Geoscience Frontiers 7 (2016) 45–52

HOSTED BY

Contents lists available at ScienceDirect

China University of Geosciences (Beijing)

Geoscience Frontiers

journal homepage: www.elsevier.com/locate/gsf

Research paper

Multivariate adaptive regression splines and neural network models for prediction of pile drivability

Wengang Zhang, Anthony T.C. Goh*

School of Civil and Environmental Engineering, Nanyang Technological University, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 19 June 2014
Received in revised form
15 October 2014
Accepted 26 October 2014
Available online 13 November 2014

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

ABSTRACT

Piles are long, slender structural elements 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 solution with regard to the phenomena involved. In situations where measured data or numerical hypothetical results are available, neural networks stand out in mapping the nonlinear interactions and relationships between the system's predictors and dependent responses. In addition, unlike most computational tools, no mathematical relationship assumption between the dependent and independent variables has to be made. Nevertheless, neural networks have been criticized for their long trial-and-error training process since the optimal configuration is not known a priori. This paper investigates the use of a fairly simple nonparametric regression algorithm known as multivariate adaptive regression splines (MARS), as an alternative to neural networks, to approximate the relationship between the inputs and dependent response, and to mathematically interpret the relationship between the various parameters. In this paper, the Back propagation neural network (BPNN) and MARS models are developed for assessing pile drivability in relation to the prediction of 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.

1. Introduction

Piles are long, slender structural elements used to transfer the loads from the superstructure through weak strata onto stiffer soils or rocks. The selection of the type of pile depends on the type of structure, the ground conditions, the durability (e.g., to corrosion) and the installation costs. 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. One common method of calculating driving stresses is based on the stress-wave theory (Smith, 1960) which involves the discrete idealization of the hammer-pile-soil system. As the conditions at each site is 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: (i) hammer stroke vs. blow per foot (BPF) (1/set) for required bearing capacity; (ii) maximum compressive stresses vs. BPF; (iii) maximum tension stress vs. 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.

* Corresponding author. Tel.: +65 6790 5271; fax: +65 6792 0676.
E-mail address: ctcgoh@ntu.edu.sg (A.T.C. Goh).
Peer-review under responsibility of China University of Geosciences (Beijing).

http://dx.doi.org/10.1016/j.gsf.2014.10.003

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 (Rumelhart et al., 1986) 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 multivariate adaptive regression splines (MARS) (Friedman, 1991) 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, and assessing the ultimate and serviceability performances of underground caverns (Attoh-Okine et al., 2009; Mirzahosseini et al., 2011; Samui, 2011; Samui and Karup, 2011; Samui et al., 2011; Zarnani et al., 2011; Lashkari, 2012; Zhang and Goh, 2013, 2014a, b; Goh and Zhang, 2014). 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. Neural network 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 produce an actual output. In the second phase, the errors 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 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 (Demuth and Beale, 2003) was adopted for neural network modeling.

3. MARS algorithm

MARS was first proposed by Friedman (1991) 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

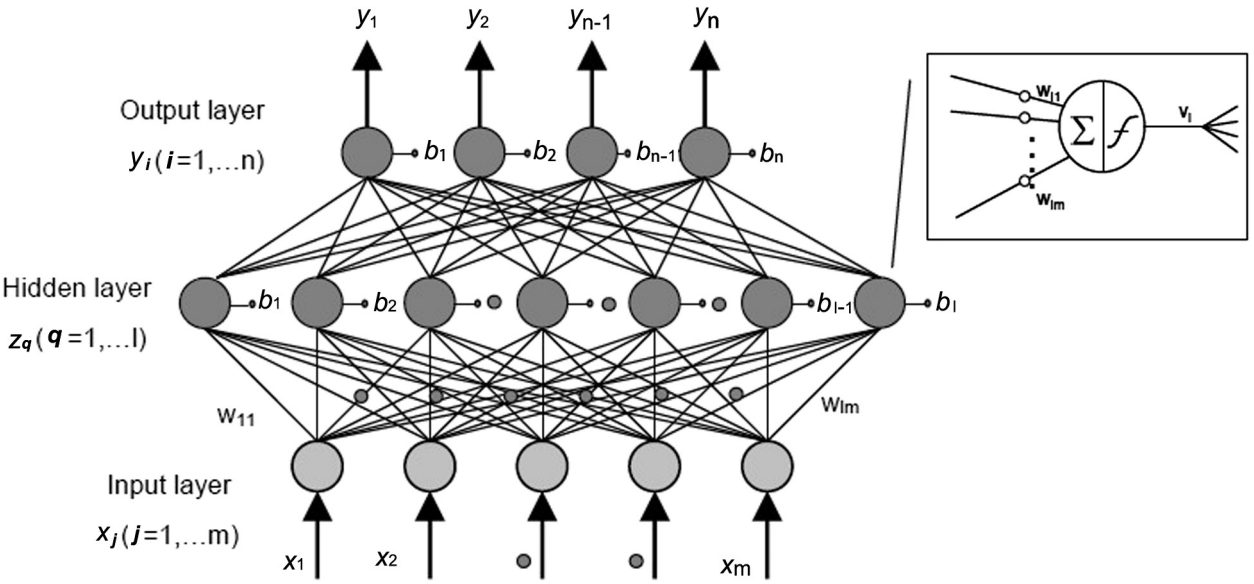

Figure 1. Back-propagation neural network architecture used in this study.

another, the candidate knots are placed at random positions within the range of each input variable.

