INTERNATIONAL JOURNAL FOR NUMERICAL AND ANALYTICAL METHODS IN GEOMECHANICS
Int. J. Numer. Anal. Meth. Geomech. 2016; **40**:1556–1569
Published online 18 January 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nag.2496

Neural network prediction of the reliability of heterogeneous cohesive slopes

Y.H. Chok[1,4,*,†], M.B. Jaksa[1], W.S. Kaggwa[1], D.V. Griffiths[2] and G.A. Fenton[3]

[1] *School of Civil, Environmental and Mining Engineering, The University of Adelaide, Adelaide, SA 5005, Australia*
[2] *Department of Civil and Environmental Engineering, Colorado School of Mines, Golden, CO 80401, U.S.A.*
[3] *Department of Engineering Mathematics, Dalhousie University, Halifax, NS B3J 2X4, Canada*
[4] *AECOM, Fortitude Valley, QLD 4006, Australia*

SUMMARY

The reliability of heterogeneous slopes can be evaluated using a wide range of available probabilistic methods. One of these methods is the random finite element method (RFEM), which combines random field theory with the non-linear elasto-plastic finite element slope stability analysis method. The RFEM computes the probability of failure of a slope using the Monte Carlo simulation process. The major drawback of this approach is the intensive computational time required, mainly due to the finite element analysis and the Monte Carlo simulation process. Therefore, a simplified model or solution, which can bypass the computationally intensive and time-consuming numerical analyses, is desirable. The present study investigates the feasibility of using artificial neural networks (ANNs) to develop such a simplified model. ANNs are well known for their strong capability in mapping the input and output relationship of complex non-linear systems. The RFEM is used to generate possible solutions and to establish a large database that is used to develop and verify the ANN model. In this paper, multi-layer perceptrons, which are trained with the back-propagation algorithm, are used. The results of various performance measures indicate that the developed ANN model has a high degree of accuracy in predicting the reliability of heterogeneous slopes. The developed ANN model is then transformed into relatively simple formulae for direct application in practice. Copyright © 2016 John Wiley & Sons, Ltd.

Received 21 June 2015; Revised 5 December 2015; Accepted 9 December 2015

KEY WORDS: artificial neural network; finite element method; random field; probability of failure; heterogeneous slope

1. INTRODUCTION

Probabilistic slope stability analysis has been widely used to evaluate the reliability of heterogeneous slopes. The aim is to incorporate uncertainties due to soil properties, that is, soil spatial variability, into the slope stability analysis. One of the available probabilistic slope stability analysis methods is the random finite element method (RFEM) developed by Griffiths and Fenton [1, 2]. The RFEM combines random field theory with the non-linear elasto-plastic finite element slope stability analysis method. This approach computes the probability of failure of a slope via the Monte Carlo simulation process. The spatial variability of soil properties is modelled by a random field, which can be described concisely by the mean, coefficient of variation (COV) and the scale of fluctuation. The Monte Carlo simulation involves repeated finite element slope stability analysis of a large number of generated random fields that simulate the spatial variability of the soil properties. In order to obtain

*Correspondence to: Y.H. Chok, AECOM, 540 Wickham Street, PO Box 1307, Fortitude Valley QLD 4006, Australia.
†E-mail: yunhang.chok@aecom.com

a representative and reproducible estimation of the probability of failure of a slope, a very large number of repetitions or realizations are required. Hence, the major drawback of the RFEM and Monte Carlo simulation is that they are computationally intensive. Performing parametric studies or sensitivity analysis using this approach is therefore a time-consuming task.

This paper investigates the feasibility of developing a meta-model using a computational tool known as artificial neural networks (ANNs) to predict the reliability of heterogeneous slopes. Ideally, the developed ANN model is capable of predicting the probability of failure of a slope close to that calculated by using the RFEM approach. The advantage of the developed ANN model is that it has a significantly less computational time than the RFEM approach, although a large number of RFEM analyses are initially required to compile a database that is used to develop and validate the ANN model.

2. OVERVIEW OF ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are a class of non-linear computational tools that attempt to mimic the way in which the human brain processes information. ANNs have been successfully applied to a wide range of geotechnical problems [3–5]. The most widely used ANNs are the multi-layer perceptrons (MLPs) that are trained with the back-propagation algorithm [6]. A comprehensive description of back-propagation MLPs is beyond the scope of this paper and can be found in the many publications (e.g.[7–10]). The typical structure of MLPs consists of a number of interconnected neurons or processing elements. The neurons are logically arranged into layers: an input layer, one or more intermediate layers called hidden layers and an output layer. Each neuron is connected to all the neurons in the next layer via weighted connections. The scalar weights determine the strength of influence between the interconnected neurons. The input from each neuron in the previous layer (x_i) is multiplied by an adjustable connection weight (w_{ji}). The weighted input values, at each neuron, are summed, and a threshold value or bias (B_j) is added or subtracted. The result (I_j) is then passed through a non-linear transfer function ($f(.)$) (e.g. *sigmoid* or *tanh* function) to produce the output of the neuron (y_i). The output of one neuron provides the input to the neurons in the next layer. This process is summarized in Eqns (1) and (2) as follows:

$$I_j = \sum_{i=1}^{n} w_{ji} x_i + B_j \tag{1}$$

$$y_i = f\left(I_j\right) \tag{2}$$

where I_j = the activation level of node j; w_{ji} = the connection weight between nodes i and j; x_i = the input from node i, $i = 0, 1, \ldots, n$; B_j = the bias for node j; y_j = the output of node j; and $f(I_j)$ = the transfer function.

