

## Use of machine learning techniques for predicting the bearing capacity of piles

Yago Ferreira Gomes<sup>1</sup> , Filipe Alves Neto Verri<sup>2</sup> , Dimas Betioli Ribeiro<sup>1#</sup> 

Article

### Keywords

Concrete precast piles  
Bearing capacity  
Semi-empirical methods  
Machine learning

### Abstract

Geotechnical engineers frequently rely on semi-empirical methods like Décourt-Quaresma and Meyerhof's to estimate the bearing capacity of piles. This paper proposes alternatives to these methods, presenting an approach using machine learning models for predicting the bearing capacity of precast concrete piles. It uses data samples including 165 load tests, each one accompanied with a SPT sounding. This study proposes two types of analysis using two separated datasets, one based on the Décourt-Quaresma method and the other based on the Meyerhof method. Six machine learning algorithms of distinct biases are trained and tested with a leave-one-out cross validation procedure and the models' predictive performance is assessed through two metrics: root mean squared error (RMSE) and coefficient of determination ( $R^2$ ). The best performing technique was random forest (RF) using Décourt-Quaresma dataset, with an RMSE of 642.38. All other machine learning techniques obtained a RMSE below 710, overcoming Meyerhof's and Décourt-Quaresma's semi-empirical methods, which both obtained RMSE values close to 900. This study proposes 95% and 90% confidence intervals for the best technique employing a graphical interpretation, so that geotechnical engineers can choose which level of safety they wish to work with. Finally, the study presents a case study showing that the best performing models achieve a reasonable accuracy, surpassing the semi-empirical methods in two of the three piles considered. The representativity of the new examples within the used datasets explain the accuracy of the techniques.

## 1. Introduction

Designers need to estimate the load bearing capacity of piles and the most precise way is through static pile load tests. The Brazilian Standard (ABNT 2019) define procedures for this type of test, which basically consists on applying an increasing load to an executed pile and measuring its displacement. Designers can obtain the load bearing capacity by examining the load-displacement graph, using criteria defined by the standard. Nonetheless, they cannot rely only on static pile load tests because they are expensive, time consuming and usually executed when part of the piles of the project are already in place. The most popular approach to estimate pile bearing capacity beforehand is to use semi-empirical methods, like those proposed by Aoki & Velloso (1975), Décourt & Quaresma (1978, 1998) and Meyerhof (1976). Most semi-empirical methods propose two separate estimates: one for the shaft resistance and another for the tip resistance. The total pile bearing capacity given by the sum of them. These methods usually estimate the bearing

capacity through results of in situ tests and pile geometric features. In several countries (including Brazil), contractors usually only make available the standard penetration test (SPT). The main reasons are cost and simplicity when compared to methods like the cone penetration test, making the SPT popular in those countries. Even when designers do have access to other in situ tests, they sometimes rely on correlations to convert data into SPT values.

In recent years, machine learning techniques are increasingly gaining space within a wide variety of engineering applications. Their advantages include the capability to deal with large amounts of data and to find complex and highly nonlinear relationships among different parameters. In geotechnics many works have been using these algorithms to solve different kinds of problems with good results over traditional methods. Some of these problems are: soil classification (Bhattacharya & Solomatine, 2006; Kovacevic et al., 2010; Bonini et al., 2017; Carvalho & Ribeiro 2019), soil profiling (Arel, 2012), soil liquefaction (Juang & Chen, 1999; Hanna et al., 2007; Goh & Goh,

<sup>#</sup>Corresponding author. E-mail address: [dimasbetioli@gmail.com](mailto:dimasbetioli@gmail.com)

<sup>1</sup>Instituto Tecnológico de Aeronáutica, Divisão de Engenharia Civil, São José dos Campos, SP, Brasil.

<sup>2</sup>Divisão de Ciência da Computação, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brasil.

Submitted on September 6, 2021; Final Acceptance on Outubro 27, 2021; Discussion open until February 28, 2022.

<https://doi.org/10.28927/SR.2021.074921>



This is an Open Access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

2007; Livingston et al., 2008; Kohestani et al., 2015), slope stability (Ferentinou & Sakellariou, 2007; Bui et al., 2019; Maxwell et al., 2020), foundation settlement (Samui, 2008; Nejad et al., 2009; Nejad & Jaksa, 2017) and bearing capacity. For the latter, authors have used well-known algorithms with different kinds of inputs, such as wave stress data (Lok & Che, 2004; Pal & Deswal, 2008; Maizir & Kassim, 2013; Momeni et al., 2015), soil parameters (Pal & Deswal, 2010; Moayed & Hayati, 2019; Kardani et al., 2020), CPT results (Ardalan et al., 2009; Shahin, 2010; Kordjazi et al., 2014; Kordjazi et al., 2013; Alkroosh et al., 2015) and SPT measures (Lee & Lee, 1996; Nawari & Liang, 2000; Ismail & Jeng, 2011; Benali et al., 2018; Jesswein & Liu, 2018; Pham et al., 2020). These studies achieved reasonable results for the proposed problems. Nevertheless, most of them include weaknesses like comparing few techniques or using very homogeneous datasets, with all soundings taken from the same construction site.

The main objective of this paper is to propose a new approach for the use of machine learning techniques, using classical semi-empirical methods as a basis for estimating the bearing capacity of piles. It is better than previous machine learning models from the literature concerning generality for tropical soils and ease of use. The used datasets include only static load tests (slow maintained load) of pre-cast concrete piles executed accordingly to the Brazilian standard (see ABNT 2019) and accompanied with SPT soundings. The investigation starts with the training of six machine learning techniques, producing two models for each one: the first using the inputs from the Décourt-Quaresma method and the second the inputs from the Meyerhof method. A multiple linear regression (LR) is also included as a baseline for performance. The authors selected the Décourt-Quaresma method because it is commonly used in Brazilian foundation projects and Meyerhof for being widely used worldwide. Both sets of inputs include pile diameter and length, the mean SPT along the shaft and the mean SPT at the pile tip. The main difference of the two sets is how mean SPT values are calculated.

It is shown in a general application that the precision of all machine learning techniques surpassed both Meyerhof (1976) and Décourt & Quaresma (1978, 1998) semi-empirical methods with respect to RMSE. This work proposes a graphical method to provide 90% and 95% confidence intervals for the results of the best technique. A case study applies the top two machine learning models and the two semi-empirical methods to three new examples, from one site that was not included in the training dataset. The machine learning techniques presented reasonable performance, and were better than the semi-empirical methods in two of the three piles.

## 2. Semi-empirical methods

Semi-empirical methods work based on empirical correlations of in situ tests data and adjustments with load test

results. Results can vary for these methods due to their implicit subjectivity. For the Meyerhof method, little subjectivity was included because it uses  $N_{SPT}$  and pile geometry as inputs, which are not sensitive to interpretation. On the other hand, the Décourt-Quaresma method relies on soil types as presented in Table 1, which are sensitive to interpretation.

In these methods, the pile load capacity  $R_t$  is usually divided into two parts: lateral friction  $R_l$  and tip resistance  $R_p$ , as shown in Equation 1. Different expressions are proposed for  $R_l$  and  $R_p$  in the literature, using information such as soil type, pile type, pile geometry and in situ test results.

$$R_t = R_l + R_p \quad (1)$$

The authors selected two methods for this study: the Décourt-Quaresma method for being popular in Brazil and the Meyerhof method for being widely used around the world. The next sections describe these methods.