MARS generates BFs by stepwise searching overall 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 Jekabsons (2010) is adopted in performing the analyses presented in this paper.

Let y be the target dependent responses and $X = (X_1, \ldots, X_P)$ 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:

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

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(.) means that only the positive part of (.) is used otherwise it is assigned a zero value. Formally,

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

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 term β is constant coefficients, estimated using the least-squares method.

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

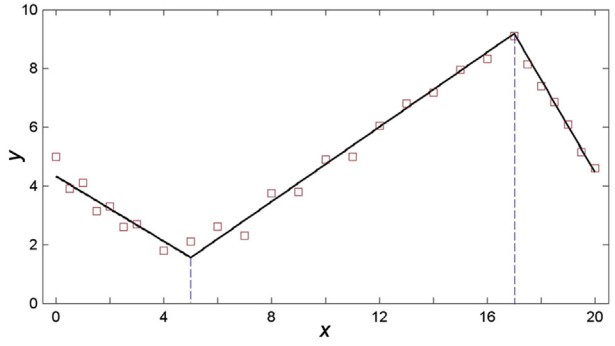

Figure 2. Knots and linear splines for a simple MARS example.

Table 1
Summary of performance measures.

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

$\overline{y}$ 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 Gandomi and Roke (2013).

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

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

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

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 (Hastie et al., 2009):

Table 2
Summary of input variables and outputs.

Inputs and outputs	Parameters and parameter descriptions		
Input variables	Hammer	Hammer weight (kN)	Variable 1 (x_1)
		Energy (kN·m)	Variable 2 (x_2)
	Hammer cushion	Area (m^2)	Variable 3 (x_3)
	material	Elastic modulus (GPa)	Variable 4 (x_4)
		Thickness (m)	Variable 5 (x_5)
		Helmet weight (kN)	Variable 6 (x_6)
	Pile information	Length (m)	Variable 7 (x_7)
		Penetration (m)	Variable 8 (x_8)
		Diameter (m)	Variable 9 (x_9)
		Section area (m^2)	Variable 10 (x_{10})
		L/D	Variable 11 (x_{11})
	Soil information	Quake at toe (m)	Variable 12 (x_{12})
		Damping at shaft (s/m)	Variable 13 (x_{13})
		Damping at toe (s/m)	Variable 14 (x_{14})
		Shaft resistance (%)	Variable 15 (x_{15})
	Ultimate pile capacity Q_u (kN)		Variable 16 (x_{16})
	Stroke (m)		Variable 17 (x_{17})
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

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

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 Friedman (1991). 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 (Friedman, 1991) can be used to assess the parameter relative importance based on the contributions from the input variables and the BFs.

4. Performance measures

Table 1 shows the performance measures utilized for prediction comparison of the two metahueristic methods.

5. Pile drivability analysis

5.1. Database

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 (Jeon and Rahman, 2008). Seventeen variables including hammer

Table 4
Sample training and testing data sets for category Q_1 type.

x_1 (kN)	x_2 (kN·m)	x_3 (m^2)	x_4 (GPa)	x_5 (m)	x_6 (kN)	x_7 (m)	x_8 (m)	x_9 (m)	x_{10} (m^2)	x_{11}	x_{12} (m)	x_{13} (s/m)	x_{14} (s/m)	x_{15} (%)	x_{16} (kN)	x_{17} (m)	MCS (MPa)	MTS (MPa)	BPF
Training data																			
12.9	34.5	0.18	3.35	0.09	9.37	10.67	9.75	0.36	0.014	27	0.0030	0.66	0.33	95	138	1.26	78.7	11.7	5.5
17.8	43.4	0.26	1.21	0.05	12.54	12.19	10.36	0.30	0.010	34	0.0025	0.59	0.49	95	178	1.18	89.2	6.1	5.3
14.7	36.7	0.15	2.81	0.11	5.83	15.24	15.24	0.30	0.010	50	0.0025	0.59	0.39	85	178	1.19	114.5	12.3	6.3
18.6	58.1	0.26	1.21	0.05	5.19	10.67	10.67	0.30	0.010	35	0.0025	0.72	0.33	75	200	1.17	123.5	2.3	6.4
7.8	27.3	0.20	1.21	0.05	12.41	18.29	18.29	0.30	0.010	60	0.0025	0.59	0.59	92	200	1.95	102.2	8.8	11.1
13.3	35.3	0.26	1.21	0.05	9.46	9.14	9.14	0.30	0.010	30	0.0025	0.66	0.33	15	222	1.70	87.5	0.0	6.6
18.6	58.1	0.16	1.21	0.05	4.00	4.57	4.57	0.30	0.010	15	0.0025	0.59	0.33	45	222	1.05	111.0	0.0	7.3
31.1	81.3	0.26	1.21	0.05	9.52	5.50	5.50	0.30	0.010	18	0.0025	0.50	0.33	57	225	1.22	60.7	0.0	3.0
29.4	102.3	0.27	1.93	0.05	4.00	8.50	8.50	0.30	0.010	28	0.0025	0.66	0.33	87	265	1.23	138.7	0.0	5.1
Testing data																			
12.9	34.5	0.18	3.35	0.09	9.37	10.67	9.75	0.36	0.014	27	0.0030	0.66	0.33	95	138	1.68	112.1	23.8	4.7
14.7	36.7	0.15	2.81	0.11	5.83	15.23	15.23	0.30	0.010	50	0.0025	0.59	0.39	85	178	1.68	154.5	29.4	5.3
18.6	58.1	0.26	1.21	0.05	5.19	12.19	6.40	0.36	0.014	18	0.0031	0.49	0.33	80	200	1.52	142.9	42.3	4.4

