PILE DRIVEABILITY PREDICTION WITH MACHINE LEARNING ALGORITHMS

Source: SteKrueBe - https://commons.wikimedia.org/w/index.php?curid=17009450

In recent years, Data Science has rapidly transformed businesses and industries in a great range of sectors. Data Science has emerged as a powerful tool revolutionizing how civil engineers approach infrastructure design, planning and management.
In the context of the ISFOG 2020 Conference in Austin, Tx, a prediction event was launched inviting geotechnical engineers to gain hands-on experience implementing machine learning models. The proposed problem was the prediction of blowcounts vs depth required to install jacket piles in North Sea soil conditions.

A driveability analysis involves three stages: First the soil resistance to driving (SRD) is assessed for short increments of pile penetration until the target embedment. Second, the dynamic driving process is simulated by a numerical method (1d wave equation analysis) and a bearing graph is calculated for selected depths. Third, the bearing graphs are interpolated to derive blow counts. Figure 1 shows a flowchart describing the main inputs required and the analysis procedure used to conduct a driveability stud (Byrne et al, 2018)

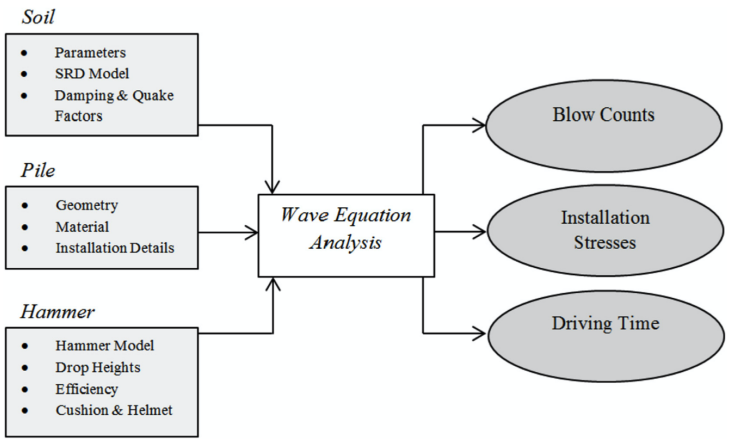

Figure 1: Flowchart of principal inputs and outputs for a wave equation based driveability analysis

To ensure that the modeling is as accurate as possible, information from the monitoring reports related to hammer type, input energy, driving delays, etc. for each pile is carefully considered in each driveability analysis to ensure the predictions for each proposed soil resistance are genuine

The prediction event explored whether machine learning models could be trained on a set of pile driving records to provide reliable estimates of blow counts for unseen locations.

DATA SET

The available dataset (provided by Cathie) consists of installation records for 114 piles in the North Sea with the following data:

- CPT profiles with cone tip resistance (qc), sleeve friction (fs) and pore pressure measurements (u)
- Pile outer diameter and wall thickness
- Blow count and hammer energy

For the prediction event the data was partitioned into a training set of 94 piles and 20 piles were used as a validation data set. Only training data was provided and the validation data was used to assess the quality of the proposed machine learning models. Figure 2 shows processed data for one pile location, data has been mapped to a regular grid with nodes every 0.5 m (ISFOG2020 pile driving prediction, kaggle.com)

