Capacity prediction and design optimization for laterally loaded monopiles in sandy soil using hybrid neural network and sequential quadratic programming

Article *in* Computers and Geotechnics · November 2023

DOI: 10.1016/j.compgeo.2023.105745

3 authors, including:

Amir Hosein Taherkhani
University of New Hampshire

Fei Han
University of New Hampshire

Capacity prediction and design optimization for laterally loaded monopiles in sandy soil using hybrid neural network and sequential quadratic programming

Amir Hosein Taherkhani[1], Qipei Mei[2], Fei Han[1]

[1] Department of Civil and Environmental Engineering, University of New Hampshire, Durham, NH, USA
[2] Faculty of Engineering - Civil and Environmental Engineering Dept, University of Alberta, Edmonton, AB, Canada

ABSTRACT

Data driven methods have gained momentum in recent years in solving highly non-linear engineering problems that are challenging to solve using conventional methods. In this paper, we present a hybrid neural network model to predict the lateral response of large-diameter monopiles in multi-layered soil. The hybrid neural network consists of a mixture of convolutional and fully-connected layers, which capture the impacts of the soil profile, the pile geometry, and loading conditions on the lateral load response of monopiles. To train the neural network model, we produced data from high-fidelity three-dimensional (3D) finite element (FE) models that are validated against full-scale pile load tests. To ensure consistent model performance across the entire range of pile capacities considered in the dataset (ranging from approximately 100 kN to 100,000 kN), we utilize the relative error (percentage error) as the criterion for training the model. To achieve this goal, we explored six different combinations of data transformation methods (i.e., natural logarithm and root transformations) and cost functions. Among these models, the model trained with Mean Squared Error (MSE) using natural logarithm transformation yielded the most accurate and consistent predictions of the lateral capacities of monopiles. To demonstrate the strengths of the developed neural network model, it was used as a surrogate model to perform pile design optimization using sequential

quadratic programming. In addition, a design example is provided to show how the developed method can be easily implemented.

1 INTRODUCTION

The offshore wind industry has grown exponentially in the last decade driven by the increasing demand for clean renewable energy. Large-scale offshore wind turbines are often supported on monopiles, a foundation type that is made of a single open-ended steel pipe driven or jacked into the seabed. The lateral load design for these large-diameter monopiles has traditionally relied on the *p*−*y* method. This method considers the pile as a one-dimensional Euler-Bernoulli beam while the soil medium is modeled as a series of non-linear springs attached to the beam. Since this method was originally developed and used for slender pile design, its applicability for large-diameter monopiles is not well established (Doherty and Gavin 2012; Farahani et al. 2022; Suryasentana and Lehane 2016). Evidence has shown that the p-y method may result in inaccurate predictions for the lateral response of monopiles (Han et al. 2015, 2017; Suryasentana and Lehane 2016).

In order to improve the design for large-diameter monopiles, researchers have developed new design approaches for monopiles based on three-dimensional finite-element (3D FE) analyses. When compared with the *p*−*y* method, a 3D FE analysis can fully capture the 3D pile-soil interactions when a pile is laterally loaded, producing more accurate capacity predictions. 3D FE analysis has been widely used to study the lateral load response of single piles (Ahmadi and Ahmari 2009; Brown and Shie 1990; Chatterjee et al. 2015; Cheng et al. 2021; Kementzetzidis et al. 2018; Murphy et al. 2018; Peng et al. 2010). Recently, Byrne et al. (2020) conducted a series of 3D FE analyses for monopiles placed in stiff and overconsolidated clay, based on which the updated $p-y$ method was proposed. Taborda et al. (2020) conducted a 3D FE analysis for laterally loaded monopiles placed in dense sands, accounting for a constitutive model that captures the behavior of sand with a range of relative densities. Despite its accurate results, 3D FE analysis is computationally costly, often taking several days to complete a full-scale FE simulation (Han et al. 2015; Xu et al. 2013). In addition, the accuracy of the FE analysis is highly dependent on the knowledge and experience of the modeler. Proper meshing and selection of an appropriate constitutive model for the soil are both critical to ensure the quality of 3D FE analyses.

Unlike physics-based models (e.g., FE analyses), which are constructed based on physics laws and appropriate assumptions, Artificial Neural Network (ANN) models are developed to recognize the underlying patterns within the data collected for the problem of interest (Shahin 2016). When properly trained, ANN models can produce fast predictions (taking split-second computational time) with high accuracies. Inspired by the biological neural network, ANN was initially developed for simulating neurological networks in the 1940s (McCelloch and Pitts 1943). In recent years, ANN has gained popularity in solving complex geotechnical engineering problems due to the rapidly growing computational capacity and data availability (Chan and Low 2012; Huang et al. 2022; Kordjazi et al. 2014; Makasis et al. 2018; Moeinifard et al. 2022). For instance, Pham et al. (2020) and Kardani et al. (2020) developed ANN models to estimate the axial bearing capacity of driven piles. Xiao et al. (2022) developed a machine-learning based spatio-temporal forecasting model to predict landslide locations and consequences. Zhang et al. (2022) used convolutional neural network (CNN) to characterize the soil spatial variability from limited cone penetration test (CPT) data. Lai et al. (2022) developed an ANN-based framework to detect particle contacts for discrete element method

(DEM). ANN has been demonstrated through these studies to be a promising instrument for solving geotechnical engineering challenges.

In this paper, we developed an ANN model to predict the lateral load response of monopiles installed in multi-layered sandy soil. The model takes pile geometries, loading conditions, and the cone penetration test (CPT) data as inputs, generating predictions for the lateral pile capacities corresponding to different pile rotation levels at the mudline. We investigated the effectiveness of various data transformation techniques and cost functions in reducing the skewness of training datasets and improving model performance. In addition, the developed model was used as a surrogate model to perform pile design optimization.

2 METHODOLOGY

In this section, we first briefly explain the fundamental ideas of fully-connected (FC) neural work and convolutional neural network (CNN). Based on these two ANN methods, we propose the hybrid neural network architecture designed for this study.

2.1. Hybrid Neural Network

Deep neural network or Deep Learning (DL) is an ANN model with a structure of more than three layers (Patterson and Gibson 2017). Fully-connected (FC) neural network, also known as feed-forward neural network, is the simplest yet most widely-used type of DL. FC neural networks are often used for feature extraction for non-sequential data. As shown in Figure 1a, a FC neural network is composed of an input layer, several hidden middle layers, and an output layer. Each layer contains one or multiple nodes (also known as artificial neurons), the most fundamental units in an FC neural network. Resembling the function of biological neurons, nodes are designed to process the information received from upstream nodes and then pass the processed information to the downstream nodes. To achieve this goal, a node first uses a linear function to map the outputs of all nodes from the previous layer into a single value, which is then passed into a non-linear activation function to generate the output for this node (Figure 1b). The output values for all nodes in a layer are then used as the input for the next layer. The non-linear activation function, even if it is as simple as a bilinear function [e.g., Rectified Linear Unit: ReLU(z)=max(0,z)], plays a critical role that empowers an FC neural network to capture the complex non-linear patterns in the data.

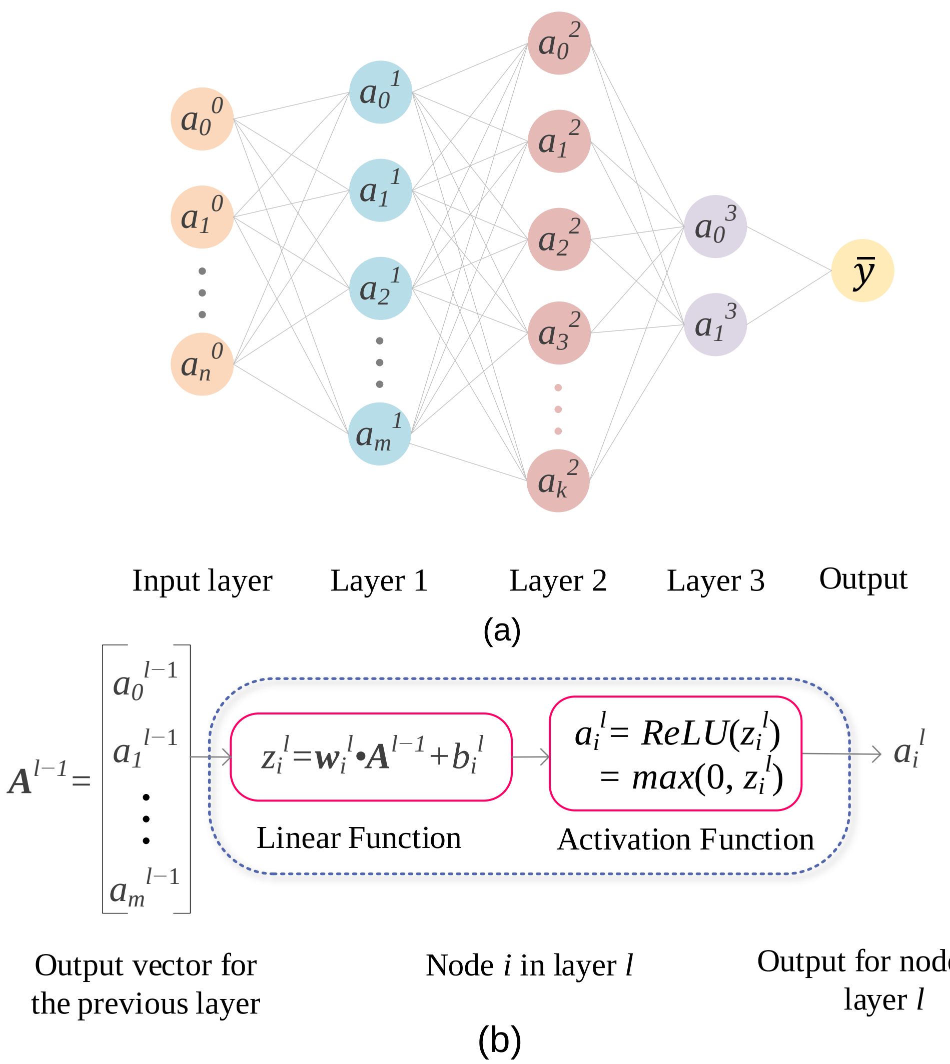

Figure 1 (a) Architecture of a typical FC neural network. a_i^l denotes the output of node i in layer l. Layer 0 is the input layer in the network.; (b) The operation of a node in an FC neural network. $\boldsymbol{A}^{l-1}$ is the output vector for layer l–1, which consists of nodes denoted by a_0^{l-1} to a_m^{l-1}. $\boldsymbol{w}_i^l$ is a vector of weights for node i in layer l. b_i^l is the bias for node i in layer l. z_i^l is an intermediate variable for node i in layer l.

