Machine Learning Approach for Approximating Design Parameters from Engineering Graphs

# Abstract

Geotechnical engineers have traditionally relied on engineering graphs for the analysis and design of specific geotechnical problems. However, interpolating target design parameters, particularly on logarithmic scale graphs, can be time consuming and susceptible to human error. Recent advancements in machine learning enable engineers to efficiently approximate design parameters by training models on extensive datasets, thereby minimizing both time and manual intervention. Furthermore, coefficients for closed-form equations can be derived from these models, streamlining computational analysis and enhancing design workflows. This paper presents two case studies: one focused on shallow footing settlement assessment and the other on single pile settlement assessment. It illustrates the application of non-linear regression, high-degree polynomial regression, Gaussian Process Regression, and Fully Connected Neural Networks in developing effective machine learning models for graphical approximation.

# Introduction

For many years, engineers have depended on engineering graphs for assessments and designs related to geotechnical problems. These practices often originate from academic research or field observations, leading to the development of specialized graphs that provide valuable guidelines for practitioners, significantly simplifying engineering processes. Even with the development of powerful numerical methods such as the finite element method, chart-based methods remain important for checking the results of complex numerical analyses.

Nowadays, it remains common for engineers to extract data points from graphs in prominent publications and convert them into tabular formats, facilitating linear interpolation of intermediate target parameters. However, this method is time-consuming and prone to inaccuracies due to human error, particularly with data presented on logarithmic scales.

As artificial intelligence (AI) gains popularity, its application in geotechnical engineering is becoming increasingly prevalent. A fundamental application involves fitting regression models to obtain optimal lines or surfaces from engineering graphs. This regression can be executed using machine learning (ML) techniques where data is input into an ML algorithm, and the model is trained to minimize a loss function. This paper demonstrates the digitization and approximation of engineering graphs through machine learning algorithms, illustrated by two case studies that showcase enhanced efficiency in geotechnical analysis.

# Case 1: Settlement of a shallow footing

To assess the settlement of a flexible circular footing, Mayne and Poulos (1999) recommend the following equation, which accounts for homogeneous to Gibson soil modulus profiles, finite layer thickness, foundation flexibility, undrained and drained loading conditions, and embedment:

where q is the average applied loading, d the equivalent footing diameter, IG the displacement influence factor, IF the foundation flexibility correction factor, IE the foundation embedment correction factor, v the soil Poisson’s ratio and Eo is the soil Young’s modulus at the surface.

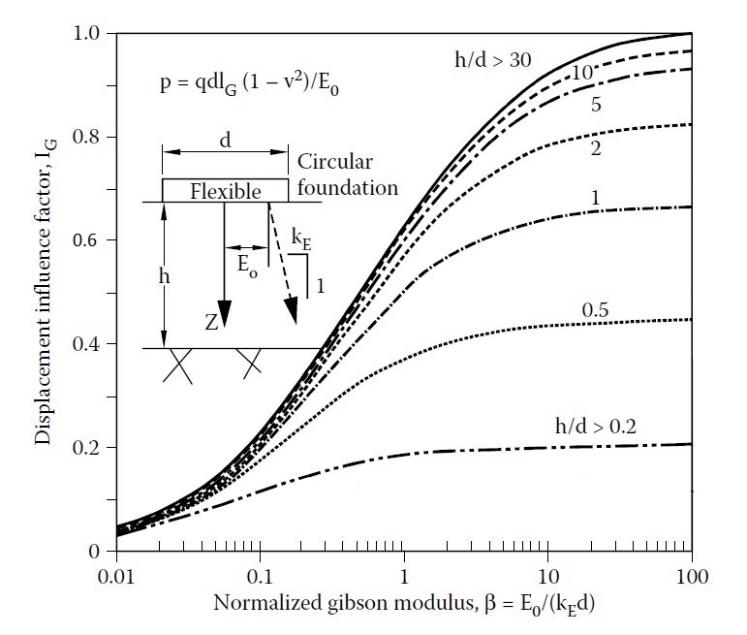
The foundation flexibility correction factor IF is approximately given by:

where , Ef is the footing Young’s modulus, Esav the average soil Young’s modulus and t is the footing thickness.

The foundation embedment correction factor IE is approximately given by

The graph for the displacement influence factor IG is extracted from Mayne and Poulos (1999) and is illustrated in Figure 1.