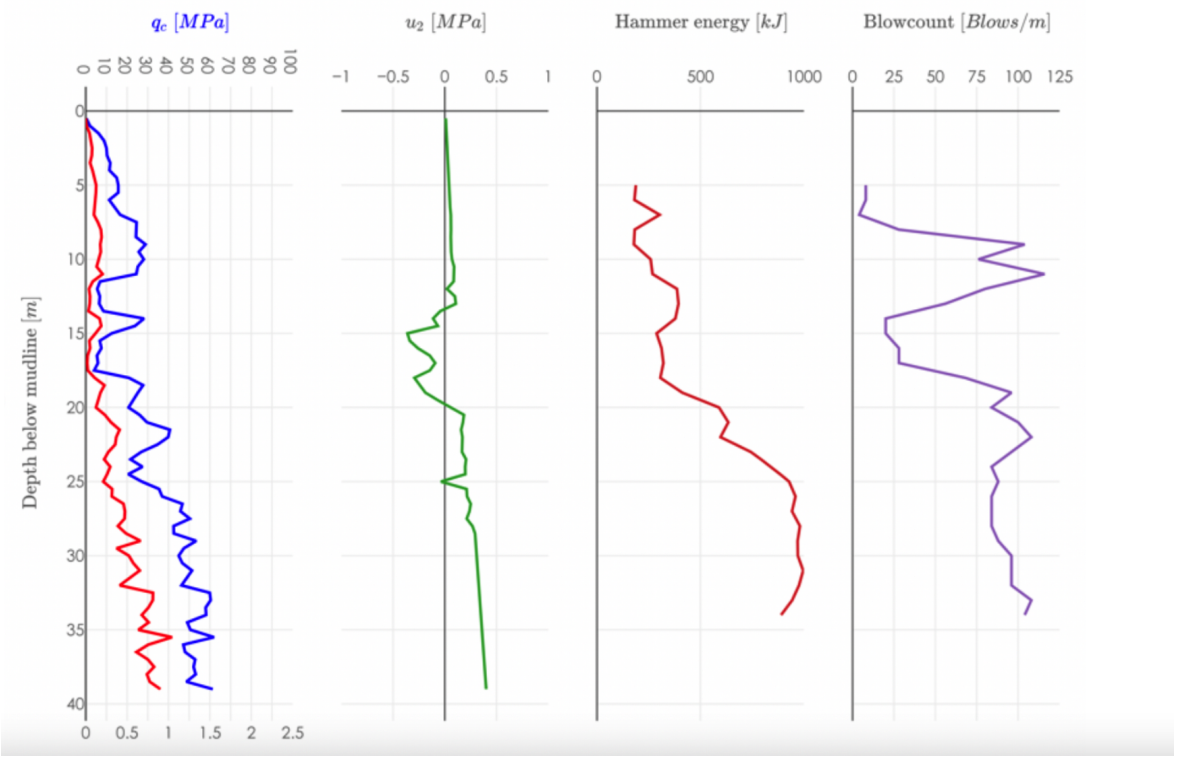

Figure 2: Data processing and grid mapping for one pile location (ISFOG2020, pile driving prediction, kaggle.com)

DATA ANALYSIS AND PREPARATION

The dataset provided is organized in 28 columns containing PCPT data (qc, fs, u), recorded hammer data (blowcounts, normalized hammer energy, normalized ENTHRU, which is the transferred energy at the pile top, and total number of blows), pile data (pile diameter, external wall thickness and final pile penetration). The pile location is identified by a unique ID and z defines the depth below the mudline.

Plotting the cone tip resistance, blow counts and normalized ENTHRU energy for all pile locations shows how the data varies with depth (Figure 3)

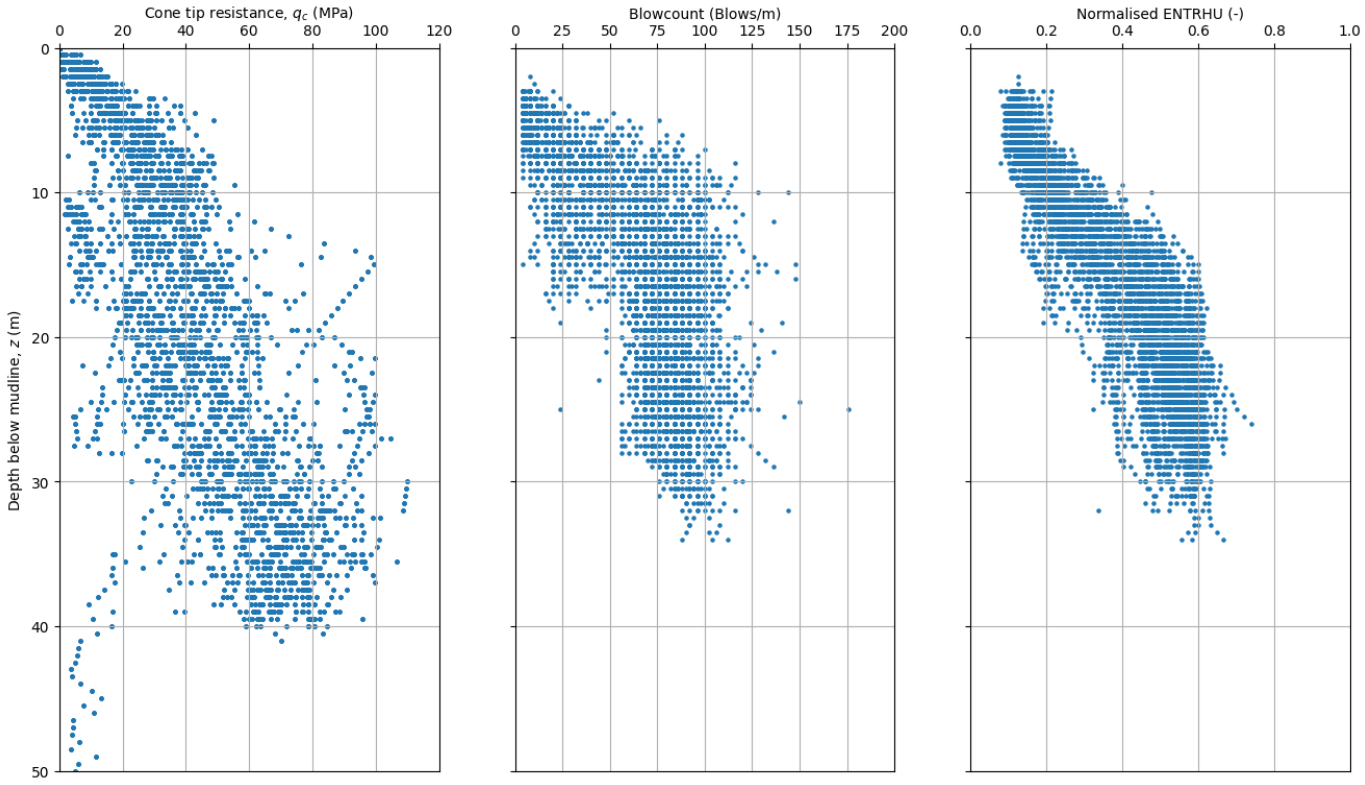

Figure 3: Cone tip resistance (qc), Blowcount (blows/m) and Normalized ENTHRU vs depth

