MACHINE LEARNING APPROACH FOR APPROXIMATING DESIGN PARAMETERS FROM ENGINEERING CHARTS

WaiLeung Ng
Tetra Tech Coffey, Melbourne, Australia

https://doi.org/10.56295/AG6112

ABSTRACT

Geotechnical engineers have traditionally relied on engineering charts for the analysis and design of specific geotechnical problems. However, interpolating target design parameters, particularly on logarithmic scale charts, 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 in some cases, 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.

1 INTRODUCTION

For many years, engineers have depended on engineering charts for assessments and designs related to geotechnical problems. These practices often originate from academic research or field observations, leading to the development of specialized charts 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 charts 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 charts. This regression can be performed using machine learning (ML) techniques where data is input into an ML algorithm, and the model is trained to minimize a defined loss function, thereby enhancing predictive accuracy.

Once trained, ML models can be reformulated into simplified closed-form equations or visual representations, improving interpretability and facilitating their integration into engineering workflows. This paper presents a practical approach to digitising and approximating geotechnical design charts using ML algorithms. Two case studies are provided to demonstrate the improved efficiency and scalability of this method in routine geotechnical analysis.

2 LITERATURE REVIEW

The integration of ML methods into geotechnical engineering has evolved significantly, progressing from early theoretical studies to practical and field-ready applications. Initial efforts in the 1990s were largely confined to basic regression models, limited by both sparse datasets and restricted computational capabilities. However, with the rapid expansion of data availability and advances in computing power, the application of ML techniques in geotechnical engineering has grown substantially (Shahin, 2025). Over the past three decades, a range of sophisticated ML models including artificial neural networks (ANN), support vector machine (SVM), Gaussian process regression (GPR) and decision trees (DT) have been employed to address a wide variety of geotechnical problems.

Shao et al. (2023) assessed the performance of various ML algorithms in tasks such as soil classification, landslide susceptibility mapping, rock type classification, rock deformation forecasting and compressive strength prediction. Similarly, Zhao et al. (2024) provided a comparative review of ML methods for optimising the prediction of key geotechnical properties such as compressibility parameters, permeability coefficients and uniaxial compressive strength. In an earlier study, Shahin (2014) utilised data from load tests on piles to train ANN models for estimating the ultimate bearing capacity of piles. More recently, Shen and She (2023) applied a hybrid approach combining Independent Component Analysis and Multivariate Adaptive Regression Splines (ICA–MARS) to model the load–displacement behaviour of pre-stressed ground anchors.

While much of the existing literature emphasises the use of complex ML models to capture intricate and non-linear relationships within large datasets, this paper instead focuses on the development of simple and scalable ML applications customised to everyday geotechnical engineering practice. In particular, it addresses the automation and enhancement of Category 2 design and analysis workflows, which are widely used in geotechnical design due to their balance of theoretical rigour and practical simplicity (Poulos, 1989).

Category 2 methods typically rely on semi-empirical procedures involving engineering design charts derived from analytical or experimental studies. Although these charts are conceptually straightforward, they often require manual interpretation, interpolation between curves or repetitive calculations when applied to multiple design cases or parametric studies. Such processes are not only time-consuming but also prone to user interpretation error, especially when dealing with low-resolution charts or large-scale design tasks.

This paper proposes a structured approach to digitising these traditional design charts, transforming them into machine-readable datasets through coordinate extraction, curve fitting and numerical interpolation techniques. Once digitised, the charts can be integrated into lightweight ML models that allow for real-time querying, automated chart reading and bulk design evaluations. This not only enhances computational efficiency but also preserves the interpretability and traceability in Category 2 designs, thereby facilitating a practical step toward modernising geotechnical workflows.

3 DIGITISATION AND MACHINE LEARNING WORKFLOW

A workflow diagram summarising the machine learning process is presented in Figure 1. It illustrates the key stages from data acquisition to model deployment. The process begins with the collection and digitisation of engineering data from design charts, followed by data preprocessing which may include cleaning, normalisation and log transformation. Model development involves selecting suitable algorithms and tuning their hyperparameters. After training and validation, the model is tested across various design scenarios to assess its robustness. Finally, the model is deployed to support Category 2 geotechnical design tasks with outputs translated into design parameters and integrated into practical engineering workflows.

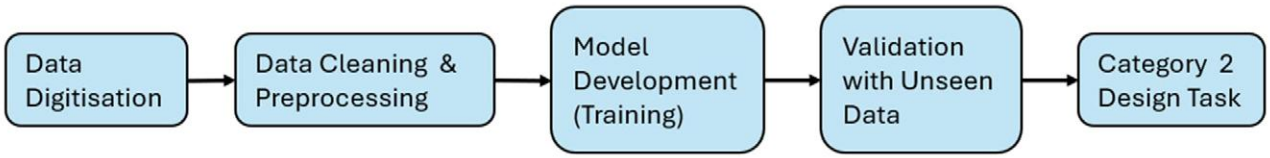


Figure 1: Workflow diagram of machine learning process

Data acquisition is carried out using WebPlotDigitizer which is an open-source tool that applies machine learning and computer vision to extract data from images. Being widely used across disciplines including astrophysics (Marin et al., 2017), it enables efficient and accurate digitisation of graphical plots. This makes it particularly suitable for converting engineering design charts into structured and machine-readable formats, streamlining the data preparation process.

Data cleaning is a critical step to remove noise, inconsistencies and outliers that may adversely affect model performance. The choice of preprocessing strategy depends on the nature and distribution of the extracted data. In some cases, models perform adequately using raw and unprocessed values, particularly when the data is well-structured and uniformly distributed. However, for datasets exhibiting skewness or large magnitude variations, normalisation can improve learning efficiency and convergence. In scenarios where relationships span several orders of magnitude, log transformation has proven effective in enhancing model performance by linearising trends.

Selecting appropriate ML algorithms is essential to model development and depends on the complexity and characteristics of the dataset. In this study, models such as Gaussian process regression (GPR) and artificial neural networks (ANN) were considered for their ability to capture nonlinear relationships common in geotechnical design problems. Once selected, each model's performance is optimised through hyperparameter tuning, which involves adjusting settings such as kernel functions in GPR or the number of layers and neurons in ANN. This process aims to enhance model accuracy and generalisation across different design scenarios.

