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.

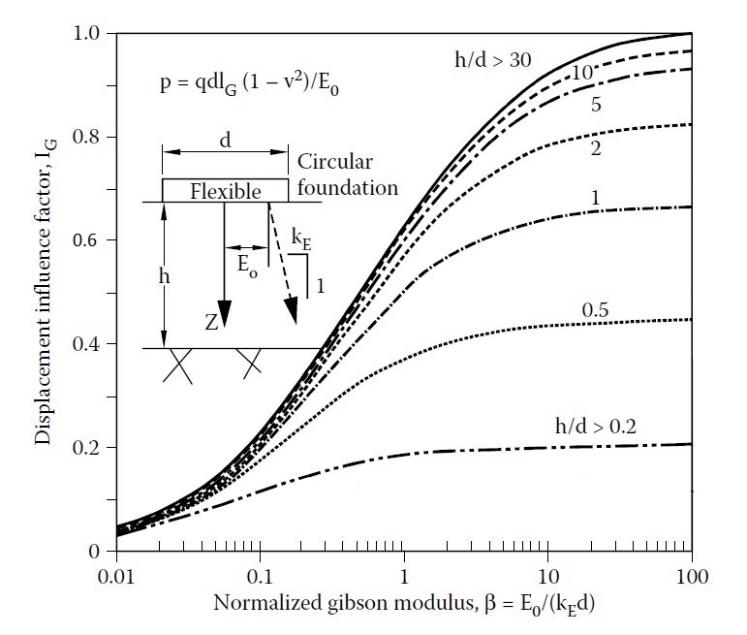
The foundation flexibility correction factor IF is approximately given by:

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 in Table 1.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| h/d | t = ln (h/d) | mt | nt | kt |
| 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 |

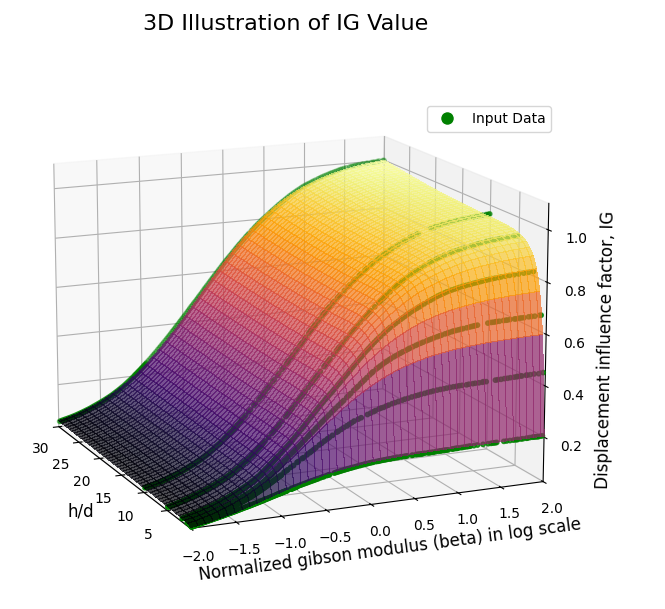
The next step involves formulating the values presented in Table 1 into a generalized function. To achieve this, each constant (mt, nt, and kt) is treated as a dependent variable correlated with the independent variable t through a closed-form equation. The rationale for associating m, n, and k with 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 the following polynomials.

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

Finally, all data are input into Equation (1) to assess overall accuracy. The resulting R² score of 0.9996 and mean squared error of 3.08e-5 indicate an excellent fit of the input data to the constructed equation. This is further corroborated by Figure 2, where the input data points (represented as green dots) align perfectly with the curved surface generated by Equation (1). Moreover, the smooth transitions observed between the known data points suggest that the equation is capable of delivering reliable and continuous predictions for unseen inputs.

Figure 2 - Three-Dimensional Illustration of IG Value.

****

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

The approach used in Case 1 requires 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.