The basic methodology of ANNs consists of three stages; (i) network training, (ii) network testing and (iii) network validation. ‘Training’ of the neural network involves adjusting the connection weights by repeatedly presenting a historical set of model inputs and the corresponding (desired) outputs. The objective is to minimise the errors between the predicted and desired output values. The number of training samples presented between weight updates is called an epoch. This iterative process of correcting the weights at the completion of each epoch until the errors are minimal is based on the gradient-descent technique. The process of training continues until some stopping criterion is achieved so that the network can obtain a set of weights that will produce the smallest possible error. In this study, the cross-validation technique [11] is used as a stopping criterion to ensure that overfitting does not occur [12]. Cross-validation also known as the testing stage of the ANN model’s development in which the data set is used to decide when to stop training to avoid overfitting. Finally, a validation data set, which has not been used in training nor testing of the ANN models, is used to assess the accuracy and predictive ability of the model.

3. DATA GENERATION USING RANDOM FINITE ELEMENT METHOD

To generate data for developing ANN models, a parametric study is carried out, using the RFEM, to investigate the effect of soil variability on the reliability of cohesive slopes with different geometries by varying the slope angle (β) and depth factor (D). The geometry of the slope problem is shown in Figure 1.

The slope is assumed to consist entirely of purely cohesive soil under undrained conditions ($\phi_u = 0$), with the soil strength described by the undrained shear strength (c_u). In the context of probabilistic analysis, c_u is assumed to be lognormally distributed, which is characterized by a mean (μ) and a standard deviation (σ). The mean and standard deviation can be expressed in terms of the dimensionless COV, defined as follows:

$$COV = \sigma/\mu \tag{3}$$

The spatial variability of soil properties is modelled by the scale of fluctuation (θ). A large value of θ implies a more smoothly varying field, while a small value of θ indicates one that varies more randomly. In this study, the correlation structure of soil properties is defined by an exponentially decaying (Markovian) correlation function as follows:

$$\rho(\tau) = \exp\left(-\frac{2|\tau|}{\theta}\right), \tag{4}$$

where ρ = correlation coefficient between the underlying random field values at any two points separated by a lag distance τ.

In the interest of generality, the scale of fluctuation (θ) was normalised with respect to the slope height (H) (i.e. θ/H), while the mean value of the c_u was expressed in terms of a dimensionless stability coefficient, N_s, similar to Taylor's [13] stability number, and expressed as follows:

$$N_s = \frac{\mu_{c_u}}{\gamma H}, \tag{5}$$

where μ_{c_u} = the mean value of c_u; γ = unit weight of the soil; and H = slope height as shown in Figure 1.

The random field of shear strength values is simulated using the local average subdivision method [14]. The local average subdivision algorithm generates correlated random variables in which their correlation is governed by the correlation function in Eqn (4). These random variables are then mapped onto the finite element mesh. The element size used in the finite element analysis is fixed at 1 m by 1 m for all slope geometries. Therefore, each element in the mesh is assigned a random variable, and neighbouring elements are correlated with each other. Figure 2 shows two typical random field realizations of c_u with different scales of fluctuation. The darker elements indicate stronger soils.

The finite element analysis is based on an elastic-perfectly plastic stress–strain law with a Tresca failure criterion. It uses eight-noded quadrilateral elements and reduced integration in both the stiffness and stress distribution parts of the algorithm. The plastic stress distribution is accomplished by using a visco-plastic algorithm. The theoretical basis of the method is described more fully in the

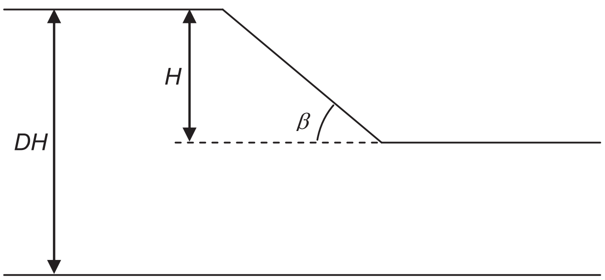

Figure 1. Geometry of the cohesive slope problem.

DOI: 10.1002/nag

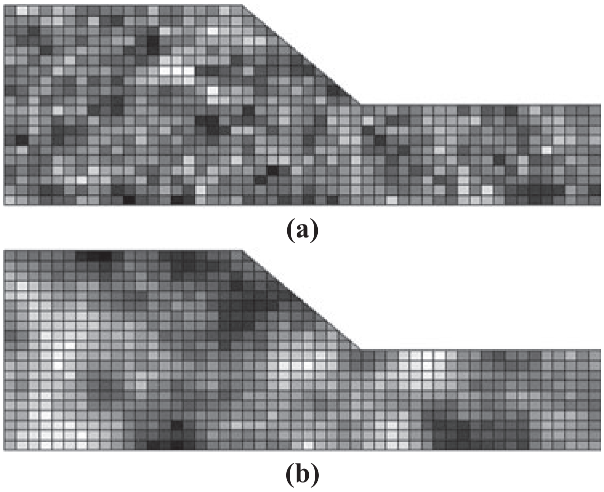

Figure 2. Typical random field realizations of undrained shear strength (*cu*) with scales of fluctuation of (a) $\theta/H=0.1$ and (b) $\theta/H=1$.

text by Smith and Griffiths [15]. The application of the finite element method in slope stability analysis is described more fully by Griffiths and Lane [16].

Briefly, the analyses involve the application of gravity loading and the monitoring of stresses at all Gauss points. If the Tresca failure criterion is violated, the programme attempts to redistribute excess stress to neighbouring elements that still have reserve strength. This iterative process continues until the Tresca failure criterion and global equilibrium are satisfied at all points within the mesh under strict tolerances.

