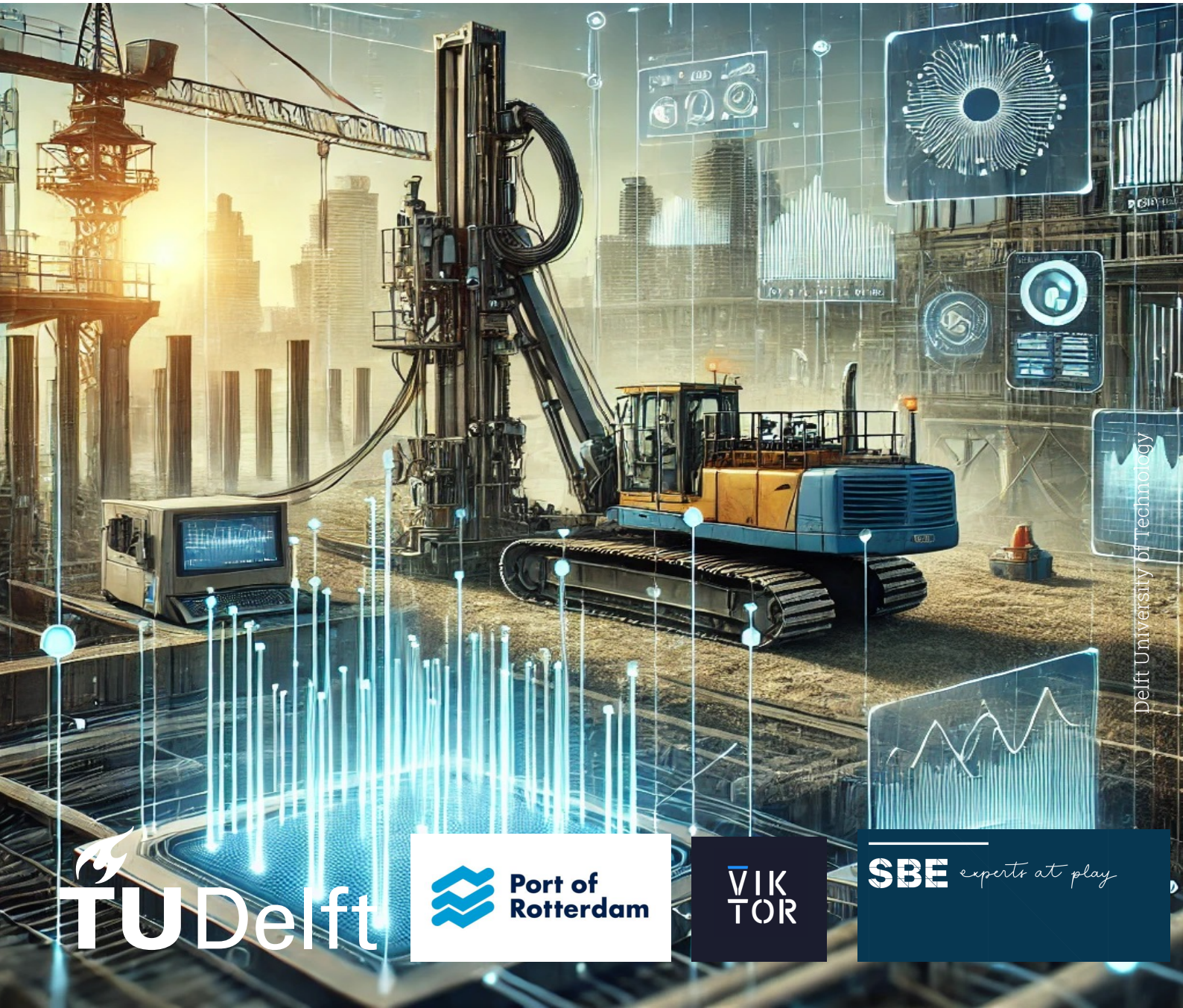
# Cone Resistance Predictions from Pile Installation Records using Machine Learning

MSc. Thesis

A.M. Yusuf



# Cone Resistance Predictions from Pile Installation Records using Machine Learning

by

A.M. Yusuf

In partial fulfillment of the requirements for the degree
**Master of Science in Cognitive Robotics at the Delft University of Technology**,
to be defended publicly on Monday January 27th, 2025.

| | |
|---|---|
| Primary Supervisor: | Dr. C. Pek |
| External Faculty Supervisor: | Dr. L. Flessati |
| Company Supervisors: | Dr. Ir. A. Roubos (PoR),<br>MSc. M. Slootweg (VIKTOR)<br>Ir. J. Putteman (SBE),<br>PDEng. P. Le Lan (SBE) |
| Student Number: | 4361504 |
| Project Duration: | 09, 2023 - 12, 2024 |
| Faculty: | Faculty of Mechanical Engineering (ME) |
| Department: | Cognitive Robotics (CoR) |

| | |
|---|---|
| Cover: | DALL-E generated cover image by ChatGPT plus with the prompt: Futuristic robotic pile-driving machines working and an engineer inspecting the construction site. |
| Style: | TU Delft Report Style, with modifications by Daan Zwaneveld and Amos M. Yusuf |

TU Delft Port of Rotterdam VIKTOR SBE experts at play

# Acknowledgments

The journey of completing this research has been one of the most challenging and rewarding experiences of my life, and I would like to take this opportunity to express my deepest gratitude to everyone who has been a part of it.

First and foremost, I would like to thank my supervisors, Dr. C. Pek, Dr. L. Flessati, and Dr. Ir. A. Roubos, MSc. M. Slootweg, Ir. J. Putteman and PDEng. P. Le Lan for their invaluable guidance, patience, and support throughout this project. Your willingness to engage in countless discussions, provide insightful feedback, and empower me as the principal researcher of my work has been instrumental in leading this project to its successful completion. Your mentorship has not only enhanced my technical understanding but also instilled in me a greater sense of confidence as I prepare for the next stages of my career.

I extend my heartfelt thanks to the companies and organizations that allowed me to immerse myself in the world of engineering. These opportunities have been transformative, enabling me to develop the technical expertise required for an engineer while also honing my interpersonal and networking skills. Collaborating with diverse teams and tackling real-world challenges has profoundly shaped my perspective on engineering and its impact.

To my family and friends, thank you for standing by me through the highs and lows of this journey, without you I would not be the person I am today. Your unwavering encouragement and understanding during periods of health challenges and life stress have been a source of strength for me. I am especially grateful for your patience and kindness as I navigated this demanding phase of my life.

This experience has been truly fulfilling, not only for the knowledge and skills I have gained but also for the personal growth it has fostered. I leave this chapter of my life with a renewed sense of motivation, armed with new insights and prepared to face the challenges of the future.

Thank you all for being a part of this journey. Finally, I leave you with the following quote that resonates with me, especially in retrospect to this research.

*"The best way to predict the future is to create it." – Peter Drucker*

*A.M. Yusuf*
*Delft, January 2025*

# Cone Resistance Predictions from Pile Installation Records using Machine Learning

A.M. Yusuf
Delft University of Technology
4361504

## Abstract

Geotechnical engineering faces challenges in subsurface characterization due to reliance on sparse site investigations and empirical correlations, which limit the accuracy of soil property estimations. This study addresses this issue by leveraging machine learning (ML) methods to predict cone penetration test (CPT) measurements using data from driven cast in-situ (DCIS) pile installations. A comprehensive dataset, including cone resistance, sleeve friction, and friction ratio measurements, was used to train and evaluate ML models—Random Forest (RF), eXtreme Gradient Boosting (XGB), and TabNet. Among these, XGB trained at a 30 cm depth resolution demonstrated a strong balance of predictive accuracy, efficiency, and granularity, achieving $R^2$ values exceeding 80% for cone resistance predictions even with limited data availability. By improving predictions at unseen locations, the study showcases the feasibility of integrating ML models into geotechnical workflows to enhance real-time decision-making, optimize construction practices, and support the development of digital twins for pile-supported structures. Interpretability methods, specifically SHAP values, provided insights into feature significance and highlighted the critical roles of depth, spatial coordinates, machine-related features and representative sampling strategies in achieving reliable predictions. These findings demonstrate ML's potential to align predictive capabilities with engineering principles, bolstering confidence in its adoption for practical applications.