The cone resistance data shows that the site mainly consists of sand of varying relative density. It also shows that there are locations with very high cone resistance (above 70 MPa). The blow count profile with depth shows blow counts that are relatively well clustered and following an increasing trend with depth. This increasing trend with depth is also observed in the normalized ENTHRU profile.

Plotting the data, showing the relationship between blowcount and other variables like cone tip resistance, normalized ENTHRU and depth to mudline for one of the pile locations (“EK”) shows significant scatter and a non-linear behavior, this will be taken into account when the machine learning model for the prediction is implemented. (Figure 4)

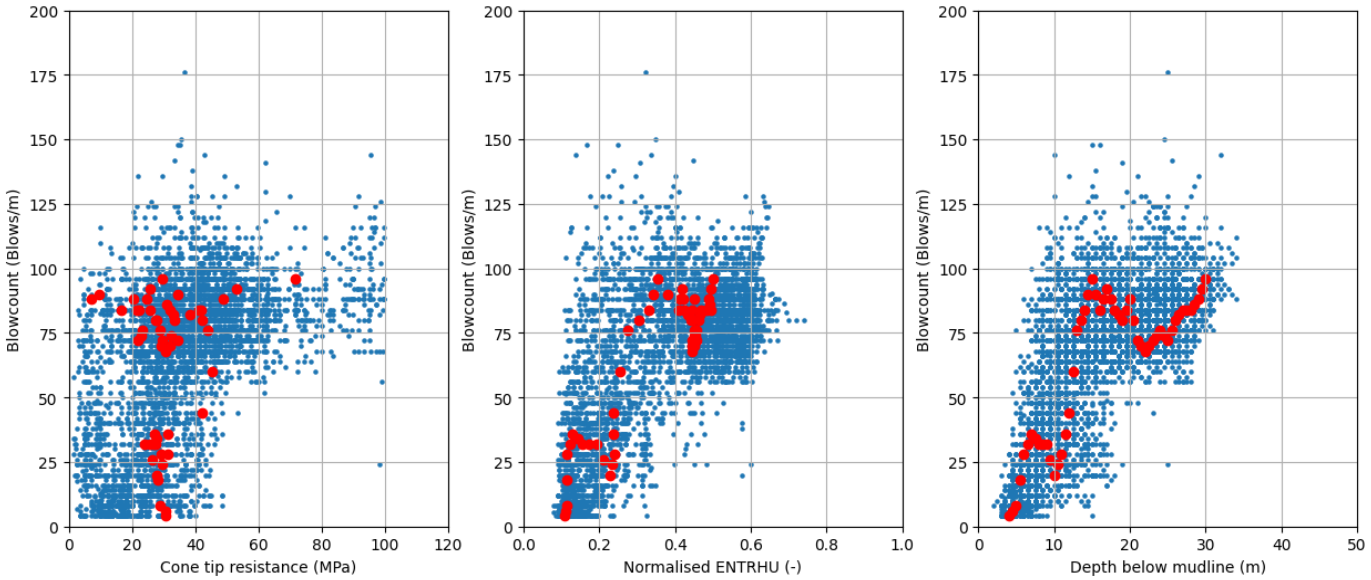

Figure 4: Cone tip resistance, Normalized ENTHRU and depth below mudline vs blowcount, plotted for pile at location “EK”

In order to understand the correlation of the variable that we want to predict (Blowcount (blows/m) with other variables, we can prepare a correlation matrix. The matrix shows that the variables that have a greater influence on Blow Count are cone tip resistance, shaft friction, normalized ENTHRU, normalized hammer energy and depth. This is shown in Figure 5:

