A new CPT virtual calibration chamber in sand based on Machine learning algorithms

Mingpeng Liu[a], Enci Sun[b], Ningning Zhang[a], Fengwen Lai[c,*], Raul Fuentes[a]

[a] Institute of Geomechanics and Underground Technology, RWTH Aachen University, Germany.

[b] School of Qilu Transportation, Shandong University, Jinan 250061, China.

[c] Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, 2628 CN Delft, The Netherlands.

* Corresponding author. Email address: F.Lai-1@tudelft.nl (Fengwen Lai)

Abstract

Interpretation of Cone Penetration Tests (CPTs) still relies greatly on empirical correlations that are mostly developed in resource-demanding and time-consuming calibration chambers. This paper presents a CPT virtual calibration chamber using machine learning approaches. For such purpose, the multilayer perceptron (MLP) and long short-term memory (LSTM) neural networks are implemented to predict the cone resistance (q_c) profiles under various soil states and testing conditions. The Bayesian optimization (BO) is first adopted to find the optimal neural network hyperparameters of MLP and LSTM. Thereafter, the BO-MLP and BO-LSTM networks are trained with the available data from published datasets. Further comparison and validation of the prediction results are carried out against numerical results obtained from a Coupled Eulerian-Lagrangian (CEL) model. The results show that BO reduces the prediction error of the neural networks by 73.1% (MLP) and 59.5% (LSTM) in the training set as well as 44.4% (MLP) and 40% (LSTM) in the testing set compared to that without BO. The established machine learning models are proven competent to reproduce the q_c profiles with the coefficient of determination (R^2) of 98.65% (MLP) and 98.51% (LSTM) in the training set as well as 95.13% (MLP) and 94.65% (LSTM) in the testing

set. Apart from matching the numerical model results in terms of accuracy, the proposed methods show a much greater computational efficiency. Eventually, to showcase the use of this new virtual calibration chamber, the predicted q_c are used to obtain a new relationship to predict the relative density, D_r, of the sand. The improved correlation has an R^2 of 92.7% compared to all data, including those generated by the machine learning method and experiments, and 88.3% compared to the pure experimental data. This is a better generalization than other previously suggested relationships.

Keywords: cone penetration test, virtual calibration chamber, Bayesian optimization, multilayer perceptron neural network, long short-term memory network

1. Introduction

The cone penetration test (CPT) is one of the most common and popular in-situ test tools for site characterization [1]. One notable advantage of CPT is to quickly obtain continuous and reproducible soil testing records (cone resistance q_c and sleeve friction f_s) with minimal disturbance compared to laboratory element testing [2]. The cone resistance profile from the CPT is widely used to interpret soil properties. However, many CPT-based interpretations of soil parameters still greatly rely on empirical correlations [3]. The majority of these correlations are presented from the results of calibration chamber tests where the soil state and properties can be well-controlled [4]. However, calibration chamber testing is resource-demanding and time-consuming. As a consequence, available CPT data from calibration chamber tests is still scarce, which leads to some of the presented empirical correlations being only valid for specific soil types or specific conditions (e.g., over-consolidation ratio) [5].

To remediate the above shortage of CPT data in calibration chambers, analytical modeling (e.g., cavity expansion theory [6, 7]) and numerical simulations (e.g., Coupled Eulerian-Lagrangian (CEL) method [8, 9] and discrete-element method (DEM)

[10]) have been proposed to estimate q_c profiles for various types of soils. However, the complicated formulations contribute to a gap between analytical solutions and practical use. The numerical solutions are also notoriously difficult due to expensive computational costs and non-trivial calibration for constitutive models [11, 12], although the DEM has been previously presented as a solution for virtual calibration chambers [10].

With the advances made in hardware and software over the past decades, machine learning (ML) approaches may be an attractive alternative possessing a strong ability to utilize the raw (existing) data for prediction without any prior assumptions [13]. Multiple attempts have been made towards the application of ML-based models in geotechnical practice relevant to the CPT, e.g., prediction of bearing capacity of piles [14], soil type classification [15-17], soil parameter identification [18], and evaluation of soil liquefaction potential [19, 20]. However, for CPT profile prediction that depends largely on the depth (overburden stress) and soil type, a non-linear mapping capable of catering to sequence loading situations may be more suitable. In this regard, two possible ML options are multilayer perceptron (MLP) and long short-term memory (LSTM) neural networks [21]. Zhang et al. [22] and Guan and Yang [23] successfully applied the LSTM model to reproduce the constitutive responses of sands under both monotonic and cyclic loading. Wang and Sun [24] combined LSTM with a multiscale framework to capture the hydro-mechanical coupling effects of porous media. Habibagahi and Bamdad [25] and Kohestani and Hassanlourad [26] employed MLP to describe the mechanical properties of carbonate sand and unsaturated soil. Their results demonstrated that MLP and LSTM both have great potential in predicting nonlinear-mapping datasets. Hence, we hypothesize that MLP and LSTM neural networks are capable of reproducing and predicting CPT resistance profiles in calibration chamber tests.

The selection of hyperparameters is a critical task for the construction of ML-based models (particularly for neural networks). There are currently no well-established methods to tune the optimal hyperparameters, which means that typically these are obtained by a trial-and-error [23]. To address this drawback, some researchers [27, 28] have used Bayesian optimization (BO) that has been applied to geotechnical problems to give optimal hyperparameters for various networks. For example, Tao et al. [28] integrated Bidirection LSTM neural networks with BO to predict excavation-induced responses. Zhang et al. [29] used the BO-optimized neural network to present a modeling strategy for developing prediction models of soil properties. Hence, this paper combines the MLP and LSTM neural networks with Bayesian optimization (i.e., BO-MLP and BO-LSTM) to quickly reproduce and predict the CPT profile in calibration chamber tests, and to discuss their applications in soil interpretation based on the predicted CPT data.

The paper aims at developing accurate, and computationally efficient, virtual calibration chambers to generate q_c profiles of the CPT in sand. First, we briefly introduce MLP, LSTM, and BO algorithms to illustrate the fundamentals of Bayesian-optimized neural networks. Subsequently, the BO-MLP and BO-LSTM models are developed and trained by feeding observed data, and their performance is then evaluated by the testing dataset. The developed models are further compared with the validated solutions from the Coupled Eulerian Lagrangian (CEL) method. Eventually, an example is given to show how the developed models are used to correlate cone resistance with soil properties and to enhance the empirical equations.

2. Fundamentals of machine learning approaches

2.1 Multilayer perceptron (MLP)

The multilayer perceptron is one of the most popular artificial neural networks for

modelling and predicting complex non-linear responses and processes [26]. Fig. 1 shows a typical MLP structure consisting of one input layer, one or more hidden layers, and one output layer. A set of neurons are arranged in each layer and connected through weights and bias. The numbers of hidden layers and neurons in each layer are not constant and need to be optimized. The input data are firstly presented through the input layer and then pass through the hidden layer(s) being processed by the neural network to eventually predict values for the output layer. This process can be expressed mathematically by considering a feedforward propagation process that uses input **x** to estimate the output **y**:

$$\mathbf{h}_1 = f(\boldsymbol{W}_1\mathbf{x} + \boldsymbol{b}_1) \tag{1-1}$$

$$\mathbf{h}_2 = f(\boldsymbol{W}_2\mathbf{h}_1 + \boldsymbol{b}_2) \tag{1-2}$$

$$\mathbf{y} = \boldsymbol{W}_3\mathbf{h}_2 + \boldsymbol{b}_3 \tag{1-3}$$

The formulas (1-1) and (1-2) represent two hidden layers, where $\mathbf{h}_1$ and $\mathbf{h}_2$ are the results of the first and second hidden layer, respectively; $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{W}_3$ represent the weight matrices, and $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, $\boldsymbol{b}_3$ are the bias vectors; f represents the activation function. The $\mathbf{h}_1$ and $\mathbf{h}_2$ include the result of each neuron and $\mathbf{h}_1 = [h_{11}, h_{12}, \ldots, h_{1i}, \ldots, h_{1N\mathrm{n}1}]$ and $\mathbf{h}_2 = [h_{21}, h_{22}, \ldots, h_{2i}, \ldots, h_{2N\mathrm{n}2}]$, where $N_{\mathrm{n}1}$ and $N_{\mathrm{n}2}$ are the number of neurons of the first and second hidden layer.

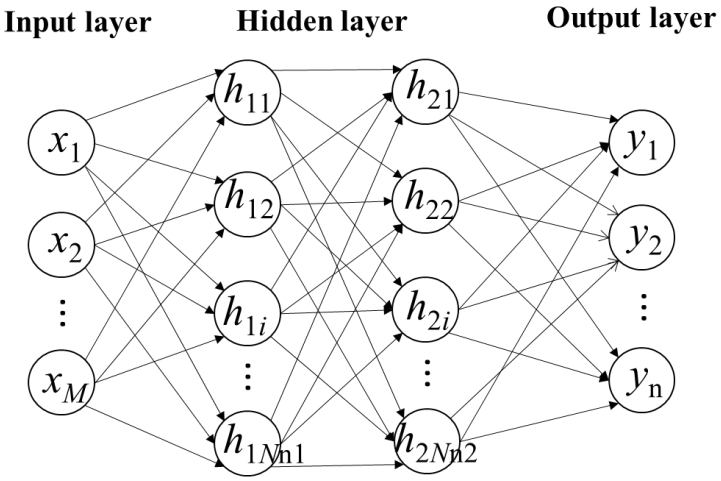


Fig. 1. Structure of a general MLP. $x_1, x_2, \ldots, x_M$ represent the elements in input vector **x**; y_1, $y_2, \ldots, y_\mathrm{n}$ represent the elements in output vector **y.**

Typically used activation functions are sigmoid, hyperbolic tangent (tanh), rectified linear unit (Relu), and leaky Relu activation functions [30]. The weight matrices $\boldsymbol{W}$ as well as the bias vectors $\boldsymbol{b}$ are randomly initialized first and then updated through a training process. The training implementation of MLP can be considered into two phases: forward calculation and backward propagations using the Back-propagation algorithm. During the forward process, the value of each hidden neuron is calculated by summing up the values of input neurons multiplied by corresponding connection weights. The error between the output and the real values can be calculated and then minimized by the backward algorithm that updates the connection weights. A full explanation of MLP neural networks can be found in the literature [31].

