Geotech Geol Eng (2020) 38:2271–2291
https://doi.org/10.1007/s10706-019-01085-8

TECHNICAL NOTE

Estimation of Bearing Capacity of Piles in Cohesionless Soil Using Optimised Machine Learning Approaches

Navid Kardani · Annan Zhou · Majidreza Nazem · Shui-Long Shen

Received: 16 August 2019 / Accepted: 9 October 2019 / Published online: 29 November 2019

Abstract Accurate estimation of the bearing capacity of piles requires complex modelling techniques which are not justified by timeframe, budget, or scope of the projects. In this study, six advanced machine learning algorithms including decision tree, k-nearest neighbour, multilayer perceptron artificial neural network, random forest, support vector regressor and extremely gradient boosting are employed to model the bearing capacity of piles in cohesionless soil, and the particle swarm optimisation algorithm is used to optimate the hyper-parameters of machine learning algorithms. A dataset comprising of 59 cases is employed and the R-squared value, root mean square error and variance accounted for are used as performance metrics to compare the performance of optimised machine learning methods. The comparison reveals that the optimised machine learning methods have great potential to estimate bearing capacity of piles and the particle swarm optimisation algorithm is efficient in the hyper-parameter tuning. The results show that R-squared values of six optimised machine learning approaches on the testing set vary from 0.731 to 0.9615. Also, the optimised extremely gradient boosting (R-squared value = 0.9615) shows the best performance compared with other algorithms. Furthermore, the relative importance of influential variable is investigated, which shows that effective stress is the most influential variable for bearing capacity of piles with an importance score of 30.9%. In addition, the results by the optimised machine learning method are compared to the β-method which is a popular empirical method. It is revealed the prominent performance of optimised machine learning approaches.

Keywords Optimised machine learning algorithms · Particle swarm optimisation algorithm · Bearing capacity of piles · Relative variable importance

N. Kardani · A. Zhou (✉) · M. Nazem · S.-L. Shen
Civil and Infrastructure Engineering Discipline, School of Engineering, Royal Melbourne Institute of Technology (RMIT), Victoria 3001, Australia
e-mail: annan.zhou@rmit.edu.au

N. Kardani
e-mail: navid.kardani@rmit.edu.au

M. Nazem
e-mail: majidreza.nazem@rmit.edu.au

S.-L. Shen
e-mail: shensl@stu.edu.cn

S.-L. Shen
Department of Civil and Environmental Engineering, College of Engineering, Shantou University, Shantou 515063, Guangdong, China

1 Introduction

Conducting a parametric study of bearing capacity of bored and driven piles by utilising static and dynamic

load tests is time-consuming and costly (Abu-Farsakh and Titi 2004; Randolph and Murphy 1985). Furthermore, owing to inherent soil variability (Wu et al. 2018; Zhang et al. 2018), pile installation disturbance and limitation of the scaling effects at the model scale, uncertainty and complexity should always be considered when estimating the bearing capacity of piles by load tests. This fact has given great stimulus for many geotechnical engineers and researchers to conduct alternative methods to estimate the bearing capacity of piles (Cao and Zhou 2019). Pile driven formulas which are widely used in practice are amongst the researchers' interests in this area; however, the accuracy of these formulas is relatively low in general, showing unsatisfactory correlations with load test results (Randolph 2003; Teh et al. 1997). Another substitute is utilising the stress-wave matching technique to estimate pile capacity from dynamic test data. Nevertheless, this technique still has some drawbacks such as the time-consuming process due to the presence of many degrees of freedom in the problem. Furthermore, measurements of stress and acceleration in the field are prone to noise and error contamination (Chow et al. 1988; Gao et al. 2015; Jin et al. 2018; Svinkin 2002).

Due to their efficiency, artificial intelligence (AI) methods have recently been applied to various complex problems to find nonlinear relations amongst several parameters (Das and Basudhar 2006; Ma et al. 2017; Muduli et al. 2013; Yao et al. 2015), making them appropriate alternatives for empirical parametric studies. Since evaluating the bearing capacity of driven piles is a very challenging problem, using AI algorithms can be introduced as a proper solution. Over the past few years, numerous studies have been carried out to estimate the bearing capacity of piles by using machine learning (ML) approaches. For example, Kiefa (1998) applied the general regression neural network (GRNN) that contains three layers (one input, one hidden and one output) to estimate the vertical load-bearing capacity of driven piles in cohesionless soil. It was found that the proposed GRNN model showed better performance in comparison with empirical models previously proposed by Randolph and Murphy (1985) and Coyle and Castello (1981). Recently, Pal and Deswal (2010) utilised the Gaussian process regression to model pile capacity. The results from their study indicated that the Gaussian process regression can provide an improved estimation compared with empirical relations. Alkroosh and Nikraz (2011) employed the gene expression programming to estimate pile axial capacity and then an evolutionary algorithm to predict pile dynamics capacity (Alkroosh and Nikraz 2014). Samui (2011) employed support vector machine (SVM) as a predictive tool for pile bearing capacity and showed that SVM outperforms the artificial neural network (ANN). Samui (2012a) also estimated the ultimate capacity of driven piles in cohesionless soils by using relevance vector machines (RVM) and multivariate adaptive regression splines (MARS) (Samui 2012b). Muduli et al. (2013) and Suman et al. (2016) adopted extreme learning machine (ELM) and functional networks (FN) to respectively investigate lateral load capacity and frictional capacity of piles. Momeni et al. (2014) proposed a hybrid model by using a genetic algorithm (GA) along with an artificial neural network (ANN) to estimate the bearing capacity of piles. Zhang and Goh (2016) applied back propagation neural network (BPNN) and multivariate adaptive regression splines (MARS) on the dataset of 4072 samples (17 input variables) to estimate the pile drivability, which showed that MARS had a better performance than BPNN. Armaghani et al. (2017) used PSO-ANN model to predict the bearing capacity of pile, and showed that it has a superior performance compared with the individual ANN. Moayedi and Armaghani (2018) employed ANN and imperialism competitive algorithm (ICA) to estimate the bearing capacity of piles. Moayedi and Hayati (2018), Moayedi et al. (2019) adopted various evolutionary neural network (ENN) models to predict the ultimate bearing capacity.

As evidenced above, various studies have been performed to evaluate the predictive performance of ML algorithms for bearing capacity of piles. However, several problems still need to be addressed properly; (1) only a limited number of advanced ML methods have been utilised in pile bearing capacity estimation and the feasibility of other methods such as extreme gradient boosting (XGB) and random forest (RF) have not been widely explored, (2) it is necessary to tune the hyper-parameters of ML algorithms properly before implementing them on the datasets, while the capability of metaheuristic optimisation algorithms such as particle swarm optimisation (PSO) in tunning hyper-parameters have not been fully explored on pile bearing capacity dataset, and (3) a systematic,

comprehensive comparative study of the available ML methods is still required. While applied to estimation of bearing capacity of piles, the performance of advanced ML methods may substantially be different, demanding further investigation.

By a comparative manner, this study implements six optimised ML (OML) methods by using the python programming language to address the research gap in the literature regarding the bearing capacity of piles. These six ML algorithms consist of decision tree (DT), K-nearest neighbour (KNN), multilayer perceptron artificial neural network (MLPANN), random forest (RF), support vector machine (SVM) and extreme gradient boosting (XGB). In addition, particle swarm optimisation (PSO) is used to determine the hyper-parameters of these ML algorithms. In this study, a dataset based on the study of Mohanty et al. (2018), which consists of 59 multinational data samples, was used. Soil shear resistance angle at the shaft of the pile, (φ_s), soil shear resistance angle at the tip of the pile φ_t, length of pile (L), cross-sectional area of pile (A) and effective overburden pressure at the tip of the pile (σ'_v) are the input parameters of the models and will be explained in the data preparation section. R-squared value (R^2), root mean square error ($RMSE$) and variance accounted for (VAF) are used as performance metrics to compare the performance of six ML methods. In addition, the relative importance of influential variable is also investigated for bearing capacity of piles. This research can be used for pile bearing capacity estimation using soft computing methods.

2 Methodology

In this study, six OML algorithms are used to investigate the relationship between the bearing capacity of the driven pile and its influencing variables while PSO algorithm was utilised to tune the hyper-parameters for these methods. In this section, we briefly introduce the six ML algorithms and the PSO algorithm.

2.1 Machine Learning Algorithms

2.1.1 Decision Tree

Decision tree (DT) is a supervised ML method in which models are built in a flow–chart-like structure (see Fig. 1). The DT is built using recursive partitioning which is generally known as divide and conquer approach. As shown in Fig. 1, a DT consists of two or more branches, in which a leaf node represents a classification or decision. The first decision node in a DT which corresponds to the best predictor is called the root node. It is notable that a relatively large DT increases the risk of overfitting towards the training data and poorly predicting towards new samples, whereas a relatively small DT may not capture significant structural information about the sample space. This problem is referred to as the horizon effect. One of the widely used strategies is to continue growing the DT to the point that each node contains a small number of examples and then to conduct pruning without additional information. The main purpose of pruning is to reduce the size of a decision tree without decreasing predictive accuracy as measured by a cross-validation set. There are many techniques for DT pruning that differ in the measurements used to optimise performance (Rokach and Maimon 2008).

2.1.2 K-Nearest Neighbour

As a non-parametric supervised method, K-nearest neighbour (KNN) can be used in classification as well as regression problems. The algorithm works based on the feature similarity, viz., a value is assigned to the new point based on how close it is to the points in the training set. The KNN is a type of lazy learning or instance-based learning in which function works as a local approximation tool and all computation is deferred until classification (Altman 1992). Three mains steps to perform KNN are summarised as follows:

1. Calculate the distance between the point to be valued and its neighbours by one of the distance functions including Minkowski, Manhattan and Euclidean.
2. Select the K data points with the lowest distance (K nearest data items).