Table 5
Comparison of performance measures for BPNN and MARS.

Data sets		BPNN								MARS							
		R^2		R		RRMSE (%)		ρ (%)		R^2		R		RRMSE (%)		ρ (%)	
		Tr.	Te.	Tr.	Te.	Tr.	Te.	Tr.	Te.	Tr.	Te.	Tr.	Te.	Tr.	Te.	Tr.	Te.
Q_1	MCS	0.996	0.974	0.998	0.988	1.508	3.825	0.755	1.924	0.955	0.823	0.977	0.916	5.172	9.943	2.616	5.189
	MTS	0.993	0.966	0.996	0.984	11.98	22.58	6.001	11.38	0.966	0.944	0.983	0.972	25.89	28.99	13.06	14.70
	BPF	0.999	0.970	0.999	0.987	0.741	6.361	0.371	3.202	0.986	0.983	0.993	0.994	3.949	4.795	1.981	2.405
Q_2	MCS	0.998	0.991	0.999	0.996	0.935	1.689	0.468	0.846	0.995	0.986	0.997	0.994	1.398	1.949	0.700	0.977
	MTS	0.995	0.944	0.998	0.975	11.96	32.54	5.986	16.47	0.908	0.770	0.953	0.901	52.04	66.08	26.65	34.76
	BPF	0.982	0.988	0.991	0.995	5.691	4.495	2.859	2.253	0.968	0.977	0.984	0.989	7.564	6.141	3.813	3.088
Q_3	MCS	0.981	0.965	0.991	0.982	2.310	3.106	1.160	1.567	0.938	0.939	0.968	0.970	4.180	4.076	2.123	2.069
	MTS	0.946	0.802	0.973	0.915	34.53	62.30	17.50	32.53	0.876	0.728	0.936	0.869	52.13	73.08	26.93	39.10
	BPF	0.981	0.950	0.990	0.975	7.051	9.673	3.543	4.897	0.945	0.881	0.972	0.943	11.86	14.87	6.015	7.653
Q_4	MCS	0.968	0.963	0.984	0.981	3.064	3.227	1.545	1.629	0.963	0.959	0.981	0.979	3.309	3.377	1.670	1.706
	MTS	0.982	0.711	0.991	0.876	23.05	65.36	11.58	34.84	0.979	0.533	0.989	0.823	25.20	83.14	12.67	45.60
	BPF	0.926	0.982	0.962	0.946	13.99	16.93	7.128	8.703	0.892	0.867	0.944	0.931	16.90	18.98	8.692	9.827
Q_5	MCS	0.999	0.993	0.999	0.997	0.702	1.684	0.351	0.843	0.994	0.988	0.997	0.994	1.682	2.136	0.842	1.071
	MTS	0.998	0.906	0.999	0.956	4.662	33.71	2.332	17.23	0.933	0.886	0.966	0.943	28.11	37.10	14.30	19.09
	BPF	0.993	0.929	0.996	0.965	3.534	10.29	1.770	5.238	0.955	0.855	0.977	0.924	8.811	14.66	4.456	7.619
Combined	MCS	0.970	0.973	0.985	0.987	4.238	3.762	2.135	1.894	0.957	0.956	0.978	0.978	5.076	4.810	2.566	2.432
Q_1 to Q_5	MTS	0.798	0.842	0.893	0.921	80.82	62.36	42.68	32.46	0.784	0.782	0.885	0.894	83.60	73.22	44.34	38.65
	BPF	0.949	0.951	0.974	0.976	18.48	18.54	9.362	9.388	0.908	0.922	0.953	0.960	24.73	23.46	12.67	11.97

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

Geotechnical applications		BPNN	MARS
Q_1	MCS	224.88	6.16
	MTS	56.23	9.03
	BPF	200.38	8.82
Q_2	MCS	79.71	18.87
	MTS	208.37	4.54
	BPF	46.26	13.90
Q_3	MCS	285.73	13.74
	MTS	70.10	22.84
	BPF	106.02	33.46
Q_4	MCS	139.40	80.51
	MTS	144.50	83.56
	BPF	137.99	121.30
Q_5	MCS	173.12	10.30
	MTS	148.19	5.22
	BPF	203.11	16.72
Overall	MCS	203.51	131.35
	MTS	363.98	33.11
	BPF	182.68	63.96

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

Table 8
Basis functions and corresponding equations of MARS model for MCS of Q_1.