2.2 Long short-term memory neural network (LSTM)

LSTM is a typical class of recurrent neural networks (RNNs), which have been widely used to model time-dependent phenomena. The outputs of RNN depend on inputs of the network from not only the current time step but also previous time steps, thereby presenting the ability to predict future information related to previous inputs and enabling its application in modeling sequential problems. By introducing a memory cell in place of the neurons, LSTM can overcome the shortcoming of gradient vanishing or exploding in the back-propagation algorithm in traditional RNNs.

The typical structure of an LSTM memory cell is displayed in Fig. 2. The memory cell has three gates (forget gate $\mathbf{f}^t$, input gate $\mathbf{i}^t$, and output $\mathbf{o}^t$) which are employed to control the information flow. First, the output at the previous time step $\mathbf{h}^{t-1}$ in a memory cell and the input at current time step $\mathbf{x}^t$ are used to calculate the forget gate $\mathbf{f}^t$, input gate $\mathbf{i}^t$, output $\mathbf{o}^t$, and the storage cell $\tilde{\boldsymbol{c}}^t$, as shown in Eqs. (2-1) to (2-4). Then, the forget gate $\mathbf{f}^t$ acts on the memory cell state at the previous time step $\mathbf{c}^{t-1}$, while the input gate $\mathbf{i}^t$ acts on the storage cell at the current time step $\tilde{\mathbf{c}}^t$, as shown in Eq. (2-5). The $\mathbf{f}^t$ and $\mathbf{i}^t$

together determine whether the information should be discarded or stored and update the current memory cell state $\mathbf{c}^t$. Eventually, the output $\mathbf{o}^t$ decides the final output values $\mathbf{h}^t$, as shown in Eq. (2-6). The specific formulas are shown as follows:

$$\mathbf{i}^t = \sigma(\boldsymbol{W}_\mathrm{i}\mathbf{x}^t + \boldsymbol{U}_\mathrm{i}\mathbf{h}^{t-1} + \boldsymbol{b}_\mathrm{i}) \tag{2-1}$$

$$\mathbf{f}^t = \sigma(\boldsymbol{W}_\mathrm{f}\mathbf{x}^t + \boldsymbol{U}_\mathrm{f}\mathbf{h}^{t-1} + \boldsymbol{b}_\mathrm{f}) \tag{2-2}$$

$$\mathbf{o}^t = \sigma(\boldsymbol{W}_\mathrm{o}\mathbf{x}^t + \boldsymbol{U}_\mathrm{o}\mathbf{h}^{t-1} + \boldsymbol{b}_\mathrm{o}) \tag{2-3}$$

$$\tilde{\mathbf{c}}^t = \tanh(\boldsymbol{W}_\mathrm{c}\mathbf{x}^t + \boldsymbol{U}_\mathrm{c}\mathbf{h}^{t-1} + \boldsymbol{b}_\mathrm{c}) \tag{2-4}$$

$$\mathbf{c}^t = \mathbf{f}^t \otimes \mathbf{c}^{t-1} + \mathbf{i}^t \otimes \tilde{\mathbf{c}}^t \tag{2-5}$$

$$\mathbf{h}^t = \mathbf{o}^t \otimes \tanh(\mathbf{c}^t) \tag{2-6}$$

where $\boldsymbol{W}_\mathrm{i}$, $\boldsymbol{W}_\mathrm{f}$, $\boldsymbol{W}_\mathrm{o}$, and $\boldsymbol{W}_\mathrm{c}$ represent the weight matrices corresponding to the inputs within different gates; $\boldsymbol{U}_\mathrm{i}$, $\boldsymbol{U}_\mathrm{f}$, $\boldsymbol{U}_\mathrm{o}$, and $\boldsymbol{U}_\mathrm{c}$ represent the weight matrices corresponding to the output at the previous time step with different gates; $\boldsymbol{b}_\mathrm{i}$, $\boldsymbol{b}_\mathrm{f}$, $\boldsymbol{b}_\mathrm{o}$, and $\boldsymbol{b}_\mathrm{c}$ denote bias matrices of each gate, σ is the sigmoid activation function, and $\otimes$ signifies the element-wise product of vectors.

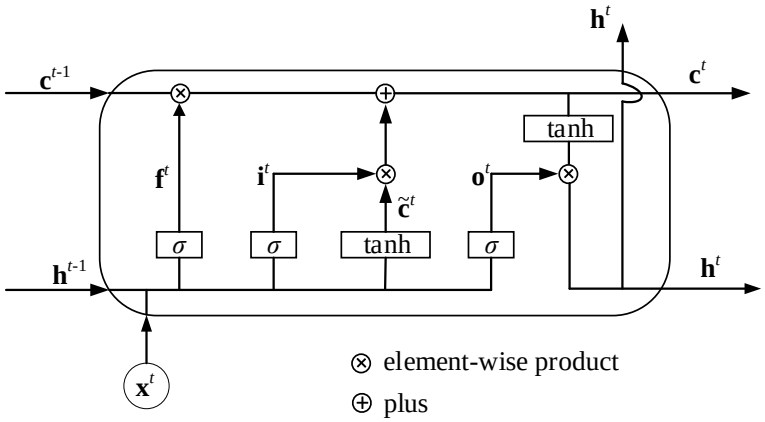


Fig. 2. Structural diagram of a LSTM memory cell

2.3 Bayesian Optimization

Although applying LSTM or MLP to predict CPT cone resistance profiles is promising, the selection of hyperparameters for a neural network often needs to be optimized by hand. To solve this problem, this paper adopts a global optimization algorithm, Bayesian optimization (BO), to select a superior combination of model

hyperparameters locating the minima of the model error. BO adopts the probabilistic surrogate model to fit real objective functions, where the next most likely point is selected for evaluation according to fitting results. The historical information is used to reasonably reduce the evaluation time and to improve search efficiency. The framework mainly includes two core parts: the probabilistic surrogate model and the acquisition function, which will be introduced in more detail in the following.

2.3.1 Probabilistic surrogate model

The surrogate model is developed using a widely-used Gaussian process [28] with the assumption that the responses obey a multidimensional normal distribution. Correspondingly, a prior distribution with the mean value being equal to 0 can be constructed by a Gaussian regression:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \sim N\left(0, \begin{bmatrix} k(x_1,x_1) & \dots & k(x_1,x_n) \\ \vdots & \ddots & \vdots \\ k(x_n,x_1) & \dots & k(x_n,x_n) \end{bmatrix}\right) \quad (3)$$

where k denotes the covariance function; x_1, …, x_n represents input values (i.e., hyperparameters in MLP and LSTM); and y_1, …, y_n are the response output values. Through the training set, the updated value y_* can be obtained through a posterior formula:

$$P(y_*|y) \sim N(K_* K^{-1} y, K_{**} - K^{-1} K_*^{\mathrm{T}}) \quad (4)$$

where K is the covariance matrix of the assumed prior distribution; K_* is the covariance matrix of the observed set and K_{**} is the covariance matrix of new-added samples. The Gaussian regression model can be continually updated through y_*:

$$\begin{bmatrix} y \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} K & K_*^{\mathrm{T}} \\ K_* & K_{**} \end{bmatrix}\right) \quad (5)$$

This process considers the relationship of y_N and y_{N+1} and builds the input-output function, providing the basics of parameter searching.

2.3.2 Acquisition function

Acquisition functions are employed to select the next probable point that enables the model's best performance. The acquisition function can obtain the posterior distribution through the observed dataset $D_{1:n}$, thereby guiding the next evaluation point x_{n+1}. Expected Improvement (EI), which is the most widely-used acquisition function, is employed in this study:

$$\alpha_n(x, D_{1:n}) = \left(v^* - \mu(x)\right)\phi\left(\frac{v^* - \mu_n(x)}{\sigma_n(x)}\right) + \sigma_n(x)\phi\left(\frac{v^* - \mu_n(x)}{\sigma_n(x)}\right) \quad (6)$$

where α_n is the expectation; v^* is the current optimum function value; ϕ is the standard normal distribution probability density function; σ and μ are variance and mean values, respectively.

2.4 Bayesian Optimized MLP/LSTM

Using BO to the selection of the hyperparameters of MLP and LSTM can be considered into the following formula:

$$x^* = \arg\min f(x), x \in X \quad (7)$$

where $f(x)$ is the objective function; x represents a group of hyperparameters; X denotes the space of hyperparameters combination; and x^* signifies the x that makes $f(x)$ obtain an optimized solution. We use the mean squared error (MSE) of the response from a neural network as the objective function f(x):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\tilde{y}_i - y_i)^2 \quad (8)$$

where n represents the number of output elements, and $\tilde{y}_i$ and y_i denotes the predicted and true values of a neural network, respectively.

In summary, the paper uses this Gaussian process to proxy the hyperparameter combination x to neural networks. The posterior distribution is obtained according to the observed dataset. Furthermore, the next evaluation point is selected by the EI acquisition function, then iteratively modifying the prior information. Thereby the

surrogate model can be improved step by step and eventually attains the optimal hyperparameter combination.

3. Implementation of BO-MLP and BO-LSTM

3.1 Data source and processing

The selected cone resistance profiles from previously reported calibration chamber tests are summarized in Table 1, in which cases 1-45 are experimental results and cases 46-64 are derived from numerical simulations. It is known that the CPT results are influenced by various factors. In this paper, five well-recognized important factors greatly affecting q_c profiles are considered: the relative density D_r, effective vertical stress σ_v, lateral earth pressure coefficient K_0, saturation condition (dry or saturated), and boundary condition (BC) [32, 33]. The five commonly used boundary conditions (BC1 - BC5) were further summarized in Table 2. Of note, these boundary conditions are represented with numbers 1 – 5 in the dataset, respectively. It is assumed that the above five influencing factors of CPT in calibration chambers are considered constant throughout the penetration process [33].

One essential requirement for applying neural networks is the one-to-one mapping relationship between input data and output data. For the two neural networks MLP and LSTM, the input data consisted of the above variables, as well as the depth z^t at the current step t, while the output was the current-step cone resistance q_c^t. In this way, the q_c - z profiles were processed into the corresponding relationship. For each q_c curve, the initial whole profile was processed into 100 points using interpolation. As can be seen in Table 1, a total of 64 groups of data were used for training and testing the network. 52 groups (5200 points) were employed to train the machine learning network, while the rest 12 groups (1200 points) with the "*" symbol in Table 1 were selected as the testing dataset to evaluate whether the model overfits or underfits. Note that the testing

252 set has considered different relative densities, vertical stresses, and K_0 conditions, as
253 well as saturation and boundary conditions.

