Modelling of Pile Drivability Using Soft Computing Methods

Wengang Zhang and Anthony T. C. Goh

Abstract Driven piles are commonly used to transfer the loads from the superstructure through weak strata onto stiffer soils or rocks. For driven piles, the impact of the piling hammer induces compression and tension stresses in the piles. Hence, an important design consideration is to check that the strength of the pile is sufficient to resist the stresses caused by the impact of the pile hammer. Due to its complexity, pile drivability lacks a precise analytical theory or understanding of the phenomena involved. In situations where measured or numerical data are available, various soft computing methods have shown to offer great promise for mapping the nonlinear interactions between the system's inputs and outputs. In this study, two soft computing methods, the Back propagation neural network (BPNN) and Multivariate adaptive regression splines (MARS) algorithms were used to assess pile drivability in terms of the Maximum compressive stresses, Maximum tensile stresses, and Blow per foot. A database of more than four thousand piles is utilized for model development and comparative performance of the predictions between BPNN and MARS.

Keywords Back propagation neural network · Multivariate adaptive regression splines · Pile drivability · Computational efficiency · Nonlinearity

W. Zhang
Key Laboratory of New Technology for Construction of Cities in Mountain Area,
Chongqing University, Chongqing 400045, China

W. Zhang
School of Civil Engineering, Chongqing University, Chongqing 400045, China

A. T. C. Goh (✉)
School of Civil and Environmental Engineering, Nanyang Technological University,
Singapore 639798, Singapore
e-mail: ctcgoh@ntu.edu.sg

S. S. Roy et al. (eds.), *Big Data in Engineering Applications*,
Studies in Big Data 44, https://doi.org/10.1007/978-981-10-8476-8_14

1 Introduction

Driven piles are commonly used to transfer the loads from the superstructure through weak strata onto stiffer soils or rocks. For these piles, the impact of the piling hammer induces compression and tension stresses in the piles. Hence, an important design consideration is to ensure that the strength of the pile is sufficient to resist the stresses introduced by the impact of the pile hammer. One common method of calculating the driving stresses is based on the stress-wave theory [18] which involves the discrete idealization of the hammer-pile-soil system. Considering that the conditions at each site are different, generally a wave equation based computer program is required to generate the pile driving criteria for each individual project. The pile driving criteria include:

- Hammer stroke versus Blow per foot BPF (1/set) for required bearing capacity,
- Maximum compressive stresses versus BPF,
- Maximum tension stress versus BPF.

However, this process can be rather time consuming and requires very specialized knowledge of the wave equation program.

The essence of modeling/numerical mapping is prediction, which is obtained by relating a set of variables in input space to a set of response variables in output space through a model. The analysis of pile drivability involves a large number of design variables and nonlinear responses, particularly with statistically dependent inputs. Thus, the commonly used regression models become computationally impractical. Another limitation is the strong model assumptions made by these regression methods.

An alternative soft computing technique is the artificial neural network (ANN). The ANN structure consists of one or more layers of interconnected neurons or nodes. Each link connecting each neuron has an associated weight. The “learning” paradigm in the commonly used Back-propagation (BP) algorithm [14] involves presenting examples of input and output patterns and subsequently adjusting the connecting weights so as to reduce the errors between the actual and the target output values. The iterative modification of the weights is carried out using the gradient descent approach and training is stopped once the errors have been reduced to some acceptable level. The ability of the trained ANN model to generalize the correct input-output response is performed in the testing phase and involves presenting the trained neural network with a separate set of data that has never been used during the training process.

This paper explores the use of ANN and another soft computing technique known as multivariate adaptive regression splines (MARS) [3] to capture the intrinsic nonlinear and multidimensional relationship associated with pile drivability. Similar with neural networks, no prior information on the form of the numerical function is required for MARS. The main advantages of MARS lie in its capacity to capture the intrinsic complicated data mapping in high-dimensional data patterns and produce simpler, easier-to-interpret models, and its ability to perform analysis on parameter relative importance. Previous applications of the MARS algorithm in civil engineering include predicting the doweled pavement performance, estimating shaft resistance of piles in sand and deformation of asphalt mixtures, analyzing shaking table tests of reinforced soil wall, determining the undrained shear strength of clay, predicting liquefaction-induced lateral spread, assessing the ultimate and serviceability performances of underground caverns, estimating the EPB tunnel induced ground surface settlement, and inverse analysis for braced excavation [1, 7, 8, 12, 13, 15–17, 19–23]. In this paper, the Back propagation neural network (BPNN) and MARS models are developed for pile drivability predictions in relation to the Maximum compressive stresses (MCS), Maximum tensile stresses (MTS), and Blow per foot (BPF). A database of more than four thousand piles is utilized for model development and comparative performance between BPNN and MARS predictions.

2 Methodologies

2.1 Back-Propagation Algorithm

A three-layer, feed-forward neural network topology shown in Fig. 1 is adopted in this study. As shown in Fig. 1, the back-propagation algorithm involves two phases of data flow. In the first phase, the input data are presented forward from the input to output layer and produces an actual output. In the second phase, the error between the target values and actual values are propagated backwards from the output layer to the previous layers and the connection weights are updated to reduce

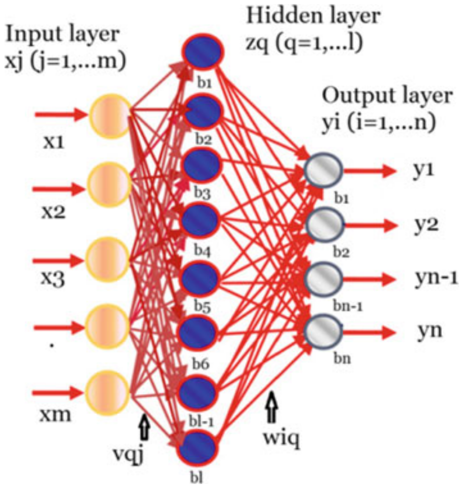


Fig. 1 Back-propagation neural network architecture used in this study

the errors between the actual output values and the target output values. No effort is made to keep track of the characteristics of the input and output variables. The network is first trained using the training data set. The objective of the network training is to map the inputs to the output by determining the optimal connection weights and biases through the back-propagation procedure. The number of hidden neurons is typically determined through a trial-and-error process; normally the smallest number of neurons that yields satisfactory results (judged by the network performance in terms of the coefficient of determination R^2 of the testing data set) is selected. In the present study, a Matlab-based back-propagation algorithm BPNN with the Levenberg-Marquardt (LM) algorithm [2] was adopted for neural network modeling.

2.2 Multivariate Adaptive Regression Splines Algorithm

MARS was first proposed by [3] as a flexible procedure to organize relationships between a set of input variables and the target dependent that are nearly additive or involve interactions with fewer variables. It is a nonparametric statistical method based on a divide and conquer strategy in which the training data sets are partitioned into separate piecewise linear segments (splines) of differing gradients (slope). MARS makes no assumptions about the underlying functional relationships between dependent and independent variables. In general, the splines are connected smoothly together, and these piecewise curves (polynomials), also known as basis functions (BFs), result in a flexible model that can handle both linear and nonlinear behavior. The connection/interface points between the pieces are called knots. Marking the end of one region of data and the beginning of another, the candidate knots are placed at random positions within the range of each input variable.