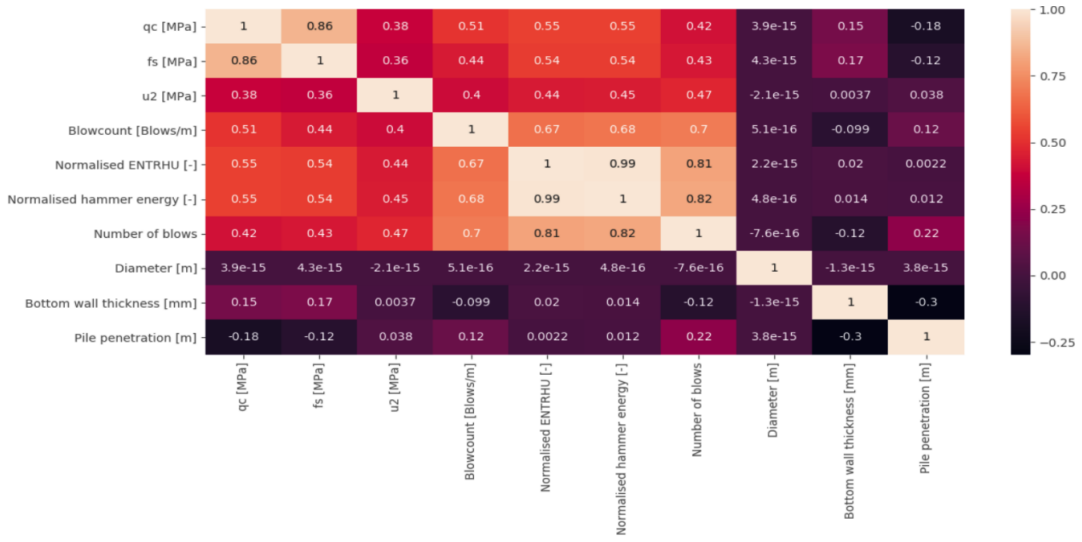

Figure 5: Correlation matrix for variables in the ISFOG 2020 dataset (Wijk et al. 2021)

FEATURE ENGINEERING: USING GEOTECHNICAL KNOWLEDGE

As part of the data provided for the prediction challenge, a more comprehensive training dataset is available, which includes additional information on the total and effective vertical stress of the soil for each depth increment, it also provided information of the pore water pressure. Using this information, along with the cone tip resistance qc and the shaft friction fs, we can derive a new feature variable for our dataset: the static resistance to driving (SRD).

The SRD is a profile of the shaft and toe resistance developed during pile installation. An SRD profile differs from a static capacity profile in that it models the cumulative increase in shaft capacity with further pile installation and has a toe resistance associated with each driving increment.

Several driveability approaches have been proposed throughout the years to calculate the SRD of piles installed in the North Sea. One of the traditional approaches is the one proposed by Alm and Hamre (2001), developed from back-calculated studies from pile installations in the North Sea. The equations are shown below, where τf is the ultimate shaft friction and qb is the unit end bearing resistance. The effective vertical stress is represented by $\sigma'v$, Patm is the atmospheric pressure and δ is the friction angle between the pile shaft and the soil.

$$\tau_f = \tau_{res} + (\tau_{f\max} - \tau_{res})e^{-kh}$$

$$\tau_{f\max} = 0.0132 q_c \left(\frac{\sigma'_{v0}}{P_{atm}}\right)^{0.13} \tan\delta$$

$$k = 0.0125\left(\frac{q_c}{P_{atm}}\right)^{0.5}$$

$$q_b = 0.15 q_c \left(\frac{q_c}{\sigma'_{v0}}\right)^{0.2}$$

$$Qs = \int_0 \tau f \cdot \pi \cdot D \cdot dz$$

$$Qb = \int_0 qb \cdot \pi \cdot D \cdot t$$

$$SRD = Qs + Qb$$

Equations for the Alm and Hamre (2001) SRD approach. (Source: Byrne et al, 2018)

BASIC MACHINE LEARNING MODEL: LINEAR REGRESSION

The most basic type of Machine learning model is the linear model. The general equation for a linear model with N features is given as:

$$y = a_0 + a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_N \cdot x_N + \epsilon$$

where ϵ is the estimation error. Based on the training dataset, the values of the coefficients (ao…… a_N) is determined using optimization techniques to minimize the difference between measured and predicted values. In order to assess how well our machine learning model is predicting our target variable, we need a measure of the error between the measured and the predicted value. The coefficient of determination R^2 is a measure of how well future samples will be predicted by the model. A good model has an R^2 close to 1. The equation for R^2 is given below

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n_{\text{samples}}-1}(y_i - \bar{y})^2} \quad \text{where } \bar{y} = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} y_i$$

$\hat{y}_i$ is the predicted value of the i-th sample and y_i is the true (measured) value.

The simplest linear model depends on only one feature, if we select one of the variables that showed more correlation to blowcount, for example ENTHRU (normalized energy transmitted to the pile) as the only feature and we fit a regression model, we can realize that the linear model is not the most appropriate choice (Figure 6)