Based on a given set of statistics (μ, σ, θ) for a soil property, multiple possible random fields can be generated. These random fields are then mapped directly to the finite elements to establish the input soil properties for slope stability analysis. For each generated random field, a single finite element analysis is performed. Hence, in this context, probabilistic analysis involves the repeated finite element analysis for each realization or repetition of soil properties as part of the Monte Carlo simulation process. The probability of failure (p_f) of a slope is therefore estimated by the following:

$$p_f \approx \frac{n_f}{n}, \tag{6}$$

where n is the total number of realizations in the simulation process and n_f is the number of realizations where slope failure occurred. Slope failure is indicated by non-convergence of the finite element algorithm after 500 iterations [2]. In this study, 2000 realizations of the Monte Carlo simulation were performed for each RFEM run to estimate the probability of failure. Based on 2000 independent realizations, the estimated probability of failure, $\hat{p}_f$, has standard error $\sqrt{\hat{p}_f\left(1-\hat{p}_f\right)/n}$, which for a probability level of 5% is 0.5%.

The input parameters β, D, N_s, COV and θ/H are varied systematically, and the values adopted in the parametric studies are given in Table I. The slope angles, β, of 14°, 18.4°, 26.6° and 45°, which represent slopes of 4H:1V, 3H:1V, 2H:1V and 1H:1V, respectively, are adopted. The scale of fluctuation (θ) is assumed to be isotropic, that is, the horizontal θ and the vertical θ are equal. Other parameters are held constant at their deterministic values, that is, slope height, $H=10\,\text{m}$; unit weight, $\gamma=20\,\text{kN/m}^3$; Young's modulus, $E_s=1\times10^5\,\text{kPa}$; Poisson's ratio, $v=0.3$; and dilation angle, $\psi=0°$.

4. DEVELOPMENT OF NEURAL NETWORK MODELS

The development of ANN models essentially involves a number of steps, which are carried out in a systematic manner [17]. These include the determination of model inputs and outputs, division and

Table I. Input values for parametric studies.

Parameters	Input values
β	14°, 18.4°, 26.6°, 45°
D	1, 2, 3
N_s	0.1, 0.2, 0.3, 0.4, 0.5
COV	0.1, 0.3, 0.5, 1.0
θ/H	0.1, 0.2, 0.5, 1, 5, 10

pre-processing of the available data, the determination of appropriate network architecture, optimization of the connection weights (training), stopping criteria and model validation. In this study, the ANN models are developed using the software *NEUFRAME* Version 4 [18]. The data used to calibrate and validate the ANN model are obtained from the parametric studies performed by the RFEM. There are a total of 1440 individual case records and each with a different combination of input and output values.

Five parameters have a significant impact on the probability of failure (p_f) of a heterogenous cohesive slope. They are the slope angle (β), depth factor (D), stability number (N_s), COV and normalised scale of fluctuation (θ/H). These five parameters are used as the input variables for the ANN model. The model output is therefore the probability of failure (p_f). The range of each parameter examined in the analyses that follow is given in Table II.

As cross-validation [11] is used as the stopping criteria in calibrating the ANN model, the data are randomly divided into three sets: training, testing and validation. In total, 80% (i.e. 1152 case records) of the data are used for model calibration and 20% (i.e. 288 case records) are used for model validation. The calibration data are further divided into 70% (i.e. 806 case records) for the training set and 30% (i.e. 346 case records) for the testing set. The data division for ANN model development is illustrated in Figure 3.

When dividing the data, it is essential to ensure that the data used for training, testing and validation represent the same statistical population, as recommended by Masters [19]. This is achieved by randomly choosing several combinations of the training, testing and validation sets until three statistically consistent data sets are obtained, as proposed by Shahin *et al.* [20]. The statistics of the data used for the training, testing and validation sets are given in Table III, which includes the mean, standard deviation, maximum, minimum and range. It can be seen that the data used for the training, testing and validation are statistically consistent, which suggest that they represent similar statistical populations.

Once the available data have been divided into their subsets, the input and output variables are pre-processed by scaling them between 0.0 and 1.0 to eliminate their dimension and to ensure that all variables receive equal attention during training [19]. A simple linear mapping of the variables' extreme to the neural network's practical extreme is adopted for scaling, as it is the most commonly used method for this purpose [19]. For a variable, x, with a maximum and minimum values of $x_{\max}$ and $x_{\min}$, respectively, the scaled value, x_n, is calculated as follows:

$$x_n = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{7}$$

Table II. Summary of range for input and output variables.

Input/output variables	Range
Slope angle (β)	14–45°
Depth factor (D)	1–3
Stability number (N_s)	0.1–0.5
Coefficient of variation (COV)	0.1–1.0
Normalised scale of fluctuation (θ/H)	0.1–10
Probability of failure (p_f)	0–1

DOI: 10.1002/nag

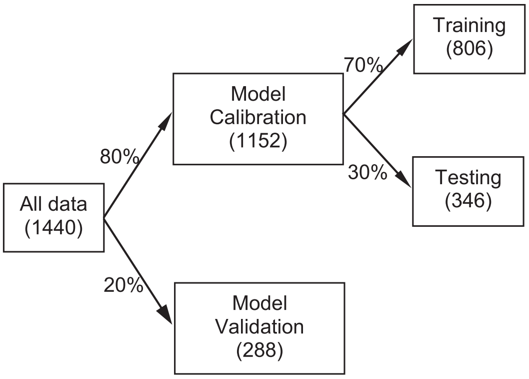

Figure 3. Data division for artificial neural network model development.

After establishing the required three data sets: training, testing and validation, the next step is to 'train' the ANN model. 'Training' involves repeatedly presenting a series of input/output pattern sets to the network. The network gradually learns the input/output relationship by adjusting the weights to minimise the error between the actual and predicted output patterns of the training sets. Cross-validation using the testing data set is adopted as the stopping criteria for the ANN model to avoid overfitting. The validation data sets are then used to assess the accuracy of the developed ANN model using data that are not used for either training or testing.

The network architecture is determined using a trial-and-error approach in which the ANN models are trained with different numbers of hidden layers and nodes. In this study, networks are developed

Table III. Input and output statistic for artificial neural network models.

