Multivariate adaptive regression splines
for analysis of geotechnical engineering systems

W.G. Zhang, A.T.C. Goh

School of Civil & Environmental Engineering, Nanyang Technological University

Block N1, Nanyang Avenue, Singapore 639798, Singapore

Corresponding author. Tel.:+65 6790-5271

E-mail address: ctcgoh@ntu.edu.sg

Abstract

With the rapid increases in processing speed and memory of low-cost computers, it is not surprising that various advanced computational learning tools such as neural networks have been increasingly used for analyzing or modeling highly nonlinear multivariate engineering problems. These algorithms are useful for analyzing many geotechnical problems, particularly those that lack a precise analytical theory or understanding of the phenomena involved. In situations where measured or numerical data are available, neural networks have been shown to offer great promise for mapping the nonlinear interactions (dependency) between the system's inputs and outputs. Unlike most computational tools, in neural networks no predefined mathematical relationship between the dependent and independent variables is required. However, neural networks have been criticized for its long training process since the optimal configuration is not known a priori. This paper explores the use of a fairly simple nonparametric regression algorithm known as multivariate adaptive regression splines (MARS) which has the ability to approximate the relationship between the inputs and outputs, and express the relationship mathematically. The main advantages of MARS are its capacity to produce simple, easy-to-interpret models, its ability to estimate the contributions of the input variables, and its computational efficiency. First the MARS algorithm is described. A number of examples are then presented that explore the generalization capabilities and accuracy of this approach in comparison to the back-propagation neural network algorithm.

Keywords: multivariate adaptive regression splines; geotechnical system; nonlinearity; basis function; multivariate problem; neural networks.

1. Introduction

Many geotechnical engineering problems rely on the use of empirical methods expressed in the form of equations or design charts, to determine the response of the system to input variables. This is usually because of an inadequate understanding of the physical phenomena involved in the multivariate problem, or the system is too complex to be described mathematically. A typical example is the determination of the undrained frictional resistance of piles in clay. Based on field load test data, empirical methods have been proposed in which the adhesion is related to the undrained shear strength as well as other factors such as the pile length by an empirical coefficient.

For problems involving several design (input) variables and nonlinear responses, particularly with statistically dependent input variables, regression methods are usually adopted. However, regression models become computationally impractical for problems involving a large number of design variables, particularly when mixed or statistically dependent variables are involved. Another criticism of regression methods lies in their strong model assumptions.

An alternative soft computing technique is the use of artificial neural networks (ANN). An ANN has a parallel-distributed architecture with a number of interconnected nodes, commonly referred to as neurons. The neurons interact with each other via weighted connections. Each neuron is connected to all the neurons in the next layer. By far the most commonly used ANN model is known as the back-propagation (BP) algorithm [1]. In the BP algorithm, the ANN “learns” the complicated model relationship from examples of input and output patterns through modifying the connection weights to reduce the errors between the actual output values and the target output values. This is carried out by minimizing the defined error function (e.g., sum squared error) using the gradient descent approach. Validation of neural network performance is carried out by “testing” with a separate set of data that was never used in training process, to assess the generalization capability of the trained neural network model to produce the correct input-output mapping.

Generalization is influenced by factors such as the size of the training data, how representative the data is of the problem to be considered, and the physical complexity of the problem. Finding the optimal BP architecture is also important. The BP algorithm has been criticized for its computational inefficiency i.e. long process to determine the optimal network configuration since this is not known a priori. Too few hidden neurons may mean that the network is unable to model the nonlinear problem correctly. An excessive number of neurons will result in unnecessary arithmetic calculations and high computation cost and may cause a phenomenon called “overfitting”, in which the network learns insignificant aspects of the training set i.e. the intrinsic noise in the data. Determining the optimal number of hidden neurons is commonly carried out by a trial-and-error approach through repeatedly increasing the number of hidden neurons till no further improvement in the network performance is obtained. Aside from finding the optimal number of hidden neurons and the number of hidden layers, finding the optimal BP architecture is a difficult task that also involves determining the optimal transfer function and learning rate, as well as the maximum number of training cycles (epochs), all of which require considerable computational effort. Various self-pruning NN algorithms have also been proposed, for example initially starting with a network that is a purposely overfit model, and then trimming it down to the appropriate size. However, neural networks implemented with these algorithms are generally just as computationally intensive since retraining is required each time a hidden neuron or weighted connection is removed.

As highlighted by Shahin et al. [2], ANN has been successfully applied to a number of geotechnical engineering problems including pile capacity, settlement of foundations, soil properties and behavior, liquefaction, site characterization, earth retaining structures, dams, blasting and mining, slope stability, geoenvironmental engineering, rock mechanics, tunneling and underground caverns.

This paper explores the use of another promising procedure known as multivariate adaptive regression spline (MARS) [3] to model nonlinear and multidimensional relationships. As with neural networks, no prior knowledge of the form of the function is required in MARS. The main advantages of MARS are its capacity to find the complex data mapping in high-dimensional data and produce simple, easy-to-interpret models, and its ability to estimate the contributions of the input variables. Previous applications of MARS algorithm in civil engineering include modeling doweled pavement performance, predicting shaft resistance of piles in sand, estimating deformation of asphalt mixtures, analyzing shaking table tests of reinforced soil wall, and determining the undrained shear strength of clay [4-9]. In this paper, a number of examples are presented to demonstrate the function approximating capacity of MARS and its efficiency in a noisy data environment. In addition, comparative performance of the predictions between BP and MARS were carried out for six practical examples in geotechnical engineering.

2. Details of MARS

MARS is a nonlinear and nonparametric regression method that models the nonlinear responses between the inputs and the output of a system by a series of piecewise linear segments (splines) of differing gradients. No specific assumption about the underlying functional relationship between the input variables and the output is required. The end points of the segments are called knots. A knot marks the end of one region of data and the beginning of another. The resulting piecewise curves (known as basis functions), give greater flexibility to the model, allowing for bends, thresholds, and other departures from linear functions.

MARS generates basis functions by searching in a stepwise manner. An adaptive regression algorithm is used for selecting the knot locations. MARS models are constructed in a two-phase procedure. The forward phase adds functions and finds potential knots to improve the performance, resulting in an overfit model. The backward phase involves pruning the least effective terms. An open source code on MARS from Jekabsons [10] is used in carrying out the analyses presented in this paper.

Let $\boldsymbol{y}$ be the target output and $\boldsymbol{X} = (X_1, \ldots, X_P)$ be a matrix of P input variables. Then it is assumed that the data are generated from an unknown "true" model. In case of a continuous response this would be

$$\boldsymbol{y} = f(X_1, \ldots, X_P) + e = f(\boldsymbol{X}) + e \qquad (1)$$

in which e is the distribution of the error. MARS approximates the function f by applying basis functions (BFs). BFs are splines (smooth polynomials), including piecewise linear and piecewise cubic functions. For simplicity, only the piecewise linear function is expressed. Piecewise linear functions are of the form $\max(0, x-t)$ with a knot occurring at value t. The equation max(.) means that only the positive part of $(.)$ is used otherwise it is given a zero value. Formally,

$$\max(0, x-t) = \begin{cases} x-t, if\ x \geq t \\ 0, otherwise \end{cases} \qquad (2)$$

The MARS model *f(**X**)*, is constructed as a linear combination of BFs and their interactions, and is expressed as

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m \lambda_m(X) \qquad (3)$$

where each $\lambda_m(X)$ is a basis function. It can be a spline function, or the product of two or more spline functions already contained in the model (higher orders can be used when the data warrants it; for simplicity, at most second-order is assumed in this paper). The coefficients β are constants, estimated using the least-squares method.

Fig. 1 shows a simple example of how MARS would use piecewise linear spline functions to attempt to fit data. The MARS mathematical equation is expressed as

$$y = 4.4668 + 1.1038*BF1 - 3.997*BF2 + 1.967*BF3 \qquad (4)$$

where BF1 = max(0, *x* – 16), BF2 = max(0, 16 – *x*) and BF3 = max(0, 25 – *x*). The knots are located at *x* = 16 and 25. They delimit three intervals where different linear relationships are identified.

Fig. 1

The MARS modeling is a data-driven process. To fit the model in Eq. (3), first a forward selection procedure is performed on the training data. A model is constructed with only the intercept, β_0, and the basis pair that produces the largest decrease in the training error is added. Considering a current model with *M* basis functions, the next pair is added to the model in the form

$$\hat{\beta}_{M+1} \lambda_m(X) \max(0, X_j - t) + \hat{\beta}_{M+2} \lambda_m(X) \max(0, t - X_j) \qquad (5)$$

with each β being estimated by the method of least squares. As a basis function is added to the model space, interactions between BFs that are already in the model are also considered. BFs are added until the model reaches some maximum specified number of terms leading to a purposely overfit model.

To reduce the number of terms, a backward deletion sequence follows. The aim of the backward deletion procedure is to find a close to optimal model by removing extraneous variables. The backward pass prunes the model by removing the basis functions with the lowest contribution to the model until it finds the best sub-model. Thus, the basis functions maintained in the final optimal model are selected from the set of all candidate basis functions, used in the forward selection step. Model subsets are compared using the less computationally expensive method of Generalized Cross-Validation (GCV). The GCV equation is a goodness of fit test that penalizes large numbers of BFs and serves to reduce the chance of overfitting. For the training data with N observations, GCV for a model is calculated as follows [11]

$$GCV = \frac{\frac{1}{N}\sum_{i=1}^{N}[y_i - f(x_i)]^2}{[1 - \frac{M + d \times (M-1)/2}{N}]^2} \qquad (6)$$

in which M is the number of BFs, d is the penalizing parameter, N is the number of observations, and $f(x_i)$ denotes the predicted values of the MARS model. The numerator is the mean squared error of the evaluated model in the training data, penalized by the denominator. The denominator accounts for the increasing variance in the case of increasing model complexity. Note that $(M-1)/2$ is the number of hinge function knots. The GCV penalizes not only the number of the model's basis functions but also the number of knots. A default value of 3 is assigned to penalizing parameter d [3]. At each deletion step a basis function is removed to minimize Eq. (3), until an adequately fitted model is found. MARS is an adaptive procedure because the selection of BFs and the variable knot locations are data-based and specific to the problem at hand.

After the optimal MARS model is determined, by grouping together all the BFs that involve one variable and another grouping of BFs that involve pairwise interactions (and even higher level interactions when applicable), the procedure known as analysis of variance (ANOVA) decomposition [3] can be used to assess the contributions from the input variables and the BFs.

As mentioned previously, the BP algorithm has been criticized for its computational inefficiency i.e. long process to determine the optimal network configuration since this is not known a priori but has to be determined through a trial-and-error approach. MARS is computationally more efficient at finding the optimal model as it essentially builds flexible models by fitting linear regressions and approximates the model by segmenting separate slopes in distinct intervals of the input variables. The variables to use and the knot locations of the intervals for each variable are determined via a fast but intensive search procedure. The forward selection and backward deletion procedure also ensures that an optimal model can be found.

3. LR_MARS

Linear Regression is commonly used statistical method for predicting values of a dependent variable from observed values of a set of predictor variables. Logistic Regression (LR) is a variation of linear regression for situations where the dependent variable is not a continuous parameter but rather a binary event (e.g., yes / no, good / bad, 0 / 1). The value predicted by LR is the probability of an event, ranging from 0 to 1. LR is more appropriate than linear regression for assessing classification problems such as the seismic liquefaction potential example presented later as it allows for binary outputs where each individual liquefaction record is classified as liquefied or non-liquefied (0 for non-liquefied case while 1 for liquefied case). Eq. (1) is applicable for the case of a continuous response of a MARS model. For a binary response, assuming P_r is the estimated probability that an individual case is liquefied, then the LR_MARS model is

$$\operatorname{logit} P_r(y=1) = f(X_1, \ldots, X_P) + \varepsilon \qquad (7)$$

in which the distribution of the error ε is an exponential. Further, Eq. (7) can be expressed as

$$\log\left(\frac{\mathrm{P_r}}{1-\mathrm{P_r}}\right) = f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m \lambda_m(X) \qquad (8)$$

or

$$e^{\log(\frac{\mathrm{P_r}}{1-\mathrm{P_r}})} = e^{f(X)} = e^{\beta_0 + \sum_{m=1}^{M} \beta_m \lambda_m(X)} \qquad (9)$$

The estimated liquefaction probability is

$$\mathrm{P_r} = \frac{1}{1+e^{-f(X)}} = \frac{1}{1+e^{-\beta_0 - \sum_{m=1}^{M} \beta_m \lambda_m(X)}} \qquad (10)$$

in which the β values are estimated using the least-squares method as in Eq. (3).

4. Neural network analysis and performance measures

In the six geotechnical examples analyzed using MARS in the next section, the same data were also analyzed using a Matlab-based back-propagation algorithm BPNN for comparative purposes. For simplicity, these BPNN models are assumed to have a single hidden layer. The optimal BPNN architecture is obtained through a trial and error procedure, by varying the number of hidden neurons and the transfer function type (logsigmoid, tansigmoid, or purelin). Table 1 shows the various performance measures used to compare the predictions of the two metahueristic methods. In addition the processing speed (CPU time) for both methods are also presented.

Table 1

For the final example (seismic liquefaction assessment) in which the dependent variable is not a continuous parameter but rather a binary event, a common measure of evaluating the performance of a pattern-classification model is to determine the success rate SR (the percentage of correctly classified cases).

5. Analyses using MARS

Nine examples are presented to illustrate the application and accuracy of MARS. Firstly, three examples consisting of fairly complicated mathematical functions (with single or two variables) are presented to demonstrate the function approximating capacity of MARS. This is followed by an example to evaluate the MARS efficiency in analyzing a hypothetical nonlinear function in which noise (error) is introduced. The last six are practical geotechnical examples that highlight the capability of MARS in modeling nonlinear multivariate problems.

5.1 Simple function approximation

In this example, MARS was used to analyze two complicated nonlinear functions consisting of a single variable:

$$y = 0.6\sin(\pi x) + 0.3\sin(3\pi x) + 0.1\sin(5\pi x) \quad (-1 < x < 1) \qquad (11)$$

$$y = e^{10x(x-1)}\sin(12\pi x) \ (0 < x < 1) \qquad (12)$$

Fig. 2a and b show the learning results of the above two functions obtained by MARS. The high coefficient of determination R^2 value indicates that MARS is highly accurate in approximating these two functions.

Fig. 2

5.2 Two-dimensional approximation

Fig. 3 shows a two-variable function (Eq. (13)), which has been widely used for model performance validation.

$$y = \sin(0.83\pi x_1)\cos(1.25\pi x_2) \quad (-1 < x_1, x_2 < 1) \qquad (13)$$

Fig. 3

To approximate this function, two MARS models with 45 BFs of linear and cubic spline functions respectively are used as shown in Fig. 4a and b. R^2 values of 0.9976 and 0.9991 show that MARS models with sufficient BFs can be used to approximate a two-dimensional function accurately.

Fig. 4

5.3 Function approximation with noise

A polynomial function $y=3x^3$ with Gaussian noise ε is used to verify the generalization capability and accuracy of MARS for the case of a rather large error variance (i.e., noisy data). The x is uniformly distributed between -1 and 1and ε is normally distributed with mean value of 0. Two cases were considered, one with a smaller error (variance of 0.25) and one with a larger error (variance of 1.0). In addition, two types of spline functions (linear spline functions and cubic spline functions with the maximum number of BFs set as 6) were used. Fig. 5 shows the scatter plots with the corresponding MARS regression curves of this example. Also displayed in the plot is the exact curve of $y = 3x^3$. The plots show that the MARS approximation almost completely overlays the exact function, yielding a very good fit to the data, even in the case of a rather large error variance.

Fig. 5

5.4 HP-Pile drivability

This HP-pile drivability example illustrates the use of MARS for analyzing a multivariate problem with a large data set. Jeon and Rahman [12] developed a BP model to predict pile drivability in terms of Maximum compressive stresses (MCS), Maximum tensile stresses (MTS), and Blow per foot (BPF). The database consisted of 4072 HP pile test results from 67 projects. Due to the large database and the variety of input variables, Joen and Rahman [12] divided the data into five categories based on the ultimate pile capacity Q_u (Q1: 133.4-355.9; Q2: 360.0-707.3; Q3: 707.4-1112.1; Q4: 1112.2-1774.8; Q5: 1774.9-3113.7; Unit: kN). For each group 70% of the observations were randomly selected for training and the remaining for testing. It should be noted that the units for MCS and MTS have been converted to MPa from the original ksi units used in [12]. Seventeen variables including hammer, hammer cushion material, pile, soil parameters, ultimate pile capacities, and stroke were used as inputs to predict the three target outputs. A summary of the input variables and outputs is listed in Table 2.