254 Table 1. Summary of used cases of calibration chamber test

Case No.	D_r	σ_v (MPa)	K_0	Saturation condition	Boundary condition (BC)	Reference
1	0.878	0.100	0.450	Dry	5	Kluger et al. [34]
2	0.748	0.200	0.450	Dry	5	
3	0.838	0.100	0.450	Saturated	5	
4*	0.78	0.200	0.450	Saturated	5	
5	0.952	0.100	0.450	Dry	5	
6	0.97	0.100	0.450	Saturated	5	
7	0.668	0.200	0.450	Saturated	5	
8*	0.606	0.200	0.450	Dry	5	
9	0.918	0.200	0.450	Saturated	5	
10	0.918	0.200	0.450	Dry	5	
11	0.650	0.056	0.393	Dry	5	Huang and Hsu. [33]
12	0.840	0.056	0.393	Dry	5	
13	0.250	0.160	0.463	Dry	5	
14	0.500	0.160	0.463	Dry	5	
15*	0.650	0.160	0.463	Dry	5	
16	0.840	0.160	0.463	Dry	5	
17	0.500	0.056	0.786	Dry	1	
18	0.500	0.056	0.786	Dry	5	
19*	0.650	0.056	0.786	Dry	1	
20	0.650	0.056	0.786	Dry	5	
21	0.840	0.056	0.786	Dry	1	
22	0.840	0.056	0.786	Dry	5	
23	0.500	0.070	0.400	Dry	1	Lech Bałachowski [35]
24*	0.500	0.100	0.400	Dry	1	
25	0.500	0.150	0.400	Dry	1	
26	0.500	0.200	0.400	Dry	1	
27	0.500	0.350	0.400	Dry	1	
28	0.500	0.400	0.400	Dry	1	
29	0.800	0.050	0.400	Dry	1	
30	0.800	0.070	0.400	Dry	1	
31	0.800	0.100	0.400	Dry	1	
32	0.800	0.150	0.400	Dry	1	
33	0.800	0.250	0.400	Dry	1	
34*	0.800	0.300	0.400	Dry	1	
35	0.800	0.400	0.400	Dry	1	

36	0.800	0.050	0.400	Dry	3	
37	0.800	0.070	0.400	Dry	3	
38*	0.800	0.150	0.400	Dry	3	
39	0.800	0.200	0.400	Dry	3	
40*	0.800	0.250	0.400	Dry	3	
41	0.330	0.025	1.000	Saturated	1	Pournaghiazar et al. [5]
42	0.330	0.050	1.000	Saturated	1	
43*	0.330	0.100	1.000	Saturated	1	
44	0.610	0.030	1.000	Saturated	1	
45	0.610	0.050	1.000	Saturated	1	
46	0.718	0.150	0.500	Dry	3	Chen et al. [4]
47*	0.615	0.150	0.500	Dry	3	
48	0.395	0.150	0.500	Dry	3	
49	0.231	0.150	0.500	Dry	3	
50	0.23	0.100	1.000	Dry	3	Schnaid. [36]
51*	0.63	0.100	1.000	Dry	3	
52	0.86	0.100	1.000	Dry	3	
53	0.65	0.075	0.500	Dry	3	
54	0.65	0.150	0.500	Dry	3	
55	0.65	0.300	0.500	Dry	3	
56	0.752	0.060	1.000	Dry	1	Arroyo et al. [37]
57	0.752	0.100	1.000	Dry	1	
58	0.768	0.200	1.000	Dry	1	
59*	0.776	0.300	1.000	Dry	1	
60	0.784	0.400	1.000	Dry	1	
61	0.907	0.100	1.000	Dry	1	
62	0.914	0.140	1.000	Dry	1	
63	0.922	0.200	1.000	Dry	1	
64	0.929	0.300	1.000	Dry	1	

Note: '*' represents the testing dataset.

Table 2. Five boundary conditions in the calibration chamber

Boundary conditions	Top and bottom boundary		Lateral boundary	
	Stress	Strain	Stress	Strain
BC1	Constant	-	Constant	-
BC2	-	0	-	0
BC3	Constant	-	-	0
BC4	-	0	Constant	-
BC5	Constant	-	Servo–controlled	

It is known that the quality of input data could significantly affect the predictive performance of machine learning models. Normalization is preferably adopted to rescale variables with different scales to lower their influence on the model performance and also reduce computational costs. The following equation was used to normalize the input data to the common range of 0 ~ 1:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (9)$$

where x is the raw input variables before normalization, x_{norm} is the input variables after normalization, x_{min} and x_{max} are the minimum and maximum values of the input variables, respectively. Besides, in many cases, especially DEM simulations [37], the q_c profile usually exhibited large oscillations, which can decrease the quality of training data. Therefore, a sliding window approach was used to smooth data before input into neural networks and therefore reduce the fluctuation in q_c curves to all groups [38]. The value of smoothed x_s can be calculated by:

$$x_s = \frac{1}{t}\sum_{i=n-t+1}^{n} x_i \quad (10)$$

where t = window size. The average value of datasets within a window is assigned as the new value of the studied parameter. The datasets within a window consist of current and former (t–1) values. It should be noted that the first (t–1) and the last (t–1) points cannot form a complete window, thereby values of such points maintain constant. Larger window size can generate a smoother sequential curve, but it is much more likely to deviate from the original curve. The window size is thus set as four in this study for maintaining the reliability and smoothness of the datasets.

3.2 Neural network structures

3.2.1 MLP

The MLP structure in this paper contains 4 layers: one input layer, two hidden layer layers, and one output layer, as shown in Fig. 1. The main hyperparameters of the MLP are listed in Table 3. For training the MLP, three important hyperparameters are to be

optimized using BO: the number of nodes in the first and second hidden layer (N_{n1} and N_{n2}) as well as the initial learning rate (η_{MLP}). Furthermore, we set the range of N_{n1} and N_{n2} as 1-20 to decrease the complexity of the network structure, thereby preventing overfitting in the training set. The Levenberg-Marquardt algorithm is employed to optimize the weight and bias matrices [39], and the Tanh activation function is also adopted for hidden layers. The maximum number of training epochs was chosen as 200, but we also used the early-stop method to prevent overfitting, by which the training process will stop if the normalized MSE (MSE after data normalization) reaches the target precision of 5E-4. The two neural networks are implemented in Matlab R2022b toolboxes.

Table 3. Main hyperparameters of MLP network.

Hyperparameter	Description	Value
N_h	Number of hidden layers	2
N_{n1}	Number of nodes in the first hidden layer	[1, 20]$^{\#}$
N_{n2}	Number of nodes in the second hidden layer	[1, 20]$^{\#}$
Optimizer	Algorithm for optimizing weights and biases	Levenberg–Marquardt
MaxEpochs	Maximum number of rounds used for training	200
η_{MLP}	Initial learning rate	[1E-5, 1E-2]$^{\#}$

Note: "#" represents the hyperparameters to be optimized by Bayesian Optimization.

The discrepancy between prediction values and target values was measured by using the mean squared error (MSE, see Eq. 8) as the loss function. Besides, the fitting effect is also quantified by the coefficient of determination:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{11}$$

where $\bar{y}$ is the mean true value and other parameters are the same as in Eq. (8).

3.2.2 LSTM

The LSTM structure contains four layers: a sequence input layer, one LSTM layer, followed by a fully connected layer, and eventually the regression layer. The main hyperparameters of the LSTM are listed in Table 4. Herein, three hyperparameters: the

number of nodes in the hidden layer, the initial learning rate, and the L2 regularization parameter are determined with the aid of BO. The optimization ranges are also given in Table 4. During the optimization process, Relu was chosen as the activation function for the LSTM layer, and the adaptive moment estimation (Adam) optimizer is utilized due to its superiority [40]. The batch size used for each training iteration is set to 100 because every q_c profile has 100 points [38]. The number of epochs needs to be sufficiently large to ensure the loss value that can converge at a constant value. However, due to the use of BO in this paper, the epoch should not be too large to prevent overfitting and therefore was set to 300 [22, 23, 38].

Table 4. Main hyperparameters of LSTM model.

Hyperparameter	Description	Value
N_h	Number of hidden layers	1
N_n	Number of nodes in the hidden layer	[1, 20]#
Optimizer	Algorithm for optimizing weights and biases	Adam
η_{LSTM}	Initial learning rate	[1E-4, 1E-2]#
L2	L2 regularization parameter	[1E-5, 1E-2]#
MaxEpochs	Maximum number of rounds used for training	300
N_b	The batch size used for each training iteration	100

Note: "#" represents the hyperparameters to be optimized by Bayesian optimization.

3.3 BO-MLP/BO-LSTM modelling

Fig. 3 presents a flowchart showing the BO-MLP and BO-LSTM modelling process. It can be described by six main steps:

- Step 1: Obtain the raw datasets under different soil and penetration conditions, as presented in Section 3.1. Then, normalize the dataset, and separate it between training and testing datasets.
- Step 2: Establish the MLP and LSTM neural networks. Then determine the hyperparameters of network models and set the range of hyperparameters to be optimized. The details for this step can be found in Section 3.2.
- Step 3: Build the Bayesian optimization model. Neural network models are regarded as the objective function in optimization, while the MSE given in Eq. (8)

is determined as the evaluation function.

- Step 4: Training the neural network under the current combination of hyperparameters with the training set. The value of the evaluation function under the current combination of hyperparameters is calculated and then returned to BO. According to the probabilistic surrogate model and acquisition function, the next group of hyperparameters is selected for a new round of training until reaching the maximum number of iterations.
- Step 5: Output the combination of hyperparameters with the best model performance after BO. Then use these hyperparameters to train the neural network.
- Step 6: Input the testing dataset into the trained network to evaluate the working performance of the established neural network.