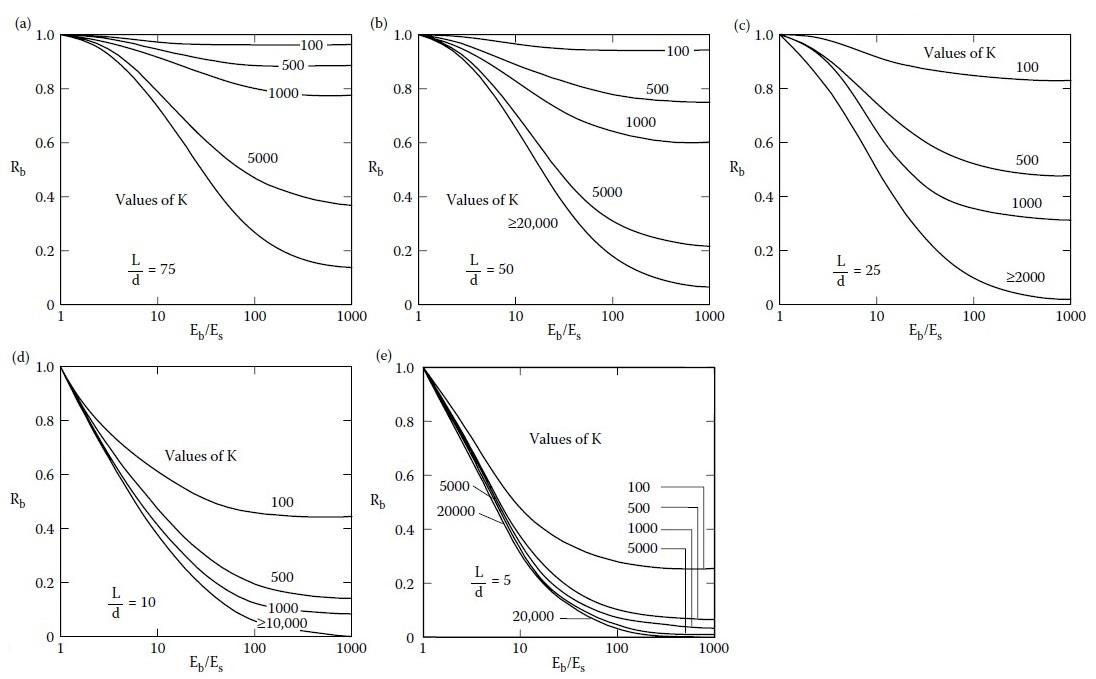
To assess the settlement of a single end-bearing pile, Poulos and Davis (1980) recommend the following equation:

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 determined in the same way as IG Case 1. 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 appears 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 extracted data is then used to train the two selected machine learning algorithms, resulting in the development of two digital models that can identify the underlying relationships among all input variables. Rather than producing a simple equation, the digital models encapsulate these relationships within specific internal mathematical structures, which will be discussed further in the following paragraphs. Once constructed, these digital models can be saved for future predictions.

## 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 the model to the input data. To mitigate this issue, supplementary data points are added by extending the original dataset, as illustrated in Figures 4(c) and 4(d). Furthermore, for the case where L/d equals 25, cross lines are generated using polynomial regression at K-Rb planes for log(Eb/Es) values of 1, 1.5, 2, 2.5 and 3 as shown in Figure 4(c).

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: 1

Length scale parameter: 1

Three-dimensional plots illustrating the data relationships for five different L/d ratios are presented in Figure 4. Each L/d layer represents predictions for 400 x 400 grid points along two axes. The maximum MSE and mean absolute error (MAE) of these trials are 1.25e-6 and 0.0006 respectively, indicating that the GPR model fits the training data well.

The next step involves constructing a comprehensive four-dimensional model that incorporates all four variables simultaneously, using the optimal kernel hyperparameters identified in the three-dimensional trials. The respective MSE and MAE for 10% testing data are 1.54e-6 and 0.0005. To visualise the trained four-dimensional model, a combined three-dimensional plot for various L/d values is presented in Figure 5. It is noteworthy that the layers corresponding to L/d values of 60, 35 and 20 represent completely unseen data not included in the training dataset. The smooth transitions in the inter-layer predictions confirm the model's capability to assess unseen data effectively.

Figure 4 - Approximation of Rb for various L/d values using Gaussian Process Regression.