Following model selection, the dataset is divided into training and validation subsets to assess the model's ability to generalise beyond the data it was trained on. The training phase involves fitting the model to a portion of the data, allowing it to learn patterns and relationships between input features and target outputs. The validation phase then evaluates the model's performance on unseen data to detect overfitting. Key performance metrics include the coefficient of determination (R^2 score) which measures how well the predicted values align with actual observations, and the mean squared error (MSE) which quantifies the average squared difference between predicted and true values. A high R^2 value (closer to 1) and a low MSE indicate strong predictive performance and robustness, both of which are essential for reliable application in engineering design tasks

4 MACHINE LEARNING ALGORITHMS

Three ML algorithms were used across the two case studies presented in this paper: non-linear regression, GPR and fully connected neural networks (FCNN). Each was selected based on its suitability for the structure and complexity of the underlying data, as well as the characteristics of the design charts being digitised. Further details on these algorithms can be found in Nield (2022) and on the scikit-learn website at https://scikit-learn.org.

The non-linear regression approach used in Case 1 is well-suited to problems where the functional form of the relationship can be inferred from engineering principles or visual chart interpretation. In this case, a modified logistic function was used to represent the displacement influence factor I_G which varies with the dimensionless variable β and footing geometry. Polynomial regression was applied to fit the constants in the function to known data, resulting in a generalised closed-form expression. This approach provides a transparent and interpretable model which is desirable in many engineering applications.

GPR was applied in Case 2 to approximate the complex relationships associated with the bearing stratum correction factor R_b for end-bearing piles. GPR is a probabilistic modelling technique that is particularly effective for small to medium-sized datasets with non-linear patterns. It provides not only point predictions but also associated uncertainty estimates. In this study, several kernel functions were evaluated and optimised through a grid search procedure, with the best-performing kernel selected to produce the final model predictions.

The third algorithm, FCNN was also implemented in Case 2 to model complex and multi-dimensional relationships. Unlike GPR, FCNN does not rely on predefined functional forms or kernel structures. Instead, they learn underlying patterns through iterative adjustments of connection weights across multiple layers of neurons. In this study, a five-layer architecture was adopted, demonstrating strong predictive performance and robustness across unseen data.

The ML models developed in this study were implemented using widely adopted open-source Python libraries: scikit-learn, scipy and keras. Scikit-learn provides a robust framework for supervised learning including tools for regression, classification, model validation and hyperparameter tuning. Scipy complements this with advanced numerical optimisation capabilities which are particularly useful for non-linear curve fitting. Keras, built on top of TensorFlow, offers a high-level interface for designing and training neural networks, making it suitable for constructing and experimenting with deep learning models such as FCNN. These libraries are well-documented, actively maintained and widely used in both academic research and industry, ensuring transparency, reproducibility and efficient model deployment.

5 CASE 1: SETTLEMENT OF A SHALLOW FOOTING

This case study addresses the assessment of surface settlement beneath the centre of a shallow footing. A simplified non-linear regression approach was employed to approximate one of the key parameters in the current analytical formulation.

5.1 CURRENT EQUATIONS FOR A CIRCULAR 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:

$$\rho_{center} = \frac{q \cdot d \cdot I_G \cdot I_F \cdot I_E \cdot (1 - \nu^2)}{E_o} \tag{1}$$

where q is the average applied loading, d the equivalent footing diameter, I_G the displacement influence factor, I_F the foundation flexibility correction factor, I_E the foundation embedment correction factor, v the soil Poisson's ratio and E_o is the soil Young's modulus at the surface.

The foundation flexibility correction factor I_F is approximately given by:

$$I_F = \frac{\pi}{4} + \frac{1}{4.6 + 10 \cdot K_F} \tag{2}$$

where $K_F = \frac{E_f}{E_{sav}} (\frac{2t}{d})^3$, E_f is the footing Young's modulus, E_{sav} the average soil Young's modulus and t is the footing thickness.

The foundation embedment correction factor I_E is approximately given by:

$$I_E = 1 - \frac{1}{3.5 \cdot exp(1.22\nu - 0.4) \cdot \left(\frac{d}{z_e} + 1.6\right)} \tag{3}$$

where z_e is the depth of embedment of footing base below surface.

The chart for the displacement influence factor I_G is extracted from Mayne and Poulos (1999) and is illustrated in Figure 2.

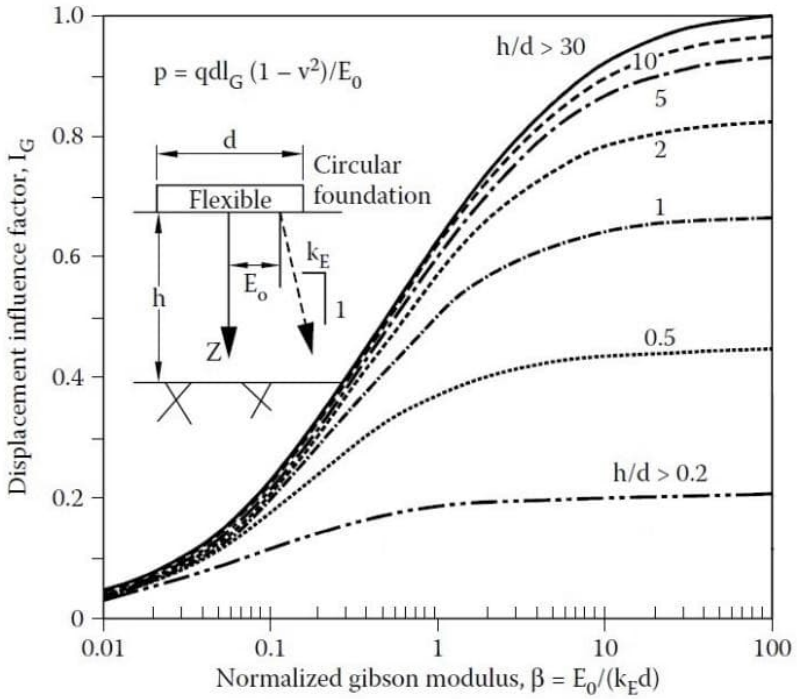


Figure 2: Displacement influence factor I_G (Adapted from Mayne and Poulos 1999)

To enable automation of the settlement calculation, a closed-form expression for I_G is required that approximates the family of S-shaped curves shown in Figure 2. These curves can be represented using a modified logistic function. The parameter β is plotted on a base-10 logarithmic scale, and the proposed form of the equation is:

$$I_{G,x} = f(\beta) = \frac{1}{m_x + n_x \cdot 10^{-k_x(log\beta)}} \tag{4}$$

which is simplified as:

$$I_{G,x} = f(\beta) = \frac{1}{m_x + n_x \cdot \beta^{-k_x}} \tag{5}$$

where $x = ln\left(\frac{h}{d}\right)$

5.2 MODEL DEVELOPMENT USING NON-LINEAR REGRESSION

For a specific value of x, there exists a corresponding modified logistic function characterized by the constants m_x, n_x, and k_x. To determine these constants, data points from each curve were extracted using WebPlotDigitizer. These data points were then analysed using a selected non-linear regression algorithm implemented in the open-source Python modules 'sklearn' and 'scipy'. The resulting values of m_x, n_x, and k_x are summarized in Table 1.

