Data-driven site characterization – Focus on small-strain stiffness

Haris Felić[1#], *Tobias* Peterstorfer[1], *Islam* Marzouk[1], and *Franz* Tschuchnigg[1]

[1]*Graz University of Technology, Institute of Soil Mechanics, Foundation Engineering and Computational Geotechnics, Graz, Austria*
[#]*Corresponding author: h.felic@tugraz.at*

ABSTRACT

Non-linear soil behaviour adds complexity in accurate parameter selection for numerical modelling. One of these parameters is the small-strain shear stiffness. This parameter depends strongly on the soil mass density and the shear wave velocity; the latter can be determined through in-situ tests or laboratory tests. The paper focuses on training various machine learning models to predict shear wave velocity estimates based on raw data from cone penetration test soundings. Three decision tree algorithms are considered for the analysis: *XGBRegressor*, *HistGradientRegressor*, and *RandomForest*. Various data preprocessing approaches are investigated, including noise removal and outlier identification, to assess their impact on the model performance. The results indicate that different data preprocessing approaches yield significant differences in the model performances. When applied to unseen raw data from a sand site of the Norwegian GeoTest Site, the model demonstrates promising predictive capabilities and is in a good agreement with well-known correlations. This study underlines the importance of data quality and preprocessing for reliable machine learning models. To enhance transparency and reproducibility, a GitHub repository with all the used files is made available online.

Keywords: shear wave velocity, machine learning, data preprocessing, site characterization.

1. Introduction

It is well known that soil behaviour is highly non-linear, which adds complexity to the parameter selection in numerical modelling. One of these parameters is the shear stiffness that decreases with increasing strain level. This has led to the development of constitutive models which explicitly account for strain-induced shear stiffness degradation. In these constitutive models, the stress-dependent shear stiffness at very small strains is commonly incorporated as input parameter, denoted as G_0. This parameter depends on the soil mass density ρ_t and the shear wave velocity v_s; the latter can be determined through in-situ tests (e.g. seismic cone penetration tests) or laboratory tests (e.g. bender element tests) (Mayne 2014). The focus of this paper is on in-situ tests.

While an understanding of the local geology and experience with comparable sites can provide valuable insights into ground conditions, data analysis through site investigations provides more precise quantitative details of the ground conditions at a particular site (Marzouk et al. 2024). Geotechnical practice has traditionally relied on empirical methods. However, bridging the gap between data and decision-making often involves statistical tools and engineering judgement (Phoon et al. 2022a).

Machine learning (ML) has been applied to the development of prediction models for soil correlations due to its ability to extract valuable insights from large and multidimensional datasets (Phoon et al. 2022b). There has also been increasing interest in the application of ML to soil parameter determination (e.g. Zhang et al. 2022; Patino-Ramirez et al. 2023). The use of ML can increase the reliability of parameter determination and consequently improve the fidelity of numerical simulations. To reduce the uncertainty in the ML-based parameter calibration process, the Computational Geotechnics Group at Graz University of Technology (TU Graz) aims to determine soil parameters for constitutive models using ML algorithms applied to standard soil tests and in-situ tests (e.g. Erharter et al. 2023).

In this paper, supervised ML algorithms are applied using raw in-situ data (mainly from CPT tests) as training data to predict shear wave velocity estimates. The paper begins with a description of the database used for ML model training, data preprocessing, and evaluation of model metrics. The performance of the trained ML model is tested with respect to Norwegian GeoTest sites (L’Heureux and Lunne 2020).

2. Database and data preprocessing

The database for ML model training contains 1339 cone penetration tests (CPT) carried out by Premstaller Geotechnik in basins and valleys in various Alpine regions and foothills, including Austria and Germany (van Hused 2000). These basins were formed during the last glacial period, remained as lakes after the melting of the ice masses and are often filled with fine-grained sediments. As a result, their characteristics can vary greatly within a basin and are often overlaid by coarse-grained top layers. In contrast, valley fills tend to have a coarser grain size distribution and more heterogeneous subsurface properties compared to basins (Oberhollenzer et al. 2021). The database mainly contains data of silts, and sands. However, mixtures of different soil types are also available (Oberhollenzer et al. 2023). Further information can be found in Oberhollenzer et al. 2021.