Convolutional neural network (CNN) is another branch of DL that is often used to capture higher-order features from structured data such as images (Patterson and Gibson 2017). The application of CNN models for image recognition is one of the primary reasons for the thriving of CNN and DL models in the past decade. Convolutional blocks are the fundamental elements in a CNN model. A convolutional block often consists of three types of operations:

convolution, non-linear activation, and pooling. Figure 2 illustrates the detailed mechanisms of these operations using one one-dimensional (1D) convolutional layer with a single kernel as an example. In contrast to the two-dimensional (2D) CNNs that are often used for image classification, we use a 1D CNN in this study to capture patterns in the CPT data (in the format of a 1D vector). When compared with an FC network, a CNN involves significantly fewer model parameters that need to be learned considering inputs of the same size, substantially reducing the computational cost for model training and inference.

Pooling layers are useful for reducing the dimensions of the data, the number of parameters to learn, and consequently the computational cost. In this study, the MaxPooling function is used, which selects the maximum value within a specified region of data after the convolution. Pooling also make models more robust to positional variations in the data.

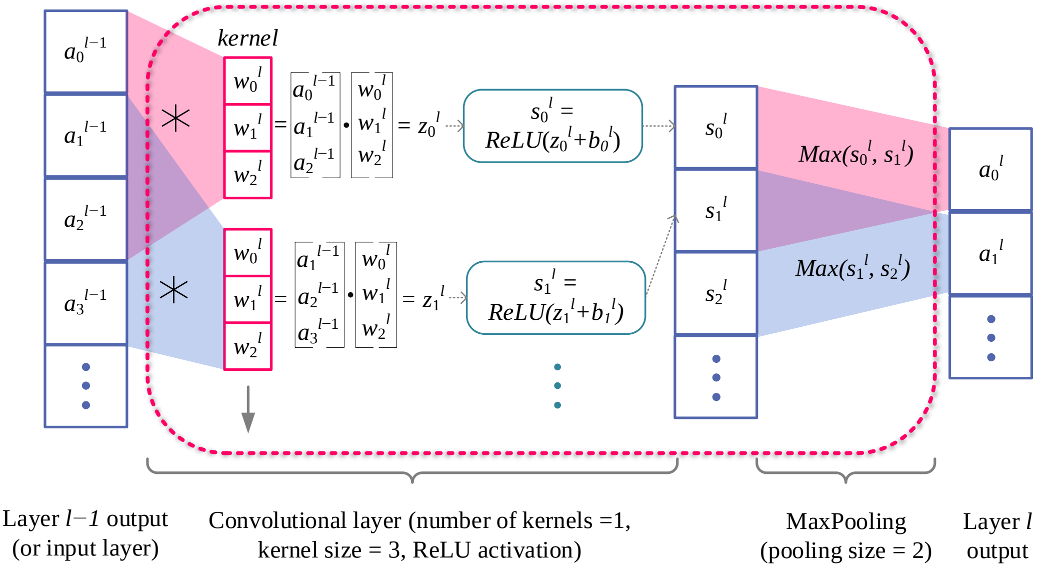

Figure 2 Components for a one-dimensional convolutional neural network. a_i^{l-1} is the output i in layer l−1. w_i^l represents the i^{th} weight in a kernel vector (also known as filter) in layer l. The symbol ✶ denotes the convolution operation. The result of convolution operation (=z_i^l) along with a bias term b_i^l is passed into the non-linear activation function, outputting s_i^l. Both w_i^l and b_i^l are parameters that need to be learned for layer l.

A key process in the development of a deep learning model is designing the neural network architecture and selecting the hyperparameters of the model, such as the network's depth (number of layers), width (number of nodes/kernels), and the type of layers. Deeper and wider models are more effective in identifying complex patterns in the dataset, but it is more computationally costly to train the large number of parameters in these more complex models.

If the model's performance on the training set continues to improve while its performance on the validation set starts to deteriorate, it is a sign of overfitting. Training should be stopped. Complex models are more prone to overfitting the training data, leading to poor model generalization on new unseen data that is different from the training data. Regularization techniques (e.g., dropout, L2 regularization and batch normalization) can be used to reduce the chance of overfitting and improve the model's generalization ability. Simple models, on the other hand, are easier to train and optimize, but they might not be able to capture the more complex patterns in a dataset. Often, the development of an ideal model requires repetitive experimentation guided by monitoring the model errors during training and validating.

Figure 3 shows the architecture of the hybrid neural network designed in this study. The hybrid model consists of two threads of neural network layers. In one thread, convolutional layers followed by three FC layers are used to capture the patterns (feature extraction: peaks and valleys that represent strong and weak soil layers) in the CPT cone resistance profile. In the other thread, FC layers are used to capture the impact of the pile geometry (i.e., area moment of inertia I_p and pile length L) and loading condition (load eccentricity h) on the lateral capacity of monopiles. Near the end of the model, the two threads are merged into one using FC layers to capture the impact of the soil-pile interaction on the lateral load response of monopiles. The final outputs of the model are the lateral pile capacities $H_{0.5}$ and H_1 corresponding to pile rotation θ=0.5° and θ=1°, respectively, at the mudline.

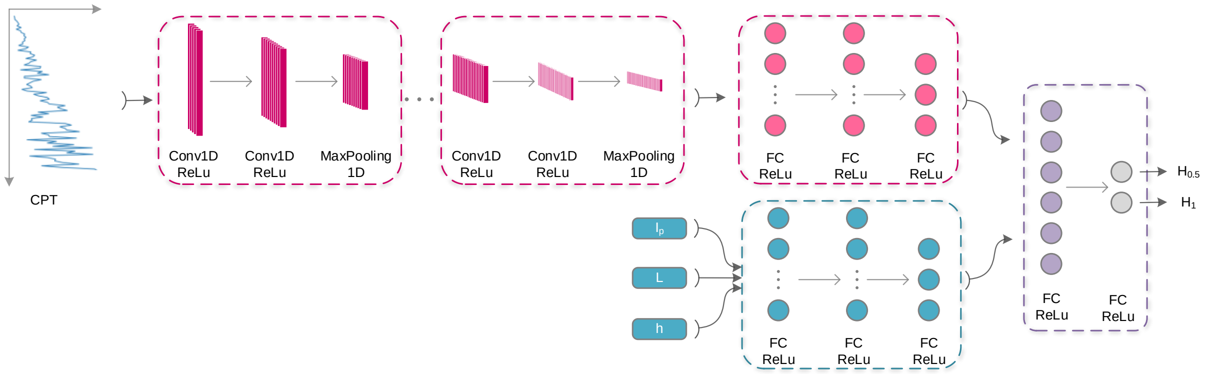

Figure 3 The architecture of the hybrid neural network model. Convolutional blocks are incorporated to extract patterns in the CPT cone resistance profile. Each convolutional block contains two one-dimensional convolutional followed by a one-dimensional pooling layer. In addition to the CPT data, three geometric attributes of the pile and loading condition are introduced to the model with three fully connected layers. In the end, two branches are combined to generate the lateral capacities $H_{0.5}$ and H_1.

2.2. Data generation and processing

2.2.1. Data generation

As a data-driven method, a deep learning model often requires a large amount of training data to learn the complex relationships between input and output variables (e.g. pile geometry, soil profile, loading condition and the lateral pile capacities). Yet, data is limited for full-scale lateral load tests on large-diameter monopiles accompanied by full site characterization (Byrne et al. 2019; Spill et al. 2017). Alternatively, we can rely on data generated from high-quality 3D finite element (FE) analyses. To obtain a sufficient amount of data for training the proposed DL model, we used the 3D FE analysis results from Hu et al. (2021, 2022). In their study, high-fidelity 3D FE analyses were performed in Abaqus Explicit (ABAQUS 2014) to model the response of laterally loaded monopiles in multi-layered sandy soil. As shown in Figure 4, only half of the soil-pile domain was modeled in the FE analyses due to the symmetric nature of the boundary value problem. Hexahedral linear elements with reduced integration (i.e., C3D8R elements) were used in the FE model. To minimize the boundary effect, the width of the soil domain was set at 20 times the pile diameter, and the thickness was set at twice the pile length. Given the large pile diameters of monopiles, fully-coring mode was assumed, and thus both the soil inside and outside the pipe pile was modeled. The pile-soil interface was modeled following the perfect-contact approach, where the common nodes of the soil and pile are tied to each other. This was chosen due to the negligible difference in lateral load response of monopiles modeled by the perfect-contact approach and the contact-pair approach, as demonstrated by Hu et al. (2021).

A two-surface-plasticity sand model developed by Loukidis and Salgado (2009) was used in the analyses. Developed under a critical-state soil mechanics framework, the constitutive model closely captures mechanical behavior of sand under various stress paths. The model was calibrated against elemental test results (e.g., triaxial compression, triaxial extension, and simple shear) for Ottawa sand and Toyoura sand. With the properly-prepared mesh (Figure 4) and the realistic constitutive model, the FE analyses are able to accurately capture the stress-path dependent soil behavior, strain localization in soil, and the soil-pile interactions. The analyses were performed under fully drained condition using the effective stress approach given the relatively small loading rate for sandy soil (Han et al. 2017). Furthermore, the strain localization is controlled by the mesh size: the minimal mesh size is needs to be comparable to the shear band thickness expected in the sand. This technique has been used and verified by Loukidis and Salgado (2008) and Han et al. (2017).

To validate the 3D FE analyses, Hu et al. (2021) and Hu et al. (2022) compared the results obtained from their FE analyses with those obtained from the full-scale lateral pile load tests performed as a part of the Pile Soil Analysis (PISA) project (Byrne et al. 2019; McAdam et al. 2019). The pile load test was performed on a 10.5-m-long, 2-m-diameter open-ended steel pipe pile in medium dense to dense marine sand. The predicted and measured load–deflection curves at the mudline are in very close agreement.