Table 1: Values of h/d with the associated constants

h/d	x = ln (h/d)	m_x	n_x	k_x
30	3.401	0.980	0.604	0.753
10	2.303	1.018	0.601	0.779
5	1.609	1.059	0.595	0.787
2	0.693	1.201	0.541	0.831
1	0.000	1.493	0.490	0.869
0.5	-0.693	2.240	0.446	0.892
0.2	-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 (m_x, n_x, and k_x) is treated as a dependent variable correlated with the independent variable x 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 R^2 scores and MSE. The results of the regression exercise are expressed in the following polynomials.

$$m(x) = (-0.0059x^3 + 0.0411x^2 - 0.0978x + 1.0855)^5 \quad (6)$$

$$n(x) = -0.0032x^3 + 0.0011x^2 + 0.0655x + 0.4927 \quad (7)$$

$$k(x) = 0.0043x^2 - 0.0465x + 0.8603 \quad (8)$$

Consequently, the displacement influence factor I_G can be expressed in a generalized form as follows:

$$I_G = f(\beta, x) = \frac{1}{m(x) + n(x) \times \beta^{-k(x)}} \quad (9)$$

Finally, all data points are input into Equation (9) to assess the overall accuracy. The resulting R^2 score of 0.9996 and MSE of 3.08e-5 indicate an excellent fit of the input data to the constructed equation. This is also corroborated by Figure 3, where the input data points (represented as green dots) align perfectly with the curved surface generated by Equation (9). 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.

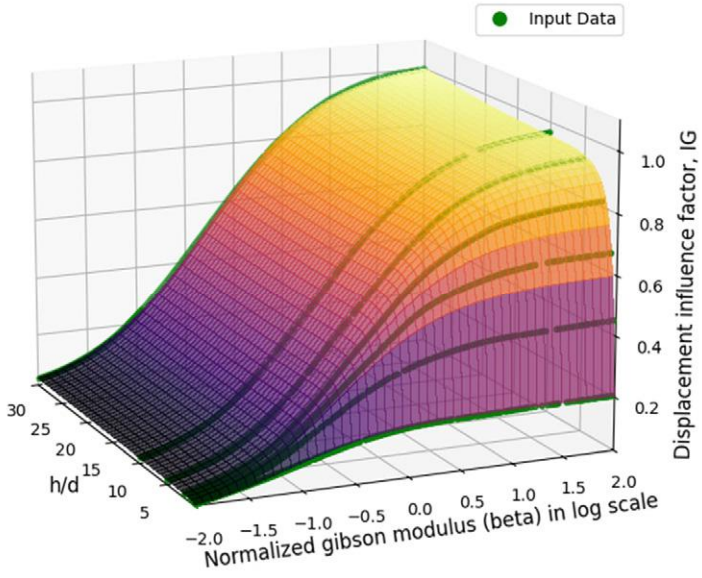


Figure 3: Three-Dimensional Illustration of I_G Value

5.3 MODEL VALIDATION AGAINST COMMERCIAL SOFTWARE – PLAXIS 3D

The reliability of Equation (9) is further verified using Plaxis 3D, a commercial finite element analysis (FEA) software developed by Bentley, specifically designed for advanced geotechnical engineering applications. As the derivation of Figure 2 is based on elastic theory, a linear elastic soil model was adopted in Plaxis 3D to ensure a consistent basis for comparison. To isolate the influence of the displacement factor I_G, the effects of embedment and footing flexibility were eliminated by positioning the footing at the ground surface and neglecting its thickness, thereby setting both I_E and I_F equal to 1.

Four scenarios with varying geometries and parameter combinations, none of which lie directly on the eight predefined design curves in Figure 2, were selected to test the generalisation capability of the model. Settlements calculated using Equations (1) and (9) are compared with those obtained from Plaxis 3D, as summarised in Table 2. Despite minor differences at the decimal level, the results from both methods are in close agreement. This comparison justifies the ability of the proposed non-linear regression model to produce reliable predictions, even for unseen combinations of geometric and material parameters.

Table 2: Comparison of settlements predicted by the non-linear regression model and Plaxis 3D

Scenario	Load, q (kPa)	Influence depth, h (m)	Footing diameter, d (m)	h/d	E_o (MPa)	k_E (MPa/m)	β	I_G	Assessed settlement (mm)	
									Non-linear regression	Plaxis 3D
1	150	15	2	7.5	25	0.2	62.5	0.94	10	10
2	300	10	4	2.5	25	1	6.3	0.79	35	35
3	300	5	4	1.25	35	0.5	17.5	0.70	22	22
4	200	5	3	1.67	35	1.5	7.8	0.74	12	12

6 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.

6.1 CURRENT CHART-BASED APPROACH FOR AN END-BEARING PILE

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

$$S = \frac{P}{dE_s} I_1 R_K R_b R_v \tag{10}$$

where P is the applied load, d the pile diameter, E_s the average soil modulus along the pile shaft, I_1 the influence factor for a rigid pile in a semi-infinite mass and R_k, R_b, R_v 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 R_b will be utilised to demonstrate the approximation exercise. The charts for R_b are extracted from Poulos and Davis (1980) and are presented in Figure 4. The value of R_b depends on three independent variables: the ratio of E_b/E_s, the ratio of L/d and the K value. Here, E_b represents the Young's modulus of the founding stratum and L is the pile length. The variable K is defined as:

$$K = \frac{E_p R_A}{E_s} \tag{11}$$

where E_p is the Young's modulus of pile and R_A is the area ratio of pile.