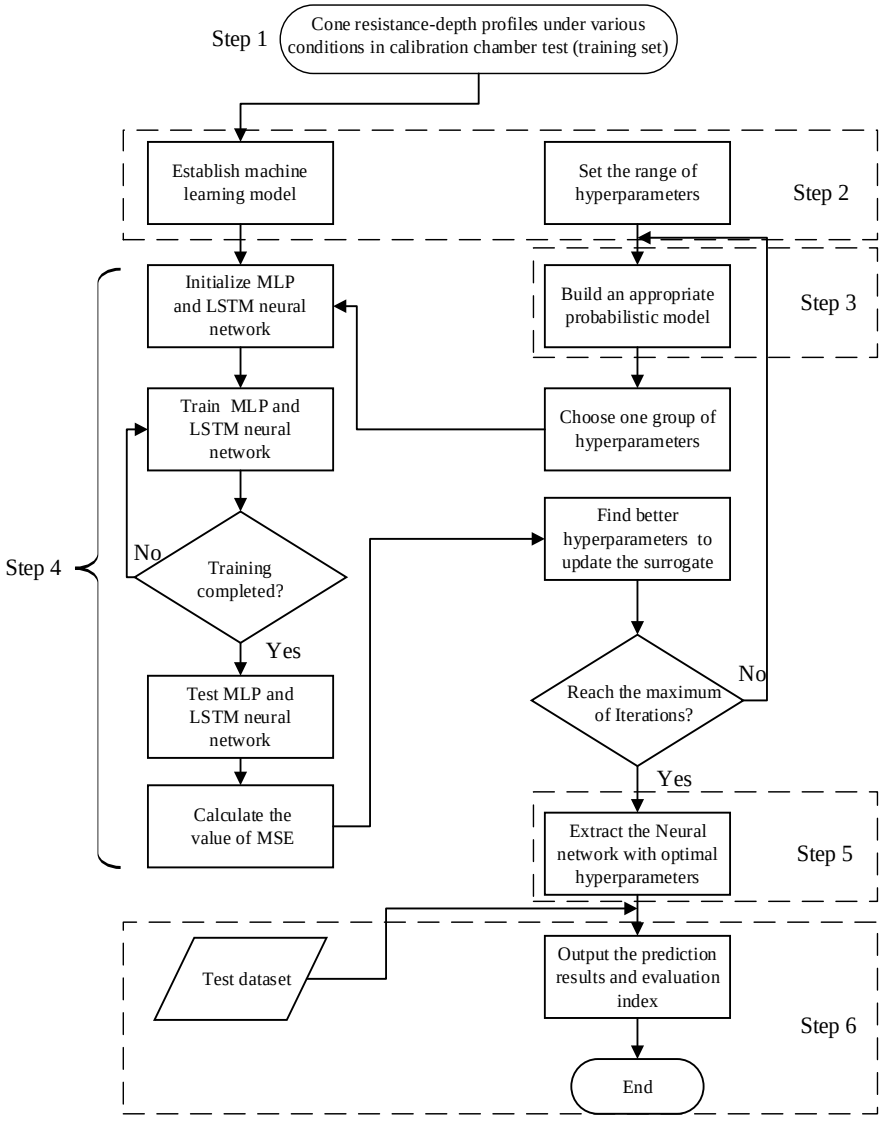


Fig. 3. Flow chart of prediction procedure for BO-MLP/BO-LSTM model.

4. Performance of the machine learning models

4.1 Effectiveness of Bayesian optimization

When using BO to optimize the neural networks, the iteration number of the objective function is set to 10 for preventing overfitting [41]. The process is displayed in Fig. 4. With the process of Bayesian optimization, the minimum objective value (i.e., the MSE of the neural network) decreases to a small value, signifying a good performance achieved by using BO. The combination of hyperparameters, after optimization, for the MLP neural network was determined to be N_{n1} =8, N_{n2} = 4, η_{MLP} = 1.31E-3, while for the LSTM neural network they are N_n = 16, η_{LSTM} = 5.2E-3, L2 = 1.2E-3. These hyperparameters were then used to train the neural networks.

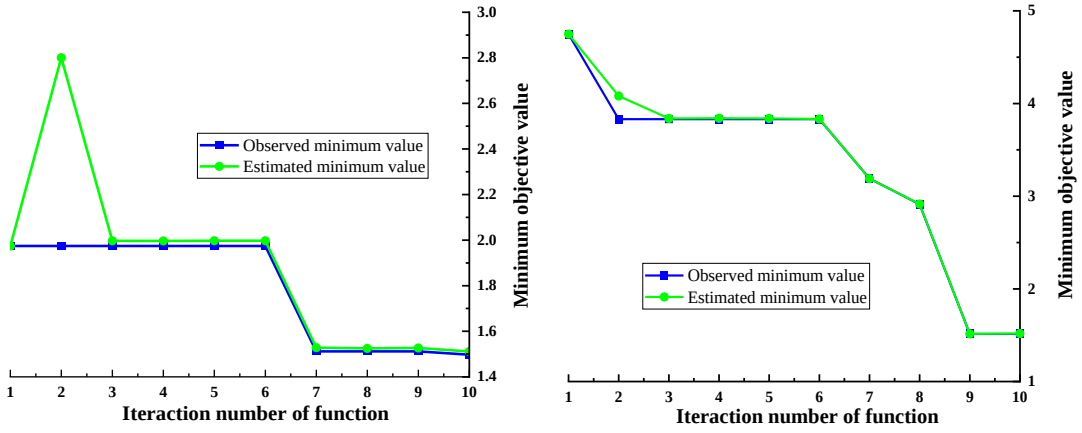
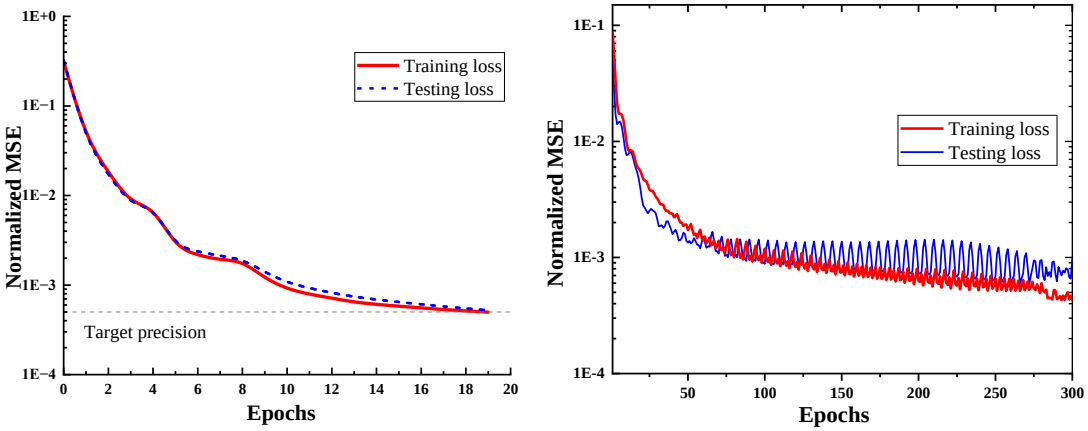


Fig. 4. Process of Bayesian optimization for (a) MLP model and (b) LSTM model

The normalized loss function MSE of the two models with the optimized hyperparameters are shown in Figs. 5(a) and 5(b), respectively. For both models, the normalized MSE rapidly reduces to a relatively small value. For MLP, the normalized MSE reaches the target precision at epoch 19, which mainly benefits from the adopted Levenberg-Marquardt algorithm that can rapidly converge to optimal solutions [39]; the normalized MSE of the LSTM reaches a relatively constant value at the end of

training. The evaluation indexes of BO-MLP and BO-LSTM are listed in Table 5. Besides, the MSE and R^2 of MLP and LSTM without optimization are also shown in Table 5, in which the errors of neural networks achieve an obvious reduction while the fitting degree R^2 has an increase after the Bayesian optimization. The MSE reduces by 73.1% (MLP) and 59.5% (LSTM) in the training set as well as 44.4% (MLP) and 40% (LSTM) in the testing set; the R^2 increases by 0.83% (MLP) and 0.88% (LSTM) in the training set, and 1.05% (MLP) and 0.76% (LSTM) in the testing set. The loss functions and evaluation indexes show that the Bayesian optimization performs well in selecting hyperparameters for the neural networks.



(a) (b)

Fig. 5. Evolution of normalized MSE against epochs during model training for (a) MLP model and (b) LSTM model.

Table 5. Evaluation index for different models

Model	MSE (MPa)		R^2	
	Training	Testing	Training	Testing
BO-MLP	2.0006	4.8272	0.9865	0.9513
BO-LSTM	2.2200	5.0109	0.9851	0.9465
MLP	3.4720	6.9720	0.9782	0.9408
LSTM	3.5412	7.0001	0.9763	0.9389

To further evaluate whether the optimized hyperparameters in neural networks are suitable to the used cases, Fig. 6 shows the variation of R^2 with the number of data used in neural networks. It can be observed that R^2 in the training set is maintained at a high

level exceeding 95% for two neural networks. However, the R^2 of the testing set increases with the number of used data from a low level (59.6% and 51.5%) to a relatively high level of more than 90%. But the used cases 41 to 55 (corresponding to 4100-5500 on x-axial in Fig. 6) decrease the R^2 of the training set, which may attribute to that the results of cases 41 to 55 are not so identical to other research, thereby reducing the overall quality of the dataset. For the eventually used 64 cases, the R^2 of both training and testing sets reach the highest level, indicating that the adopted cases in this study are adaptive to the constructed neural networks.

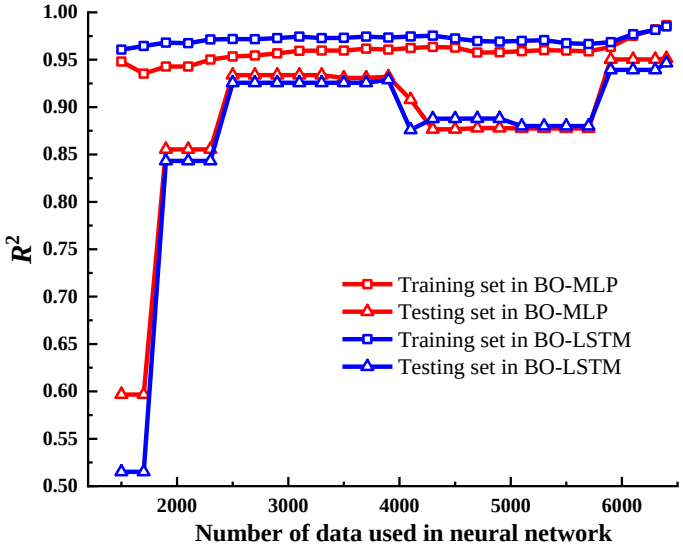


Fig. 6. R^2 of training and testing sets with the number of data used in neural networks

4.2 Evaluation of the developed models

To evaluate the performance of the established machine learning models, the predicted q_c values obtained from the two neural networks versus the measured data of all the 64 groups are presented in Fig. 7 for the BO-MLP and BO-LSTM models. As expected in the training set, the predicted q_c values from the neural networks are close to the measured values. The R^2 of the training set is 98.65% and 98.51%, respectively for MLP and LSTM models. Besides, the regression errors MSE in the training set are 2.0006 and 2.2200 for the two networks, respectively. The R^2 values for the evaluation against the unseen testing set are 95.13% and 94.65%, respectively for MLP and LSTM, while the MSE values are 4.8272 and 5.0109, respectively. In summary, the regression

plots in Fig. 7 show that the MLP and LSTM neural networks both perform well in predicting cone resistances among the selected raw datasets. But in some cases, the predictions still have some discrepancies with the measured values. A reasonable explanation might be that the fitting degree of neural networks cannot reach 100% and thereby hard for the ML model to fit all discrepancies [38]. This means there are some prediction points deviated from the original results.