The database consists of 50 seismic cone penetration tests (SCPTs) and 46 SCPTu, which forms the basis for model training. Raw data are rarely available in the quality required for “optimal” performance of a ML model (Raschka 2015). Various preprocessing techniques are explored in this paper. One of these approaches is the moving average technique, which is used to remove small-scale noise, similar to Ceccato et al. (2022). In this case, a 50 cm window is applied to cone resistance q_c, sleeve friction f_s, and friction ratio R_f. Furthermore, the performance of different ML models is investigated concerning different approaches for data preprocessing.

Table 1 presents statistical measures for various (raw) measurements of a CPT, derived from data collected from the 96 in-situ tests without and with applied moving average on the data points. The table provides insight into the dataset through parameters such as the mean value μ, standard deviation σ, and interquartile range *IQR*, which represents the distance between the 1st quartile (Q1) and 3rd quartile (Q3). The raw data of q_c demonstrates a relatively low standard deviation and interquartile range, indicating a homogenous distribution of data points. Conversely, sleeve friction and friction ratio exhibit higher standard deviation, presumably due to the influence of outliers. The interquartile range distance is relatively high for f_s and quiet low for R_f. Applying a moving average to the dataset has very small effect on the statistical measures, except for the standard deviation of R_f, which increases.

Table 1. Statistical characteristic measures for SCPT’s and SCPTu’s (*with rolling mean)

measurements	Unit	μ	σ	*IQR*
q_c	MPa	4.89	8.54	3.90
f_s	kPa	55.43	74.59	54.40
R_f	%	2.68	58.07	1.73
q_c^*	MPa	4.89	8.15	3.91
f_s^*	kPa	55.33	66.45	54.26
R_f^*	%	2.18	7.53	1.74

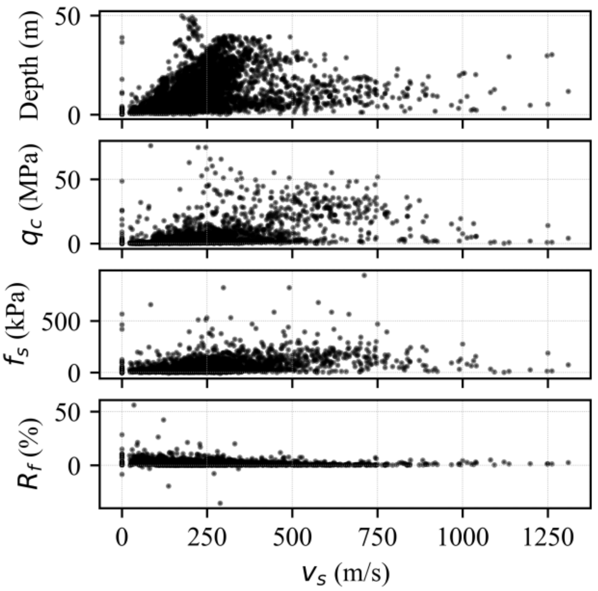

Figure 1. SCPT’s data points with moving average

Fig. 1 illustrates the distribution of all data points for the four features (**depth**, $\boldsymbol{q_c}$, $\boldsymbol{f_s}$, $\boldsymbol{R_f}$), after applying a moving average to the raw data, along the shear wave velocity (x-axis). It is important to note that negative v_s values are deleted from the database at this stage. Upon observation, no distinct correlation between the features is visible.

Input data (features) are often measured or provided in different units. Consequently, feature scaling is an essential step before training a ML model, such as for neural networks, to achieve an “optimal” model performance. Decision trees represents one class of ML algorithms that operate without the necessity for feature scaling (Raschka 2015; Ahmed Ouameur et al. 2020).

3. Machine learning models

The ML models are discussed in this section, covering feature selection, the overall workflow, model selection, and model evaluation. This paper focuses on the use of decision trees. The following ML algorithms have been used for investigation:

- *RandomForest* (Pedregosa et al. 2011)
- *HistGradientBoosting* (Pedregosa et al. 2011)
- *XGBoost* (Chen and Guestrin 2016)

RandomForest creates an ensemble of decision trees, which are built from a sample with replacement (i.e., a bootstrap sample) from the training set. When the training set for the present tree is generated by sampling with replacement, approximately one-third of the instances are excluded from the sample. This out-of-bag (OOB) data is employed to continuously obtain an unbiased estimate of the classification error as trees are incorporated into the forest (Pedregosa et al. 2011; Breiman 2001).

HistGradientBoosting is a gradient boosting framework that uses tree-based learning algorithms. In general, the main optimization target is the pseudo-residuals calculated at each iteration step. Furthermore, the computed residuals are reassigned to each instance. This framework can handle more than 10,000 samples (Friedman 2001; Pedregosa et al. 2011).