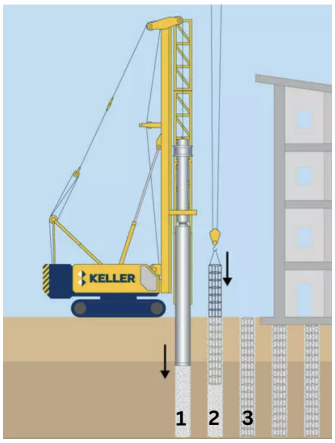
## 1. Introduction

The rapid advancement of artificial intelligence (AI) and robotics has driven significant progress in various branches of engineering. In geotechnical engineering, the integration of machine learning (ML) is a relatively recent development that has the potential to revolutionize traditional practices by enhancing predictive capabilities, such as estimating soil parameters, mapping landslide susceptibility, and integrating big data to address uncertainty and variability in soil behavior [1]. Geotechnics inherently exhibits varied and uncertain behaviors due to the complex physical processes involved in the behavior of soils under the influence of loads and hydro-mechanical coupling, making traditional modeling techniques, such as deterministic approaches, challenging in certain cases. ML has emerged as a promising solution for addressing complex geotechnical problems, and in this research, it is applied to predict cone resistance ($q_c$) from pile installation records, making it possible to update the soil profile. Using its ability to capture non-linear relationships from data, ML demonstrates superior predictive capabilities compared to conventional methods for similar applications [2]–[5]. However, while ML has significant advantages, it also faces challenges related to transparency and integration of domain knowledge [6]. The geotechnical community often views ML applications with skepticism, primarily because they are employed as "black boxes" without incorporating underlying physical principles. This lack of transparency and physical grounding can result in predictions that, while numerically accurate, may not provide insights consistent with established geotechnical principles or the underlying physics of soil behavior. For geotechnical engineers, trust in these models hinges not only on predictive accuracy but also on their ability to offer physically interpretable results that align with foundational principles, ensuring reliability and applicability in practical engineering scenarios.

Moreover, purely data-driven approaches often require large amounts of high-quality data and fail to generalize reliably outside their training range. They lack a rigorous framework grounded in the laws of physics, which is critical for modeling material interactions, particularly when pile driving. To address these downfalls, researchers for example have proposed hybrid frameworks such as Thermodynamics-based Artificial Neural Networks (TANNs), which embed thermodynamic laws directly into the network architecture. By integrating physics-based constraints, these approaches not only ensure consistency with fundamental principles but also provide deeper insights into material behavior, enhancing the reliability of predictions and their usefulness for downstream engineering decisions [7]. By adopting such combined data-driven and physics-informed frameworks, the geotechnical field can harness the predictive power of ML while maintaining alignment with established engineering practices, ultimately fostering greater trust and utility in practical applications.

In the context of this research, ML methods have proven useful in geomechanics, particularly in the evaluation of the physical and mechanical parameters of geomaterials [8]–[11]. These methods encompass a wide range of regression techniques, intelligent optimization methods ranging from swarm intelligence and evolutionary algorithms to nature-inspired computations and deep learning techniques [12]. Many studies construct data-driven models with the aid of ML, where the key to success often lies in obtaining suitable datasets that are relevant to the application, accurate to minimize prediction errors, diverse to capture a wide range of scenarios, and sufficiently large to ensure robust training and generalization of the model. These datasets

are derived from various sources, including field tests, laboratory experiments, and numerical simulations [13].




**(a)** Example of DCIS piles during installation, taken from [14].

**(b)** Illustration of various cone penetrometers used for soil data acquisition, taken from [15].

**Figure 1:** Comparison highlighting the similarity between the installation procedure of driven cast in-situ (DCIS) piles and the cone penetration test (CPT) measurements.

Building upon the advances of ML in pile foundations and geomechanics, this research aims to generate a proof of concept dataset that captures useful information about soil behavior during pile foundation installation. By analyzing data from the installation process and field tests conducted prior to installation, the study seeks to extract insights into soil characteristics post-installation. This, in turn, enables improved accuracy when predicting and assessing geotechnical properties, facilitating real-time updates to pile design, installation procedures, and/or digital twins of structures supported by these pile foundations.

This study focuses on driven cast in-situ piles (DCIS), illustrated in Figure 1a, which provides a visual representation of the key steps in the DCIS installation process. These steps are as follows: (1) driving a hollow steel casing with a base plate into the ground to the desired depth, filling it with concrete, and subsequently extracting the casing; (2) installing the rebars into the concrete; and (3) leaving the concrete pile in place. This process ensures consistent, comprehensive, and voluminous data, which aids in learning the relationship between DCIS piles and CPTs through ML methods. The initial part of the process resembles the measurement procedure of CPTs, where a cone or probe is pushed into the ground. The key difference lies in the potential deformation of the pile during DCIS installation due to the impact of the hammer, whereas CPTs involve seemingly continuous penetration with less likelihood of such deformation [16]. This resemblance offers an avenue to combine the CPT measurement and the pile installation records to create a dataset where useful insights can potentially be extracted.

An example of the probe used for these measurements, shown in Figure 1b, the specific CPT probe used in this study is the second from the right, featuring a 15 $cm^2$ cone with a 60° apex. The cone size varies according to the investigation type, smaller cones are used for shallow investigations due to their sensitivity and suitability for low-depth soils. While larger cones are employed for deeper or gravelly soils as they can withstand the higher forces required for penetration and ensure durability in dense or coarse-grained materials.

CPT measurements are less frequent than pile installations, because they require specialized equipment, operators and are time-consuming. Thus they are only performed at critical areas where detailed subsurface data is essential. The limited frequency of CPTs is also influenced by project budget making it impracticable to conduct a CPT at every desired location. Therefore, accurately predicting CPT parameters at locations without CPT data allows for the estimation of subsurface characteristics, enabling engineers to infer soil behavior and properties for downstream tasks. Installation data can also serve as post-installation CPTs, revealing changes in the substrata due to installation. Additionally, optimizing the pile driving process enhances construction efficiency by reducing noise pollution, fuel consumption and minimizing equipment wear. This, in turn, lowers costs and paves the way for autonomous robotic systems, aiming to transform construction by shifting workers' roles to supervision while significantly improving overall efficiency.

This study explores the application of ML in predicting CPT parameters (particularly $q_c$) from driven cast in-situ (DCIS) pile installation data. Using the similarities between the DCIS installation process and cone penetration tests (CPTs), the research aims to bridge the gap between AI advancements and practical geotechnical challenges. Specifically, it seeks to enhance decision-making in pile design and installation by generating accurate $q_c$ predictions at locations where CPT data is unavailable, enabling real-time updates to soil profiles, pile geometries, and installation procedures. The primary research question is

> "How well can machine learning models predict cone resistance for unseen locations using pile installation data?"

To address this question, the study investigates the following sub-questions:

1. What machine learning models are most promising for baseline performance?
2. What are these learning models basing their predictions on (explainability)?
3. How can we use ML predictions to improve subsoil estimate updates?

