COMPARISON OF MACHINE LEARNING MODELS IN A DATA-DRIVEN APPROACH FOR SCALABLE AND ADAPTIVE DESIGN OF LATERALLY-LOADED MONOPILE FOUNDATIONS

Suryasentana, S. K.[1], Burd, H. J.[1], Byrne, B. W.[1], Aghakouchak, A.[2] & Sørensen, T.[3]

[1]University of Oxford, Oxford, UK
[2]Ørsted Wind Power, London, UK
[3]Ørsted Wind Power, Copenhagen, Denmark

ABSTRACT

The design of monopile foundations under lateral loading is typically conducted using the Winkler modelling approach. Design models incorporating this approach include the '*p-y*' method specified in the design codes or the more recent 'PISA' design model. These design models have predefined soil reactions that are appropriate only for the pile and soil characteristics considered in the calibration. One of the biggest uncertainties is thus defining the appropriate soil reactions for design soils with different characteristics from the calibration cases. This paper proposes a data-driven approach, in which machine learning models are used to determine the most appropriate soil reactions directly from data. As these models can automatically adapt and increase in complexity to new data, it provides for a more scalable approach than existing predefined design models. Three machine learning models (polynomial regression, neural network regression (NN) and Gaussian Progress (GP) regression) are compared. The results demonstrate that the polynomial regression model is the simplest to interpret but is the least accurate. The NN regression model has high accuracy, but is the hardest to interpret. The GP regression model is the most accurate and is reasonably easy to interpret. Moreover, the GP regression model has the additional benefit of providing intrinsic uncertainty estimates, making it appropriate for quantifying and reducing design risks.

Keywords: PISA, Winkler, laterally loaded monopile, data-driven, machine learning, neural network, Gaussian process regression

INTRODUCTION

The monopile foundation is the dominant foundation choice for offshore wind developments in shallow waters. These foundations are typically designed using simplified one dimensional (1D) models such as the '*p-y*' method and its variants, where the foundation is modelled as an embedded beam and the lateral soil response on the pile is represented by using Winkler-type lateral soil reactions (or 'springs') called p-y curves (API 2010). The recently completed PISA (PIle Soil Analysis) project (Byrne et al. 2019) employed field testing and computational modelling to develop a new design approach for monopile foundations for offshore wind applications. In this new approach, four distinct Winkler-type soil reactions are included in the PISA 1D design model (see Fig. 1), consisting of distributed lateral soil reactions p, distributed moment soil reactions m, base horizontal force H_B and base moment M_B.

Two implementations of the PISA design model have been proposed; the 'rule-based' method and the 'numerical-based' method. The rule-based method employs predefined functions or 'rules' that represent soil reaction curves for certain calibration soil profiles; this approach therefore cannot be used reliably for ground conditions that differ significantly from the calibration cases. The rule-based method is intended principally for preliminary design. The numerical-based approach employs bespoke soil reaction curves that have been calibrated using three dimensional (3D) finite element calculations based on site-specific soil profiles and

laboratory element testing data. This approach therefore uses soil reactions that better correspond to the site conditions being analysed, allowing for more accurate predictions of the monopile behavior; it is therefore suited to detailed design analyses.

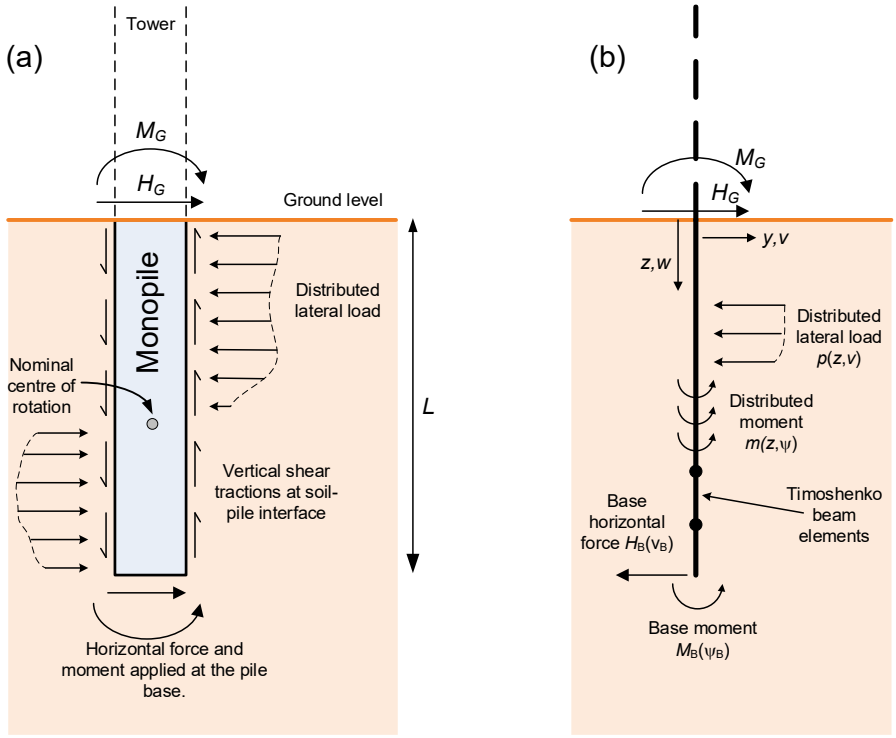

Fig. 1. PISA monopile model (a) assumed soil reactions acting on monopile (b) 1D design model. (Byrne et al. 2020)

Although the rule-based approach is fast and straightforward to apply, its applicability depends on the similarity between the current site of interest and the representative sites employed in the calibration process. The numerical-based approach, while capable of delivering high fidelity analyses, requires a significant investment of resources to conduct the bespoke 3D finite element calibration analyses; thus making it a more time-consuming approach.