MARS generates BFs by stepwise searching over all possible univariate candidate knots and across interactions among all variables. An adaptive regression algorithm is adopted for automatically selecting the knot locations. The MARS algorithm involves a forward phase and a backward phase. The forward phase places candidate knots at random positions within the range of each predictor variable to define a pair of BFs. At each step, the model adapts the knot and its corresponding pair of BFs to give the maximum reduction in sum-of-squares residual error. This process of adding BFs continues until the maximum number is reached, which usually results in a very complicated and overfitted model. The backward phase involves deleting the redundant BFs that made the least contributions. An open MARS source code from [10] is adopted in performing the analyses presented in this paper.

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

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

in which e is the fitting error. f is the built MARS model, comprising of BFs which are splines piecewise polynomial functions. For simplicity, only the piecewise linear function is expressed and considered in this paper. Piecewise linear functions follow the form $\max(0, x-t)$ with a knot defined at value t. Expression $\max(\cdot)$ means that only the positive part of (.) is used otherwise it is assigned a zero value. Formally,

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

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

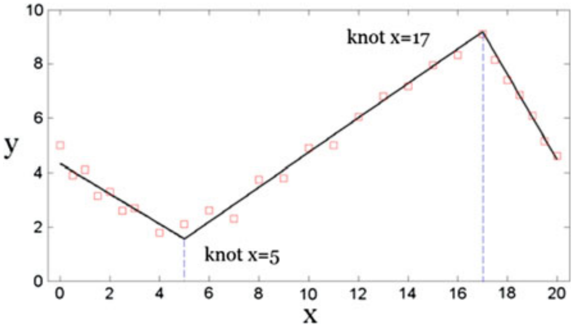
$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m \lambda_m(X) \tag{3}$$

where each λ_m is a BF. It can be a spline function, or interaction BFs produced by multiplying an existing term with a truncated linear function involving a new/different variable (higher orders can be used only when the data warrants it; for simplicity, at most second-order is adopted). The terms β are constant coefficients, estimated using the least-squares method.

Figure 2 shows an example illustration of how the MARS algorithm would make use of piecewise linear spline functions to fit provided data patterns. The MARS mathematical equation is as follows

$$y = -5.0875 - 2.7678 \times BF1 + 0.5540 \times BF2 + 1.1900 \times BF3 \tag{4}$$

Fig. 2 Knots and linear splines for a simple MARS example



in which BF1 = max(0, x – 17), BF2 = max(0, 17 – x) and BF3 = max(0, x – 5) and max is defined as: max(a, b) is equal to a if $a > b$, else b. The knots are located at $x = 5$ and 17. These two knots delimit/cut the x range into three intervals where different linear relationships are identified.

The MARS modeling is a data-driven process. To construct the model in Eq. (3), first the forward phase is performed on the training data starting initially with only the intercept β_0. At each subsequent step, the basis pair that produces the maximum reduction in the training error is added. Considering a current model with M basis functions, the next pair to be added to the model is in the form of

$$\beta_{M+1}\,\lambda_l(X)\max(0, X_j - t) + \beta_{M+2}\,\lambda_l(X)\max(0, t - X_j) \tag{5}$$

with each β being estimated by the least-squares method. This process of adding BFs continues until the model reaches some predetermined maximum number, generally leading to a purposely overfitted model.

The backward phase improves the model by removing the less significant terms until it finds the best sub-model. Model subsets are compared using the less computationally expensive method of Generalized Cross-Validation (GCV). The GCV is the mean-squared residual error divided by a penalty that is dependent on model complexity. For the training data with N observations, GCV is calculated as [9]

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

in which M is the number of BFs, d is a penalty for each basis function included in the developed sub-model, N is the number of data sets, and $f(x_i)$ denotes the MARS predicted values. Thus the numerator is the mean square error of the evaluated model in the training data, penalized by the denominator which 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 BFs but also the number of knots. A default value of 3 is assigned to penalizing parameter d and further suggestions on choosing the value of d can be referred to [3]. At each deletion step, a basis function is pruned to minimize Eq. (3), until an adequately fitting model is found.

After the optimal MARS model is determined, by grouping together all the BFs involving one variable and another grouping of BFs involving pairwise interactions, the analysis of variance (ANOVA) decomposition procedure [3] can be used to assess the parameter relative importance based on the contributions from the input variables and the BFs.

Table 1 Summary of performance measures

Performance measure	Definition
Coefficient of determination (R^2)	$R^2 = 1 - \frac{\frac{1}{\mathrm{n}}\sum_{i=1}^{n}(Y_i - \overline{Y})^2}{\frac{1}{\mathrm{n}}\sum_{i=1}^{n}(y_i - \overline{y})^2}$
Coefficient of correlation (r)	$r = \frac{\sum_{i=1}^{N}(Y_i - \overline{Y})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N}(Y_i - \overline{Y})^2}\sqrt{\sum\limits_{i=1}^{N}(y_i - \overline{y})^2}}$
Relative root mean squared error (RRMSE)	$RRMSE = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - y_i)^2}}{\frac{1}{N}\sum_{i=1}^{N} y_i} \times 100$
Performance index (ρ)	$\rho = \frac{RRMSE}{1 + r}$

$\overline{y}$ is the mean of the target values of y_i; $\overline{Y}$ is the mean of the predicted Y_i; N denotes the number of data points in the used set, training set, testing set or the overall set; Definitions of RRMSE, r and ρ are based on [4].

3 Performance Measures

Table 1 shows the performance measures and the corresponding definitions utilized for prediction comparison of the two surrogate methods.

4 Pile Drivability Data Sets

In this paper, a database containing 4072 piles with a total of seventeen variables is developed from the information on piles already installed for bridges in the State of North Carolina [11]. Seventeen variables including hammer characteristics, hammer cushion material, pile and soil parameters, ultimate pile capacities, and stroke were regarded as inputs to estimate the three dependent responses comprising of the Maximum compressive stresses (MCS), Maximum tensile stresses (MTS), and Blow per foot (BPF). A summary of the input variables and outputs is listed in Table 2.

For purpose of simplifying the analyses considering the extensive number of parameters and large data set, Joen and Rahman [11] divided the data into five categories (Q_1–Q_5) based on the ultimate pile capacity, as detailed in Table 3. In this paper, for each category 70% of the data patterns were randomly selected as the training dataset and the remaining data were used for testing. For details of the entire data set as well as each design variable and responses, the report by Joen and Rahman [11] can be referred to.

Table 2 Summary of input variables and outputs

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

Table 3 Division of data with respect to ultimate pile capacities

Pile type	Q_u range (kN)	Data		
		No. of training data	No. of testing data	Total
Q_1	133.4–355.9	270	90	360
Q_2	360.0–707.3	428	144	572
Q_3	707.4–1112.1	808	249	1057
Q_4	1112.2–1774.8	1296	421	1717
Q_5	1774.9–3113.7	276	90	366
Total		3078	994	4072

5 BPNN Models

For simplicity, only BPNN models with one single hidden layer structure are considered. The optimal BPNN model is selected from models with different hidden neurons since the other main parameters for BPNN algorithms have been fixed as:

logsig transfer function from the input layer to the hidden layer;
tansig transfer function from the hidden layer to the output layer;
maxepoch = 500;
learning rate = 0.01;
min_grad = 1 × 10^{-15};
decrease factor mu_dec = 0.7;
increase factor mu_inc = 1.03.