The dataset contains a large number (100000) of data points (also known as samples), each consisting of the input values (i.e., area moment of inertia for the pile cross section, pile length, CPT q_c profile, and load eccentricity) and the corresponding outputs (i.e., lateral pile capacities $H_{0.5}$ and H_1 corresponding to pile rotation θ=0.5° and θ=1° at the mudline). In this research, CPT profile is directly used in the model instead of D_R for two reasons. First, to use a D_R-based model (or other soil property-based models), in-situ test results (such as q_c) must be converted to D_R. This conversion may introduce errors originating from factors such as K_0, unit weight, water table, and friction angle. A CPT-based model can bypass the calculation of D_R from q_c and directly take q_c profile as the model input. Furthermore, a CPT-based model is convenient for the users to apply, where they can directly input the CPT results into the model to obtain predictions for the lateral pile capacities without the need to estimate or assume soil properties. The CPT cone resistance profile is obtained using the Salgado and Prezzi (2007) equation developed based on cavity expansion theory:

$$q_c = 1.64 p_A \exp[0.1041\phi_c + (0.0264 - 0.0002\phi_c) D_R] \left(\frac{\sigma'_h}{p_A}\right)^{0.841-0.0047 D_R} \qquad \text{Eq. 1}$$

where p_A = reference stress = 100kPa, ϕ_c = critical-state friction angle in sand, D_R = relative density in sand (D_R=60 for 60% relative density), and σ'_h = horizontal effective stress. Eq. 1 has been tested in many prior studies (Han et al. 2017, 2019; Sakleshpur et al. 2021), and it provides q_c values that are consistent to those estimated using the NGI method (Clausen et al. 2005):

$$q_c = \exp(D_R/0.4)[22(\sigma'_v p_A)^{0.5}] \qquad \text{Eq. 2}$$

where D_R is expressed as fractions (e.g. D_R=0.6 for 60% relative density), and σ'_v = vertical effective stress.

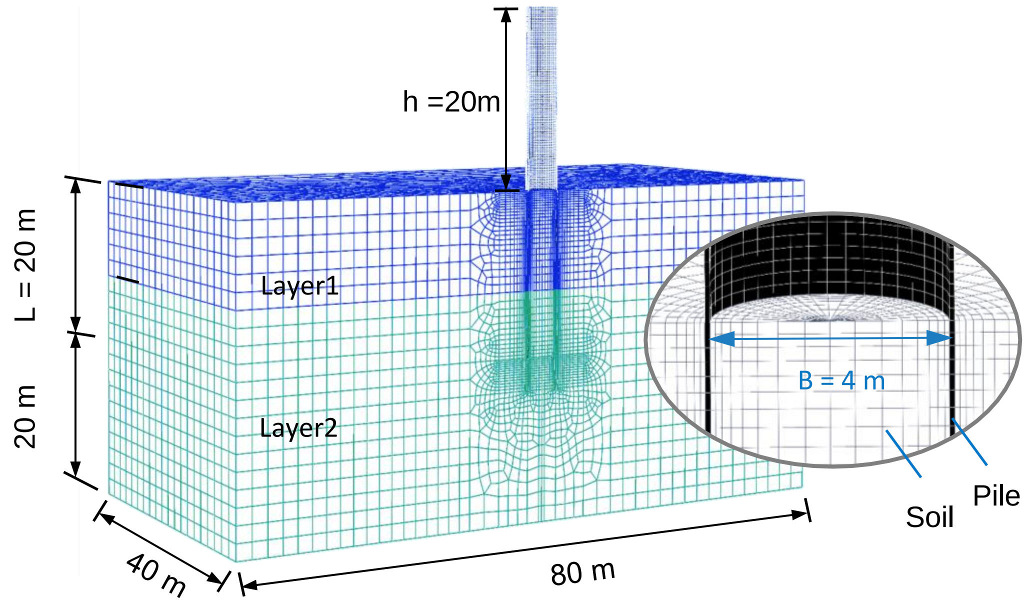

Figure 4 Finite element (FE) analyses of laterally loaded monopiles (modified after Hu et al. 2022)

In order to obtain a robust DL model that is applicable to common design scenarios, the training samples should cover an extensive range of values for the input variables. Therefore, we consider broad-ranging values for pile length (6 m to 60 m), pile diameter (2 m to 10 m), load eccentricity (15 m to 30 m), pile diameter-to-wall-thickness ratio (40 to 100), and the relative density (35% to 90%) for the multi-layered sandy soils. All raw data used in this study is published in an open-access data repository (details provided in the data availability section).

Since the CPT data (in the format of 1D vectors) vary in size depending on the pile length, we used Zero Padding (i.e., appending zeros to the end of the original data) to convert all CPT data into the same size. This is done because CNN accepts data of the same size as the input. This technique does not affect the predictions as the soil profile in the top layers controls the behavior of laterally loaded piles. Alternatively, Recurrent Neural Network (RNN), which is often used to deal with sequential data, can be used to handle input data of different sizes.

2.2.2. Data transformation

When generating the training samples, we maintained a uniform distribution for the input variables. For example, a pile diameter within the range from 2 m to 10 m is randomly selected for each sample. However, the uniform distribution for the input variables results in a highly non-symmetric, non-uniform distribution for the output values (i.e., pile capacities). In statistics, symmetry of a distribution about its mean can be measured by skewness $\bar{\mu}_3$ (e.g., skewness = 0 represents a completely symmetric distribution):

$$\bar{\mu}_3 = \mathbb{E}\left[\left(\frac{x-\mu}{\sigma}\right)^3\right] \qquad \text{Eq. 3}$$

where x is the value of a data point in the distribution, μ is the mean of the distribution, and σ is the standard deviation of the distribution. As shown in Figure 5a and Figure 5b, the distribution of the outputs (i.e., pile capacities) for the training dataset is positively skewed (the distribution is concentrated toward the left side). There is significantly larger amount of data for piles with smaller lateral capacities, leading to better model performance and higher prediction accuracies for these piles. Conversely, piles with larger lateral capacities have less data for model training, leading to greater prediction errors for those piles. To reduce the dataset skewness and improve overall model performance, we will explore the effectiveness of two transformation methods (Atkinson et al. 2021):

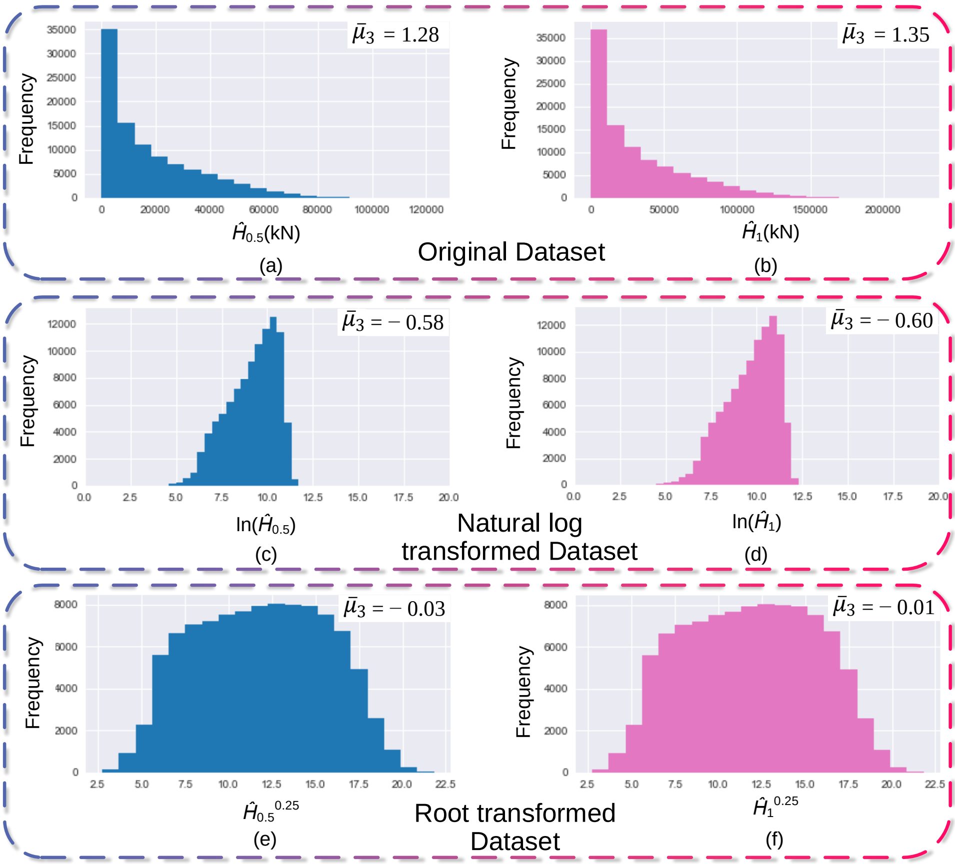

Figure 5 Histograms of the outputs and transformed outputs in the generated dataset: (a) The distribution of the lateral capacity $\hat{H}_{0.5}$ corresponding to θ=0.5°; (b) The distribution of the lateral capacity $\hat{H}_1$ corresponding to θ=1°; (c) The distribution of the intermediate output ln($\hat{H}_{0.5}$) after the natural logarithm transformation. (d) The distribution of the intermediate output ln($\hat{H}_1$) after the natural logarithm transformation. (e) The distribution of the intermediate output $\hat{H}_{0.5}^{0.25}$ after the root transformation. (f) The distribution of the intermediate output $\hat{H}_1^{0.25}$ after the root transformation.

Natural log transformation

The natural log transformation is one of the most widely used methods to reduce skewness of a dataset when it is positively skewed, which is the case for the outputs ($\hat{H}_{0.5}$ and $\hat{H}_1$) in our dataset. Thus, we introduce a pair of intermediate outputs $\hat{Y}_{0.5}$ = ln($\hat{H}_{0.5}$) and $\hat{Y}_1$ = ln($\hat{H}_1$) that are used to train the DL model. Note that $\hat{H}$ and $\hat{Y}$ refer to the ground-truth values whereas H and Y refer to the predicted values. As seen in Figure 5c and Figure 5d, the natural log transformation reduced the absolute value of skewness ($|\bar{\mu}_3|$) for the original dataset by more than 60% from about 1.3 to 0.6. After DL model training was done, the predicted intermediate outputs were transformed back to their original form to provide predictions for pile capacities: $H_{0.5} = \exp(Y_{0.5})$ and $H_1 = \exp(Y_1)$.

Root transformation

Another widely-used transformation method for positively skewed data is root-transformation. In this study, the fourth root transformation was implemented to convert the original outputs (pile capacities $\hat{H}_{0.5}$ and $\hat{H}_1$) into intermediate outputs $\hat{Y}_{0.5} = (\hat{H}_{0.5})^{0.25}$ and $\hat{Y}_1 = (\hat{H}_1)^{0.25}$ for model training. The root transformation was successful in bringing the skewness of the original outputs down to almost zero (Figure 5e and Figure 5f). After the DL model was trained, these intermediate outputs were transferred back to their original form: $H_{0.5} = (Y_{0.5})^4$ and $H_1 = (Y_1)^4$.

2.2.3. Input normalization

When training a DL model, the learning algorithm (e.g., Gradient Descent) iteratively updates the model parameters such that the prediction error is minimized. When the ranges for the input variables are significantly different (e.g., one input ranges from 1 to 2, whereas another input ranges from 10 to 10,000), the learning algorithm becomes slow or unstable, sometimes causing the learning to fail (Bishop 1995). To resolve this issue, input variables are often scaled (e.g., using normalization or standardization) into the same or similar ranges before they are used for training. In this paper, each input variable (X) is normalized with respect to its maximum and minimum values (X_{max} and X_{min}) using the MinMax normalization function:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad \text{Eq. 4}$$

After the normalization, all scaled input variables X_{scaled} fall between 0 and 1.