BF	Equation	BF	Equation
BF1	$\max(0, x_{17} - 2.44)$	BF14	$\max(0, 3.24 - x_4) \times \max(0, x_{15} - 15)$
BF2	$\max(0, x_6 - 9.34)$	BF15	$\max(0, x_{17} - 1.44)$
BF3	$\max(0, 9.34 - x_6)$	BF16	$\max(0, 1.44 - x_{17})$
BF4	$\max(0, 17.8 - x_1)$	BF17	$\max(0, 2.44 - x_{17}) \times \max(0, x_3 - 0.18)$
BF5	$\max(0, 4.57 - x_8)$	BF18	$\max(0, 2.44 - x_{17}) \times \max(0, 0.18 - x_3)$
BF6	BF5 $\times$ $\max(0, 2.29 - x_{17})$	BF19	$\max(0, 3.24 - x_4) \times \max(0, x_3 - 0.26)$
BF7	$\max(0, 2.44 - x_{17}) \times \max(0, x_1 - 29.4)$	BF20	$\max(0, 3.24 - x_4) \times \max(0, 0.26 - x_3)$
BF8	$\max(0, 2.44 - x_{17}) \times \max(0, 29.4 - x_1)$	BF21	$\max(0, 3.24 - x_4) \times \max(0, x_2 - 57.0)$
BF9	BF5 $\times$ $\max(0, x_{17} - 2.13)$	BF22	$\max(0, 3.24 - x_4) \times \max(0, 57.0 - x_2)$
BF10	BF5 $\times$ $\max(0, 2.13 - x_{17})$	BF23	$\max(0, 3.24 - x_4) \times \max(0, x2 - 43.4)$
BF11	$\max(0, 3.24 - x_4) \times \max(0, x_2 - 54.2)$	BF24	$\max(0, 3.24 - x_4) \times \max(0, 43.4 - x_2)$
BF12	$\max(0, x_1 - 17.8) \times \max(0, 0.18 - x_3)$	BF25	BF5 $\times \max(0, 5.83 - x_6)$
BF13	$\max(0, 0.0030 - x_{12})$		

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 are listed in Table 2.

For purpose of simplifying the analyses considering the extensive number of parameters and large data set, Jeon and Rahman (2008) divided the data into five categories 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 data set and the remaining data were used for testing.

5.2. Pile category Q_1

Table 4 lists some sample training (Tr.) and testing (Te.) data sets for Category Q_1 piles. Q_1 piles have been reanalyzed using MARS and BPNN. For the developed BPNN models, the optimum numbers of hidden neurons are 8, 7, and 9 for MCS, MTS and BPF, respectively. The MARS models predicting MCS, MTS and BPF comprise of 25, 43 and 40 BFs, respectively.

Comparisons of R^2, r, RRMSE and ρ in rows 4, 5 and 6 of Table 5 indicate that the differences in the accuracy of the BPNN and MARS models are marginal. Therefore, both methods serve as reliable tools for prediction of pile drivability for Q_1 piles. 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 algorithm is obvious, which indicates that the distinct advantage of MARS over BPNN lies in its convergence speed.

Table 7 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

Table 7
ANOVA decomposition of MARS model for MCS, MTS and BPF of Q_1.