XGBoost is an optimized version of a gradient boosting framework, specifically designed to be highly efficient. In this framework, the objective function is based on a second-order approximation which considers the gradient and hessian of residuals between the true and predicted values (Chen and Guestrin 2016).

3.1. Feature selection

In ML, a model, or predictor, is a function that generates an output based on a specific input (Deisenroth et al. 2021). In this case, a feature matrix (input matrix) consists of ***depth***, $\boldsymbol{q_c}$, $\boldsymbol{f_s}$, $\boldsymbol{R_f}$. Fig. 2 illustrates exemplary the raw data of CPT ID 1243 from the database in black, while the smoothed input features - *depth*, q_c, f_s, and R_f - are depicted by red dots. Performed preliminary studies (not discussed in this paper) indicated that the use of ***depth***, $\boldsymbol{q_c}$, $\boldsymbol{f_s}$, and $\boldsymbol{R_f}$ results in a very good model performance.

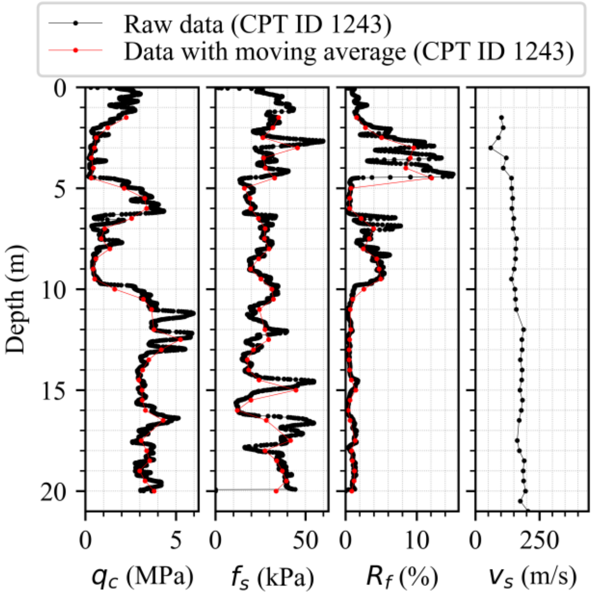

Figure 2. Raw and smoothed CPT data (CPT ID 1243)

3.2. Model training workflow

The ML workflow typically follows the steps outlined in Fig. 3, compare Deisenroth et al. (2021). To assess the model performance, a preliminary ste0p involves splitting the database into training and test sets, typically with an 80 % to 20 % ratio, respectively (Rauter and Tschuchnigg 2021; Masi 2023)

Initially, the ML model is trained using the training data and initial hyperparameters. At each workflow step, the loss is computed by evaluating an objective function that compares the ML predictions to the actual dataset. After training, the model performance can be assessed through the training loss with the training set. Subsequently, the model's generalization is assessed through a validation step, where its performance on a validation set – derived from the 80 % training data – is analysed by the validation loss. To mitigate overfitting, cross validation is used. This process iteratively partitions the data into K subsets, with K-1 used for training and the remaining one for validation. The model's performance across these K runs is averaged to measure its overall performance. A 10-fold cross validation approach is employed (Deisenroth et al. 2021).

As each ML algorithm builds on multiple hyperparameters, an optimization algorithm is necessary to obtain optimal hyperparameters based on an objective metric. In this study, Optuna is employed, an automatic hyperparameter optimization software framework designed specifically for ML tasks (Akiba et al. 2019). It assesses the training and validation loss for various hyperparameter settings to obtain optimal settings for each ML algorithm by maximizing (or minimizing) an objective function. The termination criterion for the optimization process is based on the iteration number. Details regarding the training metrics of each iteration, along with the objective function, are available in the GitHub repository (link to repository) associated with this study.

The final step is to train the test model using the training and validation sets, along with the best hyperparameters, and then to evaluate its performance using the test set. This evaluation allows for the computation of the test loss and the assessment of the generalization of the ML model. If an acceptable test loss is achieved, a final training can be performed using all data points from the database including the best hyperparameters after the optimization.

