Arab J Sci Eng (2015) 40:1565–1578
DOI 10.1007/s13369-015-1624-y

RESEARCH ARTICLE - CIVIL ENGINEERING

Prediction of Lateral Load Capacity of Pile in Clay Using Multivariate Adaptive Regression Spline and Functional Network

Sarat Kumar Das · Shakti Suman

Received: 19 September 2014 / Accepted: 19 February 2015 / Published online: 10 March 2015

Abstract This paper discusses the use of multivariate adaptive regression splines (MARS) and functional networks (FN) for prediction of the lateral load capacity of piles in clay. The results obtained from MARS and FN have been compared with different empirical models and artificial neural network in terms of statistical parameters such as correlation coefficient (R), Nash–Sutcliff coefficient of efficiency (E), absolute average error, maximum average error and root mean square error. Based on the statistical parameters, MARS and FN were found to have a better predictive capacity. Predictive equations are provided based on the MARS and FN model. A sensitivity analysis is also presented to determine the importance of inputs in prediction of the lateral load capacity of piles.

Keywords Load capacity of piles · Artificial intelligence · Multivariate adaptive regression splines · Functional networks

1 Introduction

Pile foundations are frequently subjected to lateral loads due to earth pressure, earthquake, wave or wind forces in different structures along with axial load. Thus, the design of pile foundations has found the attention of researchers more than any other foundation structure. The axially loaded piles are more frequently used. The earliest attempts at prediction of laterally loaded piles was by Hansen [1] and Broms [2,3] based on earth pressure theories. Poulos and Davis [4] used dynamic equations based on Winkler's soil model.

However, design of laterally loaded piles is complex and requires solving nonlinear differential equations. Elastic analysis as adopted by Poulos and Davis [4] is not suitable for nonlinear behaviour of soil. Matlock and Reese [5] used nonlinear $p-y$ curves to predict the lateral load capacity of pile. Portugal and Seco e Pinto [6] used nonlinear $p-y$ curves and the finite element method for prediction of the behaviour of laterally loaded piles. Muthukkumaran et al. [7] proposed a modified method to draw the $p-y$ curves for piles in horizontal ground under surcharge load. The effect of slope on these $p-y$ curves was also studied. But these methods were found to have uncertainty in predictions owing to the variations in soil properties. Hence, some empirical methods such as Hansen [1] and Broms [2,3] are in use. Begum and Muthukkumaran [8] studied the behaviour and proposed correction factors to calculate the lateral load capacity and maximum bending moment of piles on sloping ground under lateral loads. Muthukkumaran [9] performed laboratory model tests to study the effect of slope and loading direction on laterally loaded piles in cohesion less soil. He observed that the lateral load capacity of piles in sloping ground is less than that in horizontal ground for both forward and reverse loading.

Artificial intelligence (AI) techniques are considered as an alternate statistical method by many researchers and is found to be better in prediction as compared to the empirical methods [10–14]. Goh [12] used back propagation neural networks (BPNN) to predict the skin friction in piles. Further, Goh [15,16] observed that the ultimate load capacity of piles predicted by artificial neural network (ANN) had better performance than the Engineering News formula, Hiley formula and Janbu formula. Other later attempts at predicting the pile load capacity in both cohesionless soil and clayey

S. K. Das (✉) · S. Suman
Civil Engineering Department, National Institute of Technology Rourkela, Rourkela 769008 Odisha, India
e-mail: saratdas@rediffmail.com

S. Suman
e-mail: shaktisuman7@gmail.com

soil using ANN showed a better prediction capability of ANN compared to traditional empirical methods [10,17–20]. Das and Basudhar [10] found ANN to be better than Broms and Hansen. Samui [21] found a better prediction than ANN using another AI technique, support vector machine (SVM). Pal and Deswal [22] developed Gaussian process regression (GPR) and SVM models using the data set of Das and Basudhar [10] and observed a better prediction for GPR. However, their observation was based upon correlation coefficient (R) and root mean square error (RMSE) only. Alkroosh and Nikraz [23] developed a gene expression programming model (GEP) using the same data and found that GEP is good at predicting the lateral load capacity of piles embedded in clay.

ANN assures better performance by iteration of learning algorithms. This process runs the risk of termination of the learning step if local minima is attained and hence leads to poor generalization in ANN. On the other hand, SVM is found to show a better generalization, but the user needs to set the parameters C (margin parameter) and ε (insensitive loss function) for a better result. The user is not able to get a comprehensive mathematical model of ANN and SVM, and hence these techniques are categorized as 'black box' systems. Giustolisi et al. [24] divided mathematical models into three types, black box, grey box and white box in order of the ease of explanation of the functional form of the relationships between the variables. The white box models are based on physical laws where model variables and parameters are known and the underlying physical relationship can be easily explained. On the other hand, the functional relationships between model variables is unknown for a black box system and needs to be determined. Black box models are data driven, and the relationship between input and output is based on data. Grey box systems are conceptual, and a mathematical model can be derived for them.

A modified statistical technique called multivariate adaptive regression spline (MARS) has been popularized by Friedman [25] for solving regression-type problems. MARS is also called a 'white box' system of predictive model, as it is based on physical laws and underlying physical relationships of the system can be explained. The MARS technique is very popular in the area of data mining because it does not assume or impose any particular type or class of relationship (e.g. linear and logistic) between the predictor variables and the dependent (outcome) variables of interest. This makes MARS particularly suitable for problems with a greater number of variables. It has an increasing number of applications in many areas of economy, science and technology. However, its use in geotechnical engineering is very limited [26].

In the recent past, a new prediction tool, functional network (FN), which is based upon the structure of the physical world has found its use in many fields of science and engineering including petroleum engineering [27], signal processing, pattern recognition, function's approximations [28], real-time flood forecasting, science, bioinformatics, medicine [29], mining, structural engineering [30] and transportation engineering [31]. FNs were introduced by Castillo [32], Gomez [33] and Castillo et al. [34–36]. Though FN is similar to ANN, it has added advantages, making it a powerful alternative to ANN.

The present paper is an attempt to develop prediction models for lateral load capacity of piles in clay using MARS and FN. Different statistical criteria such as correlation coefficient (R), Nash–Sutcliff coefficient of efficiency (E) [37], absolute average error (AAE), maximum absolute error (MAE) and RMSE are used to compare the MARS and FN models with ANN, Broms [2] and Hansen [1] methods. A ranking system [38] using rank index (RI) has also been followed to compare the different models based on four criteria: (i) R and E for predicted lateral load capacity (Q_p) and measured lateral load capacity (Q_m), (ii) mean and standard deviation of the ratio, Q_p/Q_m (iii) 50 and 90 % cumulative probabilities (P_{50} and P_{90}) of the ratio, Q_p/Q_m and (iv) the probability of pile load capacity within 20 % accuracy level in percentage using histogram and lognormal probability distribution of Q_p/Q_m.

2 Methodology

In the present paper, MARS and FN have been used to train models used to predict the lateral load capacity of piles. Since the use of MARS is limited in geotechnical engineering and FN has not been used in geotechnical engineering, the following section briefly describes the underlying principles of MARS and FN with examples.

2.1 Multivariate Adaptive Regression Splines

Multivariate adaptive regression splines, as the name suggests, is an adaptive regression technique used to fit the relationship between a set of input variables and an independent variable. MARS uses a nonparametric regression technique for prediction of the dependent variable, i.e. no prior assumption is made about the relationship between the dependent and independent variables. This relation is constructed from a set of coefficients and basis functions (BFs) determined entirely from the data in hand. Thus, MARS is advantageous over other statistical techniques for problems with a greater amount of input data.

MARS uses a divide and conquer strategy to determine the relation between the dependent and independent variables. This includes the division of the training data set into a number of piecewise linear segments called splines of different gradients. The end points of splines are known as knots, and the piece-wise linear functions or piece-wise cubic func-

tions between two knots are known as a BF. For simplicity, only piece-wise linear functions have been discussed here and used for the prediction model of lateral load capacity of piles.