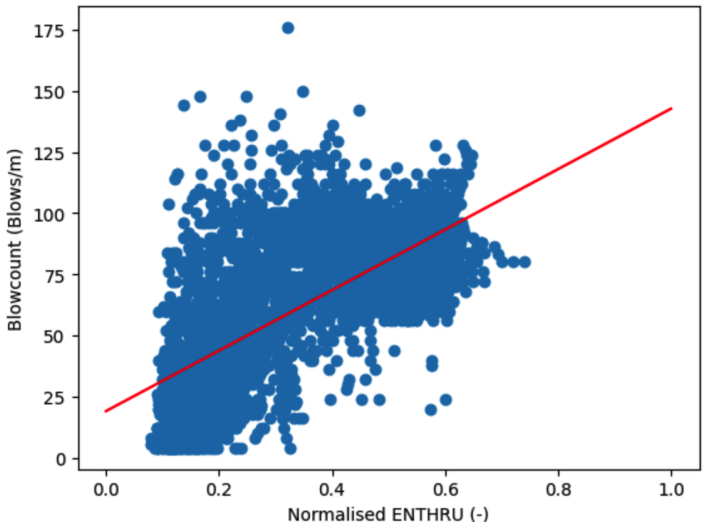

Figure 6: Linear regression model for Normalized ENTHRU as only feature

The relationship between ENTHRU and blowcount is clearly non-linear, we can linearize the feature ENTHRU using a tangent hyperbolic law as follows:

$$(\text{ENTHRU})_{lin} = \tanh(5 \cdot \text{ENTHRU}_{norm} - 0.5)$$

The model with the linearized feature can be written as

$$BLCT = a_0 + a_1 \cdot (\text{ENTHRU})_{lin}$$

Fitting (training) a linear regression model using scikit-learn results in the model shown in Figure 7, with an $R^2 = 0.554$

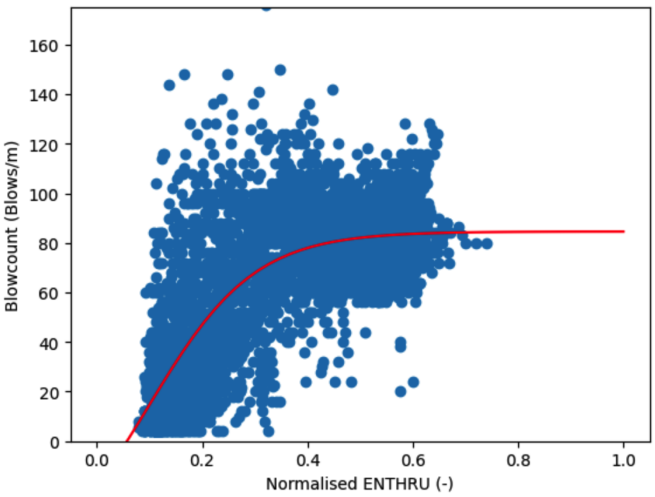

Figure 7: Regression model for linearized ENTHRU (R^2=0.554 for training)

The linear regression model could be improved by adding multiple features. Adding a feature can improve the model if it has a meaningful relation to the output (Blowcount). In this case, the features to be added will be the linearized ENTHRU, linearized z and the linearized SRD. The model with the combined features will take the next mathematical form:

$$BLCT = a_0 + a_1 \cdot \tanh(5 \cdot \text{ENTHRU}_{norm} - 0.5) + a_2 \cdot \tanh\left(\frac{SRD}{1000} - 1\right) + a_3 \cdot \tanh\left(\frac{z}{10} - 0.5\right)$$

Calculating the R^2 value after training yields a value of R^2=0.619, which is better than the initial case. Using the pile locations that we reserved for testing the model, we can check how well the model performs on these unseen locations, yielding a R^2=0.732

We can also use the model to make predictions of the blowcounts at locations reserved for the testing dataset (not previously seen by the model in training). For location “BM” the comparison between the predicted blowcounts and the measured values is shown in Figure 8. It can be seen that even though the R^2 score was reasonable for such a scattered dataset, the model overpredicts the blowcounts at location “BM”.

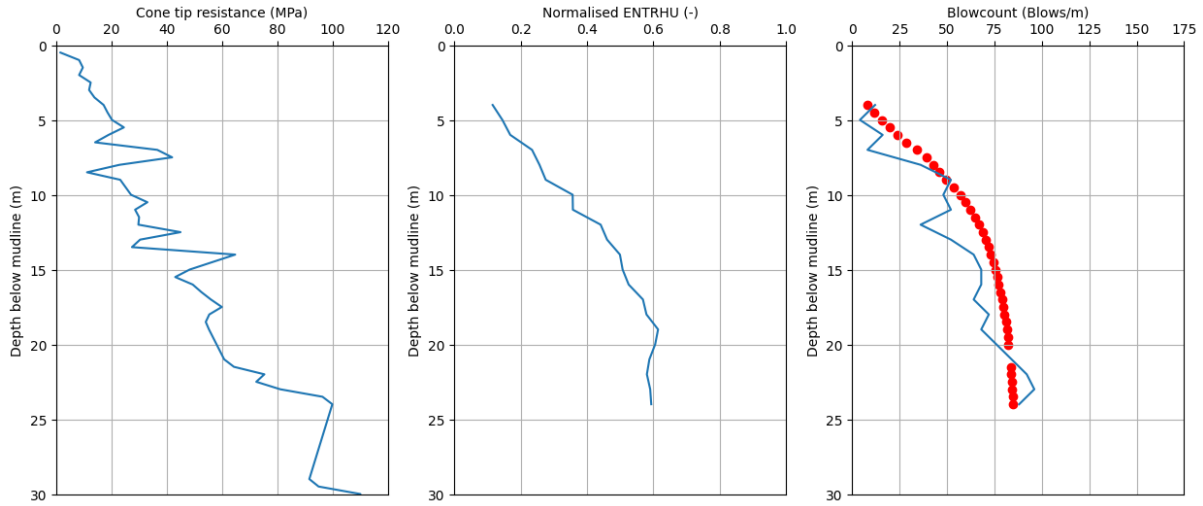

Figure 8: Predicted blowcounts and measured blowcounts vs depth for location “BM”