Identifying the complex relationship between the target R_b 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, GPR and FCNN implemented in the open-source Python modules 'sklearn' and 'keras' are utilised to train ML models in this case.

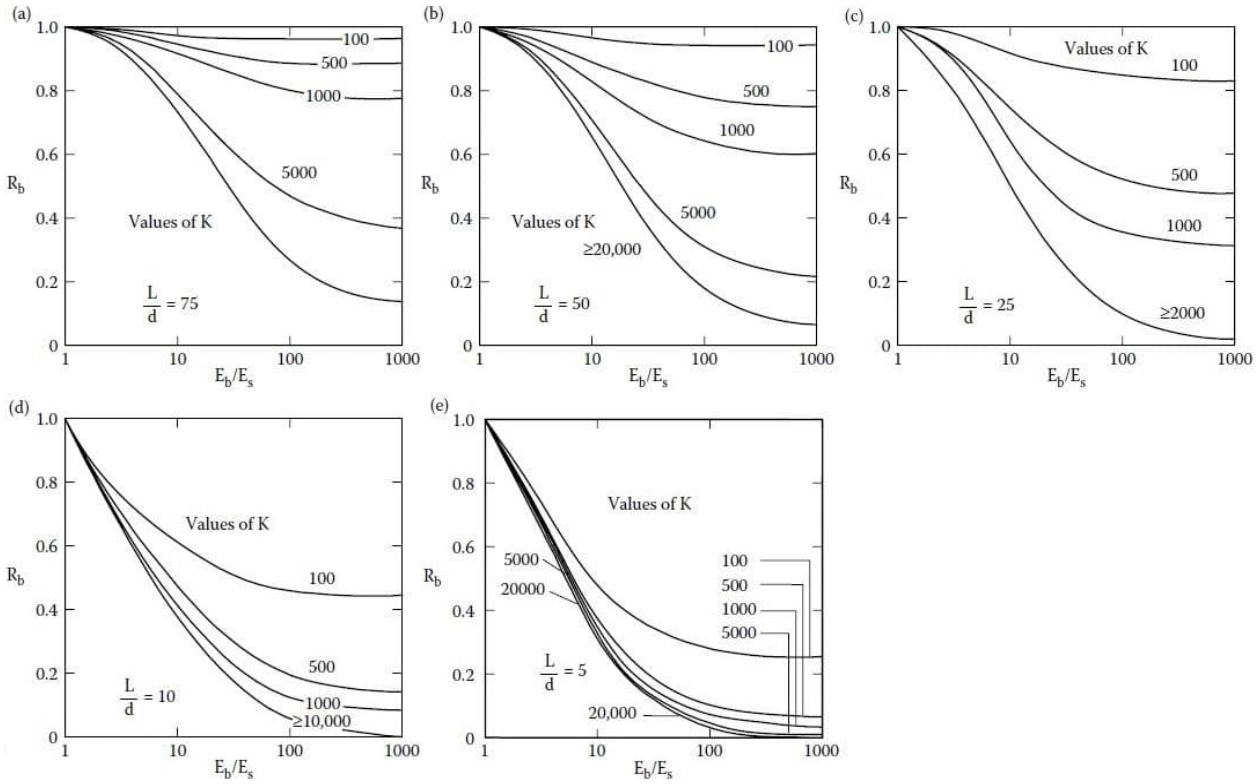


Figure 4: Bearing stratum correction factor R_b for values of L/d of 75, 50, 25, 10 and 5 (Adapted from Poulos and Davis, 1980)

To initiate this process, data points were extracted from Figure 4 using WebPlotDigitizer. The extracted data was then used to train the two selected ML algorithms, resulting in the development of two digital models that can recognize 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.

6.2 PERFORMANCE OF GAUSSIAN PROCESS REGRESSION

The performance of GPR is sensitive to the absence of some data in Figures 4(c) and 4(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 mitigate this issue, supplementary data points were added by extending the original dataset from the original maximum bounds, as illustrated in Figures 5(c) and 5(d). Furthermore, for the case where L/d equals 25, cross lines were generated using polynomial regression at K-R_b planes for log (E_b/E_s) values of 1, 1.5, 2, 2.5 and 3 as shown in Figure 5(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 was 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 yielded the lowest MSE was selected as the optimal estimator.

Before analysing the four-dimensional relationship, three-dimensional trials were 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