Figure 1 - Displacement influence factor IG. (Adapted from Mayne and Poulos 1999)



To automate the assessment process, it is necessary to establish a closed-form equation for IG that approximates all available curve lines in Figure 1. The curves exhibit a characteristic S-shape and can be modelled as a modified sigmoid function. The parameter β is plotted on a logarithmic scale in Figure 1, and the expected form of the equation is:

which is simplified as:

where

For a specific value of t, there exists a corresponding modified sigmoid function characterized by the constants mt, nt, and kt. To determine these constants, data points from each curve are extracted using an online tool, WebPlotDigitizer, which employs multimodal machine learning methods and computer vision algorithms. These data points are then analyzed using a selected non-linear regression algorithm implemented in the open-source Python modules ‘sklearn’ and ‘scipy’. The resulting values of mt, nt, and kt are summarized with the corresponding R2 scores and mean squared errors (MSE) in Table 1.

Table 1 - Values of h/d with the associated constants.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| h/d | t = ln (h/d) | mt | nt | kt | R2 score | MSE |
| 0.2 | 3.401 | 0.980 | 0.604 | 0.753 |  |  |
| 0.5 | 2.303 | 1.018 | 0.601 | 0.779 |  |  |
| 1 | 1.609 | 1.059 | 0.595 | 0.787 |  |  |
| 2 | 0.693 | 1.201 | 0.541 | 0.831 |  |  |
| 5 | 0.000 | 1.493 | 0.490 | 0.869 |  |  |
| 10 | -0.693 | 2.240 | 0.446 | 0.892 |  |  |
| 30 | -1.609 | 4.898 | 0.405 | 0.945 |  |  |

The next step involves formulating the values presented in Table 1 into a generalized function. To achieve this, each constant (mt, nt, and kt) is treated as a dependent variable correlated with the independent variable t through a closed-form equation. The rationale for associating m, n, and k with ln(h/d) instead of h/d is based on improved fitting performance.

Several trials of polynomial regression from first degree to third degree have been performed to search for the best fit functions based on R2 scores and mean squared errors. The results of the regression exercise are expressed in polynomials with the corresponding R2 scores and mean squared errors (MSE) in Table 2.

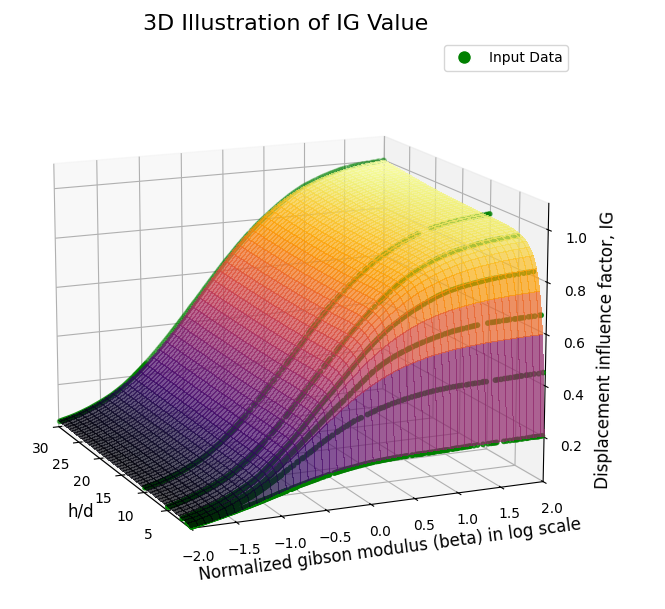
Table 2 - xxxx.

|  |  |  |
| --- | --- | --- |
| Polynomial expression | R2 score | MSE |
|  |  |  |
|  |  |  |
|  |  |  |

Consequently, the displacement influence factor IG can be expressed in a generalized form as follows:--- Eqn (1)

Finally, the overall regression is performed on Equation (x). The calculated R2 score is 0.9x and all MSE are below 0.0x. This implies that Equation (x) is representative of the input data. It can also be verified by visual inspection on Figure 1 where the green dots (input data) lie perfectly on the curved surface generated from Equations (1) (predictions). .. It is noteworthy that the inter-line prediction is smooth compared with acute changes due to linear interpolation.

Figure 2 - Three-Dimensional Illustration of IG Value.

****

# Case 2: Settlement of a single end-bearing pile

The approach used in Case 1 necessitates a certain level of mathematical insight, which enables one to make an initial guess regarding the form of the equation. However, this ability is often a privilege of mathematicians and can sometimes rely on luck. In contrast, Case 2 illustrates an approximation exercise where a closed-form equation is not strictly required.