IMPROVING THE MACHINE LEARNING MODEL

Given the non-linear behavior of the data, a different type of Machine learning model will be implemented: Support Vector Machine (SVM), with radial basis function (RBF). Support Vector Machine model can handle non-linear data using a technique called kernel function, which maps the input vectors (features) into higher dimensional space (adds more dimensions) and rearranges the data in a way which is linearly separable. There are three kernel functions that can be implemented in SVM: linear kernel, polynomial kernel and radial basis function (RBF). In this case,we will use the RBF kernel, which can create complex regions within the feature space, transforming the non-linear data into linear separable data.

For projecting the data into higher dimensional space the RBF kernel uses a radial basis function which can be written as:

$$K(X_1, X_2) = \exp(-\frac{\|X_1 - X_2\|^2}{2\sigma^2})$$

Here **||X1 - X2||^2** is known as the **Squared Euclidean Distance** and **σ** is a free parameter that can be used to tune the equation.

The implementation of SVM with RBF was performed using Scikit-learn. Implementing the model required tuning two hyperparameters: C and gamma. C is a regularization parameter that controls the trade-off between achieving a good fit to the training data and a simple decision boundary. Hyperparameter gamma determines the width of the kernel function.

Both hyperparameters were optimized using gridsearch resulting in C=4 and gamma=0.001. The SVM model implemented with these hyperparameters resulted in R^2= 0.868 for training and R^2=0.882 for testing, which is a far better result compared to the linear regression model. This is shown in Figure 9.

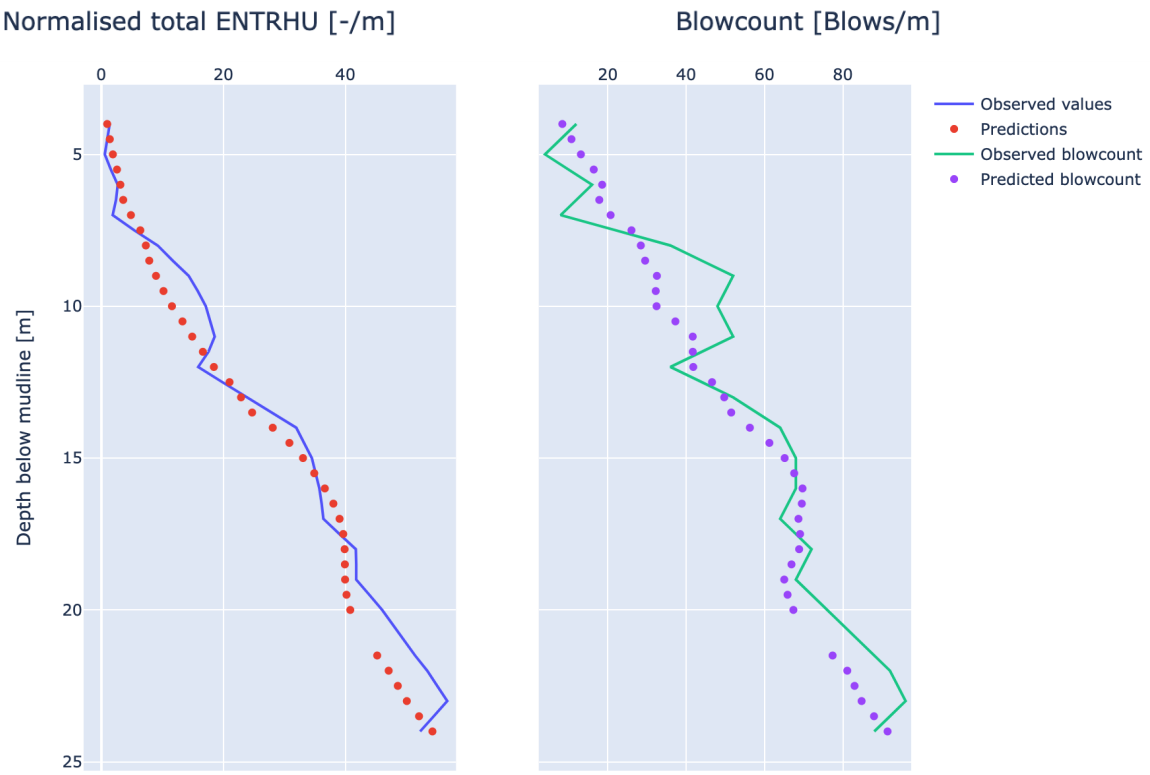

Figure 9: Observed vs Predicted values of Normalized total ENTHRU and Blowcount vs depth below mudline for the SVM with RBF model