Table 2

This problem has been reanalyzed using MARS and BPNN (with single hidden layer and sigmoid transfer function). For brevity, only analyses for category Q1 type are considered in this paper. For the BPNN models, the optimum numbers of hidden neurons are 9, 8, and 7 for MCS, MTS and BPF, respectively. Table 3 lists the MARS models to predict MCS, MTS and BPF, using 42, 43 and 40 BFs, respectively.

Table 3

Fig. 6 through to Fig. 8 show the BPNN and MARS predictions for MCS, MTS and BPF for the training and testing data patterns. For both methods, high R^2 are obtained.

Fig. 6

Fig. 7

Fig. 8

Comparisons of R^2, MSE and MAE in Rows 4, 5 and 6 of Table 4 indicate that BPNN gives only marginally better predictions than MARS. For testing samples of MTS and BPF, the MARS model is more accurate than BPNN. Therefore, both MARS and BPNN can serve as reliable tools for the prediction of HP-Pile drivability. Table 5 displays the ANOVA decomposition of the developed MARS models for MCS, MTS and BPF respectively. The first column in Table 5 lists the ANOVA function number. The second column gives an indication of the importance of the corresponding ANOVA function, by listing the GCV score for a model with all BFs corresponding to that particular ANOVA function removed. This GCV score can be used to evaluate whether the ANOVA function is making an important contribution to the model, or whether it just marginally improves the global GCV score. The third column provides the standard deviation of this function. The fourth column gives the number of BFs comprising the ANOVA function. The last column gives the particular input variables associated with the ANOVA function. Fig. 9 gives the plots of the relative importance of the input variables for the three MARS models, which is evaluated by the increase in the GCV value caused by removing the considered variables from the developed MARS model. It can be observed that variable 17 (Stroke) is the most important parameter in the MCS model, followed by variable 1(Hammer weight). Variable 6 (Helmet weight) and variable 8 (Penetration) are significantly important in determining MTS. Variable 1 (Hammer weight) and variable 2 (Hammer energy) are the two most important parameters for estimating BPF. Apart from the ability to estimate the contributions of the input variables, another distinct advantage of MARS lies in its convergence speed. Rows 2, 3 and 4 of Table 6 list the CPU processing time using BPNN and MARS. The advantage of the processing speed of MARS is obvious.

Table 4

Table 5

Fig. 9

Table 6

Table 7 lists the BFs of the MARS model for BPF and their corresponding equations. The interpretable MARS model to predict BPF is given by

$$\begin{aligned} \mathrm{BPF} = {} & 7.469+0.177*\mathrm{BF1}+0.059*\mathrm{BF2}+0.268*\mathrm{BF3}-0.103*\mathrm{BF4}-0.554*\mathrm{BF5}+1.024*\mathrm{BF6}-3.306*\mathrm{BF7}+2.165*\mathrm{BF8} \\ & +0.042*\mathrm{BF9}-0.032*\mathrm{BF10}+0.012*\mathrm{BF11}-0.007*\mathrm{BF12}+0.376*\mathrm{BF13}-0.002*\mathrm{BF14}-0.027*\mathrm{BF15}+16.685*\mathrm{BF16} \\ & -41.224*\mathrm{BF17}+0.376*\mathrm{BF18}-1.247*\mathrm{BF19}-1.935*\mathrm{BF20}-0.0323*\mathrm{BF21}+0.065*\mathrm{BF22}-0.165*\mathrm{BF23}-0.0004*\mathrm{BF24} \\ & +0.001*\mathrm{BF25}+0.0004*\mathrm{BF26}+0.25*\mathrm{BF27}-0.0001*\mathrm{BF28}-0.0003*\mathrm{BF29}-0.042*\mathrm{BF30}+0.139*\mathrm{BF31}-42.828*\mathrm{BF32} \\ & +0.002*\mathrm{BF33}-10.708*\mathrm{BF34}+54.887*\mathrm{BF35}+15.119*\mathrm{BF36}-0.054*\mathrm{BF37}+0.025*\mathrm{BF38}+0.019*\mathrm{BF39}+0.057*\mathrm{BF40} \end{aligned} \tag{14}$$

Table 7

5.5 Prediction of surface settlement associated with tunneling operation

Ground movements and surface settlements associated with tunnel operations are a major concern in the design of tunnels in urban areas as excessive movements can damage nearby building and utilities. Using instrumented data and data from the tunnel operational parameters and the geological parameters, various procedures including the use of empirical equations, simple equations based on the theory of elasticity or the numerical tools such as the finite element method are available for prediction of surface settlements. Using a total of 148 instrumented sections of settlement data obtained from three separate mass rapid transit projects in Singapore, Goh and Hefney [13] developed an ANN model to predict the tunnel settlement. This example has been reanalyzed using BPNN and MARS. A total of eight inputs that represented the tunnel geometry, geological conditions and the earth pressure balance (EPB) operation factors as shown in Table 8 were considered. A total of 115 observations of the settlement were randomly selected as the training data and the remaining 33 data samples were used for testing the validity of the developed neural network. The optimal BPNN model consisted of five hidden neurons.

Table 8

For the MARS model, the logarithmic values of parameters *EP*, *E*, *GP* and *St* were used as it was found that this substantially improved the MARS's training process. The tunnel settlement analysis using MARS adopted 16 BFs of linear spline functions with second-order interaction. A plot of MARS and BPNN predicted *St* values versus the measured for the training and testing samples is shown in Fig. 10. A comparison between BPNN and MARS in Row 7 of Table 4 indicates that MARS model is slightly more accurate than BPNN. Row 5 of Table 6 suggests that MARS outperforms BPNN in terms of processing speed.

Fig. 10

The ANOVA parameter relative importance assessment indicates that the two most important variables are *MC* (Mean moisture content) and *EP* (Earth pressure). For brevity, the ANOVA decomposition data has been omitted. Table 9 lists the BFs and their corresponding equations. It is noted from Table 9 that of the 16 BFs, 13 BFs with interaction terms are integrated in this model (excluding BF1, BF5 and BF10), indicating that the model is not simply additive and that interaction terms play a significantly important role. The interpretable MARS model is given by

$$\log(St) = 1.4764 - 4.0327*\mathrm{BF1} - 13.466*\mathrm{BF2} + 3.8852*\mathrm{BF3} + 0.7561*\mathrm{BF4} - 0.0619*\mathrm{BF5} - 0.3573*\mathrm{BF6} - 12.328*\mathrm{BF7} - 2.9412*\mathrm{BF8} + 0.082*\mathrm{BF9} + 0.4218*\mathrm{BF10} - 1.7684*\mathrm{BF11} - 3.8127*\mathrm{BF12} - 88.602*\mathrm{BF13} + 0.4682*\mathrm{BF14} - 0.0036*\mathrm{BF15} + 0.0256*\mathrm{BF16} \quad (15)$$

Table 9

5.6 Prediction of collapse potential for compacted soils

Collapse, which is the additional deformation of compacted soils when wetted, is responsible for considerable damage to buildings, embankments and earth dams that rest on compacted fills. The influence of various parameters on the amount of collapse has been discussed by many investigators. Collapse potential is assessed through different methods, including simple empirical equations based on statistical regression, experimental procedures such as single and double oedometer tests and neural networks. After compiling a more comprehensive database consisting of 192 oedemeter tests and 138 similar data sets available in the literature, Habibagahi and Taherian [14] employed neural networks to predict the amount of collapse. Table 10 lists the ten inputs for neural network model to predict the collapse potential.

Table 10

Using the same training and testing datasets, this problem is reanalyzed by means of BPNN (three hidden neurons) and MARS. The MARS model adopted 19 BFs of linear spline functions with second-order interaction. A plot of the BPNN and MARS predicted collapse potential values versus the actual values for the training and testing patterns are shown in Fig. 11. Comparison between BPNN and MARS in Row 8 of Table 4 indicates that the MARS model is slightly more accurate. Row 6 of Table 6 suggests that MARS also outperforms BPNN in processing time.

Fig. 11

The ANOVA parameter relative importance assessment indicates that the two most significant variables are P_w (Pressure at wetting) and C_U (Coefficient of uniformity). For brevity, the ANOVA decomposition data has been omitted. Table 11 lists the BFs and their corresponding equations. 16 of the total 19 BFs are interaction terms with second-order (excluding BF1, BF2 and BF3), indicating that the model is not simply additive and that interaction terms play a significantly important role. The interpretable MARS model is given by

$$CP\ (\%) = -0.2524 + 0.7743*BF1 + 0.2376*BF2 + 0.2131*BF3 - 0.1755*BF4 - 0.057*BF5 - 0.0184*BF6 + 0.01*BF7 + 0.0032*BF8 + 0.0206*BF9 + 0.0343*BF10 - 0.001*BF11 + 0.0843*BF12 + 0.0645*BF13 - 0.4218*BF14 - 0.1665*BF15 + 0.0007*BF16 + 0.0019*BF17 - 0.0026*BF18 - 0.7086*BF19 \quad (16)$$

Table 11

5.7 Estimating the undrained side resistance for drilled shafts

The determination of the side resistance of drilled shafts is essentially based on the total stress or alpha (α) method developed by Tomlinson [15], in which the side resistance or adhesion is related to the undrained shear strength s_u by an empirical coefficient denoted α, the adhesion factor. Subsequently, Randolph and Murphy [16], Semple and Rigden [17], Kulhawy and Jackson [18], and Chen and Kulhawy [19] demonstrated that α is influenced by the mean effective overburden stress σ'_{vm}, s_u, the effective stress friction angle $\bar{\phi}$, etc. Goh et al. [20] analyzed the undrained side resistance database from Chen and Kulhawy [19] using a hybrid Bayesian neural network. Using the database of Goh et al [20], this problem has been reanalyzed using BPNN and MARS. The database was compiled from 127 field load tests on drilled shafts in a variety of cohesive soil profiles. The dataset was separated randomly into a training set of 85 observations and a testing set of 42 observations. The BPNN structure consisted of two input neurons representing σ'_{vm} and s_u, five hidden neurons, and an output neuron representing α. The MARS model adopted 10 BFs of linear spline functions with second-order interaction. A plot of the BPNN and MARS predicted α values versus the measured values for the training and testing patterns are shown in Fig. 12. The comparison between BPNN and MARS in Row 9 of Table 4 indicates that the performance measures for the MARS and BPNN models are similar. Row 7 of Table 6 suggests that MARS outperforms BPNN in computational speed.

Fig. 12

The ANOVA parameter relative importance assessment indicates that s_u is more important than σ'_{vm} in determining α. For brevity, the ANOVA decomposition has been omitted. Table 12 lists the BFs and their corresponding equations. The interpretable MARS model is given by

$$\alpha = -0.6512 - 14.103*BF1 + 16.684*BF2 + 14.095*BF3 - 1259.6*BF4 - 85.959*BF5 + 77.076*BF6 + 1.7772*BF7 - 31.175*BF8 - 40.356*BF9 - 3.063*BF10 \quad (17)$$

Table 12

5.8 Prediction of diaphragm wall deflections in soft clays

For excavations in built-up urban areas with deep deposits of soft clays, it is essential to control ground movements to minimize damage to adjacent structures and facilities. Diaphragm walls are often used to minimize ground movements and damage to adjacent structures. The limiting wall deflection is typically taken to be a percentage of the excavation height. Many empirical methods (e.g., Mana and Clough [21]) have been proposed for estimating wall movements.

Xuan [22] carried out extensive plane strain finite element analyses to examine the excavation-induced wall deflections for a deep deposit of soft clay supported by diaphragm walls and bracing. The cross-sectional soil and wall profile are shown in Fig. 13. The major parameters influencing excavation performance and the ranges of these parameters are shown in Table 13. Because of symmetry, only half of the cross-section was considered.

Fig. 13

Table 13

Finite element analyses were carried out for a total of 1120 cases to determine the maximum diaphragm wall deflection. Of the 1120 cases, 840 observations were randomly chosen as the training patterns and the remaining as the testing patterns for the BPNN and MARS analyses. The BPNN structure consisted of seven input neurons, three hidden neurons and the output neuron representing the maximum wall deflection. The optimal MARS model consisted of 22 BFs of linear spline functions with second-order interaction. A plot of the MARS and BPNN predicted wall deflection values versus the FEM calculated values for the training and testing patterns are shown in Fig. 14. The results in in Row 10 of Table 4 indicates that the BPNN gives slightly better predictions than MARS. However, Row 8 of Table 6 suggests that MARS outperforms BPNN in computational speed.

The ANOVA parameter relative importance assessment indicates that the two variables which contribute most to the diaphragm wall deflection are h (Excavation depth) and B (Excavation width). For brevity, the ANOVA decomposition has been omitted. Table 14 lists the BFs and their corresponding equations for the optimal MARS model. The interpretable MARS model is given by

$$\delta_{h0} = 165 - 50.889*BF1 + 66.598*BF2 - 0.1914*BF3 + 0.4956*BF4 - 10.324*BF5 + 19.135*BF6 - 326.34*BF7 + 815.69*BF8 + 4.9881*BF9 - 6.1891*BF10 + 7.4897*BF11 - 7.0073*BF12 - 13.712*BF13 + 24.131*BF14 + 540.93*BF15 - 331.28*BF16 + 2.7716*BF17 - 4.5821*BF18 - 1.1808*BF19 + 0.8612*BF20 + 0.5114*BF21 - 1.5474*BF22 \quad (18)$$

Fig. 14

Table 14

5.9 Evaluating seismic liquefaction potential

Simplified techniques based on an in-situ testing measurement index are commonly used to assess seismic liquefaction potential. Most of these simplified charts or equations rely on the analysis of liquefaction case histories. Statistical methods were commonly adopted to assign probabilities of liquefaction through various statistical classification and regression analyses [23-26].

Based on the cone penetration test case records, Goh [27] adopted a Probabilistic neural network to evaluate seismic liquefaction potential. The case records represent 104 sites that liquefied and 66 sites that did not liquefy. These case records are reanalyzed using BPNN (six hidden neurons) and MARS based on Logistic Regression (LR_MARS). The inputs consisted of six neurons representing the earthquake magnitude M, the peak acceleration at the ground surface a_{max}, total vertical stress σ_v, the effective stress σ'_v, measured cone tip resistance q_c, and the mean grain size D_{50}. The data sets are divided into two parts: 114 are randomly selected for training and the remaining for testing. The training and testing results using BPNN are shown in Fig. 15.

Fig. 15

LR_MARS model adopted 6 BFs of linear spline functions with second-order interaction. The training and testing results are shown in Fig. 16. A comparison between BPNN and MARS in

Row 11 of Table 4 indicates that BPNN performs slightly better than MARS. The BPNN model has an overall success rate SR of 97.1%. The model accuracy in predicting liquefied cases is as high as 98.1%. The MARS model has an overall success rate of 90.6% and an accuracy of 95.2% in predicting liquefied cases. However, Row 9 of Table 6 suggests that MARS outperforms BPNN in computational speed and provides an easy to interpret model, which is shown in Table 15. The ANOVA parameter relative importance comparison indicates that q_c and M are the two most significant parameters in assessing liquefaction potential. For brevity, the ANOVA decomposition data has been omitted.

Fig. 16

Table 15

6. Conclusions

This paper demonstrates the viability of MARS to model nonlinear geotechnical engineering problems involving a multitude of variables, as an alternative to BPNN. The examples presented have demonstrated that in general the MARS and BPNN predictions are similar in terms of accuracy and generalization, even in a rather noisy environment. As mentioned previously, the BP algorithm has been criticized for its computational inefficiency i.e. long process to determine the optimal network configuration such as the number of hidden neurons since this is not known a priori but has to be determined through a trial-and-error approach. MARS is computationally more efficient at finding the optimal model as it essentially builds flexible models using linear regression and approximates the model by segmenting separate slopes in distinct intervals of the input variables. The variables to use and the knot locations of the intervals for each variable are determined via a fast but intensive search procedure. The forward selection and backward deletion procedure also ensures that an optimal model can be found. In addition, the resulting MARS model can be easily interpreted. Since MARS explicitly defines the intervals (boundaries) for the input variables, the model enables engineers to have an insight and understanding of where significant changes in the data may happen. Additionally, MARS is able to assess the relative importance of each variable through the ANOVA decomposition process.