### 2.1. Décourt-Quaresma

This method obtains the tip resistance using a factor related to the soil type, as presented in Table 1. It also uses the tip area  $A_p$  and the mean  $N_{SPT}$  index around the pile tip  $SPT_p$ , considering the value at the tip, the one above and the one below. It obtains the lateral resistance using pile geometry and the mean  $N_{SPT}$  index along the pile shaft  $SPT_l$ . The latter is subjected to  $3 < SPT_p < 15$  and  $N_{SPT}$  values used to obtain  $SPT_p$  cannot be included. The final expression is

$$R_t = \alpha \cdot K \cdot SPT_p \cdot A_p + U \cdot \beta \cdot 10 \left( \frac{SPT_l}{3} + 1 \right) L \quad (2)$$

where  $\alpha$  and  $\beta$  refer to soil and pile type, respectively.  $U$  is the pile perimeter and  $L$  is the pile length.

### 2.2. Meyerhof

This method uses the  $N_{SPT}$  index, pile length  $L$  and pile diameter  $D$  to estimate the pile bearing capacity. It calculates  $SPT_l$  as the mean of the whole pile shaft and  $SPT_p$  as the mean including  $8D$  above the tip and  $3D$  below it (Meyerhof, 1976). The expression proposed by Meyerhof is

**Table 1.** Values for  $K$  (Décourt & Quaresma, 1978, 1998).

Soil type	$K$ (KN/m <sup>2</sup> )
Clays	120
Clayey silts	200
Sandy silts	250
Sands	400

$$R_t = A_p \cdot q_p + U \cdot L \cdot q_s \quad (3)$$

where

$$q_p = 40 \cdot L \frac{SPT_p}{D} \leq 400 \cdot SPT_p \quad (4)$$

and

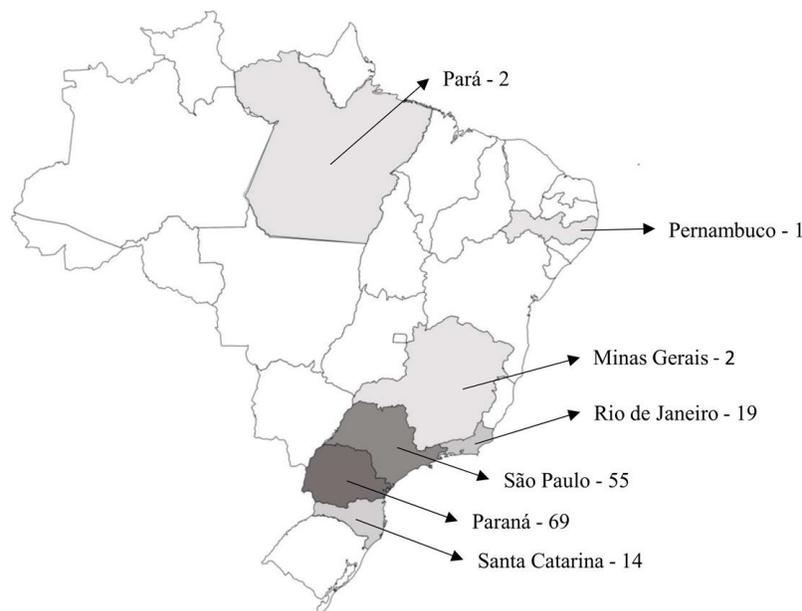
$$q_s = 2 \cdot SPT_l \quad (5)$$

### 3. Dataset

The information used in this work includes 165 precast concrete pile load tests and their respective SPT measures collected from many different construction sites in Brazil. It was obtained from the works of Lobo (2005), Vianna (2000) and Santos (1988) and all load tests were performed according to the Brazilian Standard (ABNT, 2019). When the maximum applied load was not achieved, the load-settlement curve was extrapolated using the Van der Veen method (Van Der Veen, 1953). Interested readers can find further detail about these load tests in Lobo (2005), Vianna (2000) and Santos (1988). In specific cases, information about pile rupture and comparisons between applied and ultimate loads is available.

Figure 1 presents the location of the soundings, most of them from the south and southeast regions of Brazil. The country presents a predominant tropical climate and high temperatures, with 65% of its territory covered by non-homogeneous lateritic soils. The clay-ferruginous soil is the most common type (Morais et al., 2020). The authors had access to some details about the set provided by Vianna (2000), which is composed by soundings taken from the city of Curitiba, in Paraná state. The geology of this region can be divided into three groups: a metamorphic rock complex from the Precambrian; sedimentary deposits from Tertiary; and a more recent sedimentary deposit (Holocene), as a result of a partial removal of older sediments (Cenozoic). This entire sequence of Cenozoic sediments in the Curitiba Basin is named Guabirotuba Formation in the literature (Bigarella & Salamuni, 1962).

After assembly, raw data was preprocessed into two datasets. The first, named Décourt dataset, uses  $SPT_p$ ,  $SPT_l$ ,  $D$  (in cm) and  $L$  (in m) as calculated in the Décourt-Quaresma method. The second uses the same inputs, but defined accordingly to the Meyerhof method. Notice that the difference is how each method calculates  $SPT_p$  and  $SPT_l$ , as presented in previous sections. The authors did not include soil type among the inputs because, based on their previous experience, these variables do not contribute to improve accuracy and include too much human error. Thus, although the authors consider the position of the water table relevant for the problem, they decided not to include it because many of the used soundings did not include this information. Figure 2 illustrates the variables used in each set. The target feature is the pile bearing load capacity  $Q_u$ .



**Figure 1.** Number of soundings at each Brazilian state.

in kN, obtained from the load test. Tables 2 and 3 present a sample of each dataset. Unity  $b$  refers to the number of blows needed for the sampler to penetrate 30 cm into the soil (Salgado, 2008).

Tables 4 and 5 present correlation matrices generated for each dataset. Inputs are not severely correlated, with all values within the interval  $[-0.7, 0.7]$ . This indicates that they can be all considered informative, occurring few redundancies

between them. Notice that the correlation between  $SPT_l$  and  $SPT_p$  is only 0.35 for the Décourt dataset, while it rises to 0.7 for the Meyerhof dataset. This can be explained by the way each method obtains these variables, with completely separated soil layers considered for the Décourt dataset and an intersection of common soil layers considered for the Meyerhof dataset (see Figure 2).  $D$  and  $L$  are the ones with stronger correlation to the output  $Q_u$ , which was expected.

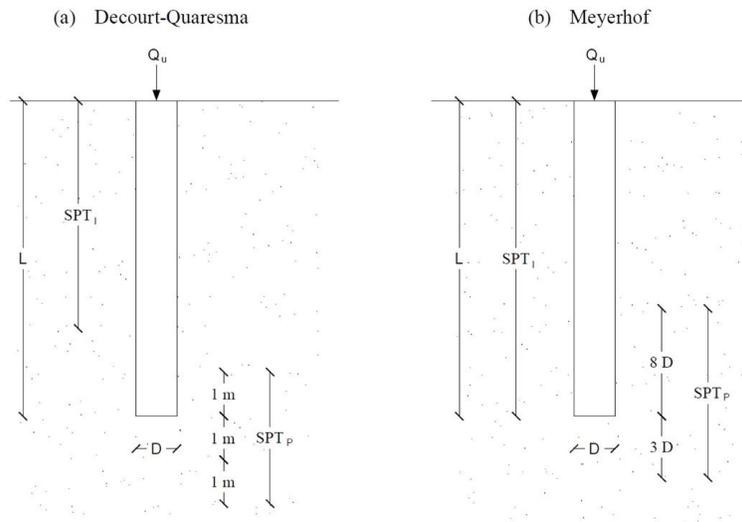


Figure 2. a) Décourt-Quaresma parameters. b) Meyerhof parameters.

Table 2. Sample from Décourt dataset.