In the current paper, a new implementation of the PISA design model – termed 'data-driven approach' – is proposed. This new approach provides the benefits of the efficiency of the rule-based approach and the accuracy of the numerical-based approach. It also offers advantages such as high scalability, as it is able to automatically accommodate new soil profiles. This supports the development of more optimal foundation designs, especially in the earlier design phases.

The data-driven approach employs machine learning models to determine the most appropriate soil reaction curve parameters directly from data. To explore this new approach, the paper compares three machine learning models (polynomial regression, neural network regression (NN) and Gaussian Progress (GP) regression); these machine learning models are assessed in terms of accuracy and interpretability. Based on this assessment, the most promising model is then suggested for the data-driven approach.

DATA-DRIVEN APPROACH

The PISA design model is specified in detail in Byrne et al. (2020) and Burd et al. (2020); key features are briefly reviewed here. In the PISA design approach, a set of 3D finite element analyses is carried out based on selected (typically idealised) soil profiles for a range of

monopile geometries. The soil reactions are then extracted from the 3D finite element results and parameterised using a 4-parameter $(\bar{y}_u, \bar{x}_u, k, n)$ function (see Fig. 2), where $\bar{y}_u$ is the maximum value for the normalised soil reaction, $\bar{x}_u$ refers to the normalised displacement variable at which $\bar{y}_u$ is attained, k is the initial stiffness of the soil reaction curve and n is the curvature.

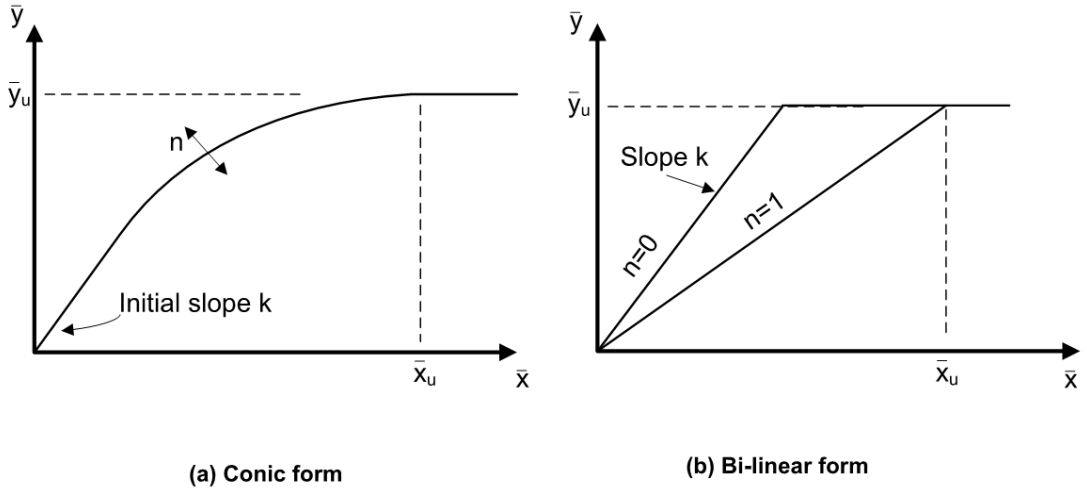

Fig. 2: 4-parameter function to represent a generic soil reaction curve (Byrne et al. 2020)

Suitable rules for these parameters are then identified to generalise the soil reactions for other monopile geometries (within the parameter space). These rules form the basis of the rule-based approach. The process of identifying these rules is typically a time-consuming, manual process but it is a one-time process; subsequent design calculations for different monopile geometries can be done very quickly. This makes the rule-based approach a very efficient modelling procedure.

However, the rule-based approach is appropriate only if the soil profiles at the sites of interest are similar to the soil profiles corresponding to the rules. Otherwise, the numerical-based approach should be used. The numerical-based approach requires running a series of bespoke 3D finite element analyses based on the actual site conditions, as determined by the site investigation and laboratory test data. Site-specific soil reactions are then extracted from the 3D finite element results and parameterised using the 4-parameter function, which can then be used for monopile design calculations. This process may be carried out using commercial software such as PLAXIS MoDeTo (Panagoulias et al. 2018). As these site-specific soil reactions correspond to the actual site conditions, the monopile design calculations will likely be more accurate and reliable; however, the need to carry out 3D finite element analyses to obtain these site-specific soil reactions adds to the complexity of this design approach.