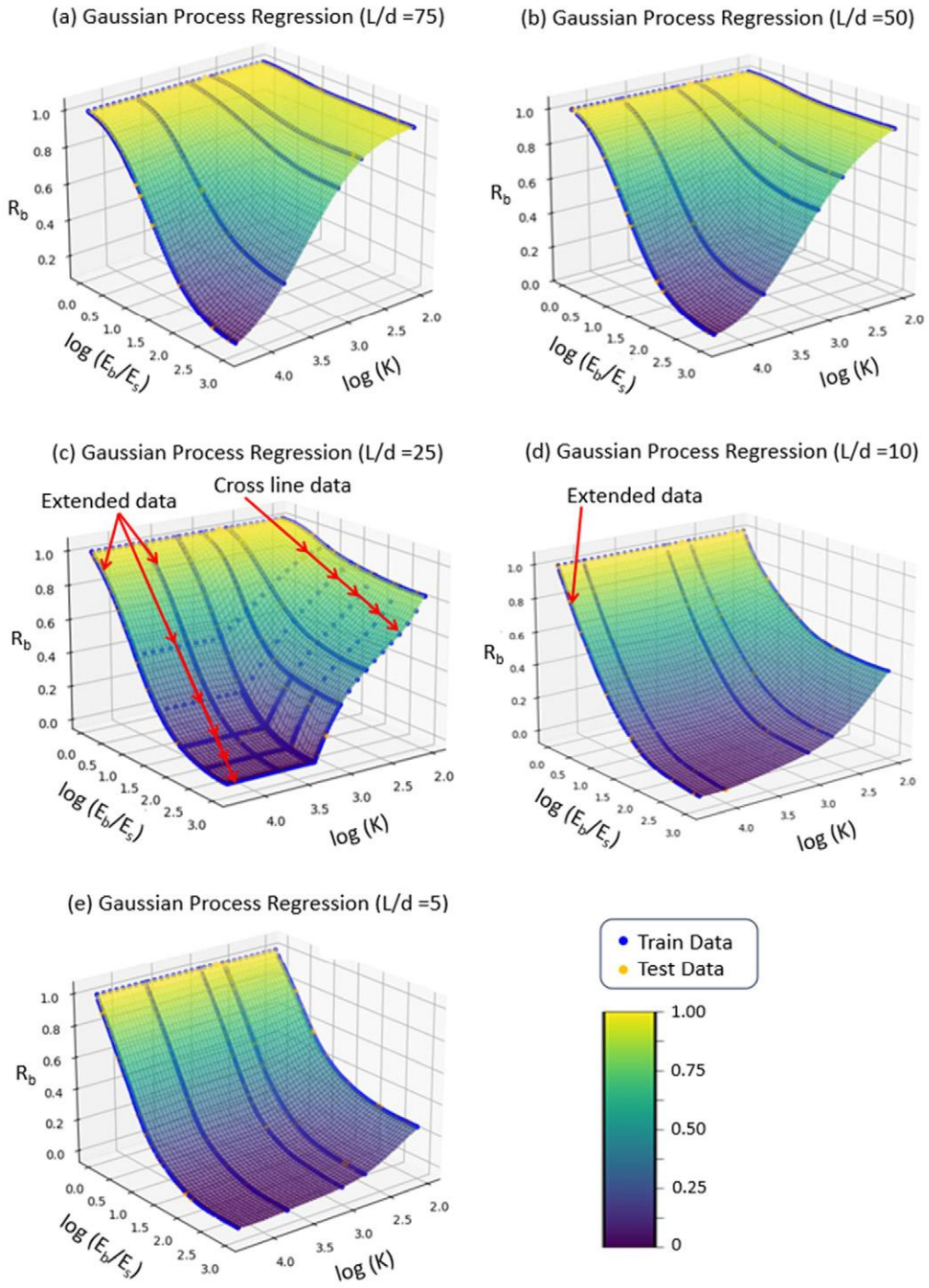


Figure 5: Approximation of R_b for various L/d values using GPR

Three-dimensional plots illustrating the data relationships for five different L/d ratios are presented in Figure 5. Each L/d layer represents predictions for 200 x 200 grid points along two axes, i.e. log (K) and log (E_b/E_s). The maximum MSE of these trials is 1.25e-6, indicating that the GPR model fits the training data well.

The subsequent stage involves the construction of a comprehensive four-dimensional model that simultaneously incorporates all four input variables, utilising the optimal kernel hyperparameters determined from prior three-dimensional trials. The resulting model achieves a MSE of 1.54e-6 on the 10% testing dataset.

As the model depends on three independent input variables, it represents a four-dimensional function that cannot be directly visualised using standard graphical methods. To facilitate interpretation, eight discrete values of the L/D ratio are selected, while the remaining variables log (K), log (E_b/E_s) and R_b are varied continuously along the orthogonal axes.

A series of simplified three-dimensional plots corresponding to each selected L/D value is presented in Figure 6. Although only eight values or layers are depicted, it should be noted that L/D is treated as a continuous variable within the model framework. Importantly, three of the selected L/D values (60, 35, and 20) correspond to data points entirely excluded from the training set. The smooth transitions observed across the plotted layers indicate the model's robustness and its ability to generalise effectively to previously unseen data.

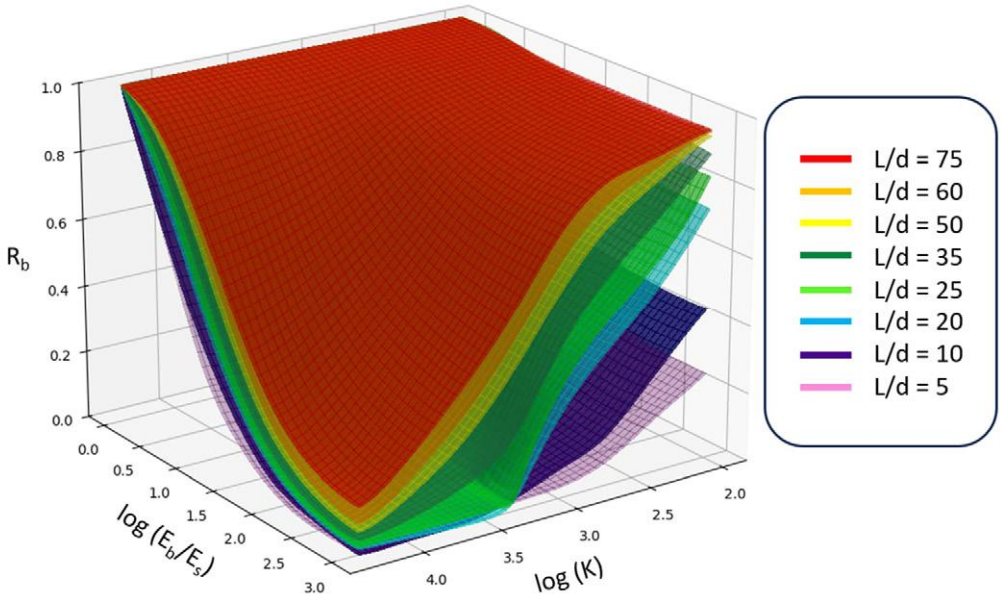


Figure 6: Combined three-dimensional plot of R_b for various L/d values using GPR

A key advantage of GPR is its ability to provide not only a mean prediction but also an associated variance estimate, enabling the construction of prediction intervals. For the 10% test dataset, the average predictive standard deviation was approximately 0.0009, resulting in a 95% confidence interval of around ±0.0018. This narrow interval indicates a high level of model confidence within the prediction domain.

6.3 PERFORMANCE OF FULLY CONNECTED NEURAL NETWORK

A fully connected neural network (FCNN) is a type of artificial neural networks 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 R^n to any desired degree of accuracy, provided that a suitable non-linear activation function is employed. Here, R^n 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 were configured with the following number of neurons: 128, 64, 64, 32 and 8 respectively. The Rectified Linear Unit (ReLU) was selected as the activation function for each hidden layer, while a linear activation function was 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 for 10% testing data is 1.81e-5, indicating strong predictive performance. Similarly, a series of simplified three-dimensional plots corresponding to eight selected L/D value is presented in Figure 7, providing a visual representation of the trained FCNN model. Notably, unlike GPR, cross line data is not required for the FCNN, as overfitting is not a concern with this model.