$N$	$SPT_l$ (b)	$SPT_p$ (b)	$L$ (m)	$D$ (cm)	$Q_u$ (kN)
1	6.46	27.33	18.90	26	1115
2	7.94	27.33	21.12	26	1005
...	...	...	...	...	...
165	22.71	42.25	7.00	40	1800

Table 3. Sample from Meyerhof dataset.

$N$	$SPT_l$ (b)	$SPT_p$ (b)	$L$ (m)	$D$ (cm)	$Q_u$ (kN)
1	4.16	27.33	18.90	26	1115
2	6.04	27.33	21.12	26	1005
...	...	...	...	...	...
165	12.80	51.67	7.00	40	1800

Table 4. Correlation matrix for Meyerhof dataset.

	$SPT_l$ (b)	$SPT_p$ (b)	$L$ (m)	$D$ (m)	$Q_u$ (KN)
$SPT_l$ (b)	1	0.70	-0.25	0.11	0.13
$SPT_p$ (b)	0.70	1	0.11	0.33	0.39
$L$ (m)	-0.25	0.11	1	0.62	0.66
$D$ (m)	0.11	0.33	0.62	1	0.84
$Q_u$ (KN)	0.13	0.39	0.66	0.84	1

**Table 5.** Correlation matrix for Décourt dataset.

	$SPT_l$ (b)	$SPT_p$ (b)	$L$ (m)	$D$ (m)	$Q_u$ (kN)
$SPT_l$ (b)	1	0.35	-0.15	0.06	0.09
$SPT$ (b)	0.35	1	0.28	0.51	0.56
$L$ (m)	-0.15	0.28	1	0.62	0.66
$D$ (m)	0.06	0.51	0.62	1	0.84
$Q_u$ (kN)	0.09	0.56	0.66	0.84	1

## 4. Model description

After pre-processing, this study uses both datasets to train a set of selected machine learning algorithms. First step is organizing each dataset as a matrix, where each column represents an input or the output and each line represents an example. In other words, each dataset becomes a  $165 \times 5$  matrix. Next, it divides the examples (lines) of each dataset into two portions: the training set and the test set. This work uses the leave-one-out cross validation approach, using the full dataset for training except for one example kept apart for test. The procedure tests all examples and the final accuracy is given by the mean (Wong, 2015).

The coefficient of determination  $R^2 \in [0,1]$  is one metric used in this work to evaluate the performance of the algorithms. It is obtained using Equation 6, where  $\hat{y}_i$  is a predicted value obtained from the model,  $y_i$  is an observed value from dataset,  $\bar{y}$  the mean of all observed values and  $ne$  is the number of examples. In this work,  $y_i$  is the pile bearing load capacity ( $Q_u$ ) of a specific pile  $i$ .  $R^2$  values close to 1 imply that the target variable is completely explained by the used model. 0 means no connection between predicted and observed values. The literature considers this metric a meaningful indicator of accuracy (Draper & Smith, 1998).

$$R^2 = \frac{\sum_{i=1}^{ne} (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{ne} (y_i - \bar{y})^2} \quad (6)$$

Other performance metric adopted in this work is the root mean square error (RMSE). It is calculated for all machine learning models and is given by Equation 7.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{ne} (\hat{y}_i - \bar{y})^2}{ne}} \quad (7)$$

The machine learning techniques used in this work are k-nearest neighbor (KNN), kernel-KNN (KKNN), decision tree (DT), random forest (RF), artificial neural networks (ANN) and support vector machines (SVM). The following subsections present them, with a brief overview of its functionality. They were chosen considering their popularity within machine learning applications, their different biases and

their reasonable results towards this work dataset. Multiple linear regression (LR) is also included as a baseline for the performance of the techniques.

### 4.1 KNN and KKNN

The KNN technique understands each example as a point whose coordinates are the inputs. It expects that a new example would have an output similar to those that are close in this input space. The regression problem can use Equation 8, which defines the output of the new example as the average value of its  $k$  nearest neighbors.

$$\hat{y}_i = \frac{1}{k} \sum_{j=1}^k y_j \quad (8)$$

This work weights the output of each neighbor with respect to its distance to the new example, giving more weight to closer ones to improve accuracy (Dudani, 1976). It calculates the distance using the Minkowski metric, as presented in Equation 9. In this work  $p = 2$ , which leads to the Euclidian metrics. Equation 9 gives the distance between arbitrary points represented by vectors  $\mathbf{a}$  and  $\mathbf{b}$ , with components  $(a_1, \dots, a_n)$  and  $(b_1, \dots, b_n)$ , considering an n-dimensional space.

$$d(\mathbf{a}, \mathbf{b}) = \left( \sum_{i=1}^n |a_i - b_i|^p \right)^{1/p} \quad (9)$$

KNN has the disadvantage of poor performance for some type of complex problems (Kuo et al., 2008). The KKNN technique solves this problem by changing the distribution of samples, mapping them into a higher dimensional input space. The objective is to obtain a linear problem in this new space, equivalent to the nonlinear problem of the original space. Equation 10 presents an example of mapping a n-dimensional input space into a m-dimensional space:

$$\begin{aligned} \mathbf{a} = (a_1, \dots, a_n) &\xrightarrow{\text{mapping}} \psi(\mathbf{a}) = (\varphi_1(\mathbf{a}), \dots, \varphi_m(\mathbf{a})), \\ \mathbf{a} \in S_1, \psi(\mathbf{a}) \in S_2 \end{aligned} \quad (10)$$

$S_1$  is the original n-dimension space and  $S_2$  the new m-dimension space.  $\mathbf{a}$  is a vector in  $S_1$  and  $\psi(\mathbf{a})$  is its corresponding vector in  $S_2$ .  $\psi$  defines the mapping from  $S_1$  to  $S_2$  and  $\varphi_i, i = 1, \dots, m$ , are input mapping functions. One problem in this approach is that finding  $\psi$  is usually impracticable. Nevertheless, the mapping does not require  $\psi$  if the internal product  $\psi(\mathbf{a}) \cdot \psi(\mathbf{b})$  is known for arbitrary vectors  $\mathbf{a}$  and  $\mathbf{b}$ . This inner product is called kernel (Yu et al., 2002).

The most commonly used kernel functions are: polynomial, radial basis and sigmoid, as shown in Equations 11, 12 and 13, respectively:

$$K(\mathbf{a}, \mathbf{b}) = (1 + \mathbf{a} \cdot \mathbf{b})^\rho \tag{11}$$

$$K(\mathbf{a}, \mathbf{b}) = \exp\left(-\frac{\mathbf{a} \cdot \mathbf{b}^2}{\sigma^2}\right) \tag{12}$$

$$K(\mathbf{a}, \mathbf{b}) = \tanh(\gamma \cdot \mathbf{a} \cdot \mathbf{b} + \omega) \tag{13}$$

where  $\rho, \sigma, \gamma$  and  $\omega$  are adjustable parameters and  $\mathbf{a} \cdot \mathbf{b}$  is the inner product between vectors  $\mathbf{a}$  and  $\mathbf{b}$ . This work uses the radial basis kernel based on preliminary tests.

#### 4.2 DT and RF

A DT model is a flow-chart-like structure, with nodes that create ramifications dividing the dataset. It starts with a single root node that receives the complete dataset and distributes it to other nodes using a rule, which is usually an inequality applied to one of the inputs. New nodes receive portions of the dataset, subjects them to their rules and

distributes them to other nodes, forming the branches of the tree. The last nodes, called leaves, assign outputs to the examples. Figure 3 presents a scheme of a DT.

