PILE CAPACITY PREDICTION BASED ON MACHINE LEARNING USING SUPPORT VECTOR REGRESSION

Jakob T. Udengaard[1], Emil Skouboe[2], Kenny K. Sørensen[3], Umberto Alibrandi[4], and Lars V. Andersen[5]

KEYWORDS

Machine Learning, Pile bearing capacity, Support Vector Regression

ABSTRACT

This study investigates the capability of a Machine Learning algorithm, Support Vector Regression (SVR), to predict the bearing capacity of driven concrete piles. Restrikes conducted to make Dynamic Load Tests (DLT) were used for training the algorithm as the DLT-derived capacity was used as the target value. The performance is compared to the Danish Driving Formula. The research explores the importance of domain knowledge and feature engineering in the process of selecting, manipulating, and creating relevant input variables (features) that improve model performance and generalization. The trained model demonstrates a very high correlation with the target and strong confidence in its bearing capacity predictions, outperforming the Danish Driving Formula. Sensitivity analyses show that a feature-engineered model has improved generalization and interpolation capabilities in parameter regions with few or no data in the training set.

1. INTRODUCTION

Accurate assessment of the bearing capacity of driven piles is a critical aspect of geotechnical engineering, influencing the design of foundations and the economy of the structure. Data from pile driving records can be used to predict pile bearing capacities, but the capacities are significantly affected by the

[1] Per Aarsleff A/S, Denmark
[2] Geo, Denmark
[3] Aarhus University, Denmark
[4] Aarhus University, Denmark
[5] Aarhus University, Denmark

build-up of excess pore water pressure during pile installation. Restrikes after a short time provide a valuable source of information that is significantly less affected by set-up. Restrikes can be made at low cost on piles after end-of-driving, where some combination of pile set and drop height is registered. This study aims to utilize the restrike data, which are readily available or can be obtained at low cost.

Often, *pile driving formulas,* e.g., the Danish Driving Formula [1] are used to determine pile capacities. These semi-empirical methods are based on energy considerations and use the driving data (end-of-driving or restrike). Other pile capacity prediction methods include Static Load Test (SLT) and Dynamic Load Testing (DLT) with signal matching, e.g., the CAPWAP method [2]. Where SLTs are often scarce due to their circumstantial setup and time consumption, DLTs are fast and therefore often preferred.

The Danish Driving Formula is simple to use and remains popular as it allows for immediate estimation of the present pile capacity during driving. However, it assumes that all energy related to the hammer impact is converted into pile penetration except for the strain energy related to elastic pile compression, and it has not been recalibrated for decades. Now, more data types are accessible, and piling equipment has become more advanced. Machine Learning (ML) offers a data-driven approach and enables the inclusion of other relevant variables, e.g., if a pile is coupled, without the need to establish physics-based equations.

Several ML techniques have been applied in geotechnical engineering with Artificial Neural Networks (ANNs) being a popular choice in research. In 1995, Chan et al. demonstrated the feasibility of ANN on the prediction of the ultimate pile bearing capacity from pile driving records, outperforming pile driving formulas [3].

Another popular ML technique is Support Vector Machines (SVM) [4] which has been found to work well for geotechnical applications in comparison with back-propagation neural networks [5]. In 2010, Pal and Deswal compared SVM to the Gaussian Processing (GP) algorithm, which suggests that the latter has a slightly improved prediction capability [6]. Both SVM and GP were shown to significantly outperform pile driving formulas.

This paper aims to verify previous findings and compare the performance of ML to the Danish Driving Formula. This study's focus is, by applying domain knowledge to the data science approach, to include other variables and to manipulate features to improve the performance and generalization of the trained algorithm. Hence, only a single algorithm, SVM, will be applied. It is chosen for its relatively simple and logical formulation and because it is supervised and popular for solving similar geotechnical problems, where it has shown promising results.

Section 2 provides more detailed insight into the background of the paper, and Section 3 outlines the methodology. The results are presented in Section 4 and discussed in Section 5, while Section 6 provides the overall conclusions.