5.1 *The Optimal BPNN Model*

The BPNN with the highest coefficient of determination R^2 value for the testing data sets is considered to be the optimal model. Figure 3 plots the R^2 values of the testing data sets for BPNN models with different neurons (from 5 to 15) in the hidden layer for MCS, MTS and BPF predictions. It can be observed that for the optimal MCS, MTS, and BPF models, the number of the neurons in the hidden layer is 9, 7 and 11, respectively.

5.2 *Modeling Results*

Figures 4, 5 and 6 show the BPNN predictions for the training and testing data patterns for MCS, MTS, and BPF, respectively. For the MCS predictions, considerably high R^2 (>0.97) are obtained for both the training and testing patterns. Compared with the MCS predictions, the developed BPNN model is slightly less

Fig. 3 R^2 for different neuron numbers for MCS, MTS and BPF models

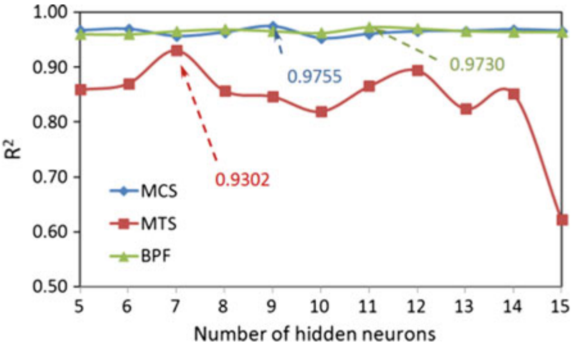


Fig. 4 Prediction of MCS using BPNN

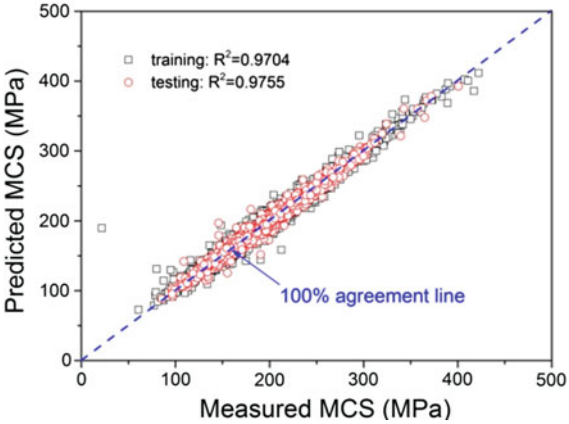


Fig. 5 Prediction of MTS using BPNN

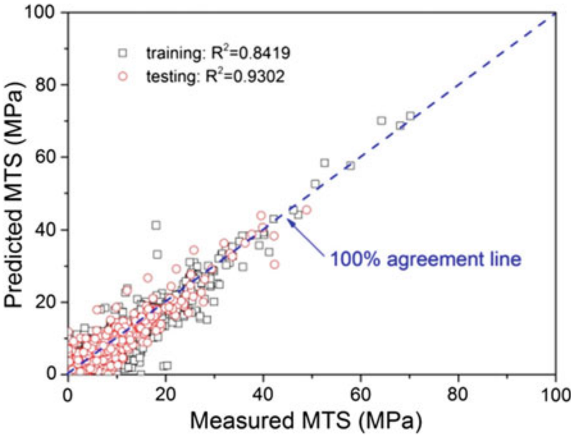
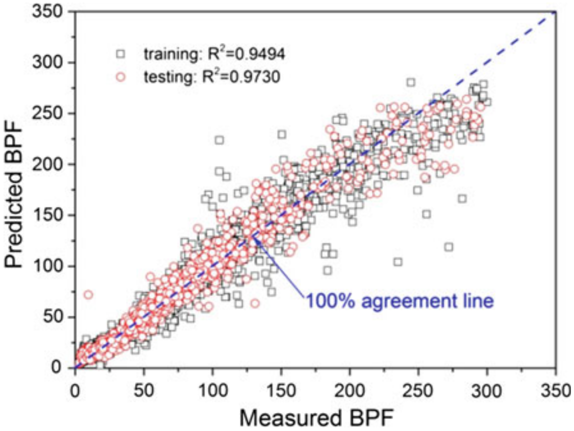


Fig. 6 Prediction of BPF using BPNN



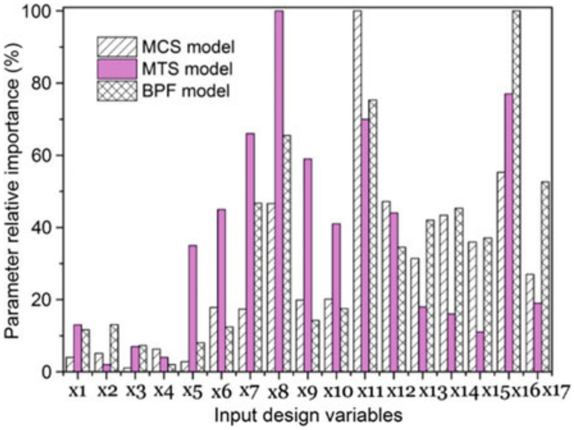


Fig. 7 Prediction of BPF using BPNN

accurate in predicting the MTS mainly as a result of the bias (errors) due to the significantly smaller tensile stress values in comparison to the compressive stresses. For the BPF estimation, high R^2 are also obtained for both the training and testing patterns, with the latter slightly greater than the training sets. In addition, the three optimal BPNN models can serve as reliable tools for prediction of MCS, MTS and BPF.

5.3 Parameter Relative Importance

The parameter relative importance determined by BPNN is based on the method by [5] and discussed by Goh [6]. Figure 7 gives the plot of the relative importance of the input variables for the three BPNN models. It can be observed that MCS is mostly influenced by the input variable x11 (Slenderness) and MTS is mostly influenced by the input variable x8 (Penetration). Interestingly, BPF is primarily influenced by the input variable x16 (Ultimate pile capacity).

5.4 Model Interpretability

For brevity, only the developed BPNN MCS model is expressed in mathematical form through the trained connections weights, the bias, and the transfer functions. The Mathematical expression for MCS obtained by the optimal MCS analysis is shown in the Appendix 1.

6 MARS Models

It is assumed that at most the 2nd order interaction is considered for the prediction of MCS, MTS and BPF using MARS. The number of basis functions changes from 2n to n^2 (n = 17 in this study, numerical trials indicate that overfitting occurs when the number of BFs exceeds 80).

6.1 *The Optimal MARS Model*

The MARS model with the highest R^2 value and less BFs for the testing data set is considered to be the optimal. Figure 8 plots the R^2 values of the testing data sets for the MARS models with different BFs (from 34 to 78) in the hidden layer for the MCS, MTS and BPF predictions. It can be observed that for the optimal MCS, MTS, and BPF models, the number of BFs is 52, 36 and 38, respectively.