Acknowledgements

The authors would like to express their appreciation to the Defense Science and Technology Agency Singapore for providing the funding for this research. The authors are most thankful to the two anonymous reviewers for their valuable comments and suggestions.

References

[1] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representation by error propagation. In Parallel Distributed Processing, Rumelhart DE, McClelland JL (eds). MIT Press: Cambridge, 1986; 1: 318–362.

[2] Shahin MA, Jaksa MB, Maier HR. State of the art of artificial neural networks in geotechnical engineering. Electronic Journal of Geotechnical Engineering 2008; 8:1–26.

[3] Friedman JH. Multivariate adaptive regression splines. The Annals of Statistics 1991; 19:1–141.

[4] Attoh-Okine NO, Cooger K, Mensah S. Multivariate Adaptive Regression (MARS) and Hinged Hyperplanes (HHP) for Doweled Pavement Performance Modeling. Journal of Construction and Building Materials 2009; 23: 3020–3023.

[5] Lashkari A. Prediction of the shaft resistance of nondisplacement piles in sand. Int. J. Numer. Anal. Meth. Geomech 2012 doi: 10.1002/nag.1129/.

[6] Mirzahosseinia M, Aghaeifarb A, Alavic A, Gandomic A, Seyednour R. Permanent deformation analysis of asphalt mixtures using soft computing techniques. Expert Systems with Applications 2011; 38 (5): 6081–6100.

[7] Zarnani S, El-Emam M, Bathurst RJ. Comparison of numerical and analytical solutions for reinforced soil wall shaking table tests. Geomechanics & Engineering 2011; 3(4): 291–321.

[8] Samui P. Determination of ultimate capacity of driven piles in cohesionless soil: A Multivariate Adaptive Regression Spline approach. Int. J. Numer. Anal. Meth. Geomech 2011 doi: 10.1002/nag.1076.

[9] Samui P, Karup P. Multivariate Adaptive Regression Spline and Least Square Support Vector Machine for Prediction of Undrained Shear Strength of Clay. Applied Metaheuristic Computing 2011; 3(2): 33–42.

[10] Jekabsons G. VariReg: A software tool for regression modelling using various modeling methods. Riga Technical University, 2010. <http://www.cs.rtu.lv/jekabsons/>

[11] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: Data mining, inference and prediction, 2nd edition, Springer, 2009.

[12] Jeon JK, Rahman MS, Fuzzy Neural Network Models for Geotechnical Problems. Research Project FHWA / NC / 2006–52. North Carolina State University, Raleigh, N.C., 2008.

[13] Goh ATC, Hefney AM. Reliability assessment of EPB tunnel-related settlement. International Journal Geomechanics and Engineering 2010; 2(1): 57–69.

[14] Habibagahi G, Taherian M. Prediction of collapse potential for compacted soils using artificial neural networks. Scientia Iranica 2004; 11: 1–20.

[15] Tomlinson MJ. Adhesion of piles driven in clay soils. Proc.,4th Int. Conf. on Soil Mechanics and Foundations Engineering, 1957; 2: 66–71, London.

[16] Randolph MF, Murphy BS. Shaft capacity of driven piles in clay. Proc., 17th Offshore Technical Conf., 1985; 1: 371–378, Houston.

[17] Semple RM, Ridgen WJ. Shaft capacity of driven pipe piles in clay. Ground Eng 1986; 19(1): 11–17.

[18] Kulhawy FH, Jackson C. Some observations of undrained side resistance of drilled shafts. Foundation Engineering: Current principles and practices 1989; 2: 1011–1025, New York.

[19] Chen YJ, Kulhawy FH. Case history evaluation of the behavior of drilled shafts under axial and lateral loading. Rep. No. TR–104601 1994, Electric Power Research Institute, Palo Alto, Calif.

[20] Goh ATC, Kulhawy FH, Chua CG. Bayesian neural network analysis of undrained side resistance of drilled shafts. Journal of Geotechnical and Geoenvironmental Engineering 2005; 131(1): 84–93.

[21] Mana AI, Clough GW. Prediction of movement for braced cuts in clay. Journal Geotechnical Engineering Division ASCE 1981; 107 (6): 759–777.

[22] Xuan F. Behavior of diaphragm walls in clays and reliability analysis. M. Eng. Thesis, 2009 Nanyang Technological University, Singapore.

[23] Liao SC, Veneziano D, Whitman RV. Regression models for evaluating liquefaction probability. J Geotech Eng ASCE 1988; 114(4): 389–411.

[24] Juang CH, Rosowsky DV, Tang WH. Reliability-based method for assessing liquefaction potential of soils. J Geotech Geoenviron Eng ASCE 1999; 125(8): 684–689.

[25] Lai SY, Hsu SC, Hsieh MJ. Discriminant model for evaluating soil liquefaction potential using cone penetration test data. J Geotech Geoenviron Eng ASCE 2004; 130(12): 1271–1282.

[26] Tosun H, Seyrek E, Orhan A, Savas H, Turkoz M. Soil liquefaction potential in Eskisehir, NW Turkey. Natural Hazards and Earth System Sciences 2011; 11: 1071–1082.

[27] Goh ATC. Probabilistic neural network for evaluating seismic liquefaction potential. Can Geotech J 2002; 39: 219–232.

Tables

Table 1 Calculation of error measures.

Table 2 Summary of HP-pile input variables and outputs.

Table 3 MARS models to predict MCS, MTS and BPF.

Table 4 Modeling accuracy comparison between BPNN and MARS.

Table 5 ANOVA decomposition of MARS models.

Table 6 Processing time comparison between BPNN and MARS (Units: s).

Table 7 Basis functions and corresponding equations of MARS model for BPF prediction.

Table 8 Summary of tunnel settlement input variables and output.

Table 9 Basis functions and corresponding equations of MARS model for settlement prediction.

Table 10 Summary of collapse potential input variables and output.

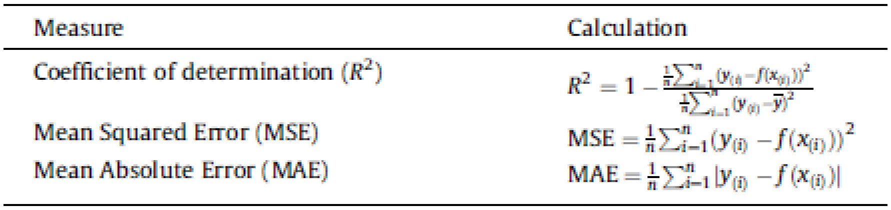
Table 11 Basis functions and corresponding equations of MARS model for *CP* prediction.

Table 12 Basis functions and corresponding equations of MARS model for α prediction.

Table 13 Summary of range of finite element parameters for braced excavation analysis.

Table 14 Basis functions and corresponding equations of MARS model for diaphragm wall deflection prediction.

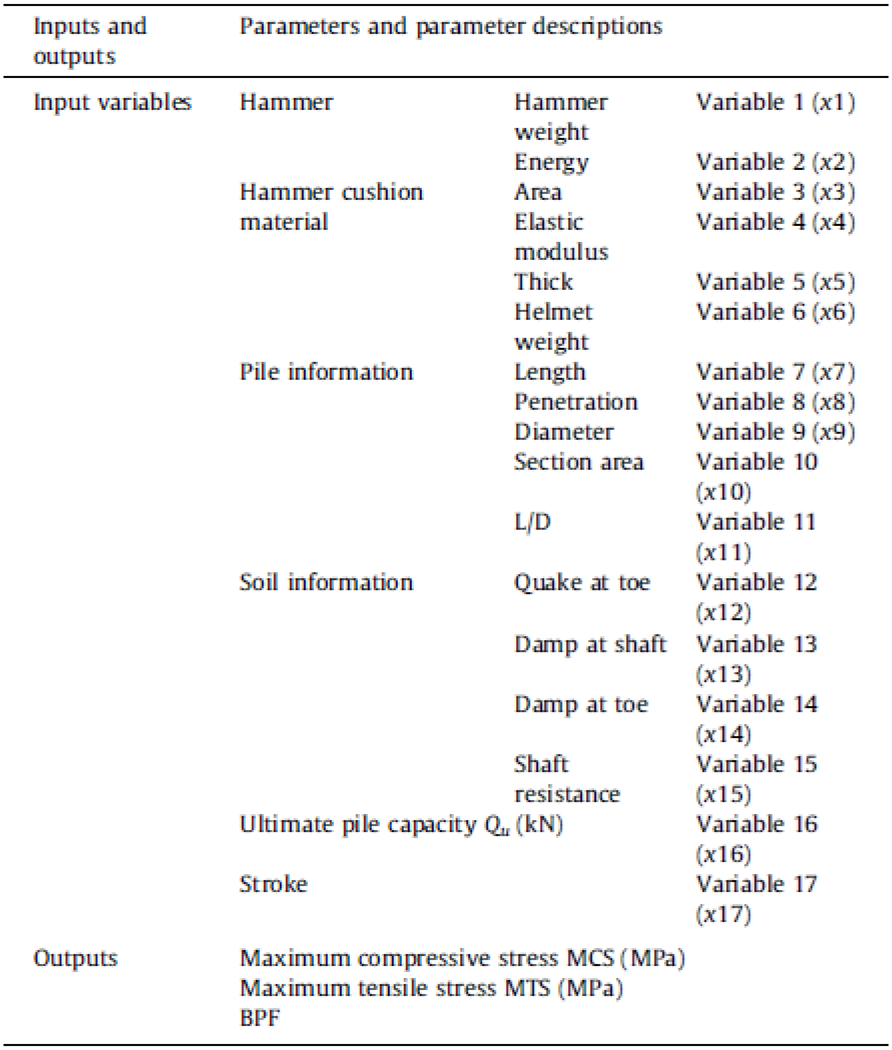
Table 15 Basis functions and corresponding equations of MARS model for seismic liquefaction potential assessment.



Measure	Calculation
Coefficient of determination (R^2)	$R^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_{(i)} - f(x_{(i)}))^2}{\frac{1}{n}\sum_{i=1}^{n}(y_{(i)} - \overline{y})^2}$
Mean Squared Error (MSE)	$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_{(i)} - f(x_{(i)}))^2$
Mean Absolute Error (MAE)	$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\lvert y_{(i)} - f(x_{(i)})\rvert$

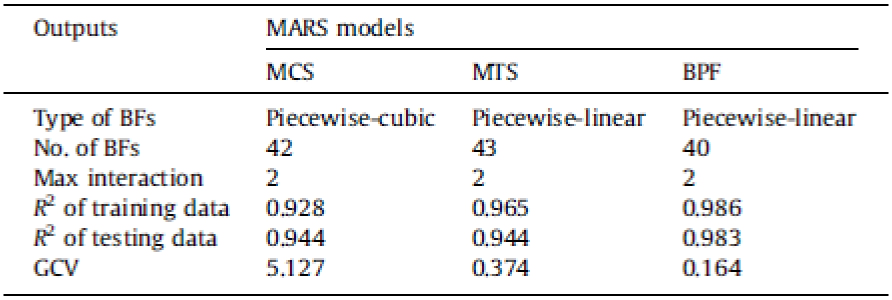
$\overline{y}$ is the mean of the target values of $y_{(i)}$; $f(x_{(i)})$ is model predictions; n denotes the number of data points in the training or testing set.

Table 1 Calculation of error measures.



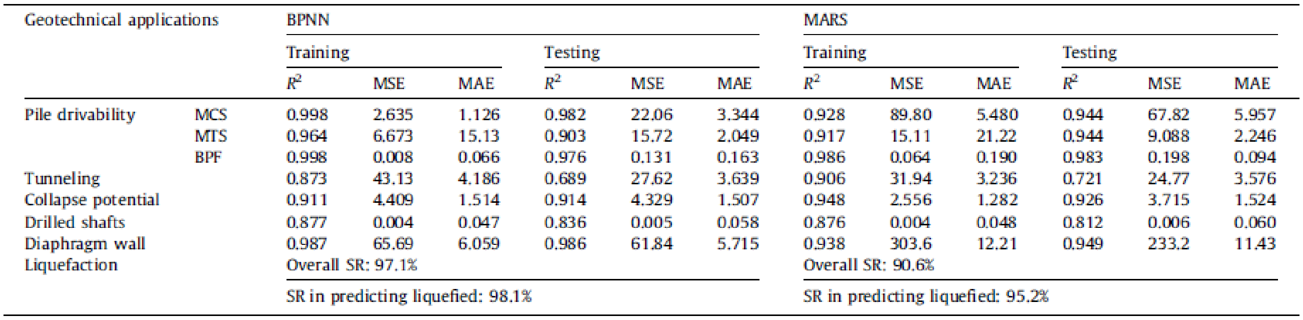
Inputs and outputs	Parameters and parameter descriptions		
Input variables	Hammer	Hammer weight	Variable 1 ($x1$)
		Energy	Variable 2 ($x2$)
	Hammer cushion material	Area	Variable 3 ($x3$)
		Elastic modulus	Variable 4 ($x4$)
		Thick	Variable 5 ($x5$)
		Helmet weight	Variable 6 ($x6$)
	Pile information	Length	Variable 7 ($x7$)
		Penetration	Variable 8 ($x8$)
		Diameter	Variable 9 ($x9$)
		Section area	Variable 10 ($x10$)
		L/D	Variable 11 ($x11$)
	Soil information	Quake at toe	Variable 12 ($x12$)
		Damp at shaft	Variable 13 ($x13$)
		Damp at toe	Variable 14 ($x14$)
		Shaft resistance	Variable 15 ($x15$)
	Ultimate pile capacity Q_u (kN)		Variable 16 ($x16$)
	Stroke		Variable 17 ($x17$)
Outputs	Maximum compressive stress MCS (MPa) Maximum tensile stress MTS (MPa) BPF		

Table 2 Summary of HP-pile input variables and outputs.



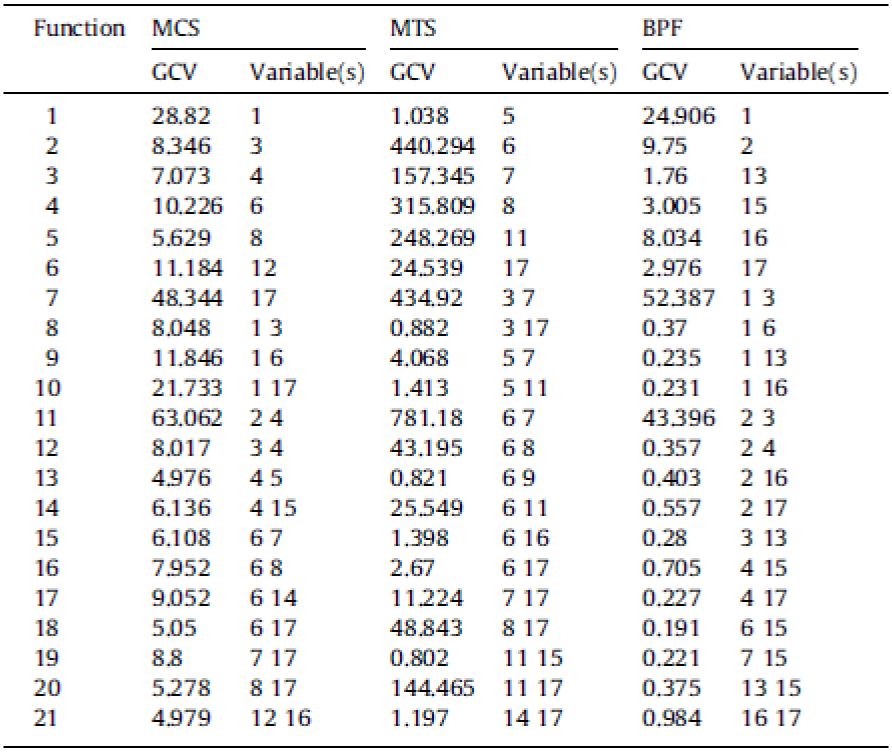
Outputs	MARS models		
	MCS	MTS	BPF
Type of BFs	Piecewise-cubic	Piecewise-linear	Piecewise-linear
No. of BFs	42	43	40
Max interaction	2	2	2
R^2 of training data	0.928	0.965	0.986
R^2 of testing data	0.944	0.944	0.983
GCV	5.127	0.374	0.164

Table 3 MARS models to predict MCS, MTS and BPF.