Model variables and data sets	Statistical parameters				
	Mean	Standard deviation	Minimum	Maximum	Range
Slope angle (β)					
Training set	25.75	11.75	14.00	45.00	31.00
Testing set	25.97	11.98	14.00	45.00	31.00
Validation set	26.73	12.08	14.00	45.00	31.00
Depth factor (D)					
Training set	1.98	0.82	1.00	3.00	2.00
Testing set	2.08	0.80	1.00	3.00	2.00
Validation set	1.95	0.81	1.00	3.00	2.00
Stability number (N_s)					
Training set	0.29	0.14	0.10	0.50	0.40
Testing set	0.31	0.14	0.10	0.50	0.40
Validation set	0.31	0.14	0.10	0.50	0.40
Coefficient of variation (COV)					
Training set	0.48	0.34	0.10	1.00	0.90
Testing set	0.46	0.32	0.10	1.00	0.90
Validation set	0.49	0.34	0.10	1.00	0.90
Dimensionless scale of fluctuation (θ/H)					
Training set	2.87	3.69	0.10	10.00	9.90
Testing set	2.76	3.59	0.10	10.00	9.90
Validation set	2.65	3.53	0.10	10.00	9.90
Probability of failure (p_f)					
Training set	0.301	0.413	0.000	1.000	1.000
Testing set	0.268	0.394	0.000	1.000	1.000
Validation set	0.276	0.397	0.000	1.000	1.000

DOI: 10.1002/nag

with one and two hidden layers, and 2, 4, 6, 8, 10 and 12 nodes in each layer. It should be noted that an equal number of nodes is used in each hidden layer for the two hidden layers models. The *tanh* and *sigmoid* transfer functions are used in the hidden and output layers, respectively. Other internal parameters such as the learning rate and momentum term are fixed at the software's default values, that is, learning rate = 0.2 and momentum term = 0.8. The effect of varying the learning rate and momentum term on the performance of ANN models is also investigated.

The performance of the developed ANN models with different numbers of hidden layers and nodes is summarized in Table IV. The ANN models developed in this study are designated by the names 'A1' to 'A12'. The performance of the ANN models is measured by three standard performance measures: the correlation coefficient (r), the root mean square error (RMSE) and the mean absolute error (MAE). It should be noted that models that perform better are ones with a higher value of r and a lower value of RMSE and MAE. It can be seen from Table IV that models with two hidden layers generally perform better than the models with one hidden layer, except for Model A7, which has only two nodes in each hidden layer. It is also observed that models with a larger number of hidden layer nodes perform better than the models with a lesser number of nodes in the hidden layers.

The effect of the number of hidden layers and nodes on the performance of ANN models for the validation data is plotted in Figure 4. It can be seen that the performance of the ANN models improves significantly when the number of hidden layer nodes increases from two to six. However, the improvement is minimal when the hidden layer nodes further increases from 6 to 12. Figure 4 also shows that the ANN models with two hidden layers perform better than the models with only one hidden layer when there are six or more nodes in the hidden layer. Increasing the number of hidden layers has less impact for models with two and four nodes in the hidden layer.

Based on the results shown in Table IV, the optimum ANN model can be determined. According to Shahin and Jaksa [21], a model is deemed to be optimal if it combines three attributes: (i) the model provides good performance with respect to the testing set; (ii) the model has a minimum number of hidden layers and nodes; and (iii) the model has consistent performance on the validation set with that obtained on the training and testing sets. Based on the computed values of r, RMSE and MAE of the developed ANN models, Model A12 appears to be the optimum model. However, Model A12 has two hidden layers with 12 nodes in each layer, which become impractical to transform into formulae. This is because a total of 25 equations are required to describe the relationship between the input and output variables. Therefore, for the purpose of developing a set of practical equations, Model A3 is considered to be optimal. Model A3 has one hidden layer with six nodes, which requires only eight equations to describe the relationship between the input and output variables. It should be noted that the relative error between the Model A3 and A12 for the correlation coefficient

Table IV. Performance of artificial neural network models with different number of hidden layers and nodes.

Model no.	No. of hidden layers	No. of hidden nodes	Performance measures								
			r			RMSE			MAE		
			T	S	V	T	S	V	T	S	V
A1	1	2	0.971	0.970	0.972	0.100	0.097	0.094	0.051	0.049	0.051
A2	1	4	0.988	0.982	0.982	0.064	0.074	0.075	0.028	0.033	0.034
A3	1	6	0.992	0.987	0.988	0.053	0.064	0.062	0.024	0.029	0.030
A4	1	8	0.994	0.988	0.989	0.046	0.061	0.059	0.021	0.027	0.026
A5	1	10	0.996	0.991	0.990	0.036	0.052	0.056	0.015	0.021	0.024
A6	1	12	0.996	0.991	0.990	0.038	0.054	0.058	0.015	0.023	0.024
A7	2	2	0.969	0.961	0.967	0.103	0.111	0.102	0.054	0.057	0.054
A8	2	4	0.990	0.983	0.983	0.059	0.073	0.073	0.029	0.034	0.036
A9	2	6	0.999	0.994	0.995	0.021	0.045	0.038	0.009	0.017	0.016
A10	2	8	0.998	0.998	0.996	0.024	0.028	0.034	0.010	0.013	0.014
A11	2	10	0.998	0.995	0.996	0.025	0.040	0.037	0.010	0.017	0.016
A12	2	12	0.999	0.996	0.996	0.022	0.035	0.035	0.009	0.015	0.014

T = training; S = testing and V = validation.
RMSE, root mean square error; MAE, mean absolute error.