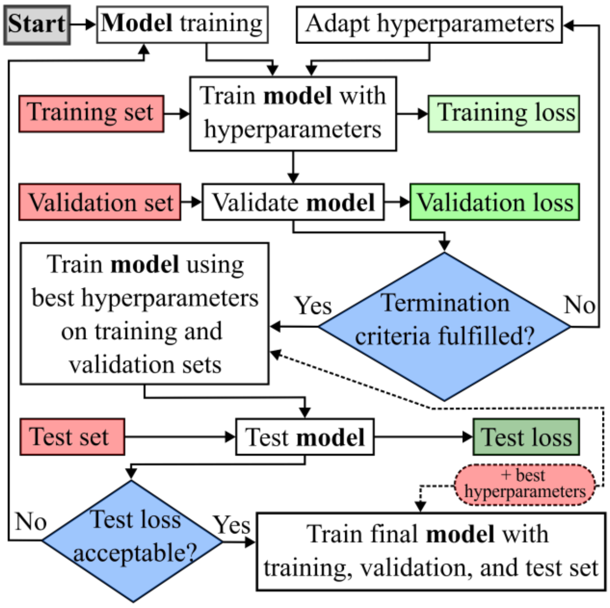
Start
Model training
Adapt hyperparameters
Training set
Train model with hyperparameters
Training loss
Validation set
Validate model
Validation loss
Train model using best hyperparameters on training and validation sets
Yes
Termination criteria fulfilled?
No
Test set
Test model
Test loss
+ best hyperparameters
No
Test loss acceptable?
Yes
Train final model with training, validation, and test set

Figure 3. Model training workflow

3.3. Model selection

ML performance relies highly on the quality and representativeness of the training dataset (Raschka 2015). Training a ML model involves adjusting internal parameters to optimize its performance on unseen input data. However, achieving satisfactory performance on familiar data (training set) may merely indicate effective memorization of the dataset, without guaranteeing good performance on unseen data (test set) (Deisenroth et al. 2021). Consequently, assessing the influence of data preprocessing on ML model performance is crucial.

In the following, the impact of outlier removal on ML performance is investigated using the method presented by Chala and Ray (2023), where outliers are identified as data points lying beyond twice the interquartile range. The input database is adapted using the following data preprocessing approaches (DPA):

- DPA 1: Raw data
- DPA 2: Moving average
- DPA 3: Removal of outliers (only for $\boldsymbol{v_s}$)
- DPA 4: Moving average and removal of outliers (only for vs)
- DPA 5: Removal of outliers (for $\boldsymbol{q_c}$, $\boldsymbol{f_s}$, $\boldsymbol{R_f}$)
- DPA 6: Removal of outliers (for $\boldsymbol{q_c}$, $\boldsymbol{f_s}$, $\boldsymbol{R_f}$, $\boldsymbol{v_s}$).

3.4. Evaluation of model performance

The upcoming section discusses the performance of each ML model and the various data preprocessing approches. For the evaluation, the coefficient of determination R^2 serves as the objective metric. The formula of R^2 is typically expressed as:

$$R^2 = 1 - \frac{\Sigma(y_i - f_i)^2}{\Sigma(y_i - \bar{y})^2} \quad (1)$$

where y_i represents the true value, f_i represents the predicted values and $\bar{y}$ represents the mean of true values. R^2 quantifies a model's predictive capacity, typically falling within the range of 0 to 1. A high R^2 value indicates a strong alignment between the model prediction and the (raw) data, while a low R^2 signifies a low alignment. Negative R^2 values suggest that the mean of the (raw) data offers a more accurate representation of the outcomes than the model prediction.

3.4.1. Performance with different data preprocessing approaches

Tables 2 to 4 present the obtained R^2 values (objective metric) on the test and training sets for all three investigated ML models and data preprocessing approaches. The R^2 value quantifies the predictive capacity of the model compared to the raw data. For instance, a R^2 value of 0.70 indicates that the model's outcome can predict 70 % of the variability in the compared variable, leaving 30 % unexplained. It is important to mention that the R^2 obtained from the test set (test loss) holds more significance than the R^2 value on the training set (training loss) in model evaluation, as the ML model should perform well on unseen data (test set), see Deisenroth et al. (2021).

As shown in Table 2, *XGBRegressor* (XGB) shows a range of the objective metrics from the test set, ranging from 0.217 for DPA 5 (removal of outliers - for $\boldsymbol{q_c}$, $\boldsymbol{f_s}$, $\boldsymbol{R_f}$) to 0.493 for DPA 2 (moving average); the corresponding R^2 values from the training set range from 0.506 for DPA 5 to 0.757 for DPA 2, respectively.