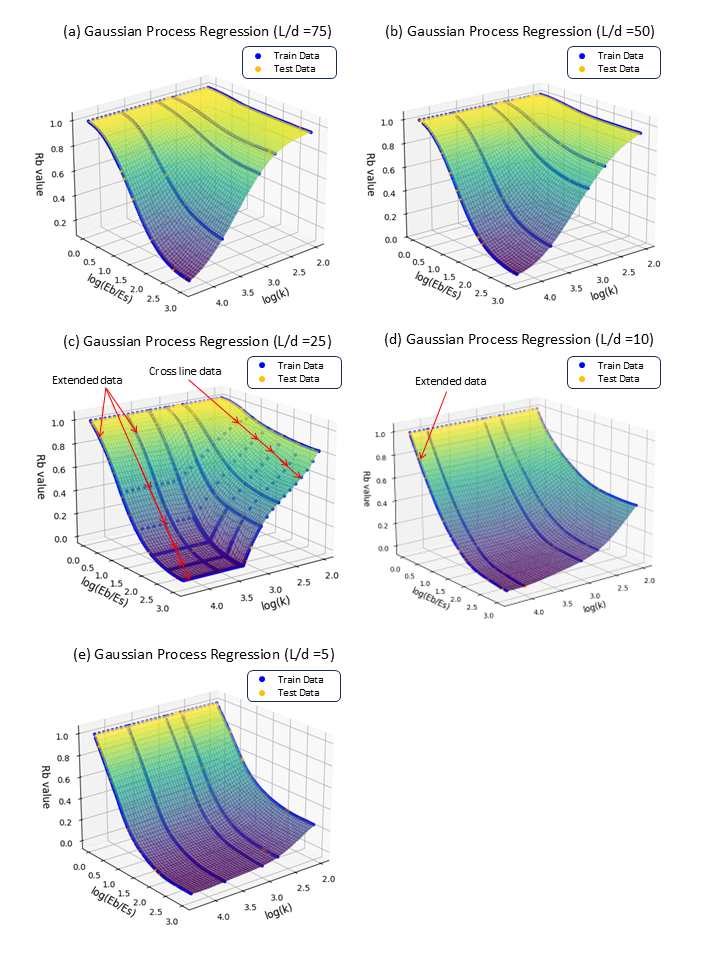
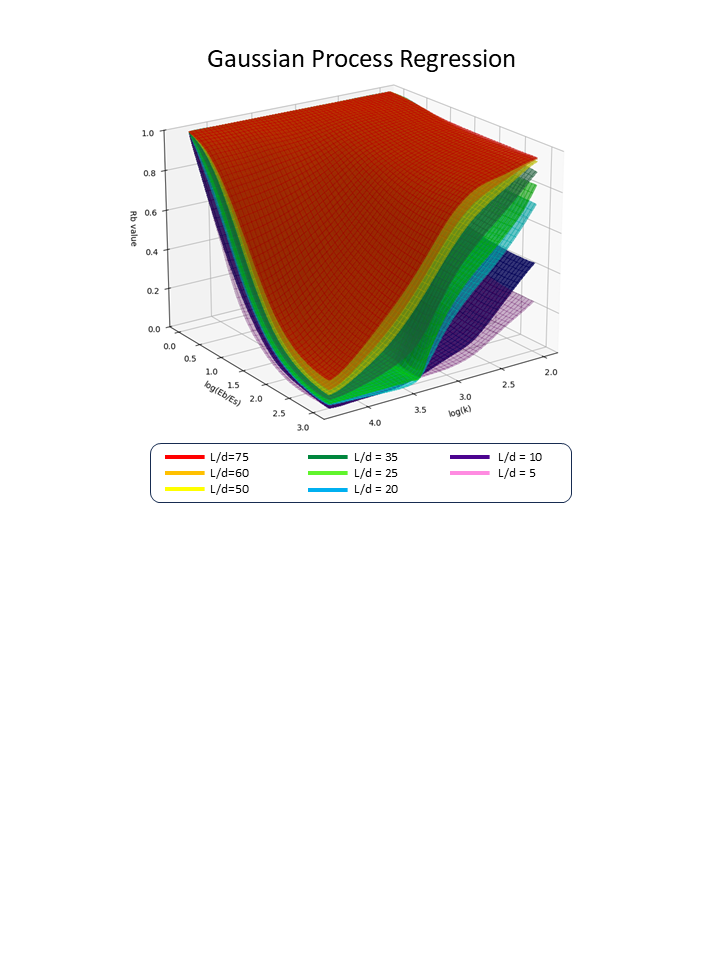


Figure 5 – Combined three-dimensional plot of Rb for various L/d values using Gaussian Process Regression.



## 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 10% testing data are approximately 0 and 0.0045 respectively, indicating strong predictive performance. Figure 6 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