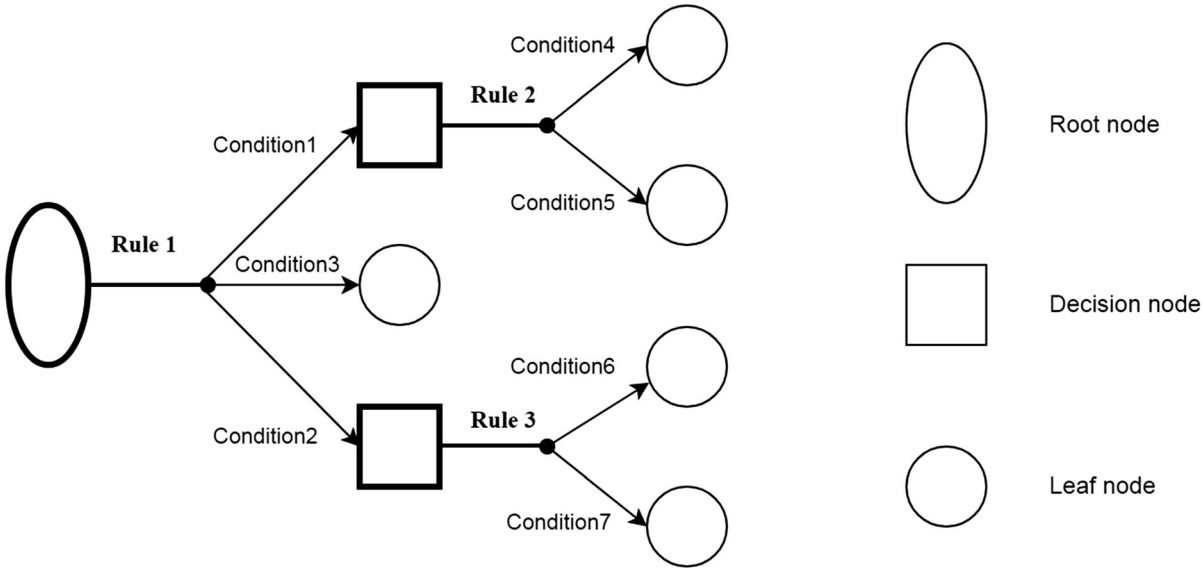

Fig. 1 Scheme illustration of a DT

3. Conduct a "majority vote" among those data points, and the dominating values in that pool are decided as the final values.

K is a key hyper-parameter used in the KNN. Hyper-parameter techniques and inspecting data are some of the available methods to choose the optimal value of K (Garcia et al. 2011).

2.1.3 Multilayer Perceptron Artificial Neural Network

As a data-driven method inspired by a brain structure, artificial neural network (ANN) uses different layers of mathematical processing to learn from observing datasets, which can be used as a classifier, regressor and random function approximation tool. As shown in Fig. 2, the multilayer perceptron artificial neural network (MLPANN or MLP for short) is a feedforward ANN which consists of one input layer, one or more hidden layer and one output layer (Das and Sivakugan 2010). The input layer receives the signal, hidden layers are the computational engine of the system and the output layer makes the prediction based on the input. As shown in Fig. 2, weights (e.g. W in Fig. 2) and biases (e.g. b in Fig. 2) are the two important parameters in the MLP. Weights define the interconnected relationships between layer neurons and biases are used to define the degree of freedom for the systems. Each node, except for the input nodes, uses a nonlinear transfer function (also known as activation function) to define the output of that node, or neuron, given an input or set of inputs. This output is then used as input for the next node and so on until a desired solution to the original problem is found. The backpropagation algorithm is then used to calculate the error which is the result of comparing the expected outcome and the outcome of the network. This error is then propagated back through the network, one layer at a time, and the weights are updated according to their contribution to the error (Baghban et al. 2018; Zhang et al. 2019).

2.1.4 Random Forest

As an ensemble learning method based on the divide-and-conquer approach, random forest (RF) is widely used for real-world classification and regression problems. The RF uses bootstrap aggregation technique (also known as bagging method) to ensemble base learners (i.e. unpruned DTs) which are different from each other and are built randomly. The key point of the RF is to build a collection of DTs with controlled variations. Another important aspect of the process of training is a random feature selection. The algorithm selects a subset comprised of M influencing variables for each node of each tree. To ensure the variety of DTs, selected influencing variables should

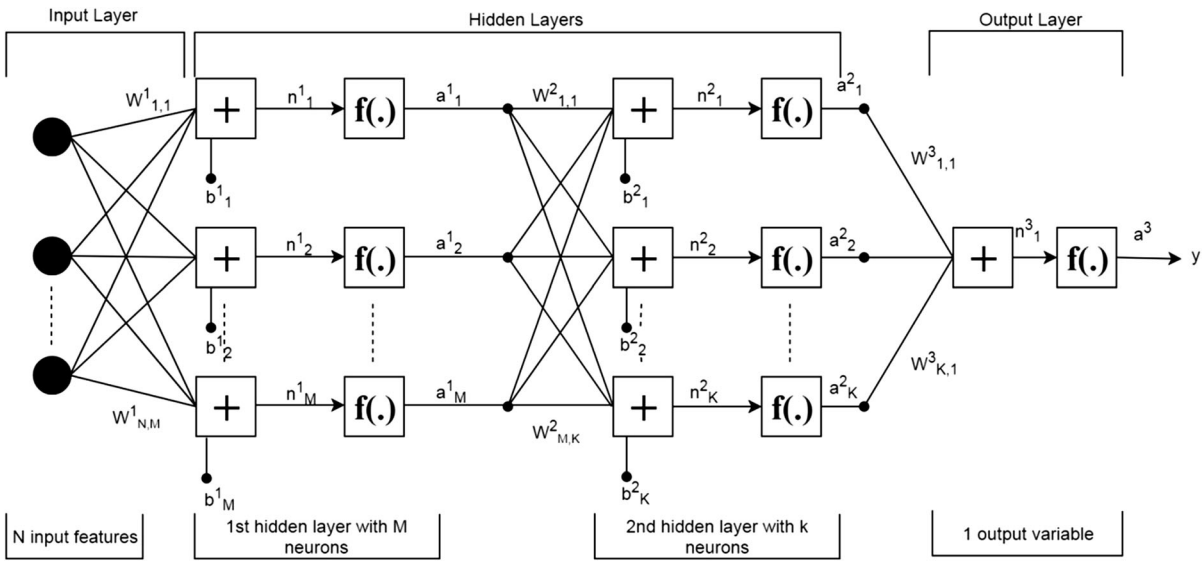

Fig. 2 Schematic plot of MLP

be different. Thus, a judgment is made by each DT in the forest separately and final results can be reached by voting of all trees (Ho 1995). Generally, as shown in Fig. 3, the RF algorithm consists of the following steps:

(i) Select samples from a dataset by a random selection (t = number of samples, see Fig. 3).
(ii) Build a decision tree for each sample and get a prediction result from each decision tree.
(iii) Perform a vote for the result of each prediction and select the most voted prediction result as the final prediction.

2.1.5 Support Vector Regression

Support vector machine (SVM) is a supervised ML method which is commonly used for three main purposes including classification (SVC), and pattern recognition and regression (SVR) problems. In the SVM, by assuming a dataset of N samples and supposing that represented type of result is y a set of training vectors (D) can be written as follow:

$$\mathrm{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}, x_i \in R^n, y_i \in \mathrm{R} \quad (1)$$

In the n-dimensional space, there are n influencing features to address the decision boundary. Soft-margin SVM applying penalty along with kernel trick can be efficiently used for nonlinear classification. Hence, any function $f(x)$ in SVM kernel trick can be shown as follows:

$$f(x) = w^T \phi(x) + b \quad (2)$$

where w^T represents the vector of output layer in a transposed form, b is a bias, x which is defined as an $N \times n$ matrix (N: number of data points, n: number of input variables) is the input variable and $\phi(x)$ is a kernel function. Vapnik introduced a minimization problem to find w and b:

$$\text{Minimize}: \ \frac{1}{2} w^T + c \sum_{i=1}^{N} \left(\xi_i - \xi_i^*\right) \quad (3)$$

$$\text{Subjected to}: \begin{cases} y_i - w^T \phi(x_i) - b \le \varepsilon + \xi_i, \ N = 1, 2, \ldots, N \\ w^T \phi(x_i) + b - y_i \le \varepsilon + \xi_i^*, \ N = 1, 2, \ldots, N \end{cases}$$

where c is a penalty coefficient, $\xi_i, \xi_i^* \ge 0$ are slack variables that refer to the effects of misclassification and ε refers to the precision of the function estimation. In this paper, we use the gaussian radial basis function as a kernel function:

$$K(x_k, x_l) = \exp\left[-\frac{(x_k - x_l)^T (x_k - x_l)}{2\sigma^2}\right] \quad (4)$$

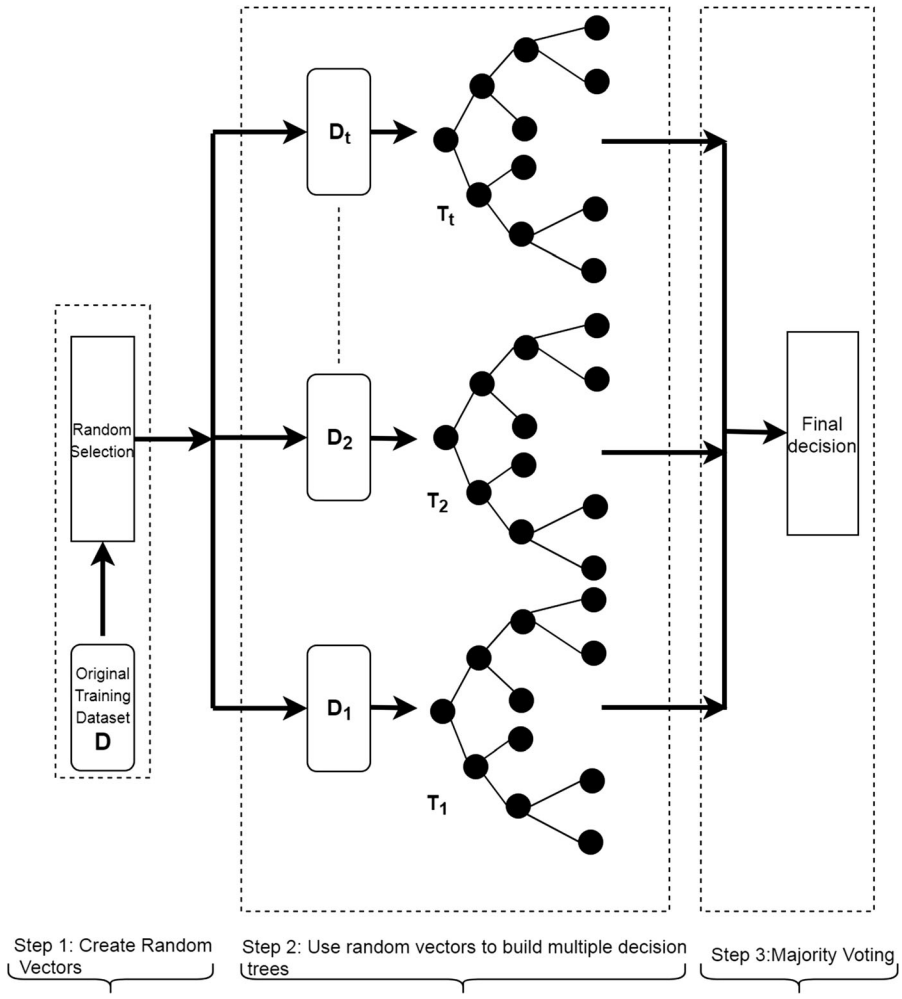

Fig. 3 Schematic diagram of a RF

where σ is the width of the radial basis function. The SVM is a powerful ML tool since it shows two considerable features: (1) fast calculation process to attain global optimum and (2) robust performance to overcome overfitting problem (Baghban et al. 2018; Wang 2005).