Table 2. Model performance of XGB for different data preprocessing approaches

Investigation	R^2 value from test set (1)	R^2 value from training set (1)
DPA 1	0.460	0.540
DPA 2	**0.493**	**0.757**
DPA 3	0.463	0.673
DPA 4	0.481	0.781
DPA 5	0.217	0.506
DPA 6	0.471	0.636

Tables 3 and 4 illustrate the R^2 values associated with different data preprocessing approaches using *HistGradientBoostingRegressor* (HGBR) and *RandomForestRegressor* (RFR), respectively. The objective metrics for HGBR from the test set range from 0.215 for DPA 5 (removal of outliers - $\boldsymbol{q_c}$, $\boldsymbol{f_s}$, $\boldsymbol{R_f}$) to 0.472 for DPA 3 (removal of outliers - $\boldsymbol{v_s}$); R^2 values from the training set vary from 0.304 for DPA 5 to 0.726 for DPA 4 (moving average and removal of outliers - $\boldsymbol{v_s}$). RFR model obtains the best R^2 value from the test set for DPA 3 with 0.468 and from the training set with 0.411. The worst values, with 0.238 and 0.186 from the test and training set, respectively, are obtained for DPA 5 using RFR.

The R^2 values indicate that the model's performance is highly influenced by the data preprocessing methods used in this paper. XGB shows slightly better performance than HGBR and RFR in terms of both R^2 values obtained from the test and training sets. The best objective metrics are obtained with DPA 2 (moving average) for XGB, and with HGBR and RFR for DPA 3 (removal of outliers - $\boldsymbol{v_s}$). Due to the performance, only the XGB model with DPA 2 is used in further analysis. The final model is again trained on the total database with the already determined hyperparameters, see Fig. 3. This model is then applied to a sand site in Øysand, Norway in the next section to compare the model performance with in-situ measurements.

Table 3. Model performance of HGBR for different data preprocessing approaches

Investigation	R^2 value from test set (1)	R^2 value from training set (1)
DPA 1	0.458	0.512
DPA 2	0.468	0.650
DPA 3	**0.472**	**0.642**
DPA 4	0.459	0.726
DPA 5	0.215	0.304
DPA 6	0.456	0.578

Table 4. Model performance of RFR for different data preprocessing approaches

Investigation	R^2 value from test set (1)	R^2 value from training set (1)
DPA 1	0.461	0.369
DPA 2	0.461	0.351
DPA 3	**0.468**	**0.411**
DPA 4	0.364	0.393
DPA 5	0.238	0.186
DPA 6	0.383	0.388

4. Application of the machine learning model

In this section, the performance of the trained ML model is shown using real CPT data from a sand site in Øysand, Norway. At the beginning, the origin of the data is presented. Subsequenlty, the model's performance on a sand test site is discussed. It has to be pointed out that the trained ML model is mainly based on silts, and sands (see Oberhollenzer et al. 2023)

4.1. Datamap

"Datamap" is an innovative web application designed to collect, categorize, and manage geotechnical data effectively. This application enables collaboration for researchers and practitioners to share knowledge. The web application can be accessed at www.geocalcs.com/datamap (Doherty et al. 2018).

4.2. Norwegian GeoTest Sites (NGTS)

The Norwegian Geotechnical Institute (NGI), in collaboration with various institutions including the Norwegian University of Science and Technology (NTNU), SINTEF Building and Infrastructure, the University Centre in Svalbard (UNIS), and the Norwegian Public Roads Administration (NPRA), has established five GeoTest Sites (NGTS) in Norway between 2016 and 2019 (L'Heureux and Lunne 2020). These sites represent different soil types, including clay, silt, quick clay, sand, and permafrost. For various geological conditions, in-situ measurements are provided. In this study the sand site is considered.