## 2. Related Work

Recent research in geotechnical engineering has increasingly focused on the application of machine learning techniques, particularly for enhancing soil classification methods and optimizing pile installation workflows, where traditional approaches face limitations due to the inherent complexity and variability of soil behavior.

**ML for Soil Classification**
In a review and comparison of the application of popular machine learning algorithms, such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Decision Trees (DT), to various geotechnical problems [2], these algorithms have demonstrated effectiveness in tasks such as soil and rock classification, landslide susceptibility prediction, and soil settlement forecasting. Notably, Random Forest (RF) was highlighted for its superior performance in soil classification compared to SVM, ANN, and DT, achieving higher classification accuracy due to its ability to handle complex datasets with diverse features. The datasets utilized in the study encompassed chemical and physical properties from sampled soil profiles, as well as images of the soil profiles. This analysis highlights that the choice of ML algorithm is crucial for achieving high predictive accuracy and generalization in geotechnical applications. RF's performance, particularly in soil classification, underscores the importance of selecting algorithms tailored to specific geotechnical challenges to maximize model effectiveness and practical utility.

Building on this effectiveness of ML algorithms, further studies have explored their utility to analyze specific data sources, such as CPTs. These advances suggest that ML approaches can improve soil classification tasks using raw CPT data. A notable study investigated the application of ML algorithms to predict soil classifications from raw CPT data [4]. The dataset included CPT measurements and adjacent borehole-derived soil classifications, which were used to train five ML models: logistic regression, SVM, RF, K-nearest neighbors (KNN), and extreme gradient boosting (XGBoost). Among these, the RF model demonstrated superior performance, followed by XGBoost and KNN.

This review of available research highlights RF and XGBoost as promising candidates for soil classification tasks due to their accuracy, robustness in handling complex and diverse datasets, and interpretability compared to more complex models. These qualities make them well-suited for aligning with the specific objectives of this research.

**ML for Soil Composition and Schematization**
In the area of neural networks, several studies have demonstrated their potential to improve the accuracy of soil classification from CPT data, providing geotechnical engineers with richer and more precise information about subsurface conditions. For example, a study implemented a General Regression Neural Network (GRNN) to predict soil composition [17]. Their model combined cone resistance ($q_c$) and sleeve friction ($f_s$) measurements with grain-size distributions obtained from standard penetration tests to train the GRNN. The model was tested with new CPT data and evaluated against conventional methods, such as Robertson's soil behavior classification [18] and the Unified Soil Classification System (USCS). The GRNN demonstrated an 86% success rate in classifying soils as coarse-grained or fine-grained. This success highlights the potential of neural networks as a promising ML algorithm for geotechnical applications. However, for this research, consideration must be given to a neural network architecture that is robust to smaller datasets, ensuring reliable predictions even with limited data availability while maintaining interpretability and accuracy.

Another study explored the use of Generative Adversarial Networks (GANs) to enhance subsoil schematization based on CPT data [19]. Subsoil models, also known as subsoil schematizations, represent the subsurface as layers with defined geometry, thickness, depth, and physical or mechanical properties. These models are crucial for geotechnical engineering, as they provide simplified yet informative representations of the heterogeneous and complex nature of the subsoil. Accurate subsoil schematizations guide critical decisions in infrastructure design, construction, and risk management. The study introduced SchemaGAN, a novel GAN architecture adapted from the Pix2Pix framework, to generate subsoil schematizations. SchemaGAN was trained on a synthetic database of 24,000 subsoil models and evaluated against traditional interpolation methods, such as inverse distance weighted (IDW), Nearest Neighbor (NN), and Kriging. Results demonstrated that SchemaGAN outperformed these methods by producing clearer layer boundaries and more accurate stratigraphic features. This study highlights the effectiveness of the SchemaGAN architecture in automating subsoil schematization, using CPT data to generate more accurate and detailed stratigraphic models than traditional methods, showcasing its potential for complex geotechnical applications.

**Pile Bearing Capacity & Load-Settlement Response**
Several studies have explored the use of ML techniques for predicting pile-related characteristics, such as bearing capacity and load-settlement behavior. One study developed ANN models to predict the axial capacity of driven piles and drilled shafts using CPT data [20]. Based on data from 80 driven piles and 94 drilled shaft load tests, the ANN models outperformed traditional CPT-based methods by providing more accurate predictions and eliminating simplifying assumptions about soil properties. Traditional methods often rely on fixed influence zones around the pile tip or shaft for calculating resistance, assume uniform soil layers despite natural variability, and presume linear relationships between soil properties and pile capacity. Additionally, the ANN models were translated into simplified design equations for practical use and validated against experimental data, demonstrating superior accuracy, particularly for larger pile capacities. These findings highlight the potential of ANNs for scalable, reliable, and practical pile design applications.

Similarly, a study investigated the optimal ANN model for predicting the bearing capacity of shallow foundations, particularly when trained on small datasets [21]. In this context, small datasets refer to limited experimental datasets, with some cases including as few as 6 samples, derived from specific geotechnical scenarios like shallow foundation load tests. Despite their size, these datasets are of high quality, containing precise measurements of variables such as soil unit weight, internal friction angle,

and foundation dimensions. The study compared the performance of both shallow (one or two hidden layers) and deep (more than two hidden layers) neural networks and found that ANNs with 5 to 7 hidden layers outperformed shallow neural networks when trained on such limited data, achieving higher accuracy in bearing capacity predictions. It emphasized that the number of layers had a more significant impact on prediction accuracy than the number of neurons in each layer. This demonstrates that deep ANNs can effectively predict pile bearing capacity even with small, high-quality datasets, highlighting their potential for addressing challenges in data-limited scenarios.

Further extending this potential, ANNs have been applied to predict the complete load-settlement behavior of single piles using CPT data [22]. Using a database of 499 field measurement records from 56 pile load tests, incorporating various soil types, pile types, and other geotechnical conditions. The ANN model outperformed traditional methods, achieving a correlation coefficient ($r$) of 0.991 and a Root Mean Square Error (RMSE) of 3.21 mm for settlements up to 138 mm, outperforming traditional methods [23], which exhibited larger errors. Similarly, ANN model was developed to estimate the load-settlement behavior of driven concrete piles using CPT data [24]. The study compared the ANN approach with traditional methods defined by the Russian national standard SP 24.13330 and found that the ANN significantly reduced the prediction error. Specifically, the ANN model achieved a Mean Average Percentage Error (MAPE) of 23.4%, compared to 43.4% for the traditional method.

These examples illustrate the robustness and accuracy of ANNs in predicting pile-related characteristics across various scenarios, emphasizing their role as a valuable tool for addressing both data-rich and data-limited challenges in pile foundation engineering.

## 3. Methodology

This research employs a systematic approach to perform a multi-output regression task from installation data. In Figure 3 an overview of the end-to-end workflow, encompassing data acquisition, preprocessing, model training, validation, and evaluation. Using a structured workflow, the study ensures clarity and reproducibility of the results. The data gathered from pile installations and CPTs performed in the western part of the Port of Rotterdam, where approximately 3000 piles were installed for the construction of new quay walls. The CPTs were conducted prior to pile installation. The data was collected using the Inpieq Data Logging System (IDLS) [25], a framework for recording data during pile and anchor installation. This system was further customized to efficiently store the data in the SBEGEO Standard [26], a cross-platform compatible data standard that facilitates smooth database processing. The CPT dataset was obtained from the Basisregistratie Ondergrond (BRO) datasets, available through BROloket [27].