2.1.6 *Extreme Gradient Boosting*

Gradient boosting method (GBM) is an ensemble ML technique that uses weak learners to make a strong prediction model. The GBM uses a boosting technique in which predictors are made sequentially (i.e. in stage-wise fashion), but not independently (Chen et al. 2015). Like any supervised learning, the objective of GBM is to define a cost function and minimise it. Extreme gradient boosting (XGB) which belongs to the family of GBM algorithms was first introduced by Chen and Guestrin (2016). It is also an implementation of gradient boosted decision trees. However, the XGB uses a more regularised model formalisation to control over-fitting, which therefore gives better performance (Chen et al. 2018). Regarding XGB, the objective function (f) consists of cost function and a term of regularisation as below

$$f(\theta) = C(\theta) + \Omega(\theta) \tag{5}$$

where θ is the indicator of parameters, C represents the cost function, and Ω denotes the term of regularisation. Commonly, the cost function is chosen to be the mean squared error (MSE). The term of regularisation (Ω helps to avoid overfitting by controlling the complexity of the model using bias-variance trade-off to keep the model both simple and predictive. It penalises

building a complex tree with several leaf nodes to limit the hypothesis space of the base functions at any iteration. The model prefers a simple function over a complex function at any iteration for the same cost. Another difference between GBM and XGB is that the GBM uses the negative gradient to optimise the cost function while the XGB uses Taylor's expansion for this purpose.

2.2 Particle Swarm Optimisation Algorithm

Particle swarm optimisation (PSO) is part of the collective and swarm intelligence field which is a subfield of computational intelligence. The PSO is inspired by the flocking patterns of birds and schooling patterns of fish. It is proposed by Eberhart and Kennedy (1995) for the finding of global optimum solutions in a multidimensional space. In the PSO, all particles should be assigned with initial random position and initial random velocities. Then, each particle moves to find the best position in the hyper-volume based on its own best-known position (i.e. personal best position), its velocity and the best-known global position in the problem space. The global best position is the best position achieved by any particle while the personal best is the best position a particle obtains. Over time, particles not only try to achieve their personal best position but move in the global best direction. Meanwhile, the velocity of particles is revised based on the distance between the global best position and its personal best position (Baghban et al. 2018). Through a combination of exploration and exploitation, the particles converge together around the optima.

$$V_i^{t+1} = \omega V_i^t + c_1 r_1^t \left(P_i^t - X_i^t\right) + c_2 r_2^t \left(P_g^t - X_i^t\right) \quad (6)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (7)$$

where V is the velocity, r_1 and r_2 stand for random numbers in the range of $[0, 1]$, P denotes the best position while i and g stand for personal and global, respectively. Other parameters depend on the problem, c_1 and c_2 are called "trust" parameters, indicating how sure the particle toward its own position and swarm position is, and ω represents inertia weight. Equation (9) shows the decreasing of inertia weight along with the time:

$$\omega^t = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{t_{max}} t \quad (8)$$

where ω_{max} and ω_{min} stand for final and primary inertia weight values, respectively. Suitable values for ω_{min} and ω_{max} are 0.4 and 0.9, respectively (Eberhart et al. 1996; Kennedy 2010). t_{max} is in accordance with a maximum number of iterations. The PSO and Genetic algorithm (GA) are similar in many aspects. However, in the PSO algorithm, particles help each other instead of competing against each other (like what happens in the GA algorithm). The most important feature of the PSO algorithm which makes it different from other optimisation approaches is the high amount of particles in the PSO model which causes it to be significantly suitable to define global minima (Mantovani et al. 2015; Sarir et al. 2019).

2.3 PSO-MLs

In this study, six advanced ML algorithms were used for the estimation of the bearing capacity of the driven pile. All these algorithms include hyper-parameters that need tuning. Using experimenting with the hyper-parameters may obtain a better result; however, doing so manually can be a tedious task. In addition, the scope of parameters being searched may be limited by the computational ability of computers. To address this, the PSO algorithm can be utilised for exploring the optimum hyper-parameters of ML algorithms (Guo et al. 2008). Figure 4 shows the procedure for the implementation of PSO for hyper-parameter tuning. The learning error of the network is calculated through the root mean squared error (*RMSE*). The error of the network decreases to a minimum as net *RMSE* reaches zero.

3 Data Summary and Preparation

In this research, six OML algorithms are applied to the dataset of 59 cases of driven (or displacement) pile in cohesionless soil to estimate the bearing capacity through a comparative study. We use the multinational dataset collected by Mohanty et al. (2018) (see "Appendix 1"). The dataset used in the development of machine learning algorithm was drawn from 59 load test records for driven piles in sand and covered a wide spectrum of variation in soil relative density;

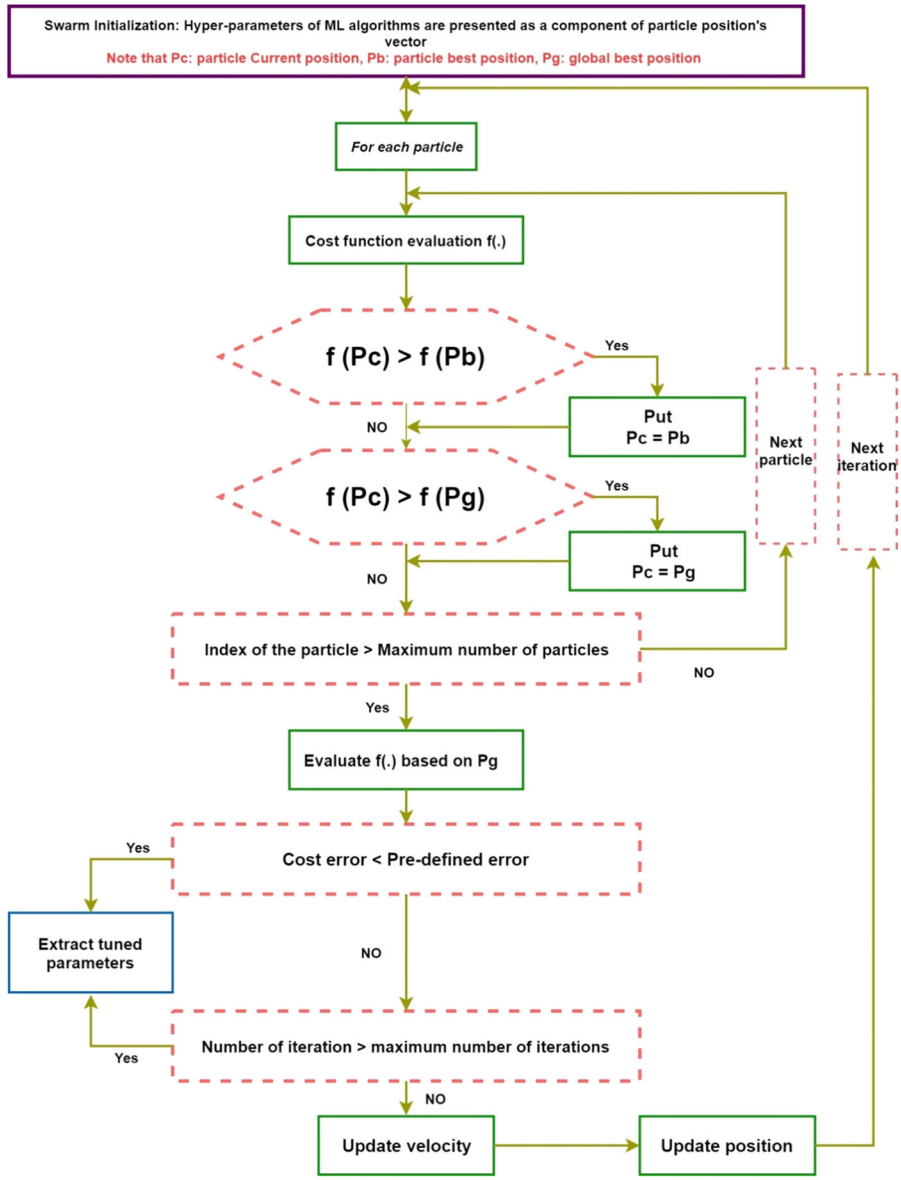

Fig. 4 Schematic diagram of PSO-MLs strategy

stress history; geographic locations; and pile types, lengths, and diameters. It should be noted that the available dataset is quite comprehensive in that it contains many parameters that are relevant to the determination of the pile capacity. It is important to select the most influencing variables to estimate the bearing capacity of piles. Five variables including soil shear resistance angle at the shaft of the pile (φ_s), soil shear resistance angle at the tip of the pile (φ_t), length of pile (L), cross-sectional area of pile (A) and effective stress at the tip of the pile (σ'_v) are selected as an input features (see Table 1).