6.2 *Modeling Results*

Figures 9, 10 and 11 show the MARS predictions for the training and testing data patterns for MCS, MTS, and BPF, respectively. For the MCS prediction, considerably high R^2 (>0.95) are obtained for both the training and testing patterns. As in the BPNN analysis, the developed MARS model is less accurate in predicting MTS compared with the MCS predictions, mainly due to the bias brought about by the smaller tensile stress values. For the BPF estimation, high R^2 (>0.90) are also obtained for both the training and testing patterns, with the latter slightly greater than the training sets. Consequently, the three optimal MARS models can serve as reliable tools for prediction of MCS, MTS and BPF.

Fig. 8 R^2 for different number of BFs for MCS, MTS and BPF models

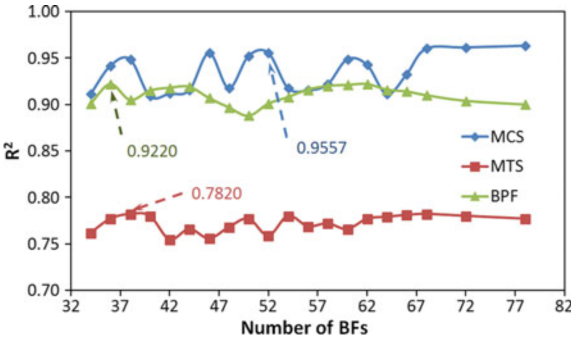


Fig. 9 Prediction of MCS using MARS

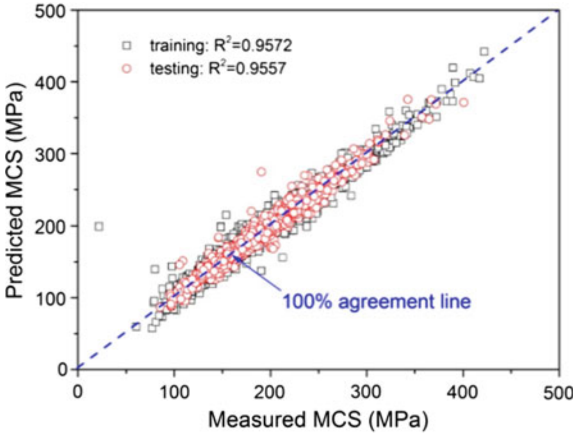


Fig. 10 Prediction of MTS using MARS

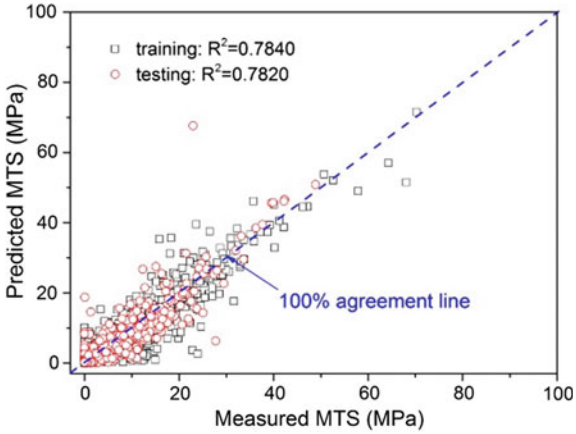
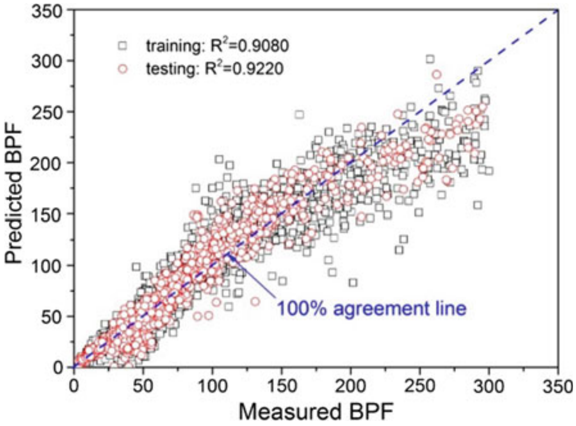


Fig. 11 Prediction of BPF using MARS



6.3 Parameter Relative Importance

Table 4 displays the ANOVA decomposition of the built MARS models for MCS, MTS and BPF respectively. For each model, the ANOVA functions are listed. The GCV column provides an indication on the significance of the corresponding ANOVA function, by listing the GCV value for a model with all BFs corresponding to that particular ANOVA function removed. It is this GCV score that is used to assess whether the ANOVA function is making a significant contribution to the model, or whether it just marginally improves the global GCV score. The #basis column gives the number of BFs comprising the ANOVA function and the variable (s) column lists the input variables associated with this ANOVA function.

Table 4 ANOVA decomposition of MARS model for MCS, MTS and BPF

Function	MCS			MTS			BPF		
	GCV	# Basis	Variable (s)	GCV	# Basis	Variable (s)	GCV	# Basis	Variable (s)
1	28.82	1	1	1.047	2	5	39.657	2	1
2	8.346	2	6	575.191	1	6	9.750	2	2
3	7.073	1	8	109.688	2	7	1.760	2	13
4	10.226	1	12	305.352	1	8	3.005	2	15
5	5.629	3	17	251.585	2	11	8.034	2	16
6	11.184	1	1 3	25.373	1	17	2.976	2	17
7	48.344	2	1 17	0.441	1	1 6	66.894	3	1 3
8	8.048	5	2 4	337.341	2	3 7	0.370	2	1 6
9	11.846	2	3 4	0.893	2	3 17	0.235	2	1 13
10	21.733	2	3 17	5.626	2	5 7	0.231	1	1 16
11	63.062	1	4 15	2.229	1	5 11	43.396	2	2 3
12	8.017	1	6 8	795.122	4	6 7	0.357	1	2 4
13	4.976	3	8 17	92.069	3	6 8	0.403	2	2 16
14				6.797	1	6 9	0.557	4	2 17
15				48.170	4	6 11	0.280	2	3 13
16				1.472	1	6 16	0.705	2	4 15
17				2.593	2	6 17	0.227	1	4 17
18				0.626	1	7 8	0.170	1	5 13
19				11.173	1	7 17	0.191	1	6 15
20				0.447	1	8 16	0.221	2	7 15
21				50.089	2	8 17	0.375	1	13 15
22				0.828	1	11 15	0.984	1	16 17
23				148.475	2	11 17			
24				1.472	2	14 17			
25				0.466	1	15 17			

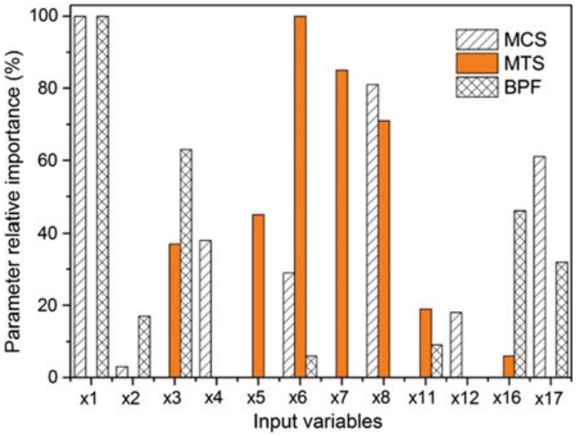


Fig. 12 Relative importance of the input variables in MARS pile drivability models

Figure 12 gives the plot of the relative importance of the input variables for the three HP drivability models developed by MARS. It can be observed that both MCS and BPF are mostly influenced by the input variable x1 (hammer weight). Interestingly, MTS is primarily influenced by the input variable x6 (the weight of helmet). It should be noted that since the BPNN and MARS algorithms adopt different methods in assessing the parametric relative importance, it is understandable that the two algorithms give different results.

