HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Soils and Foundations 63 (2023) 101321

www.elsevier.com/locate/sandf

Technical Paper

A pre-trained deep-learning surrogate model for slope stability analysis with spatial variability

Haoding Xu [a], Xuzhen He [a,*], Biswajeet Pradhan [b], Daichao Sheng [a]

[a] School of Civil and Environmental Engineering, University of Technology Sydney, NSW 2007, Australia
[b] Centre for Advanced Modelling and Geospatial Information Systems (CAMGIS), School of Civil and Environmental Engineering, University of Technology Sydney, NSW 2007, Australia

Received 17 October 2022; received in revised form 1 April 2023; accepted 11 April 2023
Available online 22 May 2023

Abstract

This paper presents a pre-trained deep-learning surrogate model for the slope stability problem, which can be used to accelerate the stochastic analysis of slope stability with spatial variability. One major innovation is that the model is trained with a big dataset (>12000 data) covering common soil properties, spatial variabilities, and slope shapes such that the trained model is ready to make predictions without additional training or numerical simulations required. Other two minor contributions are: (1) special treatments for the irregular and varying boundaries of slopes and (2) novel techniques that allow the use of non-uniform mesh in data acquisitions. The proposed model is accurate with a mean-absolute-percentage-error of about 6% for the testing dataset. Seven cases of unseen data are also used to verify the model performance, including cases of different soil parameters, slope angles, and even different slope surfaces (e.g., concave and convex slopes, which are not used in training). The results show that the predict slope factor of safety is high consistent with the values from finite element simulations, and so is the obtained probability density functions. But the surrogate model takes much less computational effort (several minutes compared with hours of computing) – proving the effectiveness of our model for efficient stochastic analyses.

Keywords: Deep learning; Deep neural network; Factor of safety; Slope stability

1. Introduction

In geotechnical engineering, slope stability is given as a factor of safety (FS), which is calculated with limit equilibrium methods such as the method of slides, or numerical methods such as limit analysis (Chen, 1976), finite elements limit analysis (Sloan, 1988) and displacement-based numerical methods with the strength reduction method (Dawson et al., 1999; Griffiths & Lane, 1999).

The required model parameters in these numerical models cannot be accurately determined, which is due to the poorly known model parameters and complex system of equations that cannot be represented exactly. Therefore, numerical simulation is an approximation of the actual situation (Atamturktur et al., 2011; Shields, 2013). Additionally, soil properties vary spatially dramatically due to sediment history and weathering attack. So stochastic analysis is often conducted to evaluate the probability density function (PDF) of the FS (Pang et al., 2018; Rice & Polanco, 2012). The spatial variability is often modelled by the random field theory (Griffiths & Fenton, 2001).

Brute-force stochastic analysis typically relies on Monte Carlo sampling (Kroese et al., 2014). As shown in Fig. 1

Peer review under responsibility of The Japanese Geotechnical Society.
* Corresponding author.
E-mail addresses: haoding.xu@student.uts.edu.au (H. Xu), xuzhen.he@uts.edu.au (X. He), biswajeet.pradhan@uts.edu.au (B. Pradhan), daichao.sheng@uts.edu.au (D. Sheng).

https://doi.org/10.1016/j.sandf.2023.101321
0038-0806/

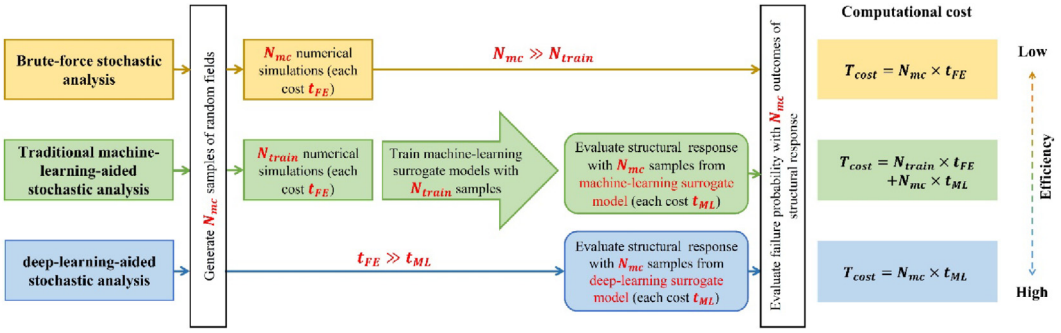


Fig. 1. Flowchart of different stochastic analyse methods.

(highlighted in yellow), this method involves generating N_{mc} samples, evaluate their FS with numerical simulations, and finally evaluating the failure probability. Each simulation necessitates a calculation time of t_{FE}. However, when the sample size is large (i.e., for small probability events or when the PDF is needed), the brute-force Monte Carlo approach – the FS of each Monte Carlo sample is evaluated by numerical models – is time-consuming and computationally intensive (Chen & Li, 2017; Zhang et al., 2017). A variety of methods have been proposed to reduce the computational cost, such as the subset simulation method (Beck & Au, 2002) or using surrogate models to replace the numerical simulations (He et al., 2020).

In recent years, training machine-learning models as surrogate models (green in Fig. 1) has gained immense popularity (Phoon & Zhang, 2022; Zhang et al., 2017). Kang et al., (2016) trained support vector machines as surrogate models in slope reliability analysis. He et al., (2020) used artificial neural networks as surrogate models for slope stability analysis with spatial variability. Wang et al., (2020) used extreme gradient boosting method to analyse the earth dam slope stability problem. In these studies, a size of N_{train} simulations are still required to for each specific engineering case (Fig. 1). These N_{train}data are used to train the machine-learning surrogate model for the specific engineering case that can subsequently evaluate the structural response with time of t_{ML}. Due to the fact that N_{train} and t_{ML} are considerably less than N_{mc} and t_{FE}, this method (machine-learning-aided stochastic analysis) is far more efficient than brute-force stochastic analysis methods. He et al., (2022) developed a new framework in which deep-learning models are trained as surrogate models with a big dataset that covers all possible material properties and boundary conditions for a particular kind of problem (the bearing capacity of shallow strip footings in their study). For any new engineering case, the deep-learning model is ready to make predictions and is also very accurate. Therefore, the small number of numerical simulations and training processes are not needed anymore, leading to a dramatic increase in the computational efficiency of the deep-learning-aided stochastic analysis compared to the traditional machine-learning-aided stochastic analysis and brute-force stochastic analysis methods (as illustrated in Fig. 1). Additionally, this framework separates the training of models and the use of models such that the training of models is left to academics so that practitioners can directly use the models. This framework is made possible mainly due to the rapid development of deep-learning techniques, particularly the convolutional neural network (CNN) and its variants. CNNs have successfully been used in image/video classification (SravyaPranati et al., 2021), image/video recognition(Szegedy et al., 2015) and even the program 'AlphaGo' in mastering the game of Go (Silver et al., 2016). CNNs are also widely applied in geotechnical engineering, such as the prediction of wall deflection (Wu et al., 2022), multi-layered soil slope stability (Wang & Goh, 2022), and bearing capacity failure (He et al., 2022; Wang et al., 2022). More recently, He et al., (2022) showed that the locally connected neural networks (a variant of CNN) perform better than CNNs alone for spatially variable problems that are aligned.

In a very recent work, He et al., (2022) demonstrated the success of their framework with the bearing capacity problem, which "luckily" has regular boundaries (rectangle computation domain) and can be easily mapped into matrices. After normalisation and discretisation, the inputs of the problem (and also the inputs of the target function to be approximated by machine learning) can be arranged exactly as matrices, regardless of material properties and boundary conditions. This framework cannot be readily applied to other problems like slope stability problems because of irregular boundaries – spatially variable material parameters will not make matrices anymore. Most importantly, when boundary condition changes (e.g., when the slope angle changes), the number (or size) of inputs also changes. Therefore, to use this framework, special treatment is required.

Another limitation of He et al., (2022)'s work is that to make input matrices, the soil domain is discretised as uniform quadrilaterals. When creating a training dataset with numerical simulations, a uniform mesh must be used. However, to obtain accurate results in numerical simulations, nonuniform meshes that are coarser in some regions and finer in others are usually used.

Two innovations are presented in this paper: (1) special treatments for the irregular and varying boundaries of

slopes and (2) novel techniques that allow the use of non-uniform mesh in data acquisition. We take the slope stability problem as an example and train a deep-learning model that can predict the slope safety factor of any shape. Additionally, we show that the model is valid for slopes represented by very complex curves. In the same as the previous study (He et al., 2022), the model is valid for any soil properties encountered in practice.

In Section 2, we introduce the slope stability problems. The output is the FS as a scalar and the inputs are material parameters represented as matrices with boundary conditions embedded in these matrices. How the FS is calculated from finite element (FE) simulations is explained in Section 3. In Section 4, we explain the whole process of building deep-learning models, from a simple condition to the final full problem. In Section 5, we verify the accuracy of the trained deep-learning model and prove that it can be used as an ideal surrogate model in slope stability problems.

2. Slope stability: Inputs and outputs

Natural hills are often cut into straight slopes (angled at β in Fig. 2), which are connected with the flat open lands at the slope toe (origin O in Fig. 2), and are connected with slope tops (angled at α) at the slope shoulder. We define the vertical distance between slope toe and shoulder as the slope height H.

The shear resistance between soil elements can sustain a certain gravity load such that slopes remain stable. When the shear resistance is not enough to sustain the gravity load, landslides happen. In geomechanics, slope stability is defined by a factor of safety (FS), which is the ratio between the available shear strength and the acting shear stress due to gravity along a potential (or actual) sliding surface. The FS results from two competing effects – the shear resistance and gravity load. If the soil elements are modelled with the Mohr–Coulomb model with c denoting the cohesion and ϕ denoting the friction angle, the FS can be conceptually expressed as

$$FS = f[c(\boldsymbol{x}), \phi(\boldsymbol{x}), \gamma(\boldsymbol{x})] \tag{1}$$

where γ is the unit weight and $\boldsymbol{x} = (x, y)$ is the position vector. The dependence on $\boldsymbol{x}$ indicates the spatial variability of soil properties and the soil parameters are all fields. The heatmap in Fig. 2 illustrates a typical spatially variable field. The soil parameters are spatially variable – the cohesion, friction angle and the unit weight are all fields that are functions of the coordinates (e.g., $c(\boldsymbol{x}), \phi(\boldsymbol{x}), \gamma(\boldsymbol{x})$). However, these soil parameters fields are only defined in the soil domain. If we assume that at any position not in the soil domain these soil parameters are still defined but with value zeros (e.g., $c(\boldsymbol{x}) = \phi(\boldsymbol{x}) = \gamma(\boldsymbol{x}) = 0$), then these fields are defined in the unbounded domain. As a consequence, soil slopes of different slope angles will have different soil parameter fields, and Eq.1 takes only three functions (e.g., $c(\boldsymbol{x}), \phi(\boldsymbol{x}), \gamma(\boldsymbol{x})$) as arguments and the dependence of the FS on slope angles is embedded in the dependence of these fields.