Figures 5 and 6 illustrate the correlation matrix as a heat map and pair panel for the influencing variables, respectively. Correlation matrix indicates the correlation coefficient between variables while pair panel shows the distribution of each variable as well as scatter plots between a pair of variables.

3.1 Splitting Data

In this study, approximately 80% of the cases are included in the training set (47 samples) and 20% of samples (12 cases) are assigned to the testing set using

Table 1 Summary of the dataset

	φ_s (°)	φ_t (°)	L (m)	A (m^2)	σ'_v (KN/m^2)	Q_m (KN)
Count	59	59	59	59	59	59
Mean	34.87	36.406	20.65	0.1349	179.796	2145.97
SD	2.13	2.108	25.92	0.102	82.391	1248.14
Min	28	31	3	0.0061	38	75
Max	39	41	47.2	0.6568	475	5604

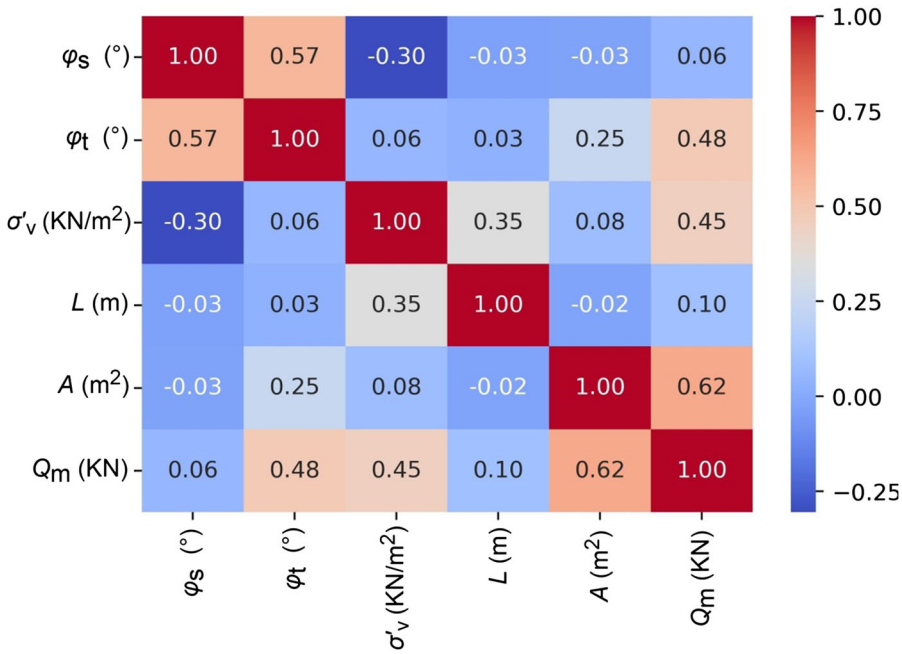

Fig. 5 Correlation matrix of five input variables and one output variable

a random selection, in order to retain the generalisation capability of the model as well as to overcome the overfitting problem. It should be noted that we normalized the dataset before carrying out any modelling. The goal of normalization is to change the values of the dataset to a common scale, without distorting differences in the ranges of values.

3.2 Cross-Validation

In this study, K-fold cross-validation (K-fold CV) is used to evaluate the prediction ability of six OML algorithms on the same dataset. Cross-validation methods can be applied to the dataset in order to overcome the risk of overfitting and selection bias of the ML methods (Rodriguez et al. 2010). In the K-fold CV, the dataset is divided into K equal size subsets. In the K subsets, K − 1 subsets are used as a training data set, whereas the single remaining subset is utilised as a testing dataset. This process is then repeated K times with different subsets being used as a testing subset. In other words, CV is a resampling method used to evaluate OML models on a limited data sample. In this study, 5-fold CV which is one of the most popular CV is applied to the dataset.

3.3 Performance Metrics

In this work, R-squared (R^2), root mean squared error (*RMSE*) and variance accounted for (*VAF*) are examined between actual values and estimated values of bearing capacity of piles to reveal the estimation ability of six OML algorithms. R-squared is a statistical metric that can be used to evaluate the goodness of the fit. In other words, R-squared expresses how well a model approximates the real data points (Cameron and Windmeijer 1997).

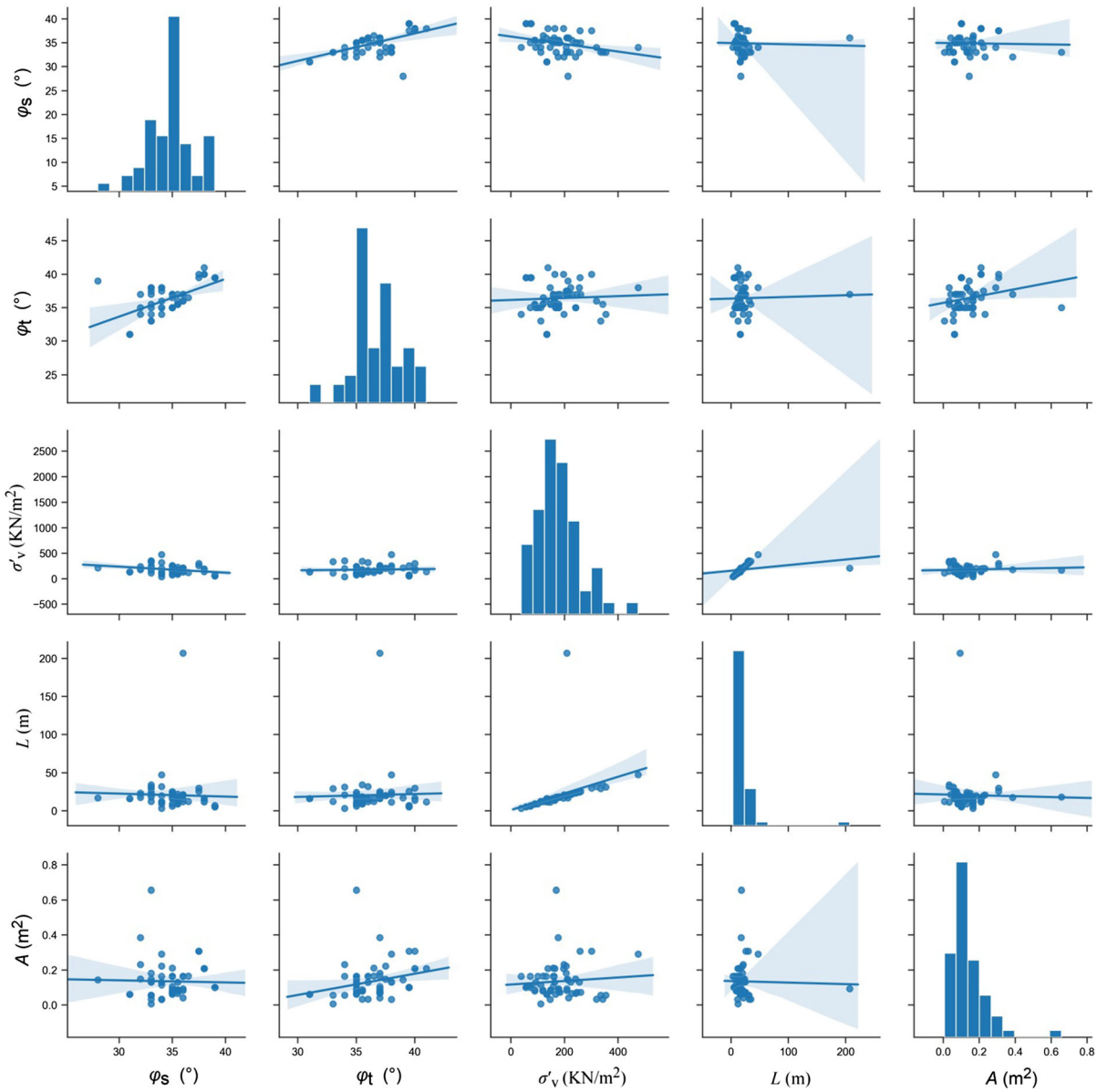

Fig. 6 Pair panel of input variables

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(X_i^{actual} - X_i^{predicted}\right)^2}{\sum_{i=1}^{N}\left(X_i^{actual} - \overline{X^{predicted}}\right)^2} \quad (9)$$

where N is the number of data points, X_i^{actual} and $X_i^{predicted}$ are actual and predicted values for ith X and $\overline{X^{actual}}$ is the average of actual values.

Mean squared error is the most popular metric to evaluate models and is defined as below:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(X_i^{actual} - X_i^{predicted}\right)^2 \quad (10)$$

In other words, the squared difference between predicted outcomes and actual values is determined and the average of those values will be calculated for each data point. In a case that the dataset consists of unexpected values such as very high or low value, the *MSE* can be a useful metric to apply. However, when it comes to noisy data, i.e., the data are not entirely