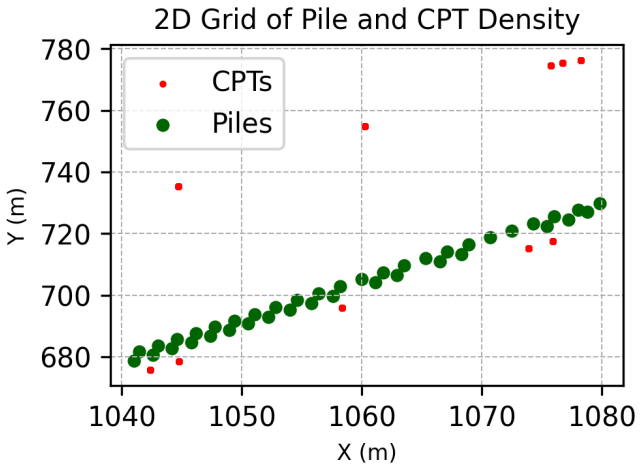



**Figure 2:** Illustration of the Pile and CPT grid layout, highlighting the dense distribution of pile installation points compared to the sparse distribution of CPT measurements.

In large-scale geotechnical projects, it is common to have extensive pile installation data but very limited site testing, in this case CPTs. This imbalance often poses challenges in accurately characterizing the subsoil due to the sparse and localized nature of the site tests. Leveraging the abundant pile installation records provides an opportunity to enhance subsoil characterization by treating these records as post-installation CPTs.

The goal is to use the fine-meshed pile installation grid to perform a multi-output prediction of cone resistance ($q_c$), sleeve friction ($f_s$), and friction ratio ($R_f$) at the pile locations based on the coarse-meshed CPTs. Figure 2 illustrates the CPTs are spaced approximately 25 meters apart. Additional CPTs are performed locally in cases of measurement errors, unexpected findings, or when the probe cannot penetrate the soil due to encountering unexpected substrata. These additional tests ensure a detailed understanding of the subsurface conditions and address any anomalies detected during the standard CPT procedures. In contrast, pile spacing during this project ranged from 3 to 5 meters, demonstrating the significantly higher density of pile installations. This higher density of pile installations offers the potential to update the soil profile by leveraging the installation data, thereby addressing the limitations of sparse CPT measurements and providing a more refined understanding of the subsurface conditions.

Furthermore, Figure 2 highlights the potential of using pile installations as post-installation CPTs to infer subsoil changes. While this research frames the problem as a multi-output regression task, the primary focus is on predicting $q_c$, as it is critical for updating soil profiles. By leveraging pile installation records, this approach aims to provide geotechnical engineers with detailed insights into substrata sequences, enhancing real-time design and installation decisions.

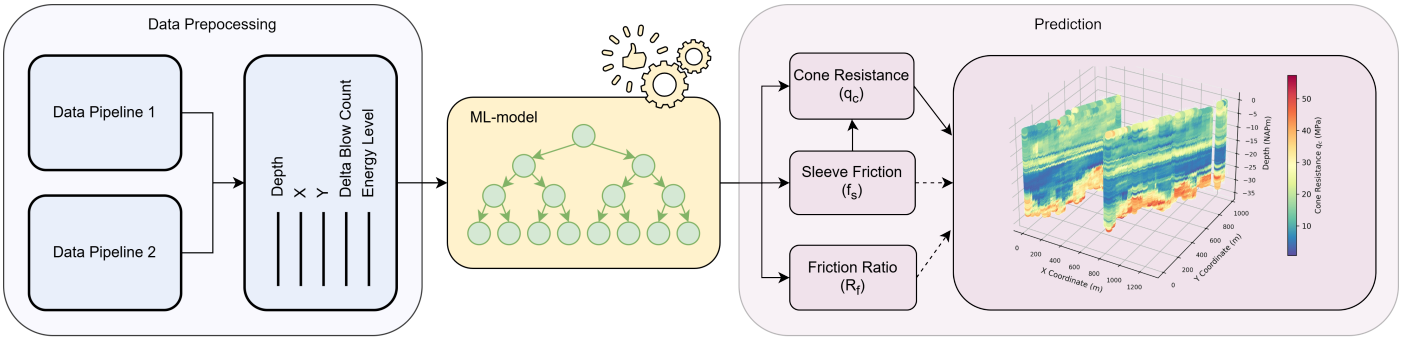



**Figure 3:** End-to-End Workflow for Multi-Output Regression.

### 3.1 Dataset Preparation

In order to prepare the data for ML model training, several preprocessing steps were applied to both the CPT data and the pile installation data. For the CPT data, the preprocessing involved parsing and converting files to CSV format, data cleaning, downsampling measurements to 10 cm intervals, and ensuring consistency in probe usage. The use of the same probe across all measurements was verified to ensure data consistency, as this information was available in the GEF metadata. This allowed us to thoroughly study whether a relationship exists between the CPT data and the pile installation records, while maintaining a consistent form-factor (for pile type and probe) across the dataset. For the pile installation data, steps included parsing and converting files to CSV format, data cleaning, feature engineering, and feature selection. The datasets were then aligned, with CPT data interpolated to match pile installation depths, creating a consistent dataset for model training. These steps were organized into two distinct data pipelines and subsequently combined into a unified dataset. The datasets were ultimately combined into a single CSV format, as this approach offers several critical advantages in the ML context. First, combining the datasets simplifies the data handling process, allowing for the creation of unified input-output pairings necessary for supervised learning. CSV format ensures compatibility with widely-used ML libraries and facilitates preprocessing, visualization, and reproducibility of the experiments. By consolidating the data into a unified structure, the model can efficiently learn the underlying relationships between input features and outputs, ultimately facilitating accurate predictions of subsurface characteristics at unseen locations.

**Data Pipeline 1: CPT Data**

The CPT data is stored in the Geotechnical Exchange Format (GEF), which is a standardized file format in the Netherlands, used for the exchange and storage of geotechnical data. For this project, a total of 355 GEF files were utilized, providing high-resolution measurements taken every 2 cm in depth (seemingly continuous with respect to depth) of soil properties and cone position. To prepare the CPT data for ML, the GEF files (see Excerpt 1) were parsed and converted into CSV files (see Excerpt 2) to facilitate manipulation and data processing (see Appendix A.1 for an explanation of pipeline 1). Furthermore, the resolution of these measurements was reduced to 10 cm intervals to match the pile installation data. Furthermore, to ensure that representative values of $q_c$, $f_s$, and $R_f$ were available at every pile location, inverse distance weighted (IDW) interpolation was employed. Using the five nearest CPTs for each pile location, IDW generated synthetic CPT data (ground-truth), enabling predictions at pile positions where CPT measurements were unavailable.

**Data Pipeline 2: Pile Installation Data**

The pile installation data includes measurements recorded at 10 cm depth intervals by the IDLS. Two pile installation methods were used to acquire this data, namely impact driving and vibratory driving. The impact driving method is the predominant standalone method used in this dataset, particularly for the entire installation process of specified DCIS piles. Approximately 70% of all impact-driven DCIS piles, equivalent to approximately 1500 piles, were installed using the diesel impact hammer, for which machine data was available. This research focuses on the machine data gathered from piles installed using the diesel impact hammer, as it constitutes the majority of the available data and closely resembles the CPT procedure in terms of soil penetration dynamics. The piles used in this study are cylindrical auxiliary tubes with inner and outer diameters of 549 mm and 609 mm, respectively. The piles were installed to depths of approximately 28 m, but this ranged from 5 to 35 m, depending on various installation challenges. For this research, only piles that were successfully installed were retained in the final dataset, with a minimum installation depth of 5 m (1,469 DCIS piles remain). Furthermore, each pile was installed with a base plate of varying diameters ranging from 680 mm to 720 mm. During data preparation, the final dataset was filtered to include only piles with base plates measuring 720 mm in diameter and 30 mm in thickness. These checks were performed during data preparation (metadata check) to ensure consistency in the dataset, enabling a reliable study of the relationship between input-output pairings when the geometry of the two data sources is consistent.