The FS is influenced by the relative magnitude of shear resistance and gravity load. For homogenous slopes, this relative magnitude is characterised by two dimensionless numbers: $\frac{\gamma H}{c}$ and ϕ (Chen, 1976). For spatially variable fields, we define some dimensionless fields, i.e., the dimensionless unit weight field $\gamma'(\boldsymbol{x}') = \frac{\gamma(\boldsymbol{x})H}{c_r}$, friction angle field $\phi(\boldsymbol{x}')$ and dimensionless cohesion field $c'(\boldsymbol{x}') = \frac{c(\boldsymbol{x})}{c_r}$), so a dimensionless equation for the FS is obtained:

$$FS = f[c'(\boldsymbol{x}'), \phi(\boldsymbol{x}'), \gamma'(\boldsymbol{x}')] \tag{2}$$

Here, c_r is a reference strength and is chosen as the mean of the cohesion field. $\boldsymbol{x}' = \frac{\boldsymbol{x}}{H}$ is the dimensionless position vector.

The FS ranges from zero (no shear resistance) to infinity (gravity load is negligible). In risk analysis, we are only interested in slopes with FS close to 1, but not interested in very safe slopes, say FS > 5. Therefore, in this study, we calculate a normalised factor of safety as $NFS = 1 - e^{-FS}$ (Fig. 3). When the slope is very safe (FS > 5), no matter how large the FS is, the NSF is the same and is close to 1. So, when conducting numerical simulations, we just stop the simulation at FS = 5 and do not need to calculate the exact FS when FS > 5. Additionally, the NFS is within a defined range (from 0 to 1) and using it as a target in machine learning helps to get more accurate models.

The target function to be approximated (Eq. (2) takes unbounded continuous fields as inputs (with a size of infinity). We will use some assumptions and conduct discretisation to transform them into inputs with finite sizes. The FS is a ratio evaluated along a potential sliding surface, which is often in a limited region close to the slope. Only when the

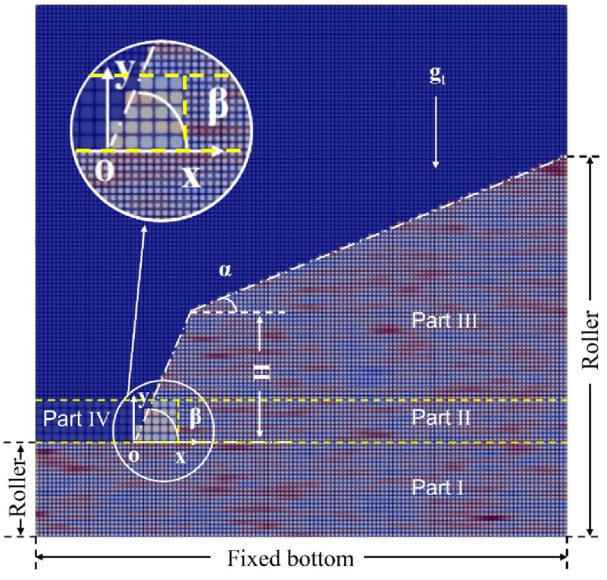


Fig. 2. Shape of slopes and spatially variable fields.

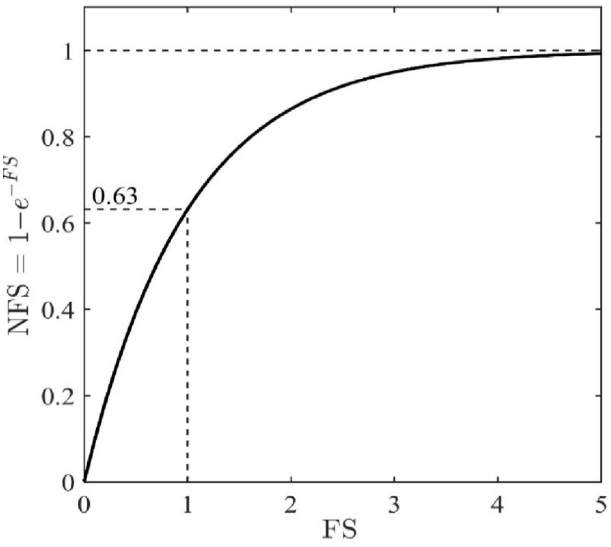


Fig. 3. Normalised factor of safety (NFS) and the factor of safety (FS).

slope angle is small and the friction angle is very small, the sliding surface may potentially extend beyond the slope toe to the left and deep below the toe (Fig. 7d). Therefore, we limit the spatial dependence to a region of $-0.7 \le x' \le 3.3$ and $-0.7 \le y' \le 3.3$. It is shown in the next section that finite element (FE) simulations with this size of domain give accurate results against analytical solutions, even when the friction angle is small. The whole domain is discretised into a mesh with squares of size $\Delta l' = 0.0312$, leading to a mesh with 128 × 128 elements. The straight slope surfaces (angled at α and β) are represented by stair-step-shaped lines (Fig. 2). After adopting finite-region dependence and finite resolution, a dimensionless equation for the NFS can be expressed below.

$$NFS = f[\mathbf{C}', \Phi, \Gamma'] \tag{3}$$

where $\mathbf{C}'$, Φ and Γ' are discretised dimensionless fields of soil properties and are represented by matrices of size 128 × 128.

To avoid building an over-complex model, we impose some limits on these dimensionless fields by including values only possible in practice. (Phoon & Kulhawy, 1999) reported typical soil parameters in Table 1. Because the reference strength is chosen as the mean of cohesion, the mean of dimensionless cohesion fields is always 1 (Table 2). If we limit the height of slopes up to 50 m, the dimensionless unit weight has a mean γ' of 0–100 (Table 2). The upper limit for the coefficient of variation (COV) of the dimensionless parameters is chosen the same as the typical soil parameters (Table 1). But the lower limit is extended further to 0 to include the homogenous cases (COV = 0) and near-homogenous cases (COV is very small).

Table 2
Dimensionless material parameters and slope shape parameters.

Parameters	Mean μ	COV v	l'_y	l'_x
c'	1	0–0.55	0.0312-∞	0.44-∞
ϕ	0-40°	0–0.15		
γ'	0–100	0–0.1		
α	0-40°			
β	45-90°			

The lower limit for the dimensionless vertical scale of fluctuation should be $l'_y = \frac{l_y}{H} \ge \frac{0.2\ m}{50\ m} = 0.004$. However, to accommodate such a small spatial variability, we need a similar resolution ($\Delta l' = \frac{\Delta l}{H} = 0.004$) in FE models, which will result in a discretised domain of size 1000 × 1000. It will be a great burden to obtain input–output pairs with such a big FE model. Therefore, in this study, we use a slightly larger resolution $\Delta l' = 0.0312$, and impose this as a lower limit for the dimensionless vertical scale of fluctuation as in Table 2. Slopes can be made into different shapes. Typical slope angle is β = 45-90° and the slope top is α = 0-40°. Sample input–output pairs with slope angles in these ranges (Table 2) will be prepared.

Next, the spatially variable inputs are prepared with their dimensionless parameters defined in Table 2, and the corresponding outputs (FSs) are calculated with FE models. These input–output pairs are then fed into deep-learning algorithms to train surrogate models.

3. FE models and calculation of safety factors

Displacement-based FE models (within Abaqus) are used to estimate the output (NFS) for each input ($\mathbf{C}'$, Φ and Γ'). Instead of employing the strength reduction method, which requires a rough guess of the FS beforehand, we use the method of increasing gravity (Kaur & Sharma, 2016; Pourkhosravani & Kalantari, 2011). The FS is the ratio between the assumed work of external loads (from gravity) and internal energy dissipation (from soil strength) (Chen 1976). Thus, the FS could be intuitively obtained by increasing the gravity, which should give similar results as the strength reduction method. Also, we conduct some validations, which prove that the FSs from our simulations agree with analytical solutions within a wide range of parameters.

Table 1
Typical soil parameters.

Parameters	Mean μ	Coefficient of variation v	Vertical scale of fluctuation l_y	Horizontal scale of fluctuation l_x
c or s_u	10–700 kPa	0.1–0.55	0.2–6.2 m	23–66 m
ϕ	0-40°	0.05–0.15		
γ	13–20 kN/m^3	0–0.1		

The FE mesh is the same as in Fig. 2 with the straight slope lines represented by stair-step-shaped lines. The bottom boundary is fixed. The left and right boundaries can move vertically. The elements not occupied by soils do not enter the calculation.

When close to failure, the high gradient of deformation near the slope toe often causes failure of convergence. So, we choose a larger mesh size near the slope toe ($\Delta l' = 2 \times 0.0312$ as in the inset of Fig. 2). In the implementation, the whole domain is made of four parts, three parts (I, II, and III) have smaller resolutions and Part IV ($-0.7 \le x' \le 0.4$ and $0 \le y' \le 0.3$) has a larger resolution. These four parts are connected into a whole assembly with tie constraints. Another technique used to mitigate the numerical instability (particularly profound when the friction angle is large) is numerical damping, which is introduced such that the viscous dissipation is a fraction (0.0002) of the strain energy.