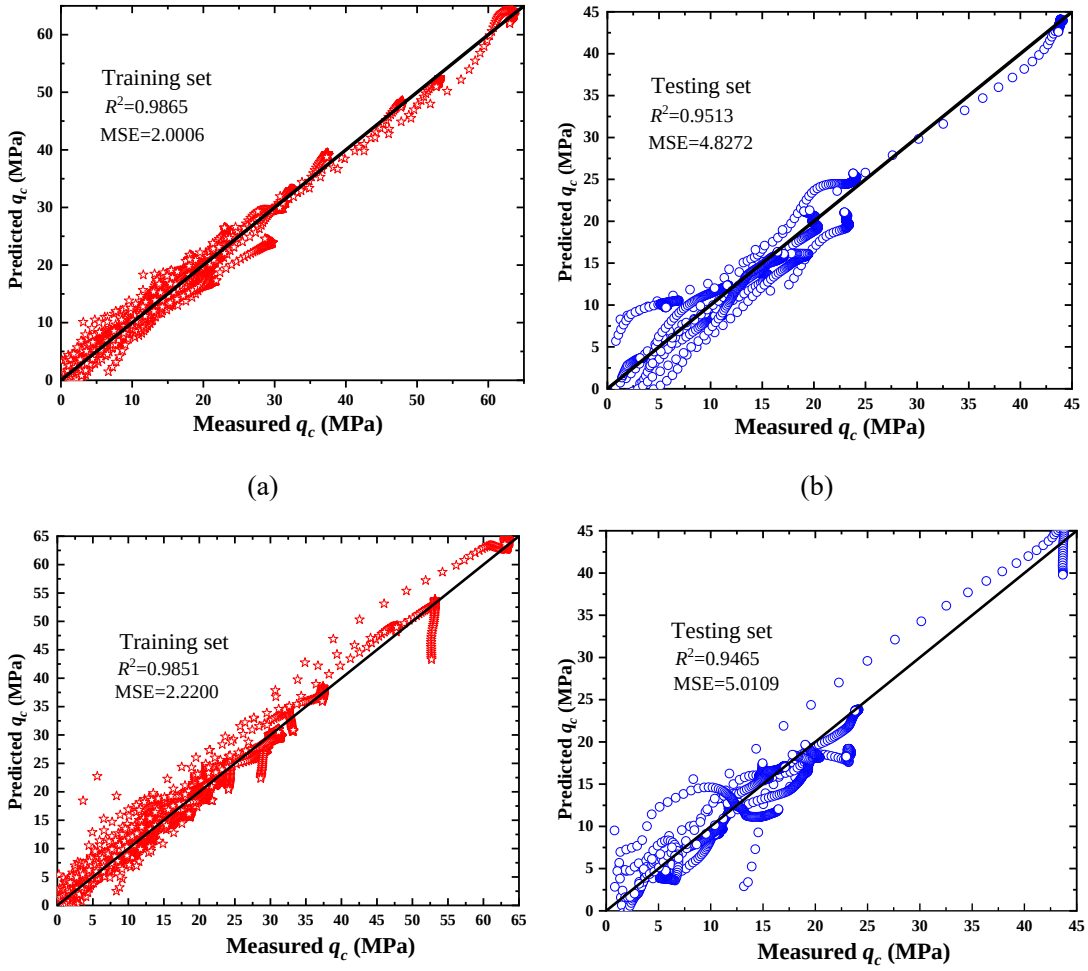


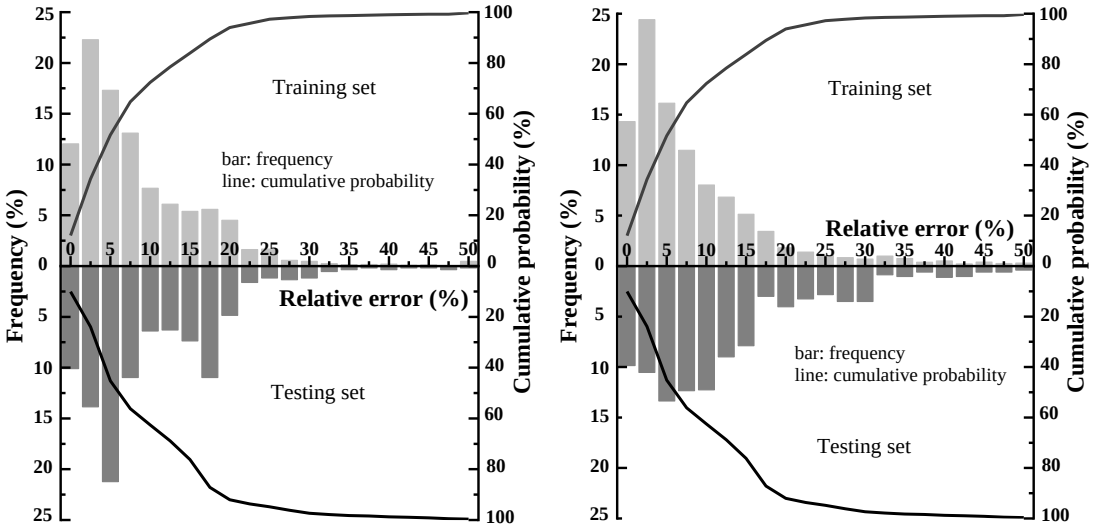
Fig. 7. Measured cone resistances against predicted values from the two neural networks in (a) training set in BO-MLP network, (b) testing set in BO-MLP network, (c) training set in BO-LSTM network, and (d) testing set in BO-LSTM network

To evaluate the performance of developed models more comprehensively, the relative error (RE) is employed to reflect the deviation percentage between measured values

and predictions:

$$\mathrm{RE} = \left|\frac{\tilde{y}_i - y_i}{y_i}\right| \quad (12)$$

where the parameters are the same as in Eq. (8). Fig. 8 depicts the frequency and cumulative probability of relative error distribution for all data in BO-MLP and BO-LSTM models. In the training set of two neural networks, about 13.5% of total points show a high convergence around the standard line of 0% error. This value is about 10% for the testing set. The relative errors under 7.5% account for the majority of datasets with a cumulative probability larger than 60% for both training and testing sets. Besides, most (about 95%) of the total points show a relative error lower than 20%, proving that the prediction by the ML-based models can be highly accepted for calibration chamber tests. However, there are still a small number of points with errors greater than 30%. These are believed to be within the discrepancies between different scholars even under similar experimental situations. For example, cases 38 and 46 in Table 1 have similar experimental conditions, but the stable q_c of case 38 is 11.4 MPa while in case 46 the value is 21.5 MPa. This discrepancy would decrease the precision of machine learning to the overall dataset, thereby causing a large error. Only if other variables that explained this discrepancy were included (i.e., instrument accuracy in the measurements) the machine learning algorithm could be improved. This is however impractical within the scope of this paper.



(a) (b)

Fig. 8. Frequency and cumulative probability of relative error in (a) BO-MLP network and (b) BO-LSTM network.

Fig. 9 shows the comparison between the predicted q_c profiles by MLP and LSTM neural networks with the experimental data. Every subplot in Fig. 9 shows three good predictions (red lines) and three worst cases (black lines). It can be seen from Fig. 9 that both in the training and testing set, the variation tendency of cone resistance with the normalized depth z/z_{total} (ratio of the current penetration depth over the total penetration depth) can be successfully reproduced by two neural networks during penetration processes. Given that coefficient of determination R^2 has reached relatively large values for training and testing datasets, many cases can be well predicted by neural networks. But some cases still display obvious errors between measured values and predictions, especially for these worst cases, as displayed in the black lines in Fig. 9. Such as the example of case 43 in Fig. 9(b), the true stable q_c is about 5 MPa while the prediction reaches 10 MPa, resulting in the relative error of 50%. These worst cases lead to the RE in Fig. 8 larger than 30%. Besides, the cone resistance at the bottom of penetration, in some cases, presents the tendency of reduction, as displayed in cases 16 and 58 in Fig. 9(c). The main reason behind this phenomenon might be that some calibration chamber tests with flexible bottom boundaries cause a decrease in cone resistance around the bottom of the chamber [34]. The established neural networks learned this feature, thereby causing the resistance reduction at the bottom of penetration in some cases. In addition, some experimental results like case 8 show large errors and singular points where the q_c curve did not start from the original point, which may explain the points outside the 30% error in Fig. 8. In summary, the established machine learning models are proven capable of predicting the cone resistance profiles in calibration chamber test with an acceptable fitting degree.