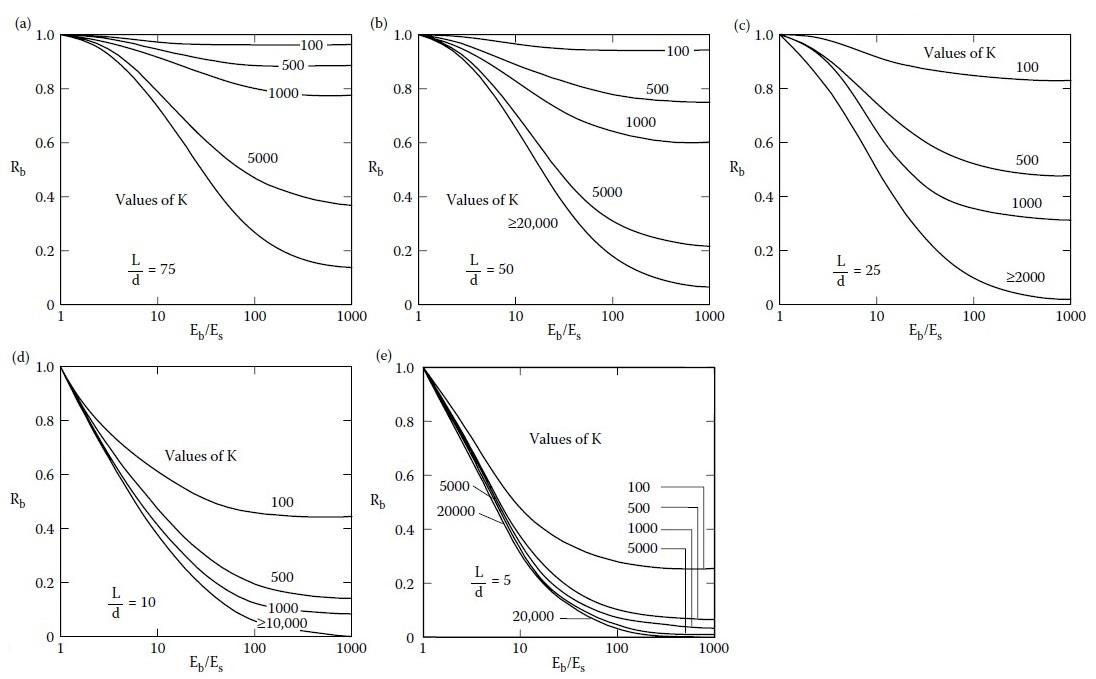
To assess the settlement of a single end-bearing pile, Poulos and Davis (1980) recommend the following equation:

where P is the applied load, d the pile diameter, Es the average soil modulus along the pile shaft, I1 the influence factor for a rigid pile in a semi-infinite mass and Rk, Rb, Rv are the correction factors for the effect of pile compressibility, bearing stratum stiffness and Poisson’s ratio respectively.

While this assessment involves four factors, only the factor Rb will be utilized to demonstrate the approximation exercise. The other three factors can be approximated either in the same way as IG in Case 1 or Rb in this case. The graphs for Rb are extracted from Poulos and Davis (1980) and are presented in Figure 3. The value of Rb depends on three independent variables: the ratio of Eb/Es, the ratio of L/d, and the K value. Here, Eb represents the Young’s modulus of the founding stratum, and L is the pile length. The variable K is defined as:

where Ep is the Young’s modulus of pile and RA is the area ratio of pile

Figure 3 - Bearing stratum correction factor Rb for values of L/d of 75, 50, 25, 10 and 5. (Adapted from Poulos and Davis, 1980)



Identifying the complex relationship between the target Rb and the other three variables through a closed-form equation is impractical. An alternative approach is to employ supervised machine learning algorithms directly, without the necessity of fully understanding the underlying data patterns. Two common methods, Gaussian Process Regression (GPR) and Fully Connected Neural Networks (FCNN) implemented in the open-source Python modules ‘sklearn’ and ‘keras’ are utilized to train machine learning models in this study.

To initiate this process, data points are extracted from Figure 3 using WebPlotDigitizer. The data are trained using ML algorithms to form a digital model. This digital model is then used for prediction for any unseen new inputs. The underlying functions linking all variables exist in a form of hidden mathematical relationships inside the digital model instead of being expressed by a clear formula. These hidden mathematical relationships will be discussed in later paragraphs.