XGBOOST MODEL

Extreme gradient boosting (XGBOOST) is an open source implementation of the gradient boosting algorithm. Gradient boosting is a class of ensemble machine learning algorithms which can be used for classification or regression problems. Our case is a regression problem. Ensemble algorithms are built by adding decision trees one at a time to the ensemble model and fit to correct the prediction errors resulting from the previously added trees. This form of constructing ensemble models is referred to as “boosting” (Brownlee, 2021). The models are fit using a differentiable loss function with gradient descent optimization algorithm, resulting in the technique’s name: “gradient boosting”

Similarly to the SVM model, grid search was employed to tune the hyperparameters of the model (find the optimal combination of parameters that yields the best model performance). For the case of XGBoost, these hyperparameters are the n_estimators (number of boosting stages to perform), learning_rate (controls how much of a contribution each new estimator will make to the ensemble prediction) and the max_depth (limits the number of nodes in the decision trees). The best hyperparameters were evaluated as: learning_rate=0.16, n_estimators=65, max_depth=4. Regularization was also employed to reduce overfitting (to prevent the model from ‘memorizing’ the training set and not generalizing for new data in the test set) Regularization helps in controlling the complexity of the model and to reduce variance.

The optimized XGBoost model resulted in R^2= 0.883 for training and R^2=0.886 for testing. The observed and predicted values for the Normalized total ENTHRU and Blowcount vs depth below mudline are shown in Figure 10. A comparison of the observed vs predicted values for Blowcount vs depth below the mudline for the linear model, SVM with RBF and XGBOOST is shown in Figure 11.

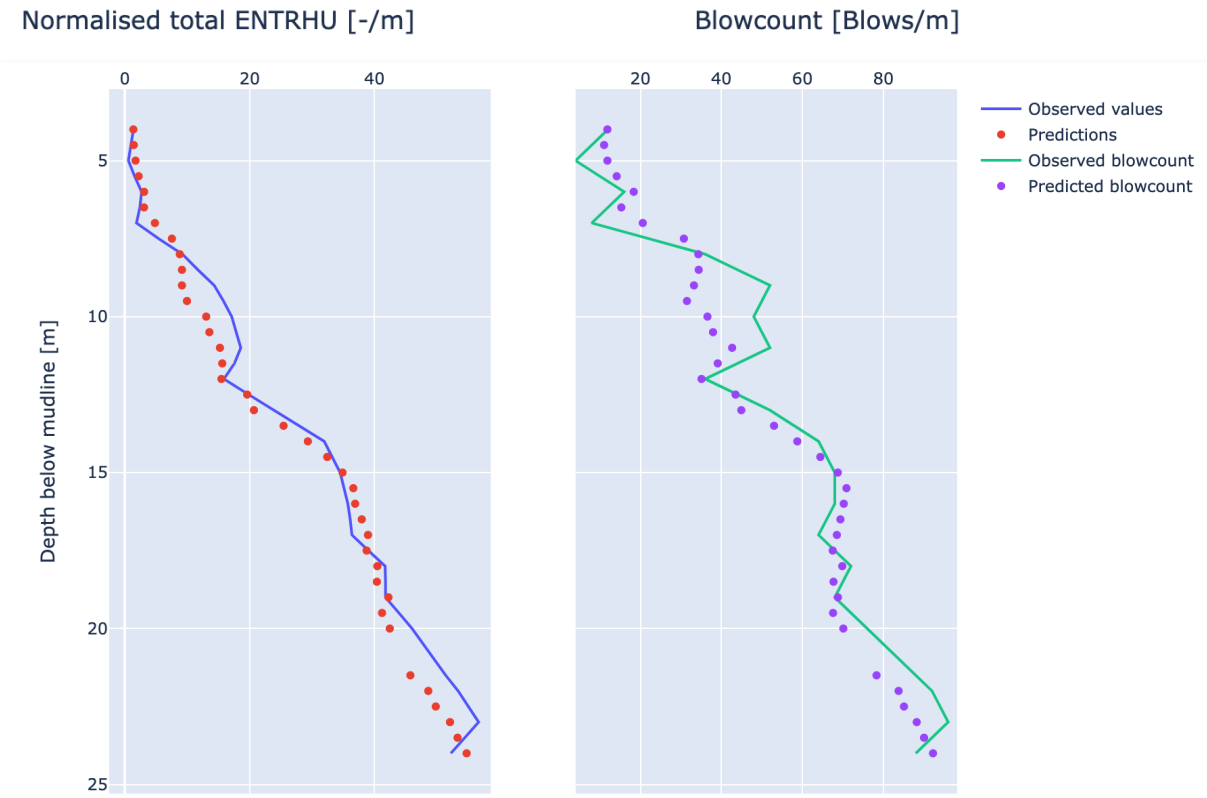

Figure 10: Observed vs Predicted values of Normalized total ENTHRU and Blowcount vs depth below mudline for the XGBoost model