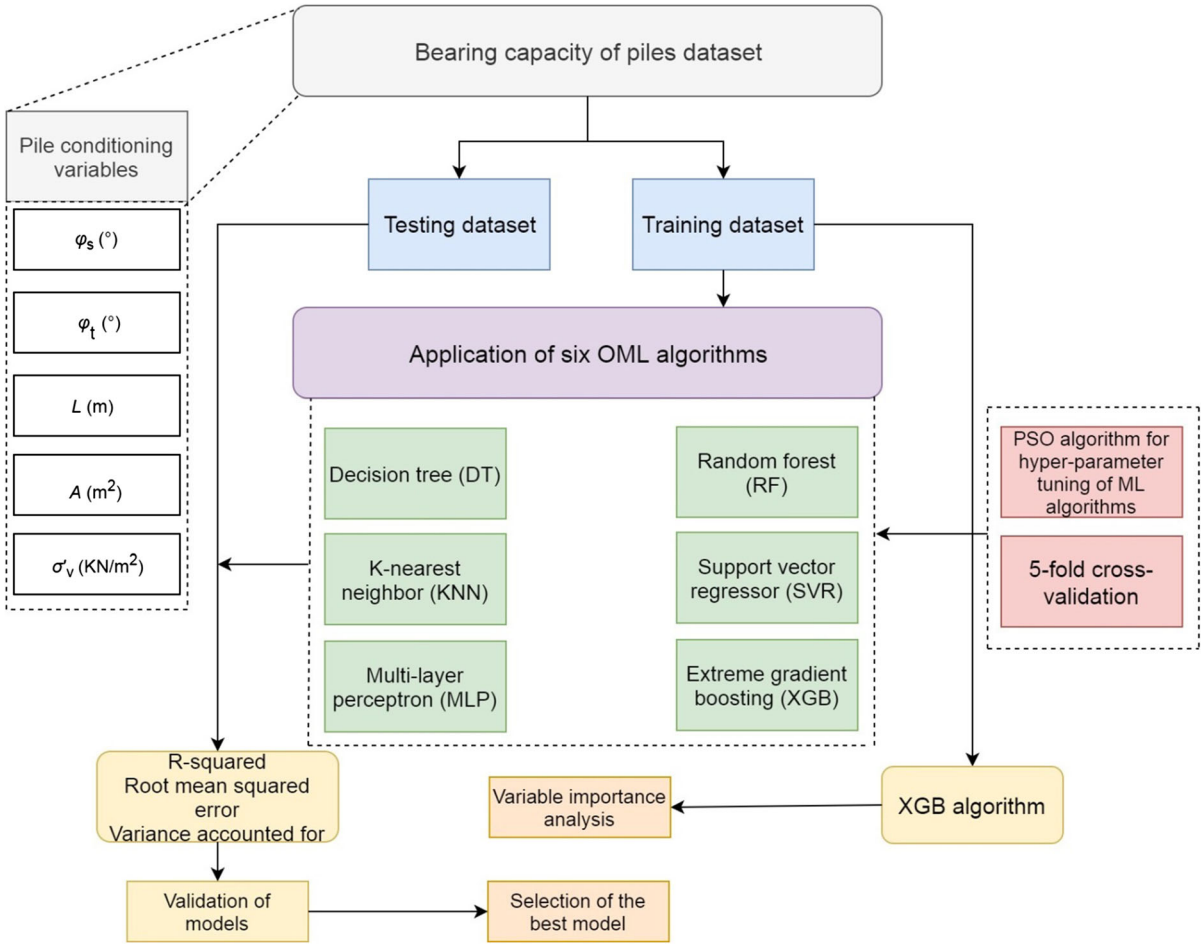

Fig. 7 Methodological flowchart of this study

reliable, the *MSE* can either overestimate or underestimate the badness of the prediction. The *RMSE* is defined as a root of *MSE* (Chai and Draxler 2014). In this study normalised *RMSE* is used to evaluate the performance of OML algorithms, which is defined by

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_i^{actual} - X_i^{predicted}\right)^2} \quad (11)$$

Variance accounted for is another metric for regression problems and calculate the variance between the actual values and predicted values as follow:

Table 2 Hyper-parameter tuning

ML algorithms	Optimum value
DT	min_samples_split = 4, mins_samples_leaf = 4, max-depth = 5,
KNN	n_neighbours = 7, function = distance
MLP	Hidden layer size = (10, 12), alpha = 0.001
RF	min_samples_split = 5, mins_samples_leaf = 3, max-depth = 6
SVR	kernel = rbf, C = 1
XGB	max_depth = 5, gamma = 0.25

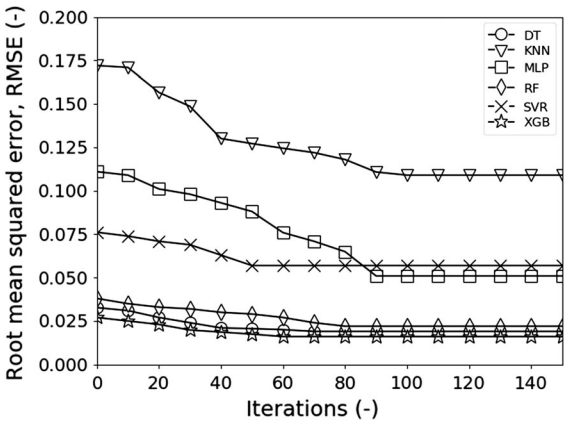

Fig. 8 RMSE values along with iterations on the training dataset

$$VAF = \left[1 - \frac{var\left(X_i^{actual} - X_i^{predicted}\right)}{var\left(X_i^{actual}\right)}\right] \times 100 \quad (12)$$

4 Results and Comparisons

The tunning of hyper-parameters, comparison of six ML algorithms on the bearing capacity estimation and the importance of influencing variables are presented in this section. The procedure for bearing capacity estimation using OML techniques is shown in Fig. 7. The initial dataset is partitioned into training and testing dataset. Six advanced ML algorithms which are trained to the training dataset are optimised by using PSO. Then, to compare the performances of OML algorithms, they are applied to the testing dataset.

4.1 Results of Hyper-Parameter Tuning

As shown in Fig. 7, the PSO is used in this study for hyper-parameter tuning of six ML algorithms. Table 2 presents the hyper-parameters of each ML algorithms and the tuned values of hyper-parameters. Figure 8 shows the change of root mean squared error value along with iterations on the training dataset. Figure 8 also represents that the hyper-parameter tuning has a remarkable influence on the performance of the ML algorithms, particularly for MLP and KNN.

4.2 Evaluation and Comparison of Six ML Algorithms

4.2.1 Results of the Training Dataset

Six OML algorithms are applied to the training dataset of 47 cases of driven piles. Figure 9 shows the regression plots of each of these algorithms. PSO-XGB shows the best performance among the six OMLs (R^2 = 0.9807) which is an outstanding performance. This is followed by PSO-DT, PSO-RF and PSO-MLP with R-squared value of 0.9428, 0.9235 and 0.8408, respectively. However, SVR (R^2 = 0.8222) and KNN (R^2 = 0.7706) only demonstrate acceptable performance.

4.2.2 Result of the Testing Dataset

Six OML algorithms have been trained in the previous section and now in this section they are used to estimate the bearing capacity of cases in the testing dataset. Twelve examples in the testing dataset, on which no training process was utilised, are used in order to evaluate the performances of six OML techniques. Figure 10 shows the regression plots for the testing set of six OML algorithms. By comparing Fig. 9 with Fig. 10, we can see that the ranking of the performance of OML techniques for the testing dataset is not similar to the one for the training dataset and also R-squared values vary in the training and testing datasets. For example, PSO-XGB shows the best performance on the training dataset (R^2 = 0.9615). PSO-RF and PSO-DT also show outstanding performance ($R^2 > 90\%$). This is followed by PSO-SVR (R^2 = 0.8182), PSO-MLP (R^2 = 0.7991) and PSO-KNN (R^2 = 0.731).

4.2.3 Comparison of Results

Table 3 indicates R-squared values, root mean squared error values and variance accounted for values of each OML algorithm for training and testing datasets. In addition, the performance ranking of each technique is shown. As shown in Table 3, PSO-XGB and PSO-KNN obtain respectively the highest and lowest R-squared values, and are ranked as first and last OML algorithms respectively, for both training and testing datasets. For training dataset, PSO-DT and