## Gaussian Process Regression

The performance of Gaussian Process Regression (GPR) is sensitive to the absence of some data in Figures 3(c) and 3(d), where no data is available for K values beyond 2,000 and 10,000 respectively. This lack of data complicates predictions beyond these bounds and may lead to overfitting of the model to the input data. To address this issue, supplementary data points are added for K values being 5,000 and 20,000, as illustrated in Figures 4(c) and 4(d).

Another important factor influencing regression performance is the choice of kernel (covariance function) and its hyperparameters. Selecting an appropriate kernel requires knowledge of data science, which is beyond the scope of this paper. In this study, a grid search approach is employed to identify the best-performing kernel among the Radial Basis Function (RBF), Matern and Rational Quadratic kernels. The combination of hyperparameters for each kernel that yields the lowest mean squared error (MSE) is selected as the optimal estimator.

Before analyzing the four-dimensional relationship, three-dimensional trials are conducted for each value of the L/d ratio separately, allowing for a review of preliminary results with the optimal kernel. The settings for the optimal kernel are as follows:

Kernel: Rational Quadratic

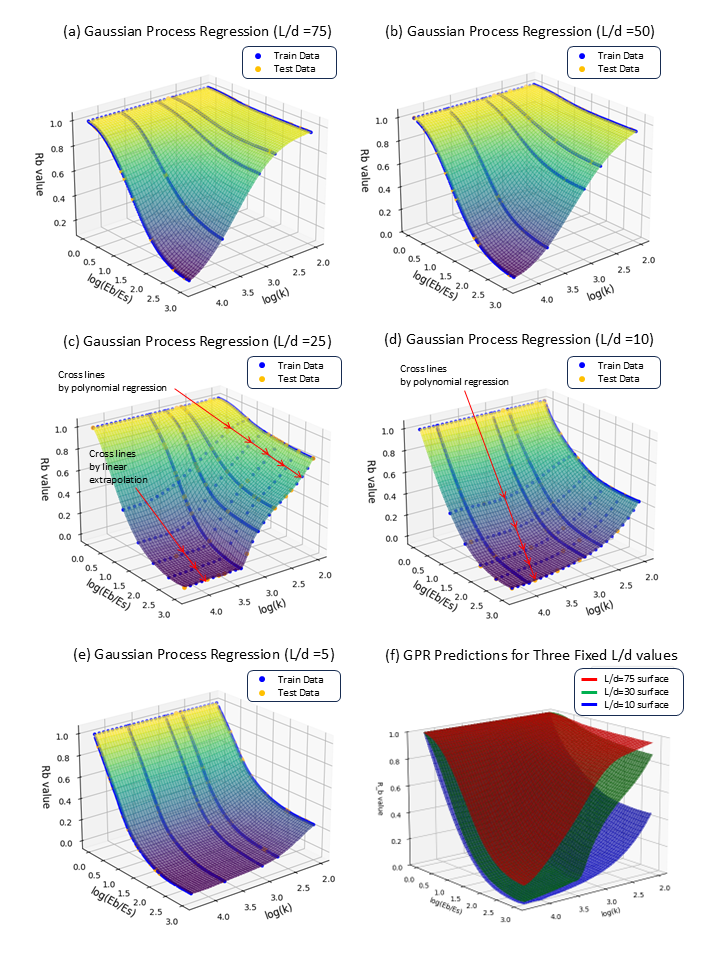
Scale mixture parameter α: 0.1

Length scale parameter: 0.5

Three-dimensional plots illustrating the data relationships for five different L/d ratios are presented in Figures 4(a) through 4(e). Each L/d layer represents predictions for 400 x 400 grid points along two axes. The maximum MSE and mean absolute error (MAE) from these trials are approximately 0 and 0.0018 respectively, indicating that the GPR model fits the training data well in three-dimensional space. From visual inspection, there is a smooth transition for the predictions in the unseen intermediate areas bounded by discrete blue lines.

The next step is to construct a comprehensive four-dimensional model incorporating all four variables simultaneously, using the optimal kernel hyperparameters identified in the three-dimensional trials. The respective MSE and MAE for the four-dimensional model are approximately 0 and 0.0006. To visualise the trained four-dimensional model, four combined three-dimensional plots for various trained L/d layers and unseen intermediate layers (in green colour) are presented in Figures 5(a) through 5(d). It is shown that the transitional predictions from one trained layer to another trained layer is very smooth.

Figure 4 - Approximation of Rb using Gaussian Process Regression.