DOI: 10.1002/nag

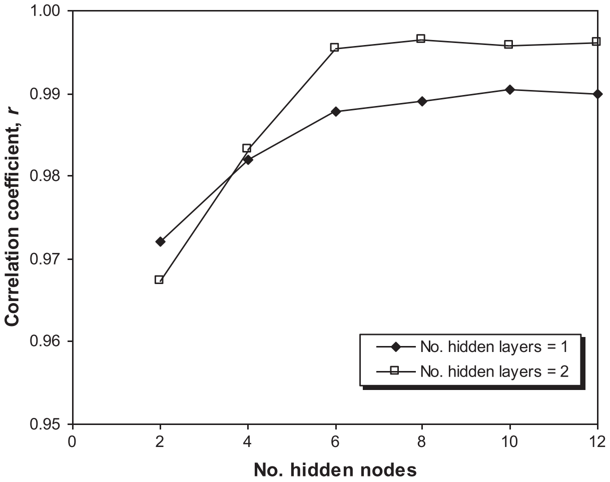

Figure 4. Effect of the numbers of hidden layers nodes on correlation coefficient (r) for the validation data set of artificial neural network models A1–A12 (learning rate = 0.2; momentum term = 0.8).

(r) is less than 1%, which is negligible. The details of transforming Model A3 into a set of practical equations for predicting the probability of failure of a heterogenous cohesive slope is described later.

The effect of the learning rate and momentum term on the performance of the ANN models is investigated by varying these two parameters within the range of 0.01–0.95 (i.e. 0.01, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9 and 0.95). The number of hidden layers and nodes are fixed at 1 and 6, respectively, which has the same network architecture to that of Model A3, as previously shown in Table IV. Figures 5 and 6 show the effect of varying the learning rate and momentum term on the correlation coefficient (r) for the validation data set, respectively. It can be seen from both Figures 5 and 6 that the effect of varying the learning rate and momentum term on model performance is not as significant as varying the number of hidden layers and nodes. It is found that the optimum learning rate = 0.2 and momentum term = 0.8, which are the software's default values.

5. DEVELOPMENT OF ARTIFICIAL NEURAL NETWORK MODEL EQUATIONS

One of the aims of this study is to develop a simple input/output relationship that can be used for predicting the reliability of a heterogeneous cohesive slope. ANNs have been successfully used to develop

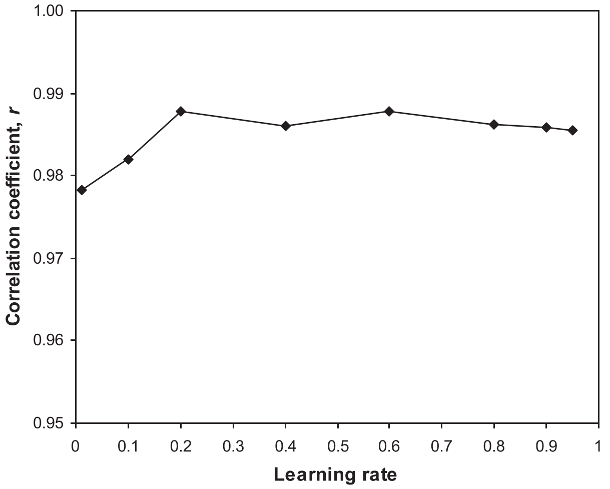

Figure 5. Effect of the learning rate on correlation coefficient (r) for the validation data set of the artificial neural network model A3 (momentum term = 0.8).

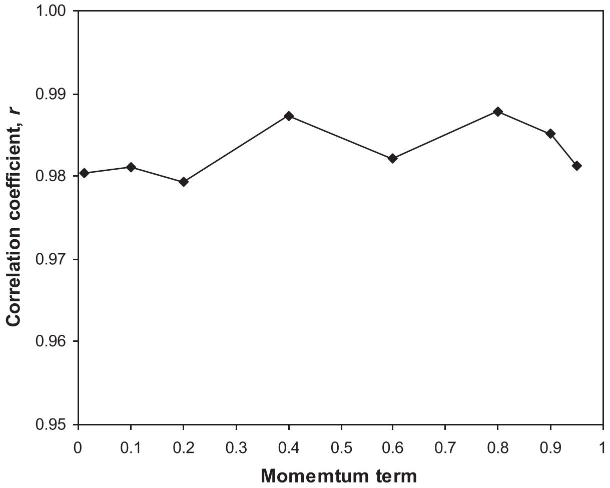

Figure 6. Effect of the momentum term on correlation coefficient (r) for the validation data set of the artificial neural network model A3 (learning rate = 0.2).

mathematical equations for predicting complex behaviour of geotechnical problems (e.g. [17, 21, 22]). More recently, another modelling procedure called evolutionary polynomial regression has also been demonstrated that is capable of developing mathematical expressions for reproducing the performance and behaviour of complicated systems (e.g. [23–25]). These studies have demonstrated that mathematical equations developed using ANNs and evolutionary polynomial regression generally performed better than the available empirical formulae.

It has been determined that Model A12, which has two hidden layers with 12 hidden nodes in each hidden layer, is the most accurate ANN model. However, as discussed previously, translating this ANN model into simple equations becomes impractical because of the large number of hidden layers and nodes. The use of this ANN model for predicting the probability of failure (p_f) of a cohesive slope is therefore limited to those users who have access to neural network software, such as *NEUFRAME*.

For the purpose of developing a relatively simple equation that can be used for predicting the probability of failure (p_f) of a heterogeneous cohesive slope, Model A3 is considered suitable. Model A3 has one hidden layer with six nodes and performs reasonably well, although not as well as Model A12. However, considering the large uncertainties in estimating soil properties, the difference in accuracy between Models A3 and A12 is considered acceptable. The structure of Model A3 is shown in Figure 7, and its connection weights and biases are shown in Table V.