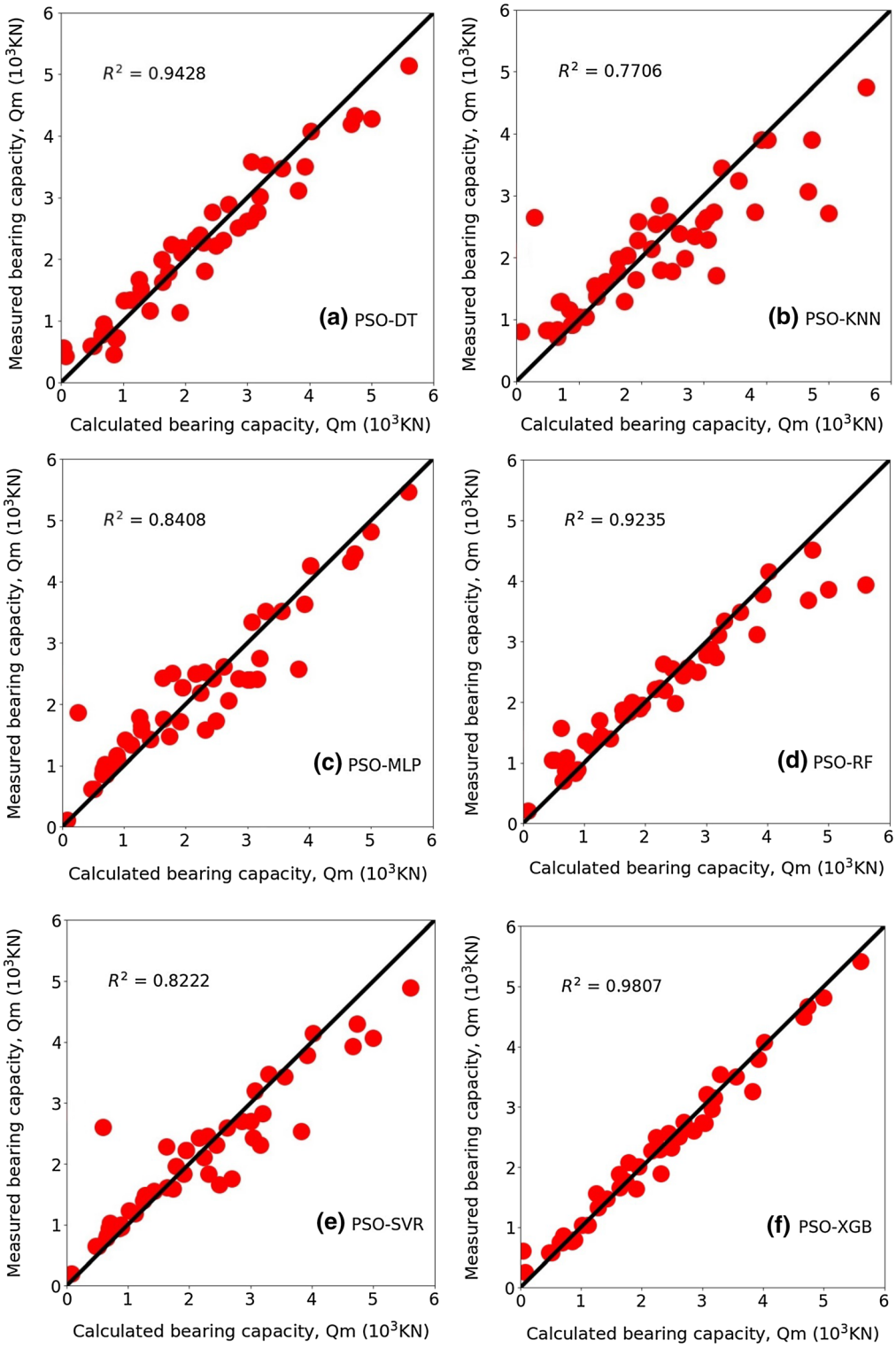

Fig. 9 Regression plots for the training set of six OML algorithms

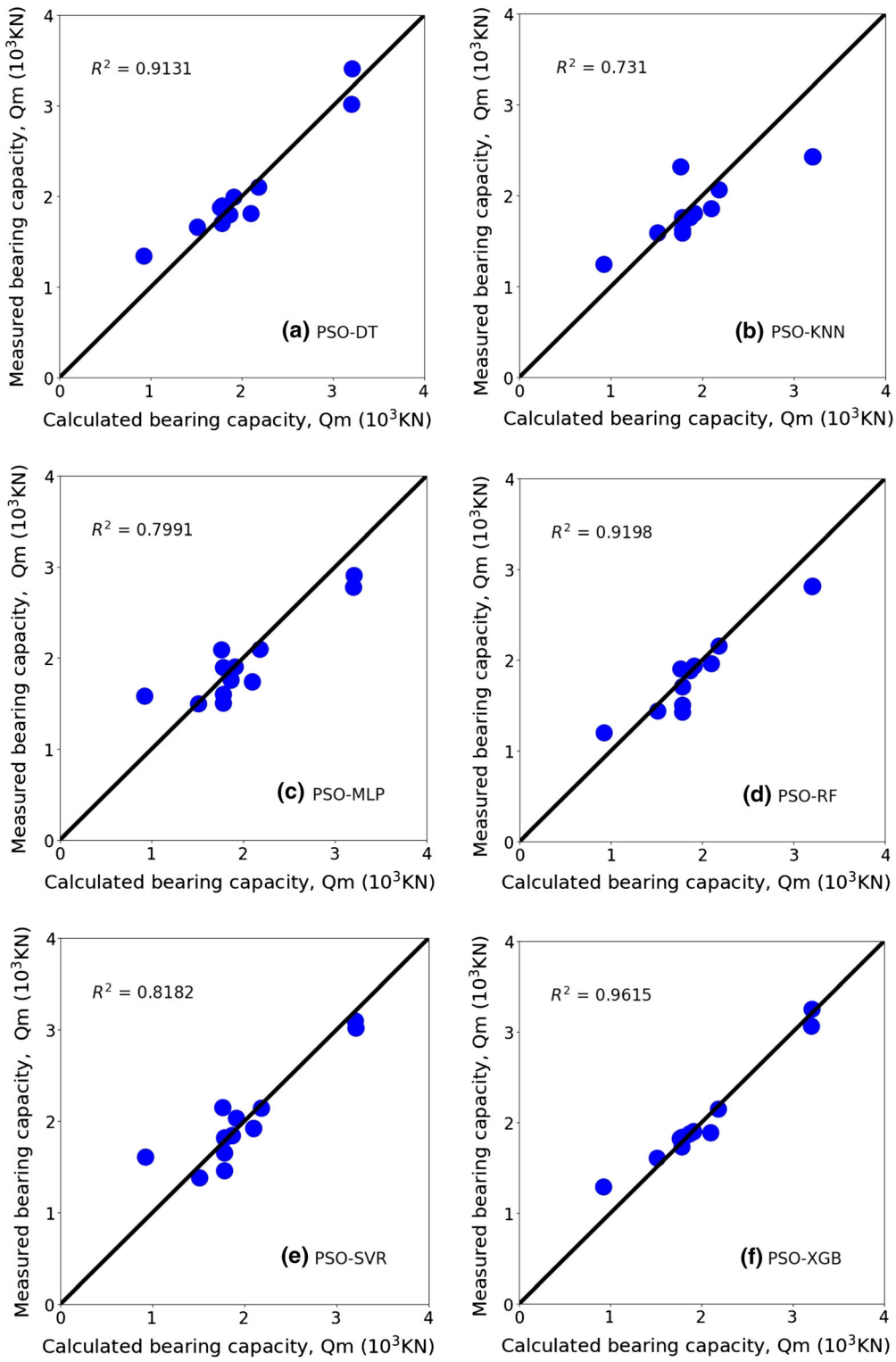

Fig. 10 Regression plots for the testing set of six OML algorithms

Table 3 Statistical analyses of six ML algorithms

Algorithms	R^2	*RMSE*	*VAF*	Ranking
PSO-DT				
Training	0.9428	0.019	94.37	2
Testing	0.9131	0.027	92.52	3
PSO-KNN				
Training	0.7706	0.109	74.08	6
Testing	0.731	0.148	62.71	6
PSO-MLP				
Training	0.8408	0.051	83.77	4
Testing	0.7991	0.080	78.84	5
PSO-RF				
Training	0.9235	0.022	87.32	3
Testing	0.9198	0.026	87.28	2
PSO-SVR				
Training	0.8222	0.057	81.3	5
Testing	0.8182	0.069	78.85	4
PSO-XGB				
Training	0.9807	0.011	97.78	1
Testing	0.9615	0.016	95.21	1

PSO-RF show the second best and third best performances among all the OML algorithms, respectively, according to their R-squared values. However, for the testing dataset, PSO-RF shows better estimation performance than PSO-DT. The same trend can be seen for PSO- MLP and PSO-SVR, where PSO-MLP shows a better performance for training dataset while PSO-SVR shows a better performance for the testing dataset. Since the testing dataset can be considered as an example of the real-world problem, the results of the performance of OML techniques for the testing dataset is more important to be considered as an application of each OML algorithms.

4.3 Results of Variable Importance

Based on the comparison, the PSO-XGB shows the best performance in both training and testing datasets. Therefore, PSO-XGB is used to analyse the importance of influencing variables for the bearing capacity estimation. Figure 11 shows the normalised scores of variable importance. As shown in Fig. 11, effective stress is identified as the most influencing feature (score = 0.309) for bearing capacity of piles in this paper. Importance values of the remaining variables on bearing capacity of pile decrease according to the following order: cross-sectional area of the pile (score = 0.216) > soil friction angle at shaft (0.2) > length of the pile (0.155) > soil friction angle at pile tip (0.12). It is important to note that, scores may be different when different dataset and models are utilised (Guyon et al. 2008). Also, it is worth to mention that the entire five input features have non-ignorable importance values as all these features constitute the basic input parameters in most engineering projects.

4.4 Comparison Between PSO-XGB and a Semi-Empirical Method

In this section, a semi-empirical method based on effective stress (β-method) is used to analyse the bearing capacity of piles and result are then compared to the PSO-XGB algorithm. Statistic capacity calculations in cohesionless soils can be performed using an effective stress-based method (Fellenius 1991).

In this method bearing capacity of a pile (Q_m) may be taken as the sum of the shaft resistance (Q_s) and base resistance (Q_p), viz,

$$Q_m = Q_s + Q_p \tag{13}$$

In an effective stress analysis, the unit shaft resistance is calculated from the following expression:

$$Q_s = \int_0^L C\left(\beta\sigma'_{\mathrm{v}} + c_a\right)dz \tag{14}$$

where $\beta = K'\tan(\delta)$, σ'_{v} is the effective vertical stress and C is the pile circumference. Usually c_a is set to zero for driven piles but may be non-zero for cast-in-place piles (for a complete explanation see "Appendix 2").

The unit base resistance is calculated from:

$$Q_p = A_p N_t \sigma'_{\mathrm{v}} \tag{15}$$

where N_t = base bearing capacity coefficient and A_p = cross-sectional area (Mayerhof 1976; Poulos and Davis 1980) (for a complete explanation see "Appendix 2").

As it can be seen from Fig. 12, PSO-XGB shows the prominent performance in comparison with the empirical method. Table 4 shows the statistical values

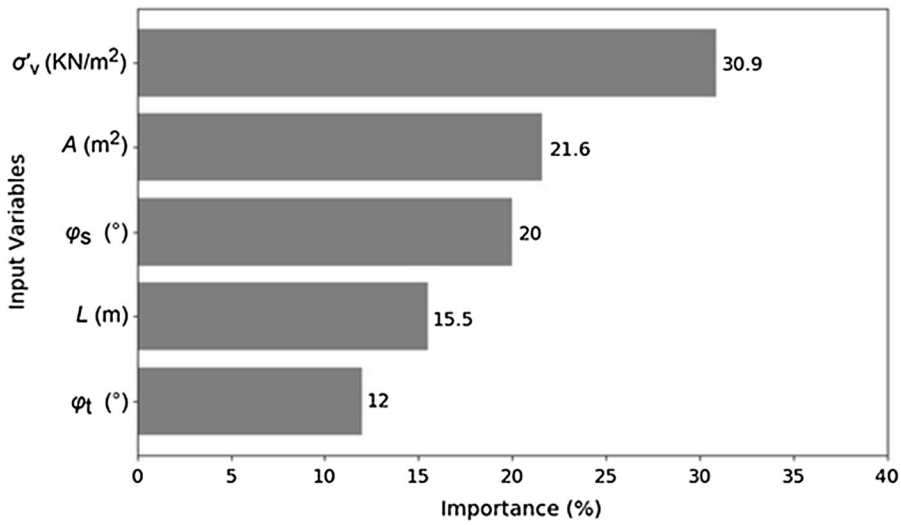

Fig. 11 Relative variable importance for bearing capacity of piles

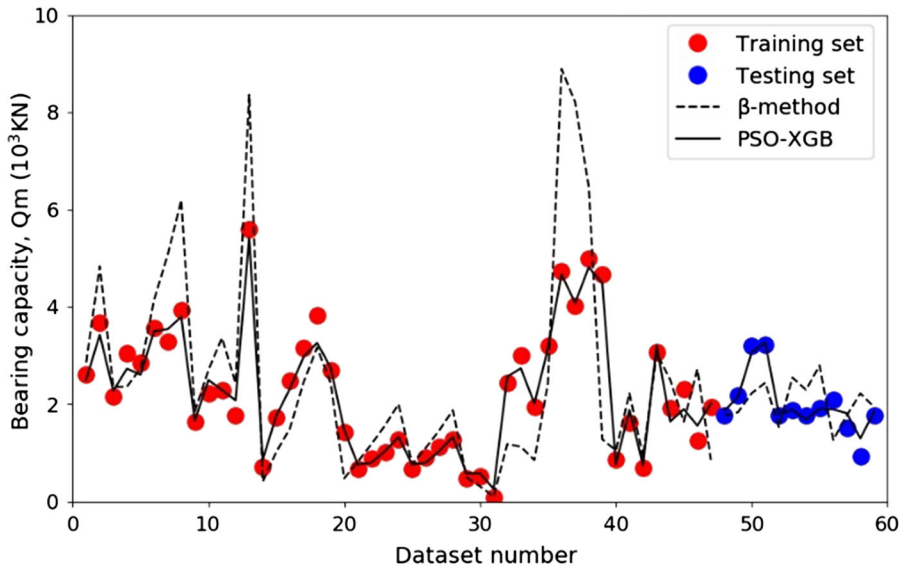

Fig. 12 Measured and calculated bearing capacity at training and testing stages

(R-squared and root mean squared error) of β-method and PSO-XGB algorithm.