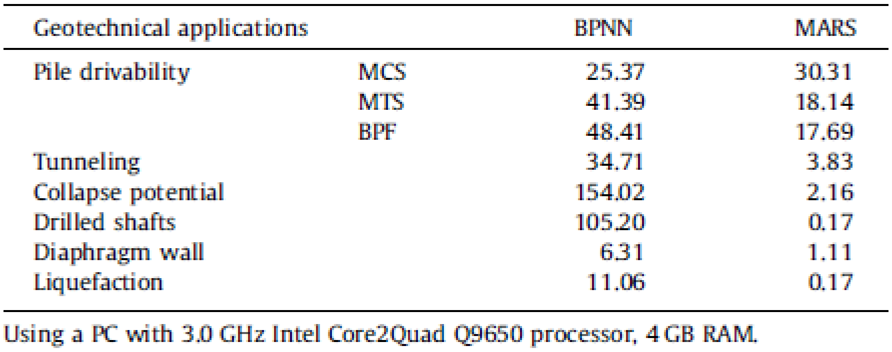
Geotechnical applications		BPNN						MARS					
		Training			Testing			Training			Testing		
		R^2	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE	R^2	MSE	MAE
Pile drivability	MCS	0.998	2.635	1.126	0.982	22.06	3.344	0.928	89.80	5.480	0.944	67.82	5.957
	MTS	0.964	6.673	15.13	0.903	15.72	2.049	0.917	15.11	21.22	0.944	9.088	2.246
	BPF	0.998	0.008	0.066	0.976	0.131	0.163	0.986	0.064	0.190	0.983	0.198	0.094
Tunneling		0.873	43.13	4.186	0.689	27.62	3.639	0.906	31.94	3.236	0.721	24.77	3.576
Collapse potential		0.911	4.409	1.514	0.914	4.329	1.507	0.948	2.556	1.282	0.926	3.715	1.524
Drilled shafts		0.877	0.004	0.047	0.836	0.005	0.058	0.876	0.004	0.048	0.812	0.006	0.060
Diaphragm wall		0.987	65.69	6.059	0.986	61.84	5.715	0.938	303.6	12.21	0.949	233.2	11.43
Liquefaction		Overall SR: 97.1%						Overall SR: 90.6%					
		SR in predicting liquefied: 98.1%						SR in predicting liquefied: 95.2%					

Table 4 Modeling accuracy comparison between BPNN and MARS.



Function	MCS		MTS		BPF	
	GCV	Variable(s)	GCV	Variable(s)	GCV	Variable(s)
1	28.82	1	1.038	5	24.906	1
2	8.346	3	440.294	6	9.75	2
3	7.073	4	157.345	7	1.76	13
4	10.226	6	315.809	8	3.005	15
5	5.629	8	248.269	11	8.034	16
6	11.184	12	24.539	17	2.976	17
7	48.344	17	434.92	3 7	52.387	1 3
8	8.048	1 3	0.882	3 17	0.37	1 6
9	11.846	1 6	4.068	5 7	0.235	1 13
10	21.733	1 17	1.413	5 11	0.231	1 16
11	63.062	2 4	781.18	6 7	43.396	2 3
12	8.017	3 4	43.195	6 8	0.357	2 4
13	4.976	4 5	0.821	6 9	0.403	2 16
14	6.136	4 15	25.549	6 11	0.557	2 17
15	6.108	6 7	1.398	6 16	0.28	3 13
16	7.952	6 8	2.67	6 17	0.705	4 15
17	9.052	6 14	11.224	7 17	0.227	4 17
18	5.05	6 17	48.843	8 17	0.191	6 15
19	8.8	7 17	0.802	11 15	0.221	7 15
20	5.278	8 17	144.465	11 17	0.375	13 15
21	4.979	12 16	1.197	14 17	0.984	16 17

Table 5 ANOVA decomposition of MARS models.

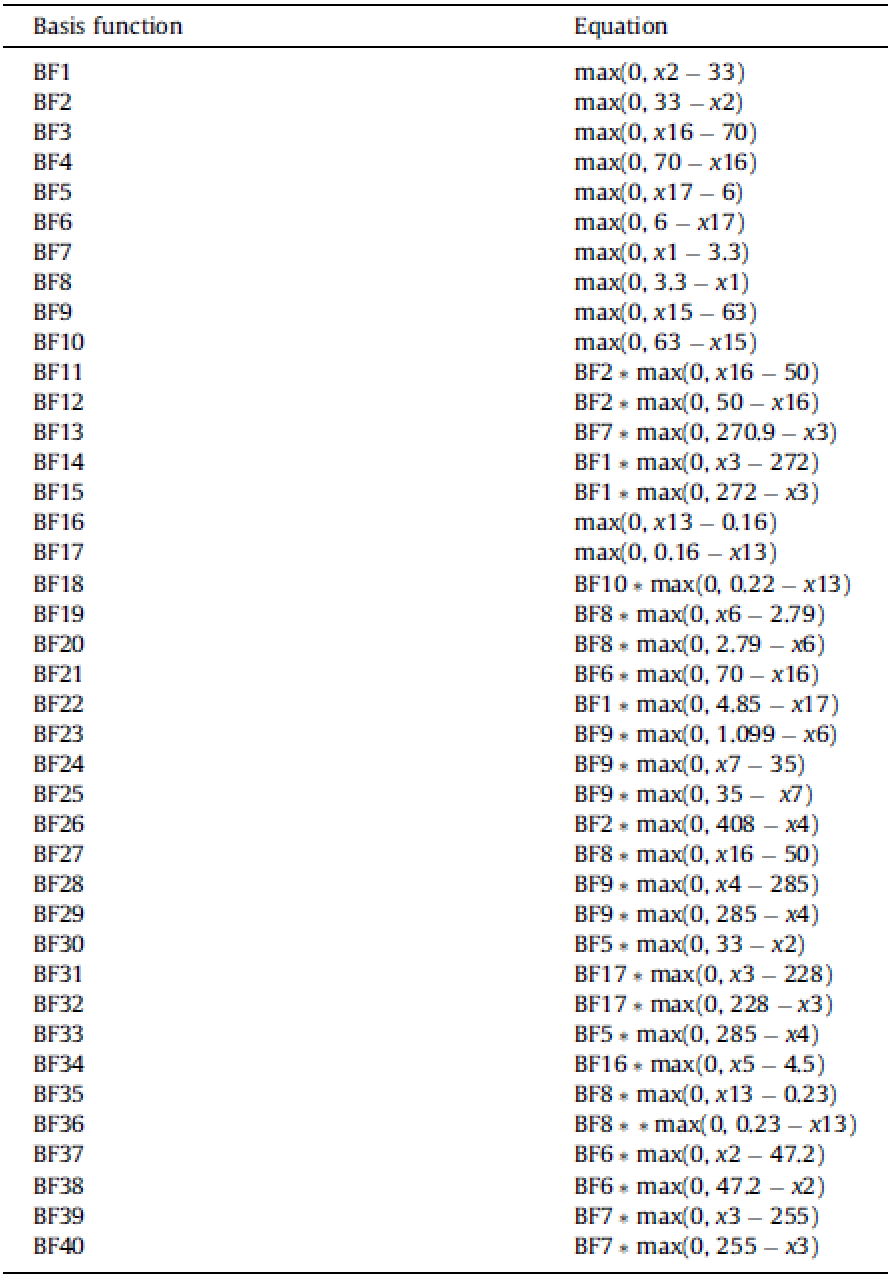


Geotechnical applications		BPNN	MARS
Pile drivability	MCS	25.37	30.31
	MTS	41.39	18.14
	BPF	48.41	17.69
Tunneling		34.71	3.83
Collapse potential		154.02	2.16
Drilled shafts		105.20	0.17
Diaphragm wall		6.31	1.11
Liquefaction		11.06	0.17

Using a PC with 3.0 GHz Intel Core2Quad Q9650 processor, 4 GB RAM.

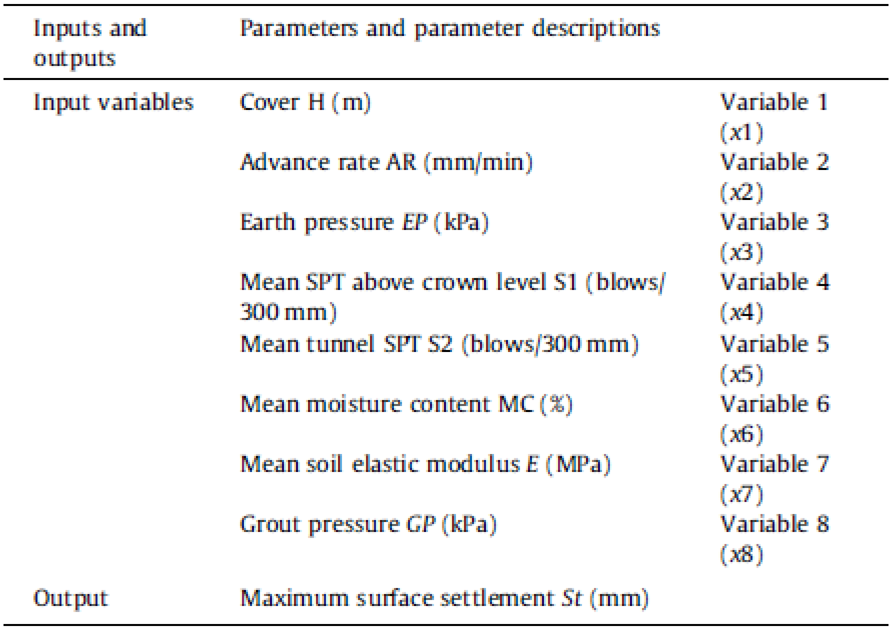
(using a PC with 3.0 GHz Intel Core2Quad Q9650 processor, 4GB RAM)

Table 6 Processing time comparison between BPNN and MARS (Units: s).



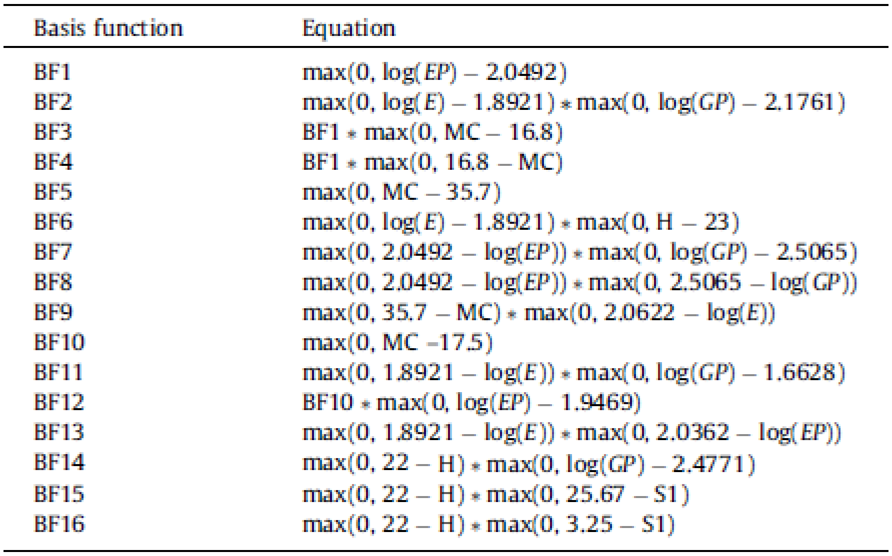
Basis function	Equation
BF1	$\max(0, x2 - 33)$
BF2	$\max(0, 33 - x2)$
BF3	$\max(0, x16 - 70)$
BF4	$\max(0, 70 - x16)$
BF5	$\max(0, x17 - 6)$
BF6	$\max(0, 6 - x17)$
BF7	$\max(0, x1 - 3.3)$
BF8	$\max(0, 3.3 - x1)$
BF9	$\max(0, x15 - 63)$
BF10	$\max(0, 63 - x15)$
BF11	$BF2 * \max(0, x16 - 50)$
BF12	$BF2 * \max(0, 50 - x16)$
BF13	$BF7 * \max(0, 270.9 - x3)$
BF14	$BF1 * \max(0, x3 - 272)$
BF15	$BF1 * \max(0, 272 - x3)$
BF16	$\max(0, x13 - 0.16)$
BF17	$\max(0, 0.16 - x13)$
BF18	$BF10 * \max(0, 0.22 - x13)$
BF19	$BF8 * \max(0, x6 - 2.79)$
BF20	$BF8 * \max(0, 2.79 - x6)$
BF21	$BF6 * \max(0, 70 - x16)$
BF22	$BF1 * \max(0, 4.85 - x17)$
BF23	$BF9 * \max(0, 1.099 - x6)$
BF24	$BF9 * \max(0, x7 - 35)$
BF25	$BF9 * \max(0, 35 - x7)$
BF26	$BF2 * \max(0, 408 - x4)$
BF27	$BF8 * \max(0, x16 - 50)$
BF28	$BF9 * \max(0, x4 - 285)$
BF29	$BF9 * \max(0, 285 - x4)$
BF30	$BF5 * \max(0, 33 - x2)$
BF31	$BF17 * \max(0, x3 - 228)$
BF32	$BF17 * \max(0, 228 - x3)$
BF33	$BF5 * \max(0, 285 - x4)$
BF34	$BF16 * \max(0, x5 - 4.5)$
BF35	$BF8 * \max(0, x13 - 0.23)$
BF36	$BF8 * * \max(0, 0.23 - x13)$
BF37	$BF6 * \max(0, x2 - 47.2)$
BF38	$BF6 * \max(0, 47.2 - x2)$
BF39	$BF7 * \max(0, x3 - 255)$
BF40	$BF7 * \max(0, 255 - x3)$

Table 7 Basis functions and corresponding equations of MARS model for BPF prediction.



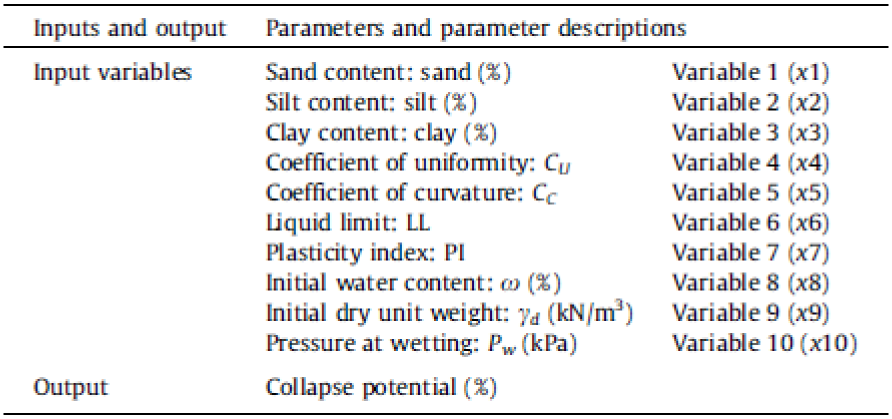
Inputs and outputs	Parameters and parameter descriptions	
Input variables	Cover H (m)	Variable 1 ($x1$)
	Advance rate AR (mm/min)	Variable 2 ($x2$)
	Earth pressure *EP* (kPa)	Variable 3 ($x3$)
	Mean SPT above crown level S1 (blows/300 mm)	Variable 4 ($x4$)
	Mean tunnel SPT S2 (blows/300 mm)	Variable 5 ($x5$)
	Mean moisture content MC (%)	Variable 6 ($x6$)
	Mean soil elastic modulus *E* (MPa)	Variable 7 ($x7$)
	Grout pressure *GP* (kPa)	Variable 8 ($x8$)
Output	Maximum surface settlement *St* (mm)	

Table 8 Summary of tunnel settlement input variables and output.