2. BACKGROUND

Driving a pile induces excess pore water pressure within the soil penetrated. When the pore pressures dissipate, the effective stresses increase which in turn increases the pile capacity. The time-dependent increase in pile bearing capacity arising from pore water dissipation and other secondary mechanisms such as creep and thixotropy is referred to as set-up [7]. DLTs, for which restrikes are made to record stress waves, are most often conducted a period of time after end-of-driving to more accurately verify the pile capacity. CAPWAP analysis of the recorded stress waves provides a very reliable determination of the ultimate capacity of driven piles [2].

Therefore, the DLT with CAPWAP analysis is used as the target value in this study. By comparing data of the restrike for DLT with the DLT itself, the comparison is not affected by time, i.e. pile set-up. A model can be trained with this data set to predict DLT results, and if applied at restrikes at a different (earlier) time, the result reflects the capacity at that specific time.

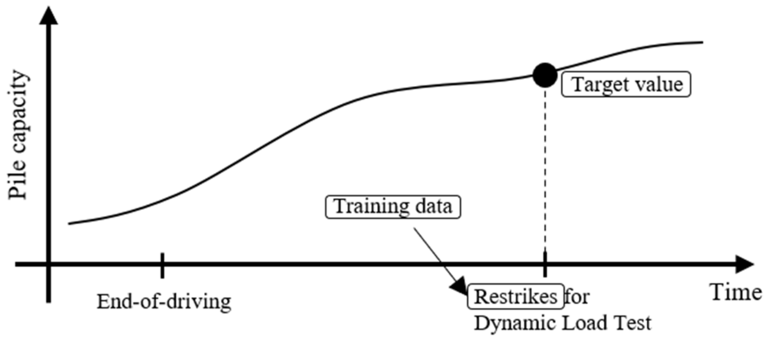

Figure 1 Schematic pile capacity development due to set-up.

Different ML techniques have previously shown improved capabilities of predicting the ultimate pile capacity in comparison with different empirical driving formulas not including the Danish Driving Formula. However, these research projects have focused on comparing prediction performance between different machine learning techniques and relating it to empirical methods [3][5][6]. Little attention was paid to the features used by the algorithms to learn if any additional information will benefit the performance, or how features may be manipulated to achieve generalization and increase performance.

This study attempts to improve the performance and generalization capabilities of the trained ML algorithm using a domain knowledge-based feature engineering approach.

3. METHODOLOGY

For building the machine learning model, Support Vector Regression (SVR) is used. SVR finds a hyperplane in a high-dimensional space that best fits the data points while allowing some errors within a margin. A kernel function transforms the data points into a higher-dimensional space from where the optimal hyperplane is found, and for this non-linear problem, the radial basis function (rbf) kernel is chosen. The data points closest to the hyperplane are the support vectors which have the greater influence on the hyperplane. In regression, the margin that minimizes the distance between the hyperplane and the data points is determined by the hyperparameters (C regularization, ε margin of tolerance, and γ kernel coefficient) which must be tuned in a grid search for optimal performance [8].

The SVR model is compared to the Danish Driving Formula [1], which takes the form:

$$R_{\mathrm{dyn;m}} = \frac{\eta h G}{s + 0.5 s_0} \quad ; \quad s_0 = \sqrt{\frac{2\eta h G_{\mathrm{h}} L_{\mathrm{p}}}{AE}}$$

and uses six variables: the hammer drop height h (m), the hammer weight G_{h} (kN), pile set per hammer blow s (m), pile length L_{p} (m), and cross-sectional area of the pile A (m^2). The nondimensional efficiency factor of the hammer η is equal to 1.0 for modern hammers, and Young's modulus E is set at a constant uniform value of 20 GPa for concrete. The subscript "dyn;m" refers to a *dynamic* resistance that is *measured* and subsequently regulated by partial factors to provide a design value.