Gravity is initially set to zero, and the stress within slopes is zero. In the loading step, the gravitational acceleration increases in linear increments. With increasing gravity, the stress field will redistribute and is always in equilibrium with gravity. When the stress field cannot balance the increasing gravity, failure occurs. The ratio between the gravitational acceleration at the time of failure (g_t) and the real gravitational acceleration ($g = 9.81$ m/s^2) is the FS from our simulations. Due to the damping applied, convergence is still possible after a failure happens. Therefore, failure is determined in terms of a sudden change of displacement. In Fig. 4, solid lines show the displacement of the slope head u_{head} for three homogenous slopes, and this displacement is a function of the gravity, i.e., $u_{head}(g_t)$. Following He et al., (2020), the failure can be defined at the maximum second-order derivative of $\log(u_{head})$ with respect to g_t (dots in Fig. 4). Dashed lines are the estimated failure positions.

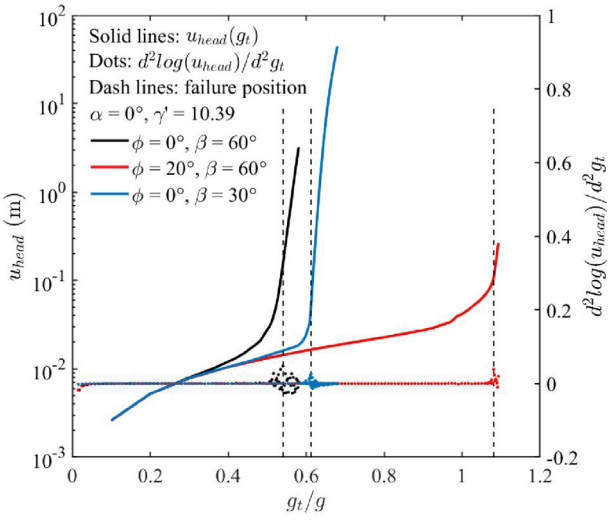


Fig. 4. Displacement/gravity curve for three homogenous slopes.

(Chen, 1976) gave a stability factor $N_s(\phi, \alpha, \beta)$ for homogeneous slopes (no spatial variability), and a critical height is calculated as $H_c = \frac{c}{\gamma} N_s$. When the slope height H is smaller than this critical height, slopes are stable. Slopes become unstable if the height is higher than this critical height. So the FS from his limit analysis is expressed in Eq. (4).

$$FS = \frac{H_c}{H} = \frac{c}{\gamma H} N_s = \frac{N_s(\phi, \alpha, \beta)}{\gamma'} \tag{4}$$

From the analytical solution, when $\phi = 20°$, $\alpha = 0°$, and $\beta = 60°$, the stability factor N_s is 10.39. So slopes with $\gamma' = 10.39$ will have FS = 1 and NFS = 0.632. We conducted 50 simulations with c and γ generated with the quasi-random sequence (He et al., 2021) according to the limits in Table 1. The generated $\boldsymbol{c}$ and $\boldsymbol{\gamma}$ sequences are shown in Fig. 5. The slope height H is chosen such that $\gamma' = 10.39$. FSs and NFSs are presented in Fig. 6a and Fig. 6b, respectively. Our method can accurately predict the FS compared with the analytical solution (only 5.11% error for the FS and 2.91% for the NFS). These results also show that our dimensional analysis is correct, the same dimensionless fields will lead to the same FSFig. 5.

When the dimensionless unit weight varies, the FS varies according to Eq.5. We conducted another 50 simulations with $\phi = 20°$, $\alpha = 0°$, and $\beta = 60°$ while γ' varies from 0 to 100 (the selected range in Table 2). The NFSs from FE simulations (Fig. 6c) are accurate compared with analytical results – only 0.58% mean absolute percentage error (MAPE).

Fig. 6d shows 50 results with $\gamma' = 10.39$, $\alpha = 0°$, and $\beta = 60°$ while ϕ varies from 0 to 40°. The MAPE for these NFSs is 2.7%. When the friction angle ϕ is small (less than30°), the FSs from FE simulations agree well with analytical solutions. However, when the friction angle ϕ is

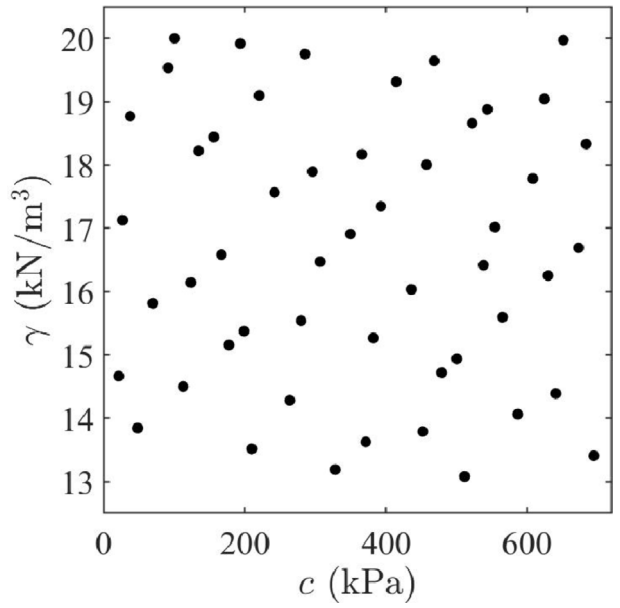
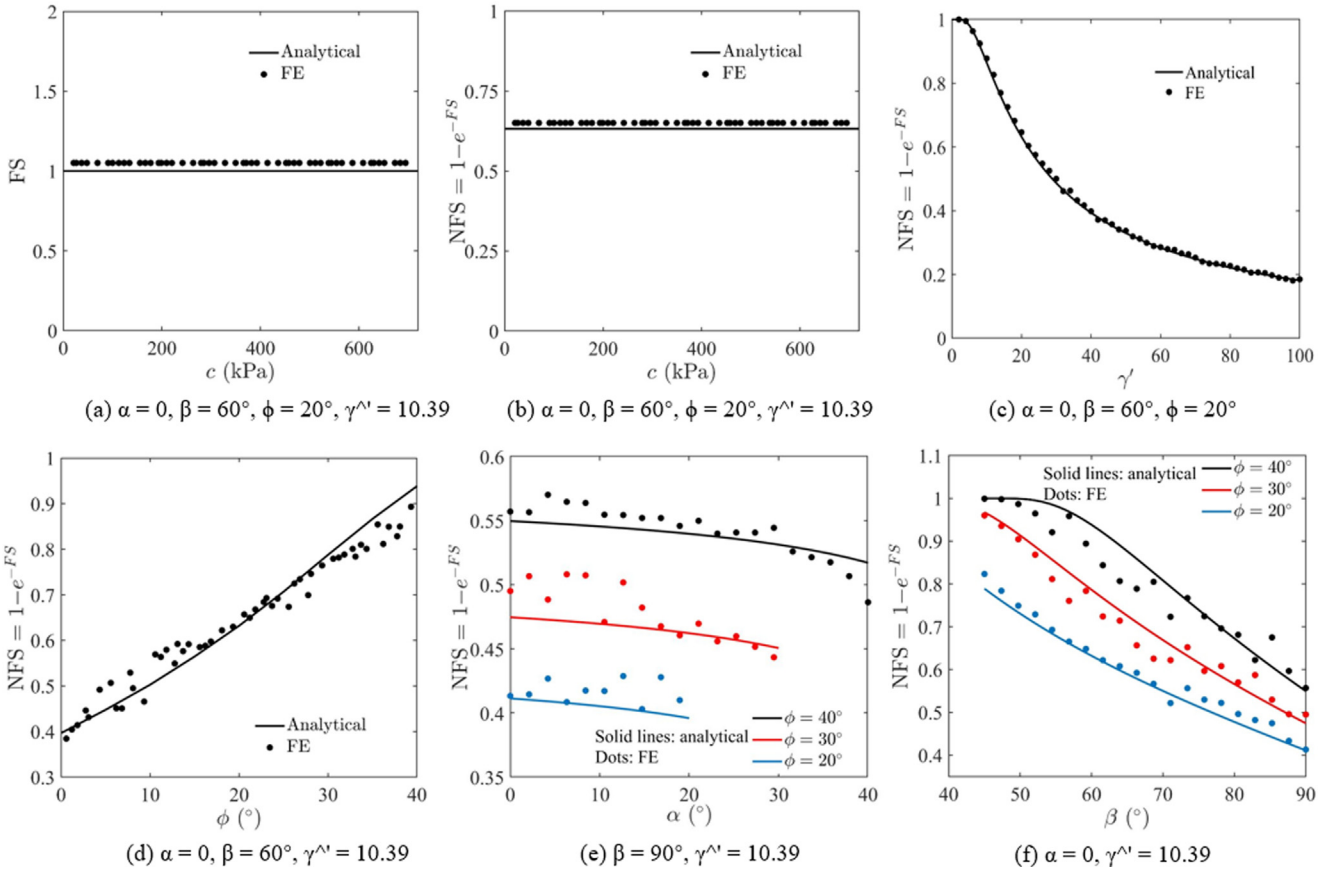


Fig. 5. Generate 50 combinations of $\boldsymbol{c}$ and $\boldsymbol{\gamma}$ with the quasi-random sequence.



(a) α = 0, β = 60°, ϕ = 20°, γ^′ = 10.39 (b) α = 0, β = 60°, ϕ = 20°, γ^′ = 10.39 (c) α = 0, β = 60°, ϕ = 20°

(d) α = 0, β = 60°, γ^′ = 10.39 (e) β = 90°, γ^′ = 10.39 (f) α = 0, γ^′ = 10.39

Fig. 6. Safety factors from FE simulations against analytical solutions.

large (from 30° to 40°), the FE model predicts lower NFSs than analytical solutions (i.e., conservative NFSs). The FSs are even more conservative due to the exponential operation in the definition of NFS. However, this conservative FS is preferred in practice. Additionally, these FS are>1.5 (NFS > 0.8). As we mentioned, engineers usually are not very interested in very safe slopes. So the misprediction of these large friction angles is not an issue for practical applications.

Fig. 6e and f show results when $\gamma' = 10.39$, ϕ is 20°, 30° or 40°, and α, β vary from 0 to 40° and 45-90°, respectively. All results in Fig. 6 show that the FE model can give accurate NFSs. There are in total 255 non-duplicate results ($N_{data1} = 255$) as shown in Fig. 6. Compared with analytical solutions, the MAPE of these NFSs is 3.6%, and these data form part of the dataset to train the deep-learning models.