The data-driven approach can be considered as a 'bridge' between the rule-based approach and the numerical-based approach, as it attempts to combine the efficiency of the rule-based approach and the accuracy of the numerical-based approach. The rule-based approach is governed by rules that have been manually crafted; the data-driven approach uses machine learning models that can automatically learn the rules directly from the data (which are obtained from the numerical-based approach), allowing it to accommodate and adapt to new soil profiles with ease. Typically, these models can learn the rules more reliably than the manual process.

The data-driven approach employs a database of inputs and outputs from previous 3D finite element analyses. For each soil profile, the inputs are *z/D* and *L/D*, where *z* is the depth (below ground level) of the soil reaction, *L* is the embedded monopile length and *D* is the monopile outer diameter. The outputs are the 4 parameters (indicated in Fig. 2) for the soil reactions curves that have been extracted from the corresponding finite element results.

Then, an appropriate machine learning model is employed to learn the mapping between the inputs and outputs. As the database grows, there is less chance of encountering a soil profile that is significantly different from the existing soil profiles stored in the database and thus, the need to run bespoke 3D finite element analyses reduces. Fig. 3 shows the links between these three approaches; the data-driven approach allows one to build on and accumulate past design experiences to improve the preliminary designs of future projects.

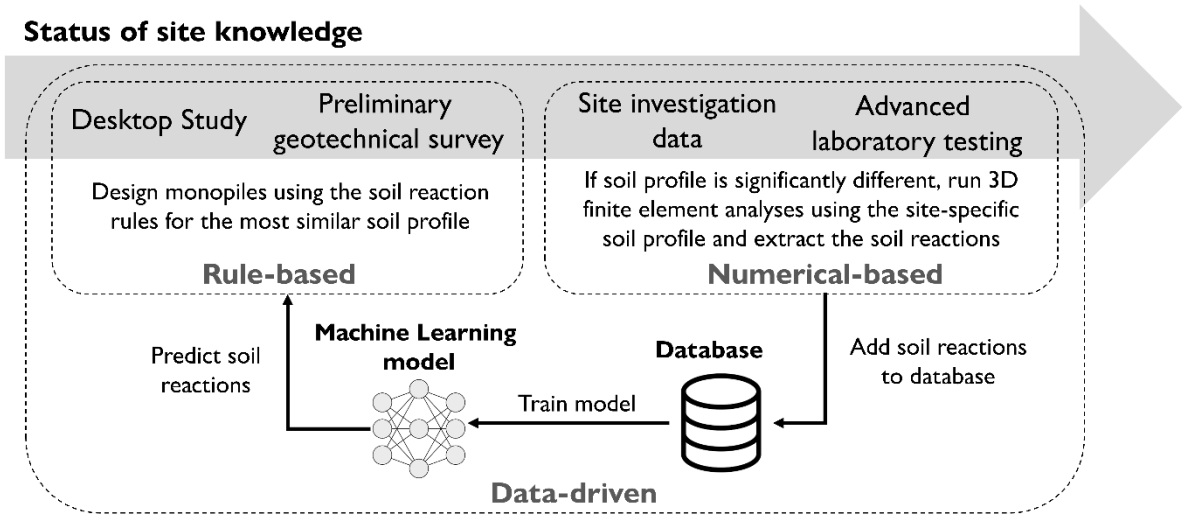

Fig. 3: Links between the PISA rule-based, numerical-based and data-driven approaches

MACHINE LEARNING MODELS

Three machine learning (ML) models are assessed for the data-driven approach: polynomial regression, neural network (NN) regression and Gaussian Process (GP) regression. For the purpose of this section, $\boldsymbol{x}$ denotes a vector of input variables and y denotes the output variable (assumed to be a scalar unless otherwise stated). Vectors and matrices are denoted using lowercase (e.g. $\boldsymbol{x}$) and uppercase (e.g. $\boldsymbol{X}$) boldface letters respectively.

Polynomial Regression

Despite being a non-linear model, polynomial regression can be considered as a special case of linear regression, as linear regression refers to 'linear in the parameters'. The non-linearity of the model comes from the non-linear transformation $\boldsymbol{\phi}(\boldsymbol{x})$ of the inputs $\boldsymbol{x}$ before applying the linear regression to the transformed inputs. This model can be formulated as:

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) + b \qquad [1]$$

where $\boldsymbol{w}$ is a vector of weights, b is a bias parameter and each component $\boldsymbol{\phi}_i(\boldsymbol{x})$ is a monomial basis function (Bishop, 2006). The optimised values of the parameters (i.e. $\boldsymbol{w}$ and b) are those that minimise the cost function (typically the residual sum-of-squares function).

While higher order polynomials will result in a better fit to the training set than lower order polynomials, it may suffer from the problem of overfitting, which occurs when the model fits the noise in the data instead of generalising. One way to resolve the overfitting problem is to apply regularisation to penalise large values of $\boldsymbol{w}$ (Bishop, 2006). This is done by adding a regularisation term to the cost function. Depending on whether the L1 or L2 norm (also known as the Manhattan or Euclidean norm respectively) is employed for the regularisation term, the regression analysis is called LASSO regression or ridge regression respectively.