2.2.4. *Training, validation, and test sets*

In the development of a DL model, the full dataset is typically split into three separate subsets (i.e., training, validation, and test sets) that are used for different purposes. The largest portion of the full dataset is used to train the model parameters (e.g., the weights and biases shown in Figure 1) in the DL model, hence the name training set. Then, a small portion of the full dataset (separate from the training set), known as the validation set, is used to evaluate the model performance to avoid overfitting and underfitting issues. In case of unsatisfactory model performance, the hyperparameters, such as the learning rate, number of layers, and number of nodes in each layer, are adjusted, and the model is trained again using the training set. These two steps are repeated iteratively until satisfactory model performance is obtained for both the training and validation datasets. Finally, the trained model is assessed against the test dataset that has never been seen by the model. In this project, the training, validation, and test sets contain 90%, 5%, and 5% of the full dataset, respectively.

The hybrid neural network was implemented using Python 3.9.7 programming language with Pytorch 1.10.2 as the DL framework. The training was conducted on a computer with the CPU of Intel Core-i9-10920X, Memory of 128 GB, and GPU of NVidia RTX A5000. It's worth pointing out that the model's training process was time consuming, taking approximately 20 hours. However, after the training is finished, generating a prediction from input values using the trained model takes only a fraction of a second.

3 RESULTS AND DISCUSSION

3.1. Results

A DL model is trained to minimize errors in predicting the outputs. The model error is quantified using a cost function, which captures the average error between the ground truth and prediction for all samples. Proper selection of the cost function is one of the keys to obtaining a good DL model. Among all commonly-seen cost functions, Mean Squared Error (MSE) is the most widely used for regression problems:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad \text{Eq. 5}$$

where n is the number of samples, y_i is the predicted output for the ith sample, and $\hat{y}_i$ is the corresponding ground truth. Each cost function has its own limitations and strengths. MSE has a convex topology, which makes the optimization process (i.e., training) to minimize the cost function more reliable (Aravkin et al. 2014). However, a model trained with MSE equally treats the absolute error (difference between the predicted and ground-truth outputs) for each sample regardless of the magnitude of the output. This is problematic when the output variable spans a wide range of values, as is the case in this study. For example, an error of 100 kN is a 0.1% relative error for a ground-truth pile capacity of 100,000 kN, whereas the same error (=100 kN) is a 100% relative error for a pile capacity of 100 kN.

To solve the aforementioned issue, we can instead use the Mean Absolute Percentage Error (MAPE), which calculates the relative error (percentage error) between the predicted output y and the ground-truth output ŷ:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{\hat{y}_i} \times 100 \qquad \text{Eq. 6}$$

MAPE is often used in practice due to its intuitive interpretation. However, optimizing MAPE is more computationally challenging than MSE because of its non-convex topology and non-differentiability (Chen et al. 2017; De Myttenaere et al. 2016).

To investigate the effectiveness of different data transformation techniques combined with different cost functions on the model performance, we used the original dataset, natural log transformed dataset, and root transformed dataset to train the hybrid neural network model using either MAPE or MSE as the cost function. The model was trained for 600 epochs (the number of passes that the algorithm works through the entire training dataset) with a batch size (the number of training samples the algorithm works through before updating the model parameters) of 128 and a learning rate (the step size taken to adjust the parameters with respect to model's error) of 0.001. The learning algorithm is ADAM (Kingma and Ba 2014), which is an extension of stochastic gradient descent. ADAM is more efficient in training neural networks with non-convex cost functions.

Among the six trained models (i.e., three data transformation techniques combined with two cost functions), the one trained using natural log transformation and MSE cost function performs the best, providing consistent predictions with an average relative error MAPE = 2.67% for the model outputs. Figure 6 compares the predicted pile capacities $H_{0.5}$ and H_1 (for pile rotation $\theta=0.5°$ and $\theta=1°$) versus the corresponding ground truths $\hat{H}_{0.5}$ and $\hat{H}_1$ for this model. Data points are closely located near the perfect-prediction line, indicating overall good model performance. We will compare results of the six models in detail in the next section.

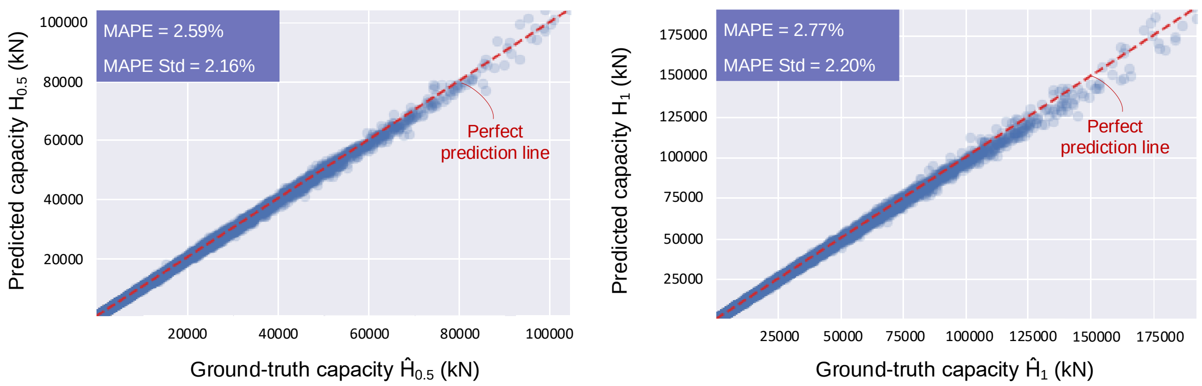

Figure 6 Performance of the model trained with natural log transformed data and MSE cost function: the predicted pile capacities vs. the ground-truth pile capacities.

3.2. **Error Analysis**

In this section, we analyze errors for each of the six models trained with different data transformation techniques and cost functions. Figure 7 shows the relative error $[(H-\hat{H})/\hat{H}]$ for the predicted pile capacity as a function of the ground-truth value for all data points in the test set. In addition, we use the concept of Confidence Interval (68% CI and 95% CI) to show the probability that the relative error falls within a certain range of values. It is ideal to have the mean relative error close to zero and narrow CIs around it, while a wide CI for a certain range of pile capacity H reflects unreliable model predictions for that range of H. Since the model performances in predicting $H_{0.5}$ and H_1 are similar (Figure 6), we show the error analysis for $H_{0.5}$ as an example.

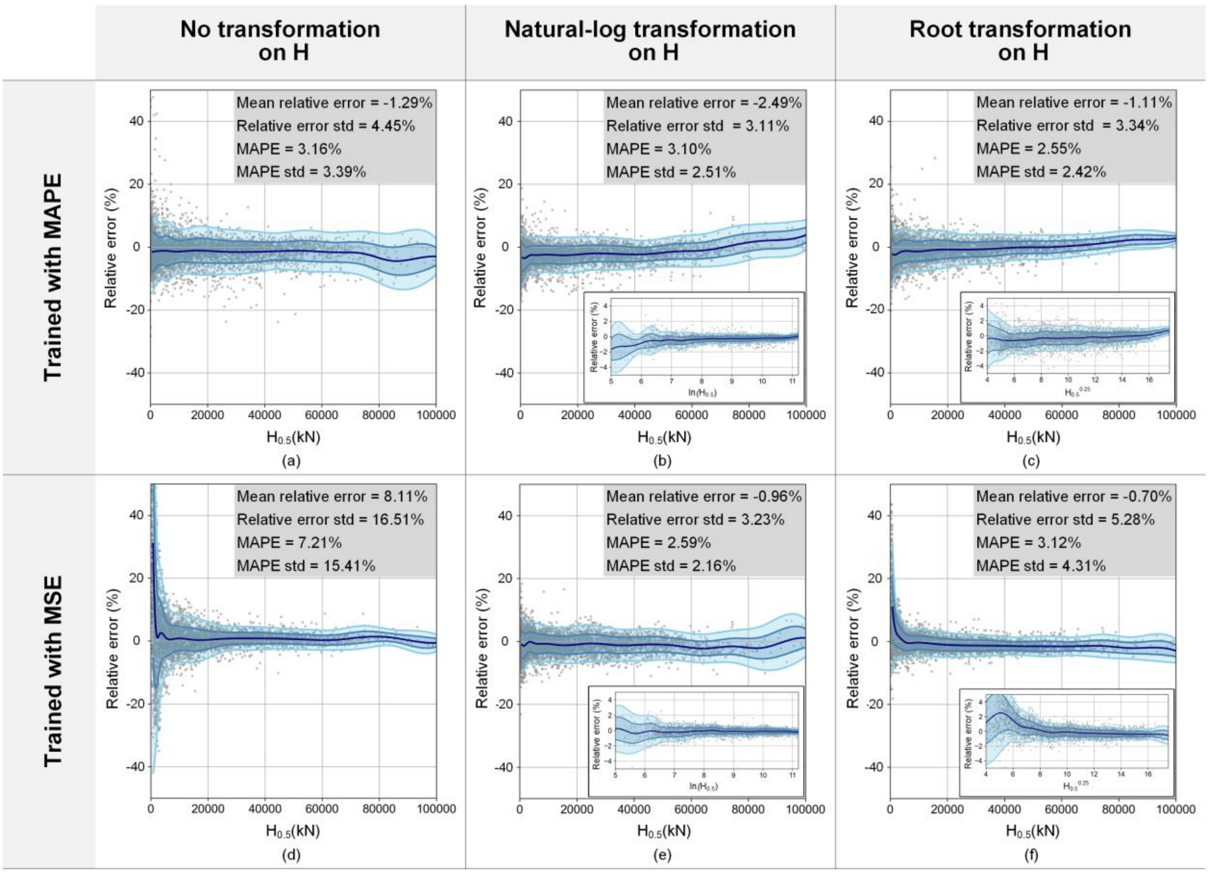

Figure 7 The mean, 68%, and 95% confidence intervals for the models' relative errors in predicting pile capacity $H_{0.5}$: (a) Relative error for the model trained with MAPE using the dataset without transformation; (b) Relative error for the model trained with MAPE using natural logarithm transformed data; (c) Relative error for the model trained with MAPE using root transformed data; (d) Relative error for the model trained with MSE using the dataset without transformation; (e) Relative error for the model trained with MSE using natural logarithm transformed data; (f) Relative error for the model trained with MSE using root transformed data. Note that the relative error for the intermediate outputs [i.e., $\ln(H_{0.5})$ and $H_{0.5}^{0.25}$] data are presented in the sub-figures.