An algorithm for MARS was developed by Friedman [25] based on the above strategy. MARS fits data through a two-step process:

i. Forward stepwise algorithm: this is the step where the BFs are added. Initially, a model is constructed only with the intercept, β_0. In each subsequent step, the BF that will produce the largest decrease in the training error is added. This process continues till a predetermined maximum number of BFs is reached. This leads to an overfitted model. An adaptive regression algorithm is used to search for knot locations among all the variables.
ii. Backward pruning algorithm: this step is applied to eliminate the overfitting of the data. In this process, the terms in the model are pruned by removing terms one by one. The least effective term is removed in each pass to achieve at the best possible sub model. Model subsets are compared using the generalized cross-validation (GCV) technique. For a data with N samples, GCV is calculated using the equation:

$$\mathrm{GCV} = \frac{\frac{1}{N}\sum_{i=1}^{N}[Y_i - f(X_i)]^2}{\left[1 - \frac{M+d(M-1)/2}{N}\right]^2} \tag{1}$$

where M is the number of BFs, d is the penalizing parameter, N is the number of data sets and $f(X_i)$ denotes the predicted values of the MARS model. The denominator of GCV is responsible for increasing variance in case of increasing model complexity. The term $(M-1)/2$ in the denominator represents the number of knots. Thus, GCV penalizes BFs as well as knots.

To understand MARS better, we consider a data set with y as an output and $X = \{X_1, X_2, X_3, \ldots, X_p\}$ as an input matrix containing p variables. A model generated by MARS would be of the form,

$$y = f\left(X_1, X_2, X_3, \ldots, X_p\right) + e = f\left(X\right) + e \tag{2}$$

where e is the distribution of error. The function $f(x)$ is approximated using BFs which may be piece-wise linear or piece-wise cubic functions. For simplicity, only piece-wise linear functions are discussed here. A piece-wise linear function is of the form $\max(0, x-t)$ where t is the location of a knot. It is defined as,

$$\max\left(0, x-t\right) = \{x-t, \quad \text{if } x > t \text{ or, } 0 \text{ otherwise}\} \tag{3}$$

Finally, $f(x)$is defined as a linear combination of BFs and their interactions and is expressed as

$$f\left(X\right) = \beta_0 + \sum_{i=1}^{M} \beta_m \lambda_m(X) \tag{4}$$

where each $\lambda_m$is a BF, which can be a spline or a product of two or more splines. The coefficients β are constants estimated using the least squares method.

Figure 1 gives an example of how MARS uses piece-wise linear splines to fit a data set. We took an example of 22 data sets one input and one output. The input $\{X\}$ consisted of random numbers between 1 and 12, and the output $\{Y\}$ was calculated using the equation

$$Y_i = \sin\left(X_i\right) + \cos(X_i) \tag{5}$$

Further, the data were normalized between 0 and 1 and were analysed with MARS. The model developed by MARS for this data set was found to be

$$\hat{Y} = 1.91 - 3.22 \times \mathrm{BF1} - 10.69 \times \mathrm{BF2} - 7.87 \times \mathrm{BF3} \tag{6}$$

where $\hat{Y}$ denotes predicted values and,

$$\mathrm{BF1} = \max\left(0, X_i - 0.40\right) \tag{7}$$

$$\mathrm{BF2} = \max\left(0, 0.40 - X_i\right) \tag{8}$$

$$\mathrm{BF2} = \max\left(0, 0.65 - X_i\right) \tag{9}$$

The knots for this MARS model are thus located at $x = 0.40$ and $x = 0.65$. A correlation value of 0.983 was found between original and predicted values of MARS. It must be noted that we need to put normalized values of X_i in Eqs. (7)–(9) and the denormalized value of predicted Y_i can be calculated using Eq. (10).

$$\hat{Y}_{\mathrm{denorm}} = \hat{Y}_{\mathrm{norm}} \times \left(X_{i_{(\max)}} - X_{i_{(\min)}}\right) + X_{i_{(\min)}} \tag{10}$$

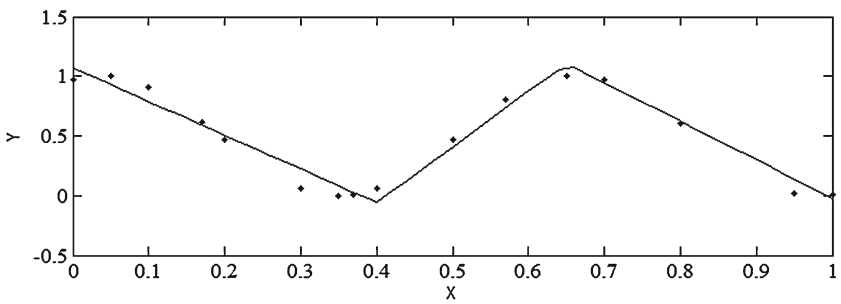

Fig. 1 Use of piece-wise linear splines by MARS to fit a data

Thus, MARS not only predicts the value of output with high accuracy it also simplifies a complex equation as Eq. (5) to a simple linear equation.

2.2 Functional Networks

Functional Networks are a recently introduced extension of neural networks. A FN is termed as a novel generalization of neural networks because of its ability to take into account both data as well as properties of the function being modelled (domain knowledge) to estimate the unknown neuron functions. Once the initial topology is available, functional equations can be used to arrive at a much simpler topology. FNs, thus, eliminate the problem of neural networks being 'black boxes' by using both the domain knowledge, i.e. associative, commutative and distributive, and the data knowledge to derive the topology of the problem. FNs use domain knowledge to determine the structure of the network and data to estimate the unknown neuron functions. In FN, arbitrary neural functions are allowed and they are initially assumed to be multiargument and vector-valued functions.

2.2.1 Differences Between FN and ANN

Figure 2 shows a typical neural network and its corresponding FN. The characteristic features of the FNs and their respective

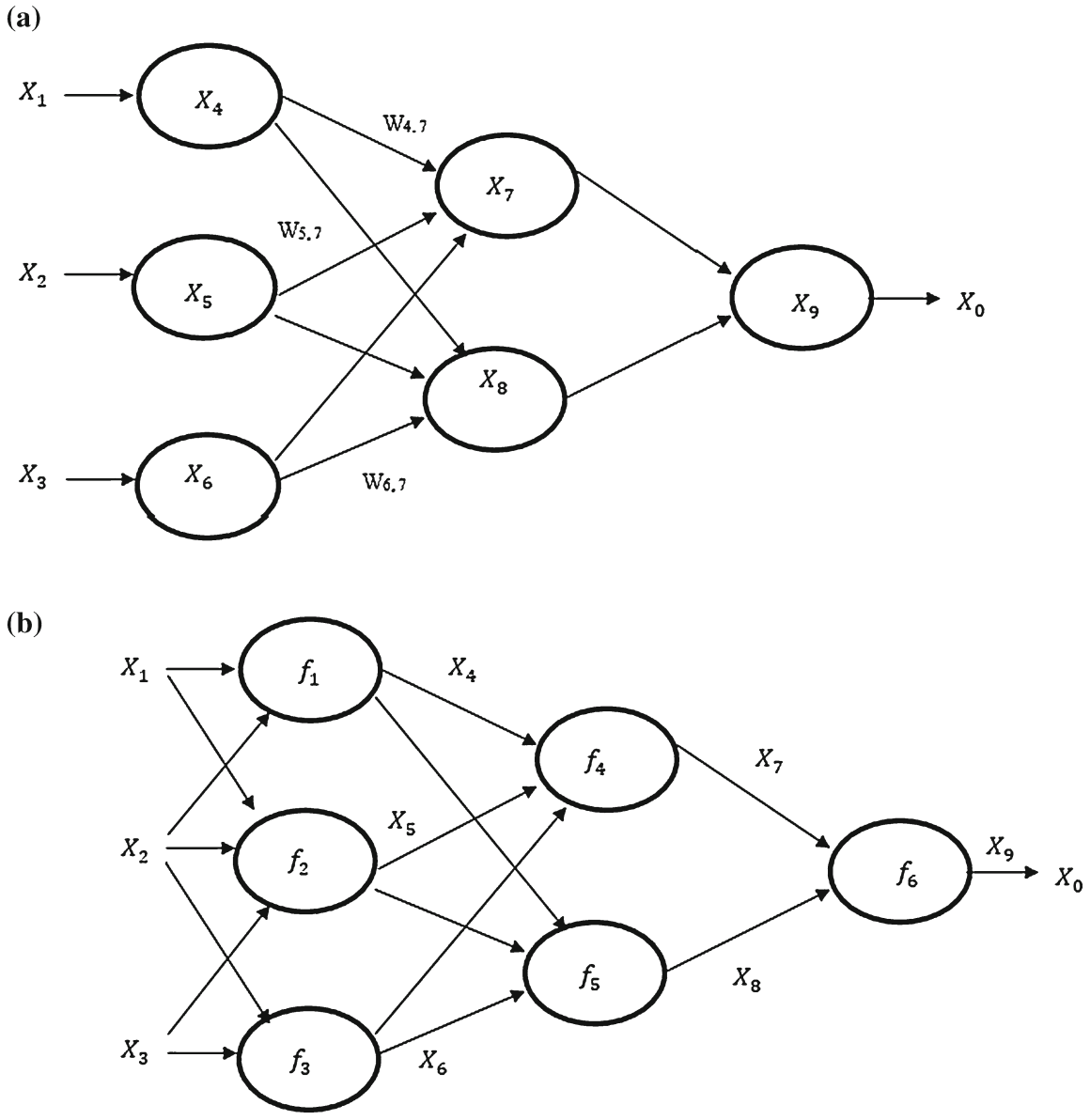

Fig. 2 A neural network and its equivalent functional network. **a** A neural network. **b** Functional network equivalent to the neural network in Fig. 1a.