Neural Network (NN) Regression

A neural network (NN) is closely related to the standard linear regression model described above. A NN model consists of: (i) the network architecture, which defines the number of layers, the number of neurons, and how the neurons are connected (ii) the parameters (weights and biases).

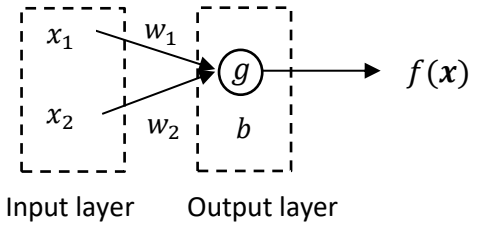

Fig. 4: A single neuron neural network mapping the inputs x to the output f(*x*), which is a (typically non-linear) transformation of the weighted inputs $\boldsymbol{f(x) = g(w^T x + b)}$.

The simplest possible NN is one that is composed of a single neuron (Fig. 4), where $f(\boldsymbol{x}) = g(\boldsymbol{w}^T\boldsymbol{x} + b)$. The terms $\boldsymbol{w}^T\boldsymbol{x} + b$ is the standard linear regression model (with monomial basis functions of degree 1) discussed above, where **w** and b are the weights and the bias of the model. g is called the activation function and is typically, but not necessarily, a non-linear transformation. Commonly used activation functions include the identity function, hyperbolic tangent function and rectified linear unit (ReLu) function. Hence, a single neuron model is just the standard linear regression model, followed by a (typically) non-linear transformation. Without non-linear activation, the NN will simply perform standard linear regression.

A more complex NN can be built by stacking together several neurons, such that one neuron passes its output as input into the next neuron, resulting in a more complex function. Typically, a NN is composed of an input layer, an output layer and a number of 'hidden layers' in between the input and output layers. Deep learning usually refers to a NN with many hidden layers. A hidden layer is called 'hidden' because there are no training values for the neurons in the hidden layers (also known as hidden units). The hidden units represent intermediate features that are automatically determined by the NN to have predictive abilities. This automated feature determination is a key attraction of the deep learning model (Goodfellow et al., 2016). However, the NN may determine features which are useful for predicting the output but are difficult to interpret. The values of the parameters (i.e. ***w*** and *b*) are taken to be those that minimise the cost function (typically the residual sum-of-squares function), which is usually done using backpropagation and gradient descent. To prevent overfitting, regularisation terms may be added to the cost function.

Gaussian Process (GP) Regression

A Gaussian Process (GP) is a generalisation of a multivariate Gaussian distribution to infinitely many variables; it is a set of random variables such that any finite number of them have a multivariate Gaussian distribution (Rasmussen and Williams, 2006). The distribution of a GP is the joint distribution of an infinite number of random variables and is thus, implicitly a probability distribution over functions (informally, an infinitely long vector is approximately a function). A random function sampled from a GP is typically denoted as:

$$f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), K(\boldsymbol{x}, \boldsymbol{x}')) \quad [2]$$

where $m(\boldsymbol{x}) = E\big(f(\boldsymbol{x})\big)$ and $K(\boldsymbol{x}, \boldsymbol{x}') = E\left(\big(f(\boldsymbol{x}) - m(\boldsymbol{x})\big)\big(f(\boldsymbol{x}') - m(\boldsymbol{x}')\big)^T\right)$ are the mean and covariance (or kernel) functions respectively. The key idea behind GP regression is that,

rather than postulating a parametric form for the function $f(\boldsymbol{x}, \boldsymbol{w}, b)$ and estimating the parameters $(\boldsymbol{w}, b)$ (as per the above ML models), $f(\boldsymbol{x})$ is assumed to be a sample from a GP.

A zero mean function $m(\boldsymbol{x}) = 0$ is usually defined as the GP is expressive enough to model the true mean using the kernel alone, especially for interpolation. A non-zero $m(\boldsymbol{x})$ is usually defined when extrapolation is required. The kernel function is chosen to express the property that, for inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$ that are similar, the corresponding outputs $y(\boldsymbol{x})$ and $y(\boldsymbol{x}')$ will be more strongly correlated than those for dissimilar inputs. The choice of an appropriate kernel is based on the 'smoothness' expected in the data. A common kernel for modelling smooth functions is the Squared Exponential (SE), also known as the Radial Basis Function, kernel:

$$K_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \theta_f^2 \exp\left(-\frac{\|x-x'\|^2}{2\theta_l^2}\right) \quad [3]$$

where θ_f and θ_l are non-negative kernel hyperparameters which scales the outputs and inputs respectively. θ_l is the characteristic length-scale of the process (i.e. 'how close' two inputs must be to influence each other significantly). The larger the length-scale, the slower the function changes, resulting in a smoother function. Another common kernel for modelling less smooth functions is the Matérn kernel:

$$K_{MA}(\boldsymbol{x}, \boldsymbol{x}') = \theta_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{x-x'}{\theta_l}\right)^{\nu} k_{\nu}\left(\sqrt{2\nu}\frac{x-x'}{\theta_l}\right) \quad [4]$$

where Γ is the gamma function, k_{ν} is the modified Bessel function of the second kind, and θ_f , θ_l and ν are non-negative kernel hyperparameters which scales the outputs, the inputs and controls the function smoothness respectively. Common choices for ν are 5/2 and 3/2. The optimal values of the kernel hyperparameters (collectively denoted as $\boldsymbol{\theta}$) may be obtained by maximising the log marginal likelihood. This operation automatically performs a trade-off between bias and variance and thus, avoids the overfitting problem.