The raw pile installation data, provided in JSON format, required thorough preprocessing to ensure compatibility with the ML workflow. The JSON files, comprising metadata and direct machine logs, contained key information such as pile ID, operator, contractor, driving energy, and cumulative blow counts. Relevant examples of the metadata and

machine logs can be found in Excerpts 3 and 4. To enable easier manipulation, these files were parsed and converted into CSV format, as demonstrated in Excerpt 5. The preprocessing involved extracting essential fields, standardizing units, and addressing missing or inconsistent entries. The preprocessing also included feature engineering steps, such as creating a Delta Blow Count and Energy Level feature calculated for every 10 cm or depending on the desired depth resolution. These features were designed to capture localized changes in blow count and energy distribution, providing more granular insights into pile behavior and enhancing the dataset's applicability for ML modeling For a comprehensive explanation of the procedure, refer to Appendix A.2.

**Combined Dataset for ML**

The dataset, combining CPT and pile installation data, includes 1,469 diesel hammer DCIS piles with interpolated soil properties, namely $q_c$, $f_s$, and $R_f$ at each pile location. As seen in Excerpt 6, an example is provided of the final combined CSV file containing all information available during this project, this is later reduced to only the relevant input-output pairings in order to perform ML predictions. Although very large by geotechnical standards, this dataset remains relatively small for typical ML applications. This research frames the problem as a multi-output regression task to enable soil classification and the extraction of information related to soil stiffness and strength, in turn for use during the design-phase decisions and refining the soil model post-pile installation. However, one notable limitation is that the corrected cone resistance ($q_t$), which adjusts the cone resistance ($q_c$) for pore water effects, could not be calculated, because pore water pressure data was unavailable from the CPT measurements as these are measured using piezocone penetration tests (CPTu). Typically, the corrected cone resistance is calculated using the formula, as defined by [15]:

$$q_t = q_c + u_2(1 - a_n) \tag{1}$$

where $u_2$ represents the pore water pressure, and $a_n$ is the net area ratio (typically between 0.7 and 0.8). The absence of pore water pressure data constrains the ability to compute $q_t$, thus limiting the dataset to uncorrected cone resistance values. However, within the first 20–25 m and below 22–27 m, depending on the location in the basin, the soil predominantly consists of sand, where the excess pore water pressure ($u_2$) is expected to be negligible for practical purposes [15], being approximately hydrostatic and on the order of hundreds of kPa instead of MPa.

The input-output pairings for this dataset are as follows:

- **The input features:**
  1. **Depth (m) w.r.t "Normaal Amsterdams Peil" (NAP):** Representing the depth at which the measurements are taken during installation of the piles w.r.t a conventional reference point established in geotechnical projects.
  2. **Coordinate X (m):** The X-coordinate relative to the Rijks-Driehoek (RD) system, representing the position of each pile with respect to an established origin, based on the spatial distribution of piles and CPTs.
  3. **Coordinate Y (m):** The Y-coordinate relative to the RD system, representing the position of each pile with respect to an established origin, based on the spatial distribution of piles and CPTs.
  4. **Energy Level (kNm) per 10 cm (30 cm or 50 cm):** The energy setting of the hammer used during the installation of the piles to travel 10 cm (30 cm or 50 cm) through the subsoil.
  5. **Delta Blow Count per 10 cm (30 cm or 50 cm):** The number of hammer blows to the pile during installation, in order for the pile to travel 10 cm (30 cm or 50 cm) through the subsoil.
- **The output or target variables:**
  1. **Cone resistance $q_c$ (MPa):** The force acting on the cone, $Q_c$, divided by the projected area of the cone, $A_c$ [15].
  $$q_c = \frac{Q_c}{A_c} \tag{2}$$
  2. **Sleeve friction $f_s$ (MPa):** The frictional force acting on the friction sleeve, $F_s$, divided by its surface area, $A_s$ [15].
  $$f_s = \frac{F_s}{A_s} \tag{3}$$
  3. **Friction ratio $R_f$ (%):** The ratio, expressed as a percentage, of the sleeve friction, $f_s$, to the corrected cone resistance, $q_t$, both measured at the same depth [15].
  $$R_f = \left(\frac{f_s}{q_t}\right) \times 100\% \tag{4}$$

In this multi-regression setup, $R_f$ is treated as an independent target variable. While partially related to the other target variables, it remains independent of the input variables. This setup allows the model to leverage any natural interdependencies between the targets while maintaining the integrity of the input-output relationships.

### 3.2 ML Model Description

From Chapter 2 a shortlist of the best models for this task are identified. The main characteristics of these models must be their capability to handle tabular data (CSV format), training time, simplicity in the structure of the ML model for better interpretability, and reduced computational complexity. Furthermore, since there is no baseline on the available dataset, this research aims to identify a baseline model that performs accurately on this dataset for future developments.

**Random Forest (RF)**

The RF algorithm was selected as the baseline ML model for this study due to its effectiveness in handling complex, non-linear relationships [28], robustness to noisy data [29],

and capacity for model explainability [30]. These qualities are critical for addressing the challenges presented by geotechnical data, which often exhibit significant variability due to soil heterogeneity and diverse material properties.

Traditional methods, such as Kriging [31], while useful in estimating CPT parameters, rely heavily on predefined semivariograms and are constrained by the spacing and number of sampled data points. These limitations often lead to instability and reduced accuracy in highly variable or sparse datasets [32]. In contrast, RF effectively models such complexities by employing an ensemble of decision trees (DTs). Each tree is trained on a bootstrapped subset of the input data, with features randomly selected at each split, ensuring diverse perspectives on the data. This is known as bagging; it enhances RF's performance by reducing variance through the creation of multiple bootstrap samples, training trees independently, and averaging their predictions for regression tasks (or majority-voting for classification tasks). This ensemble approach increases generalizability and provides robust predictions, as illustrated in Figure 4a. RF's ability to combine predictions from multiple trees allows accurate estimation of $q_c$, even in scenarios with high data variability or unsampled locations [33]. RF's design is particularly suited for predicting $q_c$ at unsampled locations, as it captures non-linear dependencies between input-output pairings. Bagging further ensures robustness to noisy data and mitigates overfitting, which is common in sparse datasets. Additionally, RF's feature importance ranking aids in identifying critical predictors, enhancing interpretability and guiding the understanding of geotechnical variability.

**eXtreme Gradient Boosting (XGBoost)**
XGBoost (XGB) is a powerful gradient-boosting algorithm that has been considered for this task due to its demonstrated effectiveness in recent studies (e.g., [4]), particularly in soil classification tasks using raw CPT data. While RF serves as a robust baseline model due to its ability to reduce variance through bagging, XGB builds on this by addressing both bias and variance. It does so by sequentially building trees that correct the errors of previous ones, as illustrated in Figure 4b. In each iteration, XGB minimizes a specified loss function by calculating pseudo-residuals, the differences between current predictions and actual target values, and training the next tree to focus on these residuals. This process ensures that subsequent trees target the most significant errors in the model's predictions [35].