With no data transformation on the pile capacity H, the model trained with MAPE cost function results in consistent relative errors for the entire range of pile capacities (Figure 7a), whereas training with MSE causes significant relative errors (wider CI) for smaller pile capacities (Figure 7d). Since using the MSE cost function minimizes the squared error [SE = $(H - \hat{H})^2$] between the predictions and ground truths, a model trained with MSE tends to produce predictions for H with consistent SE values across the entire range of H. The consistent SE values result in absolute percentage errors (APE) that are sensitive to the magnitude of $\hat{H}$:

$$MAPE = |H - \hat{H}|/\hat{H} = \sqrt{SE}/\hat{H} \quad \text{Eq. 7}$$

Thus, model trained with MSE without any data transformation tends to have larger relative error (MAPE) for piles with smaller capacity values $\hat{H}$, and vice versa (Figure 7d). This is particularly true when the range of the output values is large, such is the case in this study (the pile capacity H ranges from about 100 kN to about 100,000 kN).

Figure 7b and Figure 7e compare the performance of the two models trained with natural log transformation using MSE vs MAPE cost functions. After the natural logarithm transformation, the range for the intermediate output ln($\hat{H}$) shrinks drastically from the range for the original target output $\hat{H}$ (Figure 5). The mean decreases from 18633 for $\hat{H}_{0.5}$ to 9.14 for the intermediate output ln($\hat{H}_{0.5}$), and the associated standard deviation (std) decreases from 18621 for $\hat{H}_{0.5}$ to 1.37 for ln($\hat{H}_{0.5}$). According to Eq. 7, when the ground-truth output [i.e., the intermediate output $\hat{Y}$=ln($\hat{H}$)] for a model has a small range of values, training the model with either MSE or MAPE cost function does not make a significant difference in the model performance for predicting the intermediate output Y (sub figures in Figure 7b and Figure 7e). This eventually results in similar model performances in predicting the ground-truth $\hat{H}$ [transformed back from Y to H=exp(Y)], as shown in Figure 7b and Figure 7e.

Root transformation also reduces the range for output values (Figure 5). The mean decreases from 18633 for $\hat{H}_{0.5}$ to 10.39 for the intermediate output $\hat{Y}= \hat{H}_{0.5}^{0.25}$, and the associated standard deviation (std) decreases from 18621 for $\hat{H}_{0.5}$ to 3.23 for $\hat{Y}= \hat{H}_{0.5}^{0.25}$. Yet, root transformation does not reduce the output range as much as natural logarithm transformation does (Figure 5). Consequently, we see greater relative errors (i.e., wider CIs) for smaller intermediate outputs ($\hat{H}^{0.25}$) for the model trained with MSE cost function (sub figure in Figure 7f), leading to the large relative errors for smaller pile capacities $\hat{H}$ (Figure 7f).

Among the six models considered in this study, the model trained with MSE using natural logarithm transformed data is the most reliable one with MAPE = 2.59% and MAPE std = 2.16%. The model trained with MAPE using root transformation has a slightly smaller MAPE value (=2.55%) yet a greater std value (MAPE std = 2.42%).

3.3. **Which cost function and data transformation methods to choose?**

We often deal with geotechnical data that has a wide range of values with large data skewness due to the inherent variability and nonlinearity in the underlying geotechnical problems (e.g., pile capacities, footing settlement, and slope deformation). When developing a deep learning model to solve these problems, it is crucial to choose an appropriate combination of data transformation method and cost function to tackle the extensive range and skewed nature of the data. In this section, we derive theoretical errors and relative errors expected from using different data transformation techniques and cost functions.

Table 1 summarizes how an error E (or relative error E_r) associated with the intermediate output Y (at the end of training) is transformed to be the error (or relative error) for the target variable H. For example, a relative error E_r for the intermediate output Y=ln(H) (i.e., natural log transformation) results in a relative error of $\hat{H}^{E_r}$-1 for the target output H. This means when the model is trained with MAPE cost function, which ideally tends to produce consistent relative error E_r for the whole range of intermediate output Y, the relative error (=$\hat{H}^{E_r}$-1) for the target output H becomes dependent on the value of $\hat{H}$: The relative error for $\hat{H}$ tends to increase with increasing $\hat{H}$. As shown in Table 1, among the four combinations of data transformation methods and cost functions, training the model with natural log transformation in conjunction with MSE cost function or training with root transformation in conjunction with MAPE cost function will result in consistent relative errors for the target output H that is independent of the value of H (or $\hat{H}$).

Ideally, a model trained with the MSE cost function tends to produce predictions with zero-mean randomly distributed errors (i.e., mean of errors=0). Similarly, a model trained with MAPE cost functions tends to produce predictions with zero-mean randomly distributed relative errors (i.e., mean of relative errors=0). In Figure 8, we further demonstrate how errors or relative errors for the intermediate output Y propagate into the relative error for the target output H. We assume a zero-mean evenly-distributed error or relative error for the intermediate output Y depending on the training cost functions. The second row of sub-figures shows the corresponding relative errors for the target variable H after the intermediate variable Y is transformed to H. As suggested by Table 1, training with two specific combinations of cost function and data transformation (MSE with log transformation and MAPE with root transformation) leads to consistent relative errors for the target output H. This is reflected in Figure 8 (bottom-left and bottom right sub-figures) as randomly distributed zero-mean relative errors that are independent of the value of H. In contrast, Training the model with log transformation and MAPE loss function causes the relative error for the target output H to increase with increasing value of H. Training the model with root transformation and MSE loss function results in decreasing relative errors for the target output H as H increases. These two combinations should be avoided in data treatment and model training to prevent inconsistent relative errors for predictions depending on the output value.

Table 1 Error analysis for different data transformation methods (for model outputs) trained with different cost functions

Type of transformation	Natural log	Natural log	Root	Root
Intermediate output $\hat{Y}$ after transformation	$\hat{Y} = \ln(\hat{H})$	$\hat{Y} = \ln(\hat{H})$	$\hat{Y} = \hat{H}^{\alpha}$	$\hat{Y} = \hat{H}^{\alpha}$
Cost function	MSE	MAPE	MSE	MAPE
Error to be minimized*	$E^2 = (Y-\hat{Y})^2$	$\lvert E_r\rvert = \lvert Y-\hat{Y}\rvert/\hat{Y}$	$E^2 = (Y-\hat{Y})^2$	$\lvert E_r\rvert = \lvert Y-\hat{Y}\rvert/\hat{Y}$
Predicted intermediate output Y	$Y=\hat{Y}+E$	$Y=\hat{Y}(1+ E_r)$	$Y=\hat{Y}+E$	$Y=\hat{Y}(1+ E_r)$
Transformation back to target output H	$H=e^{Y}= e^{\hat{Y}+E}$	$H= e^{Y} = e^{\hat{Y}(1+ Er)}$	$H=Y^{(1/\alpha)} = (\hat{Y}+E)^{(1/\alpha)}$	$H=Y^{(1/\alpha)} =[\hat{Y}(1+ E_r)]^{(1/\alpha)}$
Error for target output H (= H-$\hat{H}$)	$e^{\hat{Y}+E}- e^{\hat{Y}}$	$e^{\hat{Y}(1+ Er)}- e^{\hat{Y}}$	$(\hat{Y}+E)^{(1/\alpha)}-(\hat{Y})^{(1/\alpha)}$	$[\hat{Y}(1+ E_r)]^{(1/\alpha)}-\hat{Y}^{(1/\alpha)}$
Relative error for target output H [=(H-$\hat{H}$)/$\hat{H}$]	$e^{E}-1$**	$\hat{H}^{Er}-1$	$(1+E/\hat{H}^{\alpha})^{(1/\alpha)}-1$	$(1+ E_r)^{(1/\alpha)}-1$**

*E is the error for the predicted intermediate output Y: $E=Y-\hat{Y}$; E_r is the relative error for the predicted intermediate output Y: $E_r = (Y-\hat{Y})/\hat{Y}$.

**These two combinations of cost function and data transformation result in consistent relative errors for the target output H that is independent of the value of H (or $\hat{H}$).

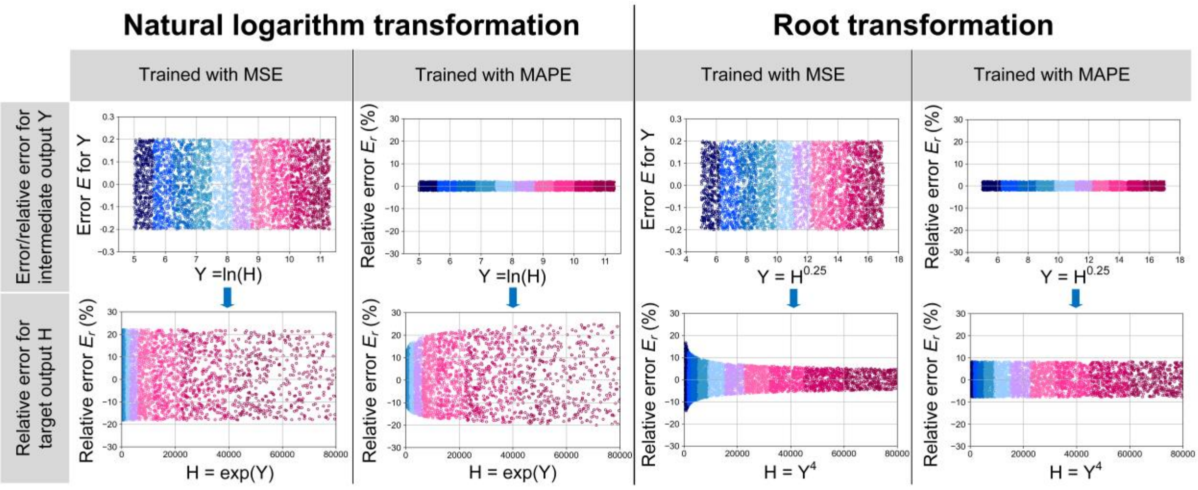

Figure 8- The propagation of error/relative error for the intermediate output Y to the relative error for the target output H. Random error E ranging from −0.2 to 0.2 is assumed for the intermediate output Y when training with the MSE loss function. Random relative error E_r ranging from −2% to 2% is assumed for the intermediate output Y when training with the MAPE loss function. Data points marked with the same color are associated with the same samples before and after data transformation.

3.4. Sensitivity analysis for the q_c profile

To evaluate the susceptibility of the proposed ANN model to noise in the CPT data, a sensitivity analysis was done for a specific case characterized by a pile diameter of 8.5 m, a length of 25.43 m, and a load eccentricity of 26.5 m. The influence of the CPT data noise on the model performance was assessed by creating 1000 variations of the qc profile. This was achieved by introducing random noise ranging from -10% to +10% into the baseline CPT data (Figure 9a). The lateral capacities H_1 predicted based on these 1000 q_c profiles follows a normal distribution (Figure 9b) with a remarkably low standard deviation (=0.6% of the mean value). This means that 68% of the predictions have relative errors less than 0.6%. This example shows the robustness of the proposed ANN model to the noise and uncertainties in the CPT data. The robustness of the ANN model is due to both the wide range of diverse data used for model training and the data augmentation technique used for data preparation (by adding noise to the q_c profiles in training data).