Model Specifications

This paper aims to compare the performance of the above ML models for the PISA data-driven approach. The dataset to train and assess these ML models are obtained from a series of 3D finite element analyses for two monopile geometry ratios (*L/D* = 2 and 6), based on the Cowden till soil profile (Byrne et al. 2020). Note that the ground-related inputs for the 3D finite element analyses are non-stochastic. For more realistic modelling of ground-related inputs, random finite element analyses may be implemented but that is beyond the scope of the current paper. The ML inputs are *z/L* and *L/D* and the ML outputs are the soil reaction parameters (indicated in Fig. 2). The training set and test set are obtained by a random 80:20 split of this dataset. For this study, two polynomial regression models (of degrees 1 and 2) are tested. L1 regularisation and 5-fold cross validation was implemented for training. Two NN regression models (with the tanh and Relu activation functions) are tested. One hidden layer is used, as trials using more hidden layers indicate minimal improvement. The optimal number of neurons in the hidden layer is obtained as 32, after implementing 5-fold cross validation over the choices of 8, 16, 32, 64 or 128 neurons. For the GP regression model, two kernels (SE and Matérn ν=5/2 kernels) with zero mean function are tested. As the log marginal likelihood is not convex, 10 random initialisations are used to avoid local optima. The Scikit-Learn library (Pedregosa et al. 2011) is used to implement the above ML models.

RESULTS

For conciseness, only the more interesting results of the highly nonlinear parameters (k, $\bar{y}_u$) (i.e. the initial stiffness and ultimate value) for the distributed soil reactions are shown here;

(k_p, p_u) and (k_m, m_u) are the corresponding $(k, \bar{y}_u)$ parameters for the distributed lateral and moment soil reactions. Fig. 5 compares the polynomial regression predictions of these parameters with the actual values in the training set and test set. Both the 1st and 2nd degree polynomial models cannot capture the highly nonlinear nature of the actual data. The added model complexity provided by the higher polynomial degree does not provide much benefit as the training data are not quadratic in nature.

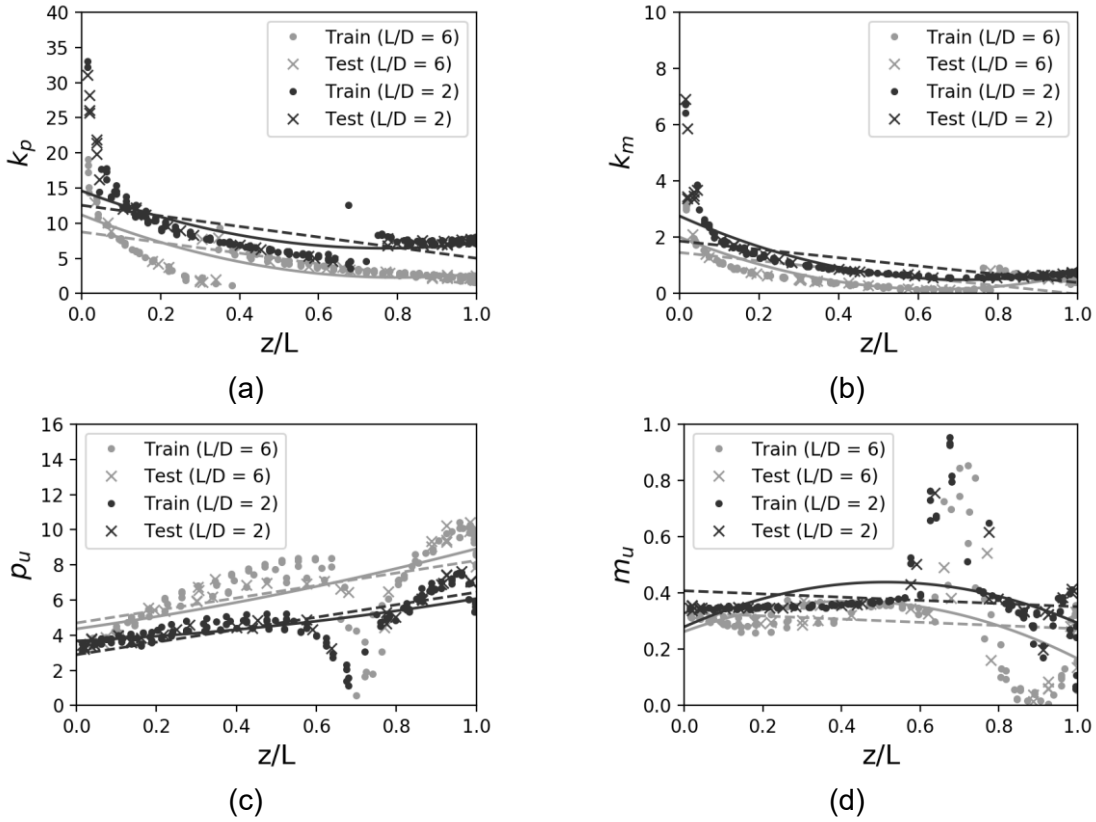

Fig. 5: Predictions of the polynomial regression models (dashed lines = 1st degree, solid lines = 2nd degree) for two *L/D* ratios against the actual data in the training set and test set.