Data set

The data set that is used to build the SVR model is a sample of DLT reports of 367 impact-driven piles installed for 16 commercial projects scattered across the western part of Denmark. The target value is the derived pile bearing capacity of these reports based on the CAPWAP method which ranges from 600 to 4610 kN. This includes concrete piles with square cross-sections having side lengths ranging from 25 to 40 cm. The pile lengths range from 8 to 30 m where piles longer than 18 m are coupled with a pile joint. The piles are embedded in various common Danish soils of sand, clay, silt, gravel, clay till, and unhardened limestone, and the environment of deposition ranges over marine, sub-glacial, and meltwater, while the ages of deposition are either

Danien or Quaternary. The hammer type used is either free-falling or accelerating. The restrikes conducted for the DLTs include the following variables: hammer weight (60 to 90 kN), drop height (0.2 to 1.2 m), and pile set (5 to 138 mm). Due to the increased significance of measurement error, pile sets lower than 5 mm have been disregarded.

Data processing and model building

Data exploration has been performed to better understand the data and its limitations. For example, the distribution of the pile lengths in Figure 2 shows a tendency for the pile capacity to increase with pile length though non-linearly. On the contrary, shorter piles do not necessarily have a lower capacity as the distance to firm ground is independent of the pile length but not vice versa. Although stresses generally increase with depth, the pile base depth should be disregarded to avoid the deduction that capacity is always directly proportional to pile length.

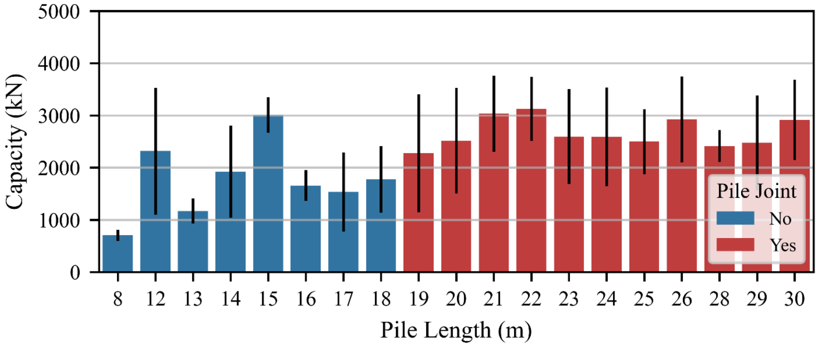

Figure 2 Mean capacity ± one standard deviation (black) vs. pile length grouped by pile joint type.

Features have been manipulated to achieve better generalization using domain knowledge of the physics behind the data and the relations between individual parameters. An example of this feature engineering is that the features *Hammer Weight* and *Drop Height* are multiplied into *Hammer Energy* as both parameters hold information about the input energy. Yet, for the hammer weight, the data set does not capture the parameter space well. In combining the features, a new parameter space is created which may close gaps in the data and provide better coverage of the population as illustrated in Figure 3.

An alternative approach of employing an unsupervised approach such as the Principal Component Analysis (PCA) submits control and domain knowledge from the model building process, and the model behavior becomes non-explanatory as opposed to a manual feature engineering approach.

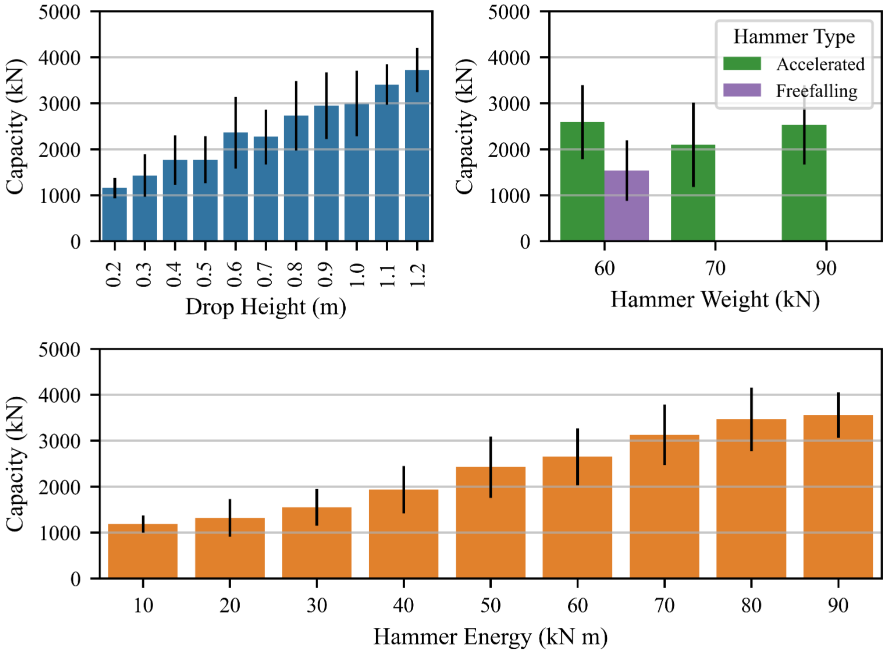

Figure 3 Exploration of initial and manipulated features.