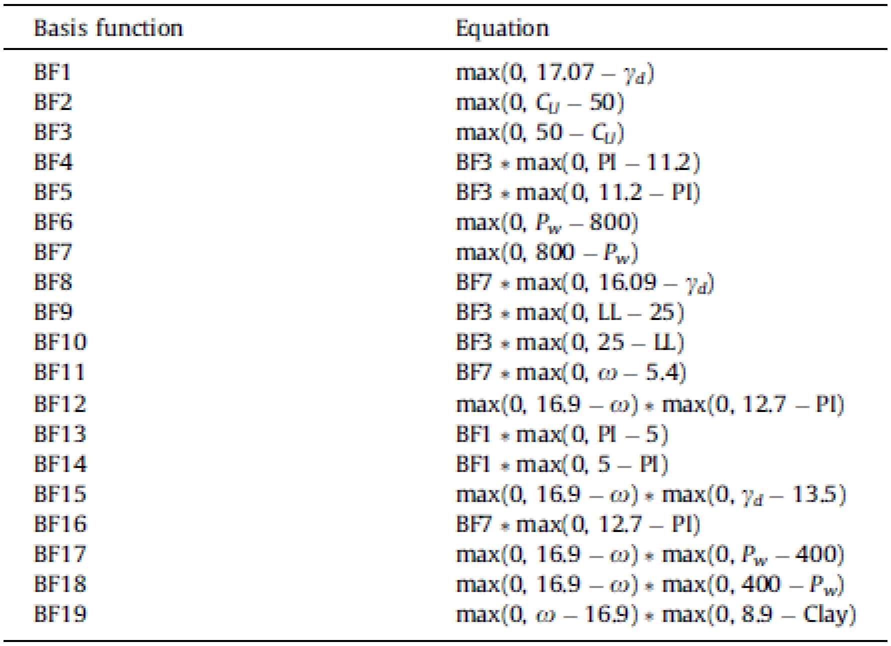
Basis function	Equation
BF1	$\max(0, \log(EP) - 2.0492)$
BF2	$\max(0, \log(E) - 1.8921) * \max(0, \log(GP) - 2.1761)$
BF3	$\text{BF1} * \max(0, \text{MC} - 16.8)$
BF4	$\text{BF1} * \max(0, 16.8 - \text{MC})$
BF5	$\max(0, \text{MC} - 35.7)$
BF6	$\max(0, \log(E) - 1.8921) * \max(0, \text{H} - 23)$
BF7	$\max(0, 2.0492 - \log(EP)) * \max(0, \log(GP) - 2.5065)$
BF8	$\max(0, 2.0492 - \log(EP)) * \max(0, 2.5065 - \log(GP))$
BF9	$\max(0, 35.7 - \text{MC}) * \max(0, 2.0622 - \log(E))$
BF10	$\max(0, \text{MC} - 17.5)$
BF11	$\max(0, 1.8921 - \log(E)) * \max(0, \log(GP) - 1.6628)$
BF12	$\text{BF10} * \max(0, \log(EP) - 1.9469)$
BF13	$\max(0, 1.8921 - \log(E)) * \max(0, 2.0362 - \log(EP))$
BF14	$\max(0, 22 - \text{H}) * \max(0, \log(GP) - 2.4771)$
BF15	$\max(0, 22 - \text{H}) * \max(0, 25.67 - S1)$
BF16	$\max(0, 22 - \text{H}) * \max(0, 3.25 - S1)$

Table 9 Basis functions and corresponding equations of MARS model for settlement prediction.



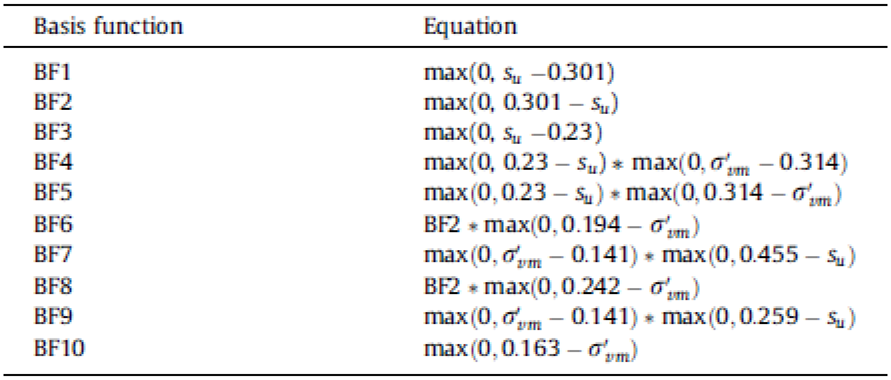
Inputs and output	Parameters and parameter descriptions	
Input variables	Sand content: sand (%)	Variable 1 ($x1$)
	Silt content: silt (%)	Variable 2 ($x2$)
	Clay content: clay (%)	Variable 3 ($x3$)
	Coefficient of uniformity: C_U	Variable 4 ($x4$)
	Coefficient of curvature: C_C	Variable 5 ($x5$)
	Liquid limit: LL	Variable 6 ($x6$)
	Plasticity index: PI	Variable 7 ($x7$)
	Initial water content: ω (%)	Variable 8 ($x8$)
	Initial dry unit weight: γ_d (kN/m^3)	Variable 9 ($x9$)
	Pressure at wetting: P_w (kPa)	Variable 10 ($x10$)
Output	Collapse potential (%)	

Table 10 Summary of collapse potential input variables and output.



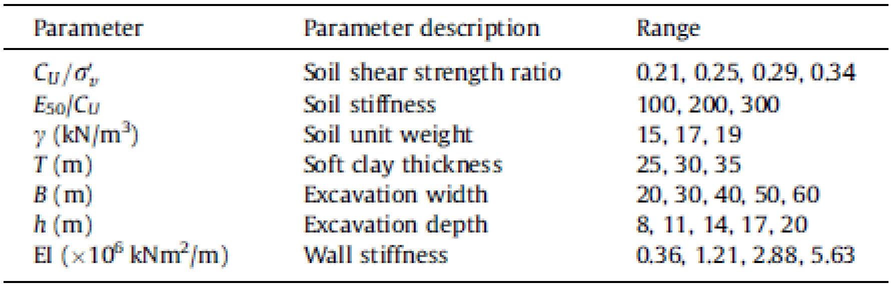
Basis function	Equation
BF1	$\max(0, 17.07 - \gamma_d)$
BF2	$\max(0, C_U - 50)$
BF3	$\max(0, 50 - C_U)$
BF4	$BF3 * \max(0, PI - 11.2)$
BF5	$BF3 * \max(0, 11.2 - PI)$
BF6	$\max(0, P_w - 800)$
BF7	$\max(0, 800 - P_w)$
BF8	$BF7 * \max(0, 16.09 - \gamma_d)$
BF9	$BF3 * \max(0, LL - 25)$
BF10	$BF3 * \max(0, 25 - LL)$
BF11	$BF7 * \max(0, \omega - 5.4)$
BF12	$\max(0, 16.9 - \omega) * \max(0, 12.7 - PI)$
BF13	$BF1 * \max(0, PI - 5)$
BF14	$BF1 * \max(0, 5 - PI)$
BF15	$\max(0, 16.9 - \omega) * \max(0, \gamma_d - 13.5)$
BF16	$BF7 * \max(0, 12.7 - PI)$
BF17	$\max(0, 16.9 - \omega) * \max(0, P_w - 400)$
BF18	$\max(0, 16.9 - \omega) * \max(0, 400 - P_w)$
BF19	$\max(0, \omega - 16.9) * \max(0, 8.9 - Clay)$

Table 11 Basis functions and corresponding equations of MARS model for *CP* prediction.



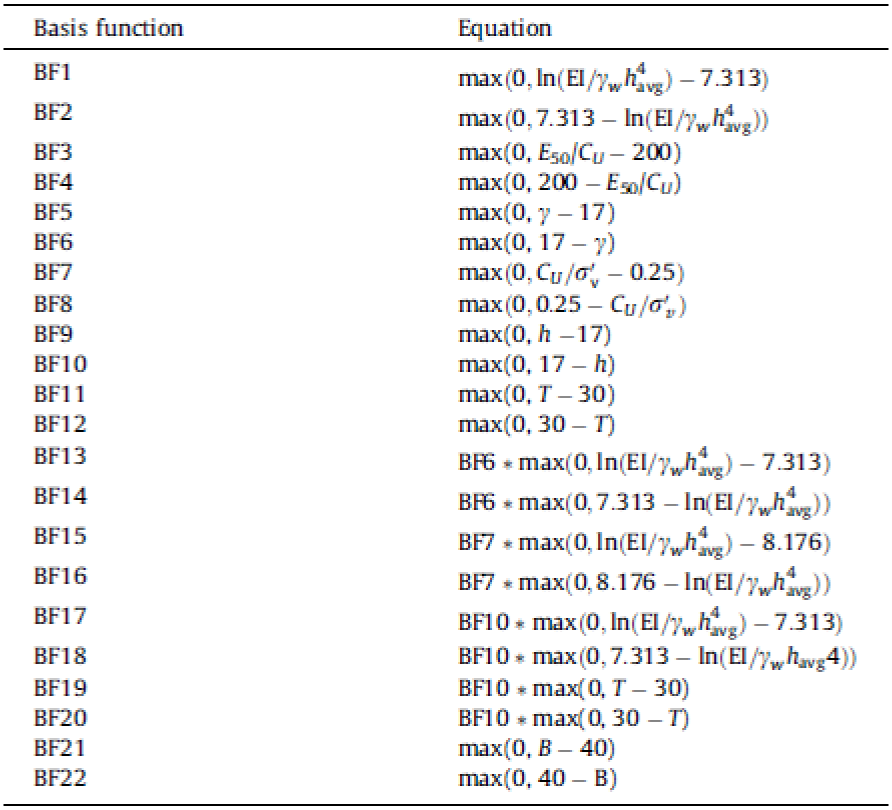
Basis function	Equation
BF1	$\max(0,\ s_u - 0.301)$
BF2	$\max(0,\ 0.301 - s_u)$
BF3	$\max(0,\ s_u - 0.23)$
BF4	$\max(0,\ 0.23 - s_u) * \max(0, \sigma'_{vm} - 0.314)$
BF5	$\max(0, 0.23 - s_u) * \max(0, 0.314 - \sigma'_{vm})$
BF6	$BF2 * \max(0, 0.194 - \sigma'_{vm})$
BF7	$\max(0, \sigma'_{vm} - 0.141) * \max(0, 0.455 - s_u)$
BF8	$BF2 * \max(0, 0.242 - \sigma'_{vm})$
BF9	$\max(0, \sigma'_{vm} - 0.141) * \max(0, 0.259 - s_u)$
BF10	$\max(0, 0.163 - \sigma'_{vm})$

Table 12 Basis functions and corresponding equations of MARS model for α prediction.



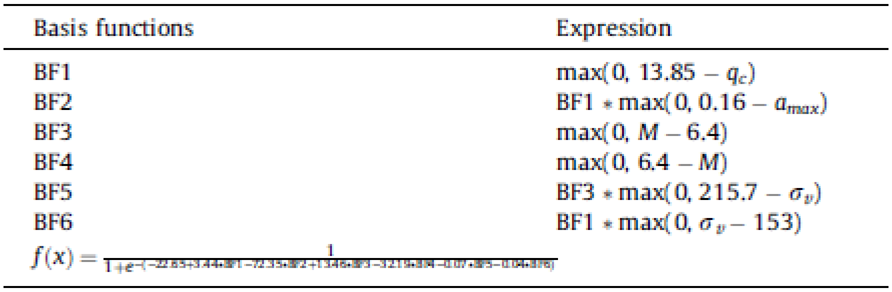
Parameter	Parameter description	Range
C_U/σ'_v	Soil shear strength ratio	0.21, 0.25, 0.29, 0.34
E_{50}/C_U	Soil stiffness	100, 200, 300
γ (kN/m^3)	Soil unit weight	15, 17, 19
T (m)	Soft clay thickness	25, 30, 35
B (m)	Excavation width	20, 30, 40, 50, 60
h (m)	Excavation depth	8, 11, 14, 17, 20
EI ($\times 10^6$ kNm^2/m)	Wall stiffness	0.36, 1.21, 2.88, 5.63

Table 13 Summary of parameters influencing diaphragm wall excavation performance.



Basis function	Equation
BF1	$\max(0, \ln(\mathrm{EI}/\gamma_w h_{avg}^4) - 7.313)$
BF2	$\max(0, 7.313 - \ln(\mathrm{EI}/\gamma_w h_{avg}^4))$
BF3	$\max(0, E_{50}/C_U - 200)$
BF4	$\max(0, 200 - E_{50}/C_U)$
BF5	$\max(0, \gamma - 17)$
BF6	$\max(0, 17 - \gamma)$
BF7	$\max(0, C_U/\sigma'_v - 0.25)$
BF8	$\max(0, 0.25 - C_U/\sigma'_v)$
BF9	$\max(0, h - 17)$
BF10	$\max(0, 17 - h)$
BF11	$\max(0, T - 30)$
BF12	$\max(0, 30 - T)$
BF13	$\mathrm{BF6} * \max(0, \ln(\mathrm{EI}/\gamma_w h_{avg}^4) - 7.313)$
BF14	$\mathrm{BF6} * \max(0, 7.313 - \ln(\mathrm{EI}/\gamma_w h_{avg}^4))$
BF15	$\mathrm{BF7} * \max(0, \ln(\mathrm{EI}/\gamma_w h_{avg}^4) - 8.176)$
BF16	$\mathrm{BF7} * \max(0, 8.176 - \ln(\mathrm{EI}/\gamma_w h_{avg}^4))$
BF17	$\mathrm{BF10} * \max(0, \ln(\mathrm{EI}/\gamma_w h_{avg}^4) - 7.313)$
BF18	$\mathrm{BF10} * \max(0, 7.313 - \ln(\mathrm{EI}/\gamma_w h_{avg}4))$
BF19	$\mathrm{BF10} * \max(0, T - 30)$
BF20	$\mathrm{BF10} * \max(0, 30 - T)$
BF21	$\max(0, B - 40)$
BF22	$\max(0, 40 - \mathrm{B})$

Table 14 Basis functions and corresponding equations of MARS model for diaphragm wall deflection prediction.



Basis functions	Expression
BF1	$\max(0, 13.85 - q_c)$
BF2	$\mathrm{BF1} * \max(0, 0.16 - a_{max})$
BF3	$\max(0, M - 6.4)$
BF4	$\max(0, 6.4 - M)$
BF5	$\mathrm{BF3} * \max(0, 215.7 - \sigma_v)$
BF6	$\mathrm{BF1} * \max(0, \sigma_v - 153)$
$f(x) = \frac{1}{1+e^{-(-22.65+3.44*BF1-72.35*BF2+13.46*BF3-32.19*BF4-0.07*BF5-0.04*BF6)}}$	

Table 15 Basis functions and corresponding equations of MARS model for seismic liquefaction potential assessment

Figures

Fig. 1. Knots and linear splines for a simple MARS example.

Fig. 2. Curve fitting using MARS: a) sine function; b) exponential function.

Fig. 3. A two-variable function for surface fitting.

Fig. 4. Surface fitting using MARS: a) using BFs of linear spline; b) using BFs of cubic spline.

Fig. 5. MARS approximations: a) large error variance of $\sigma^2 = 1$; b) small error variance of $\sigma^2 = 0.25$.

Fig. 6. Comparison of MCS using MARS and BPNN: a) training; b) testing.

Fig. 7. Comparison of MTS using MARS and BPNN: a) training; b) testing.

Fig. 8. Comparison of BPF using MARS and BPNN: a) training; b) testing.

Fig. 9. Relative importance of the input variables selected in the MARS models: a) MCS; b) MTS; and c) BPF.

Fig. 10. Comparison of measured and predicted *St*.

Fig. 11. Performance of MARS model for *CP*: a) training; b) testing.

Fig. 12. Performance of MARS model for α: a) training; b) testing.

Fig. 13. Cross-sectional soil and wall profile (figure from Xuan [17]).

Fig. 14. Performance of MARS model for diaphragm wall deflection: a) training; b) testing.

Fig. 15. Performance of BPNN model for estimating seismic liquefaction potential: a) training; b) testing.

Fig. 16. Performance of MARS model for estimating seismic liquefaction potential: a) training; b) testing.