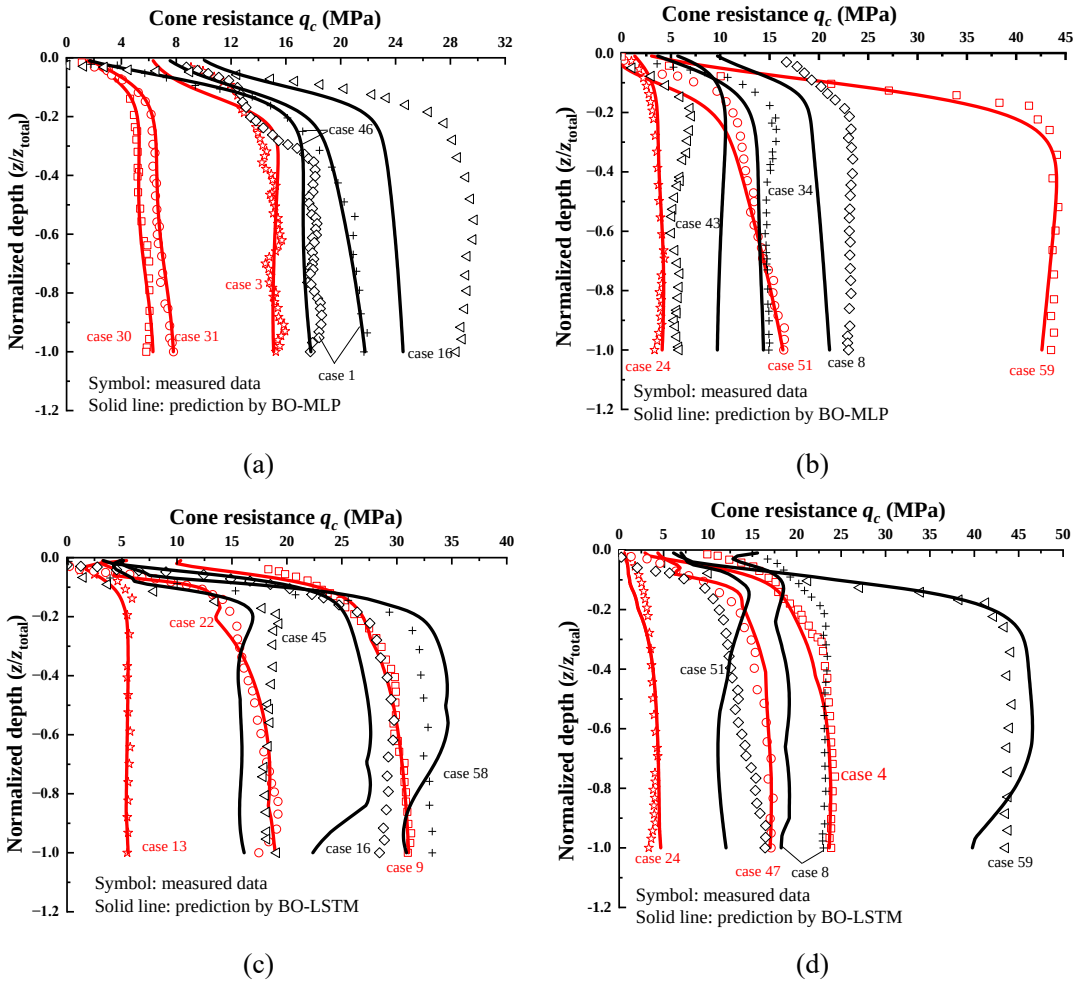


Fig. 9. Comparison of measured data with prediction profiles from ML model in (a) training set of BO-MLP; (b) testing set of BO-MLP; (c) training set of BO-LSTM; (d) testing set of BO-LSTM. Red lines represent better predictions by neural networks while black lines represent the worst cases.

5. Further validation and application of the trained models

5.1 Validation with numerical solutions

This section aims to further compare the neural networks with numerical solutions for new (not given in Table 1) cases. These new cases are designed to consider the variation of relative density, vertical effective stress, and saturation conditions of sand, as listed in Table 6. Numerical simulations to install the CPT in the calibration chamber for these cases are then performed to provide cone resistance profiles. Details on the CEL model

setup and soil constitutive relationships are shown in Appendix A and B, respectively.

Table 6. New cases to be validated by numerical modelling.

No.	D_r	σ_v (MPa)	K_0	Saturation condition
1	0.563	0.13	0.42	Dry
2	0.82	0.15	0.45	Saturated
3	0.734	0.09	0.5	Dry
4	0.8	0.16	0.38	Dry

The CEL model is first compared and validated against calibration chamber tests conducted by Kluger et al. [34] in Ticino sand. Their testing results under two vertical effective stresses σ_v of 100 kPa and 200 kPa are shown in Fig. 10. It can be seen from Fig. 10 that the numerical solutions are in good agreement with testing results, giving confidence that the CEL modelling can provide the reliable q_c profiles in calibration chamber tests.

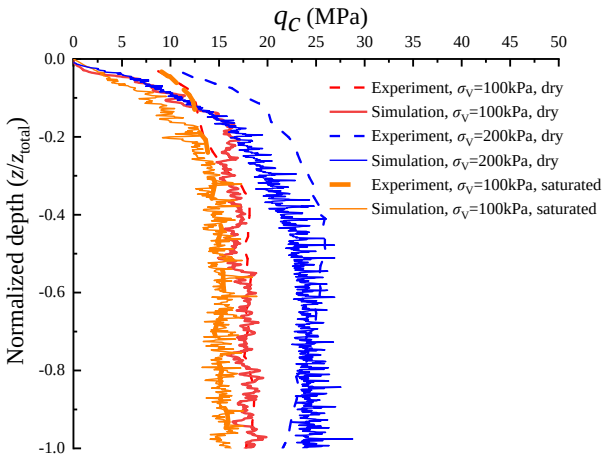


Fig. 10. Comparison of the simulated cone resistance with the experimental results of Kluger et al. [34].

Fig. 11 shows the comparisons of cone resistance profiles obtained from the two neural networks and the numerical simulation for the artificial cases in Table 6. The q_c profiles obtained from the machine learning models are fair close to the numerical simulations, although some cases like No. 2 in Table 6 still show discrepancies between the two methods. However, the stable values of cone resistance achieve a better agreement

between the numerical simulation and the neural networks. One has to be aware that the application of the validated models still requires complex model set-up and time-consuming running. In contrast, a well-trained neural network can quickly (typically a few seconds) create reliable q_c curves under certain soil conditions.

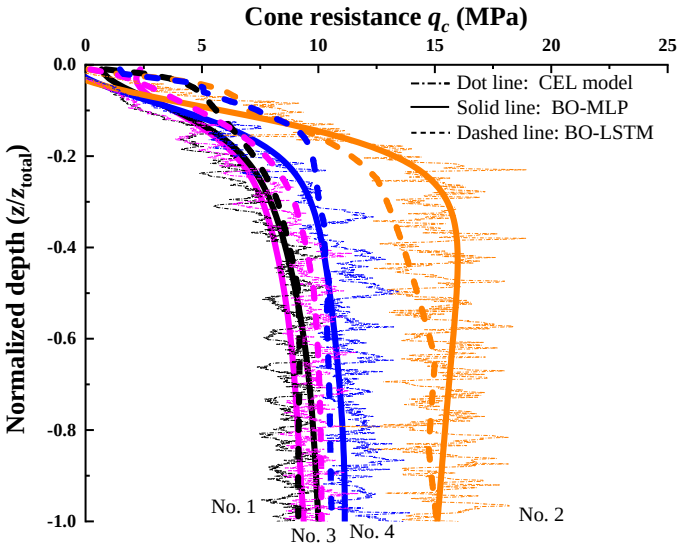


Fig. 11. Comparison of neural networks with CEL model in predicting unseen cases.

5.2 Application: relating q_c with soil properties

As mentioned before, most of the existing correlations between q_c and soil properties (e.g., relative density D_r) were presented through a limited number of datasets for specific soil types. The trained neural networks have been proven as a reliable approach to quickly extend the database of CPT in calibration chambers, therefore the corrections of these correlations can be made based on the extended database based on the machine learning models. This section shows an example to potentially improve the D_r - q_c correlation using the neural networks developed in this study.

Table 7 summarizes four reported correlations of D_r - q_c. Here, the normalized cone resistance of $Q = (q_c-\sigma_v)/\sigma_v$ was introduced to consider the influence of vertical stress whenever available. Fig. 11 presents the deduced correlations of D_r - Q from Eqs. (13) to (16). The 864 groups of D_r - Q generated from the developed neural networks are also shown in Fig. 12. Note that only the constant q_c values from cone resistance profiles

are used to relate soil properties, and the ML results are the average values of BO-MLP and BO-LSTM. The previous experimental results of Bolton and Gui [42] and Jamiolkowski et al. [43] are also presented in Fig. 12. The existing empirical formulas often only match with the corresponding experimental data used for fitting but deviate from others. Conversely, the predictive database from neural networks has considered different soil types and test conditions (see Table 1).

Table 7. Summary of representative correlations of D_r-q_c-σ_v

Empirical formula		Empirical coefficient	Reference
$D_r = \frac{1}{C_2}\ln\left[\frac{q_c}{C_0(\sigma_v)^{C_1}}\right]$	(13)	C_0 = 60, C_1 = 0.7, C_2 = 2.91	Lunne and Christoffersen (1983) [44]
$D_r = C_2\log_{10}\left[\frac{\frac{q_c}{p_a}}{\left(\frac{\sigma_v}{p_a}\right)^{C_1}} - 1\right]$	(14)	C_1 = 0.5, C_2 = 68, p_a is the atmospheric pressure	Kulhawy and Mayne (1990) [45]
$D_r = AQ + B$ and $Q = \frac{q_c - \sigma_v}{\sigma_v}$	(15)	A = 0.2831, B = 32.964	Bolton and Gui (1993) [42]
$D_r = A_0 + B_0\ln\left[\frac{q_c}{(\sigma_v)^{0.5}}\right]$	(16)	A_0 = -1.292, B_0 = 0.268	Jamiolkowski et al. (2003) [3]

Comparison with the dataset from the developed neural networks finds that the linear regression of Eq. (15) works not so well in fitting all these data, especially for D_r lower than 50%. Eq. (16) underestimates the relative density from normalized cone resistance and is close to the lower bound of the dataset. The logarithmic form of Eqs. (13) and (14) conform better to the D_r - Q relationship. Nevertheless, Eq. (14) overestimates slightly the dataset for values of D_r lower than 50% while Eq. (15) overestimates slightly the dataset for values larger than 50%. For pure experimental data in Fig. 12, the R^2 of Eqs. (13) to (16) are 68.7%, 71.3%, 75.6%, and 65.2%, respectively. This conclusion illustrates that traditional empirical equation regarding a specific soil type has some limitations when expanded to other soil states. However, with the help of the ML model, we can generate hundreds of D_r - Q points for practical application. Consequently, a new correlation of D_r - Q by curve fitting to all the data in Fig. 12 is given below:

$$D_r = 29.832\ln Q - 72.397 \quad (17)$$

Eq. (17) has a high fitting degree of 92.7% to all D_r - Q points. For the pure experimental data in Fig. 12, Eq. (17) still has a relatively high R^2 value of 88.3%. It is believed that the improved Eq. (17) is more reliable since the database used for the fitting included not only the published experimental data but also a large amount of data from the developed ML models. In the future, if more datasets of calibration chamber tests can be provided to further train the ML models, more convincing ML models can be trained for enhancing practical application.