One disadvantage of DTs is that they tend to become overspecialized in the dataset used for training, which prejudices performance for new examples. This behavior is called overfitting. RF is a technique based on DTs that minimizes this problem by using a collection of different DTs built randomly. The algorithm selects a subset of examples for each tree and node, ensuring that they are different. After RF creates the trees, each one make a separate prediction and the mean gives the final value (Ho, 1995).

#### 4.3 ANN

The interaction of neurons in the human brain inspires the ANN algorithm. Its structure consists of a number of processing elements or nodes that are arranged in layers: an input layer, an output layer and one or more hidden layers. Each node from the first layer receives an input  $x_i$ , which is multiplied by an adjustable connection weight  $w_{ij}$ . These values are inputs for the neurons of the next layer, that sum them and add a threshold value  $\theta_j$  to obtain a combined input  $I_j$ . Then, the algorithm applies an activation function  $f(I_j)$  to produce the output  $o_j$ , as expressed in Equations 14 and 15.

$$I_j = \sum w_{ij} x_i + \theta_j \tag{14}$$

$$o_j = f(I_j) \tag{15}$$

This work uses a sigmoid function for activation, which is expressed as:

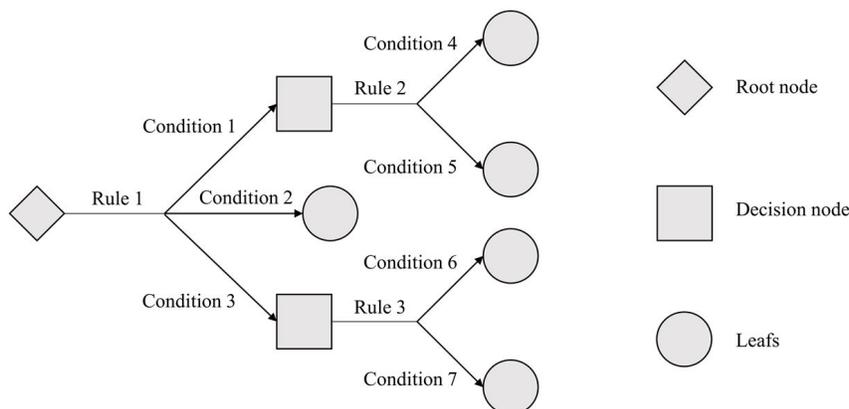


Figure 3. Example of a decision tree.

$$f(I_j) = \frac{1}{1 + e^{-\lambda I_j}} \quad (16)$$

where  $\lambda$  is a calibration parameter.

#### 4.4 SVM

SVMs use statistical learning principles as a basis. Their main objectives are minimizing errors associated with the training dataset and maximizing the generalization of the model (Vapnik, 1999). The algorithm uses a set of functions for its regression model that can have a solution as given in Equation 17:

$$\mathbf{y} = \mathbf{w} \cdot \mathbf{x} + \mathbf{t} \quad (17)$$

where  $\mathbf{x} = (x_1, y_1), \dots, (x_l, y_l)$ ,  $\mathbf{x} \in R^m$  is the input of  $l$  samples and  $m$  dimensions,  $\mathbf{y} \in R^m$  is the output,  $\mathbf{w}$  is the weight vector and  $\mathbf{t}$  is the bias. The margin is a distance from the hyperplane which is set to contain all points, as illustrated in Figure 4. This distance is the error  $\epsilon$  to be minimized, included in Equation 18 as follows:

$$\mathbf{y} = \mathbf{w} \cdot \mathbf{x} + \mathbf{t} \pm \epsilon \quad (18)$$

Equation 19 presents the function to be minimized.  $\xi_i$  and  $\xi_i^*$  are parameters introduced to penalize points outside the margins and parameter  $C$  controls these penalties (Smola & Scholkopf, 2004). The algorithm solves this optimization problem using Lagrange multipliers (Vapnik, 1998).

$$\frac{1}{2} \mathbf{w}^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (19)$$

This procedure is valid for linear problems. One can extend it to nonlinear problems using kernels to map the input data into a higher dimensional space. It is the same approach described for the KNN. The authors chose radial basis functions after observing better accuracy in preliminary tests.

#### 4.5 LR

A LR seeks a linear relationship between the input variables and the output. Equation 20 represents the model generated by this kind of regression:

$$\hat{y}_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n \quad (20)$$

where  $\hat{y}_i$  is the predicted variable,  $\beta_j$  are the coefficients determined by the model and  $x_j$  are the input values for the problem.

This technique has the advantage of being simple and widely used in geotechnical engineering practice, but it cannot reproduce non-linear behavior. Although it is not expected to obtain good results from this technique, it is included in this work as one of the baselines for the performance of the machine learning techniques.

## 5. Results and discussion

### 5.1 General application

The objective of this example is to apply the six machine learning techniques to Décourt and Meyerhof datasets, using RMSE and  $R^2$  metrics to evaluate performance. The baselines for performance are the original semi-empirical methods and LR.

Tables 6 and 7 present the performance obtained using the Décourt and Meyerhof datasets, respectively. RF was the technique with best accuracy, presenting the lowest RMSE in both tables. The second best was KNN for Décourt, followed

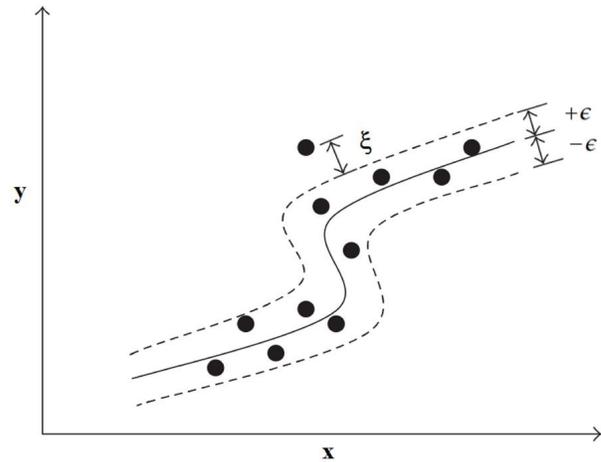


Figure 4. Graphical representation of Support Vector Machines.

Table 6. Performance metrics using Décourt dataset.

Algorithm	RMSE (kN)	$R^2$
RF	642.38	0.765
KNN	651.24	0.762
ANN	659.08	0.752
KKNN	665.37	0.750
LR	684.22	0.732
SVM	686.24	0.744
DT	704.40	0.717

by ANN and KKNN. For Meyerhof, the second best was ANN followed by KKNN and KNN. DT presented the worst performance, which can be explained by the tendency of this technique to overfit. SVM presented poor performance as well, worse than LR which is the baseline.

Table 8 presents a comparison between the performance of the semi-empirical methods (Décourt-Quaresma and Meyerhof), the LR and the RF algorithms. The subscript Dq indicates the use of the Décourt dataset for training. One can observe that even LR surpass the semi-empirical method of Meyerhof, encouraging the use of machine learning techniques

**Table 7.** Performance metrics using Meyerhof dataset.

Algorithm	RMSE (kN)	
RF	651.16	0.758
ANN	660.44	0.751
KKNN	676.31	0.742
KNN	679.90	0.741
LR	694.52	0.724
SVM	704.65	0.728
DT	706.02	0.715

**Table 8.** Performance results by method.

Method	RMSE (kN)	$R^2$
RF <sub>Dq</sub>	642.38	0.765
LR <sub>Dq</sub>	684.22	0.732
Meyerhof	896.08	0.662
Décourt-Quaresma	909.98	0.748

for this type of problem. This conclusion is corroborated by other studies (Lee & Lee, 1996; Pham et al., 2020).