differences from the neural networks can be enumerated as follows:

1. In FN, the information for selection of topology can be derived either from the data or from domain knowledge or from combinations of the two, whereas for neural networks, only the data are used.
2. In FNs, the functions are learned during the structural learning and estimated during the parametric learning, whereas in neural networks, the neuron functions are assumed to be fixed and known and only the weights are learned.
3. FN can use arbitrary multiargument and vector-valued functions, whereas in neural networks, they are fixed sigmoidal functions.
4. Intermediate layers of units are introduced in FN architectures to allow several neuron outputs to be connected to the same units. This is not possible in neural networks.

2.2.2 Working with Functional Networks

Figure 2b shows the main elements generally encountered in every FN. They can be enumerated as:

1. Storing Units
   - One layer each of input storing units and output storing units for the input data and output data, respectively. For example, in Fig. 2b, x_1, x_2, x_3, etc. are the inputs and f_4, f_5. are the outputs. In addition, there may be one or many layers of processing units, which process inputs from the previous layer and feed the output to the next layer, e.g. f_6 in Fig. 2b.
   - Intermediate storing units that contain intermediate information produced by neurons (X_4, X_5 in Fig. 2b).
2. Directed links to connect various input, output and intermediate units in accordance with the advance of the algorithm of FN.

Following are the steps required to work with FNs:

Step 1: the physical relationship between input and output.
Step 2: based on the data available in the problem, the initial topology of the FN is selected. Unlike neural networks, where the topology is selected by a trial and error method, the topology in FN is selected on the basis of properties and leads to selection of a single network structure.
Step 3: the network achieved initially is simplified using functional equations. For a given FN, it is assessed whether there exists another simpler network that gives the same output for the given set of inputs. If there exists such a network, the complex and simpler networks are called as equivalent FNs. This is known as structural learning.
Step 4: for a given topology, a unique neuron function is arrived that produces a set of output.
Step 5: this step includes collection of data for learning of the network.
Step 6: the neuron functions are estimated based on the data in step 5 and combination of given functional families. The learning may be linear or nonlinear based on the linearity of the neuron functions obtained.
Step 7: the obtained model is checked for errors and cross validated against a different set of data. The learning method of a FN consists of obtaining the neural functions based on a set of data $U = \{I_i, O_i\}, \{i = 1, 2, 3, 4 \ldots, n\}$. The learning process is based on minimizing the Euclidean norm of the error function, given by

$$E = \frac{1}{2} \sum_{i=1}^{n} (O_i - F(i))^2 \tag{11}$$

The approximate neural function $f_i(x)$ may be arranged as

$$f_i(x) = \sum_{j=1}^{m} a_{ij} \emptyset_{ij}(X) \tag{12}$$

where $\emptyset$ = shape functions with algebraic expressions $(1, x, x^2, x^3, \ldots, x^n)$, trigonometric functions such as $[1, \sin(x), \cos(x), \sin(2x), \cos(2x), \sin(3x), \cos(3x)]$, or exponential functions such as $(1, e^x, e^{2x}, \ldots, e^{nx})$. The associative optimization function may lead to a system of linear or nonlinear algebraic equations.

The knowledge of functional equations is essential while dealing with FNs. A functional equation is an equation in which the unknowns are functions, excluding differential and integral equations. The most common example of functional equation is the Cauchy's functional equation:

$$f(x + y) = f(x) + f(y); \quad x, y \in R \tag{13}$$

2.2.3 Associativity Functional Network

This paper applies the use of associativity FNs. In general, with the use of the basic theory of functional equations, any multi-input network can be transformed to an associative network [30,33]. To illustrate this, consider a property Y that depends on three inputs $\{x_1, x_2, x_3\}$. Consider first, the effect of inputs $x_1$and x_2 on Y in the form of a function $F_1(x_1, x_2)$. The effect of the third input x_3 can be incorporated by means of another function G_1 in the form $Y = G_1[F_1(x_1, x_2), x_3]$. Assuming such a relation exists for a permutation of any two inputs, we get a series of functional equation as:

$$\begin{aligned} Y &= G_1[F_1(x_1, x_2), x_3] = G_2[x_1, F_2(x_2, x_3)] \\ &= G_3[x_2, F_3(x_1, x_3)] \end{aligned} \tag{14}$$

In Eq. (14), the functions F_1, F_2, F_3, G_1, G_2 and G_3 are invertible with respect to both variables. The FN based on Eq. (14) in shown in Fig. 3a. To further simplify the FN in Fig. 3a, consider the solution of the functional equations in (6). The generalized solution of an equation of the form $G_1[F_1(x_1, x_2), x_3] = G_2[F_2(x_2, x_3), x_1]$ is given by

$$\begin{aligned} G_1(x_1, x_2) &= k[m(x_1) + n(x_2)] F_1(x_1, x_2) \\ &= m^{-1}[p(x_1) + q(x_2)] \end{aligned} \tag{15a}$$

$$\begin{aligned} G_2(x_1, x_2) &= k[p(x_1) + l(x_2)] F_2(x_1, x_2) \\ &= l^{-1}[q(x_1) + l(x_2)] \end{aligned} \tag{15b}$$

Substitution of Eq. (15a, 15b) in Eq. (14) gives us,

$$\begin{aligned} Y &= G_1[F_1(x_1, x_2), x_3] \\ &= G_2[F_2(x_2, x_3), x_1] \\ &= k[p(x_1) + q(x_2) + n(x_3)] \end{aligned} \tag{16}$$

The representation of FN we get from $k[p(x_1) + q(x_2) + n(x_3)]$ is given in Fig. 3b, which is equivalent to the FN in Fig. 3a as they both give the same output Y given the inputs $\{x_1, x_2, x_3\}$. Figure 3b is the generalized associativity model of the FN in Fig. 3a.

With two inputs x_1 and x_2 and an output x_3, we can obtain an associative FN as follows:

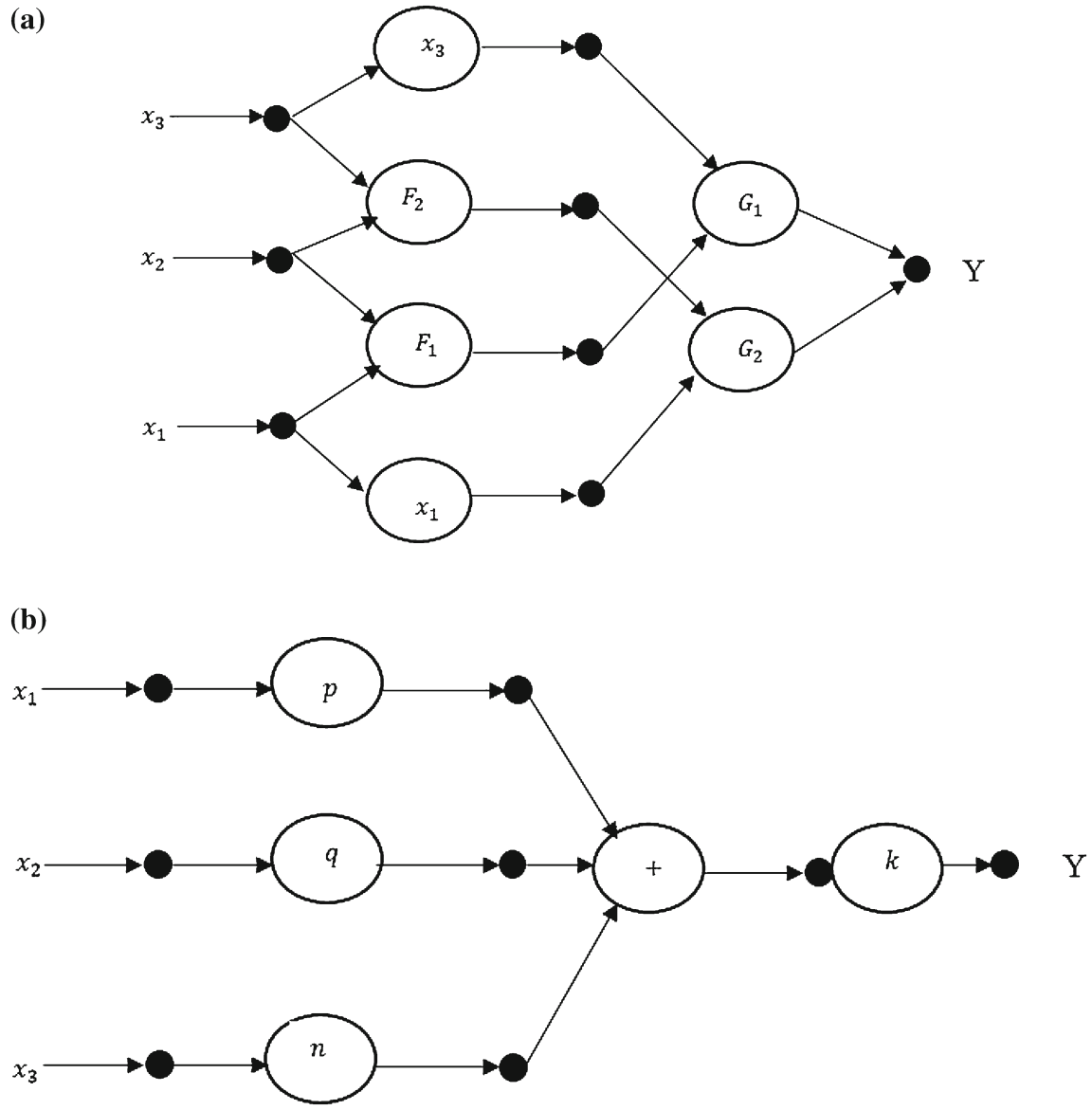

Fig. 3 A functional network and its associative model equivalent resulting from a three-input one-output problem. **a** Functional network resulting from Eq. (14). **b** Equivalent generalized associative model of functional network

$$f_s(x_s) = \sum_{i=1}^{m} a_{si}\phi_{si} \tag{17}$$

where s = number of inputs, ϕ_{si} can be polynomial, trigonometric, exponential or any acceptable function and is called as shape function and m is the degree of functions used. The function f_3 can be expressed as:

$$f_3(x_3) = \sum_{i=1}^{m} a_{3i}\phi_{3i} \tag{18}$$

From the input functions, it follows that,

$$f_3(x_3) = f_1(x_1) + f_2(x_2) \tag{19}$$

Thus, the error in the jth data is given by,

$$e_j = f_1(x_1) + f_2(x_2) - f_3(x_3) \tag{20}$$

To estimate the coefficients, $a_i, i = 1, 2, 3, \ldots m$, the sum of squared errors can be minimized as:

$$E = \sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_i[\phi_i(x_{1j}) + (\phi_i(x_{2j}) - \phi_i(x_{3j})]\right)^{2'} \tag{21}$$

$$\text{Subject to, } f(x_0) = \sum_{i=1}^{m} a_i\phi_i(x_0) = \alpha \tag{22}$$

where α is a real constant.

An auxiliary function, using the Lagrangian multiplier technique, can be built as:

$$E_\lambda = \sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_i b_{ij}\right)^2 + \lambda\left(\sum_{i=1}^{m} a_i\phi_i(x_0) - \alpha\right), \tag{23}$$

$$\text{where, } b_{ij} = \phi_i(x_{1j}) + \phi_i(x_{2j}) - \phi_i(x_{3j}) \tag{24}$$

The minimum of Eq. (23) is found from Eqs. (25) and (26).

$$\frac{\partial E_\lambda}{\partial a_r} = 2\sum_{j=1}^{n}\left(\sum_{i=1}^{m} a_i b_{ij}\right) b_{rj} + \lambda\phi_r(x_0) = 0; \quad r = 1, 2 \ldots m, \tag{25}$$

$$\frac{\partial E_\lambda}{\partial \lambda} = \sum_{i=1}^{m} a_i\phi_i(x_0) - \alpha = 0 \tag{26}$$

The above system of equations has $(m + 1)$ equations and $(m + 1)$ unknowns and can be solved to get the coefficients $a_i, i = 1, 2, 3, \ldots m$.

$$\text{In matrix form, } \begin{pmatrix} BB^T & \phi_0 \\ \phi_0^T & 0 \end{pmatrix}\begin{pmatrix} a^T \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \prime \\ \prime \\ 0 \end{pmatrix} \tag{27}$$

where B is the matrix of coefficients b_{ij} and $a = a_1, a_2, a_3, \ldots a_m$. This matrix can be written in simpler form as,

$$[B]\{u\} = \{v\} \tag{28}$$

Solving for unknowns for any given v, we get u and thus we get the coefficients $a = a_1, a_2, a_3, \ldots a_m$. For $m = 1$, a can be used to write the equation,

$$f_3(x_{3i}) = f_1(x_{1i}) + f_2(x_{2i}) = a_{31} + a_{32}x_{31} \text{or, } x_{31} = \frac{[f_3(x_{31}) - a_{31}]}{a_{32}} \tag{29}$$

2.2.4 *Example of a Two-Input, One-Output Associativity Functional Network*

Figure 4 shows an associative FN with two inputs and one output. To illustrate the use of such FN, we consider the equation,

$$x_3 = \log_{10}(x_1 + x_2) + \exp^{(x_1 \cdot x_2)} \tag{30}$$

We assume an associative FN as shown in Fig. 4. It has two inputs, x_1 and x_2 and an output x_3. A sample of 23 inputs, of randomly generated numbers between 0 and 1, was taken, and the output for each sample was calculated using Eq. (30). The input and output values were normalized between 0 to 1, and then Eq. (27) was applied to them. The following functional equations were obtained:

$$f_1(x_1) = 0.6996 - 0.4496x_1 \tag{31}$$

$$f_2(x_2) = 0.7755 - 0.3855x_2 \tag{32}$$

$$f_3(x_3) = 1.4248 - 1.0248x_3 \tag{33}$$

The value of x_3 corresponding to Eqs. (31)–(33) was calculated using Eqs. (29) and was obtained as

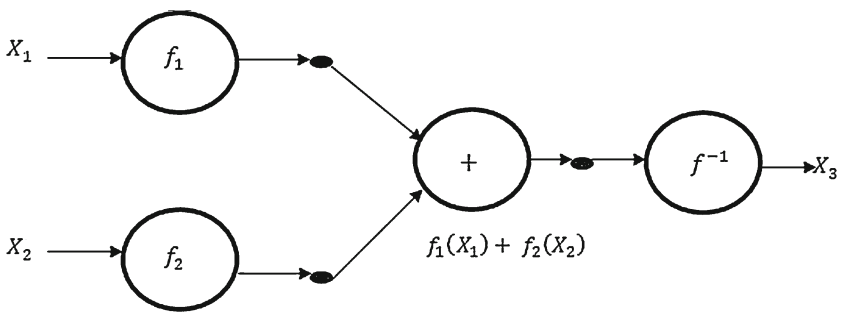

Fig. 4 An associativity functional network

Table 1 Maximum and minimum values of the data used in this study

Input	D (mm)	L (mm)	E (mm)	S_u (kPa)	Q_m (N)
Maximum	33.3	300	50	38.8	225
Minimum	6.35	130	0	3.4	29.5
Mean	17.8	278.9	44.2	9.9	72.8
Standard deviation	6.1	52.8	14.7	10.1	36.9

$$x_3 = 0.0490 + 0.4387x_1 + 0.3761x_2 \tag{34}$$

The R value between the x_3 values calculated from Eqs. (30) and (34) was found to be 0.987, showing a good prediction of the observed values by this FN. It can also be observed that the equation for x_3 obtained from FN is simpler than the original equation.