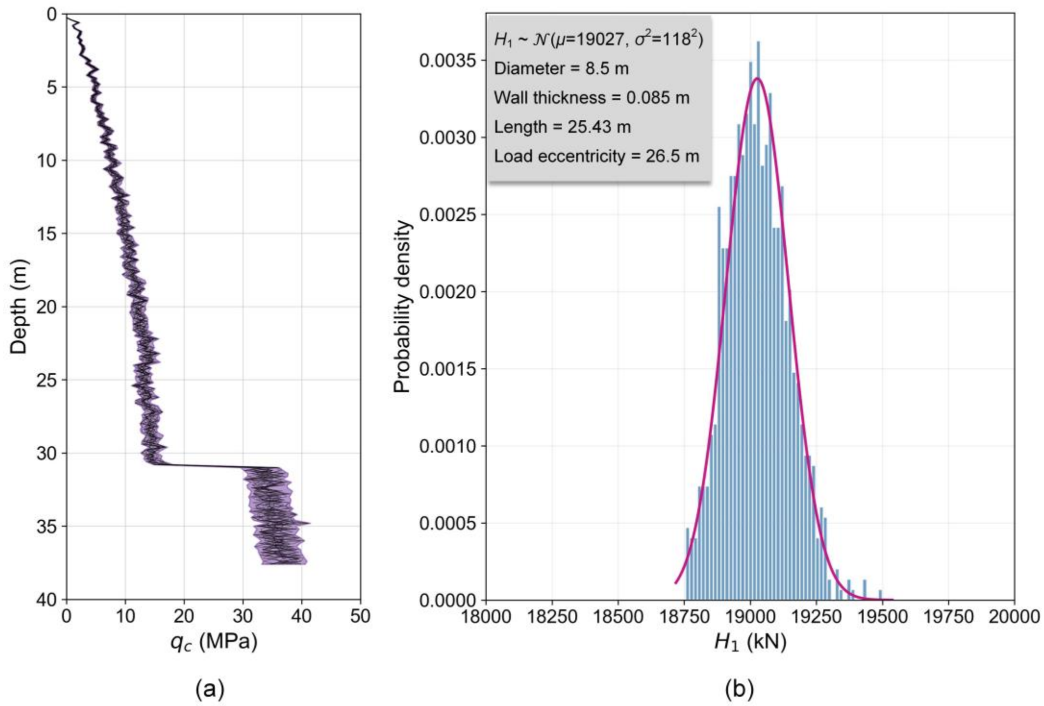

Figure 9- Sensitivity of the predicted capacity H_1 to the noise in CPT data (a) CPT with random relative noise ranging from -10% to +10%. (b) distribution of the pile lateral capacities predicted based on the noisy CPT data

3.5. **Training histories**

When tuning and evaluating a DL model, two model behaviors should be avoided: underfitting and overfitting. Underfitting means that the model cannot capture patterns in the dataset or identify the relationships between the inputs and outputs; this is reflected in large prediction errors for the training set. Overfitting, on the other hand, refers to the case when a model can produce good predictions for the training set, but it does not generalize well to the validation set, which is unseen by the model during training (Goodfellow et al. 2016; Patterson and Gibson 2017). Underfitting occurs when the DL model is too simple (i.e., with a small number of layers and nodes) or is not sufficiently trained (too few epochs), while too sophisticated model or exceedingly training a model may lead to overfitting.

Figure 10 shows the error histories for the model during the training. The MAPE error is calculated at the end of each epoch for both the training and validation datasets. As training continues, the model parameters keep being updated (by the ADAM training algorithm), and the prediction error continues to decrease. As shown in Figure 10, the errors for the training set and the validation set decrease following the same trend, suggesting that the model generalizes well to the validation set. Hypothetically, if the error for the validation set were significantly greater than that for the training set from the beginning of training, the model could be overfitted due to too complex model. If the error histories for the two sets first followed the same trend until they diverged at some point, that would suggest overfitting due to the model being overtrained beyond the divergence point.

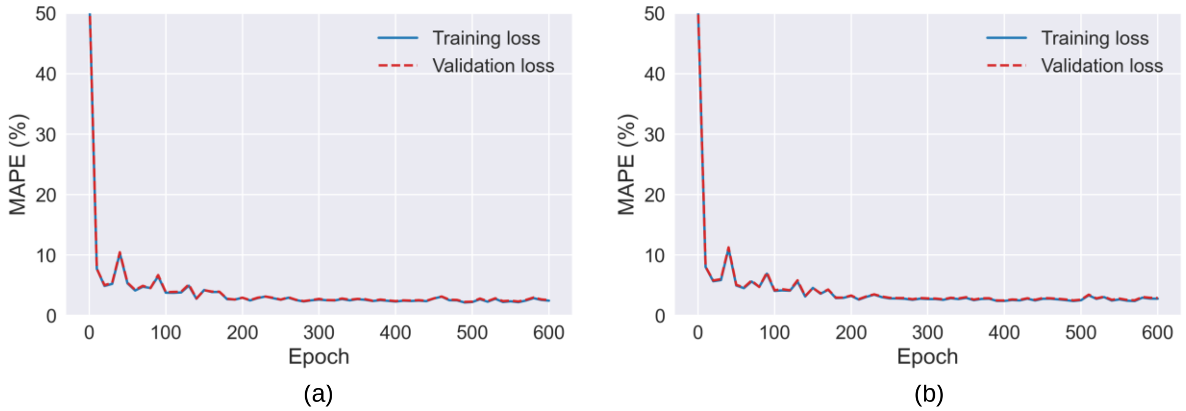

Figure 10 The loss history of learning and validation for the model trained with MSE using natural log transformation. (a) The loss history for $H_{0.5}$ (b) The loss history for H_1.

3.6. **Design example**

In order to demonstrate the performance of the developed Hybrid neural network model, we present a design example from Hu et al. (2022). A pipe pile with a diameter of 5 m, wall thickness of 5 cm, and length of 35 m is laterally loaded in multi-layered sandy soil (Figure 11a). The lateral load is applied at a height of 15 m from the ground surface (load eccentricity h = 15m). The soil is fully saturated with the water table located at the ground surface. Figure 11(b) shows the CPT cone resistance q_c profile as a function of depth. The pile geometries, load eccentricity and the CPT q_c profile were fed into the hybrid neural network model, producing predictions for the pile lateral capacities $H_{0.5}$ = 7858 kN and H_1 = 13442 kN corresponding to θ=0.5° and θ=1°. These two values of pile rotation were chosen because 1) they represent the serviceability limit states of the pile, and 2) they can be used to develop the entire load-rotation (H-θ) response for this monopile using (Eq. 8) proposed by Hu et al. (2022b):

$$H = \frac{\theta}{k + \eta\theta}$$

$$\begin{cases} \eta = 2/H_1 - 1/H_{0.5} \\ k = 1/H_{0.5} - 1/H_1 \end{cases} \qquad \text{(Eq. 8)}$$

For comparison purposes, the p-y analysis was performed for this pile using the web-based application [Lateral Analysis of Piles (LAP)] (Doherty 2017). A three-layer soil profile (Figure 11a) was considered in the p-y analysis using the API sand p-y curves. The 3D FE analysis was performed following the simulation setup detailed in Hu et al. (2022). As shown in Figure 11b, the proposed DL model provides accurate predictions for the lateral load response of the monopile. The p-y analysis, which was originally developed for long and slender piles, significantly overestimates the lateral capacities of the pile (Figure 11c).

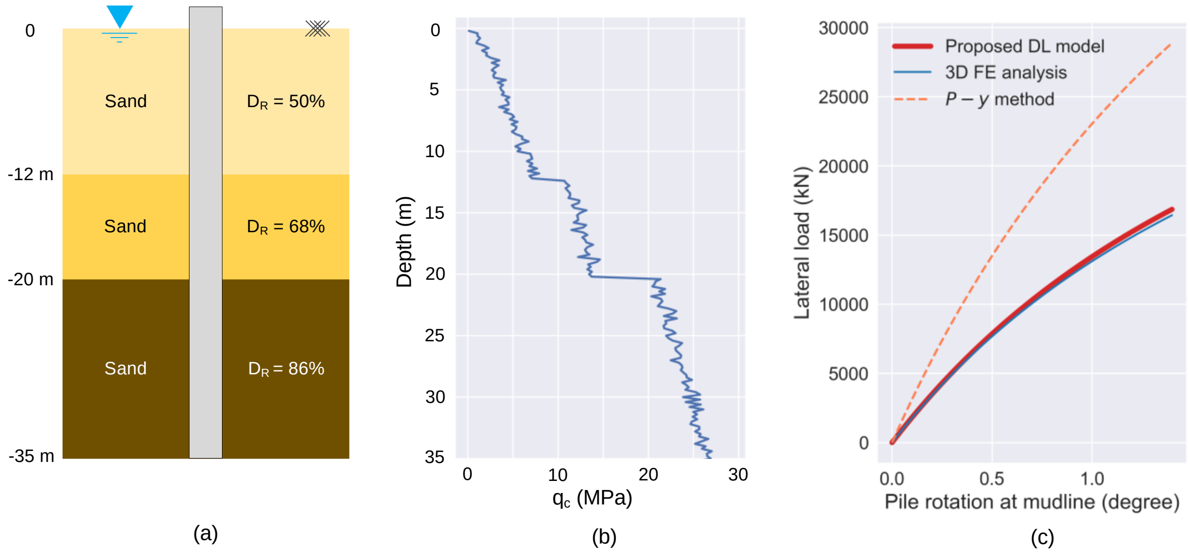

Figure 11 (a) Soil profile and the corresponding relative densities and thicknesses; (b) The CPT cone resistance profile; (c) The load rotation response curves obtained from the 3D FE model, DL model, and the p-y analysis. The p-y analysis overestimates the lateral pile capacity by 75%; the DL model could predict the capacity well with a relative error of 2.45%.