The model building itself follows the iterative process described in Figure 4. For each iteration, the data set is split randomly with 80 % distributed into a training set and the remaining 20 % into a test set (the same random split is used throughout). The model is trained on the training set with a *k*-fold cross-validation scheme. This is done to provide robustness for the estimation of model performance, reduce the risk of overfitting, and assess how well the model will perform on unseen data. The main evaluation score is set as the coefficient of determination, R^2, evaluated with Scikit-learn [8].

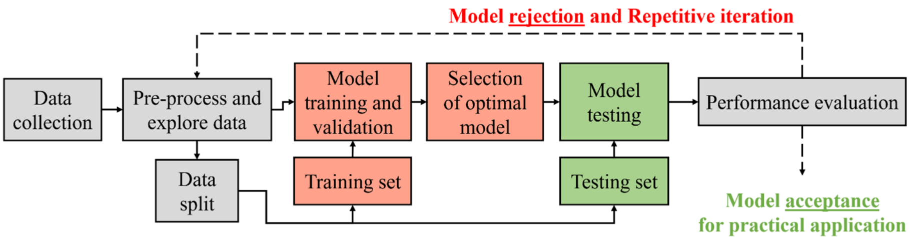

Figure 4 Development process of a supervised machine learning model.

For a random data split, two SVR models have been trained to predict pile capacity: A Baseline model with the same input features as in the Danish Driving Formula (see Eq. (1)), and a Feature-Engineered model. The hyperparameters have been optimized in a three-dimensional grid-search of 10 × 8 × 11 (C × ε × γ) with a k-fold cross-validation scheme of a random data split with 5 folds and 3 repetitions, where the models were evaluated for each iteration with the radial basis function kernel.

4. RESULTS

The feature selection relied on physical considerations of the pile that encapsulate the proportions of the pile geometry, the impact energy, and the soil resistance based on available data on the driven piles. The result of that process is the following features: The proportions and geometry are described in feature spaces of the pile dimension B (m), pile aspect ratio L_p / B (-), pile joint and hammer type (categoricals), pile reinforcement ratio $A_{steel} / A_{concrete}$, and the weight ratio of the hammer to pile G_h / G_p (-). The majority of engineering features are based on normalized parameters to spread out the data points of the feature spaces more evenly. The Hammer Energy $E_h = h \times G_h$ (kNm) represents the input energy, and the pile set per blow s (m) reflects the soil resistance.

The performance of the SVR models for pile capacity prediction as evaluated on training and test sets are shown in Figure 5, the score metrics are shown in Table 1, and the SVR models are compared against the Danish Driving Formula in Figure 6.

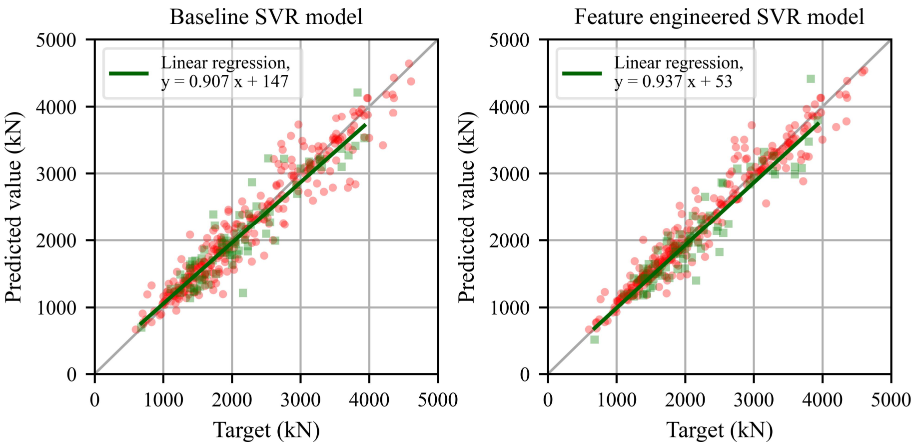

Figure 5 Pile capacity predictive performance of Baseline (left) and Feature-Engineered (right) SVR models. Models are evaluated on the training set (red) and test set (green). The regression line is the test set only.