Our numerical scheme can not only accurately calculate the FS, but also predict the sliding surface. Fig. 7 shows the displacement field of four slopes at failure along with analytical sliding surfaces. θ_0, θ_h, and length of OB and OC are calculated from the analytical method, and the log-spiral sliding surface (black lines) is determined from these parameters. Fig. 7a and b show two slopes with horizontal slope top while with different friction angles. Fig. 7c is a vertical slope with a sloped top $\alpha = 10°$. Fig. 7d is a very mild slope ($\beta = 30°$) with a low friction angle ($\phi = 0°$). From Chen, (1976)'s analysis, the sliding surface is generally a log-spiral passing through the slope toe (Fig. 7a-c), while it passes below the slope toe only when the slope is mild and the friction angle is small, which is captured by the FE model as in Fig. 7d. Additionally, the sliding surface is generally very close to the slope (Fig. 7a-c), and its extent is relatively large only when the friction angle is small. Fig. 7d shows that the domain size we chose ($-0.7 \le x' \le 3.3$ and $-0.7 \le y' \le 3.3$) is large enough to accommodate the largest possible sliding surface.

4. Deep-learning models for spatially variable slopes

The full slope stability problem (Eq. (3) is complex, so we first studied reduced problems with fewer inputs, and progressively investigated more complex conditions until the full problem was considered. These problems are: (1) undrained soil with a fixed slope shape, i.e., the friction angle is a constant of zero, and the inputs are the dimensionless cohesion field $\mathbf{C}'$ and the dimensionless unit weight field Γ'. The friction angle Φ is irrelevant. Additionally, since the slope shape is fixed, some values of matrices ($\mathbf{C}'$ and Γ') are fixed to be zero. (2) Mohr-Coulomb soil with fixed slope shape. This problem is slightly more complex than the first problem. Although some values of matrices are still fixed to be zero, the inputs have three fields ($\mathbf{C}'$,

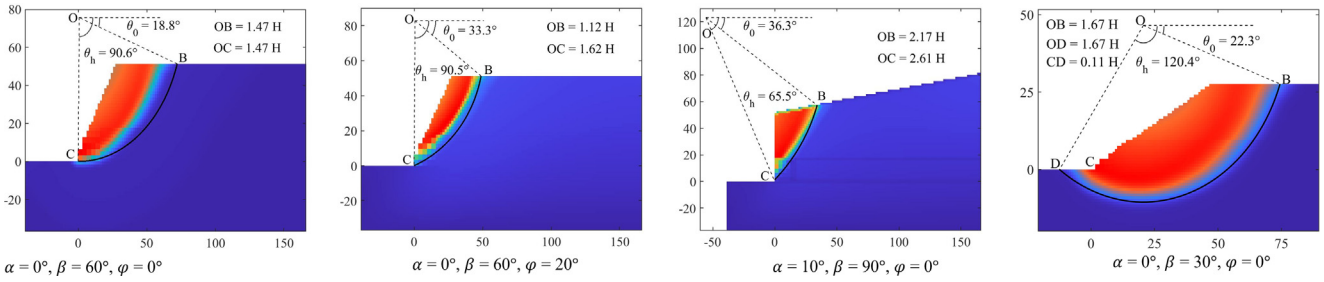


Fig. 7. Sliding surfaces from FE simulations against analytical solutions.

Φ and Γ'). (3) The full problem Eq. (3), where the inputs have three fields, and positions of zero-value entries of matrices depend on the slope shapes.

For each problem, we followed the following four steps to build machine-learning models: (1) generating representative inputs, (2) calculating NFSs, (3) choosing appropriate architecture for deep neural networks, and (4) adjusting hyperparameters.

The input for Eq. (3) can be aggregated into an array of sizes (128, 128, N_I), where N_I is the number of input fields that depends on the problem to be studied. $N_I = 2$ for undrained soil (input fields are $c'(\boldsymbol{x}')$ and $\gamma'(\boldsymbol{x}')$), and $N_I = 3$ for Mohr-Coulomb soil (input fields are $c'(\boldsymbol{x}')$, $\phi(\boldsymbol{x}')$, and $\gamma'(\boldsymbol{x}')$). In this study, three kinds of artificial neural networks are built to approximate the target function Eq. (3) and are examined. Readers can be referred to He et al., (2022) for the full description of these neural networks. In this paper, we focus on presenting technical parts that are critical for later discussion.

In this study, we investigated three different neural network structures: one-hidden-layer fully connected neural networks, convolutional neural networks and locally connected neural networks.

One-hidden-layer fully connected neural networks (FCN): the input is firstly flattened into a vector with size as 16,384 N_I, i.e., becomes a flatted layer with 16,384 N_I neurons. This flatted input layer is then connected to a hidden fully connected layer with N_F neurons. The output layer (one neuron for the NFS) is lastly connected after the hidden layer. In machine learning, deep fully connected networks (neural networks with multiple fully connected layers) are rarely used because they are computationally intensive and are prone to overfitting. The dropout layer is added with a rate $R = 0.4$.

Convolutional neural networks (CNN; Fig. 8a): the inputs first pass through several stacks of convolutional layers and average pooling layers. C_i is the size of channel in the i_{th} convolutional layer as shown in Table 3. A fully connected

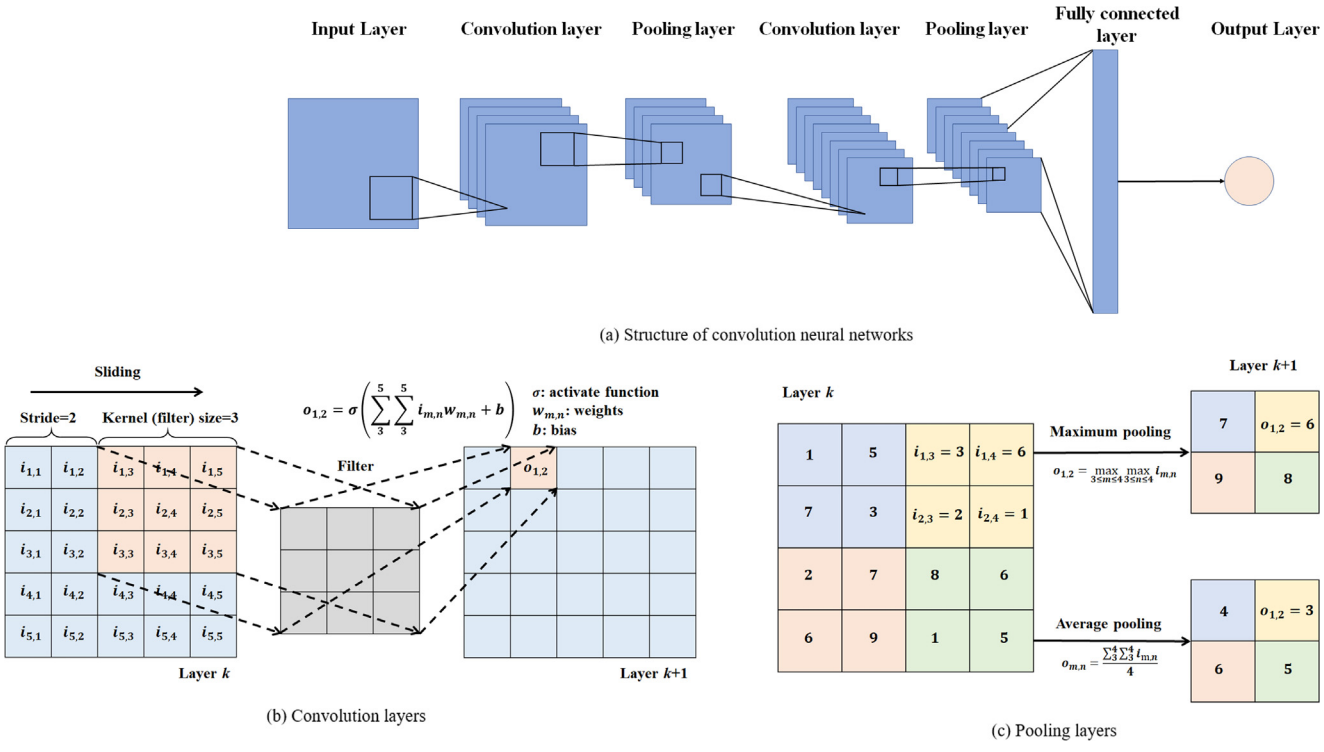
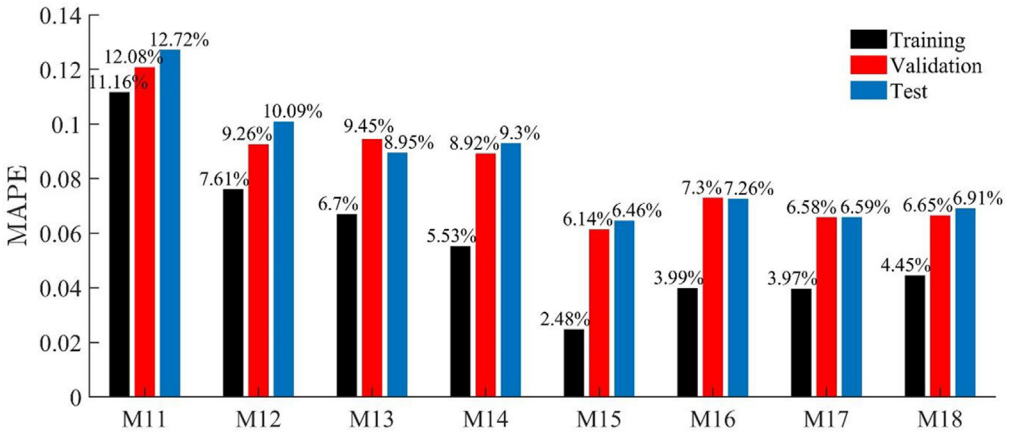
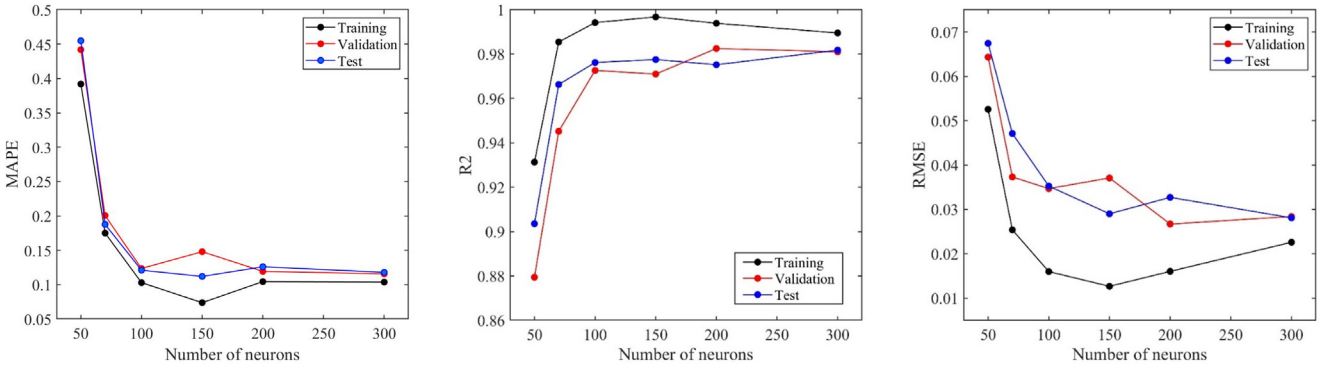


Fig. 8. Illustration of deep neural networks.