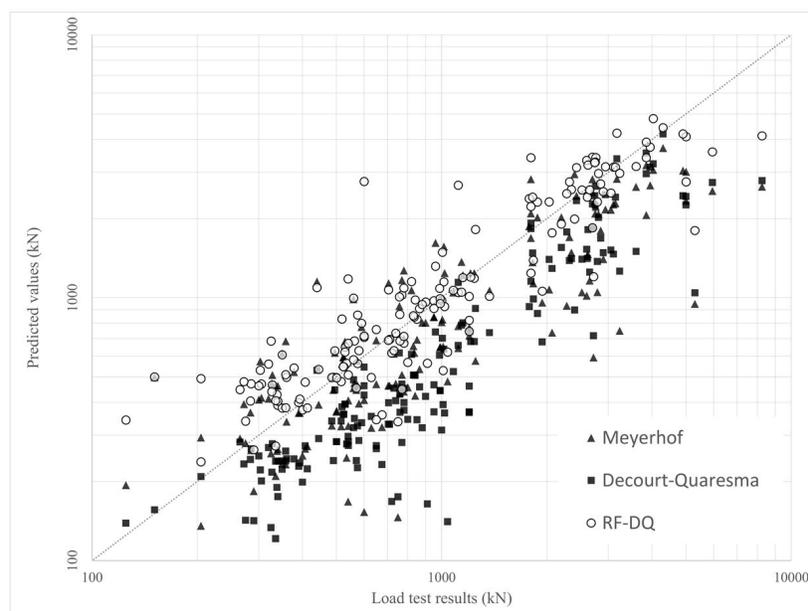
Figure 5 complements the comparisons presented in Table 8, with abscissas representing measured values and ordinates representing predicted values. This study uses logarithm scale to better represent the wide range of values. A predicted value equal to the observed one produces a point at the black line, while poor predictions tend to produce points far from it. Note that the cloud of white circles, that represents RF<sub>Mey</sub>, is clearly more concentrated around the black line than black squares and triangles, which represent semi-empirical methods. It is also shown that the semi-empirical methods tend to underestimate the load bearing capacity, while points from RF<sub>Mey</sub> tend to be split in half by the black line.

In order to present a complementary analysis, Table 9 presents predicted values using RF<sub>Dq</sub>, the corresponding measured values  $Q_u$  and the ratio between them. This ratio is organized in ascending order including all 165 load tests, with a range from 0.340 to 4.62. The objective is to produce confidence intervals for the RF<sub>Dq</sub>/ $Q_u$  values, allowing a better understanding of the accuracy of the algorithm.

The authors first verify whether the RF<sub>Dq</sub>/ $Q_u$  results follow a normal distribution using the Shapiro-Wilk test.

**Table 9.** Relation between predicted and measured values.

RF <sub>Dq</sub> (kN)	$Q_u$ (kN)	RF <sub>Dq</sub> / $Q_u$
1803	5300	0.340
1203	2720	0.442
...	...	...
2770	600	4.62



**Figure 5.** Graphical comparison between methods studied in this work.

The procedure uses a significance level of 5% and a starting null hypothesis  $H_0$  that data follows a normal distribution. However, when calculating the p-value, the result was far below 5%, indicating that the data does not have a normal behavior. This means that it is not possible to apply the confidence level theory for this distribution.

To solve this problem, this study proposes a less rigorous approach using the concept of percentiles. The  $n^{\text{th}}$  percentile is a value greater than  $n$  percent of all values in the list. The authors use the  $RF_{Dq} / Q_u$  ordered list from Table 9 to estimate the confidence interval, considering that it must be centralized in the list with respect to the percentiles. The analysis proposes two confidence intervals: one of 90%, that must be limited by the 5<sup>th</sup> and 95<sup>th</sup> percentiles, and one of 95%, that must be limited by the 2.5<sup>th</sup> and 97.5<sup>th</sup> percentiles. This procedure resulted  $[0.603, 2.185]$  for the 90% confidence interval of  $RF_{Dq} / Q_u$  and  $[0.559, 2.170]$  for its 95% confidence interval.

Figure 6 illustrates these results. Abscissa axis represents measured values, the ordinate axis represents predicted values and each point represents a  $RF_{Dq} / Q_u$  value. The continuous line is the locus of points with  $RF_{Dq} = Q_u$ , while the other lines represent the limits of the confidence intervals. Considering engineering practice, this graph can give to geotechnical

engineers a sense of which confidence interval would suit better their specific case.

### 5.2 Case study

This section presents a case study with new examples to validate the generated models. The analysis uses results taken from three SPT soundings and load tests of precast concrete piles located in a construction site in Monte Largo, Paraná state, Brazil. These examples came from Lobo (2005) and were not used to train the machine learning techniques. The objective is to evaluate what would be the accuracy of the models if applied in the future to a completely new site. Figure 7 presents the SPT values and load test information.

The study starts calculating the results obtained with the original semi-empirical methods of Décourt-Quaresma and Meyerhof, as well as for the best performing techniques for each dataset. To facilitate comparisons, Table 10 presents all relative errors. For a predicted value  $\hat{y}_i$  and an observed value  $y_i$ , the relative error is  $RE_i = |\hat{y}_i - y_i| / y_i$ .

Note that the first example seems to be more difficult than the other two. One possible explanation for this disparity is the soil of this examples, with the first underrepresented within the training datasets. This issue is investigated incorporating

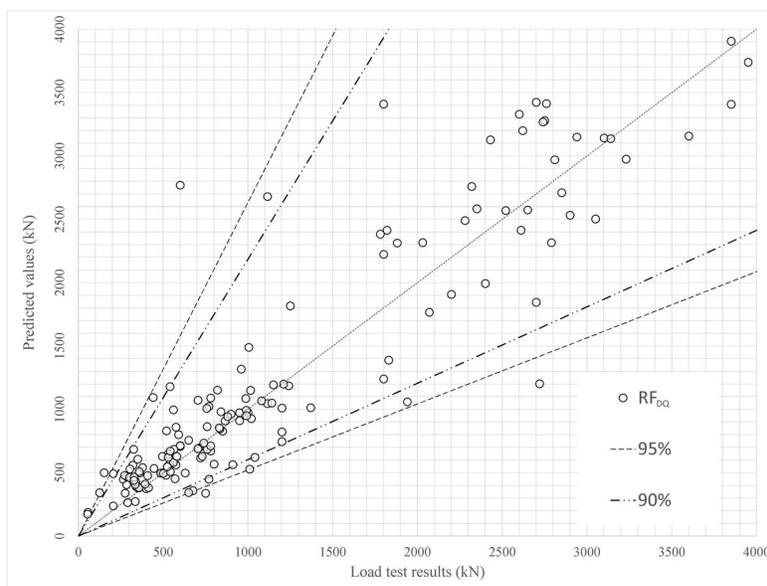
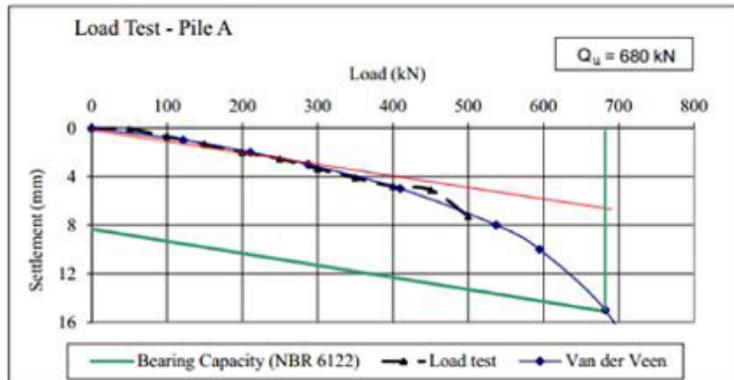


Figure 6. Confidence intervals for ratio between observed and predicted values.