4 PILE DESIGN OPTIMIZATION

Given its ability to deliver rapid and precise predictions, the proposed ANN model can function as a surrogate model, approximating complex FE models. This allows for the execution of computation-heavy procedures like design optimization or system modeling that necessitate the simulation of numerous design instances. Surrogate models substantially cut down the computational cost of optimization, paving the way for more rapid design iterations and aiding in pinpointing the most effective solutions. To demonstrate a practical application of the trained surrogate model, we conduct optimization for a design example aiming to minimize the material cost for the monopile whiling satisfying a set of constraints, including specified ranges for slenderness ratio, wall thickness ratio, and the pile diameter as well as meeting the required lateral capacity. The optimization problem is formulated as:

$$\textit{Min.}\quad \rho \pi B t_w L$$

Subject to:

(1) $H_1 \geq \hat{H}_1$

(2) $3 \leq L/B \leq 15$

(3) $40 \leq B/t_w \leq 100$

(4) $2 \leq B \leq 10$

where B, t_w, and L are the pile diameter, wall thickness, and length, respectively. H_1 is the lateral capacity of the optimized solution, and $\hat{H}_1$ is the desired capacity. One specific design example was selected for optimization. The initial pile dimensions are B = 6.87 m, t_w = 0.15 m, L = 31 m, load eccentricity h = 26.5 m, and the corresponding lateral capacity $\hat{H}_1$=18007 kN. The initial weight of the pile is 787,824 kg. The Sequential Quadratic Programming (SQP) algorithm was used to optimize the pile design. The SQP algorithm is an iterative optimization used to solve nonlinear constrained optimization problems. In each iteration, it constructs a quadratic approximation of the objective function and a linear approximation of the constraints to form a Quadratic Programming subproblem. The solution to this subproblem then provides a search direction for an iterative line search procedure, and this process repeats until the convergence criteria are met or the maximum number of iterations is reached. The SQP algorithm takes about 10 minutes to output the optimized solution: B = 8.5 m, t_w = 0.085 m, and L = 25.43 m. This optimized pile design provides the same lateral capacity (18007 kN) as the original design, but the weight of the optimized pile is 450,676 kg, which is 42% less than that for the original design. Figure 12 shows the history of the pile weight and the corresponding pile capacity during the optimization process.

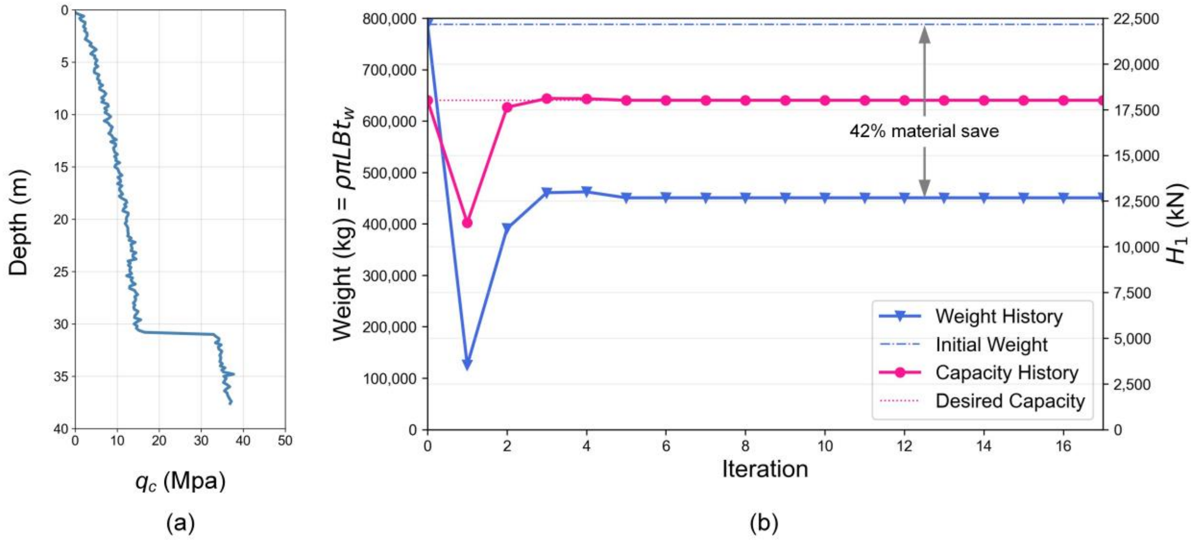

Figure 12- (a) CPT q_c profile considered for the pile design optimization example. (b) Solution history for the pile design optimization with the objective to reduce the material cost for the monopile while providing the required lateral capacity.

5 CONCLUSIONS AND DISCUSSIONS

A hybrid neural network model was developed to provide fast and accurate predictions of the lateral capacities for large-diameter monopiles. The neural network contains a series of convolutional layers that captures the soil behavior (via CPT cone resistance data) and fully-connected layers that accounts for the impact of pile geometry, load eccentricity and the pile-soil interactions on the lateral pile load response. To train the model, synthetic data were generated based on validated 3D finite element analyses covering a wide range of design scenarios. The developed neural network model is able to provide accurate capacity predictions (mean average error = 2.68%). The developed ANN model was then integrated into Sequential Quadratic Programming (SQP) to optimize pile design, minimizing material cost (by 42%) while satisfying the capacity requirement.

Highly skewed distribution of the output variables in the training dataset adversely affects the performance of deep learning models. The natural logarithm transformation and root transformation techniques can effectively reduce the skewness of the output distribution. These data transformation techniques need to be used in pairs with specific training cost functions to achieve the best model performance: The natural logarithm transformation should be used with the MSE cost function, whereas the root transformation should be used with the MAPE cost function to provide predictions with consistent relative percentage errors that are independent of the model output values. This is particularly useful when the output spans a large range of values.

The proposed neural network model has many advantages when compared with high-fidelity 3D FE model: 1) The proposed model can generate predictions with high accuracy at a much faster rate compared to a 3D FE model. While a 3D FE analysis may take 2-7 days depending on the model size (larger piles taking longer runtime due to more elements), the proposed model can generate predictions in just a fraction of a second. The proposed model is an excellent surrogate for the 3D FE model, enabling large-scale system-level modeling, optimization, and resilience analysis; 2) The proposed method is data-driven, and it eliminates the need for users to possess specialized knowledge, such as meshing and constitutive model, which are crucial for performing high-quality FE analyses; and 3) Trained with a large dataset that covers broad-ranging values for design inputs, the developed model is versatile and adaptable. It can serve as the base model that can be easily adapted for specific design scenarios (e.g., a special site condition) with only a small amount of data using the transfer learning technique.

The model can be further strengthened in terms of robustness and reliability by training with more diversified soil types (e.g., over-consolidated soils) and pile types (e.g., concrete piles). This hybrid neural network framework can also be easily extended to solve other geotechnical problems (e.g., axially loaded piles and shallow foundations).

6 DATA AVAILABILITY

The raw dataset used to train the proposed hybrid neural network is published at the open-access Zenodo data repository (https://doi.org/10.5281/zenodo.7675229).

NOTATION

L	pile length
B	outer diameter of pile
t_{w}	wall thickness of pile
I_p	area moment of inertia of pile
h	load eccentricity
D_{R}	relative density of sand
θ	pile rotation at the mudline
$\hat{H}$	ground-truth pile lateral capacity
H	estimated pile lateral capacity
$\hat{H}_{0.5}$	ground-truth pile lateral capacity corresponding to a pile rotation $\theta = 0.5°$ at mudline
$H_{0.5}$	estimated pile lateral capacity corresponding to a pile rotation $\theta = 0.5°$ at mudline
$\hat{H}_1$	ground-truth pile lateral capacity corresponding to a pile rotation $\theta = 1°$ at mudline
H_1	estimated pile lateral capacity corresponding to a pile rotation $\theta = 1°$ at mudline
$\bar{\mu}_3$	skewness of the distribution

7 REFERENCES

ABAQUS. 2014. “ABAQUS 6.14-4.” Abaqus Analysis User’s Manual (RI, USA: SIMULIA Inc).

Adeel, M. B., M. A. Jan, M. Aaqib, and D. Park. 2021. “Development of Simulation Based p-Multipliers for Laterally Loaded Pile Groups in Granular Soil Using 3D Nonlinear Finite Element Model.” Appl. Sci., 11 (1): 26. Multidisciplinary Digital Publishing Institute. https://doi.org/10.3390/app11010026.

Ahmadi, M. M., and S. Ahmari. 2009. “Finite-element modelling of laterally loaded piles in clay.” Proc. Inst. Civ. Eng. - Geotech. Eng., 162 (3): 151–163. ICE Publishing. https://doi.org/10.1680/geng.2009.162.3.151.

Aravkin, A., J. V Burke, A. Chiuso, and G. Pillonetto. 2014. “Convex vs non-convex estimators for regression and sparse estimation: the mean squared error properties of ARD and GLasso.” J. Mach. Learn. Res., 15 (1): 217–252. JMLR. org.

Atkinson, A. C., M. Riani, and A. Corbellini. 2021. “The Box–Cox Transformation: Review and Extensions.” Stat. Sci., 36 (2): 239–255. Institute of Mathematical Statistics.

Birba, D. E. 2020. “A Comparative study of data splitting algorithms for machine learning model selection.”

Brown, D. A., and C.-F. Shie. 1990. “Three dimensional finite element model of laterally loaded piles.” Comput. Geotech., 10 (1): 59–79. https://doi.org/10.1016/0266-352X(90)90008-J.

Bishop, C. M. 1995. Neural networks for pattern recognition. Oxford university press.

Byrne, B. W., H. J. Burd, L. Zdravković, R. A. McAdam, D. M. G. Taborda, G. T. Houlsby, R. J. Jardine, C. M. Martin, D. M. Potts, and K. G. Gavin. 2019. “PISA: new design methods for offshore wind turbine monopiles." Rev. Française Géotechnique, (158): 3. EDP Sciences.

Byrne, B. W., Houlsby, G. T., Burd, H. J., Gavin, K. G., Igoe, D. J., Jardine, R. J., ... & Zdravković, L. (2020). PISA design model for monopiles for offshore wind turbines: application to a stiff glacial clay till. Géotechnique, 70(11), 1030-1047.

Chan, C. L., and B. K. Low. 2012. "Probabilistic analysis of laterally loaded piles using response surface and neural network approaches." Comput. Geotech., 43: 101–110. Elsevier.

Chen, C., J. Twycross, and J. M. Garibaldi. 2017. "A new accuracy measure based on bounded relative error for time series forecasting." PLoS One, 12 (3): e0174202.

Chatterjee, K., D. Choudhury, and H. G. Poulos. 2015. "Seismic analysis of laterally loaded pile under influence of vertical loading using finite element method." Comput. Geotech., 67: 172–186. https://doi.org/10.1016/j.compgeo.2015.03.004.

Chen, L., and H. G. Poulos. 1993. "Analysis of pile-soil interaction under lateral loading using infinite and finite elements." Comput. Geotech., 15 (4): 189–220. https://doi.org/10.1016/0266-352X(93)90001-N.