From Figure 5 and Table 1, it is seen that the training scores are slightly lower than the test scores. The performance of the feature-engineered model is slightly better than the baseline model. In Figure 6, none of the methods is associated with a uniform bias. However, the Danish Driving Formula tends to overestimate the capacity for low capacities and underestimate the capacity for high capacities. This is not the case for the ML approach, though model uncertainty is also present here in the form of fluctuations.

Table 1 Performance metrics of the linear regressions between the model predictions and target. R^2: coefficient of determination, RMSE: Root Mean Square Error.

Model, Hyperparameters	$\boldsymbol{R^2}$		**RMSE**	
	Train	**Test**	**Train**	**Test**
SVR (baseline) $C = 10^{4.5}$, $\varepsilon = 70$, $\gamma = 10^{-1.4}$	0.923	0.846	258	288
SVR (feature-engineered) $C = 10^{4.5}$, $\varepsilon = 70$, $\gamma = 10^{-1.2}$	0.946	0.865	215	269
Danish Driving Formula	0.760	0.687	455	410

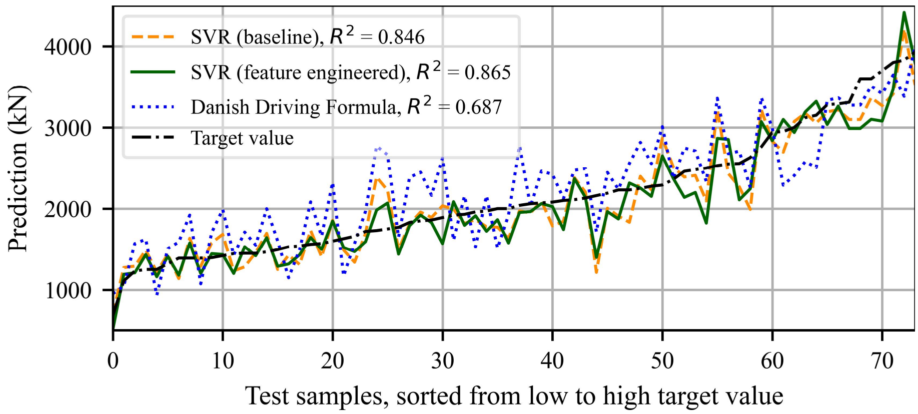

Figure 6 Comparison of the predictive capabilities of SVR models against the Danish Driving Formula on each sample in the test set.

In comparison, Pal and Deswal (2010) used eight input features (E, L_p, G_p, h, G_h, s and *hammer type* (*gravity* or *steam*)) to train an SVR model to achieve a coefficient of determination $R^2 = 0.855$ and RMSE = 372 kN on test samples [6]. The results of the baseline SVR model in this study are similar.

A second analysis is conducted with alternative split methods. The first retains all hammer weights of 60 kN (15 % of the data set), and the second retains all data from a single project (16 % of the data set) to mimic a real-world case and practical application. The results of the analyses are shown in Table 2 and Figure 7. The hyperparameters are tuned similarly besides $\varepsilon = 10$. A lower ε ensures that more predictions are penalized and more become support vectors.

Table 2 Metrics of the alternative split methods for the training process.