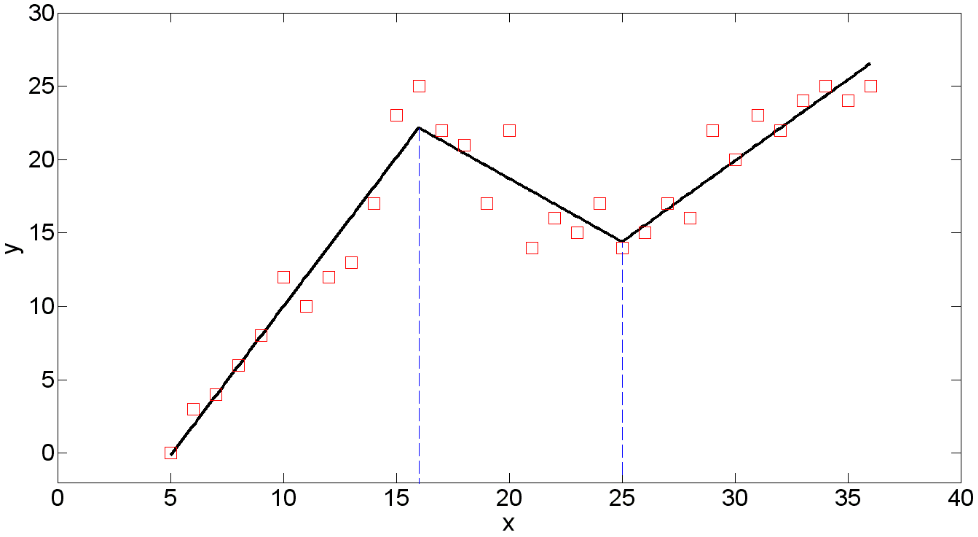


Fig. 1. Knots and linear splines for a simple MARS example.

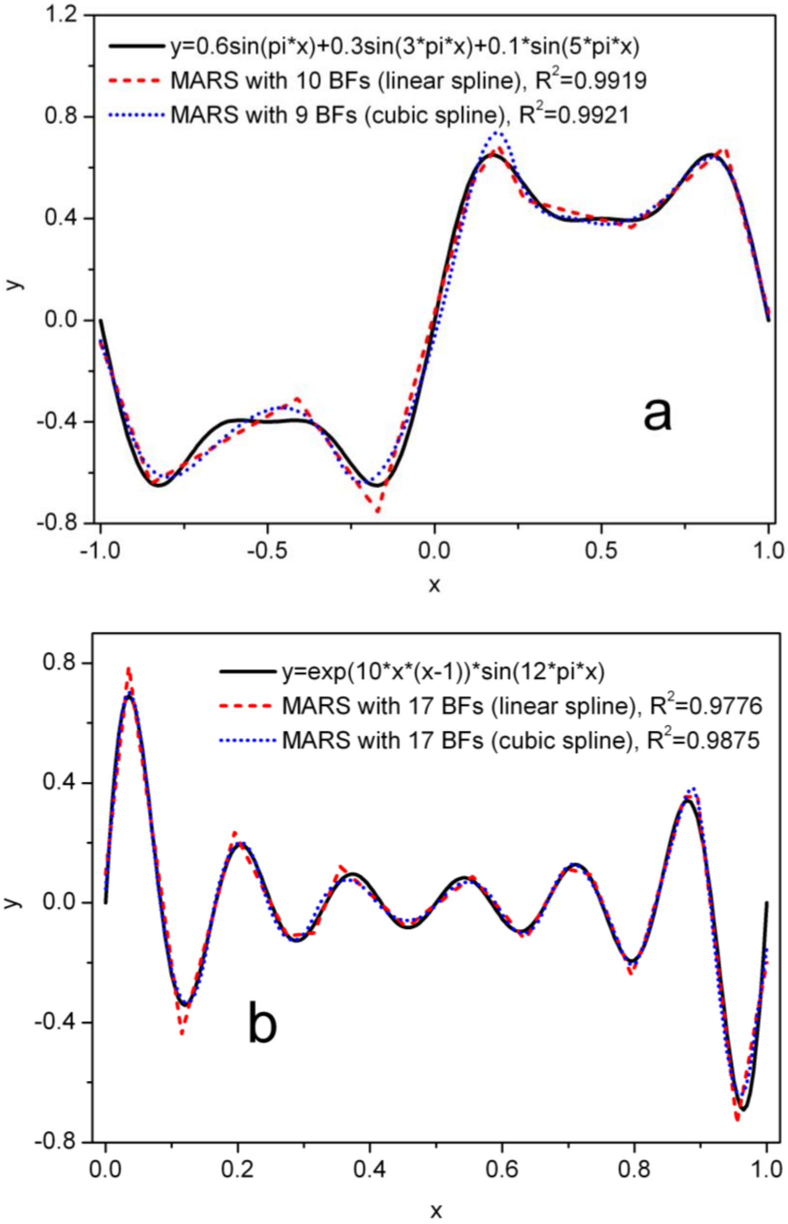


Fig. 2. Curve fitting using MARS: (a) sine function; (b) exponential function

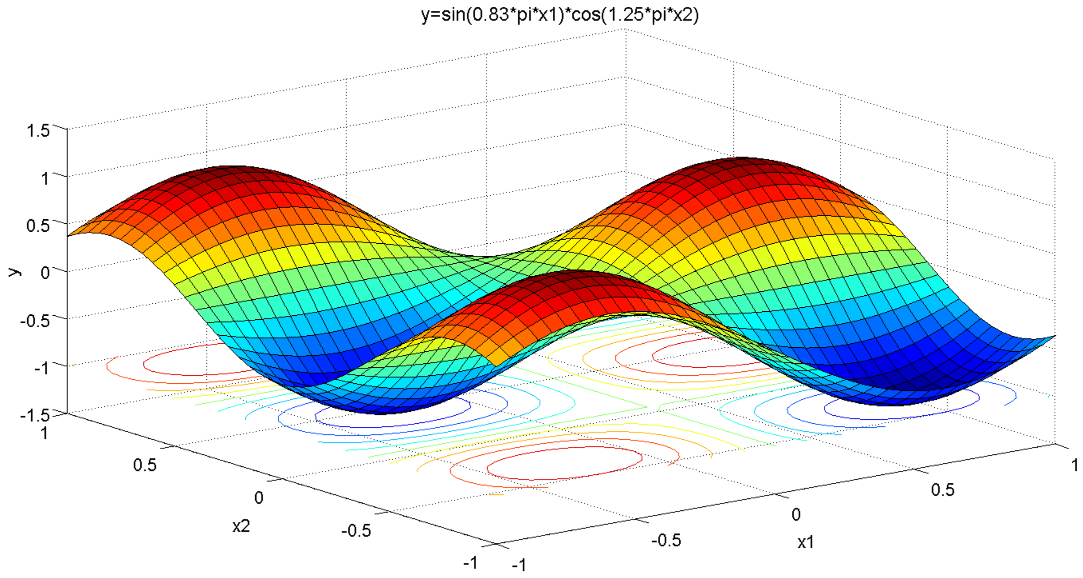


Fig. 3. A two-variable function for surface fitting.

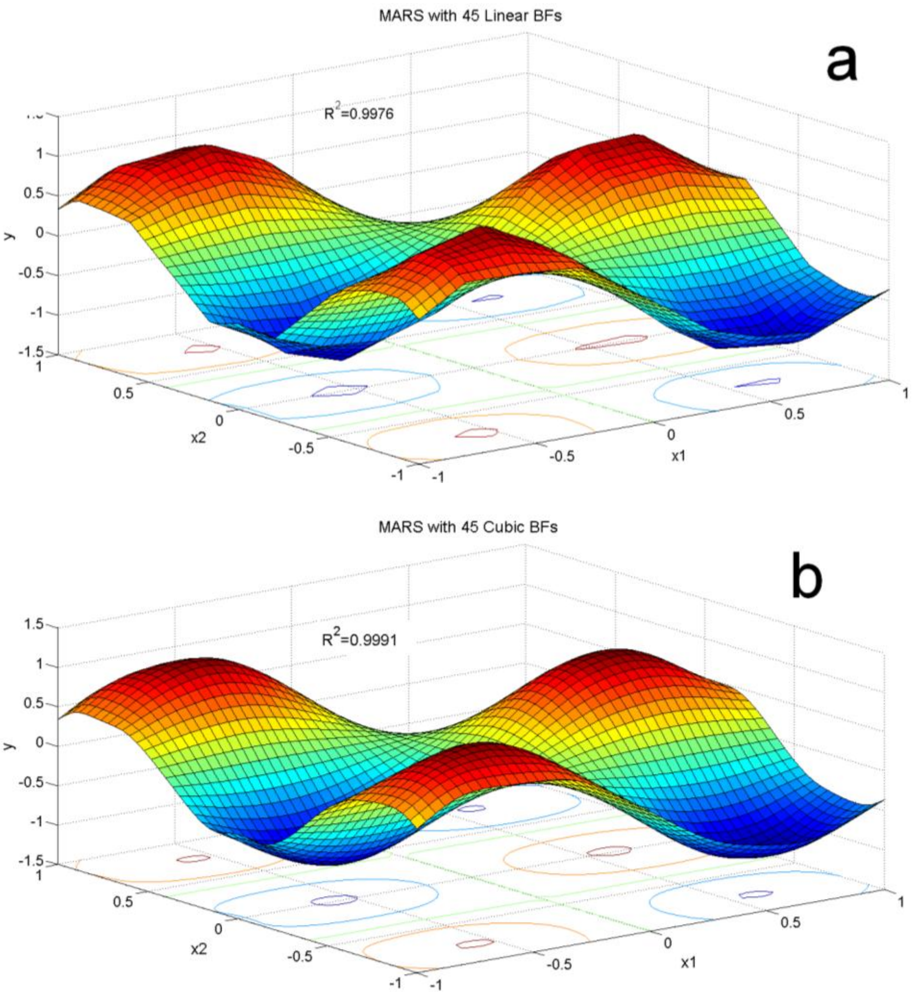


Fig. 4. Surface fitting using MARS: a) using BFs of linear spline; b) using BFs of cubic spline.

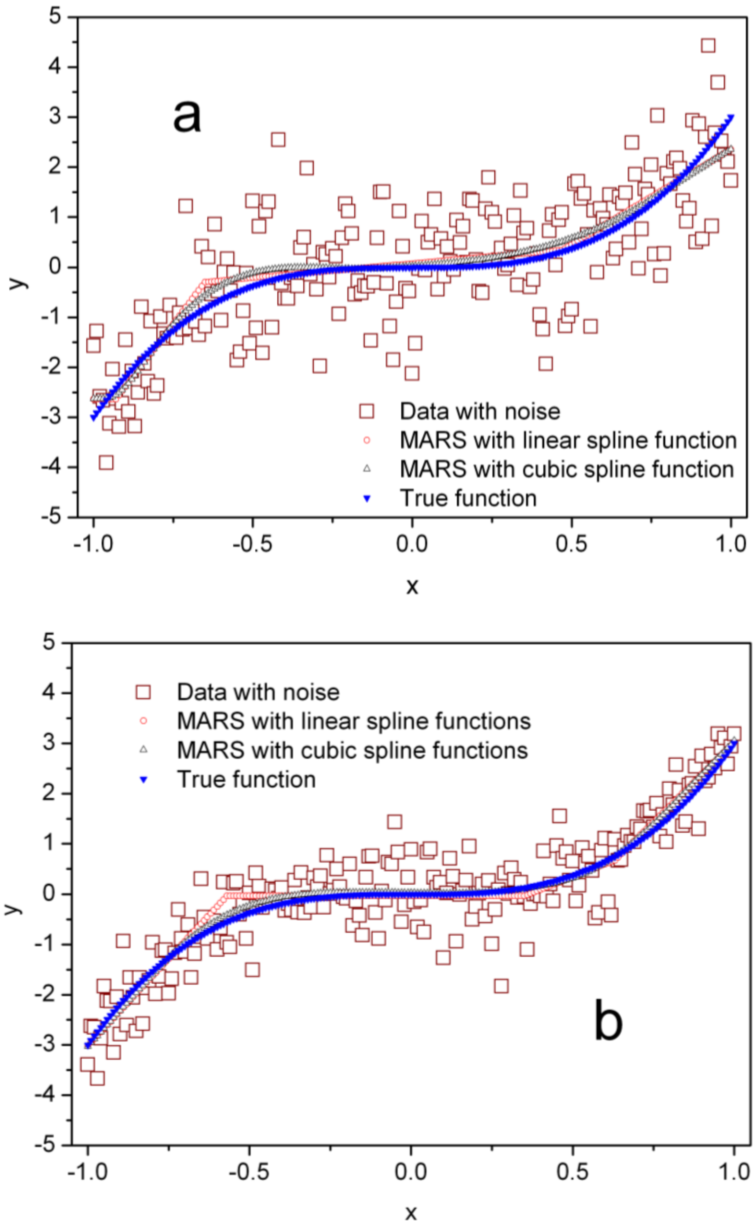


Fig. 5. MARS approximations: a) large error variance of $\sigma^2 = 1$; b) small error variance of $\sigma^2 = 0.25$.

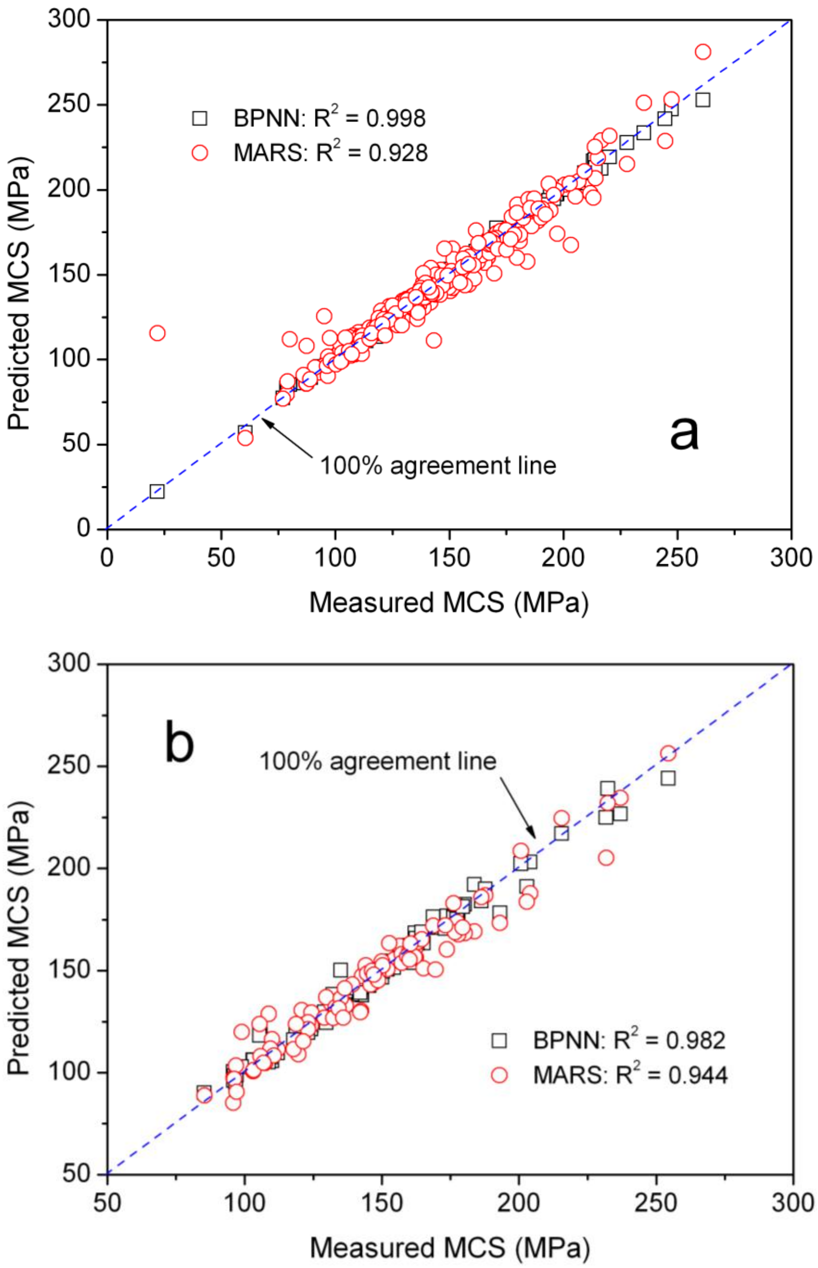


Fig. 6. Comparison of MCS using MARS and BPNN: a) training; b) testing.