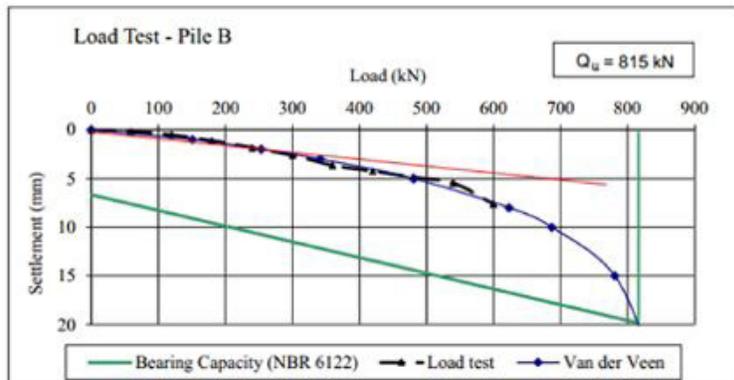
Table 10. Relative errors for the load tests from Monte Largo.

Relative error					
Décourt-Quaresma			Meyerhof		
Déc	$RF_{Dq}$	$KNN_{Dq}$	Mey	$RF_{Mey}$	$ANN_{Mey}$
6.48	26.26	24.12	35.60	33.62	53.94
40.04	10.50	19.39	31.50	14.44	1.24
43.81	2.46	8.45	32.71	11.50	0.41

Pile A		
D = 25 cm		
L = 16.2 m		
Depth (m)	NSPT	Soil type
1	8	Silty Clay
2	7	
3	5	
4	6	
5	5	
6	4	
7	5	Sandy Clay
8	6	
9	7	
10	14	
11	8	
12	7	
13	9	Sandy Clay
14	14	
15	19	
16	35	
17	40	
18	40	



Pile B		
D = 25 cm		
L = 16.8 m		
Depth (m)	NSPT	Soil type
1	8	Silty Clay
2	7	
3	6	
4	13	
5	12	
6	9	Silty Sand
7	8	
8	4	
9	5	Silty Clay
10	5	
11	7	
12	6	
13	8	Clayey Silt
14	13	
15	12	
16	14	
17	10	
18	17	
19	15	
20	18	



Pile C		
D = 25 cm		
L = 15.35 m		
Depth (m)	NSPT	Soil type
1	16	Slity Clay
2	12	
3	15	
4	8	
5	6	Silty-sandy Clay
6	2	
7	2	
8	2	Silty Clay
9	15	
10	10	
11	9	
12	7	
13	10	Clayey Silt
14	11	
15	7	
16	8	
17	19	
18	20	

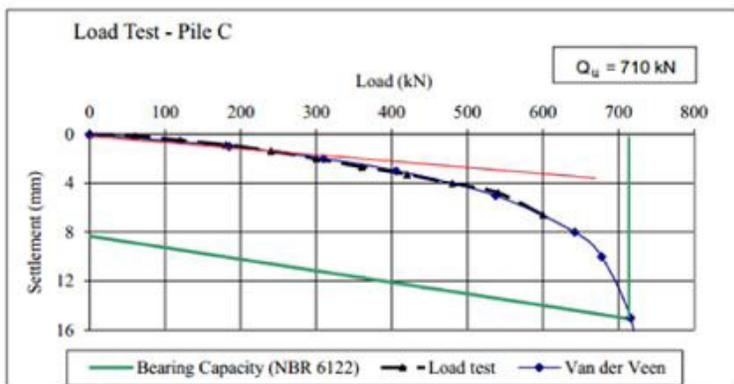


Figure 7. SPT sounding and load test for the pile tests. Adapted from Lobo (2005).

**Table 11.** Relative error for predictions made with the updated datasets.

Relative error			
Décourt-Quaresma		Meyerhof	
RF <sub>Dq</sub>	KNN <sub>Dq</sub>	RF <sub>Mey</sub>	ANN <sub>Mey</sub>
25.62	24.12	31.72	53.94
9.25	20.65	6.55	1.55
2.59	3.47	11.64	0.61

these piles to the datasets, to verify if performance changes. The objective is verifying if the inclusion of two of the load tests of this construction site helps predicting the third one, as performed in the leave-one-out methodology. Table 11 presents the result.

One can observe that most machine learning techniques presented some improvement for the first example, which is still the most difficult. For the other two, although some specific values increased, the overall performance of the techniques can be considered better. This allows concluding that the inclusion of information from the same construction site helped improving performance. In other words, the performance of the techniques for new examples depends on its representativity within the training dataset.

## 6. Conclusions

This work applies machine learning techniques to predict the bearing capacity of concrete precast piles. It presents two examples, the first with a general application and the second with a case study. The results obtained in the first example, considering all techniques applied to both datasets, allows concluding that RF is the best algorithm for this problem, with lower RMSE values. KNN and ANN also detached from the others, presenting the second best performance for Décourt-Quaresma and Meyerhof datasets, respectively. The semi-empirical methods of Décourt-Quaresma and Meyerhof presented a relatively poor performance in this example with an RMSE close to 900, being surpassed by all other techniques including LR. These results demonstrate that machine learning algorithms are a good alternative for predicting the ultimate bearing capacity of piles. The analysis proposed an approximation of the confidence intervals using the concept of percentile. A graph presented two intervals, 90% and 95%, to give engineers choices for the desired accuracy.

The second example presented a study to evaluate the effect of the representativity of the dataset. Results confirm that performance depends on representativity and also reveal the limits of these models, which tend to present poor accuracy for examples very different from the ones contained in the used datasets.

## Acknowledgements

To the Coordination of Improvement of Higher Education Personnel (CAPES) for granting the first author's scholarship.

## Declaration of interest

The authors have no conflicts of interest to declare. All co-authors have observed and affirmed the contents of the paper and there is no financial interest to report.

## Authors' contributions

Yago Ferreira Gomes: conceptualization, data curation, formal analysis, investigation, methodology, visualization, writing – original draft. Filipe Alves Neto Verrri: conceptualization, methodology, supervision, validation, visualization, writing – review & editing. Dimas Betioli Ribeiro: conceptualization, data curation, funding acquisition, methodology, project administration, supervision, validation, writing – review & editing.

## List of symbols

$a, b$ :	Arbitrary vectors
$a_p, b_i$ :	Vector components
$b$ :	Number of blows needed for the sampler to penetrate 30 cm into the soil
$A_p$ :	Area of the pile tip
$f$ :	Activation function
$I_j$ :	Combined input of a neuron
$k$ :	Number of nearest neighbors
$K$ :	Soil type factor
$K(a, b)$ :	Kernel
$l$ :	Number of samples for a SVM
$L$ :	Pile length
$n, m$ :	Space dimensions
$ne$ :	Number of examples
$o_j$ :	Output of a neuron
$p$ :	Exponent of Minowsky equation
$q_p, q_s$ :	Parameters of Meyerhof's method
$Q_u$ :	Pile bearing load capacity
$R^2$ :	Coefficient of determination
$RE_i$ :	Relative error
$R_i$ :	Lateral friction
RMSE:	Root mean square error
$R_p$ :	Tip resistance
$R_i$ :	Pile load capacity
$S_1, S_2$ :	Spaces of dimension n and m, respectfully
$SPT_i$ :	Mean $N_{SPT}$ index along the pile shaft