Fig. 6 compares the NN regression predictions of the same soil reaction parameters. The model with the tanh activation function is able to better capture the highly non-linear nature of the actual data than the model with the ReLu activation function. Fig. 7 compares the GP regression predictions of the same soil reaction parameters, where the shaded bounds represent the 95% confidence interval of the predictions (the confidence intervals for both kernels are very similar). The confidence intervals increase in size where there are no data points or the data points are noisy, which corroborates with the intuition that uncertainty increases where there is little or noisy information. Both the GP models (with the SE and Matérn kernels) can accurately capture the highly nonlinear nature of the actual data and they show very similar predictions (the predictions virtually overlap in Fig. 7).

DISCUSSION

To evaluate the accuracy of the ML models, the mean square error (MSE) is computed:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad [5]$$

where n is the data sample size, y_i is the actual value of the ith sample and $\hat{y}_i$ is the corresponding predicted value. The smaller the MSE, the better the model is performing.

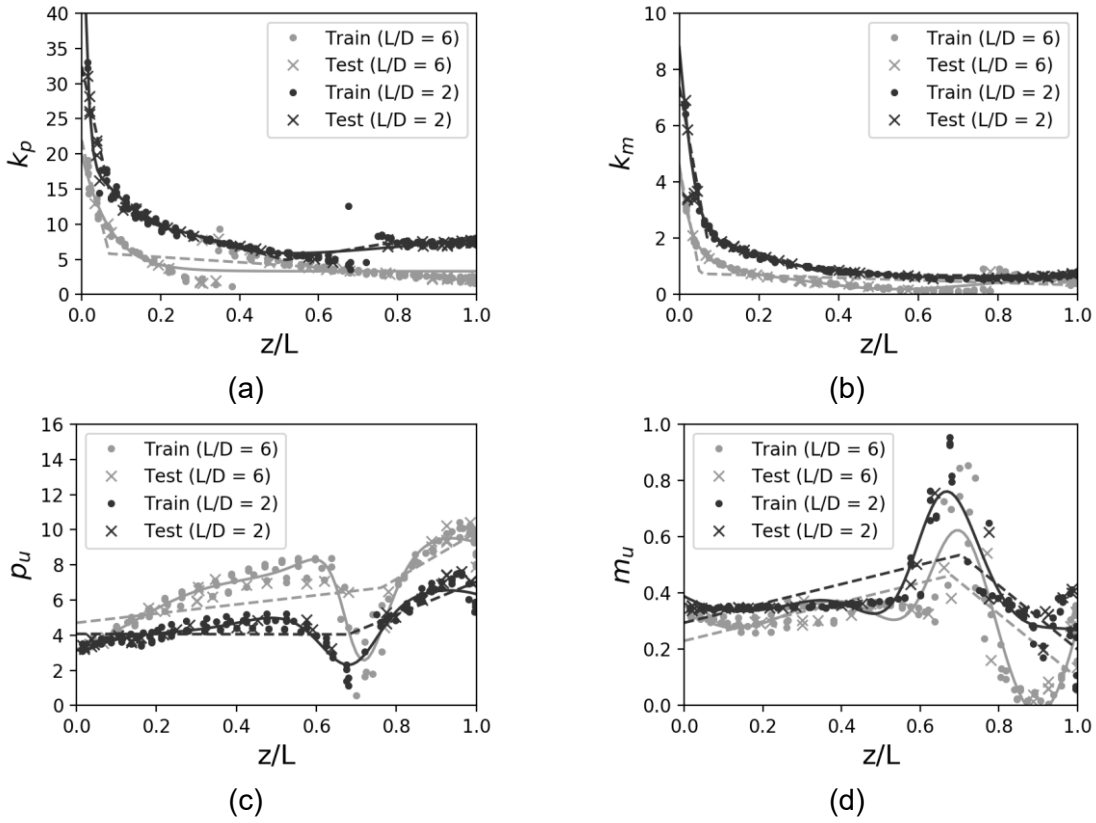

Fig. 6: Predictions of the NN regression models (dashed lines = ReLu activation, solid lines = tanh activation) for two *L/D* ratios against the actual data in the training set and test set.

Table 1 shows the MSE scores (based on the test set) for the best performing variant of each ML model. Evidently, the GP regression model is the most accurate model. The NN regression model is marginally less accurate than the GP regression model, while the polynomial regression model is the least accurate.

Table 1. MSE scores for the polynomial, NN and GP regression model predictions.