Split method	**Model, Hyperparameters**	**Train, R^2**	**Test, R^2**
60 kN hammer	SVR (baseline) $C = 10^4$, $\gamma = 10^{-0.75}$	0.942	–0.133
	SVR (feature-engineered) $C = 10^5$, $\gamma = 10^{-2}$	0.913	0.802
Single project	SVR (baseline) $C = 10^{3.5}$, $\gamma = 10^{-1}$	0.937	0.331
	SVR (feature-engineered) $C = 10^4$, $\gamma = 10^{-1.25}$	0.942	0.632

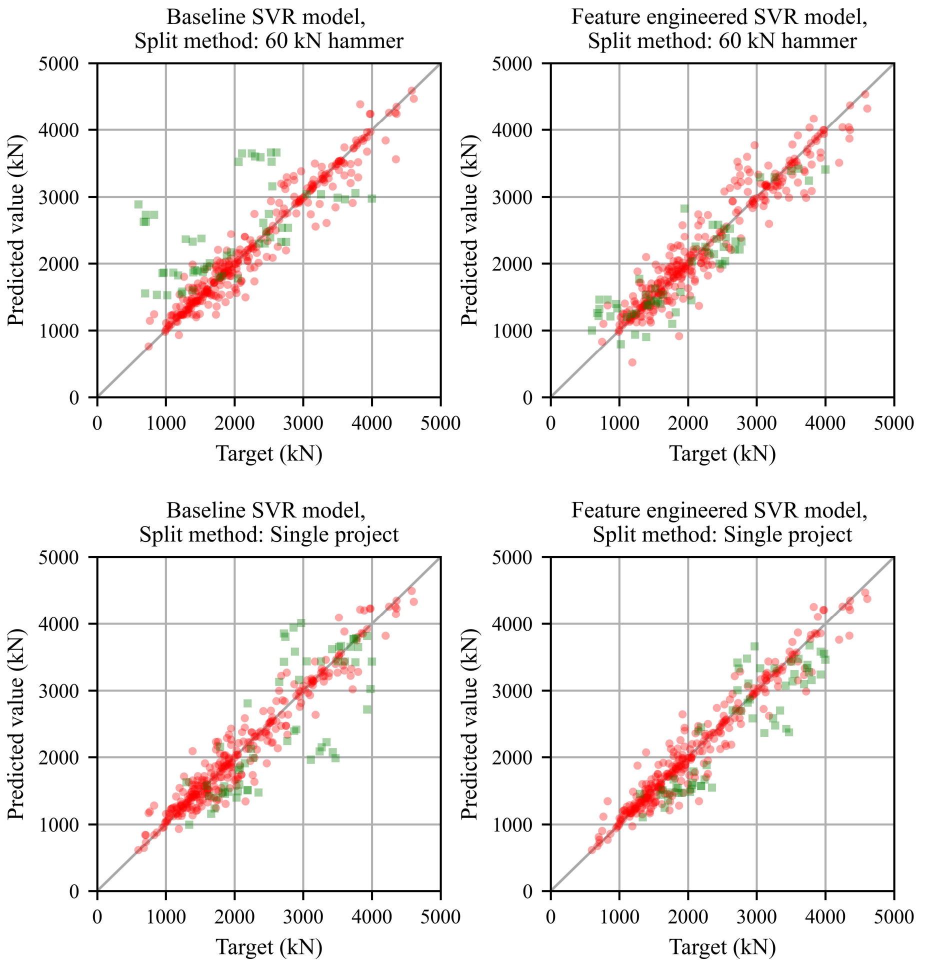

Figure 7 Pile capacity prediction performance of SVR models with 60 kN hammer data points in the test set (above), and a single project case in the test set (below). Model evaluated on training set (red) and test set (green).

5. DISCUSSION

The SVR models show increased accuracy and precision compared to the Danish Driving Formula. The performance gap for predictions on the test set compared to the training set indicates overfitting of the models. However, even the Danish Driving Formula shows poorer performance on the test set compared to the training set (Table 1), suggesting an uneven split with underlying bias or greater error sources in the test set. The relatively lower gap in the score metrics of the SVR models between the training set and the test set in Table 1 shows them to be well-balanced.

The data set comprises only 16 commercial projects, in which typically the same *hammer weight* is used for an entire project. Underlying tendencies for individual projects may be captured by the baseline model through this feature that causes overfitting. The feature engineering efforts, such as constructing *hammer energy*, mitigate this. In the alternative split method where 60 kN hammers were withheld, the baseline model is extrapolating as it was only trained on 70–90 kN hammers (Figure 3). On the other hand, feature engineering makes the model cover a wider parameter space that makes it interpolate within the range of *hammer energies*. The feature-engineered model works nearly as well as for the random split (Figure 7). Some of the performance loss may be attributed to the indirect removal of the *Hammer Type*.