Using the connection weights and biases shown in Table V, and with the input and output variables being rescaled using Eqn (7), the predicted probability of failure can be expressed as follows:

$$p_f = \frac{1}{1 + e^A} \tag{8}$$

where:

$$\begin{aligned} A = {} & 0.929 + 5.343 \tanh H_1 + 5.905 \tanh H_2 + 4.327 \tanh H_3 + 2.126 \tanh H_4 \\ & - 3.227 \tanh H_5 - 3.120 \tanh H_6 \end{aligned} \tag{9}$$

$$H_1 = 0.175 - 1.386\beta + 0.936D + 6.08N_s + 3.14COV + 1.006(\theta/H) \tag{10}$$

$$H_2 = -0.943 + 0.217\beta + 0.227D + 5.689N_s - 4.667COV - 2.092(\theta/H) \tag{11}$$

$$H_3 = 0.642 - 2.937\beta - 0.719D + 3.673N_s - 2.058COV - 1.68(\theta/H)| \tag{12}$$

$$H_4 = 0.846 - 0.682\beta - 3.259D + 0.573N_s - 0.916COV - 0.887(\theta/H) \tag{13}$$

$$H_5 = -0.099 + 1.136\beta + 1.761D - 4.868N_s + 0.416COV - 5.032(\theta/H) \tag{14}$$

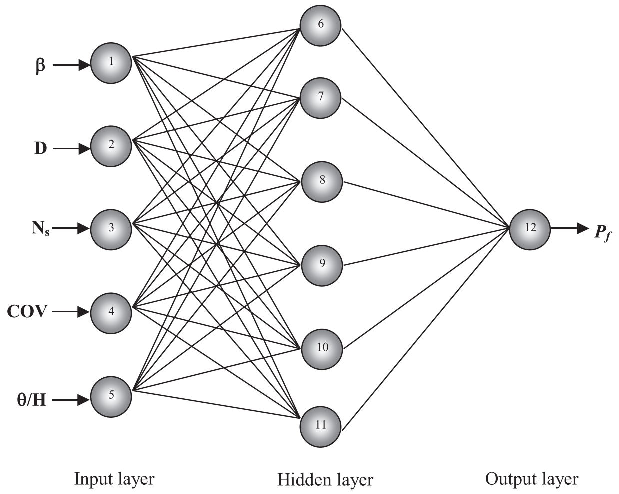

Figure 7. Structure of the artificial neural network model A3. COV, coefficient of variation.

$$H_6 = 0.152 + 0.06\beta + 0.56D – 4.715N_s + 0.177COV – 0.792(\theta/H) \quad (15)$$

It should be noted that the predicted probability of failure (p_f) obtained from Eqn (8) is scaled between 0.0 and 1.0, and in order to obtain the actual value, this p_f needs to be re-scaled using Eqn (7) and the data ranges in Table II. However, because the data range for p_f is also between 0.0 and 1.0, Eqn (8) remains unchanged. It should also be noted that, before using Eqns (10) to (15), all input variables (i.e. β, D, N_s, COV and θ/H) need to be scaled between 0.0 and 1.0 using Eqn (7) and the data ranges in Table II. Accounting for this, Eqns (10) to (15) are hence rewritten as follows:

$$H_1 = 1.494 – 0.045\beta + 0.468D + 15.2N_s + 3.489COV + 0.102(\theta/H) \quad (16)$$

$$H_2 = -2.037 + 0.007\beta + 0.114D + 14.223N_s – 5.185COV – 0.211(\theta/H) \quad (17)$$

$$H_3 = 1.656 – 0.095\beta – 0.36D + 9.183N_s – 2.287COV – 0.17(\theta/H) \quad (18)$$

$$H_4 = 2.752 – 0.022\beta – 1.63D + 1.433N_s – 1.018COV – 0.09(\theta/H) \quad (19)$$

Table V. Weights and biases for the artificial neural network Model A3.

Hidden layer nodes	w_{ji} (weight from node i in the input layer to node j in the hidden layer)						Hidden layer bias(θ_j)
	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$		
$j=6$	−1.386	0.936	6.080	3.140	1.006		0.175
$j=7$	0.217	0.227	5.689	−4.667	−2.092		−0.943
$j=8$	−2.937	−0.719	3.673	−2.058	−1.680		0.642
$j=9$	−0.682	−3.259	0.573	−0.916	−0.887		0.846
$j=10$	1.136	1.761	−4.868	0.416	−5.032		−0.099
$j=11$	0.060	0.560	−4.715	0.177	−0.792		0.152
Output layer nodes	w_{ji} (weight from node i in the input layer to node j in the hidden layer)						Output layer bias(θ_j)
	$i=6$	$i=7$	$i=8$	$i=9$	$i=10$	$i=11$	
$j=12$	−5.343	−5.905	−4.3266	−2.126	3.227	3.120	−0.929

DOI: 10.1002/nag

$$H_5 = -0.271 + 0.037\beta + 0.881D - 12.17N_s + 0.462COV - 0.508(\theta/H) \quad (20)$$

$$H_6 = 1.012 + 0.002\beta + 0.28D - 11.788N_s + 0.197COV - 0.08(\theta/H) \quad (21)$$

It should be noted that Eqn (8) is valid only for the ranges of values of β, D, N_s, COV and θ/H previously given in Table II. This is because ANN models perform best for interpolation rather than extrapolation [26, 27]. The developed ANN equations, that is, Eqns (8) and (16)–(21), can therefore be used as an alternative method to the RFEM for predicting the probability of failure of a heterogeneous cohesive slope.

To assess the accuracy of the developed equations, five predictions are performed, and the predicted probability of failure (p_f) is compared with that predicted by ANN Model A3 as summarized in Table VI. It is observed that the predictions performed by the developed equations are very close to those obtained from ANN Model A3. Further predictions are also performed using all the 288 validation data, and the correlation coefficient (r) between the predictions obtained from the developed equations and ANN Model A3 is 0.992, which further confirms that the developed equations provide almost identical predictions to those obtained from the ANN model.