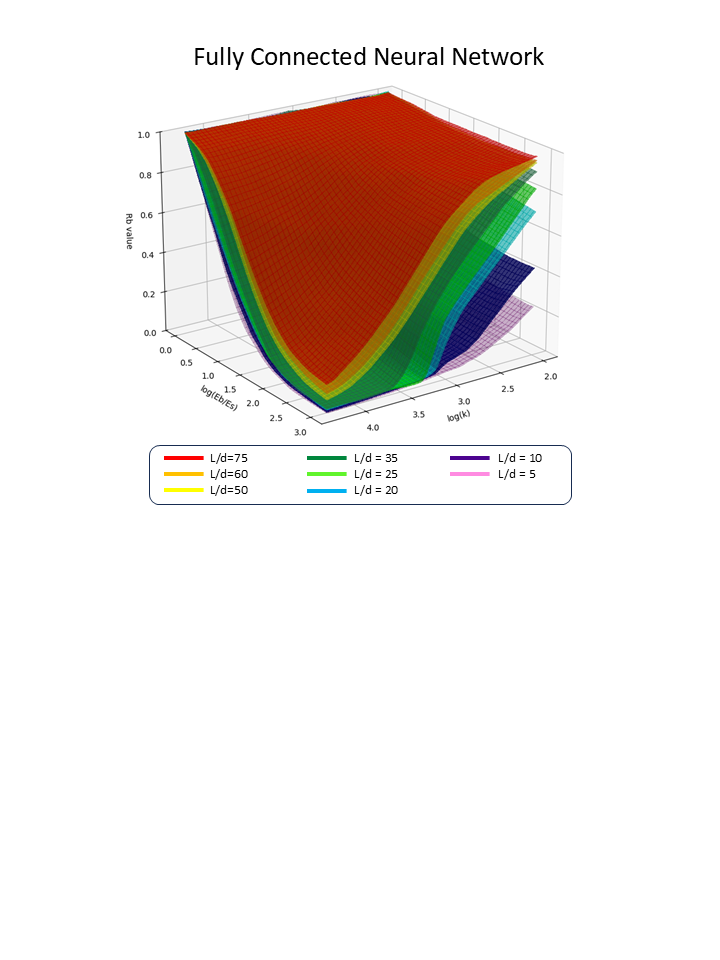
To evaluate the performance of the two constructed digital models, ten points from different clusters are randomly selected for assessment. The predictions generated by these models using a Python interpreter are compared with those obtained through a manual interpolation method performed by the author 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 | 0.68 | 0.65 | 0.03 |
| 60 | 800 | 80 | 1.9 | 0.77 | 0.76 | 0.78 | 0.02 |
| 60 | 2000 | 50 | 1.7 | 0.62 | 0.61 | 0.63 | 0.02 |
| 50 | 300 | 500 | 2.7 | 0.83 | 0.83 | 0.85 | 0.02 |
| 50 | 2000 | 50 | 1.7 | 0.54 | 0.52 | 0.55 | 0.03 |
| 40 | 300 | 10 | 1.0 | 0.90 | 0.89 | 0.86 | 0.04 |
| 25 | 300 | 10 | 1.0 | 0.83 | 0.80 | 0.80 | 0.03 |
| 25 | 800 | 500 | 2.7 | 0.39 | 0.37 | 0.36 | 0.03 |
| 15 | 300 | 30 | 1.5 | 0.52 | 0.47 | 0.50 | 0.03 |
| 10 | 300 | 30 | 1.5 | 0.37 | 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 illustrates the application of Gaussian Process Regression (GPR) and Fully Connected Neural Networks (FCNN) to explore the relationships among multiple variables. While the data reveals certain patterns, deriving a simple closed-form equation that links all variables seems impractical. Consequently, GPR and FCNN are utilized to directly predict the dependent variable Rb, eliminating the need to determine the equation's form.

The outcome of these regression analyses consists of two machine learning models available as digital files for download from GitHub at https://github.com/Opengti/papers/tree/main/ML\_graph\_approximation. The online folder also contains the raw data and simple instructions for using the digital models. Readers are encouraged to clone the entire repository and load the machine learning models directly for predictions using the Python code provided in the repository.

A key feature of the GPR model that facilitates prediction is the covariance matrix, which is computed using the specified kernel settings and the training data prepared by the author. This covariance matrix encapsulates the mathematical relationships between the trained data points. If readers apply the same kernel settings with different data points, the resulting covariance matrix will differ significantly. In essence, each GPR model is unique to its original training data, even if the output predictions may be similar.

In contrast, the hidden mathematical relationships within the FCNN model are defined by the weights and biases of each neuron in the network. The behaviour of each neuron is influenced by the activation function applied to the weighted sum of its inputs. This combination of weights, biases and activation functions enables neural networks to learn complex mappings from inputs to outputs. However, it is important to note that the author's model may not be the optimal one. The number of hidden layers in the neural network can significantly affect the smoothness of the curvature of each L/d layer. Readers should balance computation time and model performance if they wish to develop their own models.

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 transformation 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 enhance engineers' routine workflows through the use of open-source tools. The case studies presented in this paper demonstrate the integration of machine learning with geotechnical engineering, enhancing the efficiency of assessment and design tasks. The author aims to make the findings of this paper publicly accessible. Readers are encouraged to utilize the online materials, including raw data, Python code and trained machine learning models, in accordance with the license requirements specified on GitHub. As we approach the fourth industrial revolution, it is believed that the application of AI in geotechnical engineering will continue to expand..

# 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