## Fully Connected Neural Network

A Fully Connected Neural Network (FCNN), also known as a Dense Neural Network, is a type of artificial neural network in which each neuron in one layer is connected to every neuron in the subsequent layer. According to the Universal Approximation Theorem, a feedforward neural network with at least one hidden layer containing a finite number of neurons can approximate any continuous function on a compact subset of Rn to any desired degree of accuracy, provided that a suitable non-linear activation function is employed. Here, Rn denotes an n-dimensional Euclidean space, and the function in question pertains to a bounded and continuous subset of four-dimensional space.

In this study, five hidden layers are configured with the following number of neurons: 128, 64, 64, 32, and 8 respectively. The Rectified Linear Unit (ReLU) is selected as the activation function for each hidden layer, while a linear activation function is assigned to the final output layer which consists of a single neuron. The other parameters for the training configuration are as follows

Optimizer: Adam

Epochs: 100

Batch size: 32

Validation split: 0.1

The MSE and MAE for the 10% testing data are approximately 0 and 0.0045 respectively, indicating strong predictive performance. Figure 5 presents three-dimensional plots for various L/d ratios, providing a visual representation of the trained model. Notably, unlike GPR, cross line data is not required for the FCNN, as overfitting is not a concern with this model.

## Comparison of model performance

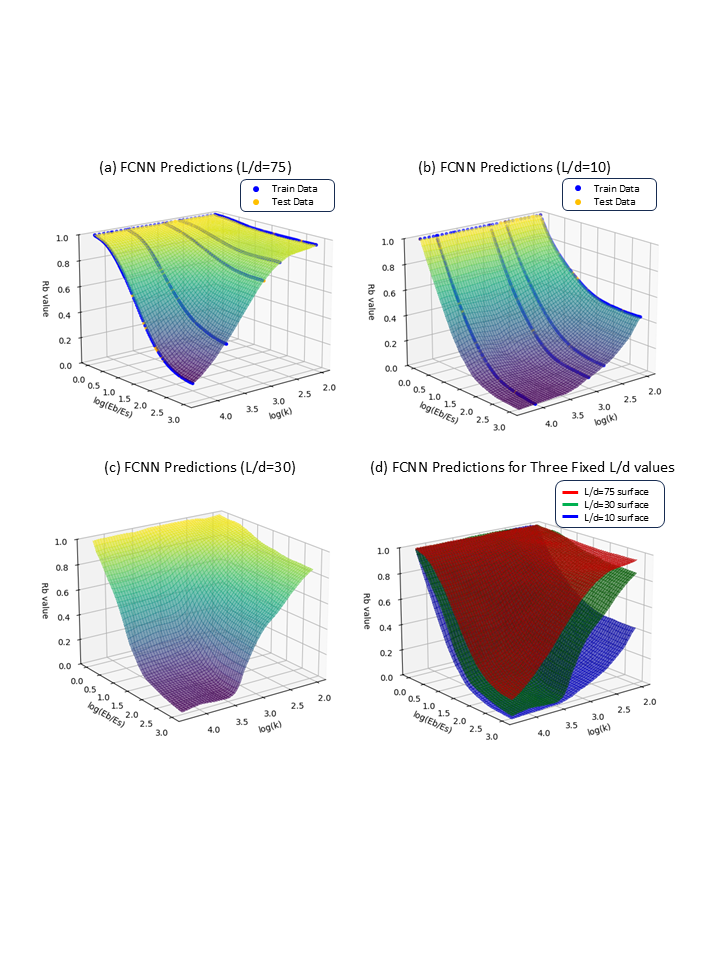
To evaluate the performance of the two models, ten points from different clusters are randomly selected for assessment. The predictions generated by these models are compared with those obtained through a manual interpolation method performed by the author using visual estimation and a ruler. The comparison results are presented in Table 2.

While there are slight deviations in the predictions, particularly for intermediate values of L/d ratios and K values, the order of magnitude for each Rb value remains generally consistent across the different methods.