6. SENSITIVITY ANALYSES OF THE ARTIFICIAL NEURAL NETWORK MODEL INPUTS

In an attempt to determine the relative importance of the various input variables, a sensitivity analysis was carried out on the ANN Model A3. This is achieved by using the method proposed by Garson [28], which has been previously adopted by other researchers (e.g. [17, 29]) for the same purpose. This method involves partitioning the hidden-output connection weights of each hidden node into components associated with each input node. As mentioned previously, Model A3 incorporates five input nodes, one hidden layer with six nodes and one output node. The connection weights were previously shown in Table V and they are rewritten as follows:

Hidden nodes	β	D	N_s	COV	θ/H	p_f
1	−1.386	0.936	6.080	3.140	1.006	−5.343
2	0.217	0.227	5.689	−4.667	−2.092	−5.905
3	−2.937	−0.719	3.673	−2.058	−1.680	−4.327
4	−0.682	−3.259	0.573	−0.916	−0.887	−2.126
5	1.136	1.761	−4.868	0.416	−5.032	−3.227
6	0.060	0.560	−4.715	0.177	−0.792	−3.120

The computation process proposed by Garson [28] is as follows:

1. For each hidden node i, obtain the products P_{ij} (where j represents the column number of the weights mentioned earlier) by multiplying the absolute value of the hidden-output layer connection weight by the absolute value of the hidden-input layer connection weight of each input variable j. As an example: $P_{11} = 1.386 \times 5.343 = 7.405$.

Hidden nodes	β	D	N_s	COV	θ/H
1	7.405	5.001	32.485	16.777	5.375
2	1.281	1.340	33.594	27.559	12.353
3	12.708	3.111	15.893	8.905	7.269
4	1.450	6.929	1.218	1.947	1.886
5	3.666	5.683	15.709	1.342	16.238
6	0.187	1.747	14.711	0.552	2.471

DOI: 10.1002/nag

2. For each hidden node, divide P_{ij} by the sum of all the input variables to obtain Q_{ij}. As an example: $Q_{11} = 7.405/(7.405 + 5.001 + 32.485 + 16.777 + 5.375) = 0.110$.

Hidden nodes	β	D	N_s	COV	θ/H
1	0.110	0.075	0.485	0.250	0.080
2	0.017	0.018	0.441	0.362	0.162
3	0.265	0.065	0.332	0.186	0.152
4	0.108	0.516	0.091	0.145	0.140
5	0.086	0.133	0.368	0.031	0.381
6	0.010	0.089	0.748	0.028	0.126

3. For each input node, sum Q_{ij} to obtain S_j. As an example:

$$S_1 = 0.110 + 0.017 + 0.265 + 0.108 + 0.086 + 0.010 = 0.596.$$

	β	D	N_s	COV	θ/H
Sum	0.596	0.895	2.465	1.003	1.041

4. Divide S_j by the sum of all the input variables to obtain the relative importance of each output weight attributed to the selected input variable. As an example, the relative importance of input node 1 is equal to:

$$(0.596 \times 100) / (0.596 + 0.895 + 2.465 + 1.003 + 1.041) = 9.9\%$$

	β	D	N_s	COV	θ/H
Relative importance (%)	9.9	14.9	41.1	16.7	17.4

The results indicate that the stability number (N_s) has the most significant effect on the predicted probability of failure (p_f) with a relative importance of 41.1%. This is followed by the normalised scale of fluctuation (θ/H) and the COV with a relative importance of 17.4% and 16.7%, respectively. The slope geometry parameters: depth factor (D) and slope angle (β) have relatively small influence on p_f, with a relative importance of 14.9% and 9.9%, respectively. It should be noted that N_s is the mean undrained shear strength (c_u) normalised by the unit weight and slope height. Furthermore, θ/H and COV are parameters used to model the spatial variability of c_u. Therefore, the sensitivity analysis suggests that spatial variability of soil properties has a more significant impact on the reliability of a heterogeneous slope than the slope geometry parameters for the geometries examined in the present analysis.

Table VI. Comparison of predictions obtained from the ANN model and the developed equations.

Case	β	D	N_s	COV	θ/H	p_f	
						ANN model	Developed equations
1	14	3	0.2	0.1	0.2	0.0000	0.0000
2	18.4	3	0.3	0.5	5	0.0131	0.0132
3	26.6	1	0.3	1	10	0.2747	0.2988
4	45	2	0.2	1	5	0.7281	0.7324
5	45	3	0.1	0.1	0.1	1.0000	0.9997

ANN, artificial neural network.

DOI: 10.1002/nag

7. SUMMARY AND CONCLUSIONS

This paper has investigated the feasibility of using ANNs as a meta-model of the RFEM and Monte Carlo simulation for predicting the probability of failure (p_f) of a heterogeneous cohesive slope. MLPs trained with the back-propagation algorithm were used in this study. The data used for the ANN model development and validation were obtained from parametric studies conducted using the RFEM and Monte Carlo simulation. A total of 1440 case records were used, and they were divided into three separate data sets: training, testing and validation. The training and testing sets were used for model development or calibration, while the validation set was used solely for model validation.

The optimum ANN model was determined using a trial-and-error approach where the number of hidden layers and nodes was varied. The performance of the developed ANN models was measured by three standard performance measures: correlation coefficient (r), RMSE and the MAE. It was found that the number of hidden layers and nodes has a significant effect on the performance of the ANN models. The ANN model with two hidden layers and 12 nodes in each hidden layer was found to be the most accurate model with an r of 0.996 for the validation data set. It was also found that the internal network parameters, such as the learning rate and momentum term, have less impact on ANN model performance when compared with the number of hidden layers and nodes.

