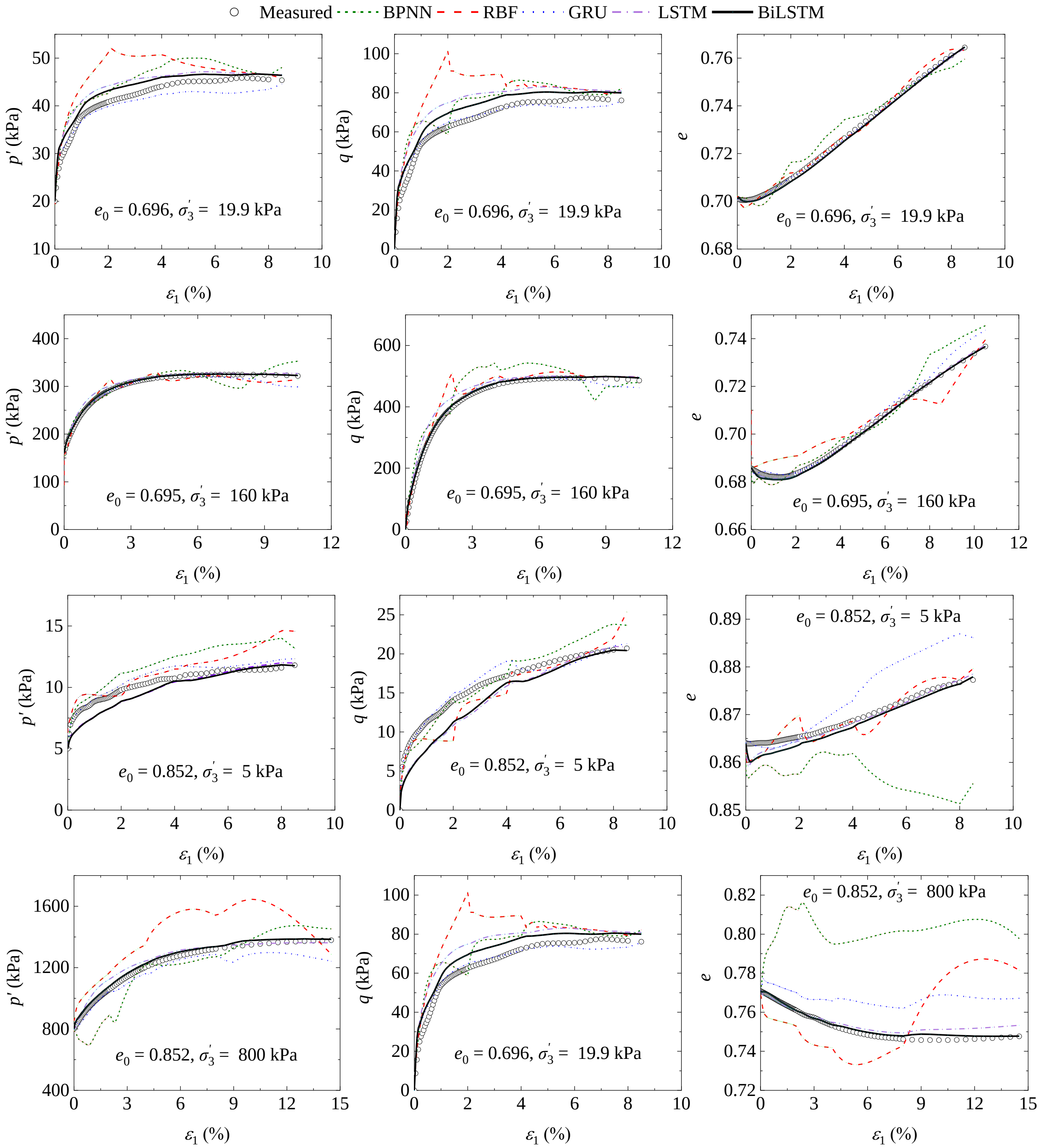

Fig. 16 Predicted stress-strain responses using four ML algorithms: (a) e_0 = 0.696, σ'_3 = 19.9 kPa; (b) e_0 = 0.695 σ'_3 = 160 kPa; (c) e_0 = 0.852, σ'_3 = 5 kPa; (d) e_0 = 0.852, σ'_3 = 800 kPa