Table 3
Deep-learning models and performance for undrained soil with fixed slope shape (FC: Fully connected; CNN: Convolutional neural network; LCN: Locally connected network).



Model	M11	M12	M13	M14	M15	M16	M17	M18
description	FC	CNN	CNN with more channels	CNN with more layers	LCN	LCN with more channels in the locally connected layer	LCN with more convolutional channels	LCN with more convolutional layers
Hyper- parameters	$N_I = 2$ $N_F = 150$	$N_I = 2$ $C_1 = 8$ $C_2 = 16$ $C_3 = 32$ $N_F = 100$	$N_I = 2$ $C_1 = 16$ $C_2 = 32$ $C_3 = 64$ $N_F = 100$	$N_I = 2$ $C_1 = 8$ $C_2 = 16$ $C_3 = 32$ $C_4 = 64$ $N_F = 100$	$N_I = 2$ $C_1 = 32$ $C_2 = 16$ $C_L = 8$ $N_F = 400$	$N_I = 2$ $C_1 = 32$ $C_2 = 16$ $C_L = 16$ $N_F = 400$	$N_I = 2$ $C_1 = 64$ $C_2 = 32$ $C_L = 8$ $N_F = 400$	$N_I = 2$ $C_1 = 64$ $C_2 = 32$ $C_3 = 16$ $C_L = 8$ $N_F = 64$ $N_F = 400$



(a) MAPE (b) R2 (c) RMSE

Fig. 9. Errors of different evaluation metrics with various hyperparameters for the fully connected networks.

layer is followed by the last convolutional and average pooling layers with N_F neurons. An example of the convolutional layers is illustrated in Fig. 8b. The convolutional layer contains a set of filters (also called kernels) whose hyperparameters are to be learned. The filters have shared weights. The size of filters is smaller than the input size. They convolve with the input data from layer k, and an activation function is applied to generate the output data from layer k, which is then passed to the subsequent layer. Convolutional layers can extract features from the small size of input data and can maintain the spatial relationships between the inputs. In this study, the filter size is fixed as (3, 3) and the stride is fixed as (1, 1) in convolutional layers.

Pooling layers are usually followed by convolutional layers to further decrease the dimensionality of the input data and enhance feature extraction (He et al., 2022; Wang & Goh, 2022; Wu et al., 2022). The filter size and the stride size are often identical in pooling layers. Fig. 8c illustrates two types of pooling layers: the maximum pooling layer and the average pooling layer. Maximum pooling extracts the maximum input data over the filter size, while average pooling calculates the average input data over the filter size. In this study, the filter size and the stride are both fixed as (2, 2) in pooling layers (He et al., 2022; Wang & Goh, 2022; Wu et al., 2022).

Locally connected neural networks (LCN): the inputs first pass through several stacks of convolutional layers and pooling layers as above. A locally connected layer with the channel size C_L is followed by these convolutional layers and pooling layers as shown in Table 3. In the locally connected layer, each neuron is locally connected to a small region in the previous layer (He et al., 2022). A fully connected layer then connects the locally connected layer and the final output layer with N_F neurons. During the convolutional operation, the filters (kernels) remain constant across all positions. In other words, the same filter (kernel) weights are shared in all the pixel positions, which may overlook some minor differences in spatial arrangements. Conversely, in a locally connected layer, the filters (kernels) are distinct in different positions. This type of layer can study different characteristics in different regions. The locally connected layer has been successfully used in the DeepFace for face recognition (Taigman et al., 2014). The activation function for all the neurons is the rectified linear unit (ReLU) in this study.

4.1. Undrained soil with a fixed slope shape

The complex full problem is firstly simplified by ignoring Φ in Eq. (3) (zero for undrained soil) and fixing $\alpha = 0°$ and $\beta = 60°$. A deep-learning model is expected to be built to accurately predict the normalized factor of safety NFS from random fields $(\mathbf{C}', \Gamma')$ with fixed slope shape.

To obtain representative inputs, the quasi-random sequence is firstly used to generate 100 combinations for the five parameters: the COV of $\mathbf{C}'$ (in the range 0–0.55), the mean of Γ' (in the range 0–100), the COV of Γ' (in the range 0–0.1), l'_x (in the range 0.44–66) and l'_y (in the range 0.03–6.2). Then for each combination, 30 random field samples are generated, which leads to a dataset of size $N_{data2} = 3000$. Log-normal distributions are used to generate random fields. It has been validated that distribution types do not affect the results (He et al., 2022). The exponential autocorrelation function is used as $\rho_G(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{2|x-x'|}{l_x} - \frac{2|y-y'|}{l_y})$. Random field samples are generated by the open-source software 'GSTools', which uses the so-called randomization method. The spatial random field is represented by a stochastic Fourier integral, and its discretised modes are evaluated at random frequencies (Müller et al., 2021).

The input–output dataset is often separated into training, validation, and test datasets in machine learning. The training set is used to train the machine-learning models, while the validation set is used to validate model performance during training. The model is trained on the training set and evaluated on the validation set after every epoch. The test set is used to test the model only after completing the training. It provides an unbiased final model perfor-

mance metric in terms of accuracy, precision, etc. The test dataset will not participate in the training process.

In this study, the ratio of the training, validation and test datasets will be 70:15:15 (Gholamy et al., 2018; Joseph, 2022). Therefore, for this condition (3000 input–output pairs), the three datasets will be 2100, 450, and 450, respectively. Three evaluation metrics are compared in this section: mean absolute percentage error (MAPE), root mean square error (RMSE), and coefficient of determination (R2).

The first network examined is the one-hidden-layer FCN. Because this reduced problem has a fixed slope shape – some values of matrices ($\mathbf{C}'$ and Γ') are fixed to be zero (blue area in Fig. 2), in the flatted layer, these zeros are removed, which results in a flatted layer of smaller size. Firstly, a fully connected network is built without the dropout layer. The number of neurons in the hidden layer is a tuneable hyperparameter. Fig. 9a-c show the MAPE, R2 and RMSE of the trained FCN surrogate models with neurons from 50 to 300. Increasing the size of hidden neurons (from 50 to 300) means a more complex model and can improve the accuracy. However, there is no obvious improvement after the size exceeds 100. Also, when the size of hidden neurons is 150, the error of the training dataset (MAPE = 7.38%, RMSE = 0.0127, R2 = 0.997) is much smaller than the validation (MAPE = 14.8%, RMSE = 0.0371, R2 = 0.971) and test dataset (MAPE = 11.2%, RMSE = 0.029, R2 = 0.978), which indicates overfitting – the model is too close to the training dataset and cannot generalise to unseen data. In addition, Therefore, to solve this overfitting problem, a model (M11) is built by adding a dropout layer after the fully connected layer. The MAPEs of model M11 are 11.16%, 12.08%, and 12.72% (also shown in the bar chart in Table 3). The overfitting problem is solved with the dropout and M11 can be considered a robust model. Subsequent tables and figures only show the robust models.

In model M11, entries of the input matrices with zero values have been removed before conducting machine learning (blue area in Fig. 2). This can be considered as a filter applied to the original input to remove the insignificant features. In fact, deep learning is a powerful tool, in which filters are learnt automatically. For example, CNNs can automatically learn filters and obtain high-level features in image recognition. Considering the similarity between the inputs of this study and images, CNNs are also built in this study. It has been shown that average pooling layers have better performance than maximum pooling layers in geotechnical predictions involving mainly continuous numbers (He et al., 2022). Therefore, a CNN is firstly built with three consecutive convolutional layers and average-pooling layers. Parameters for the convolutions layer are shown in column M12 of Table 3. The effect of the number of neurons in the fully connected layer is examined, which varies from 20 to 100. Similarly, increasing the size of hidden neurons in the fully connected layer can improve the

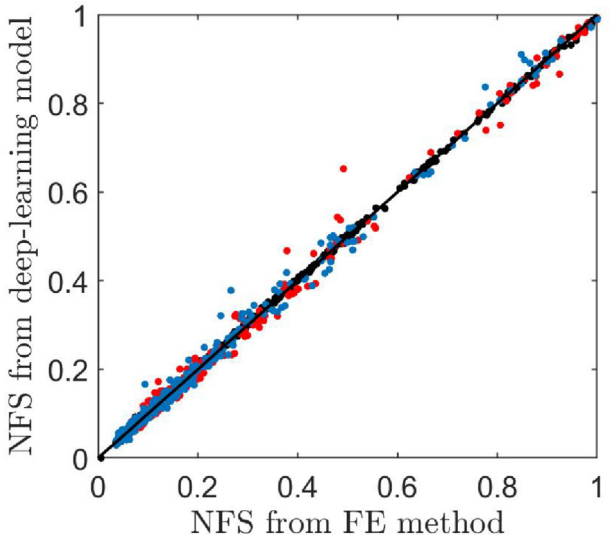


Fig. 10. NFS for undrained soil with fixed slope shape from FE simulations and from the deep-learning model (M15, 3000 data, black = training; red = validation, blue = test).

accuracy of the model, and a robust model is with N_F = 100, which is labelled as the model M12. This CNN (MAPE = 10.09%, RMSE = 0.0161, and R2 = 0.993 on the test dataset) performs better than the FCN M1 (MAPE = 12.72%, RMSE = 0.0353, and R2 = 0.976 on the test dataset).