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			

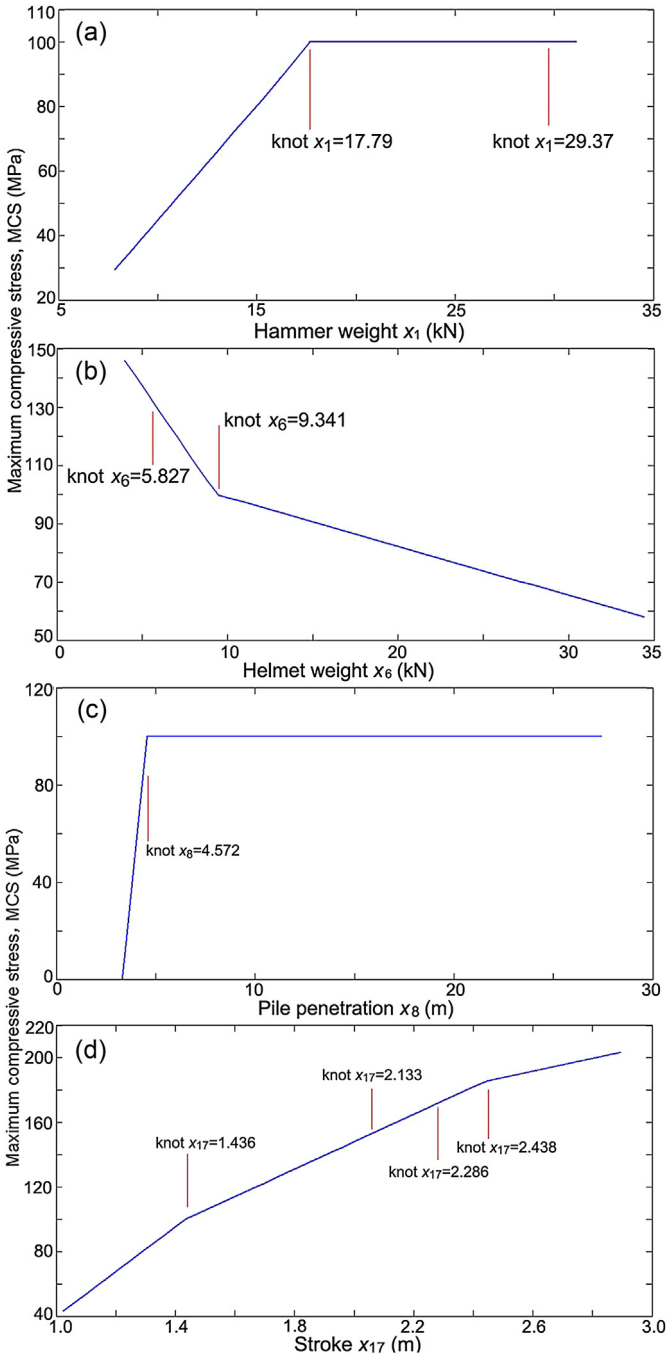

Figure 3. Knot locations of MARS model for MCS for pile category Q_1.

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 the 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 8 lists the BFs of the MCS model and the corresponding equations. The MARS model to estimate MCS for Q_1 is given by

Table 9
No. of hidden neurons and No. of BFs for modeling.

No. of models		Data			
		Q_2	Q_3	Q_4	Q_5
BFs	MCS	46	30	62	42
	MTS	33	25	65	28
	BPF	39	50	68	50
hidden neurons	MCS	9	10	9	9
	MTS	9	10	10	9
	BPF	8	10	9	9

$$\begin{aligned}\mathrm{MCS}\left(\mathrm{MPa}\right) = {} & 100.1 - 44.81 \times \mathrm{BF1} - 1.679 \times \mathrm{BF2} + 8.58 \\ & \times \mathrm{BF3} - 7.11 \times \mathrm{BF4} - 79.87 \times \mathrm{BF5} + 412 \\ & \times \mathrm{BF6} - 12.7 \times \mathrm{BF7} + 2.25 \times \mathrm{BF8} + 134 \times \mathrm{BF9} \\ & - 387 \times \mathrm{BF10} + 8.4 \times \mathrm{BF11} - 1817 \times \mathrm{BF12} \\ & + 2.58 \times 10^4 \times \mathrm{BF13} + 0.067 \times \mathrm{BF14} + 85 \\ & \times \mathrm{BF15} - 138.7 \times \mathrm{BF16} - 98.1 \times \mathrm{BF17} - 424.6 \\ & \times \mathrm{BF18} + 902 \times \mathrm{BF19} + 85 \times \mathrm{BF20} - 5.7 \\ & \times \mathrm{BF21} - 0.66 \times \mathrm{BF22} - 2.4 \times \mathrm{BF23} + 0.65 \\ & \times \mathrm{BF24} - 18.9 \times \mathrm{BF25} \end{aligned} \quad (7)$$

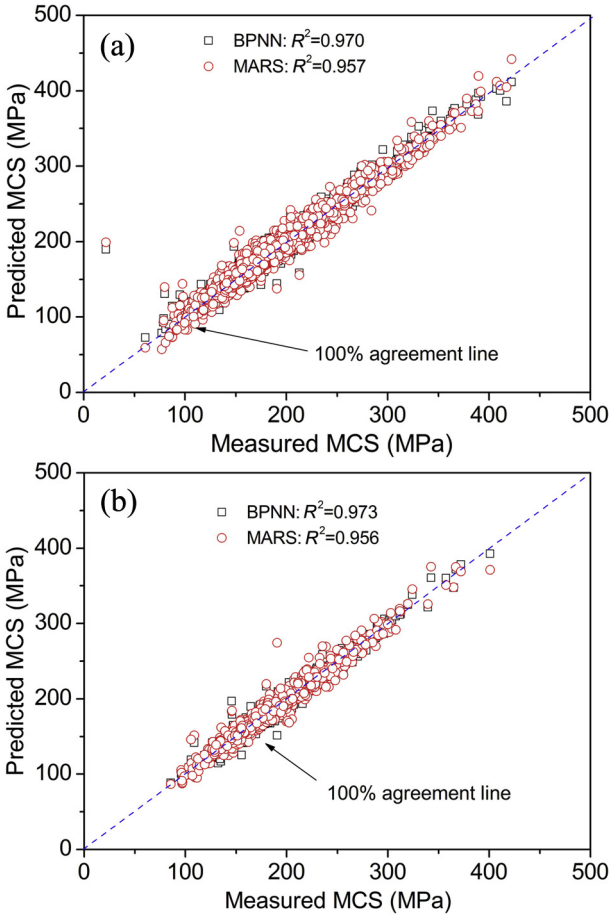

Figure 4. Comparison of MCS using MARS and BPNN.

Fig. 3a–d plots the knot locations for x_1 (Hammer weight), x_6 (Helmet weight), x_8 (pile penetration) and x_{17} (stroke) respectively.