3 Database and Preprocessing

In the present study, the database of Das and Basudhar [10] has been considered. Das and Basudhar [10] have developed an ANN model with these data. Each sample has four inputs viz. diameter of pile (D), length of pile (L), eccentricity of load (e) and undrained shear strength of soil (S_u), and one output viz. measured lateral load capacity (Q_m). The variations such as maximum, minimum, mean and standard deviation values of the input and output parameters are given in Table 1.

For MARS analysis, out of the mentioned 38 data, 29 randomly selected data were selected for training and the remaining 09 data were used for testing the developed model as per Das and Basudhar [10]. However, for FN analysis, 30 randomly selected data were selected for training and 8 data were selected for testing. The data were normalized in the range [0, 1] for analysis. In this study, both MARS [39] and FN have been implemented with MATLAB [40].

4 Results and Discussion

The results of the present study are as follows. A MARS model with six BFs was adopted in this study. A MARS model with more number of BFs can give better results, but it leads to a more complex model equation. The prediction equation for the adopted MARS model can be presented as:

$$Q_p = 0.18 + 0.28 \times \mathrm{BF1} + 1.34 \times \mathrm{BF2} - 1.79 \times \mathrm{BF3} - 1.59 \times \mathrm{BF4} + 0.50 \times \mathrm{BF5} + 5.90 \times \mathrm{BF6} \tag{35}$$

where

$$\mathrm{BF1} = \max(0, 1 - e) \tag{36}$$

$$\mathrm{BF2} = \max(S_u - 0.11) \tag{37}$$

$$\mathrm{BF3} = \max(0, 0.11 - e) \tag{38}$$

$$\mathrm{BF4} = \mathrm{BF2} * \max(0, 1 - L) \tag{39}$$

$$\mathrm{BF5} = \max(0, D - 0.43) \tag{40}$$

$$\mathrm{BF6} = \max(0, 0.52 - D) \times \max(0, D - 0.26) \tag{41}$$

The value of the inputs in Eqs. (36)–(41) is the normalized value between 0 and 1. The denormalized value of Q_p is given by:

$$Q_{p\mathrm{denorm}} = Q_{p\mathrm{norm}}(225 - 29.5) + 29.5 \tag{42}$$

The FN model was created with 30 randomly selected data as the training set, and the rest of the data were used for testing. An FN model is developed by selecting an appropriate BF and its degree. A model equation is prepared from the analysis of training data, and the model is validated with testing data. Figure 5 gives a plot of R values versus the degree for polynomial, exponential, sin, cos and tan functions.

It can be seen that, for the set of data in this problem, R value for all the functions reach a maximum value near degree five and remain constant after that. Thus, taking a degree five model would provide the most accurate result, but it leads to a complex output equation. Hence, a trade-off was made in the present study and a FN model with degree three and polynomial BF was adopted. Figure 6 gives the associative FN used in this study.

The equation for the prediction of the values is obtained as follows:

$$y = a_0 + \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} f_i(x_j) \tag{43}$$

where n = no. of variables, m = degree of variable

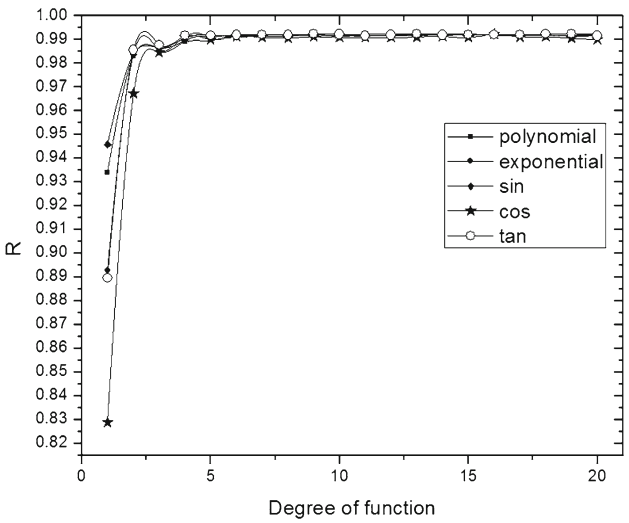

Fig. 5 Plot of R versus degree of function for FN

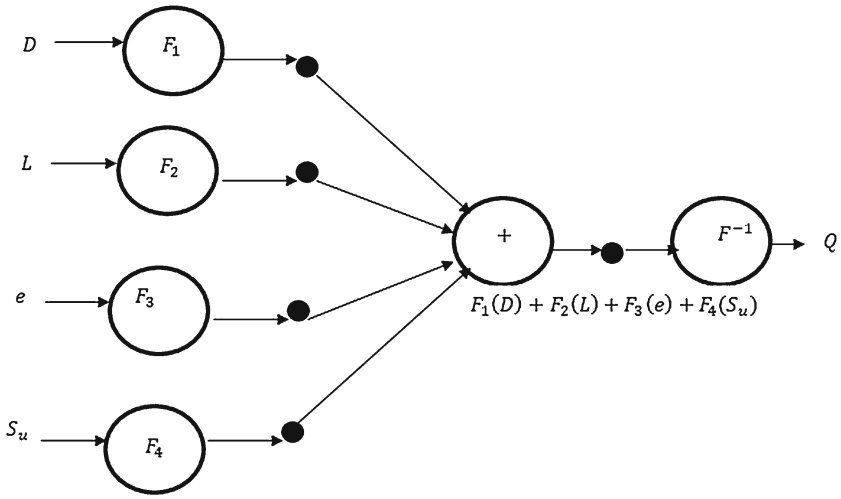

Fig. 6 Associative functional network used in this study

Here, $n = 4$ and $m = 3$, and the above Eq. (43) can be written in expanded form as:

$$a_0 = -0.434 \tag{44}$$

$$f_1(x_1) = 0.920D - 1.353D^2 \tag{45}$$

$$f_2(x_2) = 1.289L^3 \tag{46}$$

$$f_3(x_3) = -1.033e^3 \tag{47}$$

$$f_4(x_4) = 1.853S_u - 1.132S_u^2 \tag{48}$$

In Eqs. (44)–(48), the values of the inputs to be used are their normalized values between 0 and 1. The sum of values from Eqs. (44)–(48) give the normalized value of output. The denormalized value of Q_p is given by:

$$Q_{p_{denorm}} = Q_{p_{norm}}(225 - 29.5) + 29.5 \tag{49}$$

It is worth mentioning here that Eqs. (32) and (43) can be used to predict the Q for a new set of data, provided all the inputs for new set lie within the maximum and minimum range of inputs used in this study.

4.1 Comparison of Models in Terms of Statistical Parameters

Figure 7a, b shows the performance of Q_p and Q_m for MARS and FN models. It can be seen that the scatter of the data for both MARS and FN is within the 80 % prediction limit. Table 2 gives the values of different statistical parameters for both MARS and FN models.

It is observed from Table 2 that MARS and FN have approximately the same values of R for training (0.980 and 0.987, respectively) and testing data (0.991 and 0.994, respectively). Both the models have high and close values of R for training and testing data, showing a better prediction and better generalization, respectively. The data set used in this study has also been used by Pal and Deswal [22]

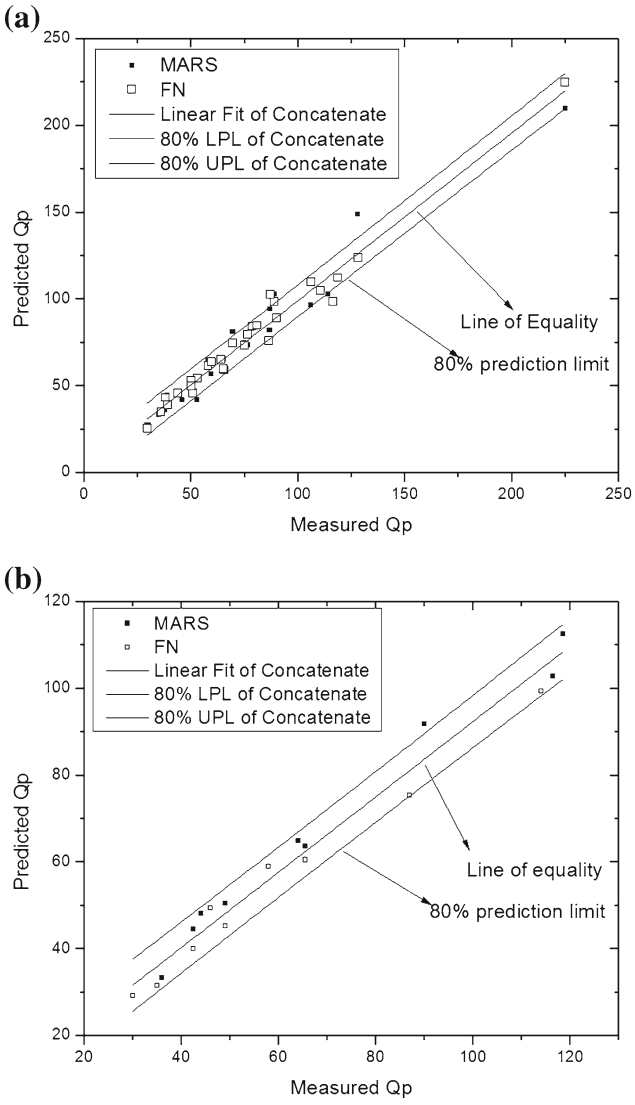

Fig. 7 Plot of measured and predicted values of lateral load capacity of pile for **a** training and **b** testing data for MARS and FN models