In models M13 and M14, the number of channels and the number of convolutional layers are increased, respectively. The MAPEs of validation and test dataset are the same as M12. So, increasing the complexity of models cannot obtain better results in this situation. A test without a dropout layer is also conducted, for which the error of the training dataset (MAPE = 0.72%, RMSE = 0.0013, R2 = 0.999) is far smaller than the validation dataset (MAPE = 8.14%, RMSE = 0.0154, R2 = 0.994) and test dataset (MAPE = 8.22%, RMSE = 0.0151, R2 = 0.995), which means overfitting. Therefore, the dropout layer is necessary in CNNs for this slope stability problem.

For this slope stability problem, CNNs performed well. However, the filters in CNNs use shared parameters, while the soil properties in different areas have different effects on the NFS. Therefore, locally connected networks are expected to have the capability to reflect this feature. Table 3 contains the key parameters of locally connected networks. Based on model M12, a locally connected network is built by removing one convolutional layer and one average pooling layer while adding a locally connected layer between the second pooling layer and the fully connected layer. A larger stride is used in the locally connected layer to keep the number of trainable parameters low and to improve computation because locally connected layers often have a large number of trainable parameters. Similarly, the hyperparameter N_F – the number of neurons in the fully connected layer is examined, which varies from 50 to 500. The MAPEs do not change after N_F exceeds

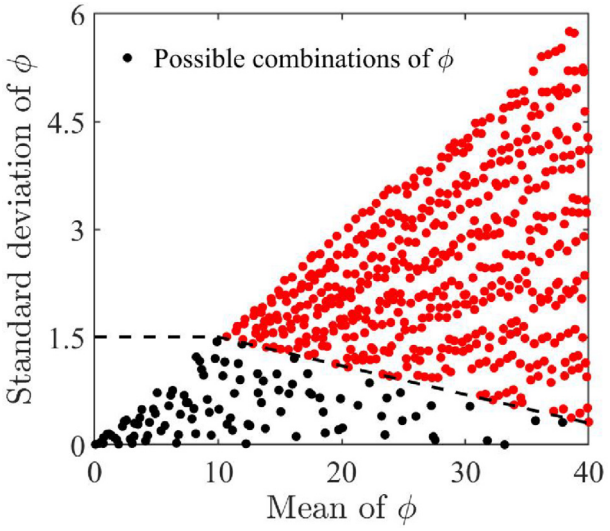


Fig. 11. Possible combination of the means and standard deviations of friction angles.

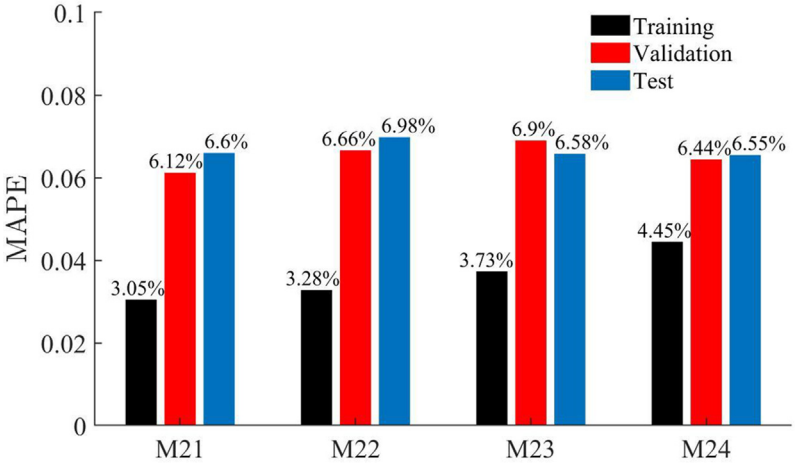
200. The neural network with 400 neurons is labelled M15. Fig. 10 compares the NFS obtained from FE simulations and from model M15, which demonstrates the accuracy of our deep-learning model. We also test three other locally connected networks: M16 adopts a larger filter in the locally connected layer, M17 and M18 have more channels and more convolutional layers, respectively. The performance of M16, M17and M18 indicate that increasing model complexity cannot improve accuracy. The bar graph in Table 3 shows the MAPE of the robust models. In general, locally connected networks give the best predictions and FCNs have the largest errors.

In this section, three evaluation metrics, MAPE, RMSE, and R2, consistently produce comparable results. R2 represents the proportion of the variation in the dependent variable that can be predicted by the independent variable, which reflects the overall fit of the model. MAPE and RMSE represent the relative and absolute differences between predicted and observed values. Fig. 9 illustrates the MAPE, R2, and RMSE of the trained FCN surrogate models. These three metrics provide the same conclusion that the model tends to be stable when the size of neurons exceeds 100. However, engineers typically place more emphasis on the accuracy of the prediction result in a specific geotechnical case rather than the overall fitting of all cases. Furthermore, RMSE is sensitive to outliers and may exaggerate the error value with a significant influence. In contrast, MAPE is less susceptible to outliers and is more robust than RMSE. Hence, we have opted to employ MAPE as the representative error metric in this study.

4.2. Mohr-Coulomb soil with a fixed slope shape

A more general problem is considered – Mohr-Coulomb soils with a fixed slope shape. The inputs contain three matrices ($\mathbf{C}'$, Φ and Γ'). Similarly, the quasi-random sequence is used to obtain 100 combinations for parameters and each combination is used to generate 30 random

Table 4
Deep-learning models, hyperparameters, and performance for Mohr-Coulomb soil with fixed slope shape (LCN: Locally connected network).



Model label & description	M21: LCN	M22: LCN with more channels in the locally connected layer	M23: LCN with more convolutional channels	M24: LCN with more convolutional layers
Model structure parameters	$N_I = 3$ $C_1 = 32$ $C_2 = 16$ $C_L = 8$ $N_F = 400$	$N_I = 3$ $C_1 = 32$ $C_2 = 16$ $C_L = 16$ $N_F = 400$	$N_I = 3$ $C_1 = 64 C_2 = 32$ $C_L = 8$ $N_F = 400$	$N_I = 3$ $C_1 = 64$ $C_2 = 32$ $C_3 = 16$ $C_L = 8$ $N_F = 400$

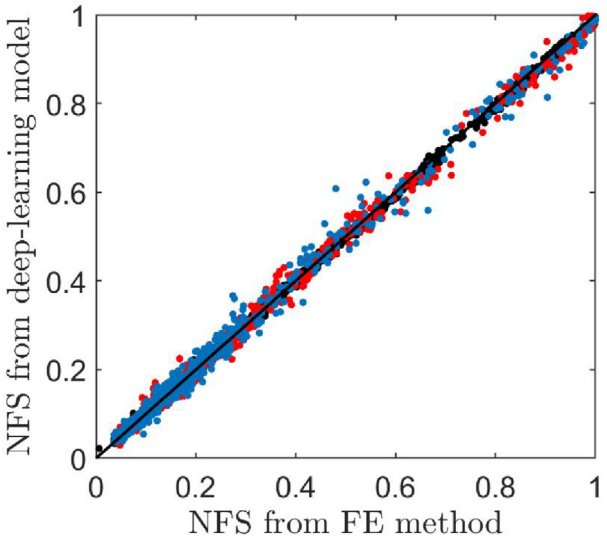
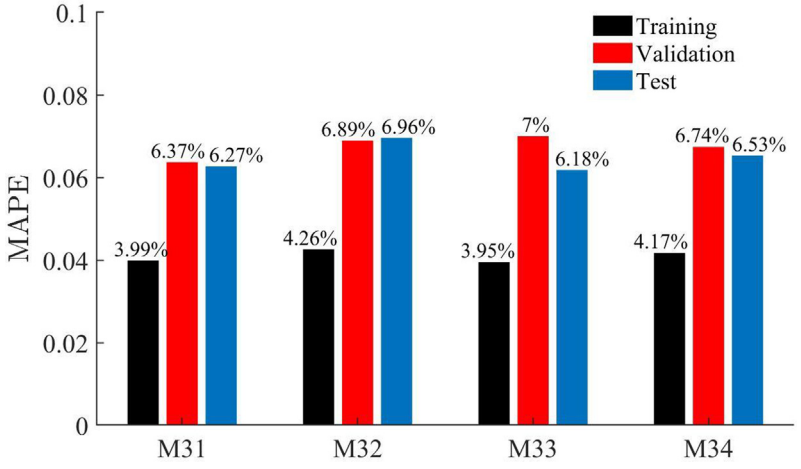


Fig. 12. NFS of Mohr-Coulomb soil with a fixed slope shape from FE simulations and from the deep-learning model (M21, 6000 data, black = training; red = validation, blue = test).

field samples, which leads to another $N_{data3} = 3000$ input data. It must be noted that Table 2 specifies the possible mean and COV of friction angles. All the dots in Fig. 11 represent possible combinations for the mean and standard deviation. When the mean and standard deviation are both very large (like 40° and 6°), it is possible to generate random field samples with friction angles as large as 60-80° and even>90°, which is impossible in practice. Therefore, we apply some constraints on the mean and standard deviation of the friction angle (Fig. 11). When the mean of the friction angle is less than 10°, the possible COV is still 0–0.15. When the mean of the friction angle is>10°, the standard deviations decrease linearly from 1.5° to 0.3°.

From the previous problem, locally connected networks perform best, so we will only examine locally connected networks for this problem. All key hyperparameters of neural networks are listed in Table 4. Considering that the first problem is a subset of the present problem, the total dataset contains $N_{data2} + N_{data3} = 6000$ data. A locally connected network is firstly built similar to model M15 but with inputs of three matrices. The optimal N_F is 400 and the MAPEs of training, validation and test are 3.05%, 6.12%, and 6.6%, respectively (also shown as M21 in the bar graph in Table 4). To examine the effect of hyperparameters on the predicted results, several models are built based on M21: Model M22 uses twice the channels in the locally connected layer, and Model M23 and M24 increase the number of channels and the number of convolutional layers, respectively. The bar graph (Table 4) shows that models M22 to M24 have no significant improvement in prediction compared with M21. The NFSs obtained from FE simulations and from the deep-learning model (M21) are compared in Fig. 12. Our deep-learning surrogate model has a low error of only 6.6% on the test dataset.