Cheng, X., A. Diambra, E. Ibraim, H. Liu, and F. Pisanò. 2021. "3D FE-informed laboratory soil testing for the design of offshore wind turbine monopiles." J. Mar. Sci. Eng., 9 (1): 101. MDPI.

De Myttenaere, A., B. Golden, B. Le Grand, and F. Rossi. 2016. "Mean absolute percentage error for regression models." Neurocomputing, 192: 38–48. Elsevier.

Doherty, P., and K. Gavin. 2012. "Laterally loaded monopile design for offshore wind farms." Proc. Inst. Civ. Eng. Energy, 165 (1): 7–17. ICE Publishing.

Doherty, J. P. (2017, June). A web-based application for the lateral analysis of pile (LAP) foundations. In International Conference on Offshore Mechanics and Arctic Engineering (Vol. 57779, p. V009T10A012). American Society of Mechanical Engineers.

Farahani, A., M. Samadzad, and R. Rafiee-Dehkharghani. 2022. “Vibration mitigation of pile groups by coupling a wave-based vibration analysis and genetic algorithm.” Taylor & Francis.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. Deep learning. MIT press.

Han, F., M. Prezzi, and R. Salgado. 2017. “Energy-based solutions for nondisplacement piles subjected to lateral loads.” Int. J. Geomech., 17 (11): 04017104.

Han, F., R. Salgado, and M. Prezzi. 2015. “Nonlinear analyses of laterally loaded piles – A semi-analytical approach.” Comput. Geotech., 70: 116–129. Elsevier.

Han, F., R. Salgado, M. Prezzi, and J. Lim. 2017. “Shaft and base resistance of non-displacement piles in sand.” Computers and Geotechnics, 83: 184–197.

Han, F., E. Ganju, R. Salgado, and M. Prezzi. 2019. “Comparison of the load response of closed-ended and open-ended pipe piles driven in gravelly sand.” Acta Geotechnica, 14 (6): 1785–1803. https://doi.org/10.1007/s11440-019-00863-1.

Hu, Q., F. Han, M. Prezzi, R. Salgado, and M. Zhao. 2021. “Lateral load response of large-diameter monopiles in sand.” Géotechnique, 1–16.

Hu, Q., F. Han, M. Prezzi, R. Salgado, and M. Zhao. 2022. “Finite-Element Analysis of the Lateral Load Response of Monopiles in Layered Sand.” J. Geotech. Geoenvironmental Eng., 148 (4): 04022001. American Society of Civil Engineers.

Huang, Y., Z. He, A. Yashima, Z. Chen, and C. Li. 2022. “Multi-objective optimization design of pile-anchor structures for slopes based on reliability theory considering the spatial variability of soil properties.” Comput. Geotech., 147: 104751. Elsevier.

Kardani, N., A. Zhou, M. Nazem, and S. L. Shen. 2020. “Estimation of Bearing Capacity of Piles in Cohesionless Soil Using Optimised Machine Learning Approaches.” Geotech. Geol. Eng., 38 (2): 2271–2291. Springer.

Kingma, D. P., and J. Ba. 2014. “Adam: A method for stochastic optimization.” arXiv Prepr. arXiv1412.6980.

Kementzetzidis, E., W. G. Versteijlen, A. Nernheim, and F. Pisano. 2018. “3D FE dynamic modelling of offshore wind turbines in sand: natural frequency evolution in the pre– to after-storm transition.” Numer. Methods Geotech. Eng. IX, 2: 1477–1484. CRC Press.

Kordjazi, A., F. P. Nejad, and M. B. Jaksa. 2014. “Prediction of ultimate axial load-carrying capacity of piles using a support vector machine based on CPT data.” Comput. Geotech., 55: 91–102. Elsevier.

Lai, Z., Q. Chen, and L. Huang. 2022. “Machine-learning-enabled discrete element method: Contact detection and resolution of irregular-shaped particles.” Int. J. Numer. Anal. Methods Geomech., 46 (1)

Loukidis, D., and R. Salgado. 2009. “Modeling sand response using two-surface plasticity.” Comput. Geotech., 36 (1–2): 166–186.

Makasis, N., G. A. Narsilio, and A. Bidarmaghz. 2018. “A machine learning approach to energy pile design.” Comput. Geotech., 97: 189–203. Elsevier.

McAdam, R. A., B. W. Byrne, G. T. Houlsby, W. J. A. P. Beuckelaers, H. J. Burd, K. G. Gavin, D. J. P. Igoe, R. J. Jardine, C. M. Martin, A. Muir Wood, D. M. Potts, J. Skov Gretlund, D. M. G. Taborda, and L. Zdravković. 2019. “Monotonic laterally loaded pile testing in a dense marine sand at Dunkirk.” Géotechnique, (July): 1–13.

McAdam, R. A. (2020). Finite-element modelling of laterally loaded piles in a dense marine sand at Dunkirk. Géotechnique, 70(11), 1014-1029.

McCelloch, W. S., and W. Pitts. 1943. “A logical calculus of the idea immanent in neural nets.” Bull. of Mathematical Biophys., 5: 115–133.

Moeinifard, P., M. S. Rajabi, and M. Bitaraf. 2022. “Lost vibration test data recovery using convolutional neural network: a case study.” arXiv Prepr. arXiv2204.05440.

Murphy, G., D. Igoe, P. Doherty, and K. Gavin. 2018. “3D FEM approach for laterally loaded monopile design.” Comput. Geotech., 100: 76–83. HTTPS://DOI.ORG/10.1016/J.COMPGEO.2018.03.013.

Nguyen, Q. H., H.-B. Ly, L. S. Ho, N. Al-Ansari, H. Van Le, V. Q. Tran, I. Prakash, and B. T. Pham. 2021. “Influence of data splitting on performance of machine learning models in prediction of shear strength of soil.” Math. Probl. Eng., 2021: 1–15. Hindawi Limited.

Patterson, J., and A. Gibson. 2017. Deep learning: A practitioner’s approach. “ O’Reilly Media, Inc.”

Peng, J.-R., M. Rouainia, and B. G. Clarke. 2010. “Finite element analysis of laterally loaded fin piles.” Comput. Struct., 88 (21): 1239–1247. https://doi.org/10.1016/j.compstruc.2010.07.002.

Pham, T. A., V. Q. Tran, H. L. T. Vu, and H. B. Ly. 2020. “Design deep neural network architecture using a genetic algorithm for estimation of pile bearing capacity.” PLoS One, 15 (12): e0243030. Public Library of Science.

Sakleshpur, V. A., M. Prezzi, R. Salgado, and M. Zaheer. 2022. CPT-Based Geotechnical Design Manual, Volume 2: CPT-Based Design of Foundations—Methods. Purdue University.

Salgado, R., & Prezzi, M. (2007). Computation of cavity expansion pressure and penetration resistance in sands. International Journal of Geomechanics, 7(4), 251-265.

Shahin, M. A. 2016. “State-of-the-art review of some artificial intelligence applications in pile foundations.” Geosci. Front., 7 (1): 33–44. Elsevier.

Spill, S., M. Kohlmeier, M. Wefer, S. Maretzki, and J. Dührkop. 2017. “Design of large-scale tests investigating the lateral load-bearing behavior of monopiles.” Proc. Int. Offshore Polar Eng. Conf., 579–586.

Suryasentana, S. K., and B. M. Lehane. 2016. “Updated CPT-based p–y formulation for laterally loaded piles in cohesionless soil under static loading.” Geotechnique, 66 (6): 445–453. ICE Publishing.

Taborda, D. M. G., L. Zdravkovic, D. M. Potts, H. J. Burd, B. W. Byrne, K. G. Gavin, G. T. Houlsby, R. J. Jardine, T. Liu, C. M. Martin, and R. A. McAdam. 2020. “Finite-element modelling of laterally loaded piles in a dense marine sand at Dunkirk.” Geotechnique, 70 (11): 1014–1029. https://doi.org/10.1680/jgeot.18.PISA.006.

Tan, C., F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. 2018. “A Survey on Deep Transfer Learning.” Artif. Neural Netw. Mach. Learn. – ICANN 2018, Lecture Notes in Computer Science, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, eds., 270–279. Cham: Springer International Publishing.

Wakai, A., S. Gose, and K. Ugai. 1999. “3-D Elasto-Plastic Finite Element Analyses of Pile Foundations Subjected to Lateral Loading.” Soils Found., 39 (1): 97–111. https://doi.org/10.3208/sandf.39.97.

Xu, Y., and R. Goodacre. 2018. “On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning.” J. Anal. Test., 2 (3): 249–262. Springer.

Xiao, T., L. M. Zhang, R. W. M. Cheung, and S. Lacasse. 2022. “Predicting spatio-temporal man-made slope failures induced by rainfall in Hong Kong using machine learning techniques.” Geotechnique.

Xu, L.-Y., F. Cai, G.-X. Wang, and K. Ugai. 2013. “Nonlinear analysis of laterally loaded single piles in sand using modified strain wedge model.” Comput. Geotech., 51: 60–71. Elsevier Ltd.

Zhang, J. Z., D. M. Zhang, H. W. Huang, K. K. Phoon, C. Tang, and G. Li. 2022. “Hybrid machine learning model with random field and limited CPT data to quantify horizontal scale of fluctuation of soil spatial variability.” Acta Geotech., 17 (4): 1129–1145. Springer Berlin Heidelberg.

8 APPENDIX

Table 2- Parameters and their ranges for the training data; i represents the layer number (3 layers of sand). SI units are used for parameters

Parameter	Symbol	Parameter value, distribution, or formula
Pile diameter (m)	B	~ Uniform (2, 8)
Pile length (m)	L	$= 3B + 12B \times$ Uniform (0,1)
Wall thickness ratio	B/t_w	~ Uniform (40, 100)
Inner diameter (m)	ID	$= B - 2 \times$ wall thickness
Area moment of inertia (m^4)	I_p	$= \pi \times (B^4 - ID^4) / 64$
Load eccentricity (m)	h	~ Uniform (15, 30)
Thickness of layer 1 (m)	t_1	$= 0.5B + (L-0.5B) \times$ Uniform (0,1)
Thickness of layer 2 (m)	t_2	$= (L - t_1) \times$ Uniform (0,1)
Thickness of layer 3 (m)	t_3	Total soil domain thickness-t_1-t_2
Relative density (%)	D_R	~ Uniform (35, 90)
Effective unit weight (kN/m^3)	γ'	$= (G_s-1)\gamma_w/(1 + e)$
Vertical effective stress, calculated each 0.2 m in depth (kPa)	σ'_v	Calculated cumulatively from γ'
Horizontal effective stress, calculated each 0.2 m in depth (kPa)	σ'_h	$= K_0 \times \sigma'_v$
CPT cone resistance (MPa)	q_c	Calculated using Eq. 1