6.4 Model Interpretability

Table 5 lists the BFs of the MCS model. The MARS model is in the form of

$$\begin{aligned}
&MCS(MPa) = 169.4 + 0.0095 \times BF1 + 35.6 \times BF2 - 47.5 \times BF3 - 0.46 \times BF4 - 2 \times BF5 + \\
&8847 \times BF6 + 9.2 \times BF7 - 8.2 \times BF8 - 0.0025 \times BF9 + 0.0062 \times BF10 - 3.2 \times BF11 + \\
&470 \times BF12 - 0.0036 \times BF13 - 0.8 \times BF14 - 0.0012 \times BF15 + 0.006 \times BF16 + 9.43 \times BF17 \\
&- 6.1 \times BF18 + 0.136 \times BF19 - 0.098 \times BF20 - 0.83 \times BF21 - 0.17 \times BF22 - 540 \times BF23 \\
&+ 1.34 \times 10^5 \times BF24 + 1.672 \times BF25 - 0.42 \times BF26 + 0.144 \times BF27 - 4.57 \times BF28 \\
&- 0.0054 \times BF29 + 0.052 \times BF30 + 87 \times BF31 + 250 \times BF32 - 763 \times BF33 - 16 \times BF34 \\
&- 28.1 \times BF35 + 0.217 \times BF36 - 0.2 \times BF37 + 34.5 \times BF38 + 31.3 \times BF39 - 50.2 \times BF40 \\
&- 425 \times BF41 + 0.0018 \times BF42 - 0.003 \times BF43 - 7.4 \times BF44 + 341 \times BF45 + 51.4 \times BF46 \\
&+ 5.67 \times BF47 + 12 \times BF48 + 0.96 \times BF49 + 100.2 \times BF50 - 0.2 \times BF51 + 0.23 \times BF52
\end{aligned} \tag{7}$$

Table 5 BFs and corresponding equations of MARS MCS model

BF	Equation	BF	Equation
BF1	max(0, x16–1550)	BF27	max(0, x15 − 15)
BF2	max(0, x17−2.29)	BF28	max(0, 15−x15)
BF3	max(0, 2.29−x17)	BF29	BF28 × max(0, x16−289)
BF4	max(0, x6−7.38)	BF30	BF28 × max(0, 289−x16)
BF5	max(0, 7.38−x6)	BF31	BF2 × max(0, x1−29.4)
BF6	max(0, 0.014−x10)	BF32	BF6 × max(0, x6−6.67)
BF7	max(0, x2−30.7)	BF33	BF6 × max(0, 6.67−x6)
BF8	max(0, 30.7−x2)	BF34	BF5 × max(0, 1.81−x17)
BF9	BF1 × max(0, x7−8.00)	BF35	BF3 × max(0, x1−29.4)
BF10	BF1 × max(0, 8.00−x7)	BF36	BF7 × max(0, x11−50)
BF11	max(0, x11−9)	BF37	BF7 × max(0, 50−x11)
BF12	max(0, 9−x11)	BF38	BF28 × max(0, x13−0.59)
BF13	max(0, 1550−x16) × max(0, x8−3.05)	BF39	BF28 × max(0, 0.59−x13)
BF14	max(0, 1550−x16) × max(0, 3.05−x8)	BF40	BF4 × max(0, x5−0.05)
BF15	max(0, 1550−x16) × max(0, x6−9.34)	BF41	BF4 × max(0, 0.05−x5)
BF16	max(0, 1550−x16) × max(0, 9.34−x6)	BF42	max(0, 1550−x16) × max(0, x11−24)
BF17	BF6 × max(0, x16−1067.5)	BF43	max(0, 1550−x16) × max(0, 24−x11)
BF18	BF6 × max(0, 1068−x16)	BF44	BF7 × max(0, 0.18−x3)
BF19	BF11 × max(0, x4−3.24)	BF45	max(0, x3−0.26)
BF20	BF11 × max(0, 3.24−x4)	BF46	max(0, 0.26−x3)
BF21	BF11 × max(0, x1−29.4)	BF47	BF5 × max(0, x4−1.97)
BF22	BF11 × max(0, 29.4−x1)	BF48	BF5 × max(0, 1.97−x4)
BF23	BF6 × max(0, x7−3.05)	BF49	BF5 × max(0, 44.7−x2)
BF24	BF6 × max(0, 3.05−x7)	BF50	BF45 × max(0, 30−x11)
BF25	BF7 × max(0, x17−2.90)	BF51	BF11 × max(0, x2−54.2)
BF26	BF7 × max(0, 2.90−x17)	BF52	BF11 × max(0, 54.2−x2)

7 Discussions

Comparisons of R^2, r, RRMSE and ρ, as well as the built model interpretability between MARS and BPNN are shown in Table 6. It can be observed that generally BPNN models are slightly more accurate than MARS. However, in terms of the model interpretability, MARS outperforms BPNN through easy-to-interpret model. Thus, both these two methods can actually be used for cross-validation.

Table 6 Comparison of performance measures for BPNN and MARS

Performance measures		MCS	BPNN	BPF	MCS	MARS	BPF
			MTS			MTS	
R^2	Training	0.9704	0.8419	0.9494	0.9572	0.7840	0.9080
	Testing	0.9755	0.9302	0.9730	0.9557	0.7820	0.9220
r	Training	0.9852	0.8931	0.9742	0.9782	0.8855	0.9534
	Testing	0.9874	0.9414	0.9762	0.9784	0.8945	0.9604
RRMSE (%)	Training	4.2382	80.828	18.487	5.0764	83.604	24.731
	Testing	3.7621	62.364	18.543	4.8102	73.222	23.464
ρ (%)	Training	2.1357	42.688	9.3624	2.5663	44.342	12.672
	Testing	1.8945	32.468	9.3884	2.4321	38.651	11.975
Model interpretability		Poor, as shown in Appendix 1			Good, Eq. (7) and Table 5		

8 Summary and Conclusions

A database containing 4072 pile data sets with a total of 17 variables is adopted to develop the BPNN and MARS models for drivability predictions. Performance measures indicate that both the BPNN and MARS models for the analyses of pile drivability provide similar predictions and can thus be used for predicting pile drivability as cross-validation. In addition, the MARS algorithm builds flexible models using simpler linear regression and data-driven stepwise searching, adding and pruning. The developed MARS models are much easier to be interpreted.