$SPT_p$ :	Mean $N_{SPT}$ index around the pile tip
$U$ :	Pile perimeter
$w$ :	Weight vector
$w_{ij}$ :	Adjustable connection weight
$x$ :	Input of a SVM
$x_i$ :	Input of a neuron
$x_i, y_i$ :	Components of $x$ and $y$ , respectfully
$y$ :	Output of a SVM
$y_i$ :	Observed value
$\hat{y}_i$ :	Predicted value
$\bar{y}$ :	Mean of all observed values
$\xi_i, \xi_i^*, C$ :	Parameters of a SVM
$\alpha, \beta$ :	Parameters of Décourt-Quaresma method
$\theta_j$ :	Threshold value
$\rho, \sigma, \gamma, \omega$ :	Kernel parameters
$\varphi_i$ :	Component of $\psi$
$\beta_j$ :	Coefficients to be determined for a LR
$\iota$ :	Bias of a SVM
$\lambda$ :	Calibration parameter
$\psi$ :	Mapping vector
$\epsilon$ :	Error to be minimized

## References

- Alkroosh, I., Bahadori, M., Nikraz, H., & Bahadori, A. (2015). Regressive approach for predicting bearing capacity of bored piles from cone penetration test data. *Journal of Rock Mechanics and Geotechnical Engineering*, 7(5), 584-592. <http://dx.doi.org/10.1016/j.jrmge.2015.06.011>.
- Aoki, N., & Velloso, D. (1975). An approximate method to estimate the bearing capacity of piles. In *Proc., 5th Pan-American Conf. of Soil Mechanics and Foundation Engineering* (pp. 367-376) Buenos Aires: International Society of Soil Mechanics and Geotechnical Engineering.
- Ardalan, H., Eslami, A., & Nariman-Zadeh, N. (2009). Piles shaft capacity from CPT and CPTu data by polynomial neural networks and genetic algorithms. *Computers and Geotechnics*, 36(4), 616-625. <http://dx.doi.org/10.1016/j.compgeo.2008.09.003>.
- Arel, E. (2012). Predicting the spatial distribution of soil profile in Adapazari/Turkey by artificial neural networks using CPT data. *Computers & Geosciences*, 43, 90-100. <http://dx.doi.org/10.1016/j.cageo.2012.01.021>.
- Associação Brasileira de Normas Técnicas – ABNT (2019). *Projeto e execução de fundações*. Associação Brasileira de Normas Técnicas, Rio de Janeiro.
- Benali, A., Nechnech, A., Boukhatem, B., Hussein, M.N., & Karry, M. (2018). Neural networks and principle component analysis approaches to predict pile capacity in sand. In MATEC Web of Conference (Org.), *2nd International Congress on Materials & Structural Stability*. Les Ulis: EDP Sciences. <https://doi.org/10.1051/mateconf/201814902025>
- Bhattacharya, B., & Solomatine, D.P. (2006). Machine learning in soil classification. *Neural Networks*, 19(2), 186-195. <http://dx.doi.org/10.1016/j.neunet.2006.01.005>.
- Bigarella, J.J., & Salamuni, R. (1962). *Caracteres texturais dos sedimentos da bacia de Curitiba*. Instituto de Geologia.
- Bonini Neto, A., Bonini, C.S.B., Bisi, B.S., Reis, A.R., & Coletta, L.F.S. (2017). Artificial neural network for classification and analysis of degraded soils. *IEEE Latin America Transactions*, 15(3), 503-509.
- Bui, D.T., Moayed, H., Gör, M., Jaafari, A., & Foong, L.K. (2019). Predicting slope stability failure through machine learning paradigms. *ISPRS International Journal of Geo-Information*, 8(9), 395. <http://dx.doi.org/10.3390/ijgi8090395>.
- Carvalho, L.O., & Ribeiro, D.B. (2019). Soil classification system from cone penetration test data applying distance-based machine learning algorithms. *Soils and Rocks*, 42(2), 167-178. <http://dx.doi.org/10.28927/SR.422167>.
- Décourt, L., & Quaresma, A. (1978). Capacidade de carga de estacas a partir de valores de SPT. In *Congresso Brasileiro de Mecânica dos Solos e Engenharia de Fundações* (pp. 45-53). São Paulo: Associação Brasileira de Mecânica dos Solos e Engenharia Geotécnica.
- Décourt, L., & Quaresma, A. (1998). Análise e Projeto de Fundações Profundas. In: W. Hachich (Eds.), *Fundações: teoria e prática* (pp. 275-276). PINI.
- Draper, N.R., & Smith, H.K. (1998). *Applied regression analysis* (3rd edition). Wiley.
- Dudani, S.A. (1976). The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(4), 325-327.
- Ferentinou, M.D., & Sakellariou, M.G. (2007). Computational intelligence tools for the prediction of slope performance. *Computers and Geotechnics*, 34(5), 362-384. <http://dx.doi.org/10.1016/j.compgeo.2007.06.004>.
- Goh, A.T.C., & Goh, S.H. (2007). Support vector machines: their use in geotechnical engineering as illustrated using seismic liquefaction data. *Computers and Geotechnics*, 34(5), 410-421. <http://dx.doi.org/10.1016/j.compgeo.2007.06.001>.
- Hanna, A.M., Ural, D., & Saygili, G. (2007). Neural network model for liquefaction potential in soil deposits using Turkey and Taiwan earthquake data. *Soil Dynamics and Earthquake Engineering*, 27(6), 521-540. <http://dx.doi.org/10.1016/j.soildyn.2006.11.001>.
- Ho, T.K. (1995). Random decision forests. In International Association for Pattern Recognition. Technical Committee 11 (Org.), *Proceedings of 3rd International Conference on Document Analysis and Recognition* (pp. 278-282). Los Alamitos: IEEE Computer Society Press.
- Ismail, A., & Jeng, D. (2011). Modelling load-settlement behavior of piles using high-order neural network (HON-PILE model). *Engineering Applications of Artificial Intelligence*, 24(5), 813-821. <http://dx.doi.org/10.1016/j.engappai.2011.02.008>.