5.3. Pile categories Q_2–Q_5

The results for the remaining four pile categories Q_2–Q_5 analyzed using MARS and BPNN are described in this section. Table 9 lists the number of BFs for MARS model and the number of hidden neurons for BPNN model for each category, respectively.

Comparisons of R^2, r, RRMSE and ρ in Table 5 between MARS and BPNN shown in rows 7–9, rows 10–12, rows 13–15, and rows 16–18 are for Q_2 to Q_5 piles, respectively, from which it is obvious that BPNN gives only slightly more accurate predictions than MARS. Both MARS and BPNN can serve as reliable tools for pile drivability prediction for Q_2 to Q_5. Rows 5–7, 8–10, 11–13 and 14–16 of Table 6 list the CPU processing time using BPNN and MARS models for Q_2 to Q_5, respectively. The advantage of the processing speed of MARS is obvious, indicating the advantage of MARS over BPNN in computational efficiency. For brevity, the comparison of parameter relative importance and the interpretable MARS models are not elaborated here.

5.4. Combined dataset Q_1 to Q_5

Additional analyses were also carried out using the entire 4072 pile data set. The MARS models to predict MCS, MTS and BPF use 52, 37 and 35 BFs, respectively. As for the BPNN models, the numbers of hidden neurons are 9, 9, and 8 for MCS, MTS and BPF, respectively.

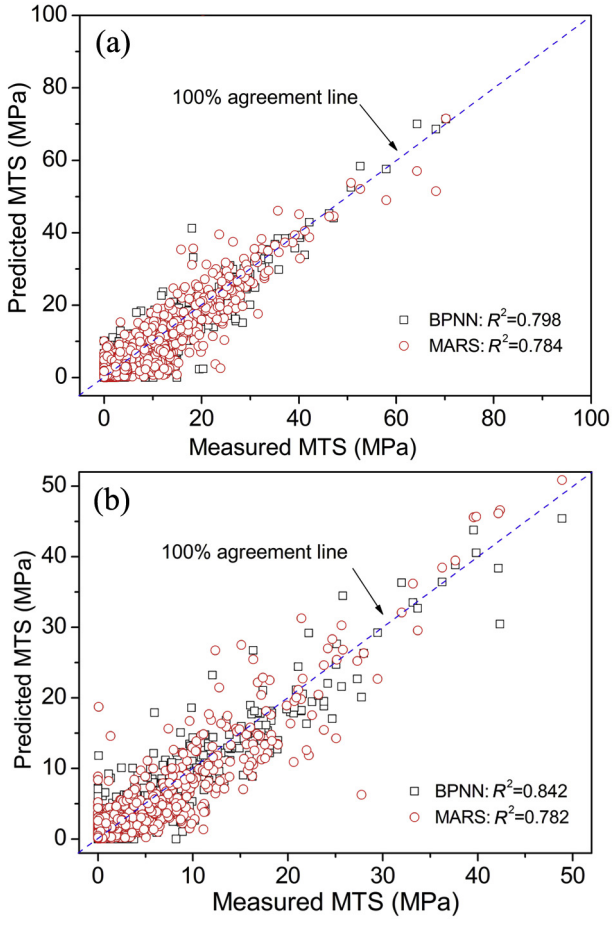

Figure 5. Comparison of MTS using MARS and BPNN.

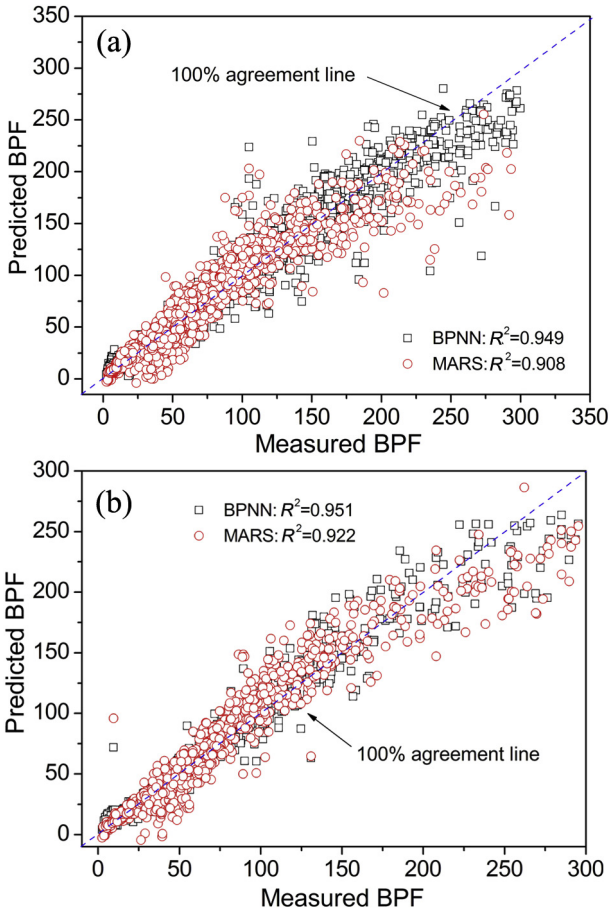

Figure 6. Comparison of BPF using MARS and BPNN.

Table 10
Basis functions and corresponding equations of MARS model for MCS overall data sets.