XGB incorporates gradient boosting, which optimizes tree training using gradient descent, and adds regularization ($L_1$ and $L_2$) to control model complexity and prevent overfitting. The model's final predictions are a summation of all trees' outputs, with each tree's contribution scaled by a learning rate (shrinkage) to prevent over-correction and ensure generalization. Additionally, XGB enhances computational efficiency through techniques such as sparsity-aware algorithms, which handle missing data effectively, and column subsampling, which accelerates training while reducing overfitting. These features enable XGB to manage noisy and sparse datasets more effectively, while improving accuracy and interpretability.

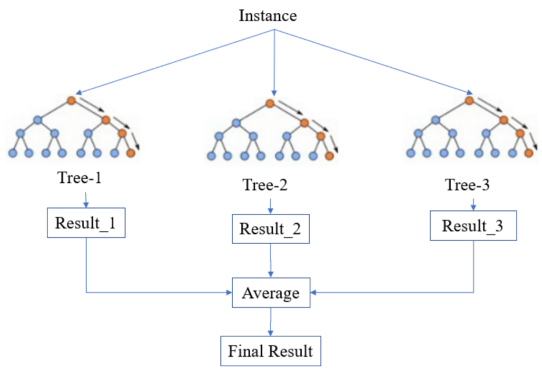
Using RF as a baseline, this study evaluates whether XGB's advanced features, such as its focus on reducing bias and its optimized training, lead to significant performance improvements. Unlike bagging in RF, which trains trees independently, boosting in XGB weights each model by its accuracy and sequentially adjusts for residual errors, creating a highly adaptive ensemble. This sequential approach allows XGB to capture intricate patterns in tabular data, making it particularly well-suited for the combined dataset.

**Attentive Interpretable Tabular Learning (TabNet):**
Neural networks have also shown success in certain geotechnical applications where complex, non-linear relationships and large datasets are involved (e.g., [19], [22]). However, traditional deep neural networks (DNNs) often underperform on tabular data compared to tree-based methods because of overparameterization and the absence of inductive bias. To address these limitations, TabNet was developed specifically for tabular data. Unlike standard DNNs, TabNet utilizes a sequential attention mechanism for feature selection, allowing the model to dynamically focus on the most relevant features at each decision step, thus enhancing interpretability and making efficient use of model capacity. As each data point passes through TabNet, the model selects a subset of features for each instance, processing only the most salient information.

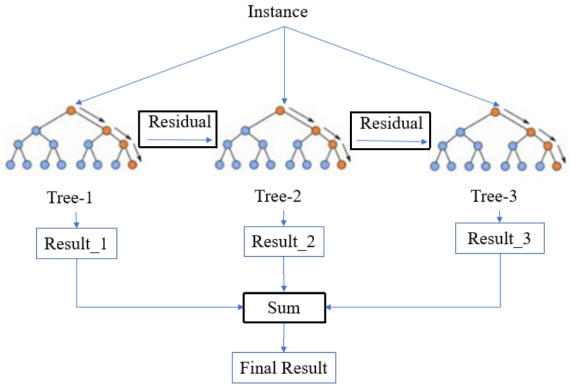
TabNet's architecture supports gradient-based optimization, enabling it to learn directly from raw tabular data without extensive preprocessing or feature engineering. Furthermore, TabNet demonstrates robustness to hyperparameter variations, as evidenced by the ablation studies in the original paper [36]. For a more in-depth understanding of the underlying principles and architectural innovations of TabNet, readers are encouraged to consult the original paper, which provides detailed explanations of the sequential attention mechanism, feature selection process, and the theoretical basis for TabNet's effectiveness on tabular data. Key parameters such as the number of decision steps ($N_{\text{steps}}$) and feature transformer units ($N_d$ and $N_a$) exhibit minimal impact on performance within reasonable ranges. This stability is attributed to its sequential attention mechanism, sparsity constraints, and the balance between shared and decision-step-specific layers, reducing the need for extensive hyperparameter tuning. TabNet outperforms ML models, including RF and XGB, across various tabular datasets. This makes TabNet a strong candidate to consider for this task.

### 3.3 Experiments on data to test ML models

In this subsection, we describe a series of experiments designed to evaluate the performance, stability, and robustness of ML models applied to the combined dataset. The goal is to identify the best-performing model for the multi-output regression task and then rigorously test its performance under varying conditions and real-world




**(a)** Illustration of Random Forest approach where multiple decision trees are trained independently using bagging, taken from [34].




**(b)** Illustration of XGBoost, which employs gradient boosting (instead of bagging) to sequentially optimize decision trees for predictions., taken from [34].

**Figure 4:** Comparison between Random Forest and XGBoost and their ensembeling techiniques. and (b) illustrates

challenges specific to geotechnical applications. The first experiment focuses on identifying the model that achieves the highest accuracy and fastest training time on the unperturbed dataset. The selected model is then tested for robustness, adaptability, and predictive power through data perturbations, feature exclusions, and region-specific training scenarios, ensuring it can generalize effectively to diverse and noisy geotechnical conditions. The experiments are conducted in the following order.

**Reduce depth resolution of data:** The combined dataset initially consists of measurements taken at 10 cm depth increments, which include significant measurement noise and outliers. To reduce these effects, the dataset's resolution was adjusted to coarser depth increments of 30 cm and 50 cm. For depth values that directly matched the new resolution, the original values were retained. For depth values not present in the dataset, linear interpolation was applied to estimate the feature values at the desired depths. This approach ensured a smooth transition between data points while maintaining the integrity of the dataset. Geotechnical experts recommended these resolutions as realistic and feasible, given the noisy nature of the data. Coarse resolutions help minimize the influence of outliers and reduce variability, particularly in blow count values, which showed limited data variation at 10 cm intervals (see Figure 15). At higher resolutions, the blow count becomes more representative of underlying soil conditions, making it a more informative feature for training ML models.

**Minimal Amount of Piles Required for Training:** As previously stated, the dataset is relatively small by ML standards, it is considered large for geotechnical engineering applications. This experiment is conducted to determine how much data is actually needed to achieve acceptable predictive performance, balancing the trade-offs between dataset size and model accuracy. By gradually increasing the size of the training set from a minimal subset and tracking performance metrics at each step, the study seeks to identify a threshold where model accuracy stabilizes. Each increment adds approximately 5% more piles to the training set, starting with approximately 74 piles (5% of the entire dataset) and progressing to the full set of 1469 DCIS piles. Since the dataset contains 134 unique segments, the smallest subsets may not fully represent all segments. However, this can potentially reveal whether there are drastic performance increases as the basin becomes fully represented in the data. Stratified sampling by segment ID is applied to ensure that as the training size increases, all segments are consistently included. This segmentation, established by geotechnical engineers on-site, ensures that model training reflects the diversity of the basin as more data is incorporated.

**Location perturbations:** The X and Y coordinates of each pile relative to the same origin (RD system) are available in the dataset. By swapping the X and Y coordinates, and rotating them by a fixed angle (45°), we test the model's ability to handle spatial transformations. Swapping the X and Y coordinates disrupts the original spatial orientation, testing whether the model relies on specific coordinate axes for its predictions or if it can adapt to changes in axis alignment. The fixed angle rotation is designed to evaluate whether the model's performance remains high when the basin is rotated but the relative distances between piles are preserved. Specifically, rotating by 45° is chosen, because it is a non-trivial angular shift away from the original axes, ensuring that the model cannot rely solely on straightforward horizontal or vertical alignments, and thus serving as a robust test of its spatial orientation adaptability.