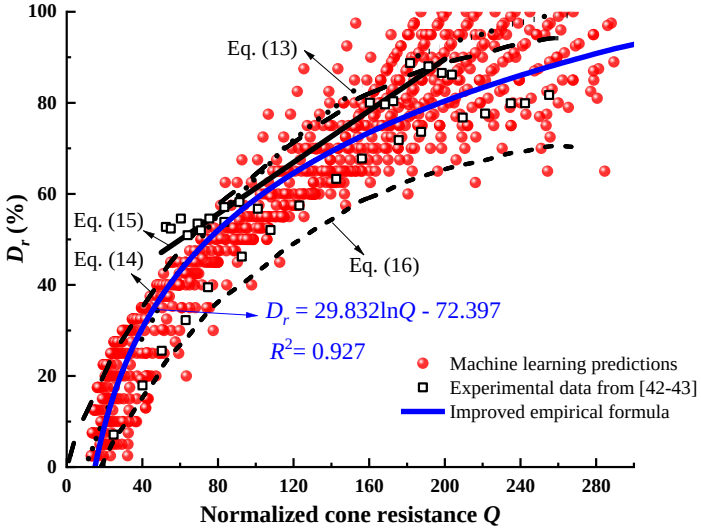


Fig. 12. Correlation between relative density and normalized cone resistance.

6. Conclusion

This study introduces a Bayesian-optimized machine learning approach using neural networks for predicting cone resistance profiles in calibration chamber tests performed on sand. We began by training the neural networks using 52 groups (5200 points) taken from the results in the literature, followed by validation of the results using a testing data set consisting of 12 groups (1200 points). Bayesian optimization was used to obtain the hyperparameters of both the MLP and the LSTM networks, and in particular the number of neurons per layer and the initial learning rate.

The prediction of the cone resistance q_c, had a coefficient of determination (R^2) for the

542 training set of 98.65% (MLP) and 98.51% (LSTM), and an MSE of 2.0006 (MLP) and
543 2.2200 (LSTM). In the training dataset, the R^2 and MSE values were 95.13% (MLP)
544 and 94.65% (LSTM), and 4.8272 (MLP) and 5.0109 (LSTM) respectively. Further
545 comparisons were carried out using the results of a Coupled Eulerian-Lagrangian model
546 for other soils not covered in the experimental results, which also gave good results.
547
548 Finally, the validated machine learning model was deployed to generate unseen
549 scenarios, correlating cone resistance (q_c) with the relative density of sand to prove the
550 validity of the derived results. The model is derived from 864 groups (cases) generated
551 using the ML model and is tested against literature models. The obtained R^2 value
552 versus experiments mentioned in the literature is 88.3%, which indicates its ability to
553 generalization.
554
555 In conclusion, we have demonstrated that neural networks can be applied to provide
556 general models that could serve the purpose of virtual calibration chambers. This
557 approach outperforms other previously published calibration chambers based on
558 numerical models as they provide pseudo-real-time models, and therefore, it is orders
559 of magnitude more computationally efficient. Besides, this work can be extended to
560 more CPT-based material parameter interpretations such as soil strength and stiffness
561 parameters using other machine learning-based approaches.

562 # Appendix A. CEL model set-up

563 The CEL model for a calibration chamber test is shown in Fig. A1. The CPT probe has
564 a standard diameter of D = 36 mm and an apex angle of 60° [4]. The cone penetrometer
565 was modeled by a Lagrangian body and discretized using C3D8R elements with a total
566 number of 202. Eulerian material was employed to model the soil with large
567 deformation. As the symmetry of the model test, only a 1/4 size model was established.
568 A soil domain of 25D in radial direction and 47D in depth direction was sufficient to

avoid boundary effects [4]. The central rectangular area of approximately 5.5D was densified to ensure the calculation accuracy. The mesh size in the dense field was set as $0.125D \times 0.27D$ in radial and vertical directions, respectively. The upper part of model represents the initially void region but could be filled by the movement of materials into elements during penetration, while the lower part is filled fully with materials. The soil domain was discredited by the EC3D8R elements with a total number of 12968 with reduced integration and hourglass control.

The bottom boundary of the soil domain was restrained in the vertical direction, and the lateral boundary was constrained in both horizontal directions (BC3). Two dynamic-explicit steps were employed to simulate the penetration process in sand, in which the first geostatic step is used to establish the initial stress field and balance the geostatic stress for sand. According to Table 6, the corresponding σ_v and K_0 were applied to generate the soil stress field. In the validation with an experiment in Fig. 10, the lateral earth pressure coefficient K_0 was set to be 0.45, which was consistent with the selected calibration chamber tests. The relative density was calculated by the initial void ratio [4]. In the second step, the modelling of CPT penetration under a constant vertical velocity of 0.2 m/s was performed. The contact between the cone/shaft and soil was modelled using a surface-to-surface discretization with a frictional coefficient of μ = 0.22.

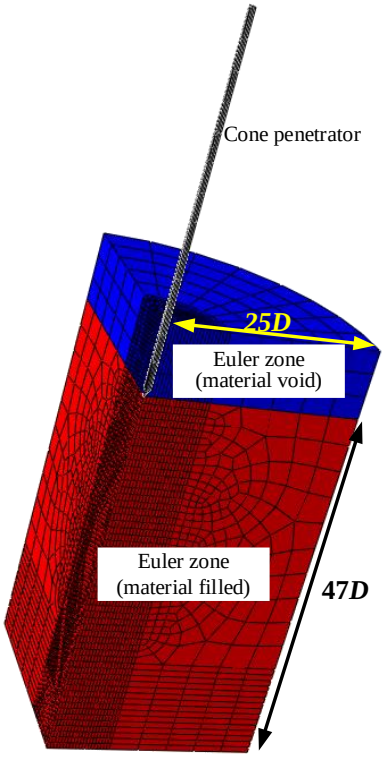


Fig. A1. CEL model for calibration chamber

Appendix B. Soil constitutive relationship

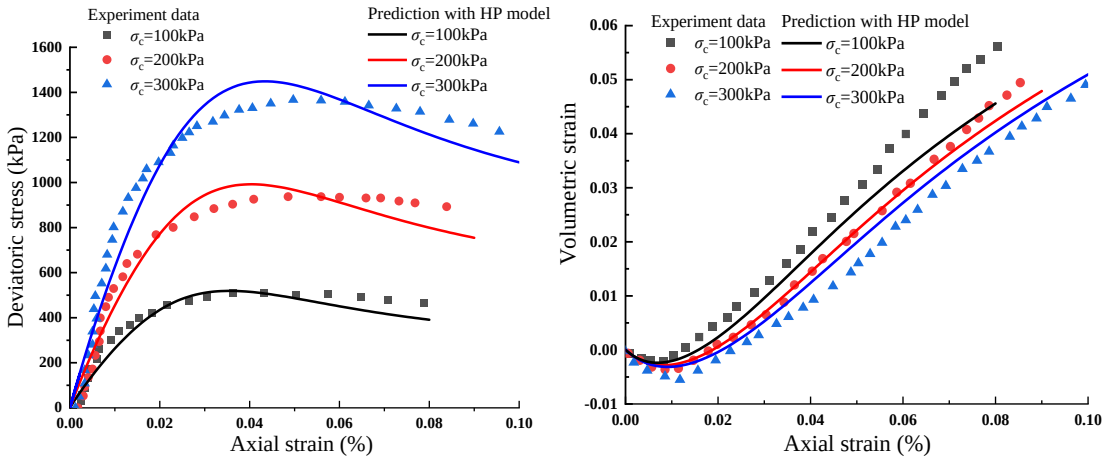
The hypoplastic (HP) constitutive model for sand proposed by Gudehus [46] and Wolffersdorff [47] was employed to model soil constitutive behavior. An extension of the concept of intergranular strain proposed by Niemunis and Herle [48] was established to account for the accumulation effects and hysteretic behavior in small strain stiffness cases. The Ticino sand was chosen to fill the chambers. The experimental drained triaxial shear data on Ticino sand conducted by Rorato et al. [49] is shown in Fig. B1 for dense sand with a constant initial void ratio e_0=0.612 under different confining pressure σ_c. A set of HP model parameters of the Ticino sand are summarized in Table B1. For calibration, Fig. B1 also gives the predicted constitutive relationships of Ticino sand modelled by HP model parameters in Table B1. From Fig. B1, although some small differences between the experimental and the numerical data are found, the HP model parameters listed in Table B1 are considered to be enough

reasonable, and hence can be used to run the following models.

To incorporate the effect of pore water pressure generated in the saturated sand, this paper employed the method by Qiu and Grabe [50]. The changes in pore water pressure p_w are determined by the state equation of pore pressures and the balance of mass:

$$\dot{p}_w = \frac{K_w}{n}\dot{\varepsilon}_v \quad \text{(B1)}$$

where K_w is the bulk modulus of water is derived 2.1 GPa; n is the porosity; ε_V is the volumetric strain. K_w=0 means dry sand while K_w>0 signifies saturated condition. The HP model and Eq. (B1) are employed to describe the constitutive relationship of sand and are incorporated using a VUMAT subroutine.



(a) Stress-strain relationship (b) Volumetric response

Fig. B1. Comparison between the experiment data and simulation with hypoplastic model

Table B1. Hypoplastic material parameters for Ticino sand

Parameter	Value
Critical state friction angle, φ_c	31°
Granular hardness, h_s (MPa)	2000
Material constant, n	0.29
Minimum void ratio at zero pressure, e_{d0}	0.57
Critical void ratio at zero pressure, e_{c0}	0.94
Maximum void ratio at zero pressure, e_{i0}	1.1
Material constant, α	0.16
Material constant, β	1.65
Maximum value of intergranular strain, R	0.00033
Material constant, m_R	5

Material constant, m_T	2
Evolution of intergranular strain, β_r	0.5
Parameter controlling stiffness degradation during monotonic deformation, χ	6

Acknowledgments

The authors would like to acknowledge the funding support by the China Scholarship Council (CSC).

Data availability

Data will be made available on request. The datasets and machine learning models are available in https://github.com/MingpengLiu/BO-MLP-and-BO-LSTM.

References

[1] Liyanapathirana, D.S. Arbitrary Lagrangian Eulerian based finite element analysis of cone penetration in soft clay. Computers and Geotechnics, 2009, 36(5): 851-860.