BF	Equation	BF	Equation
BF1	$\max(0, x_{16} - 1550)$	BF27	$\max(0, x_{15} - 15)$
BF2	$\max(0, x_{17} - 2.29)$	BF28	$\max(0, 15 - x_{15})$
BF3	$\max(0, 2.29 - x_{17})$	BF29	BF28 × $\max(0, x_{16} - 289)$
BF4	$\max(0, x_6 - 7.38)$	BF30	BF28 × $\max(0, 289 - x_{16})$
BF5	$\max(0, 7.38 - x_6)$	BF31	BF2 × $\max(0, x_1 - 29.4)$
BF6	$\max(0, 0.014 - x_{10})$	BF32	BF6 × $\max(0, x_6 - 6.67)$
BF7	$\max(0, x_2 - 30.7)$	BF33	BF6 × $\max(0, 6.67 - x_6)$
BF8	$\max(0, 30.7 - x_2)$	BF34	BF5 × $\max(0, 1.81 - x_{17})$
BF9	BF1 × $\max(0, x_7 - 8.00)$	BF35	BF3 × $\max(0, x_1 - 29.4)$
BF10	BF1 × $\max(0, 8.00 - x_7)$	BF36	BF7 × $\max(0, x_{11} - 50)$
BF11	$\max(0, x_{11} - 9)$	BF37	BF7 × $\max(0, 50 - x_{11})$
BF12	$\max(0, 9 - x_{11})$	BF38	BF28 × $\max(0, x_{13} - 0.59)$
BF13	$\max(0, 1550 - x_{16})$ × $\max(0, x8 - 3.05)$	BF39	BF28 × $\max(0, 0.59 - x_{13})$
BF14	$\max(0, 1550 - x_{16})$ × $\max(0, 3.05 - x8)$	BF40	BF4 × $\max(0, x_5 - 0.05)$
BF15	$\max(0, 1550 - x_{16})$ × $\max(0, x6 - 9.34)$	BF41	BF4 × $\max(0, 0.05 - x_5)$
BF16	$\max(0, 1550 - x_{16})$ × $\max(0, 9.34 - x6)$	BF42	$\max(0, 1550 - x_{16})$ × $\max(0, x_{11} - 24)$
BF17	BF6 × $\max(0, x_{16} - 1067.5)$	BF43	$\max(0, 1550 - x_{16})$ × $\max(0, 24 - x_{11})$
BF18	BF6 × $\max(0, 1068 - x_{16})$	BF44	BF7 × $\max(0, 0.18 - x_3)$
BF19	BF11 × $\max(0, x_4 - 3.24)$	BF45	$\max(0, x_3 - 0.26)$
BF20	BF11 × $\max(0, 3.24 - x_4)$	BF46	$\max(0, 0.26 - x_3)$
BF21	BF11 × $\max(0, x_1 - 29.4)$	BF47	BF5 × $\max(0, x_4 - 1.97)$
BF22	BF11 × $\max(0, 29.4 - x_1)$	BF48	BF5 × $\max(0, 1.97 - x_4)$
BF23	BF6 × $\max(0, x_7 - 3.05)$	BF49	BF5 × $\max(0, 44.7 - x_2)$
BF24	BF6 × $\max(0, 3.05 - x_7)$	BF50	BF45 × $\max(0, 30 - x_{11})$
BF25	BF7 × $\max(0, x_{17} - 2.90)$	BF51	BF11 × $\max(0, x_2 - 54.2)$
BF26	BF7 × $\max(0, 2.90 - x_{17})$	BF52	BF11 × $\max(0, 54.2 - x_2)$

Figs. 4–6 plot the BPNN and MARS estimations vs. the measured values for MCS, MTS, and BPF models. For MCS prediction, high R^2 are obtained from both methods. Compared with the MCS predictions, the developed BPNN and MARS models are less accurate in predicting MTS mainly as a result of the bias (errors) due to the significantly smaller tensile stress values. In addition, both BPNN and MARS models can serve as reliable tools for prediction of BPF.

Comparisons of R^2, r, RRMSE and ρ in rows 19–21 of Table 5 indicate that BPNN gives only slightly more accurate predictions than MARS. Rows 17–19 of Table 6 compare the CPU processing time. It is obvious that MARS performs better than BPNN in the convergence speed.

It should be pointed out that the parameter relative importance can also be assessed. For MARS this is carried out by evaluating the GCV increase caused by removing the considered variables from the developed MARS model. For the BPNN, this is commonly carried out using the method by Garson (1991) and discussed by Das and Basudhar (2006). For brevity, these comparisons have been omitted.