**Feature Sensitivity Analysis:** To understand the importance of individual features, we conduct a feature sensitivity analysis by systematically excluding features to evaluate their impact on model performance. The order of removal prioritizes testing Depth as the sole input feature, followed by the evaluation of spatial information (e.g. X, and Y coordinates) and machine-related data (e.g. Energy Level and Delta Blow Count). This order reflects a logical progression to isolate the contributions of individual features and combinations.

The experiments are conducted as follows:

- Keep only Depth (Excluding all features except Depth to assess the model's performance when depth is the sole input feature).
- Remove Depth (to evaluate the significance of the primary spatial feature).
- Remove X and Y (to assess the role of spatial coordinates in predictions).
- Remove X, Y, and Depth (to evaluate the combined effect of removing all spatial features).
- Remove Delta Blow Count (to test the importance of blow count as a machine-related feature).
- Remove Energy Level (to test the importance of the hammer energy setting).
- Remove Energy Level and Delta Blow Count (to examine the combined impact of removing both machine-related features).

This progression enables an understanding of both the individual and combined contributions of spatial and machine-related features to the model's predictive power, while also examining the extent to which Depth alone suffices as a predictor.

**Zone-Specific Training:** The training data and testing data will be split into different subsets, due to the fact that the studied basin has mostly homogeneous soil conditions but can deviate at certain depths, as seen in Figure 5a. To evaluate the robustness and adaptability of the ML model, areas of Interest (AOIs) were chosen to include regions with distinctive characteristics. By identifying an AOI with low target feature values, particularly for $q_c$, a specific "weak" zone of the basin is selected. This weak zone represents challenging soil conditions that could significantly impact pile installation and design, making it critical to assess the model's performance in such scenarios. The motivation for this choice is to ensure that the model can handle spatial variability and identify patterns even in situations where limited data is available. Additionally, AOI regions have been selected along an axis where CPT measurements and pile installations were performed, ensuring a strong correspondence between the CPT data and the pile installation records. This alignment allows for a more accurate evaluation of the model's predictive capabilities by providing a consistent basis for comparing the predicted soil profiles against the true $q_c$ values derived from the CPT measurements. By focusing on these regions, the analysis aims to capture the model's ability to generalize predictions to new piles in spatially consistent environments.

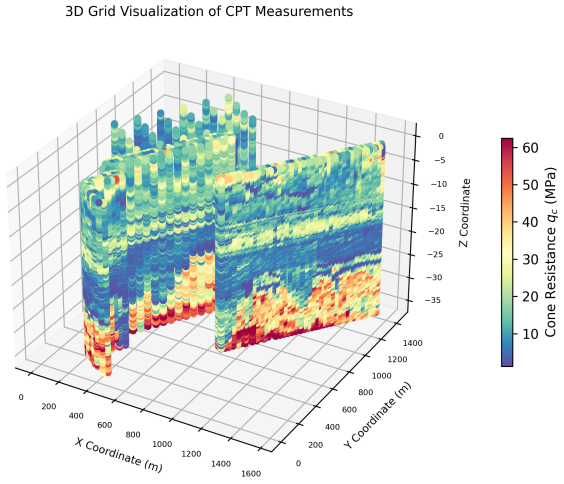
Furthermore, these AOIs allow for testing the ML model under different scenarios: predicting the soil profile of random piles within a specific area of interest, and sequentially installed piles, where the model is fed data for the first piles along a specified axis and predicts the target features for the very next pile to be installed. This dual testing approach ensures the evaluation covers both random variability and ordered dependencies. The four AOI regions studied within the available dataset were selected based on their soil conditions and spatial characteristics, namely:

- **The blue region:** Identified by geotechnical engineers as a "weak" zone of the basin, this area is characterized by smaller $q_c$ values, indicating lower cone resistance. At a depth of -25 meters below NAP, this region is distinctly visible as the "blue" area in the soil model. In Figure 5b, the area in the dotted black circle illustrates this blue region from a birds-eye view (BEV). This subset of data, containing approximately 222 DCIS piles, was carefully selected for detailed study to understand model behavior in a deviating region, enabling comparisons with other subsets.
- **The red region:** This region represents the "strong" zones of the basin, it serves as a comparative dataset to evaluate model performance against the deviating blue region. In Figure 5b, the "red" region encompasses the entire dataset minus the blue region, 1247 DCIS piles remain. This comparison clearly shows how much the accuracy improves and how it compares to a (blue) region with significantly less data, and that deviates from the rest of the basin.
- **The sequential region:** Identified by an axis, also seen in Figure 5b, along which CPT measurements were performed and followed by pile installations. Using this subset of the full dataset, we study the model's capability to predict the soil properties ($q_c$, $f_s$, and $R_f$) at the location of the next pile to be installed, based on the data from previously installed piles along the axis. This sequential dataset consists of approximately 468 DCIS piles.
- **The random region:** Similarly to the sequential region we use the same piles ($\approx$ 468 DCIS piles) along the specified axis in order to test how the ML model performs, if we feed the piles randomly. This random split has no stratified order based on segment ID, and is purely random, however ensuring that there is still sufficient representative information for the ML model to learn from, by using the commonly used splitting ratio discussed in the following Section 3.4.

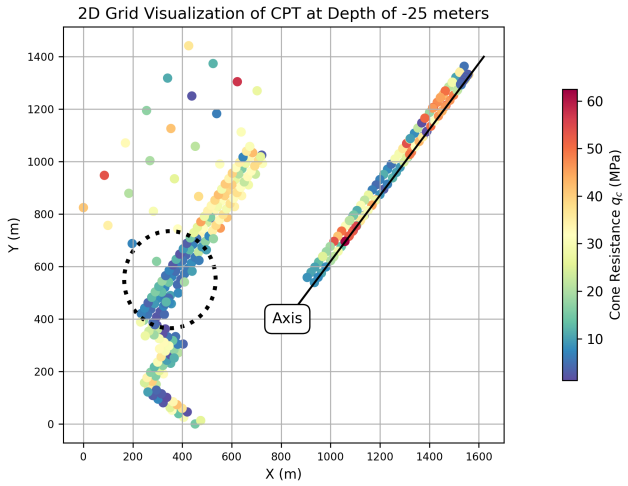
### 3.4 ML Model Development & Evaluation Metrics

The development and evaluation of the ML models followed established methodologies [37]. This section outlines the procedures used for dataset splitting, feature selection, hyperparameter tuning, and cross-validation, followed by the evaluation metrics used to quantify the model's performance.

**Dataset Splitting:** Given the regression nature of the problem and the relatively small dataset size, the data was divided into three subsets: 70% for training, 15% for validation, and 15% for testing. This split ensures that the distribution of target variables remains consistent across subsets, providing a representative sample for each phase of the modeling process. The 70-15-15 split was selected to balance the need for sufficient data for model training while preserving enough data for robust validation and testing.




**(a)** 3D spatial distribution of CPT ($q_c$) measurements in the basin prior to pile installation, showing the variation in $q_c$ values (in MPa) across the subsurface (estimated soil model).




**(b)** 2D BEV visualization of the basin at a depth of 25 meters NAP, the "blue" region illustrated by a dotted (black) circle, and the axis (black) along which the experiments for the "sequential" and "random" regions are performed.

**Figure 5:** Comparison of 3D and 2D visualizations of CPT measurements in the basin specifying a rough soil model and regions of interest for experiments. The figure emphasizes spatial variability in $q_c$ values and regional differences.

Alternative ratios, such as 60-20-20, were considered but were deemed less suitable given the relatively small size of the dataset, as reducing the training set further could negatively impact the model's ability to learn. Similarly, a larger test set would risk insufficient representation in training and validation, potentially compromising model generalization. This approach emphasizes that the choice of data splitting strategy significantly impacts model performance and should be tailored to the dataset's specific characteristics and modeling objectives [38]. The same splitting strategy was applied to both the entire combined dataset and zone-specific subsets for consistency.