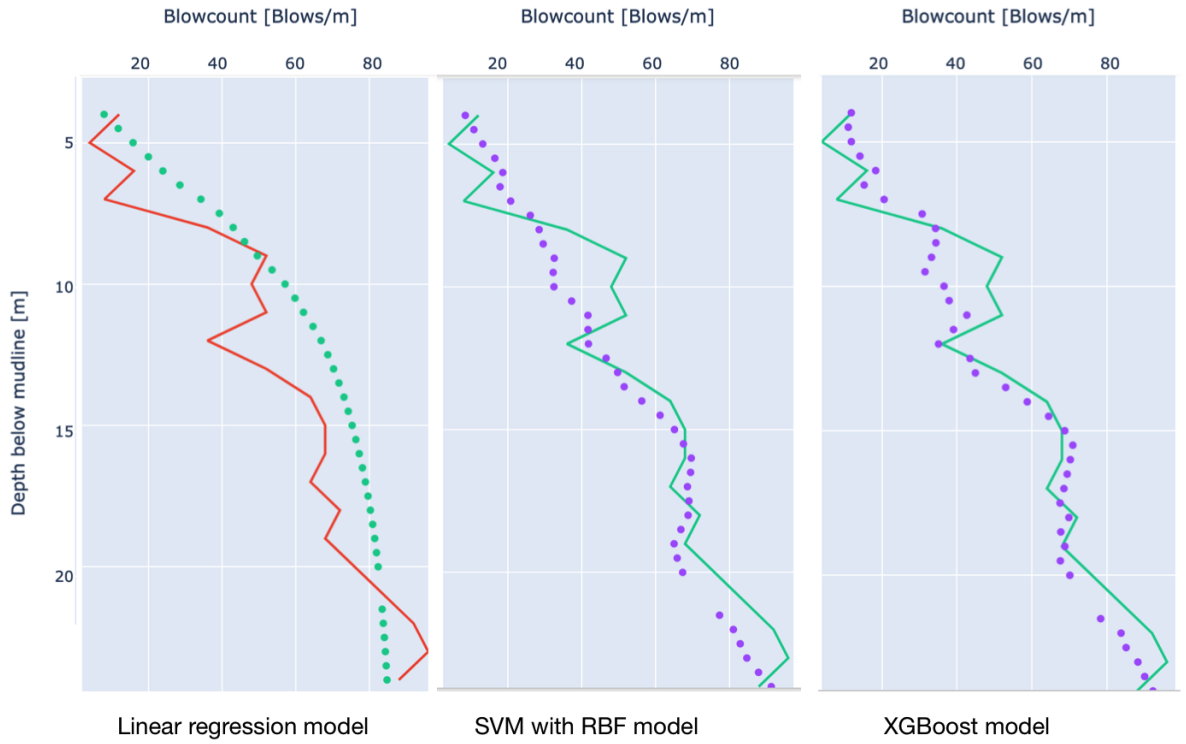

Figure 11: Comparison of Observed vs Predicted values of Blowcount vs depth below mudline for the Linear regression, SVM with RBF and XGBoost models (prediction values are scattered)

The next table shows a summary of the R^2 values for training and testing achieved by each of the algorithms. It can be seen that SVM with RBF and XGBoost reach very similar performance.

Algorithm	R^2 training	R^2 test
Linear regression	0.619	0.732
SVM with RBF	0.868	0.882
XGBoost	0.883	0.886

CONCLUSION

It can be concluded that the use of robust machine learning algorithms like SVM with RBF and XGBoost, along with the use of feature engineering based on domain knowledge allowed us to achieve a good result in the prediction of blow counts for pile driveability.

REFERENCES

Brownlee Jason XGBoost with Python, Machine Learning Mastery, 2018

Brownlee Jason machinelearningmastery.com/xgboost-for-regression, March 2021

Byrne T., Gaven K., Prendergast L.J., Cachim P, Doherty P., Chenicheri Pulukul. "Performance of CPT-based methods to assess monopile driveability in North Sea Sands", Ocean Engineering, 166 (2018) pp 76-91

github.com/snakesonabrain/isfog-workshop, based on ISFOG 2020 keynote paper "Data Science in offshore geotechnics" (Styuts, 2020)

kaggle.com/c/isfog2020-pile-driving-predictions ISFOG 2020 Data Science prediction event

scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html Sckikit-learn: RBF SVM Parameters

scikit-learn.org/stable/auto_examples/svm/plot_svm_regression.html?highlight=svr#

van Wijk J., Alvarez M., Moyo T. "Risk reduction in foundation installation project: how data and A.I. make life predictable and easy" Proceedings of Geotechniekdag, 2021 (researchgate.net/publication 356128889)