Table 10 lists the BFs of the MCS model and the corresponding equations. The MARS model is in the form of:

$$\begin{aligned}\mathrm{MCS(MPa)} = {} & 169.4 + 0.0095 \times \mathrm{BF1} + 35.6 \times \mathrm{BF2} - 47.5 \times \mathrm{BF3} \\ & - 0.46 \times \mathrm{BF4} - 2 \times \mathrm{BF5} + 8847 \times \mathrm{BF6} + 9.2 \times \mathrm{BF7} \\ & - 8.2 \times \mathrm{BF8} - 0.0025 \times \mathrm{BF9} + 0.0062 \times \mathrm{BF10} - 3.2 \\ & \times \mathrm{BF11} + 470 \times \mathrm{BF12} - 0.0036 \times \mathrm{BF13} - 0.8 \\ & \times \mathrm{BF14} - 0.0012 \times \mathrm{BF15} + 0.006 \times \mathrm{BF16} + 9.43 \\ & \times \mathrm{BF17} - 6.1 \times \mathrm{BF18} + 0.136 \times \mathrm{BF19} - 0.098 \\ & \times \mathrm{BF20} - 0.83 \times \mathrm{BF21} - 0.17 \times \mathrm{BF22} - 540 \times \mathrm{BF23} \\ & + 1.34 \times 10^5 \times \mathrm{BF24} + 1.672 \times \mathrm{BF25} - 0.42 \times \mathrm{BF26} \\ & + 0.144 \times \mathrm{BF27} - 4.57 \times \mathrm{BF28} + 0.0054 \times \mathrm{BF29} \\ & + 0.052 \times \mathrm{BF30} + 87 \times \mathrm{BF31} + 250 \times \mathrm{BF32} - 763 \\ & \times \mathrm{BF33} - 16 \times \mathrm{BF34} - 28.1 \times \mathrm{BF35} + 0.217 \times \mathrm{BF36} \\ & - 0.2 \times \mathrm{BF37} + 34.5 \times \mathrm{BF38} + 31.3 \times \mathrm{BF39} - 50.2 \\ & \times \mathrm{BF40} - 425 \times \mathrm{BF41} + 0.0018 \times \mathrm{BF42} - 0.003 \\ & \times \mathrm{BF43} - 7.4 \times \mathrm{BF44} + 341 \times \mathrm{BF45} + 51.4 \times \mathrm{BF46} \\ & + 5.67 \times \mathrm{BF47} + 12 \times \mathrm{BF48} + 0.96 \times \mathrm{BF49} + 100.2 \\ & \times \mathrm{BF50} - 0.2 \times \mathrm{BF51} + 0.23 \times \mathrm{BF52} \end{aligned} \tag{8}$$

6. Summary and conclusions

A database containing 4072 pile data sets with a total of seventeen variables is adopted to develop the BPNN and MARS models for drivability predictions in relation to the prediction of the Maximum compressive stresses, Maximum tensile stresses, and Blow per foot. The pile data were divided into five categories Q_1 to Q_5 based on the ultimate pile capacity. BPNN and MARS models were built for each category and also using the combined data sets. Performance measures indicate that BPNN and MARS models for the analyses of pile drivability provide similar predictions and can thus be used for predicting pile drivability. MARS can be considered to be more computationally efficient than BPNN, as the MARS algorithm builds flexible models using simpler linear regression and data-driven stepwise searching, adding and pruning. In addition, the developed MARS models are easier to be interpreted. Furthermore, since MARS explicitly defines the knots for each design input variables, the model enables engineers to have an insight and understanding of where significant changes in the data may occur.

Acknowledgments

The authors would like to express their appreciation to Jeon and Rahman (2008) for making the pile drivability database available for this work.

References

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.
Das, S.K., Basudhar, P.K., 2006. Undrained lateral load capacity of piles in clay using artificial neural network. Computer and Geotechnics 33 (8), 454–459.
Demuth, H., Beale, M., 2003. Neural Network Toolbox for MATLAB-user Guide Version 4.1. The Math Works Inc.
Friedman, J.H., 1991. Multivariate adaptive regression splines. The Annals of Statistics 19, 1–141.
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 and Multiscale, 12–14 Jun., Cambridge, USA.
Garson, G.D., 1991. Interpreting neural-network connection weights. AI Expert 6 (7), 47–51.
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.
Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference and Prediction, second ed. Springer.
Jekabsons, G., 2010. VariReg: a Software Tool for Regression Modelling Using Various Modeling Methods. Riga Technical University. http://www.cs.rtu.lv/jekabsons/.
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.
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.
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.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning internal representation by error propagation. In: Rumelhart, D.E., McClelland, J.L. (Eds.), Parallel Distributed Processing, vol. 1. MIT Press, Cambridge, pp. 318–362.
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.
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.
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.
Smith, E.A.L., 1960. Pile driving analysis by the wave equation. Journal of the Engineering Mechanics Division ASCE 86, 35–61.
Zarnani, S., El-Emam, M., Bathurst, R.J., 2011. Comparison of numerical and analytical solutions for reinforced soil wall shaking table tests. Geomechanics and Engineering 3 (4), 291–321.
Zhang, W.G., Goh, A.T.C., 2013. Multivariate adaptive regression splines for analysis of geotechnical engineering systems. Computers and Geotechnics 48, 82–95.
Zhang, W.G., Goh, A.T.C., 2014a. Multivariate adaptive regression splines model for reliability assessment of serviceability limit state of twin caverns. Geomechanics and Engineering 7 (4), 431–458.
Zhang, W.G., Goh, A.T.C., 2014b. Reliability assessment of ultimate limit state of twin cavern. Geomechanics and Geoengineering (under review).