Table 5
Deep-learning models, hyperparameters, and performance for the full problem (LCN: Locally connected network).



Model label & description	M31: LCN	M32: LCN with more channels in the locally connected layer	M33: LCN with more convolutional channels	M34: LCN with more convolutional layers
Model structure parameters	$N_I = 3$ $C_1 = 32$ $C_2 = 16$ $C_L = 8$ $N_F = 400$	$N_I = 3$ $C_1 = 32$ $C_2 = 16$ $C_L = 16$ $N_F = 400$	$N_I = 3$ $C_1 = 64$ $C_2 = 32$ $C_L = 8$ $N_F = 400$	$N_I = 3$ $C_1 = 64$ $C_2 = 32$ $C_3 = 16$ $C_L = 8$ $N_F = 400$

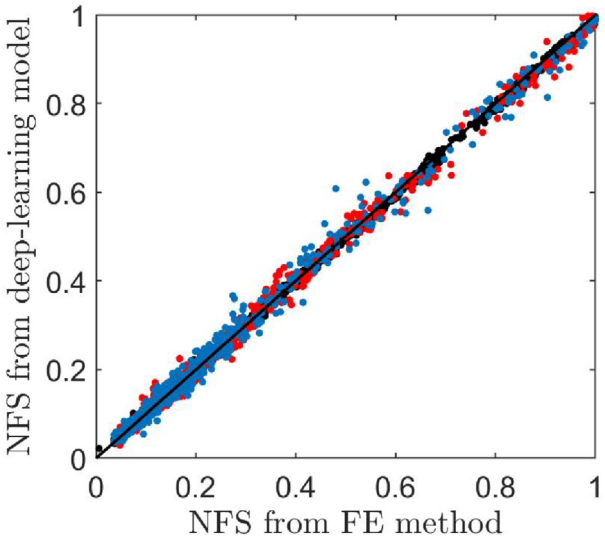


Fig. 13. NFS of the full problem from FE simulations and from the deep-learning model (M31, 12,255 data, black = training; red = validation, blue = test).

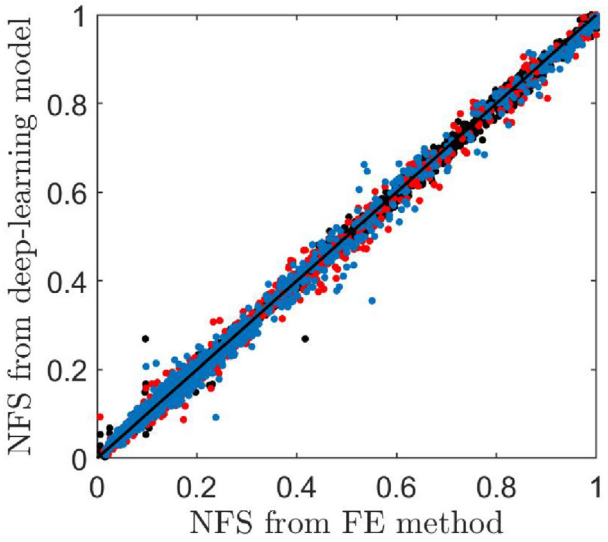


Fig. 14. Slope surface shapes and random fields of soil parameters.

Table 6
Parameters of slopes for validation.

	Parameters
Case 1	The COV of $\mathbf{C}' = 0.4$, the mean of$\Gamma' = 50$, the COV ofΓ' **= 0.05**, the mean of $\Phi = 0$, the COV of $\Phi = 0.05$, $l'_x = 33$, $l'_y = 3.1$, $\alpha = 10°$, and $\beta = 50°$
Case 2	Same as Case 1 but the mean of$\Gamma' = 90$
Case 3	Same as Case 1 but $\beta = 80°$
Case 4	Same as Case 1 but $\alpha = 20°$
Case 5	Same as Case 1 but the mean of $\Phi = 30°$
Case 6	Same as Case 5 but concave slope surface
Case 7	Same as Case 5 but convex slope surface

4.3. The full problem

The above two reduced problems are all with a fixed slope shape (α = *0°* and β = *60°*). In this section, the full problem is considered – the inputs are three matrices ($\mathbf{C}'$, Φ and Γ'), and the distributions of zeros within these matrices (position not occupied by soils) vary with different slope shapes (i.e., angles α and β). The parameters that characterise this problem are the COV of $\mathbf{C}'$ (in the range 0–0.55), the mean of Γ' (in the range 0–100), the COV of Γ' (in the range 0–0.1), the mean of Φ (in the range 0-40°), the COV of Φ (in the range 0–0.15), l'_x (in the range 0.44–66), l'_y (in the range 0.03–6.2), α (in the range 0-40°), and β (in the range 45-90°). Firstly, 200 combinations of these nine parameters are generated using the quasi-random sequence and 30 random field samples are generated for each combination, resulting in $N_{data4} = 6000$ data. All the input–output data prepared in this study is used for training, i.e., $N_{data1} + N_{data2} + N_{data3} + N_{data4} = 12255$ data. Similarly, we only test locally connected neural networks. Firstly, model M31 is a locally connected network like M15 and its accuracy is 6.27% on the test dataset (Table 5). Models M32 to M34 are built to study the effect of hyper-parameters. The results (The bar graph in Table 5) indicate that model M31 is a good model and that any more complex structures cannot improve the accuracy of predictions. The NFSs from FE simulations and from the deep-learning model (M31) are compared in Fig. 13.

For the full slope stability problem in this section, a large dataset (>10,000 input–output pairs) is used to train deep neural networks. This dataset covers all possible combinations of soil material parameters and boundary conditions (slope angles in this study), and the trained model is accurate with a MAPE of about 6%, so this deep-learning model is an ideal surrogate model for stochastic analysis of slopes considering spatial variability. Its accuracy and applications in stochastic analysis are further demonstrated in the next section.

5. Verification and application

Seven particular cases are designed to verify the predictive power of the trained deep-learning model and to demonstrate how it can increase the computational efficiency of stochastic analyses. The parameters are listed in Table 6. Case 1 is a reference case, In Cases 2–5, only one parameter is altered compared with the reference case. We also test if our trained model is able to predict the FS of slopes with curved slope surfaces. Cases 6 and 7 have the same parameters as Case 5 but Case 6 has a concave slope surface (Fig. 14a) and Case 7 has a convex slope surface (Fig. 14b). The black dotted lines connecting the top and

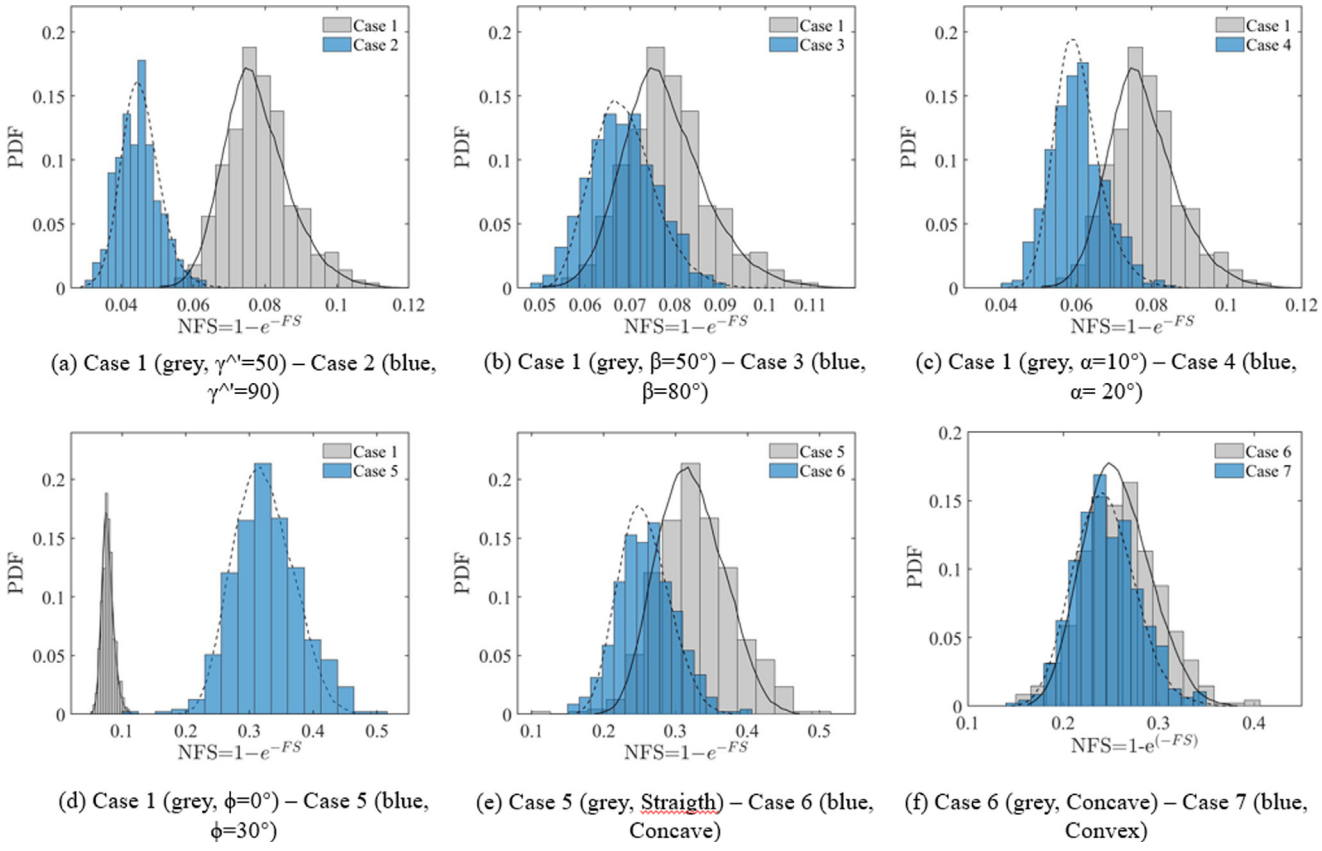


Fig. 15. PDFs for seven cases of Mohr-Coulomb soils with different parameters and slope shapes (Histograms = PDFs from brute-force analyses; solid and dotted lines = PDFs from deep-learning models).

shoulder represent the slope angles β of the two curved slope surfaces.