Table 2 Statistical parameters for MARS and FN, DENN, BRNN, Hansen and Broms models

Models	Statistical performances				
	R	*E*	AAE	MAE	RMSE
FN					
Training	0.986	0.973	4.697	17.824	6.277
Testing	0.99	0.928	5.135	14.611	6.832
MARS					
Training	0.98	0.96	6.199	20.899	7.594
Testing	0.991	0.968	7.239	13.626	5.091
ANN					
Training	0.98	0.959	5.647	18.705	7.667
Testing	0.967	0.905	7.17	18.11	8.549
Hansen					
Training	0.95	0.209	30.712	65.36	33.825
Testing	0.919	0.119	23.65	49.48	26.066
Broms					
Training	0.967	0.807	12.391	48.66	16.703
Testing	0.985	0.574	12.082	46.38	18.127

to develop GPR and SVM models. The value of R for the GPR and SVM model was 0.980 and 0.920, respectively, for the testing data. Das and Basudhar [10] obtained R value of 0.987 and 0.947 for the training and testing data, respectively, using Levenberg–Marquardt (LM) algorithm, ANN. However, as per Das et. al. [12], differential evolution algorithm is better than LM neural network. So, in this study, ANN was implemented using a differential evolution training algorithm. Thus, based on the *R* value, the performance of MARS and FN is better than of ANN (0.980 for training and 0.991 for testing), Broms (0.967 for training and 0.985 for testing) and the Hansen (0.950 for training and 0.919 for testing) model.

It is also known that *R* is a biased estimate [41]; hence, the results were also compared in terms of Nash–Sutcliff coefficient of efficiency (*E*) [37]. *E* is defined as

$$E = \frac{E_1 - E_2}{E_1} \tag{50}$$

where

$$E_1 = \sum_{t=1}^{n} \left(Q_\mathrm{m} - \overline{Q_\mathrm{m}}\right)^2 \tag{51}$$

$$E_1 = \sum_{t=1}^{n} \left(Q_\mathrm{p} - Q_\mathrm{m}\right)^2 \tag{52}$$

And Q_m, $\overline{Q_\mathrm{m}}$, Q_p are the measured, average and predicted lateral load capacity of pile. The *E* value compares the modelled and measured values of the variable and evaluates how far the network is able to explain total variance in the data set. Table 2 provides the *E* value for MARS and FN and their comparison with other models. The *E* value for training in FN (0.970) was better than the E value for MARS (0.960). However, the E value for testing in FN (0.925) is less the than *E* value for MARS (0.968). The *E* value for MARS and FN is better than for ANN (0.959 for training and 0.905 for testing), Broms (0.807 for training and 0.574 for testing) and Hansen (0.209 for training and 0.119 for testing). Only *E* for FN in testing was found to have a lower value than ANN. Table 2 also compares the MAE, AAE and RMSE for training and testing data sets for MARS, FN and other prediction models. Based upon MAE, AAE and RMSE values, the performance of MARS and FN is better than of ANN, Broms and Hansen models. However, in this case also the MAE and RMSE for FN testing data (14.581 and 7.239, respectively) were found to be more than those for MARS testing data (13.626 and 5.091, respectively).

The mean (μ) and standard deviation (σ) of $Q_\mathrm{p}/Q_\mathrm{m}$ are important indicators of the accuracy and precision of the prediction method. Under ideal conditions, an accurate and precise method gives the mean value as 1.0 and the standard deviation to be 0. A μ value >1.0 indicates over prediction and under prediction if <1. Cumulative probability of the $Q_\mathrm{p}/Q_\mathrm{m}$ has also been considered for the assessment of the model. The ratio $Q_\mathrm{p}/Q_\mathrm{m}$ is arranged in ascending order, and the cumulative probability is calculated from the formula:

$$P = \frac{i}{n+1} \tag{53}$$

where i is the order number given to the $Q_\mathrm{p}/Q_\mathrm{m}$ ratio and n is the number of data points. If the computed value of 50 % cumulative probability (P_{50}) is less than unity, under prediction is implied; values greater than unity means over prediction. The 'best' model is corresponding to the P_{50} value close to unity. The 90 % cumulative probability (P_{90}) reflects the variation in the ratio of $Q_\mathrm{p}/Q_\mathrm{m}$ for the total observations. The model with $Q_\mathrm{p}/Q_\mathrm{m}$ close to 1.0 is the better model.

Figure 8 shows the variation of $Q_\mathrm{p}/Q_\mathrm{m}$ with cumulative probability (%) for MARS and FN models. Table 3 gives the P_{50} and P_{90} values for training and testing data for MARS, FN and other prediction models.

From Table 3, it can be observed that the P_{50} and P_{90} values for testing in MARS (1.016 and 1.196, respectively), FN (0.920 and 1.009, respectively) and ANN (0.945 and 1.161, respectively) are close to 1 indicating a good prediction. The values for FN for P_{90} are marginally better than those for MARS and ANN. In comparison with other models, the values of P_{50} and P_{90} for testing in Broms (1.140 and 1.392, respectively) and Hansen (0.523 and 0.838, respectively) are not close to 1 showing a better performance for MARS and FN in terms of cumulative probability. Figure 9 gives the log-normal distribution of $Q_\mathrm{p}/Q_\mathrm{m}$ for MARS and FN.

Figure 9 shows that 100 % of the data in both training and testing lie within ±20 % accuracy. Table 3 provides a comparison of log-normal distribution between MARS, FN and other prediction models. It is observed that only ANN

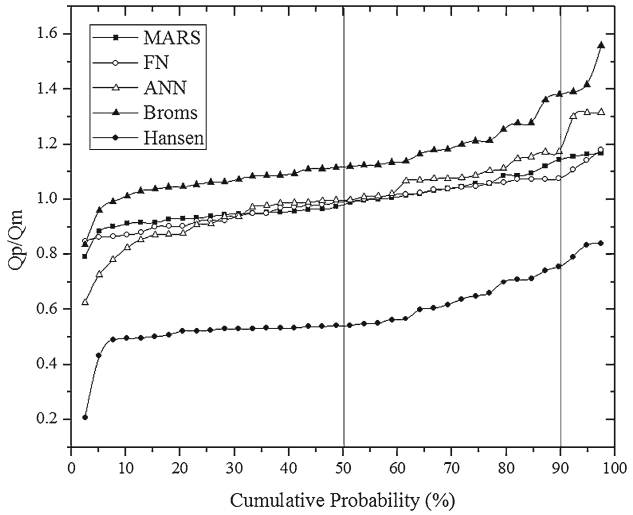

Fig. 8 Plot of cumulative probability of Q_p/Q_m for MARS, FN, ANN, Broms and Hansen models

has a satisfactory set of data lying between ±20 % accuracy (90 % for training and 84 % for testing).

4.2 Ranking of Models

As discussed above, it was observed that the performance of different prediction models was found to vary based on different statistical parameters. Hence, a ranking system as suggested by Abu-Farsakh and Titi [38] has been employed to assess the overall performance of the MARS, FN, ANN, Hansen and Broms. The RI is the sum of four individual rank criterion, i.e.

$$\mathrm{RI} = R1 + R2 + R3 + R4, \tag{54}$$

where $R1$ is based on best fit calculations (values of R and E), $R2$ is based on arithmetic calculations of Q_p/Q_m (average, μ and standard deviation, σ), $R3$ is based on cumulative probability of Q_p/Q_m($P50$ and $P90$) and $R4$ is based on prediction of lateral load capacity within ±20 % accuracy. The ranking system for different models is presented in Table 3. A lower value of RI indicates better performance of a model. Thus, it can be seen from Table 3 that FN (RI = 5) has the best performance among all models. FN is closely followed by MARS (RI = 7) as the second best model. The other models in the order of ranking are ANN (RI = 12), Broms (RI = 17) and Hansen (RI = 19).