Acknowledgements The authors are grateful to the support by the National Natural Science Foundation of China (No. 51608071) and the Advanced Interdisciplinary Special Cultivation program (No. 106112017CDJQJ208850).

Appendix 1

Calculation of BPNN Output MCS Model

The transfer functions used MCS are ‘*logsig*’ transfer function for hidden layer to output layer and ‘*tansig*’ transfer function for output layer to target. The calculation process of BPNN output for MCS is elaborated in detail as follows:

From connection weights for a trained NN, it is possible to develop a mathematical equation relating input parameters and the single output parameter Y using

$$Y = f_{sig}\left\{ b_0 + \sum_{k=1}^{h}\left[w_k f_{sig}\left(b_{hk} + \sum_{i=1}^{m} w_{ik} X_i \right)\right]\right\} \tag{8}$$

in which b_0 is the bias at the output layer, ω_k is the weight connection between neuron k of the hidden layer and the single output neuron, b_{hk} is the bias at neuron k of the hidden layer ($k = 1, h$), ω_{ik} is the weight connection between input variable i ($i = 1, m$) and neuron k of the hidden layer, xxx is the input parameter i, and f_{sig} is the sigmoid (logsig & *tansig*) transfer function.

Using the connection weights of the trained neural network, the following steps can be followed to mathematically express the BPNN model:

Step 1: Normalize the input values for x_1, x_2,… and x_{17} linearly using $X_{norm} = 2(\mathrm{x}_{\mathrm{actual}} - \mathrm{x}_{\min})/(\mathrm{x}_{\max} - \mathrm{x}_{\min}) - 1$

Let the actual $x_1 = X_{1\mathrm{a}}$ and the normalized $x_1 = X_1$

$$X_1 = -1 + 2 \times (X_{1a} - 7.8)/(31.1 - 7.8) \tag{9}$$

Let the actual $x_2 = X_{2\mathrm{a}}$ and the normalized $x_2 = X_2$

$$X_2 = -1 + 2 \times (X_{2a} - 23.9)/(102.3 - 23.9) \tag{10}$$

Let the actual $x_3 = X_{3\mathrm{a}}$ and the normalized $x_3 = X_3$

$$X_3 = -1 + 2 \times (X_{3a} - 0.15)/(0.27 - 0.15) \tag{11}$$

Let the actual $x_4 = X_{4\mathrm{a}}$ and the normalized $x_4 = X_4$

$$X_4 = -1 + 2 \times (X_{4a} - 1.21)/(3.72 - 1.21) \tag{12}$$

Let the actual $x_5 = X_{5\mathrm{a}}$ and the normalized $x_5 = X_5$

$$X_5 = -1 + 2 \times (X_{5a} - 0.0)/(0.18 - 0.0) \tag{13}$$

Let the actual $x_6 = X_{6\mathrm{a}}$ and the normalized $x_6 = X_6$

$$X_6 = -1 + 2 \times (X_{6a} - 4.0)/(34.5 - 4.0) \tag{14}$$

Let the actual $x_7 = X_{7\mathrm{a}}$ and the normalized $x_7 = X_7$

$$X_7 = -1 + 2 \times (X_{7a} - 3.0)/(30.5 - 3.0) \tag{15}$$

Let the actual $x_8 = X_{8\mathrm{a}}$ and the normalized $x_8 = X_8$

$$X_8 = -1 + 2 \times (X_{8a} - 3.0)/(30.5 - 3.0) \tag{16}$$

Let the actual $x_9 = X_{9\mathrm{a}}$ and the normalized $x_9 = X_9$

$$X_9 = -1 + 2 \times (X_{9a} - 0.30)/(0.36 - 0.30) \tag{17}$$

Let the actual $x_{10} = X_{10\mathrm{a}}$ and the normalized $x_{10} = X_{10}$

$$X_{10} = -1 + 2 \times (X_{10a} - 0.010)/(0.014 - 0.010) \tag{18}$$

Let the actual $x_{11} = X_{11\mathrm{a}}$ and the normalized $x_{11} = X_{11}$

$$X_{11} = -1 + 2 \times (X_{11a} - 8.4)/(100.1 - 8.4) \tag{19}$$

Let the actual $x_{12} = X_{12\mathrm{a}}$ and the normalized $x_{12} = X_{12}$

$$X_{12} = -1 + 2 \times (X_{12a} - 0.0025)/(0.0084 - 0.0025) \tag{20}$$

Let the actual $x_{13} = X_{13\mathrm{a}}$ and the normalized $x_{13} = X_{13}$

$$X_{13} = -1 + 2 \times (X_{13a} - 0.16)/(0.83 - 0.16) \tag{21}$$

Let the actual $x_{14} = X_{14\mathrm{a}}$ and the normalized $x_{14} = X_{14}$

$$X_{14} = -1 + 2 \times (X_{14a} - 0.20)/(0.66 - 0.20) \tag{22}$$

Let the actual $x_{15} = X_{15\mathrm{a}}$ and the normalized $x_{15} = X_{15}$

$$X_{15} = -1 + 2 \times (X_{15a} - 10)/(95 - 10) \tag{23}$$

Let the actual $x_{16} = X_{16\mathrm{a}}$ and the normalized $x_{16} = X_{16}$

$$X_{16} = -1 + 2 \times (X_{16a} - 137.9)/(2891.2 - 137.9) \tag{24}$$

Let the actual $x_{17} = X_{17\mathrm{a}}$ and the normalized $x_{17} = X_{17\mathrm{a}}$

$$X_{17} = -1 + 2 \times (X_{17a} - 1.02)/(3.46 - 1.02) \tag{25}$$

Step 2: Calculate the normalized value (Y_1) using the following expressions:

$$\begin{aligned}A_1 = & -21.3261 + 6.2318 logsig(X_1) - 3.1654 logsig(X_2) + 17.1602 logsig(X_3)\\ & - 1.459 logsig(X_4) - 4.3521 logsig(X_5) + 13.216 logsig(X_6) - 9.768 logsig(X_7)\\ & + 3.5715 logsig(X_8) - 0.1209 logsig(X_9) + 0.2208 logsig(X_{10}) + 15.9897 logsig(X_{11})\\ & - 9.4443 logsig(X_{12}) - 1.9824\, logsig(X_{13}) - 5.5972 logsig(X_{14}) - 6.3521 logsig(X_{15})\\ & + 5.4767 logsig(X_{16}) - 0.7102 logsig(X_{17})\end{aligned} \tag{26}$$

$$\begin{aligned}A_2 = & -8.4258 - 15.0031 logsig(X_1) + 11.7647 logsig(X_2) + 1.1075 logsig(X_3)\\ & - 5.1013 logsig(X_4) + 6.9054 logsig(X_5) - 10.1146 logsig(X_6) - 5.4258 logsig(X_7)\\ & - 23.0086 logsig(X_8) + 0.5226 logsig(X_9) + 0.4659 logsig(X_{10}) + 4.8115 logsig(X_{11})\\ & + 8.377 logsig(X_{12}) + 18.6713 logsig(X_{13}) - 13.0335 logsig(X_{14}) + 15.2353 logsig(X_{15})\\ & - 12.7608 logsig(X_{16}) + 2.239 logsig(X_{17})\end{aligned} \tag{27}$$