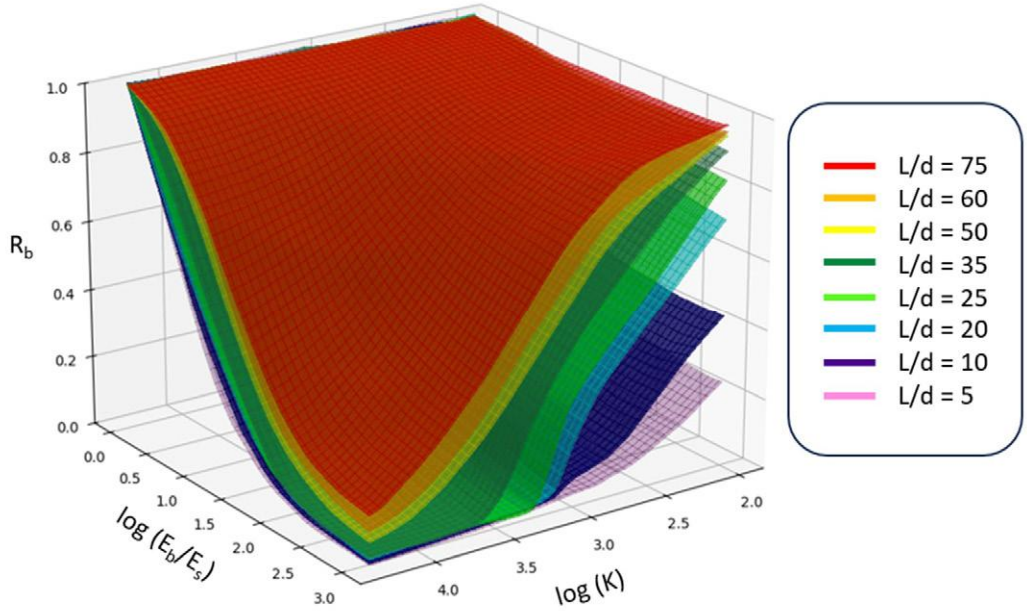


Figure 7: Combined three-dimensional plot of R_b for various L/d values using FCNN

To assess predictive uncertainty, a Monte Carlo (MC) dropout version of the deterministic FCNN model was developed. The dropout model applied a dropout rate of 0.1 at each neural layer and performed 100 stochastic forward passes for each prediction. The average standard deviation across the test dataset is approximately 0.06, indicating a low level of uncertainty relative to the target range of 0 to 1. While this standard deviation is higher than that of the GPR model, the variation remains negligible in the context of pile settlement assessment, where a reasonable degree of tolerance is generally acceptable under Category 2 design methods.

6.4 COMPARISON OF MODEL OUTPUTS

To compare the outputs 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 3.

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 R_b value remains generally consistent across the different methods.

Table 3: Comparison of R_b using different methods

L/d	K	E_b/E_s	log (E_b/E_s)	R_b			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	0.78	0.78	0.01
60	2000	50	1.7	0.62	0.62	0.63	0.01
50	300	500	2.7	0.83	0.82	0.85	0.03
50	2000	50	1.7	0.54	0.54	0.55	0.01
40	300	10	1.0	0.90	0.90	0.86	0.04
25	300	10	1.0	0.83	0.81	0.80	0.03
25	800	500	2.7	0.39	0.38	0.36	0.03
15	300	30	1.5	0.52	0.49	0.50	0.03
10	300	30	1.5	0.37	0.35	0.38	0.03

6.5 DIGITISATION OF OTHER FACTORS

The other three factors can be determined in the same way as I_G in Case 1 using non-linear regression model or R_b in this case using GPR or FCNN. After non-linear regression approximation, I_1 and R_v can be expressed by Equations (12) and (13) respectively. The plots of I_1 and R_v computed from these equations are shown in Figures 8(a) and 8(b) respectively. The excellent agreement with the original charts in Poulos and Davis (1980) validates the precision of the regression-based approximations.

$$I_1 = f\left(\frac{d_b}{d}, \frac{L}{d}\right) = A \cdot x^2 + B \cdot x + C \tag{12}$$

$$R_v = f(K, v_s) = m \cdot v_s + c \tag{13}$$

where d_b is the diameter of pile base and v_s is the Poisson's ratio

$$A = 0.005\left(\frac{d_b}{d}\right)^2 - 0.0493\left(\frac{d_b}{d}\right) + 0.139$$

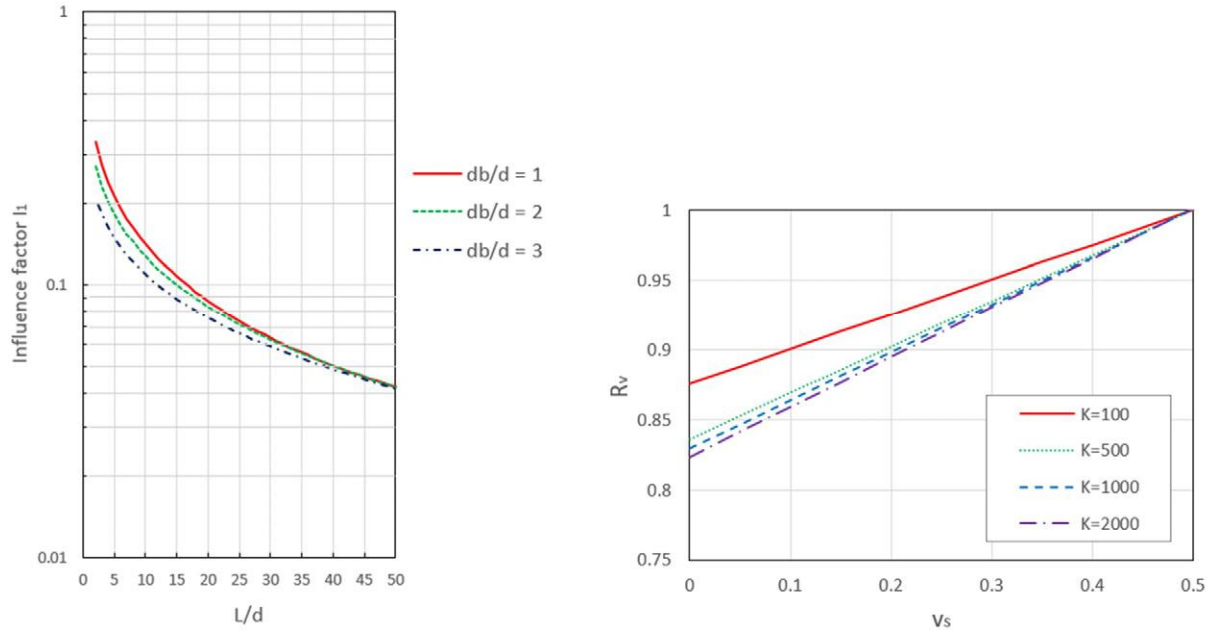
$$B = -0.0107\left(\frac{d_b}{d}\right)^2 + 0.1445\left(\frac{d_b}{d}\right) - 0.5312$$