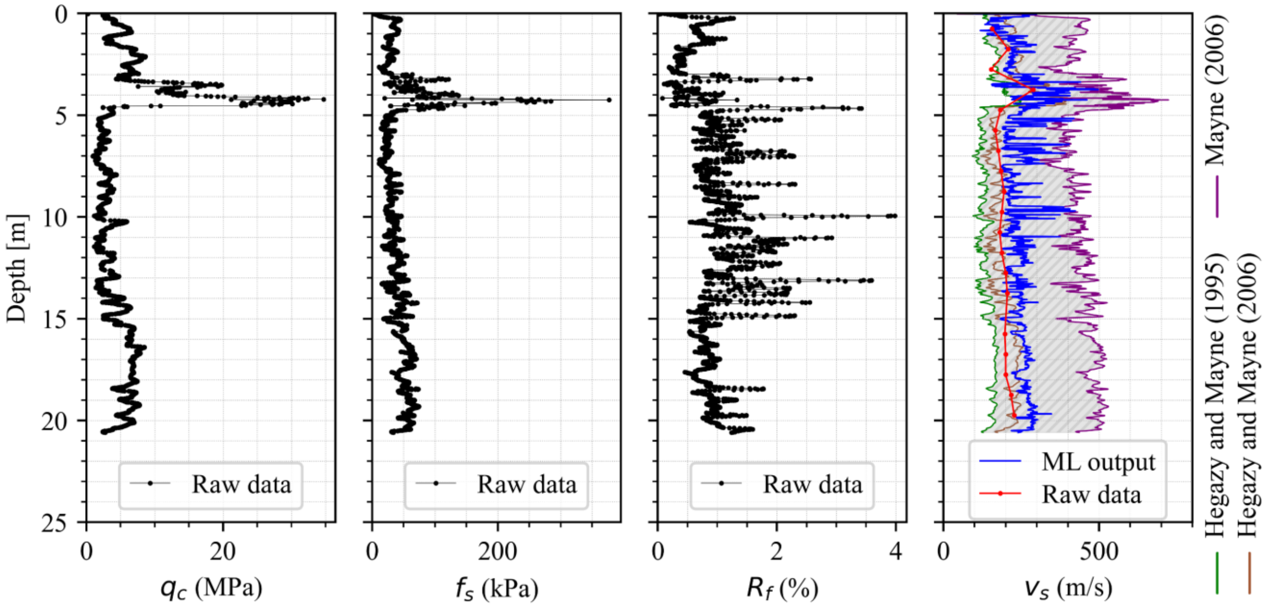

Figure 4. Raw CPT site data of OYSC35 (in black, and red), with output of XGB (in blue), and correlations (in green, brown, and purple)

4.3. Performance of machine learning model

The performance of the ML model is assessed by comparing its predictions of shear wave velocity values to the in-situ measurements from the sand site in Øysand, Norway. To improve the validity of the model, three established correlations are used for comparison:

$$v_s = 12.02 * q_c^{0.319} * f_s^{-0.0466} \qquad (2)$$

$$v_s = 118.80 * \log(f_s) + 18.50 \qquad (3)$$

$$v_s = 11.711 * q_c^{0.3409} \qquad (4)$$

Eq. (2) is more suitable for sands (Hegazy and Mayne 1995), whereas Eq. (3) (Mayne 2006) and Eq. (4) (Hegazy and Mayne 2006) can be used for various soil types (Mayne 2006; Hegazy and Mayne 2006). Investigating the influence of different data preprocessing approaches on the sand site data in Øysand is beyond the scope of this paper, as it would require retraining the ML model with the sand data, obtaining optimal hyperparameters, and reevaluating its performance. Therefore, XGB model with DPA 2 is used in this study.

Fig. 4 illustrates the raw CPT data from the sand site (CPT OYSC35 is chosen as an example), depicted by black for q_c, f_s, and R_f, and red solid lines for v_s. The output generated by the XGB model is represented by the blue solid line. Additionally, the corresponding correlations are illustrated by green, brown, and purple solid lines. The ML predictions demonstrate an overall good fit to the in-situ v_s-measurements of OYSC35 along depth and successfully capture the peak at approximately 3.50 m depth. However, due to the utilization of all raw data points from OYSC35 as input into the ML model, the output exhibits a more zig-zag behavior along depth. If the input data points were smoothed, for instance, with a moving average, the ML predictions would also exhibit a smoother output. It is worth noting that there is a slight offset between the ML model output and the raw data, especially between depths of 10 and 20 m. Eq. (2) and Eq. (3) enclose the measurements and the ML prediction; Eq. (2) tends to underestimate v_s compared to the measurements, while Eq. (3) overestimates v_s. The gray hatch between the two correlations, representing Eq. (2) and Eq. (3), illustrates a kind of uncertainty gap, which represents the uncertainty in estimating the in-situ shear wave velocity from correlations when measurements are unavailable. It is important to highlight that Eq. (4) is in a good agreement with the raw data and ML prediction.

Nevertheless, while the ML model (trained mainly with silt-sand soil types, see Oberhollenzer et al. 2021) shows good performance on the sand test site in Øysand, Norway, it is essential to acknowledge the need for more data of different soil types. Preliminary tests suggest (not shown in this paper) that the ML models indicate poor extrapolation ability when applied to a clay test site.