- Jesswein, M., & Liu, J. (2018). Predicting the side resistance of piles using a genetic algorithm and SPT n-values. In Canadian Geotechnical Society (Org.), *Proceedings of the 71st Canadian Geotechnical Conference and the 13th Joint CGS/IAH-CNC Groundwater Conference - GeoEdmonton 2018 : moving forward* (pp. 1-8). Richmond: Canadian Geotechnical Society.
- Juang, C.H., & Chen, C.J. (1999). CPT-based liquefaction evaluation using artificial neural networks. *Computer-Aided Civil and Infrastructure Engineering*, 14(3), 221-229. <http://dx.doi.org/10.1111/0885-9507.00143>.
- Kardani, N., Zhou, A., Nazem, M., & Shen, S. (2020). Estimation of bearing capacity of piles in cohesionless soil using optimised machine learning approaches. *Geotechnical and Geological Engineering*, 38(2), 2271-2291. <http://dx.doi.org/10.1007/s10706-019-01085-8>.
- Kohestani, V.R., Hassanlourad, M., & Ardakani, A. (2015). Evaluation of liquefaction potential based on CPT data using random forest. *Natural Hazards*, 79(2), 1079-1089. <http://dx.doi.org/10.1007/s11069-015-1893-5>.
- Kordjazi, A., Nejad, F.P., & Jaksa, M.B. (2013). Prediction of load-carrying capacity of piles using a support vector machine and improved data collection. In G. Cassidy (Ed.), *Proceedings of the 12th Australia New Zealand conference on geomechanics: the changing face of the earth - geomechanics & human influence* (pp. 1-8). Wellington: Opus.
- Kordjazi, A., Nejad, F.P., & Jaksa, M.B. (2014). Prediction of ultimate axial load-carrying capacity of piles using a support vector machine based on CPT data. *Computers and Geotechnics*, 55, 91-102. <http://dx.doi.org/10.1016/j.compgeo.2013.08.001>.
- Kovacevic, M., Bajat, B., & Gajic, B. (2010). Soil type classification and estimation of soil properties using support vector machines. *Geoderma*, 154(3-4), 340-347. <http://dx.doi.org/10.1016/j.geoderma.2009.11.005>.
- Kuo, B., Yang, J., Sheu, T., & Yang, S. (2008). Kernel-based KNN and gaussian classifiers for hyperspectral image classification. In Geoscience and Remote Sensing Society (Org.), *2008 International Geoscience and Remote Sensing Symposium* (pp. 1006-1008). Piscataway: IEEE.
- Lee, I., & Lee, J. (1996). Prediction of pile bearing capacity using artificial neural networks. *Computers and Geotechnics*, 18(3), 189-200. [http://dx.doi.org/10.1016/0266-352X\(95\)00027-8](http://dx.doi.org/10.1016/0266-352X(95)00027-8).
- Livingston, G., Piantedosi, M., Kurup, P., & Sitharam, T.G. (2008). An approximate method to estimate the bearing capacity of piles. In X. Zeng, M.T. Manzari, & D.R. Hiltunen (Eds.), *Geotechnical Earthquake Engineering and Soil Dynamics IV Congress* (pp. 1-10). Reston: American Society of Civil Engineers.
- Lobo, B. (2005). *Método de previsão de capacidade de carga de estacas: aplicação dos conceitos de energia do ensaio SPT* [Doctoral thesis]. Universidade Federal do Rio Grande do Sul.
- Lok, T.M.H., & Che, W.F. (2004). Axial capacity prediction for driven piles using ANN: model comparison. In M.K. Yegian, & E. Kavazanjian (Eds.), *Geotechnical Engineering for Transportation Projects* (pp. 697-704). Reston: American Society of Civil Engineers. [https://doi.org/10.1061/40744\(154\)56](https://doi.org/10.1061/40744(154)56).
- Maizir, H., & Kassim, K.A. (2013). Neural network application in prediction of axial bearing capacity of driven piles. In O. Castillo, C. Douglas, D.D. Feng & J. Lee (Eds.), *Proc of the International MultiConference of Engineers and Computer Scientists*. Hong Kong: Newswood Limited.
- Maxwell, A.E., Sharma, M., Kite, J.S., Donaldson, K.A., Thompson, J.A., Bell, M.L., & Maynard, S.M. (2020). Slope failure prediction using random forest machine learning and lidar in an eroded folded mountain belt. *Remote Sensing*, 12(3), 486. <http://dx.doi.org/10.3390/rs12030486>.
- Meyerhof, G. (1976). Bearing capacity and settlements of pile foundations. *Journal of Geotechnical Engineering*, 102, 197-228. <http://dx.doi.org/10.1061/AJGEB6.0000243>.
- Moayedi, H., & Hayati, S. (2019). Artificial intelligence design charts for predicting friction capacity of driven pile in clay. *Neural Computing & Applications*, 31(11), 7429-7445. <http://dx.doi.org/10.1007/s00521-018-3555-5>.
- Momeni, E., Nazir, R., Armaghani, D., & Maizir, H. (2015). Application of artificial neural network for predicting shaft and tip resistances of concrete piles. *Earth Sciences Research Journal*, 19(1), 85-93.
- Morais, T.S.O., Tsuha, C.H.C., Bandeira Neto, L.A., & Singh, R.M. (2020). Effects of seasonal variations on the thermal response of energy piles in an unsaturated Brazilian tropical soil. *Energy and Building*, 216, 1-19. <http://dx.doi.org/10.1016/j.enbuild.2020.109971>.
- Nawari, N.O., & Liang, R. (2000). Intelligent hybrid system for the design of pile foundations. In N.D. Dennis, R. Castelli, & M.W. O'Neill (Eds.), *New Technological and Design Developments in Deep Foundations* (pp. 312-326). Reston: American Society of Civil Engineers. [https://doi.org/10.1061/40511\(288\)22](https://doi.org/10.1061/40511(288)22).
- Nejad, F.P., & Jaksa, M.B. (2017). Load-settlement behavior modeling of single piles using artificial neural networks and CPT data. *Computers and Geotechnics*, 89, 9-21. <http://dx.doi.org/10.1016/j.compgeo.2017.04.003>.
- Nejad, F.P., Kakhki, M.B., & McCabe, B.A. (2009). Prediction of pile settlement using artificial neural networks based on standard penetration test data. *Computers and Geotechnics*, 36(7), 1125-1133. <http://dx.doi.org/10.1016/j.compgeo.2009.04.003>.
- Pal, M., & Deswal, S. (2008). Modeling pile capacity using support vector machines and generalized regressions neural network. *Journal of Geotechnical and Geoenvironmental Engineering*, 134(7), 1021-1024. [http://dx.doi.org/10.1061/\(ASCE\)1090-0241\(2008\)134:7\(1021\)](http://dx.doi.org/10.1061/(ASCE)1090-0241(2008)134:7(1021)).
- Pal, M., & Deswal, S. (2010). Modelling pile capacity using Gaussian process regression. *Computers and*

- Geotechnics*, 37(7), 924-947. <http://dx.doi.org/10.1016/j.compgeo.2010.07.012>.
- Pham, T., Ly, H., Tran, V., Giap, L., Vu, H., & Duong, H. (2020). Prediction of pile axial bearing capacity using artificial neural network and random forest. *Applied Sciences*, 10(5), 1871. <http://dx.doi.org/10.3390/app10051871>.
- Salgado, R. (2008). *The engineering of foundations*. McGraw Hill.
- Samui, P. (2008). Support vector machine applied to settlement of shallow foundations on cohesionless soils. *Computers and Geotechnics*, 35(3), 419-427. <http://dx.doi.org/10.1016/j.compgeo.2007.06.014>.
- Santos Junior, O. (1988). *Previsão do comportamento carga-recalque de estacas pré-moldadas de concreto* [Masters thesis]. Universidade de São Paulo.
- Shahin, M.A. (2010). Intelligent computing for modeling axial capacity of pile foundations. *Canadian Geotechnical Journal*, 47(2), 230-243. <http://dx.doi.org/10.1139/T09-094>.
- Smola, A.J., & Scholkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3), 199-222. <http://dx.doi.org/10.1023/B:STCO.0000035301.49549.88>.
- Van Der Veen, C. (1953). The bearing capacity of a pile. In International Conference on Soil Mechanics and Foundation Engineering (Org.), *International Conference on Soil Mechanics and Foundation Engineering* (pp. 367-376). Zurich: ICOSOMEF.
- Vapnik, V.N. (1998). *Statistical learning theory*. Wiley.
- Vapnik, V.N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5), 988-999.
- Vianna, A. (2000). *Análise de provas de carga estática em estacas pré-moldadas cravadas na cidade de Curitiba e região metropolitana* [Masters thesis]. Universidade de São Paulo. <https://doi.org/10.11606/D.18.2000.tde-18102006-101542>
- Wong, T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-outcross validation. *Pattern Recognition*, 48(9), 2839-2846. <http://dx.doi.org/10.1016/j.patcog.2015.03.009>.
- Yu, K., Ji, L., & Zhang, X. (2002). Kernel nearest-neighbor algorithm. *Neural Processing Letters*, 15(2), 147-156. <http://dx.doi.org/10.1023/A:1015244902967>.