[2] Hamann, T., G. Qiu and J. Grabe. Application of a Coupled Eulerian–Lagrangian approach on pile installation problems under partially drained conditions. Computers and Geotechnics, 2015, 63: 279-290.

[3] Jamiolkowski, M., D. Lo Presti and M. Manassero. Evaluation of relative density and shear strength of sands from CPT and DMT, in *Soil behavior and soft ground construction*. 2003. 201-238.

[4] Chen, Z., M. Huang and Z. Shi. Application of a state-dependent sand model in simulating the cone penetration tests. Computers and Geotechnics, 2020, 127: 103780.

[5] POURNAGHIAZAR, M., A.R. RUSSELL and N. KHALILI. The cone penetration test in unsaturated sands. Géotechnique, 2013, 63(14): 1209-1220.

[6] Mo, P.-Q., A.M. Marshall and H.-S. Yu. Interpretation of cone penetration test data in layered soils using cavity expansion analysis. Journal of Geotechnical and Geoenvironmental Engineering, 2017, 143(1): 04016084.

640 [7] Cudmani, R. and V. Osinov. The cavity expansion problem for the interpretation of
641 cone penetration and pressuremeter tests. Canadian Geotechnical Journal, 2001, 38(3):
642 622-638.
643 [8] Susila, E. and R.D. Hryciw. Large displacement FEM modelling of the cone
644 penetration test (CPT) in normally consolidated sand. International Journal for
645 Numerical and Analytical methods in geomechanics, 2003, 27(7): 585-602.
646 [9] Wang, D., B. Bienen, M. Nazem, et al. Large deformation finite element analyses
647 in geotechnical engineering. Computers and Geotechnics, 2015, 65: 104-114.
648 [10] Zhang, N., M. Arroyo, M.O. Ciantia, et al. Standard penetration testing in a virtual
649 calibration chamber. Computers and Geotechnics, 2019, 111: 277-289.
650 [11] Fan, S., B. Bienen and M. Randolph. Stability and efficiency studies in the
651 numerical simulation of cone penetration in sand. Géotechnique Letters, 2018, 8(1): 13-
652 18.
653 [12] Yu, H., H. Zhou, B. Sheil, et al. Finite element modelling of helical pile installation
654 and its influence on uplift capacity in strain softening clay. Canadian Geotechnical
655 Journal, 2022, 0(0): null.
656 [13] Zhang, P. A novel feature selection method based on global sensitivity analysis
657 with application in machine learning-based prediction model. Applied Soft Computing,
658 2019, 85: 105859.
659 [14] Alkroosh, I.S., M. Bahadori, H. Nikraz, et al. Regressive approach for predicting
660 bearing capacity of bored piles from cone penetration test data. Journal of Rock
661 Mechanics and Geotechnical Engineering, 2015, 7(5): 584-592.
662 [15] Wang, H., X. Wang, J.F. Wellmann, et al. A Bayesian unsupervised learning
663 approach for identifying soil stratification using cone penetration data. Canadian
664 Geotechnical Journal, 2018, 56(8): 1184-1205.
665 [16] Erharter, G.H., S. Oberhollenzer, A. Fankhauser, et al. Learning decision
666 boundaries for cone penetration test classification. Computer-Aided Civil and
667 Infrastructure Engineering, 2021, 36(4): 489-503.

668 [17] Rauter, S. and F. Tschuchnigg. Identification of soil strata from in-situ test data
669 using machine learning. in International Conference of the International Association for
670 Computer Methods and Advances in Geomechanics. 2022: Springer.
671 [18] Pham, B.T., T.-A. Hoang, D.-M. Nguyen, et al. Prediction of shear strength of soft
672 soil using machine learning methods. Catena, 2018, 166: 181-191.
673 [19] Ahmad, M., X.-W. Tang, J.-N. Qiu, et al. Application of machine learning
674 algorithms for the evaluation of seismic soil liquefaction potential. Frontiers of
675 Structural and Civil Engineering, 2021, 15: 490-505.
676 [20] Samui, P. Liquefaction prediction using support vector machine model based on
677 cone penetration data. Frontiers of Structural and Civil Engineering, 2013, 7: 72-82.
678 [21] Guan, Q.Z., Z.X. Yang, N. Guo, et al. Finite element geotechnical analysis
679 incorporating deep learning-based soil model. Computers and Geotechnics, 2023, 154:
680 105120.
681 [22] Zhang, P., Z.Y. Yin, Y.F. Jin, et al. An AI-based model for describing cyclic
682 characteristics of granular materials. International Journal for Numerical and Analytical
683 Methods in Geomechanics, 2020, 44(9): 1315-1335.
684 [23] Guan, Q.Z. and Z.X. Yang. Hybrid deep learning model for prediction of
685 monotonic and cyclic responses of sand. Acta Geotechnica, 2022.
686 [24] Wang, K. and W. Sun. A multiscale multi-permeability poroplasticity model linked
687 by recursive homogenizations and deep learning. Computer Methods in Applied
688 Mechanics and Engineering, 2018, 334: 337-380.
689 [25] Habibagahi, G. and A. Bamdad. A neural network framework for mechanical
690 behavior of unsaturated soils. Canadian Geotechnical Journal, 2003, 40(3): 684-693.
691 [26] Kohestani, V.R. and M. Hassanlourad. Modeling the Mechanical Behavior of
692 Carbonate Sands Using Artificial Neural Networks and Support Vector Machines.
693 International Journal of Geomechanics, 2016, 16(1): 04015038.
694 [27] Alizadeh, B., A.G. Bafti, H. Kamangir, et al. A novel attention-based LSTM cell
695 post-processor coupled with bayesian optimization for streamflow prediction. Journal

696 of Hydrology, 2021, 601: 126526.
697 [28] Tao, Y., H. Sun and Y. Cai. Predictions of deep excavation responses considering
698 model uncertainty: integrating BiLSTM neural networks with Bayesian updating.
699 International Journal of Geomechanics, 2022, 22(1): 04021250.
700 [29] Zhang, P., Z.-Y. Yin and Y.-F. Jin. Bayesian neural network-based uncertainty
701 modelling: application to soil compressibility and undrained shear strength prediction.
702 Canadian Geotechnical Journal, 2021, 59(4): 546-557.
703 [30] Zhang, P., Z.-Y. Yin and Y.-F. Jin. State-of-the-Art Review of Machine Learning
704 Applications in Constitutive Modeling of Soils. Archives of Computational Methods in
705 Engineering, 2021, 28(5): 3661-3686.
706 [31] Dai, H. and C. MacBeth. Effects of Learning Parameters on Learning Procedure
707 and Performance of a BPNN. Neural Networks, 1997, 10(8): 1505-1521.
708 [32] Yang, Z., R. Jardine, B. Zhu, et al. Sand grain crushing and interface shearing
709 during displacement pile installation in sand. Géotechnique, 2010, 60(6): 469-482.
710 [33] Huang, A.-B. and H.-H. Hsu. Cone penetration tests under simulated field
711 conditions. Geotechnique, 2005, 55(5): 345-354.
712 [34] Kluger, M.O., S. Kreiter, F.T. Stähler, et al. Cone penetration tests in dry and
713 saturated Ticino sand. Bulletin of Engineering Geology and the Environment, 2021,
714 80(5): 4079-4088.
715 [35] Bałachowski, L. Penetration resistance of Lubiatowo sand in calibration chamber
716 tests. Archives of Hydro-Engineering and Environmental Mechanics, 2006, 53(4): 311-
717 329.
718 [36] Schnaid, F. A study of the cone-pressuremeter test in sand. 1990, University of
719 Oxford.
720 [37] ARROYO, M., J. BUTLANSKA, A. GENS, et al. Cone penetration tests in a
721 virtual calibration chamber. Géotechnique, 2011, 61(6): 525-531.
722 [38] Zhang, P., Z.-Y. Yin, Y. Zheng, et al. A LSTM surrogate modelling approach for
723 caisson foundations. Ocean Engineering, 2020, 204: 107263.

724 [39] He, S. and J. Li. Modeling nonlinear elastic behavior of reinforced soil using
725 artificial neural networks. Applied Soft Computing, 2009, 9(3): 954-961.
726 [40] Ruder, S. An overview of gradient descent optimization algorithms. arXiv preprint
727 arXiv:1609.04747, 2016.
728 [41] Huang, Y., Y. Wang, Z. Xu, et al. Prediction and variable importance analysis for
729 small-strain stiffness of soil based on ensemble learning with Bayesian optimization.
730 Computers and Geotechnics, 2023, 162: 105688.
731 [42] Bolton, M. and M. Gui. The study of relative density and boundary effects for cone
732 penetration tests in centrifuge. 1993: University of Cambridge, Department of
733 Engineering.
734 [43] Jamiolkowski, M. New developments in field and laboratory testing or soils. in
735 Proc. 11th. Int. Conf. on SMFE., San Francisco, CA, 1985. 1985.
736 [44] Lunne, T. and H.P. Christoffersen. Interpretation of cone penetrometer data for
737 offshore sands. in Offshore Technology Conference. 1983: OTC.
738 [45] Kulhawy, F.H. and P.W. Mayne. Manual on estimating soil properties for
739 foundation design. 1990, Electric Power Research Inst., Palo Alto, CA (USA); Cornell
740 Univ., Ithaca ….
741 [46] Gudehus, G. A comprehensive constitutive equation for granular materials. Soils
742 and foundations, 1996, 36(1): 1-12.
743 [47] Von Wolffersdorff, P.A. A hypoplastic relation for granular materials with a
744 predefined limit state surface. Mechanics of Cohesive-frictional Materials: An
745 International Journal on Experiments, Modelling and Computation of Materials and
746 Structures, 1996, 1(3): 251-271.
747 [48] Niemunis, A. and I. Herle. Hypoplastic model for cohesionless soils with elastic
748 strain range. Mechanics of Cohesive-frictional Materials: An International Journal on
749 Experiments, Modelling and Computation of Materials and Structures, 1997, 2(4): 279-
750 299.
751 [49] Rorato, R., M. Arroyo, A. Gens, et al. Image-based calibration of rolling resistance

752 in discrete element models of sand. Computers and Geotechnics, 2021, 131: 103929.
753 [50] Qiu, G. and J. Grabe. Explicit modeling of cone and strip footing penetration under
754 drained and undrained conditions using a visco-hypoplastic model. geotechnik, 2011,
755 34(3): 205-217.

756