4.5 Discussion

The primary strength of this study is to propose and compare six optimised ML methods on the bearing capacity estimation of piles. This study contributes to the bearing capacity of plie analysis, as well as other fields of geotechnical engineering, in the following aspects: (*i*) the optimised ML methods and PSO are very promising for regression problems in

Table 4 Statistical analyses of empirical and ML methods

Method	R^2	*RMSE*	*VAF*
β-method			
Training	0.6221	0.191	58.43
Testing	0.3452	0.274	31.67
PSO-XGB			
Training	0.9807	0.011	97.78
Testing	0.9615	0.016	95.21

geotechnical engineering, (*ii*) the stableness and robustness of regression models can be well investigated using R^2, *RMSE* and *VAF*, (*iii*) some recommendations have been made for estimating bearing capacity of piles by using ML methods, and (*iv*) the methodology in this paper has great potential for a wider application in other geotechnical and geological engineering aspects where regression problems are widely encountered. However, some influencing variables for bearing capacity of piles, such as geological structure, and engineering disturbance, are not considered, which can be regarded a limitation of the present study. Another limitation of this research is that the pile dataset is still relatively small. The performance of the proposed optimised ML methods can be improved if more data is available.

5 Conclusion

In this study, a comprehensive comparison of six OML algorithms including DT, KNN, MLP, RF, SVR and XGB, has been presented to indicate the best model for estimating the bearing capacity of driven piles in cohesionless soil. PSO algorithm is implemented for tuning the hyper-parameters of ML algorithms. A dataset comprising of 59 cases is employed for the OML algorithms. R-squared, root mean square error and variance accounted for are used as the performance metrics and fivefold cross-validation is used as the validation method. Some important conclusions can be drawn as follows:

(1) According to the optimum scores obtained by ML algorithms within iterations, PSO is an effective algorithm for tuning the hyper-parameters of ML algorithms.
(2) According to the R-squared and root mean square error values of OML algorithms on the training dataset, PSO-XGB shows the best performance (R^2 = 0.9807). PSO-DT and PSO- RF, also show outstanding performances with a R-squared value of 0.9428 and 0.9235 respectively. However, PSO-KNN achieved R-squared of 0.7706 which is the lowest value amongst six OML algorithms. According to the results for the testing dataset, PSO-XGB achieved the best score (R^2 = 0.9615). This followed by PSO-RF and PSO-DT, respectively. All these three models (PSO-XGB, PSO-RF, PSO-DT) could be considered as an outstanding model for estimating the bearing capacity of piles while PSO-KNN is not suitable for this purpose.
(3) Results of variable importance analysis indicate that effective stress is the most influential variable (score = 0.309) for bearing capacity estimation of piles.
(4) Comparison between PSO-XGB algorithm (R^2 = 0.9615) and β-method (R^2 = 0.3452) shows the prominent performance of OML approaches in comparison to empirical methods.

Appendix 1: Dataset of the Multinational Pile Cases

See Table 5.

Table 5 Dataset of the multinational pile cases

φ_s (°)	φ_t (°)	L (m)	A (m^2)	σ'_v (KN/m^2)	Q_m (KN)
Low-sill structure, Old River, USA					
33	38	255	24.5	0.131	2615
34	37.5	206	19.8	0.223	3675
33	38	223	21.5	0.131	2164
33	37.5	210	20.2	0.1468	3042
33	37	206	19.9	0.1821	2856
Lanesville, USA					
38	41	138	11.6	0.209	3558
38	40	164	13.7	0.209	3292
38	40	196	16.5	0.209	3923
Arkansas river project LD4 and LD5, USA					
35	37	158	16.2	0.105	1637
35	37	158	16.1	0.1644	2233
35	36.5	158	16.2	0.2109	2295
36.5	36.5	120	12.3	0.1654	1779
Low arrow lake, BC, Canada					
34	38	475	47.2	0.2917	5604
Ogeeches river, USA					
34	34	38	3	0.1644	712
35	35	72	6.1	0.1644	1735
35	35	100	8.9	0.1644	2491
36	36	131	12	0.1644	3158
36	36	161	15	0.1644	3825

Table 5 continued

φ_s (°)	φ_t (°)	L (m)	A (m^2)	σ'_v (KN/m^2)	Q_m (KN)
35.5	36	163	15.2	0.1301	2695
Tokyo, Japan					
34	38	146	11.3	0.0316	1429
St. Charles river, QC, Canada					
35.5	35.5	89	9.1	0.0864	658
35.5	35.5	119	12.2	0.0864	882
35.5	35.5	148	15.2	0.0864	1014
35.5	35.5	178	18.3	0.0864	1281
35.5	35.5	89	9.1	0.0799	655
35.5	35.5	119	12.2	0.0799	894
35.5	35.5	148	15.2	0.0799	1113
35.5	35.5	178	18.3	0.0799	1281
Holmen island, Drammen, Norway					
31	31	134	16	0.0613	480
31	31	134	16	0.0613	519
Albysjon, Sweden					
33	33	111	12.2	0.0061	75
North sea, The Netherlands					
39	39.5	75	7	0.0999	2439
39	39.5	72	6.7	0.0999	3000
39	39.5	56	5.2	0.0999	1950
Zeebrugge, Belgium					
32	34	198	21	0.2313	3200
West Seattle freeway, USA					
37.5	40	301	29.9	0.3075	4733
37.5	39.5	258	25.6		0.3075
Gadiz, Spain					
33	35	169	18	0.6568	5000
Tacoma, USA					
28	39	213	16.8	0.1431	4670
Glasgow, Scotland					
34	35	111	12.2	0.1143	854
32	35	241	23.3	0.1486	1628
34	35	92	9.1	0.1291	685
Biograd, Yugoslavia					
32	37	176	17.5	0.3855	3069
Missouri, USA					
35	37	246	23.8	0.0729	1913
35	37	183	17.7	0.0729	2313
Connecticut, USA					
35	37	260	25.3	0.0729	1254
North sea, The Netherlands					
39	39.5	56	5.2	0.0999	1948
Cohasset, USA					
33	35.5	343	34.1	0.0325	1761
34	36	319	31.7	0.0325	2180
Kaohsiung, China					
33	33	335	29.3	0.0557	3203
33	34	354	31.1	0.0557	3211
Winnemucca, USA					
36	37	215	21.3	0.0409	1779
36	37	209	20.7	0.0827	1868
35	35	242	24.1	0.066	1779
36	37	209	207	0.0929	1913
35	35	166	16.5	0.0613	2100
35	35	178	17.7	0.0827	1509
35.5	37	228	21.9	0.066	922
35	35	178	17.7	0.0929	1779

Appendix 2: Calculating the Unit Shaft Resistance

In an effective stress analysis, the unit shaft resistance is calculated from the following expression:

$$Q_s = \int_0^L C(\beta\sigma'_v + c_a)dz \tag{16}$$

where $\beta = K'\tan(\delta)$, σ'_v is the effective vertical stress and C is the pile circumference. Usually c_a is set to zero for driven piles but may be non-zero for cast-in-place piles. Fellenius (1991) tells us that typical values for β depend on soil gradation, mineral composition, density and soil strength within a fairly narrow range. Some empirical ranges for β coefficients in different soil types are,

Meyerhof (1976) has proposed values of $K'\tan(\delta)$ for driven piles. Note that the shaft resistance values reflect the likely changes of stress state in the soil due to the method of installation. In using Table 6, the undisturbed value is used in all cases.

The unit base resistance is calculated from:

$$Q_p = A_p N_t \sigma'_v \tag{17}$$

where N_t = base bearing capacity coefficient, A_p = cross-sectional area and σ'_v = effective stress.

Recommended ranges of β and N_t coefficients as a function of soil type and φ_s angle from Fellenius (1991) are presented in Table 6. Fellenius notes that factors affecting the β and N_t, coefficients consist of

Table 6 Approximate range of β and N_t coefficients (Fellenius 1991)

Soil type	β	N_t	φ_s
Clay	0.23–0.40	3–30	25–30
Silt	0.27–0.50	20–40	28–34
Sand	0.30–0.60	30–150	32–40
Gravel	0.35–0.80	60–300	35–45

the soil composition including the grain size distribution angularity and mineralogical origin of the soil grains, the original soil density and density due to the pile installation technique, the soil strength, as well as other factors. Even so, β coefficients are generally within the ranges provided and seldom exceed 1.0.

For sedimentary cohesionless deposits, Fellenius states N_t, ranges from about 30 to a high of 120. In very dense non-sedimentary deposits such as glacial tills,N_t can be much higher, but can also approach the lower bound value of 30. In clays, Fellenius notes that the toe resistance calculated using a N_t of 3 that is similar to the toe resistance calculated from a traditional analysis using undrained shear strength. Therefore, the use of a relatively low N_t, coefficient in clays is recommended unless local correlations suggest higher values are appropriate.