4.3 Sensitivity Analysis

The sensitivity analysis is an important aspect of a developed model to find out important input parameters. In the present study, the sensitivity analysis was carried according to Gandomi et al. [42]. According to this concept, the sensitivity

Table 3 Evaluation of performance of different prediction models and their ranking based on rank index proposed by Abu-Farsakh and Titi [38]

Models	Best fit calculations			Arithmetic calculations of Q_p/Q_m			Cumulative probability			±20 % Accuracy (%)			Overall rank	
	R	E	$R1$	μ	σ	$R2$	Q_p/Q_m at P_{50}	Q_p/Q_m at P_{90}	$R3$	Log-normal	Histogram	$R4$	RI	Final rank
MARS														
Training	0.980	0.960	2	1.000	0.093	1	0.964	1.156	2	97	97	2	7	1
Testing	0.991	0.968		0.994	0.066		1.016	1.196		100	100			
FN														
Training	0.986	0.973	1	1.004	0.081	2	1.007	1.109	1	100	100	1	5	1
Testing	0.990	0.928		0.943	0.068		0.924	1.074		100	100			
ANN														
Training	0.980	0.959	3	1.018	0.106	3	1.012	1.156	3	90	92	3	12	2
Testing	0.967	0.905		0.948	0.125		0.945	1.161		84	88			
Hansen														
Training	0.950	0.209	5	0.580	0.111	5	0.542	0.741	4	0	0	5	19	4
Testing	0.919	0.119		0.590	0.149		0.523	0.838		8	22			
Broms														
Training	0.967	0.807	4	1.143	0.144	4	1.112	1.382	5	64	72	4	17	3
Testing	0.985	0.574		1.166	0.136		1.140	1.392		50	66			

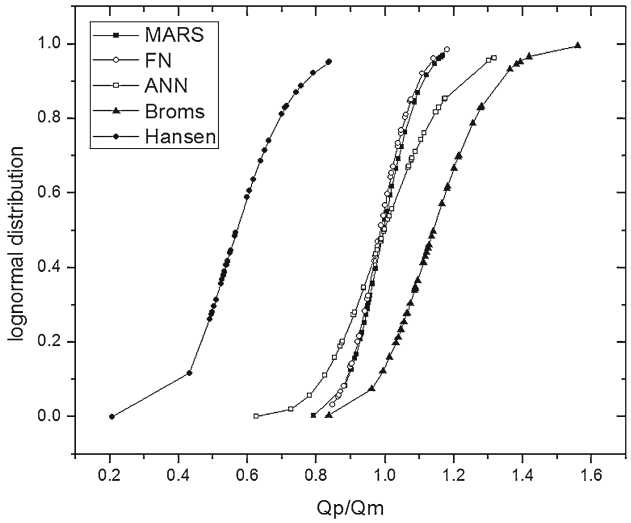

Fig. 9 Log-normal distribution of Q_p/Q_m for MARS, FN, ANN, Broms and Hansen models

of output respective to each input parameter is calculated by the use of Eqs. (55) and (56) according to which the value of the said input is varied, while the value of all other inputs are kept constant.

$$N_i = f_{\max}(x_i) - f_{\min}(x_i) \tag{55}$$

$$S_i = N_i \Big/ \sum_{i=1}^{n} N_i \tag{56}$$

where $f_{\max}(x_i)$ and $f_{\min}(x_i)$ are the maximum and minimum of the predicted output over the ith input domain, respectively, where the other variables are equal to their mean values. n is the number of variables. In the present study $n = 4$. Table 4 shows the sensitivity analysis of inputs for MARS and FN models.

For the MARS model, S_u is identified as the most important input followed by D, e and L. For FN model, L is identified as the most important input followed by e, S_u and D.

4.4 Validation of MARS and FN Modelling

To validate the MARS and FN modelling in predicting the lateral capacity of a pile accurately, two different models were developed using the data set of Rao and Suresh Kumar [43]. Out of 42 data, 34 and 8 data were used for training and testing, respectively, for both MARS and FN modelling.

A MARS model with 7 BFs was adopted with R value of 0.993 and 0.972 in training and testing, respectively. The corresponding equation can be given by:

$$\begin{aligned} Q_p = {} & 0.22 + 0.73 * \mathrm{MAX}(0, S_u - 0.22) \\ & - 0.88 * MAX(0, 0.22 - S_u) \\ & + 0.65 * \mathrm{MAX}(0, L - 0.03) \\ & - 8.97 * \mathrm{MAX}(0, 0.03 - L) \\ & - 0.55 * \mathrm{MAX}(0, D - 0.02) \\ & + 3.44 * \mathrm{MAX}(0, 0.03 - e) \\ & - 4.82 * \mathrm{MAX}(0, 0.01 - D) \end{aligned} \tag{57}$$

Equation (57) gives the normalized value of Q_p and the actual value can be calculated using Eq. (58).

$$Q_{p\text{denorm}} = Q_{p\text{norm}}(900 - 29.5) + 29.5 \tag{58}$$

The FN model was developed with degree 3 and polynomial BF. The R value of this model was found to be 0.999 and 0.986 for training and testing, respectively. The predictive equation for the developed FN model can be given by:

$$\begin{aligned} Q_p = {} & \big(-0.112 + 1.853D - 21.953D^2 - 25.327D^3 \\ & + 5.491L - 1.882L^3 - 3.137e + 23.345e^2 \\ & - 21.9e^3 + 0.981S_u - 3.85S_u^3\big) \end{aligned} \tag{59}$$

Equation (59) gives the normalized value of Q_p and the actual value can be calculated using Eq. (58).

To validate the developed model, two experimental data that were not used for training or testing were validated to get the efficiency of both the models. Equations (57) and (58) were used to calculate Q_p for two new data for to validate the MARS and FN models, respectively, as shown in Table 5.

It can be seen that for the first data the percentage errors were 15.4 and 6.60 % for the MARS and FN model, respectively. For the second data, the percentage error for MARS and FN were 12.83 and 3.80 %, respectively. Thus, it can be inferred that both MARS and FN can fairly accurately predict the value of Q for a new data set. As discussed earlier,

Table 4 Sensitivity analysis of inputs for MARS and FN models

Inputs	FN		MARS	
	Sensitivity (%)	Ranking of input as per sensitivity	Sensitivity (%)	Ranking of input as per sensitivity
D	33.184	2	38.724	2
L	29097.170	1	6.070	4
e	−23336.400	4	192.660	1
S_u	1.949	3	34.712	3

Table 5 Validation data used to test the efficiency of the developed MARS and FN Models

D (mm)	L (mm)	e (mm)	S_u (kPa)	Q_m (N)	MARS		FN	
					Q_p (N)	% Err	Q_p (N)	% Err
13.5	300	50	5.5	50	42.28	15.43	46.70	6.60
13.5	300	50	7.2	64	55.79	12.83	61.56	3.81

in this case also, FN was found to be a better prediction tool as compared to MARS.

However, the MARS and FN models developed in this study were based upon the results obtained from experiments done on scaled models in laboratory. Appropriate dimensional analysis and scaling effects have to be taken into consideration to apply the results in actual field practice.

5 Conclusions

In this paper, an attempt has been made to develop prediction models for the lateral load capacity of piles in clay using two recently developed statistical modelling methods: MARS and FN. Based on the results of MARS and FN models and the discussion that follows, the following conclusions can be drawn:

(a) According to a comparison done on the basis of statistical parameters (R, E, MAE, AAE, RMSE) and properties of Q_p/Q_m (μ, σ, cumulative distribution, log-normal distribution), the performance of MARS and FN is better than other AI techniques such as ANN and empirical methods such as Hansen and Broms.
(b) A RI technique previously presented in the literature was used to assess the overall performance of each technique. According to RI, FN is the best prediction technique followed by MARS, ANN, Broms and Hansen.
(c) A sensitivity analysis was carried out to assess the relative importance of inputs in each model. For the MARS model, S_u is the most important input, whereas for FN model, L is the most important input. The identification of S_u as the most important input for the MARS model is in agreement with Das and Basudhar [10], whereas S_u was found to be the most important input for ANN model.
(d) Model equations for both MARS and FN are given, which can be used by practicing geotechnical engineers to predict the lateral load capacity of pile in clay to a fair degree of accuracy when the appropriate field data are available.
(e) The developed model is based on only the data set pertaining to laboratory investigation results.