For each case, ten thousand samples are generated according to the parameters (Table 6). Brute-force Monte Carlo analyses are conducted for only 500 samples, and each analysis will take about 5 h. However, the evaluation of the NSF for the ten thousand samples with our deep-learning model only costs several minutes, which is a great improvement in computational efficiency.

The estimated PDFs for the seven cases are shown in Fig. 15. The histograms represent the brute-force analyses with 500 samples and the black lines represent the results from the deep-learning model with ten thousand samples. Fig. 15a shows that the NFS decreases with the increase of γ' (Case 2 compared with Case 1), which is consistent with the deterministic analysis in Section 3 (Fig. 6c). In addition, larger γ' leads to a smaller spread of NFS. Fig. 15b and c indicate that an increase in α and β will lead to a decrease in NFS, which corresponds to the deterministic analysis in Fig. 6e and f. The increase of β here leads to a wider spread of NFS while the increase of α leads to a narrower spread of NFS. Fig. 15d shows the influence of friction angle on the NFS. Similar to the deterministic analysis (Fig. 6d), a larger friction angle will lead to safer slopes.

Fig. 15e and f show how the shape of the slope surface affects the stability. A slope with straight slope surfaces (Case 5 and grey in Fig. 15e) is safer than slopes with curved surfaces (Case 6 and 7). Comparing convex and concave slopes, the concave (Case 6 and grey in Fig. 15f) is slightly safer than the convex slope (Case 7 and blue in Fig. 15f), which is because of the larger sliding mass of the convex slope. PDFs from the deep-learning model show good agreement with brute-force analyses and achieve higher accuracy and detail. Therefore, our deep-learning model is an ideal surrogate model for efficient slope stability analysis.

6. Conclusion

This paper presents an accurate deep-learning surrogate model for slope stability problems. A big dataset covering the common soil properties, spatial variabilities, and boundary conditions is used for training. Therefore, the trained model is ready to make predictions without additional numerical simulations and training required.

We conducted dimensional analysis first. The factor of safety is expressed as a function of several inputs – dimensionless fields of soil parameters (cohesion, unit weight, and friction angle), which are represented as matrices. Since

slopes are of irregular shapes, a larger square simulation domain is used to contain all possible slope shapes and is discretised into 128 × 128 quadrilateral elements. The simulation domain outside slopes is assumed to have material properties of zeros. So, different slopes are represented by the distribution of different material parameters and zeros with square matrices. Non-uniform meshes are used in the FE simulations. A coarser mesh is adopted near the toe of the slope to improve the convergence of calculation.

We rigorously verified our method of calculating the factor of safety with FE models, in which gravity is increased in linear increments. Over a wide range of parameters (e.g., various non-dimensional unit weights, friction angles, and slope angles), our method predicts the factor of safety accurately against the analytical solution.

Three conditions are considered from reduced problems with fewer inputs to the full problem. A comparison of the three kinds of artificial neural networks is conducted in the first simplified condition. The mean absolute percentage error is used as the training loss and evaluation criteria. It is found that locally connected networks perform bests (MAPE of about 6%), and CNNs come the next (about 10%). The error of fully connected networks is relatively large (about 13%). Subsequent studies only consider locally connected networks. The final full problem was trained with over 12,000 input–output data and the locally connected network gives accurate predictions (MAPE of 6.27%), so it can be used as an ideal surrogate model for the slope stability problem.

Seven cases are designed to verify the performance of the deep-learning model, including different soil parameters, slope shapes, and two different kinds of slope surfaces (e.g., concave and convex slopes). The results indicate the probability density functions from deep-learning models are highly consistent with brute-force Monte Carlo analyses with FE simulations but take much less computing effort.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Atamturktur, S., Hemez, F., Williams, B., Tome, C., Unal, C., 2011. A forecasting metric for predictive modeling. Comput. Struct. 89 (23–24), 2377–2387.

Beck, J.L., Au, S.-K., 2002. Bayesian updating of structural models and reliability using Markov Chain Monte Carlo simulation. J. Eng. Mech. 128 (4), 380–391. https://doi.org/10.1061/(ASCE)0733-9399(2002)128:4(380).

Chen, X., Li, J., 2017. A subset multicanonical Monte Carlo method for simulating rare failure events. J. Comput. Phys. 344, 23–35.

W.F. Chen. (1976). W.F. Chen, Editor, Limit Analysis and Soil Plasticity, Elsevier, Amsterdam (1975), p. 638 Dfl. 240.00. *Engineering Geology*, *10*(1), 79. 10.1016/0013-7952(76)90010-7.

Dawson, E.M., Roth, W.H., Drescher, A., 1999. Slope stability analysis by strength reduction. Geotechnique 49 (6), 835–840. https://doi.org/10.1680/geot.1999.49.6.835.

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). *Why 70/30 or 80/20 relation between training and testing sets: a pedagogical explanation.*

Griffiths, D. v., & Fenton, G. A. (2001). Bearing capacity of spatially random soil: The undrained clay Prandtl problem revisited. *Geotechnique*, *51*(4), 351–359. 10.1680/geot.2001.51.4.351

Griffiths, D. v., & Lane, P. A. (1999). Slope stability analysis by finite elements. *Geotechnique*, *49*(3), 387–403. 10.1680/geot.1999.49.3.387

He, X., Xu, H., Sabetamal, H., Sheng, D., 2020. Machine learning aided stochastic reliability analysis of spatially variable slopes. Comput. Geotech. 126 (April). https://doi.org/10.1016/j.compgeo.2020.103711 103711.

He, X., Wang, F., Li, W., Sheng, D., 2021. Efficient reliability analysis considering uncertainty in random field parameters: trained neural networks as surrogate models. Comput. Geotech. 136. https://doi.org/10.1016/j.compgeo.2021.104212 104212.

He, X., Wang, F., Li, W., Sheng, D., 2022. Deep learning for efficient stochastic analysis with spatial variability. Acta Geotech. 17 (4), 1031–1051. https://doi.org/10.1007/s11440-021-01335-1.

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*

Kang, F., Xu, Q., Li, J., 2016. Slope reliability analysis using surrogate models via new support vector machines with swarm intelligence. App. Math. Model. 40 (11–12), 6105–6120. https://doi.org/10.1016/j.apm.2016.01.050.

Kaur, A., Sharma, R.K., 2016. Slope stability analysis techniques: a review. Int. J. Eng. Appl. Sci. Technol. 1 (4), 52–57.

Kroese, D.P., Brereton, T., Taimre, T., Botev, Z.I., 2014. Why the Monte Carlo method is so important today. Wiley Interdiscip. Rev. Comput. Stat. 6 (6), 386–392.

Müller, S., Schüler, L., Zech, A., Heße, F., 2021. GSTools v1.3: a toolbox for geostatistical modelling in Python. Geosci. Model Dev. Discuss. 2021, 1–33. https://doi.org/10.5194/gmd-2021-301.

Pang, R., Xu, B., Kong, X., Zou, D., Zhou, Y., 2018. Seismic reliability assessment of earth-rockfill dam slopes considering strain-softening of rockfill based on generalized probability density evolution method. Soil Dyn. Earthq. Eng. 107, 96–107.

Phoon, K.-K., & Zhang, W. (2022). Future of machine learning in geotechnics. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 1–16. 10.1080/17499518.2022.2087884.

Phoon, K.-K., Kulhawy, F.H., 1999. Characterization of geotechnical variability. Can. Geotech. J. 36 (4), 612–624. https://doi.org/10.1139/t99-038.

Pourkhosravani, A., Kalantari, B., 2011. A review of current methods for slope stability evaluation. Electron. J. Geotech. Eng. 16, 1245–1254.

Rice, J.D., Polanco, L., 2012. Reliability-based underseepage analysis in levees using a response surface–Monte Carlo simulation method. J. Geotech. Geoenviron. Eng. 138 (7), 821–830.

Shields, P. (2013). *Evaluate the predictive capability of numerical models considering robustness to non-probabilistic uncertainty in the input parameters.*

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D., 2016. Mastering the game of Go with deep neural networks and tree search. Nature 529 (7587), 484–489. https://doi.org/10.1038/nature16961.

Sloan, S.W., 1988. Lower bound limit analysis using finite elements and linear programming. Int. J. Numer. Anal. Meth. Geomech. 12 (1), 61–77. https://doi.org/10.1002/nag.1610120105.

SravyaPranati, Bh., Suma, D., ManjuLatha, Ch., & Putheti, S. (2021). *Large-Scale Video Classification with Convolutional Neural Networks* (pp. 689–695). 10.1007/978-981-15-7062-9_69.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. IEEE Conf. Comput. Vision Pattern Recogn. (CVPR) 2015, 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1701–1708.

Wang, Z. Z., Goh, S. H., & Zhang, W. (2022). Reliability-based design in spatially variable soils using deep learning: an illustration using shallow foundation. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 1–15. 10.1080/17499518.2022.2083178.

Wang, Z.Z., Goh, S.H., 2022. A maximum entropy method using fractional moments and deep learning for geotechnical reliability analysis. Acta Geotech. 17 (4), 1147–1166. https://doi.org/10.1007/s11440-021-01326-2.

Wang, L., Wu, C., Tang, L., Zhang, W., Lacasse, S., Liu, H., Gao, L., 2020. Efficient reliability analysis of earth dam slope stability using extreme gradient boosting method. Acta Geotech. 15, 3135–3150.

Wu, C., Hong, L., Wang, L., Zhang, R., Pijush, S., Zhang, W., 2022. Prediction of wall deflection induced by braced excavation in spatially variable soils via convolutional neural network. Gondw. Res. https://doi.org/10.1016/j.gr.2022.06.011.

Zhang, J., Li, W., Lin, G., Zeng, L., Wu, L., 2017. Efficient evaluation of small failure probability in high-dimensional groundwater contaminant transport modeling via a two-stage M onte C arlo method. Water Resour. Res. 53 (3), 1948–1962.