Graphs of the ranges in β and N_t coefficients versus the range in φ_s angle as suggested by Fellenius. These graphs may be helpful in the selection of β or N_t. The inexperienced user should select conservative β and N_t coefficients. As with any design method, the user should also confirm the appropriateness of a selected β or N_t, coefficient in a given soil condition with local correlations between static capacity calculations and static load tests results.

It should be noted that the effective stress method places no limiting values on either the shaft or base resistance.

Step 1 Delineate the soil profile into layers and determine φ_s angle for each layer.

a. Construct σ'_v diagram along the depth of the pile.
b. Divide soil profile throughout the pile penetration depth into layers and determine the effective stress,σ'_v, at the midpoint of each layer.
c. Determine the φ_s angle for each soil layer from laboratory or in-situ test data.
d. In the absence of laboratory or in-situ data for cohesionless layers, determine the average corrected N_{cor} value for each layer and estimate φ_s angle from Schmertmanns (1975) SPT correlation for soils.

Step 2 Select the β coefficient for each soil layer.

a. Utilise local experience to select β coefficient for each layer.
b. In the absence of local experience, use Meyerhof's Pile Factors to estimate β coefficient from φ_s angle for each layer. NB $\beta = K'\tan(\delta)$

Step 3 For each soil layer calculate the shaft resistance Q_s (kPa) from Eq. (1). The total shaft resistance is simply the sum of the shaft resistance from each soil layer.

Step 4 Calculate the ultimate toe resistance, Q_m (kPa).

a. Utilise local experience to select N_t coefficient.
b. In the absence of local experience, estimate N_t, from Table 6 based on φ_s angle.
c. Calculate the effective stress at the pile base, σ'_v.

Step 5 Calculate the ultimate pile capacity, $Q_m = Q_s(KN) + Q_p$.

References

Abu-Farsakh MY, Titi HH (2004) Assessment of direct cone penetration test methods for predicting the ultimate capacity of friction driven piles. J Geotech Geoenviron Eng 130(9):935–944

Alkroosh I, Nikraz H (2011) Correlation of pile axial capacity and CPT data using gene expression programming. Geotech Geol Eng 29(5):725–748

Alkroosh I, Nikraz H (2014) Predicting pile dynamic capacity via application of an evolutionary algorithm. Soils Found 54(2):233–242

Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185

Armaghani DJ, Raja RSNSB, Faizi K, Rashid ASA (2017) Developing a hybrid PSO–ANN model for estimating the ultimate bearing capacity of rock-socketed piles. Neural Comput Appl 28(2):391–405

Baghban A, Kardani MN, Mohammadi AH (2018) Improved estimation of Cetane number of fatty acid methyl esters (FAMEs) based biodiesels using TLBO-NN and PSO-NN models. Fuel 232:620–631

Cameron AC, Windmeijer FA (1997) An R-squared measure of goodness of fit for some common nonlinear regression models. J Econom 77(2):329–342

Cao M, Zhou A (2019) Analytical solutions to interaction factors of two unequal length piles under horizontal loads. Int J Geomech 19(4):06019003

Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE) arguments against avoiding RMSE in the literature. Geosci Model Dev 7(3):1247–1250

Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM. pp 785–794

Chen T, He T, Benesty M, Khotilovich V, Tang Y (2015) Xgboost: extreme gradient boosting. R package version 0.4-2: 1–4

Chen X, Huang L, Xie D, Zhao Q (2018) EGBMMDA: extreme gradient boosting machine for MiRNA-disease association prediction. Cell Death Dis 9(1):3

Chow Y, Karunaratne G, Wong K, Lee S (1988) Prediction of load-carrying capacity of driven piles. Can Geotech J 25(1):13–23

Coyle HM, Castello RR (1981) New design correlations for piles in sand. J Geotech Geoenviron Eng 107:965–986

Das SK, Basudhar PK (2006) Undrained lateral load capacity of piles in clay using artificial neural network. Comput Geotech 33(8):454–459

Das SK, Sivakugan N (2010) Discussion of "Intelligent computing for modeling axial capacity of pile foundations". Can Geotech J 47(8):928–930

Eberhart R, Kennedy J (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks. 1995. Citeseer. pp 1942–1948

Eberhart R, Simpson P, Dobbins R (1996) Computational intelligence PC tools. Academic Press Professional Inc, New York

Fellenius BH (1991) Pile foundations. In: Foundation engineering handbook. Springer, Boston, pp 511–536

Gao Y, Sun DA, Zhou AN (2015) Hydromechanical behaviour of unsaturated soil with different specimen preparations. Can Geotech J 53(6):909–917

Garcia S, Derrac J, Cano JR, Herrera F (2011) Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans Pattern Anal Mach Intell 3:417–435

Guo X, Yang J, Wu C, Wang C, Liang Y (2008) A novel LS-SVMs hyper-parameter selection based on particle swarm optimization. Neurocomputing 71(16–18):3211–3215

Guyon I, Gunn S, Nikravesh M, Zadeh LA (2008) Feature extraction: foundations and applications. Springer, Berlin

Ho TK (1995) Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition, IEEE. pp 278–282

Jin X, Wang T, Cheng W-C, Luo Y, Zhou A (2018) A simple method for settlement evaluation of loess-pile foundation. Can Geotech J. https://doi.org/10.1139/cgj-2017-0690

Kennedy J (2010) Particle swarm optimization. Encycl Mach Learn. Springer, pp 760–766. https://doi.org/10.1007/978-0-387-30164-8_630

Kiefa MA (1998) General regression neural networks for driven piles in cohesionless soils. J Geotech Geoenviron Eng 124(12):1177–1185

Ma C, Lu DC, Du XL, Zhou AN (2017) Developing a 3D elastoplastic constitutive model for soils: a new approach based on characteristic stress. Comput Geotech 86:129–140

Mantovani RG, Rossi AL, Vanschoren J, Bischl B, De Carvalho AC (2015) Effectiveness of random search in SVM hyper-parameter tuning. In: 2015 international joint conference on neural networks (IJCNN), IEEE. pp 1–8

Mayerhof G (1976) Bearing capacity and settlemtn of pile foundations. J Geotech Geoenviron Eng 102:195–228

Moayedi H, Armaghani DJ (2018) Optimizing an ANN model with ICA for estimating bearing capacity of driven pile in cohesionless soil. Eng Comput 34(2):347–356

Moayedi H, Hayati S (2018) Modelling and optimization of ultimate bearing capacity of strip footing near a slope by soft computing methods. Appl Soft Comput 66:208–219

Moayedi H, Moatamediyan A, Nguyen H, Bui X-N, Bui DT, Rashid ASA (2019) Prediction of ultimate bearing capacity through various novel evolutionary and neural network models. Eng Comput. https://doi.org/10.1007/s00366-019-00723-2

Mohanty R, Suman S, Das SK (2018) Prediction of vertical pile capacity of driven pile in cohesionless soil using artificial intelligence techniques. Int J Geotech Eng 12(2):209–216

Momeni E, Nazir R, Armaghani DJ, Maizir H (2014) Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN. Measurement 57:122–131

Muduli PK, Das SK, Das MR (2013) Prediction of lateral load capacity of piles using extreme learning machine. Int J Geotech Eng 7(4):388–394

Pal M, Deswal S (2010) Modelling pile capacity using Gaussian process regression. Comput Geotech 37(7–8):942–947

Poulos HG, Davis EH (1980) Pile foundation analysis and design. Wiely, New york

Randolph MF (2003) Science and empiricism in pile foundation design. Géotechnique 53(10):847–876

Randolph MF, Murphy B (1985) Shaft capacity of driven piles in clay. In: Offshore technology conference

Rodriguez JD, Perez A, Lozano JA (2010) Sensitivity analysis of k-fold cross validation in prediction error estimation. IEEE Trans Pattern Anal Mach Intell 32(3):569–575

Rokach L, Maimon OZ (2008) Data mining with decision trees: theory and applications. World scientific, Singapore

Samui P (2011) Prediction of pile bearing capacity using support vector machine. Int J Geotech Eng 5(1):95–102

Samui P (2012a) Application of relevance vector machine for prediction of ultimate capacity of driven piles in cohesionless soils. Geotech Geol Eng 30(5):1261–1270

Samui P (2012b) Determination of ultimate capacity of driven piles in cohesionless soil: a multivariate adaptive regression spline approach. Int J Numer Anal Methods Geomech 36(11):1434–1439

Sarir P, Shen SL, Wang ZF, Chen J, Horpibulsuk S, Pham BT (2019) Optimum model for bearing capacity of concrete-steel columns with AI technology via incorporating the algorithms of IWO and ABC. Eng Comput. https://doi.org/10.1007/s00366-019-00855-5

Suman S, Das SK, Mohanty R (2016) Prediction of friction capacity of driven piles in clay using artificial intelligence techniques. Int J Geotech Eng 10(5):469–475

Svinkin MR (2002) Engineering judgement in determination of pile capacity by dynamic methods. In: Deep foundations 2002: an international perspective on theory, design, construction, and performance, pp 898–914

Teh C, Wong K, Goh A, Jaritngam S (1997) Prediction of pile capacity using neural networks. J Comput Civil Eng 11(2):129–138

Wang L (2005) Support vector machines: theory and applications. Springer, Berlin

Wu HN, Shen SL, Yang J, Zhou AN (2018) Soil-tunnel interaction modelling for shield tunnels considering shearing dislocation in longitudinal joints. Tunn Undergr Sp Technol 78:168–177

Yao YP, Hu J, Zhou AN, Luo T, Wang N (2015) Unified strength criterion for soils, gravels, rocks, and concretes. Acta Geotech 10(6):749–759

Zhang W, Goh AT (2016) Multivariate adaptive regression splines and neural network models for prediction of pile drivability. Geosci Front 7(1):45–52

Zhang N, Shen SL, Zhou AN, Arulrajah A (2018) Tunneling induced geohazards in mylonitic rock faults with rich groundwater: a case study in Guangzhou. Tunn Undergr Sp Technol 74:262–272

Zhang N, Shen SL, Zhou AN, Xu YS (2019) Investigation on performance of neural networks using quadratic relative error cost function. IEEE Access 7:106642–106652. https://doi.org/10.1109/ACCESS.2019.2930520