References

1. Hansen, B.: The ultimate resistance of rigid piles against transversal force. Copenhagen, Danish Geotechnical Institute, Bulletin no. 12., pp. 5–9 (1961)
2. Broms, B.B.: Lateral resistance of piles in cohesive soils. J. Soil Mech. Found Engg. ASCE **90**((SM. 2), 27–63 (1964a)
3. Broms, B.B.: Lateral resistance of piles in cohesionless soils. J. Geotech. Eng. **90**, 123–156 (1964b)
4. Poulos, H.G.; Davis, E.H.: Pile foundation analysis and design. Wiley, New York (1980)
5. Matlock, H.; Reese, L.C.: Generalized solutions for laterally loaded piles. Trans ASCE **127**, 1220–1248 (1962)
6. Portugal, J.C.; Seco e Pinto, P.S.: Analysis and design of pile under lateral loads.In: Proceedings of the 11th international geotechnical seminar on deep foundation on bored and auger piles, Belgium, pp. 309–313 (1993)
7. Muthukkumaran, K.; Sundaravadivelu, R.; Gandhi, S.R.: Effect of slope on p–y curves due to surcharge load. Soils Found **48**(3), 353–361 (2008)
8. Begum, N.A.; Muthukkumaran, K.: Experimental investigation on single model pile in sloping ground under lateral load. Int. J. Geotech. Eng. **3**(1), 133–146 (2009)
9. Muthukkumaran, K.: Effect of slope and loading direction on laterally loaded piles in cohesionless soil. Int. J. Geomech. ASCE **14**(1), 1–7 (2014)
10. Das, S.K.; Basudhar, P.K.: Undrained lateral load capacity of piles in clay using artificial neural network. Comput. Geotech. **33**, 454–459 (2006)
11. Hamid, M.; Reza, R.: The estimation of rock mass deformation modulus using regression and artificial neural networks analysis. Arab. J. Sci. Eng. **35**(1A), 205–217 (2010)
12. Das, S.K.; Biswal, R.K.; Sivakugan, N.; Das, B.: Classification of slopes and prediction of factor of safety using differential evolution neural networks. Environ. Earth. Sci., Springer **64**, 201–210 (2011)
13. Muduli, P.K.; Das, M.R.; Samui, P.; Das, S.K.: Uplift capacity of suction caisson in clay using artificial intelligence techniques. Mar. Georesour. Geotechnol. **31**(4), 375–390 (2013)
14. Tarawneh, B.; Imam, R.: Regression versus artificial neural networks: predicting pile setup from empirical data. KSCE J. Civ. Eng. **18**(4), 1018–1027 (2014)
15. Goh, A.T.C.: Empirical design in geotechnics using neural networks. Geotechnique **45**(4), 709–714 (1995)
16. Goh, A.T.C.: Pile driving records reanalyzed using neural networks. J. Geotech. Eng., ASCE **122**(6), 492–495 (1996)
17. Chan, W.T.; Chow, Y.K.; Liu, L.F.: Neural network: an alternative to pile driving formulas. J. Comput. Geotech. **17**, 135–156 (1995)
18. Lee, I.M.; Lee, J.H.: Prediction of pile bearing capacity using artificial neural networks. Comput. Geotech. **18**(3), 189–200 (1996)
19. Teh, C.I.; Wong, K.S.; Goh, A.T.C.; Jaritngam, S.: Prediction of pile capacity using neural networks. J. Comput. Civ. Eng., ASCE **11**(2), 129–138 (1997)
20. Abu-Kiefa, M.A.: General regression neural networks for driven piles in cohesionless soils. J. Geotech. Geoenviron. Eng., ASCE **124**(12), 1177–1185 (1998)
21. Samui, P.: Prediction of friction capacity of driven piles in clay using the support vector machine. Can. Geotech. J. **45**(2), 288–295 (2008)
22. Pal, M.; Deswal, S.: Modelling pile capacity using Gaussian process regression. Comput. Geotech. **37**, 942–947 (2010)

23. Alkroosh, I.; Nikraz, H.: Evaluation of pile lateral capacity in clay applying evolutionary approach. Int. J. GEOMATE **4**(1), 462–465 (2013)
24. Giustolisi, O.; Doglioni, A.; Savic, D.A.; Webb, B.W.: A multi-model approach to analysis of environmental phenomena. Environ. Model. Softw. **22**(5), 674–682 (2007)
25. Friedman, J.: Multivariate adaptive regression splines. Ann. Stat. **19**, 1–141 (1991)
26. Samui, P.; Das, S.; Kim, D.: Uplift capacity of suction caisson in clay using multivariate adaptive regression spline. Ocean Eng. **38**(17–18), 2123–2127 (2011)
27. El-Sebakhy, E.A.; Asparouhov, O.; Abdulraheem, A.; Al-Majed, A.; Wu, D.; Latinski, K.; Raharja, I.: Functional networks as a new data mining predictive paradigm to predict permeability in a carbonate reservoir. Expert Syst. Appl. **39**, 10359–10375 (2012)
28. Castillo, E.; Cobo, A.; Gutierrez, J.M.; Pruneda, R.E.: Working with differential, functional and difference equations using functional networks. Appl. Math. Model. **23**, 89–107 (1999)
29. El-Sebakhy, E.A.; Faisal, K.A.; Helmy, T.; Azzedin, F.; Al-Suhaim, A.: Evaluation of breast cancer tumor classification with unconstrained functional networks classifier. In: Proceeding of the 4th ACS/IEEE international conference on computer systems and applications, pp. 281–287 (2006)
30. Rajasekaran, S.: Functional networks in structural engineering. J. Comput. Civ. Eng. **18**, 172–181 (2004)
31. Attoh-Okine, N.O.: Modeling incremental pavement roughness using functional network. Can. J. Civ. Eng. **32**, 899–905 (2005)
32. Castillo, E.: Functional networks. Neural Process. Lett. **7**, 151–159 (1998)
33. Castillo, E.; Ruiz-Cobo, R.: Functional equations in science and engineering. Marcel Dekker, New York (1992)
34. Castillo, E.; Cobo, A.; Gutierrez, J.M.; Pruneda, E.: An introduction to functional networks with applications. Kluwer, Boston (1998)
35. Castillo, E.; Cobo, A.; Manuel, J.; Gutierrez, J.M.; Pruneda, E.: Functional networks: a new network-based methodology. Comput. Aided Civ. Infrastruct. Eng. **15**, 90–106 (2000a)
36. Castillo, E.; Cobo, A.; Gomez-Nesterkin, R.; Hadi, A.S.: A general framework for functional networks. Networks **35**(1), 70–82 (2000b)
37. Das, S.K.; Basudhar, P.K.: Prediction of residual friction angle of clays using artificial neural network. Eng. Geol. **100**(3–4), 142–145 (2008)
38. Abu-Farsakh, M.Y.; Titi, H.H.: Assessment of direct cone penetration test methods for predicting the ultimate capacity of friction driven piles. J. Geotech. Geoenviron. Eng. **130**(9), 935–944 (2004)
39. Jekabsons, G.; ARESLab: Adaptive regression splines toolbox for Matlab/Octave. http://www.cs.rtu.lv/jekabsons/ (2011)
40. MathWork Inc.: Matlab User's Manual, Version 6.5. Natick (MA).(2005)
41. Das, S.K.; Sivakugan, N.: Discussion of: intelligent computing for modeling axial capacity of pile foundations. Can. Geotech. J. **47**, 928–930 (2010)
42. Gandomi, A.H.; Yun, G.J.; Alavi, A.H.: An evolutionary approach for modeling of shear strength of RC deep beams. Mater. Struct. (2013) doi:10.1617/s11527-013-0039-z
43. Rao, K.M.; Suresh Kumar, V.: Measured and predicted response of laterally loaded piles. In: Proceedings of the sixth international conference and exhibition on piling and deep foundations, India, pp. 1.6.1–1.6.7 (1996)