$$C = 0.0036\left(\frac{d_b}{d}\right)^2 - 0.1029\left(\frac{d_b}{d}\right) + 0.5433$$

$$x = log\left(\frac{L}{d}\right)$$

$$m = 0.046(logK)^3 - 0.4199(logK)^2 + 1.3189(logK) - 1.0762$$

$$c = -0.023(logK)^3 + 0.2105(logK)^2 - 0.6615(logK) + 1.5405$$



(a) Influence factor I_1 **(b) Correction factor R_v**

Figure 8: Plots of I_1 and R_v using approximate equations

The remaining correction factor R_k are defined by the following function:

$$R_K = f\left(K, \frac{L}{d}\right) \tag{14}$$

Similar to R_b factor, a FCNN was trained and validated to approximate this function with engineering accuracy so that the R^2 score is close to 1. Figure 9 plots the corresponding regression surface of this trained model.

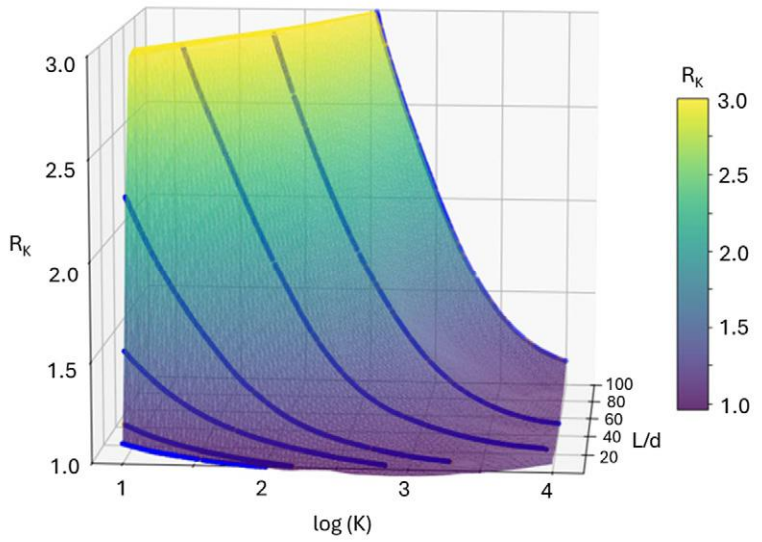


Figure 9: Graphical representation of R_K using FCNN

6.6 MODEL VALIDATION AGAINST COMMERCIAL SOFTWARE – PLAXIS 3D

The reliability of Equations (12) and (13) along with the two trained ML models was validated using Plaxis 3D. Since all four factors in Equation (10) were derived based on elastic theory originally, a linear elastic soil model was adopted in Plaxis 3D to ensure a consistent basis for comparison. The pile was modelled as a circular concrete pile with a modulus of elasticity (E_p) of 33,000 MPa and simulated as a volume element with rigid interface conditions. The Young's modulus of the shaft soils (E_s) ranges from 30 MPa to 60 MPa, while that of the bearing stratum (E_b) at the pile base varies between 360 MPa and 5,000 MPa. A uniform Poisson's ratio of 0.3 was applied across all materials.

To assess the generalisation capability of the models, geometric configurations were intentionally selected such that the resulting data points did not coincide with the predefined chart lines. The corresponding input parameters are summarised in Table 4.

Predicted settlements obtained from the ML models are compared against those computed using Plaxis 3D, as presented in Table 5. The close alignment between the two sets of results affirms the predictive reliability of the proposed ML approach, even when applied to previously unseen combinations of geometric and material parameters across all four scenarios.

Table 4: Geometric and material properties used in model validation

Scenario	Load, P (kN)	L (m)	d (m)	d_b (m)	d_b/d	L/d	E_s (MPa)	E_b (MPa)	E_b/E_s	K
1	4000	10	0.6	0.6	1	16.7	40	5000	125	825
2	6000	15	0.75	0.75	1	20	35	1000	29	943
3	4500	25	0.6	0.6	1	41.7	30	3000	100	1100
4	20,000	35	2	2.5	1.25	17.5	60	360	6	550

Table 5: Comparison of settlements predicted by ML models and Plaxis 3D

Scenario	Computed values of influence factor and correction factors				Assessed settlement (mm)	
	I_I	R_K	R_b	R_v	ML models	Plaxis 3D
1	0.093	1.074	0.221	0.933	4	5
2	0.082	1.089	0.392	0.933	8	9
3	0.053	1.272	0.549	0.932	9	10
4	0.091	1.132	0.697	0.935	11	10

7 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 I_G 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 2 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 chart reading method, establishing a closed-form solution that simplifies computation while preserving accuracy.

Case 2 illustrates the application of GPR and FCNN to explore the relationships among multiple variables. While the data reveals certain patterns, deriving a simple closed-form equation that associates all variables seems impractical. Therefore, GPR and FCNN were utilised to directly predict the dependent variable R_b, eliminating the need to determine the equation's form. The outcome of these regression analyses consists of two ML models available as digital files for download from GitHub at https://github.com/OpenGTi/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 ML models directly for predictions using the Python code provided in the repository.

7.1 RECOMMENDATION ON MODEL SELECTION

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 bias 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 a neural network 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 intending to develop their own models should consider the trade-off between computational time and model accuracy.

There is no simple rule for selecting the most appropriate ML model, as performance depends heavily on the underlying data patterns. In most cases, a trial-and-error approach is necessary. From a complexity perspective, the GPR model is generally more intricate due to the non-intuitive nature of kernel functions, particularly for those without a background in data science. In terms of training time, the GPR model used in Case 2 required approximately 30 minutes to train, whereas the FCNN model completed training in just 5 minutes for 100 epochs. These durations are based on a laptop equipped with an AMD Ryzen 7 series CPU. Considering its relative simplicity and reduced training time, the FCNN model is recommended for similar applications involving the digitisation and prediction of engineering chart data.