5. Conclusion

The primary objective of this contribution is to develop a predictive model for the stress-dependent small-strain stiffness G_0. The findings underscore the potential of ML models for the prediction of shear wave velocity estimates, as shown in Entezari et al. (2022), and Entezari et al. (2023). These predictions can be employed to determine the small-strain modulus G_0. Through the presented investigations, a dependency of the preprocessing is observed. This highlights the importance of careful data preprocessing and selection for achieving reliable model performance for each application.

Additional studies (not presented in this paper) have indicated the limited extrapolation ability of the ML models when applied to clay test sites. This underlines the necessity of adding additional training data. Future research will focus on determining further soil parameters with ML models and addressing the limitations of ML predictions for soil parameters. Despite the promising results obtained from ML studies with in-situ tests, integrating ML with laboratory tests could further enhance predictive accuracy.

While the study has provided valuable insights into data preprocessing, there are areas open for discussion and improvement. One such area is the handling of outliers, which significantly deviate from the rest of the data either in one variable (univariate) or across multiple variables (multivariate). Identifying outliers is crucial, as they can significantly impact model performance and accuracy. Another area for improvement lies in investigating bias(es) within raw data, such as the predominance of silts and sand in the database used in this study, which can influence the model performance of the ML algorithm. This aspect remains a subject of ongoing research.

A link to the GitHub repository is provided containing the Python codes utilized in this study. This repository fosters transparency, reproducibility, and facilitates further collaboration within the geotechnical research community.

Acknowledgement

The first author would like to thank Andreas-Nizar Granitzer for the discussions and review.

Data repositories

The GitHub repository is available here: github.com/harifel/ISC7_DataDrivenSiteCharacterization

References

Ahmed Ouameur, M., Caza-Szoka, M., and Massicotte, D. 2020. “Machine learning enabled tools and methods for indoor localization using low power wireless network”, Internet of Things, Volume(12), pp. 1–14. https://doi.org/10.1016/j.iot.2020.100300

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. 2019. “Optuna: A Next-generation Hyperparameter Optimization Framework”, In: KDD ‘19: The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, USA, pp. 2623–2631. https://doi.org/10.1145/3292500.3330701

Breiman, L. 2001. “Random Forests”, Machine Learning, Volume(45), (1), pp. 5–32. https://doi.org/10.1023/A:1010933404324

Ceccato, F., Uzielli, M., and Simonini, P. 2022. “Characterization of geotechnical spatial variability in river embankments from spatially adjacent SCPT”, In: Proceedings of the 5th International Symposium on Cone Penetration Testing (CPT’22), Bologna, Italy, pp. 863–869. https://doi.org/10.1201/9781003308829-128

Chala, A. T., and Ray, R. P. 2023. “Machine Learning Techniques for Soil Characterization Using Cone Penetration Test Data”, Applied Sciences, Volume(13), (14), pp. 1–20. https://doi.org/10.3390/app13148286

Chen, T., and Guestrin, C. 2016. “XGBoost”, In: KDD ‘16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco California USA, pp. 785–794. https://doi.org/10.1145/2939672.2939785

Deisenroth, M. P., Ong, C. S., and Faisal, A. A. 2021. “Mathematics for machine learning”, Cambridge University Press, Cambridge.

Doherty, J. P., Gourvenec, S., Gaone, F. M., Pineda, J. A., Kelly, R., O’Loughlin, C. D., Cassidy, M. J., and Sloan, S. W. 2018. “A novel web based application for storing, managing and sharing geotechnical data, illustrated using the national soft soil field testing facility in Ballina, Australia”, Computers and Geotechnics, Volume(93), pp. 3–8. https://doi.org/10.1016/j.compgeo.2017.05.007

Entezari, I., Sharp, J., and Mayne, P. W. 2022. “A data-driven approach to predict shear wave velocity from CPTu measurements”, In: Proceedings of the 5th International Symposium on Cone Penetration Testing (CPT’22), Bologna, Italy, pp. 374–380. https://doi.org/10.1201/9781003308829-51

Entezari, I., Sharp, J., and Mayne, P. W. 2023. “Machine Learning for CPTu Interpretation”, In: 4th International Symposium on Machine Learning and Big Data in Geoscience, Cork, Republic of Ireland.

Erharter, G. H., Nøst, H. A., Marzouk, I., Oberhollenzer, Simon, Holmen, Wilhelm, Jostad, H. P., and Tschuchnigg, F. 2023. “MLpFEM - towards Machine Learning based parameter calibration”, In: 4th International Symposium on Machine Learning and Big Data in Geoscience, Cork, Republic of Ireland.