Table 2 - Comparison of Rb using different methods.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| L/d | K | Eb/Es | Log(Eb/Es) | Rb | | | Max. Difference |
| GPR | FCNN | Manual |
| 75 | 2000 | 100 | 2.0 | 0.65 | 0.66 | 0.70 | 0.05 |
| 60 | 800 | 80 | 1.9 | 0.77 | 0.76 | 0.78 | 0.02 |
| 60 | 2000 | 50 | 1.7 | 0.62 | 0.61 | 0.69 | 0.07 |
| 50 | 300 | 500 | 2.7 | 0.82 | 0.81 | 0.85 | 0.04 |
| 50 | 2000 | 50 | 1.7 | 0.53 | 0.54 | 0.60 | 0.06 |
| 40 | 300 | 10 | 1.0 | 0.91 | 0.92 | 0.86 | 0.06 |
| 25 | 300 | 10 | 1.0 | 0.83 | 0.82 | 0.80 | 0.03 |
| 25 | 800 | 500 | 2.7 | 0.38 | 0.39 | 0.36 | 0.03 |
| 15 | 300 | 30 | 1.5 | 0.52 | 0.47 | 0.50 | 0.05 |
| 10 | 300 | 30 | 1.5 | 0.36 | 0.35 | 0.38 | 0.03 |

Figure 5 - Approximation of Rb using Fully Connected Neural Network.



# Discussion

Case 1 illustrates the procedures for applying basic machine learning techniques, specifically non-linear regression and polynomial regression, in graphical approximation. It is important to note that integral solutions for the influence factor IG are indeed available, as detailed in the works of Davis & Poulos (1968) and Mayne & Poulos (1999). While it remains a good practice to perform numerical integration to evaluate settlements precisely using the elastic displacement theory upon which Figure 1 is based, this paper does not aim to replace the original integration method. Instead, it offers an alternative to both the integration method and the traditional graph reading method, establishing a closed-form solution that simplifies computation while preserving accuracy.

Case 2 demonstrates the application of Gaussian Process Regression (GPR) and Fully Connected Neural Networks (FCNN) in understanding the relationships among multiple variables. Although the data exhibits certain patterns, developing a simple closed-form equation linking all variables is impractical. Therefore, GPR and FCNN are employed to predict the dependent variable Rb directly, bypassing the need to ascertain the form of the equation. The final products of these regression exercises are two ML models in form of digital files which can be downloaded via <https://github.com/Opengti/papers/tree/main/TTC_Symposium_2025/models>. A simple instruction of using the digital files and the raw data are also included in the folder. The reader may clone the repository and load the ML models directly for Excel calculation.

The fundamental attribute in GPR model responsible for prediction is the covariance matrix which is computed using the above kernel setting and the training data prepared by the author. This covariance matrix stores the mathematical relationships between the trained data points. If the reader uses the same kernel setting but with different data points, the convenience matrix will be totally different. In other words, every GPR model is a unique model depending on the original trained data, even though the output predictions may be similar.

On the other hand, the hidden mathematical relationships within a FCNN model are represented by the weight and bias of each neuron. The overall behavior of the neuron is influenced by the activation function applied to the weighted sum of its inputs. The combination of weights, biases, and activation functions allows neural networks to learn complex mappings from inputs to outputs. These parameters have been extracted from the trained FCNN model and saved in the above folder. The reader has the flexibility to finetune the trained FCNN model with a smaller task-specific dataset to improve the performance.

The case studies presented herein illustrate straightforward approaches to developing machine learning models for digitizing graphical data, solving non-linear regression problems, and identifying relationships among multiple variables. Compared to traditional graphical interpolation, machine learning offers a practical alternative for geotechnical engineers, enhancing efficiency in their daily workflows. In particular, FCNN provides a simple and effective method for identifying complex data relationships with minimal data handling, as demonstrated in Case 2.

# Conclusion

The rapid advancement of artificial intelligence (AI) technology has the potential to replace tedious manual processes, and the engineering industry is increasingly embracing digitization and automation. This shift is not limited to large software companies, but it is also becoming integral to the daily practices of engineers. As a subset of AI, machine learning can assist engineers in their routine workflows using open-source tools. The case studies presented in this paper demonstrate the integration of machine learning with geotechnical engineering, enhancing the efficiency of daily assessment and design tasks. It is anticipated that more AI applications in geotechnical engineering will emerge as we approach the fourth industrial revolution.

# Reference:

Davis, E.H. and Poulos, H.G. (1968). The use of elastic theory for settlement prediction under three-dimensional conditions. Géotechnique 18, Issue 1, pp. 67-91

Poulos, H.G. and Davis, E.H. (1980). Pile foundation analysis and design. New York: John Wiley

Mayne, P.W. and Poulos, H.G. (1999). Approximate displacement influence factors for elastic shallow foundations. Journal of Geotechnical and Geoenvironmental Engineering, ASCE 125, pp. 453-460