$$\begin{aligned}A_3 = & -7.9671 - 21.6681\mathit{logsig}(X_1) + 6.0201\mathit{logsig}(X_2) + 6.4033\mathit{logsig}(X_3)\\ & - 0.5677\mathit{logsig}(X_4) + 21.532\mathit{logsig}(X_5) + 11.305\mathit{logsig}(X_6) - 21.4426\mathit{logsig}(X_7)\\ & + 24.4447\mathit{logsig}(X_8) - 1.7981\mathit{logsig}(X_9) - 1.526\mathit{logsig}(X_{10}) - 6.618\mathit{logsig}(X_{11})\\ & - 32.874\mathit{logsig}(X_{12}) + 3.4611\mathit{logsig}(X_{13}) - 5.9862\mathit{logsig}(X_{14}) + 9.1232\mathit{logsig}(X_{15})\\ & - 15.876\mathit{logsig}(X_{16}) - 1.1918\mathit{logsig}(X_{17})\end{aligned} \tag{28}$$

$$\begin{aligned}A_4 = & -0.8699 + 3.7546\mathit{logsig}(X_1) - 2.2402\mathit{logsig}(X_2) - 1.2905\mathit{logsig}(X_3)\\ & + 0.2448\mathit{logsig}(X_4) - 0.1977\mathit{logsig}(X_5) + 0.0614\mathit{logsig}(X_6) + 1.206\mathit{logsig}(X_7)\\ & - 0.6279\mathit{logsig}(X_8) - 2.19\mathit{logsig}(X_9) + 2.1303\mathit{logsig}(X_{10}) - 0.3518\mathit{logsig}(X_{11})\\ & - 0.4643\mathit{logsig}(X_{12}) + 1.0234\mathit{logsig}(X_{13}) - 0.1317\mathit{logsig}(X_{14}) + 0.1105\mathit{logsig}(X_{15})\\ & - 0.2714\mathit{logsig}(X_{16}) - 0.3666\mathit{logsig}(X_{17})\end{aligned} \tag{29}$$

$$\begin{aligned}A_5 = & -1.8394 + 0.5777\mathit{logsig}(X_1) + 0.0039\mathit{logsig}(X_2) - 0.1447\mathit{logsig}(X_3)\\ & - 0.0038\mathit{logsig}(X_4) + 0.1173\mathit{logsig}(X_5) - 0.2946\mathit{logsig}(X_6) - 1.2601\mathit{logsig}(X_7)\\ & + 3.2074\mathit{logsig}(X_8) - 0.496\mathit{logsig}(X_9) + 0.4693\mathit{logsig}(X_{10}) - 2.415\mathit{logsig}(X_{11})\\ & + 0.2184\mathit{logsig}(X_{12}) + 0.5232\mathit{logsig}(X_{13}) + 0.1453\mathit{logsig}(X_{14}) + 0.1121\mathit{logsig}(X_{15})\\ & - 0.1928\mathit{logsig}(X_{16}) + 0.2347\mathit{logsig}(X_{17})\end{aligned} \tag{30}$$

$$\begin{aligned}A_6 = & -10.0517 - 1.6316\mathit{logsig}(X_1) - 9.722\mathit{logsig}(X_2) - 0.7598\mathit{logsig}(X_3)\\ & - 0.6052\mathit{logsig}(X_4) + 6.2292\mathit{logsig}(X_5) - 15.005\mathit{logsig}(X_6) + 13.487\mathit{logsig}(X_7)\\ & - 14.773\mathit{logsig}(X_8) + 0.2152\mathit{logsig}(X_9) + 0.1029\mathit{logsig}(X_{10}) - 7.4019\mathit{logsig}(X_{11})\\ & - 9.094\mathit{logsig}(X_{12}) + 0.7162\mathit{logsig}(X_{13}) + 3.5733\mathit{logsig}(X_{14}) + 5.7949\mathit{logsig}(X_{15})\\ & - 9.2971\mathit{logsig}(X_{16}) + 0.3107\mathit{logsig}(X_{17})\end{aligned} \tag{31}$$

$$\begin{aligned}A_7 = & -19.748 + 17.4566\mathit{logsig}(X_1) - 6.3092\mathit{logsig}(X_2) - 11.431\mathit{logsig}(X_3)\\ & + 2.612\mathit{logsig}(X_4) - 10.9304\mathit{logsig}(X_5) + 3.7079\mathit{logsig}(X_6) - 6.078\mathit{logsig}(X_7)\\ & - 11.0792\mathit{logsig}(X_8) - 0.0797\mathit{logsig}(X_9) + 2.5652\mathit{logsig}(X_{10}) - 13.752\mathit{logsig}(X_{11})\\ & + 21.45\mathit{logsig}(X_{12}) + 5.628\mathit{logsig}(X_{13}) + 4.9272\mathit{logsig}(X_{14}) + 0.3388\mathit{logsig}(X_{15})\\ & - 10.0783\mathit{logsig}(X_{16}) + 11.6775\mathit{logsig}(X_{17})\end{aligned} \tag{32}$$

$$\begin{aligned}A_8 = & -0.8092 + 0.8928\mathit{logsig}(X_1) + 0.2205\mathit{logsig}(X_2) - 0.1265\mathit{logsig}(X_3)\\ & - 0.0339\mathit{logsig}(X_4) + 0.2392\mathit{logsig}(X_5) - 0.5454\mathit{logsig}(X_6) - 2.6061\mathit{logsig}(X_7)\\ & + 7.1711\mathit{logsig}(X_8) - 1.018\mathit{logsig}(X_9) + 0.8755\mathit{logsig}(X_{10}) - 5.613\mathit{logsig}(X_{11})\\ & + 0.8703\mathit{logsig}(X_{12}) + 0.9022\mathit{logsig}(X_{13}) + 0.3572\mathit{logsig}(X_{14}) + 0.202\mathit{logsig}(X_{15})\\ & - 0.3748\mathit{logsig}(X_{16}) + 0.2867\mathit{logsig}(X_{17})\end{aligned} \tag{33}$$

$$A_9 = -1.3878 - 1.4052 logsig(X_1) + 0.6705 logsig(X_2) - 0.5285 logsig(X_3) - 0.0814 logsig(X_4) + 0.668 logsig(X_5) + 1.052 logsig(X_6) - 4.0946 logsig(X_7) + 3.5151 logsig(X_8) - 0.4901 logsig(X_9) + 1.104 logsig(X_{10}) + 0.3529 logsig(X_{11}) 0.9261 logsig(X_{12}) + 1.442 logsig(X_{13}) - 0.0338 logsig(X_{14}) + 0.2956 logsig(X_{15}) - 0.365 logsig(X_{16}) - 1.0274 logsig(X_{17}) \tag{34}$$