Parameter	Polynomial (2nd degree)	NN (tanh activation)	GP (Matérn kernel)
k_p	17.15	1.81	1.41
k_m	0.58	0.22	0.24
p_u	0.86	0.33	0.26
m_u	0.01	0.01	0.00

In terms of interpretability, the polynomial model is the easiest to interpret as the weights associated with each input can be regarded as the contribution of each input to the output predictions. The GP model can also be interpreted in a straightforward manner; the predictions are weighted averages of its 'nearby' data points, where the definition of 'nearby' depends on the length-scale hyperparameter. The NN model is the hardest to interpret, as it is not obvious what the transformed data (using the activation function) represents.

In general, the GP model was found to be the best performing model in terms of accuracy and interpretability. Moreover, it is a non-parametric model which allows it to automatically adapt to any non-linear trend and grow in complexity with the data. It can also account for uncertainty due to lack of data or noisy data via its confidence interval measure, which offers opportunity to add more 3D finite element analyses to the training set to reduce the design risk (should the uncertainty exceed some defined tolerance).

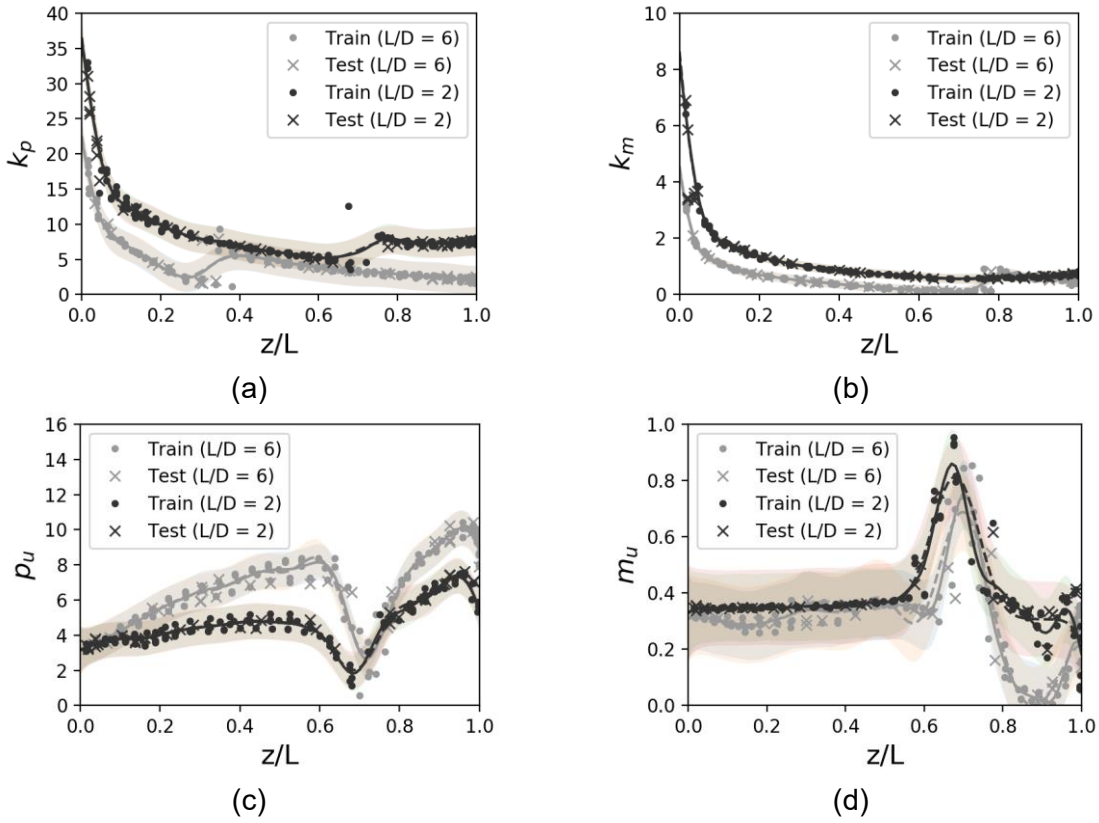

Fig. 7: Predictions of the GP regression models (dashed lines = SE kernel, solid lines = Matérn kernel) for two *L/D* ratios against the actual data in the training set and test set.

DESIGN EXAMPLE

To illustrate the data-driven approach based on the best variant of the three ML models, they are used to predict the ground-level load-displacement behaviour of a monopile in the calibration set (D = 7.5m, L = 45m, more details available under the reference monopile C11 in Byrne et al. 2020) in Cowden till soil. Fig. 8 compares these predictions with that obtained from the PISA 1D (numerical) model (which uses the numerical soil reactions computed specifically for the monopile and provides the best possible performance of the PISA 1D model). The data in Fig.8 indicates that all three data-driven methods provide a close fit with the PISA 1D (numerical) model although the GP model is seen to provide the best agreement. The GP model predictions are also in line with the predictions by the PISA 1D (parametric) model that was obtained from Stage 2 optimisation in Byrne et al. (2020), which requires additional tuning efforts than the data-driven approach.