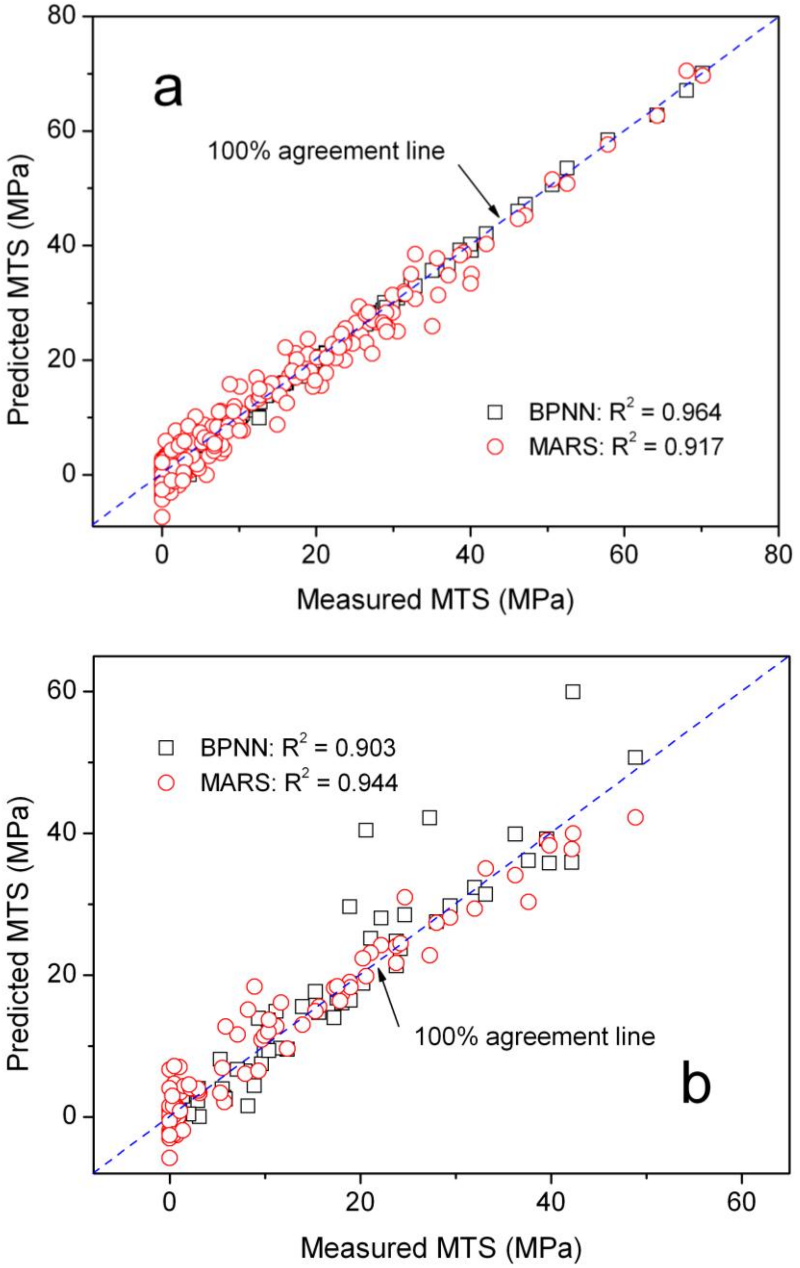


Fig. 7. Comparison of MTS using MARS and BPNN: a) training; b) testing.

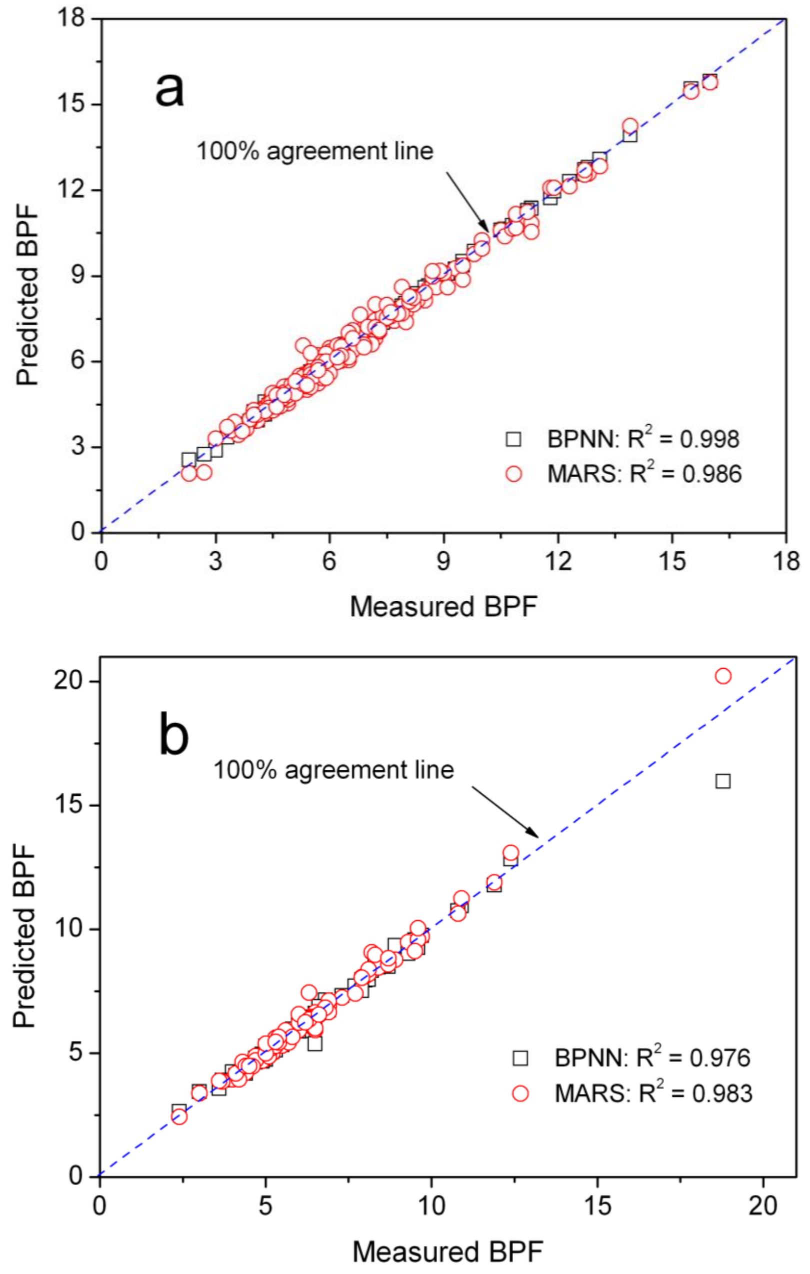


Fig. 8. Comparison of BPF using MARS and BPNN: a) training; b) testing.

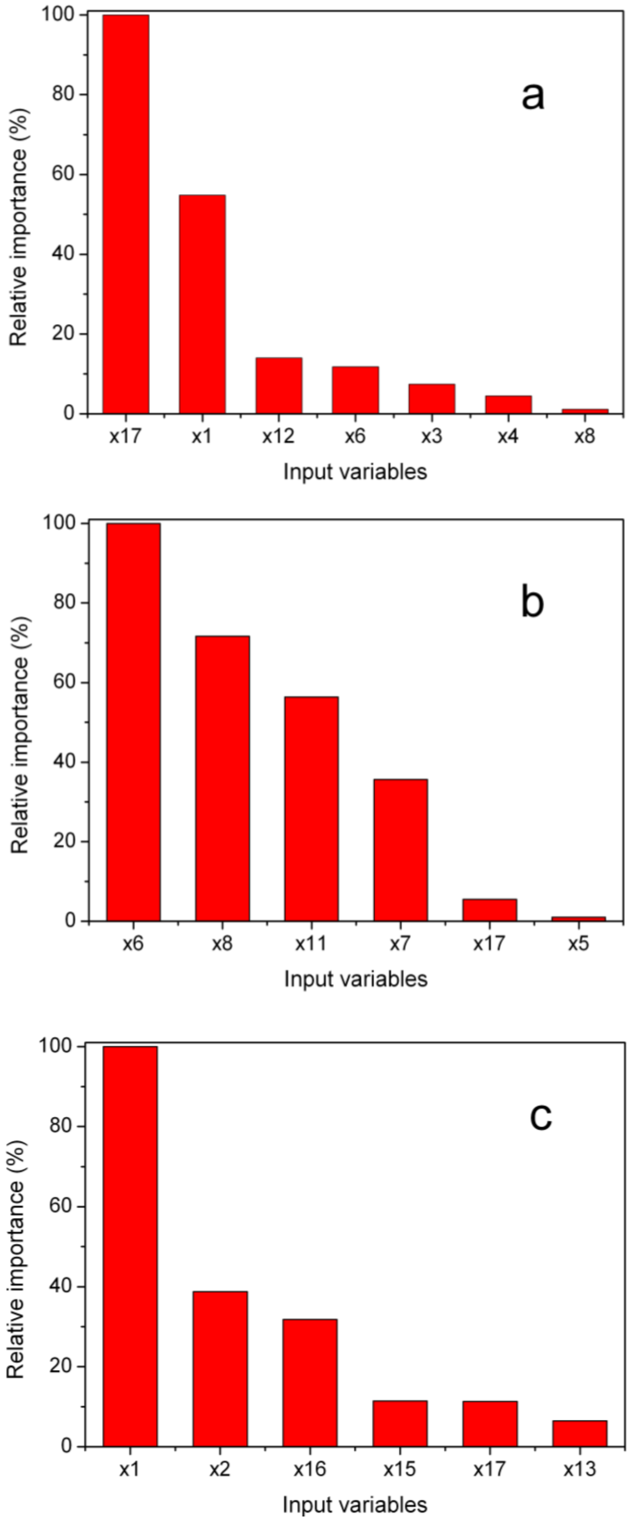


Fig. 9. Relative importance of the input variables selected in the MARS models: a) MCS; b) MTS; and c) BPF.

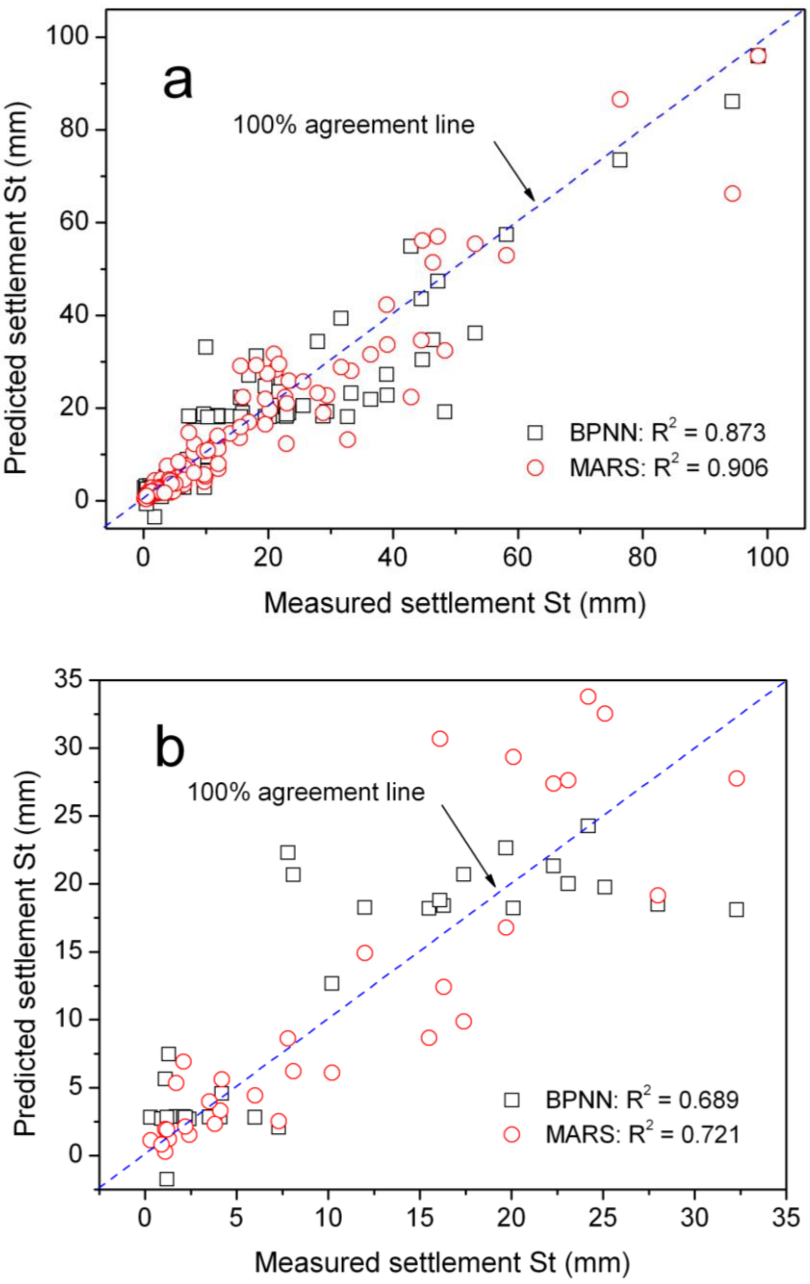


Fig. 10. Comparison of measured and predicted *St*.

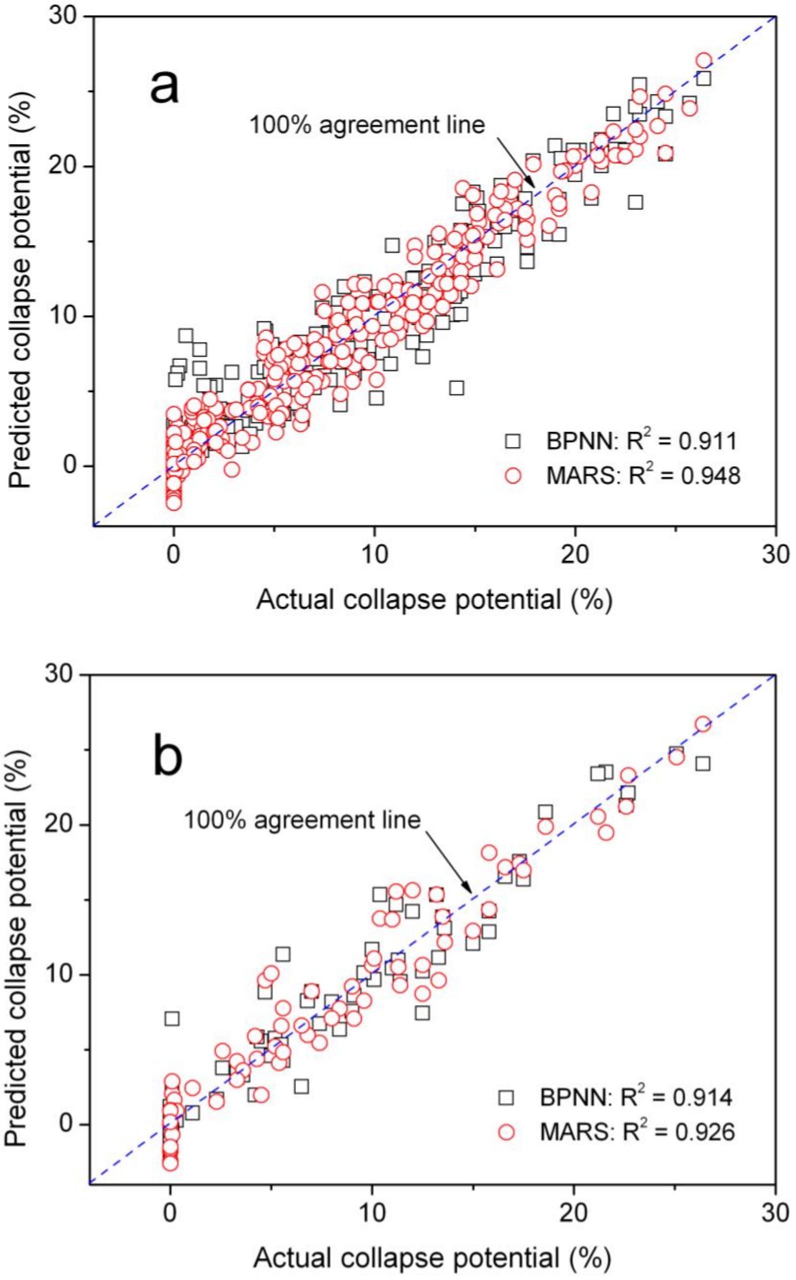


Fig. 11. Performance of MARS model for *CP*: a) training; b) testing.