In the single-project split method where data from a whole project site was withheld in training, the feature-engineered model performs better than the baseline model (Figure 7). In this split method, parts of the training set's parameter space may be insufficiently covered in training for the model to accurately predict the target values of the test set. This causes the performance to decrease in comparison to the models trained with a direct random data split, which suggests a level of extrapolation of new projects. This observation emphasizes that the practical application of SVR requires a larger data set that must be representative of the population intended for its application.

As opposed to *Deep Learning* in which feature engineering is made by the algorithm, supervised *Machine Learning* allows the human behind to apply his/her domain knowledge and modify features based on known physics behind the data. This can generalize the model, mitigate overfitting, and make the model more trustworthy. In this research project, knowledge of the driving mechanism was the basis for a feature engineering process that led to the identification of three basic parameters that represent the hammer strike, pile geometry, and soil reaction, and, in addition, ratios between parameters that are more relevant than the original values. Equally important, the data exploration and feature engineering process make the model developer aware of any boundaries present in the parameter space.

The model performances would arguably be improved had the data set been soil-specific. However, the lack of restrictions on the soil type at the pile tip or along the pile shaft enables broader application.

6. CONCLUSION

This paper has shown that feature engineering can significantly improve the accuracy and generalization capabilities of an ML regression model based on the input data generally used for the semi-empiric Danish Driving Formula. This is done using domain knowledge while maintaining control over the learning process of the models. On a direct comparison of the coefficient of determination R^2 in predicting the pile bearing capacity as determined by the CAPWAP method, the SVR model, with a random data split, outperforms the Danish Driving Formula with 0.865 against 0.687.

There are several advantages to using data-driven methods in geotechnics. Complex relations may be made between two variables dependent on a larger multi-dimensional set of features. Supervised machine learning, as opposed to deep learning, allows for further control over the training enabling implementation of domain knowledge of the data and the physics behind them. This mitigates the issue of the “black box” that is inherent in employing machine learning.

The research has shown that the Feature-Engineered model can confidently interpolate over a limited space of original parameters with poor coverage. However, extrapolation and overfitting are shown to be mitigated by a generalization of the features. Yet, the model requires more training data before practical application to enable use on new projects without recalibration.

Future research should involve training on a larger data set with dozens of project cases to learn whether the model can perform well on new project cases unseen in training. Additionally, using the same or similar feature-engineered parameters in a comparison between different ML algorithms is suggested for future research.

ACKNOWLEDGEMENT

Aarsleff provided the data of the 16 commercial project cases used in this study. This is greatly appreciated.

REFERENCES

[1] T. Sørensen and B. Hansen. Pile Driving Formulae — An Investigation based on Dimensional Considerations and a Statistical Analysis. Proceedings of the 4th International Conference on Soil Mechanics and Foundation Engineering, held in London 1956, 2, 61–65, 1957.

[2] Garland Likins and Frank Rausche, 2004. Garland Likins and Frank Rausche. Correlation of CAPWAP with static load tests. In Proceedings of the seventh international conference on the application of stresswave theory to piles, pages 153-165, 2004.

[3] WT Chan, YK Chow and LF Liu. Neural network: an alternative to pile driving formulas. Computers and geotechnics, 17(2), 135–156, 1995.

[4] Cortes, C., Vapnik, V. Support-vector networks. Mach Learn 20, 273–297 (1995)

[5] Pijush Samui. Support vector machine applied to settlement of shallow foundations on cohesionless soils. Computers and Geotechnics, 35(3), 419–427, 2008.

[6] M. Pal and S. Deswal. Modelling pile capacity using Gaussian process regression. Computers and Geotechnics, 37(7-8), 942–947, 2010.

[7] A. H. Augustesen. The Effect of Time on Soil Behaviour and Pile Capacity. DCE Thesis, Department of Civil Engineering, Aalborg University, 2006.

[8] Pedregosa et al., 2011. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and E. Duchesnay. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830, 2011.