CONCLUSION

This paper proposes a new PISA data-driven approach for monopile design, which seeks to combine the efficiency of the PISA rule-based approach and the accuracy of the PISA numerical-based approach. In this new approach, a machine learning model is used to determine the most appropriate soil reactions directly and automatically from data, which enables straightforward accommodation for new soil profiles. This allows past design experience to be integrated straightforwardly into the designs of future projects. Furthermore, this paper compares the performance of three machine learning models for the data-driven approach. The results show that the GP regression model is the best performing model based on accuracy and interpretability. Moreover, this model provides intrinsic uncertainty estimates, which offers opportunities to add to the training set to reduce design risks. It can

also automatically adapt to different non-linear trends, and grow in complexity with the data, making the data-driven approach scalable and highly adaptable to different soil profiles (which may have different non-linear trends for the soil reaction parameters).

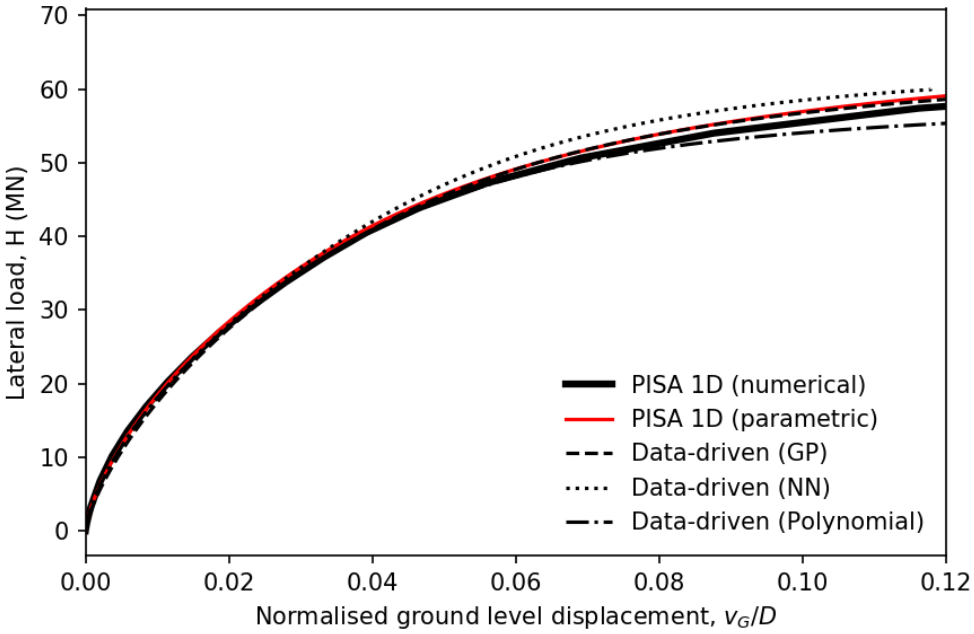

Fig. 8: Comparison of the ground-level load-displacement predictions for a monopile (*L/D*=6) by the data-driven approach, PISA 1D (parametric) and PISA 1D (numerical) model

ACKNOWLEDGEMENTS

The authors acknowledge the support of Ørsted. Byrne is supported by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme.

REFERENCES

API, 2010. RP 2A-WSD - Recommended Practice for Planning, Designing and Constructing Fixed Offshore Platforms. Washington: American Petroleum Institute.

Bishop, C. M., 2006. Pattern Recognition and Machine Learning. Springer.

Burd, H.J., Taborda, D.M.G. Zdravković, L. Abadie, C.N., Byrne, B.W., Houlsby, G.T., Gavin, K.G., Igoe, D.J.P., Jardine, R.J., Martin, C.M., McAdam, R.A., Pedro, A.M.G., Potts, D.M. (2020) PISA Design Model for Monopiles for Offshore Wind Turbines: Application to a Marine Sand. *Géotechnique*, in press.

Byrne, B.W., Houlsby, G.T., Burd, H.J., Gavin, K.G., Igoe, D.J.P, Jardine, R.J., Martin, C.M., McAdam, R.A., Potts, D.M., Taborda, D.M.G. and Zdravković, L. (2020) PISA Design Model for Monopiles for Offshore Wind Turbines: Application to a Stiff Glacial Clay Till. *Géotechnique*, in press.

Byrne, B.W., Burd, H.J., Zdravkovic, L., Abadie, C.N., Houlsby, G.T., Jardine, R.J., Martin, C.M., McAdam, R.A., Pacheco Andrade, M., Pedro, A.M.G. Potts, D.M., Taborda, D.M.G (2019) PISA Design Methods for Offshore Wind Turbine Monopiles. Offshore Technology Conference, Paper number 29373

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

Panagoulias S., Brinkgreve R.B.J., Minga E., Burd H.J., McAdam R.A., 2018. Application of the PISA framework to the design of offshore wind turbine monopile foundations. WindEurope 2018 conference at the Global Wind Summit, Hamburg, Germany.

Pedregosa F., Varoquaux, G., Gramfort A., Michel V., Thirion B., Grisel O. et al., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research. 12(Oct): 2825–30.

Rasmussen, C. E. and Williams, C. K. I., 2006. Gaussian Processes for Machine Learning. MIT Press.