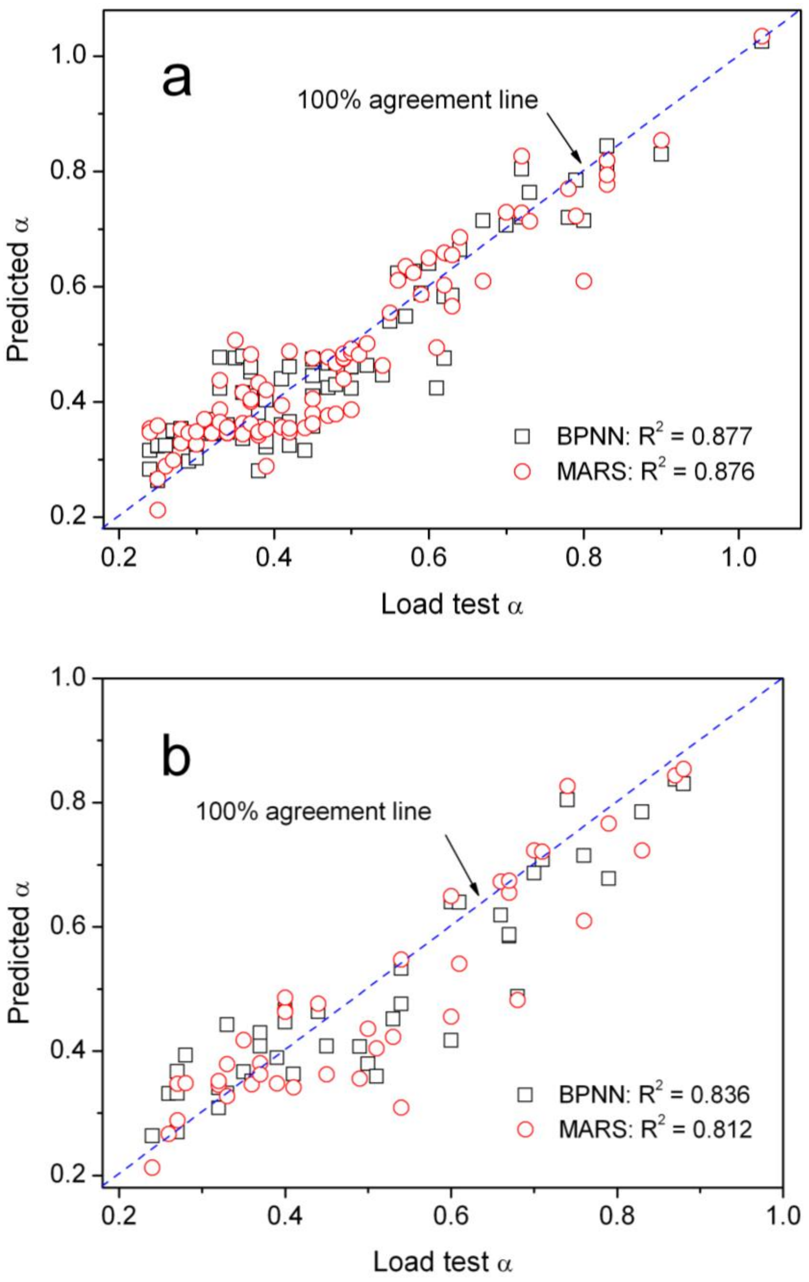


Fig. 12. Performance of MARS model for α: a) training; b) testing.

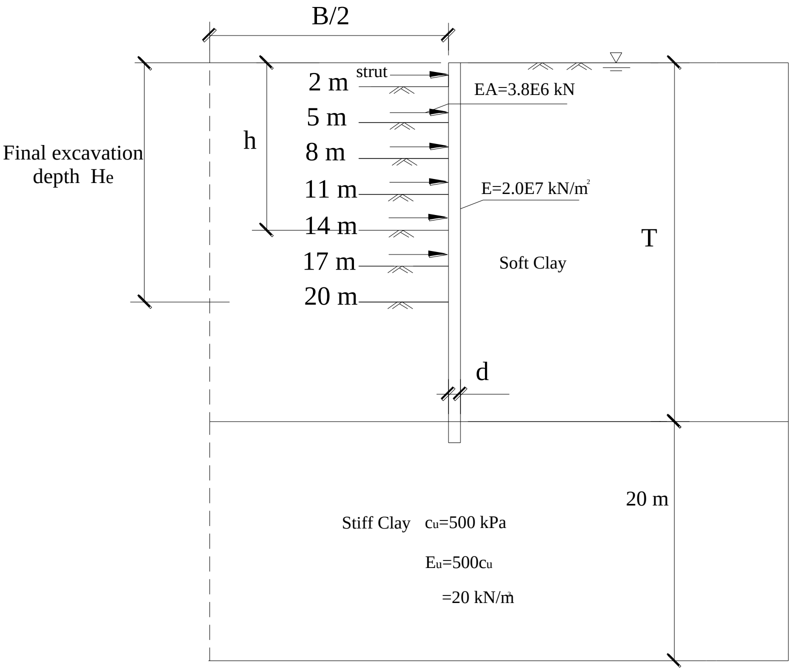


Fig. 13. Cross-sectional soil and wall profile (figure from Xuan [17]).

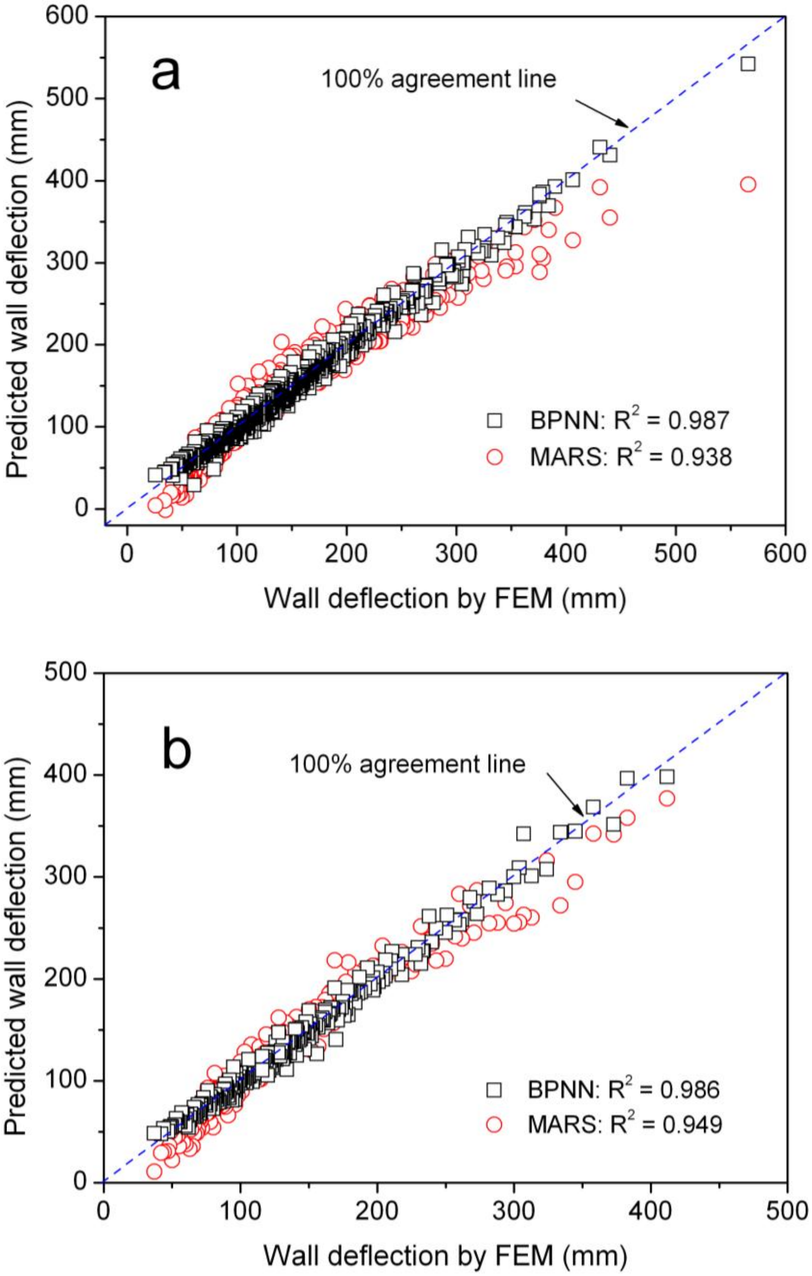


Fig. 14. Performance of MARS model for diaphragm wall deflection: a) training; b) testing.

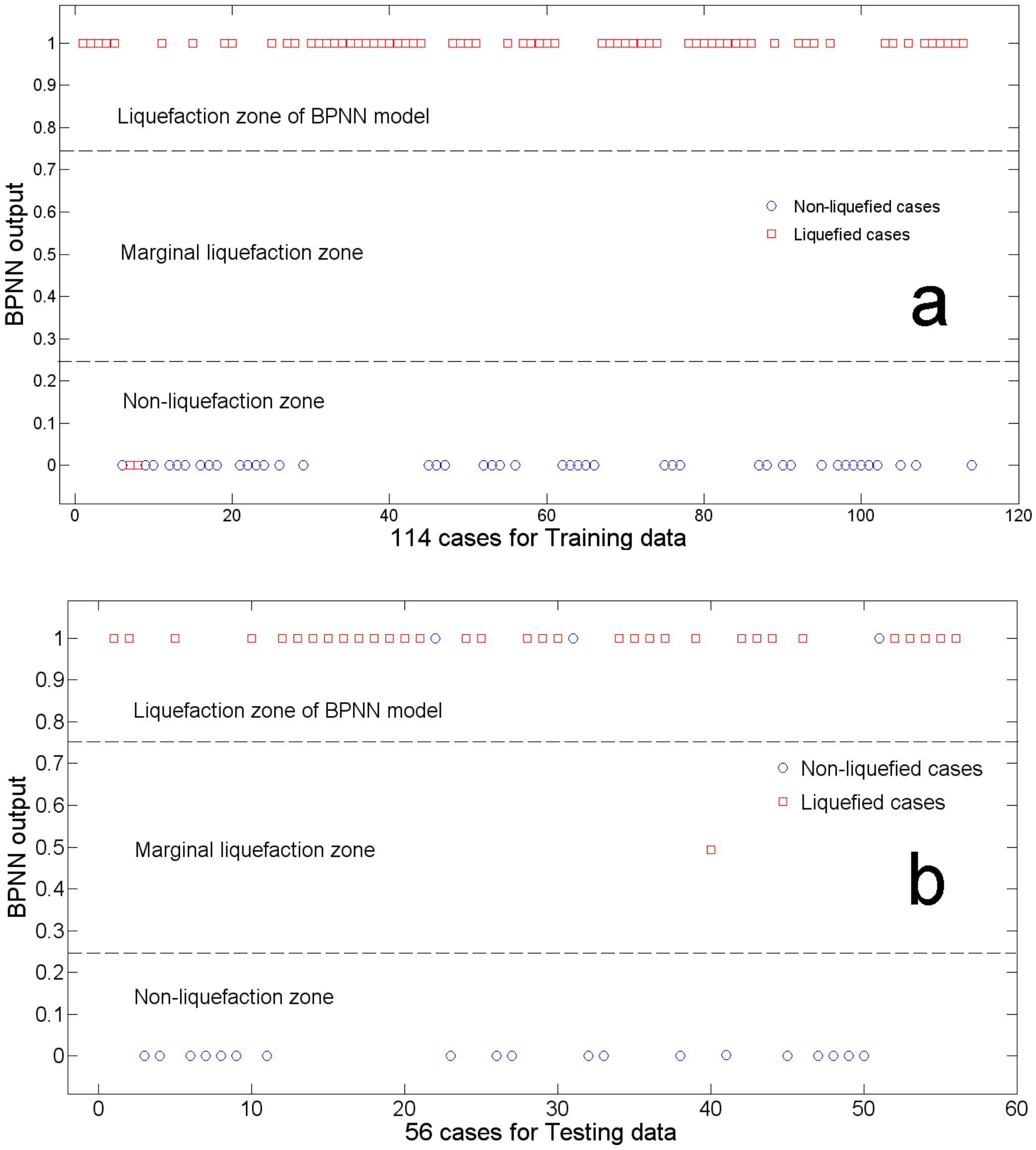


Fig. 15. Performance of BPNN model for estimating seismic liquefaction potential: a) training; b) testing.

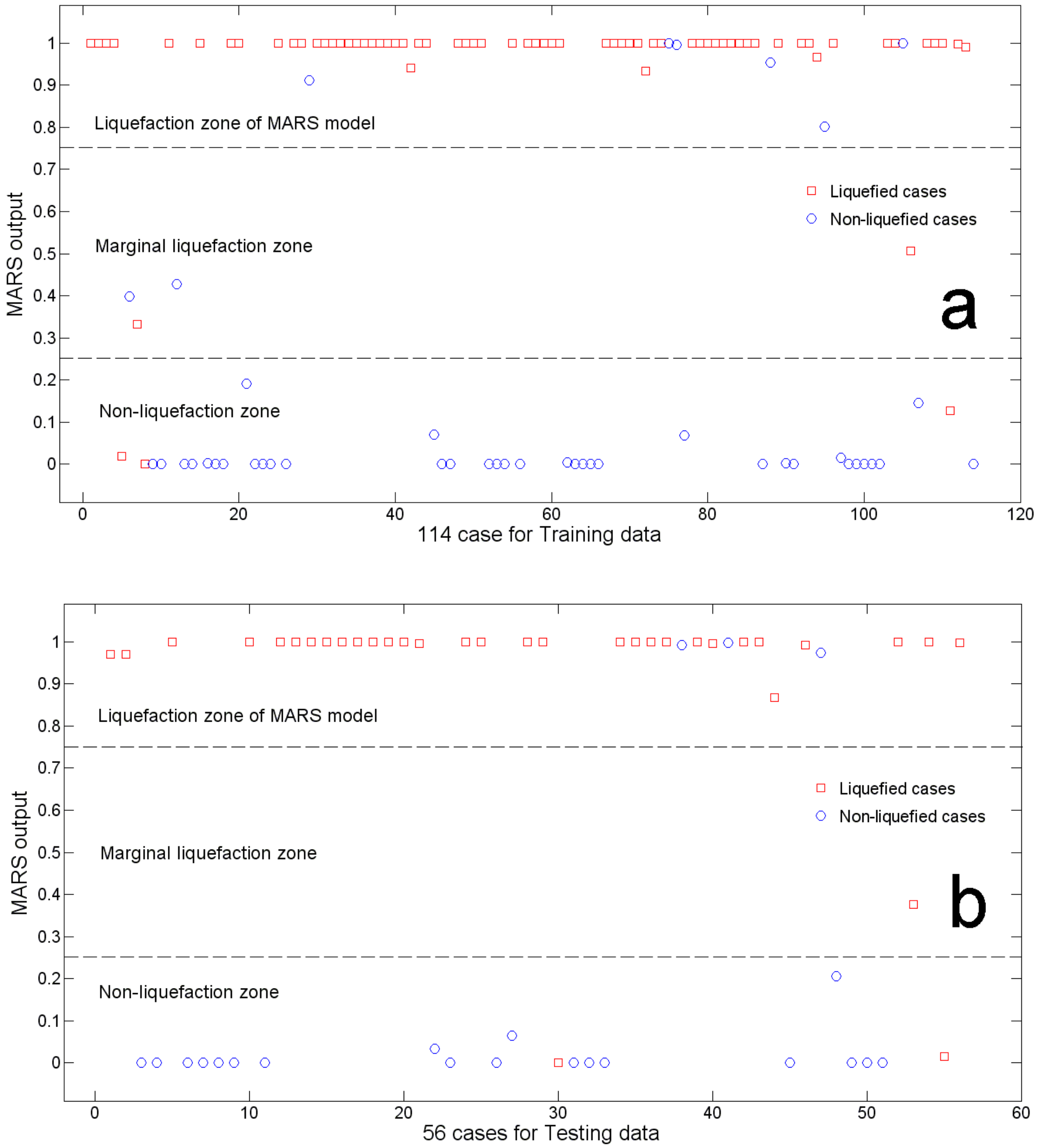


Fig. 16. Performance of MARS model for estimating seismic liquefaction potential: a) training; b) testing.