$$B_1 = -1.9078 \times tanh(A_1) \tag{35}$$

$$B_2 = -0.2020 \times tanh(A_2) \tag{36}$$

$$B_3 = 0.5773 \times tanh(A_3) \tag{37}$$

$$B_4 = -1.8211 \times tanh(A_4) \tag{38}$$

$$B_5 = 59.5399 \times tanh(A_5) \tag{39}$$

$$B_6 = -1.8844 \times tanh(A_6) \tag{40}$$

$$B_7 = -17.4645 \times tanh(A_7) \tag{41}$$

$$B_8 = -17.3515 \times tanh(A_8) \tag{42}$$

$$B_9 = -2.1634 \times tanh(A_9) \tag{43}$$

$$C_1 = -1.2957 + B_1 + B_2 + B_3 + B_4 + B_5 + B_6 + B_7 + B_8 + B_9 + B_{10} + B_{11} \tag{44}$$

$$Y_1 = C_1 \tag{45}$$

Step 3: De-normalize the output to obtain MCS

$$MCS = 21.93 + (422.18 - 21.93) \times (Y_1 + 1)/2 \tag{46}$$

Note: $logsig(x) = 1/(1 + exp(-x))$while $tanh(x) = 2/(1 + exp(-2x)) - 1$

References

1. Attoh-Okine, N. O., Cooger, K., & Mensah, S. (2009). Multivariate adaptive regression (MARS) and hinged hyperplanes (HHP) for doweled pavement performance modeling. *Journal of Construction and Building Materials, 23,* 3020–3023.
2. Demuth, H., & Beale, M. (2003). *Neural network toolbox for MATLAB-user guide version 4.1.* The Math Works Inc.
3. Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics, 19,* 1–141.

4. Gandomi, A.H., Roke, D.A. (2013). Intelligent formulation of structural engineering systems. In *Seventh MIT Conference on Computational Fluid and Solid Mechanics- Focus: Multiphysics & Multiscale*, 12–14 June, Cambridge, USA.
5. Garson, G. D. (1991). Interpreting neural-network connection weights. *AI Expert, 6*(7), 47–51.
6. Goh, A. T. C. (1994). Seismic liquefaction potential assessed by neural networks. *Journal of Geotechnical Engineering, ASCE, 120*(9), 1467–1480.
7. Goh, A. T. C., & Zhang, W. G. (2014). An improvement to MLR model for predicting liquefaction-induced lateral spread using multivariate adaptive regression splines. *Engineering Geology, 170,* 1–10.
8. Goh, A. T. C., Zhang, W. G., Zhang, Y. M., Xiao, Y., & Xiang, Y. Z. (2016). Determination of EPB tunnel-related maximum surface settlement: A Multivariate adaptive regression splines approach. *Bulletin of Engineering Geology and the Environment*. https://doi.org/10.1007/s10064-016-0937-8.
9. Hastie, T., Tibshirani, R., Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction*, 2nd ed., Springer.
10. Jekabsons, G. (2010). *VariReg: A software tool for regression modelling using various modeling methods*. Riga Technical University. http://www.cs.rtu.lv/jekabsons/.
11. Jeon, J. K., Rahman, M. S. (2008). *Fuzzy neural network models for geotechnical problems*. Research Project FHWA/ NC/ 2006–52. North Carolina State University, Raleigh, N.C.
12. Lashkari, A. (2012). Prediction of the shaft resistance of non-displacement piles in sand. *International Journal for Numerical and Analytical Methods in Geomechanics, 37,* 904–931.
13. Mirzahosseini, M., Aghaeifar, A., Alavi, A., Gandomi, A., & Seyednour, R. (2011). Permanent deformation analysis of asphalt mixtures using soft computing techniques. *Expert Systems with Applications, 38*(5), 6081–6100.
14. Rumelhart, D. E., Hinton, G. E., Williams, R. J. (1986. Learning internal representation by error propagation. In Parallel distributed processing, Rumelhart DE, McClelland JL (eds). MIT Press, (pp. 318–362) Cambridge, vol. 1.
15. Samui, P. (2011). Determination of ultimate capacity of driven piles in cohesionless soil: A multivariate adaptive regression spline approach. *International Journal for Numerical and Analytical Methods in Geomechanics, 36,* 1434–1439.
16. Samui, P., Das, S., & Kim, D. (2011). Uplift capacity of suction caisson in clay using multivariate adaptive regression splines. *Ocean Engineering, 38*(17–18), 2123–2127.
17. Samui, P., & Karup, P. (2011). Multivariate adaptive regression splines and least square support vector machine for prediction of undrained shear strength of clay. *Applied Metaheuristic Computing, 3*(2), 33–42.
18. Smith, E. A. L. (1960). Pile driving analysis by the wave equation. *Journal of the Engineering Mechanics Division ASCE, 86,* 35–61.
19. Zarnani, S., El-Emam, M., & Bathurst, R. J. (2011). Comparison of numerical and analytical solutions for reinforced soil wall shaking table tests. *Geomechanics & Engineering, 3*(4), 291–321.
20. Zhang, W. G., & Goh, A. T. C. (2013). Multivariate adaptive regression splines for analysis of geotechnical engineering systems. *Computers and Geotechnics, 48,* 82–95.
21. Zhang, W. G., & Goh, A. T. C. (2014). Multivariate adaptive regression splines model for reliability assessment of serviceability limit state of twin caverns. *Geomechanics and Engineering, 7*(4), 431–458.
22. Zhang, W. G., & Goh, A. T. C. (2017). Reliability assessment of ultimate limit state of twin cavern. *Geomechanics and Geoengineering, 12*(1), 48–59.
23. Zhang, W. G., Zhang, Y. M., & Goh, A. T. C. (2017). Multivariate adaptive regression splines for inverse analysis of soil and wall properties in braced excavation. *Tunneling and Underground Space Technology, 64,* 24–33.

Author Biographies

Zhang Wengang is Professor of Geotechnical Research Institute in the School of Civil Engineering, Chongqing University (CQU), China. He received his Ph.D. degree from the School of Civil and Environmental Engineering at Nanyang Technological University (NTU) Singapore and his B.Eng and M.Eng degrees from Hohai University (HHU) China. Before his career in CQU, he worked as a postdoctoral Research Fellow in NTU for more than two years. He received the Hulme Best Paper Award from Tunneling and Underground Construction Society Singapore (TUCSS) in 2013. Dr Zhang was invited in 2017 to act as the Lead Guest Editor for the Journal Geoscience Frontiers in view of his previously well-received paper in that journal. He joined CQU in 2016 as the Hundredth Young Professor and later in the same year was given the prestigious China Thousand Youth Talents plan award. His current research interests include: Big data geotechnical analysis, Braced excavation and deformation analysis and Rainfall induced slope deformation and instability.

Goh Anthony Teck Chee is Associate Professor in the School of Civil and Environmental Engineering (CEE) at Nanyang Technological University (NTU) Singapore. Prof Goh received both his Ph.D. and B.Eng. in Monash University Australia. Dr Goh is a registered Professional Engineer in Singapore. His teaching, research and professional practice have covered many aspects of geotechnical engineering including soft computing, finite element analysis, earth retaining structures, pile foundations, and slope stability. He is currently a Technical Committee member on Urban Geoengineering for the ISSMGE. Since 1993, Dr Goh has been at the forefront of research into the application of artificial intelligence (AI) methodologies in geotechnical engineering.