A simple equation was developed based on the ANN model with one hidden layer and six nodes, which can be used to predict the probability of failure (p_f) of a heterogeneous cohesive slope. This equation can be used as an alternative meta-model to the more advanced but computationally intensive approach of the RFEM. It is worthwhile noting that predictions from ANN models are more reliable when used for ranges of input variables similar to those utilized in model training. This is because ANNs perform best when interpolating rather than in extrapolation. The sensitivity analysis of the ANN model inputs indicated the stability number (N_s) was the most important parameter with a relative importance of 41.1%. The other parameters ranked in the order of most to least important were θ/H, COV, D and β. Finally, for the geometries examined in the present analysis, it has been observed that the spatial variability of the soil properties has a more profound effect on the reliability of a heterogeneous slope than the slope geometry parameters.

REFERENCES

1. Griffiths DV, Fenton GA. Influence of soil strength spatial variability on the stability of an undrained clay slope by finite elements. In *Slope Stability. 2000*, GSP No. 101, Proc Geo-Denver 2000, Griffiths DV, Fenton GA, Martin TR (eds.). ASCE: Denver, 2000; 184–93.
2. Griffiths DV, Fenton GA. Probabilistic slope stability analysis by finite elements. *Journal Geotechnical and Geoenvironmental Engineering ASCE* 2004; **130**(5):507–18.
3. Shahin MA, Jaksa MB, Maier HR. Artificial neural network applications in geotechnical engineering. *Australian Geomechnics* 2001; **36**(1):49–62.
4. Shahin MA, Maier HR, Jaksa MB. State of the art of artificial neural networks in geotechnical engineering. *Electronic Journal of Geotechnical Engineering* 2008; **8**:1–26.
5. Shahin MA, Maier HR, Jaksa MB. Recent advances and future challenges for artificial neural systems in geotechnical engineering applications. *Advances in Artificial Neural Systems* 2009. doi:10.1155/2009/308239.
6. Rumelhart DE, Hilton GE, Williams RJ. Learning internal representation by error propagation. In *Parallel Distributed Processing*, Rumerlhart DE, McClelland JL (eds.). MIT Press: Cambridge, 1986; 318–362.
7. Eberhart RC, Dobbins RW. *Neural Network PC Tools: A Practical Guide*. Academic Press: San Diego, 1990.
8. Zurada JM. *Introduction to Artificial Neural Systems*. West Publishing Company: St. Paul, 1992.
9. Fausett LV. *Fundamentals Neural Networks: Architecture, Algorithms, and Applications*. Prentice-Hall: New Jersey, 1994.
10. Hassoun MH. *Fundamentals of Artificial Neural Networks*. MIT Press: Cambridge, 1995.
11. Stone M. Cross-validatory choice and assessment of statistical predictions. *Journal of Royal Statistical Society* 1974; **B36**:111–47.
12. Smith M. *Neural Networks for Statistical Modelling*. Van Nostrand: New York, 1993.
13. Taylor DW. Stability of earth slopes. *Journal of Boston Society of Civil Engineering* 1937; **24**:197–246.
14. Fenton GA, Vanmarcke EH. Simulation of random fields via local average subdivision. *Journal of Engineering Mechanics, ASCE* 1990; **116**(8):1733–49.
15. Smith IM, Griffiths DV. *Programming the Finite Element Method* (4th). John Wiley and Sons: New York, 2004.

16. Griffiths DV, Lane PA. Slope stability analysis by finite elements. *Geotechnique* 1999; **49**(3):387–403.
17. Shahin MA, Maier HR, Jaksa MB. Predicting settlements of shallow foundation using artificial neural networks. *Journal Geotechnical and Geoenvironmental Engineering ASCE* 2002; **128**(9):785–93.
18. Neusciences. NEUFRAME Version 4. Southampton: Neusciences Corp; 2000.
19. Masters T. *Practical Neural Network Recipes in C++*. Academic Press: San Diego, 1993.
20. Shahin MA, Maier HR, Jaksa MB. Data division for developing neural networks applied to geotechnical engineering. *Journal of Computing in Civil Engineering, ASCE* 2004; **18**(2):105–14.
21. Shahin MA, Jaksa MB. Neural network prediction of pullout capacity of marquee ground anchors. *Computers and Geotechnics* 2005; **32**:153–63.
22. Kuo YL, Jaksa MB, Lyamin LV, Kaggwa WS. ANN-based model for predicting the bearing capacity of strip footing on multi-layered cohesive soil. *Computers and Geotechnics* 2009; **36**:503–16.
23. Ahangar-Asr A, Javadi AA, Khalili N. A new approach to thermo-mechanical modelling of the behaviour of unsaturated soils. *International Journal for Numerical and Analytical Methods in Geomechanics* 2015; **39**:539–57.
24. Ahangar-Asr A, Javadi AA, Johari A, Chen Y. Lateral load bearing capacity modelling of piles in cohesive soils in undrained conditions; an intelligent evolutionary approach. *Applied Soft Computing; an international journal* 2014; **24**:822–28.
25. Ahangar-Asr A, Faramarzi A, Javadi AA. A new approach for prediction of the stability of soil and rock slopes. *Engineering Computations: International Journal for Computer-Aided Engineering* 2010; **27**(7):878–93.
26. Flood I, Kartam N. Neural networks in civil engineering I: principles and understanding. *Journal of Computing in Civil Engineering, ASCE* 1994; **8**(2):131–48.
27. Minns AW, Hall MJ. Artificial neural networks as rainfall-runoff models. *Hydrological Science Journal* 1996; **41**(3):399–417.
28. Garson GD. Interpreting neural-network connection weights. *AI Expert* 1991; **6**(7):47–51.
29. Goh ATC. Back-propagation neural networks for modeling complex systems. *Artificial Intelligence Engineering* 1995; **9**:143–51.

DOI: 10.1002/nag