Friedman, J. H. 2001. “Greedy function approximation: A gradient boosting machine”, Ann. Statist., Volume(29), (5). https://doi.org/10.1214/aos/1013203451

Hegazy, Y. A., and Mayne, P. W. 1995. “Statistical correlations between Vs and cone penetration data for different soil types”, In: International Symposium on Cone Penetration Testing, CPT’ 95, Linköping, Sweden, pp. 173–178.

Hegazy, Y. A., and Mayne, P. W. 2006. “A Global Statistical Correlation between Shear Wave Velocity and Cone Penetration Data”, In: GeoShanghai International Conference 2006, Shanghai, China, pp. 243–248. https://doi.org/10.1061/40861(193)31

L’Heureux, J.-S., and Lunne, T. 2020. “Characterization and Engineering properties of Natural Soils used for Geotesting”, AIMS Geosciences, Volume(6), (1), pp. 35–53. https://doi.org/10.3934/geosci.2020004

Marzouk, I., Granitzer, A.-N., Rauter, S., and Tschuchnigg, F. 2024. “A Case Study on Advanced CPT Data Interpretation: From Stratification to Soil Parameters”, Geotechnical & Geological Engineering. https://doi.org/10.1007/s10706-024-02774-9

Masi, F. 2023. “Introduction to regression methods”, In: Machine Learning (ML) in Geomechanics, Alert Doctoral School 2023, pp. 29–75.

Mayne, P. 2006. “In-situ test calibrations for evaluating soil parameters”, In: Characterisation and Engineering Properties of Natural Soils. https://doi.org/10.1201/NOE0415426916.ch2

Mayne, P. W. 2014. “Interpretation of geotechnical parameters from seismic piezocone tests”, In: Proceedings from the 3rd International Symposium on Cone Penetration Testing, Las Vegas, Nevada, pp. 47-73.

Oberhollenzer, S., Premstaller, M., Marte, R., Tschuchnigg, F., Erharter, G. H., and Marcher, T. 2021. “Cone penetration test dataset Premstaller Geotechnik”, Data in brief, Volume(34), pp. 1–11. https://doi.org/10.1016/j.dib.2020.106618

Oberhollenzer, S., Hauser, L., Marte, R.; Tschuchnigg, F., Schweiger H. F. 2023. „Herausforderung Bodenklassifizierung – Möglichkeiten und Vorteile der Drucksondierung“, geotechnik, Volume(46), (2), S. 100–111. https://doi.org/10.1002/gete.202200017

Patino-Ramirez, F., Wang, Z. J., Chau, D. H., and Arson, C. 2023. “Back-calculation of soil parameters from displacement-controlled cavity expansion under geostatic stress by FEM and machine learning”, Acta Geotech., Volume(18), (4), pp. 1755–1768. https://doi.org/10.1007/s11440-022-01698-z

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. 2011. “Scikit-learn: Machine Learning in Python”, Volume(12), pp. 2825-2830.

Phoon, K.-K., Cao, Z.-J., Ji, J., Leung, Y. F., Najjar, S., Shuku, T., Tang, C., Yin, Z.-Y., Ikumasa, Y., and Ching, J. 2022a. “Geotechnical uncertainty, modeling, and decision making”, Soils and Foundations, Volume(62), (5), pp. 1–21. https://doi.org/10.1016/j.sandf.2022.101189

Phoon, K.-K., Ching, J., and Shuku, T. 2022b. “Challenges in data-driven site characterization”, Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards, Volume(16), (1), pp. 114–126. https://doi.org/10.1080/17499518.2021.1896005

Raschka, S. 2015. “Python machine learning”, Packt Publishing Ltd, Birmingham, UK.

Rauter, S., and Tschuchnigg, F. 2021. “CPT Data Interpretation Employing Different Machine Learning Techniques”, Geosciences, Volume(11), (7), pp. 265. https://doi.org/10.3390/geosciences11070265

van Hused, D. 2000. “Geological Processes during the Quaternary”, Mitteilungen der Österreichischen Geologischen 92, pp. 135–156.

Zhang, P., Yin, Z.-Y., and Jin, Y.-F. 2022. “Machine Learning-Based Modelling of Soil Properties for Geotechnical Design: Review, Tool Development and Comparison”, Arch Computat Methods Eng, Volume(29), (2), pp. 1229–1245. https://doi.org/10.1007/s11831-021-09615-5