7.2 INTERPRETABILITY OF NEURAL NETWORKS

Artificial neural networks are often perceived as "black-box" models due to their limited interpretability. To enhance interpretability of the FCNN model in Case 2, a permutation feature importance analysis was conducted. This technique evaluates the contribution of each input variable by measuring the increase in MSE when the values of a given feature are randomly shuffled. The results indicate that the model is most sensitive to log (E_b/E_s) with an associated increase in

MSE of approximately 0.12, followed by L/d value (0.07) and log (K) (0.04), which exhibits comparatively lower influence. Although log (E_b/E_s) contributes the most, the overall magnitudes are of the same order. This suggests that all three features have a similar meaningful impact on the model's predictions.

7.3 MACHINE LEARNING MODEL LIMITATIONS

There are several limitations associated with the adopted ML approaches. Both the GPR and FCNN models may suffer from overfitting, particularly when trained on limited or non-representative datasets which can reduce their ability to generalise to unseen scenarios. The performance of these models is highly dependent on the density and distribution of the training data. Sparse or unevenly spaced points can compromise prediction accuracy. Furthermore, the models are only reliable within the parameter range captured during training. Therefore, extrapolation beyond this range should be approached with caution. Another critical limitation lies in the quality of the digitised input data. Inaccuracies during the digitisation process such as resolution issues or coordinate misinterpretation may introduce systematic errors. These challenges underline the importance of robust validation procedures, careful data preparation and transparency in model development to ensure practical applicability and reproducibility.

7.4 FUTURE POTENTIALS

While digitising design charts may initially appear more complex than using traditional tools such as spreadsheets or commercial software like PIGLET or RSPile, the additional effort is well justified once the process is embedded within an automated workflow. This transformation converts hard-copy figures or image-based charts into structured, machine-readable data that can be processed programmatically.

More importantly, digitised design charts become compatible with modern generative AI tools, including large language models (LLMs). This opens new possibilities for intelligent design automation, real-time decision support and integration with advanced geotechnical analysis systems.

Since the introduction of ChatGPT, there has been a noticeable shift, with more geotechnical engineers beginning to explore programming languages such as Python. Free platforms like Google Colab and Anaconda have further lowered the barrier to entry by offering accessible environments and extensive support for coding and data analysis. As a result, basic programming, machine learning, generative AI and prompt engineering are emerging as foundational skills for the next generation of geotechnical engineers. With chart-based knowledge in a digital format, engineers are increasingly empowered to develop custom tools and applications that enhance productivity in practice.

8 CONCLUSION

The case studies presented herein demonstrate accessible approaches to developing ML models for digitising graphical data, solving non-linear regression problems and uncovering relationships among multiple variables. Compared to traditional graphical interpolation, ML offers a practical and efficient alternative for geotechnical engineers, streamlining routine design workflows. In particular, the FCNN employed in Case 2 illustrates a simple yet effective method for capturing complex data relationships with minimal manual data handling.

In recent years, generative AI has gained significant attention. However, its integration into geotechnical engineering remains constrained by the limitations of traditional chart-based methods. The digitisation of design charts serves as a crucial step in bridging the gap between established engineering practices and emerging AI technologies. The methodology demonstrated in this study is scalable well beyond the scope of the presented cases. For instance, if the equations and charts from an entire textbook are converted into structured and machine-readable datasets, they can be integrated into large language models (LLMs). A customised AI agent can then provide informed responses to queries across a wide range of geotechnical topics including retaining walls, tunnelling and slope stability. This approach represents more than the digitisation of individual charts. It signifies the transformation of geotechnical knowledge into an interactive and AI-driven tool.

The rapid advancement of AI provides opportunities to replace laborious manual processes with efficient and automated solutions. This shift is increasingly evident within the engineering sector, where digital transformation is no longer exclusive to large software developers but is becoming embedded in the everyday practices of individual engineers. It is anticipated that this evolution will continue, advancing from traditional analytical solutions toward numerical methodologies.

The author is committed to making the findings of this study publicly accessible. Readers are encouraged to explore the accompanying online resources available on GitHub, including raw datasets, Python codes and two trained machine learning models, in accordance with the specified licensing terms. As the profession moves into the era of the fourth industrial revolution, the role of AI in geotechnical engineering is expected to expand significantly, shaping the future of design, analysis and decision-making in the field.

9 ACKNOWLEDGEMENT

The author sincerely appreciates Professor Harry Poulos for his invaluable suggestions, which have significantly enhanced the quality and clarity of this paper.

CRediT authorship contribution statement

WaiLeung Ng: Conceptualization, Writing - original draft, Writing – review & editing.

10 REFERENCES

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

Marin, F., Rohatgi, A. and Charlot, S. (2017). Webplotdigitizer, a polyvalent and free software to extract spectra from old astronomical publications: application to ultraviolet spectropolarimetry. *Société Française d'Astronomie et d'Astrophysique (SF2A) 2017,* 237-241.

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

Nield, T. (2022). *Essential math for data science*. Sebastopol: O'Reilly.

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

Shahin, M. A. (2014). State-of-the-art review of some artificial intelligence applications in pile foundations, *Geoscience Frontiers (2014)*, 1-12

Shahin, M. A. (2025). Progression of artificial intelligence / machine learning in geotechnical engineering. *Machine Learning and Data Science in Geotechnics*, *Volume 1, Issue 1 (2025),* 1-5

Shao, W., Yue, W., Zhang, Y., Zhou, T., Zhang, Y., Dang, Y., Wang, H., Feng, X. and Chao, Z. (2023). The application of machine learning techniques in geotechnical engineering: a review and comparison. *Mathematics 2023, 11, 3976*, 1-16

Shen, H. and She, W. (2023). Load-displacement prediction of pre-stressed ground anchors: a hybrid machine learning approach. *Proceedings of the 14th Australia and New Zealand Conference on Geomechanics, Cairns 2023 (ANZ2023)*

Zhao, T., Shen, F. and Xu, L. (2024). Review and comparison of machine learning methods in developing optimal models for predicting geotechnical properties with consideration of feature selection. *Soils and Foundations, 64 (2024) 101523*, 1-28