**Feature Selection:** Feature selection was guided solely by domain expertise to identify the most relevant predictors for the ML model. This approach aligns with the objective of leveraging the data in its raw or minimally processed form, minimizing the need for extensive preprocessing. By focusing on features deemed most impactful by geotechnical experts, the study aims to maintain the integrity of the dataset while ensuring practicality and reproducibility. This approach underscores the importance of domain knowledge in guiding feature selection, particularly in engineering applications where physical interpretations of data are critical.

**Hyperparameter Tuning:** A grid search approach was employed to identify the optimal hyperparameters for the default best-performing ML model. This process systematically evaluated combinations of hyperparameter values to maximize model performance while minimizing computational cost. The grid search was performed by defining a range of possible values for each hyperparameter (brute force approach), and evaluating the model for each unique set of hyperparameters. See Excerpts 7 and Excerpts 8 for the exact hyperparameters and values considered.

**Cross-Validation:** To evaluate the generalization ability of the model, 5-fold cross-validation was employed. In this approach, the dataset was randomly divided into five subsets of approximately equal size. For each fold, one subset was held out as the validation set, while the remaining subsets were used for training. The process was repeated for all five folds, and the mean performance across the folds was recorded as the overall estimate of model performance. It effectively balances computational efficiency with robust error estimation, as it averages over multiple train-validation splits to reduce variability [39]. Increasing the number of folds beyond five would likely have provided little additional insight into the model's robustness, while increasing computational cost.

**Evaluation Metrics:** To assess the performance of the developed ML model, we employed regression-specific metrics that provide insights into how well the model predicts target variables from the combined dataset.

***Coefficient of Determination ($R^2$:)***
$R^2$ quantifies the proportion of variance in the target variable that is explained by the predictors. A higher $R^2$ value indicates a better fit between predicted and observed values. This metric is particularly relevant as it considers the variability in the ground truth values, providing a comprehensive measure of model accuracy [40].

***Root Mean Squared Error (RMSE):***
RMSE measures the magnitude of prediction errors, with greater emphasis on larger errors due to the squaring operation. It is particularly suitable for cases where significant deviations from the true values are undesirable [41].

***Mean Absolute Error (MAE):***
MAE calculates the average absolute difference between predicted and observed values. Unlike RMSE, MAE treats

all errors equally, making it less sensitive to outliers [42]. This metric provides a robust and interpretable measure of overall predictive accuracy.

***Training Time:***
The time required to train the ML model was recorded to assess computational efficiency, particularly in the context of real-time applications. The hardware used for training is detailed below:

- **CPU:** AMD Ryzen™ 9 7945HX processor
- **GPU:** NVIDIA® GeForce® RTX™ 4070

### 3.5 Model Explainability

Lastly, for the explainability aspect of the models, two approaches are taken in order to gain insights into what is leading a specific model to make its predictions and if this holds true, based on the information provided by the domain experts. The methods used for explainability are the following.

***Feature Importance:***
All three shortlisted models incorporate built-in feature importance functions, allowing for the ranking of features based on their effectiveness in minimizing error during inference. This provides a high-level overview of the features that contribute the most to the model's predictive power. In the case of tree-based models (e.g., RF or XGB), feature importance is often derived from the reduction in impurity achieved when a feature is used to split a node. Specifically, every split in the tree decreases a measure of impurity (e.g., Gini index), and these reductions are aggregated across all trees to produce a global importance score for each feature. However, these importance scores can be biased towards features with many levels in categorical variables or with numerous unique values in continuous features, which is particularly relevant for this dataset since the target features are continuous and contain many unique values [33], [43], [44]. Thus, these scores need to be compared to a method that can accurately represent and aggregate the impact of the feature on global and individual predictions.

***Shapley Additive Explanations (SHAP) Values:***
SHAP values provide an approach to feature importance grounded in cooperative game theory [45], that uniquely satisfies three desirable properties in the context of feature attribution:

- **Local Accuracy**: The sum of SHAP values equals the model's output for a given input, ensuring that the explanation fully accounts for the prediction.
- **Missingness**: Features with no impact on the prediction receive a SHAP value of zero, maintaining interpretability by excluding irrelevant features.
- **Consistency**: If a model changes such that a feature's contribution increases or remains the same, its SHAP value cannot decrease, ensuring logical consistency in the explanation.

Together, these properties ensure that SHAP provides a unique and mathematically consistent solution for additive feature importance. SHAP assigns each feature an importance value reflecting its contribution to the model predictions. SHAP values are derived from Shapley values, calculated using the following formula:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[ f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right] \quad (5)$$

where:

- $\phi_i$ is the SHAP value for feature $i$ in model $f$ for a given instance $x$,
- $S$ is a subset of all features except $i$,
- $f_S(x_S)$ is the model prediction using only the features in $S$.

This formula captures the marginal contribution of each feature by averaging its effect across all possible subsets of features, ensuring fairness and consistency in attribution. SHAP provides two levels of interpretability, namely:

- **Global Interpretability**: SHAP calculates the average marginal impact of each feature across the entire dataset, while incorporating theoretical consistency. This robustness makes SHAP a reliable tool for understanding how each feature influences overall model behavior, while removing the bias inherent to built-in feature importance .
- **Local Interpretability**: SHAP also offers per-instance explanations, showing the contribution of each feature to individual predictions. This is particularly valuable for examining specific cases, such as the prediction of an individual pile, where unique feature values influence the outcome. Instance-specific explanations enable visualization of the marginal impact of each feature on specific predictions.

The dual perspective offered by SHAP enables comprehensive insights into feature importance, providing both global and local interpretability. However, SHAP's computational complexity can be a significant limitation, especially for large datasets or complex models, as it requires evaluating numerous feature combinations. Additionally, SHAP assumes feature independence, which may lead to inaccuracies when features are highly correlated. Compared to other explainability techniques, such as LIME (Locally Interpretable Model-agnostic Explanations [46]), SHAP offers a more theoretically grounded approach based on Shapley values from cooperative game theory. While LIME is computationally cheaper and faster due to its use of local surrogate models, it can be less consistent and lacks the theoretical guarantees of SHAP. To balance these trade-offs, SHAP analysis is applied to the best-performing model among the shortlisted candidates, focusing on local interpretability to analyze predictions for specific instances. This choice ensures robust and theoretically sound insights into the model's behavior, despite the additional computational cost.

### 3.6 Major Assumptions and Considerations

Before proceeding to the next chapter, this section outlines the key assumptions made in this study to guide the development of a baseline model capable of effectively performing the intended multi-output regression task.

- **Hyperparameter Selection:**
  The optimal hyperparameters, determined through grid search, are consistently applied across all experiments. See Excerpt 10 for the specific hyperparameter values used for this research.
- **Data Source:**
  Only data from impact-driven DCIS piles installed using a diesel hammer are considered in this study. Other methods or data sources are excluded.
- **Experiments:**
  After determining the optimal resolution, all specified location perturbations are tested to evaluate their effect on predictive accuracy. Feature sensitivity analysis is performed to assess the importance of individual features, and zone-specific training is conducted to examine the influence of data distribution and installation order on the prediction of cone resistance.
- **Input Features:**
  The features consistently used across all experiments are Depth, X, Y, Energy Level, and Delta Blow Count, except for the feature sensitivity analysis.
- **Depth Range of Piles:**
  Only piles installed to depths ranging from 5 meters to 35 meters NAP are included in this study. Piles not meeting the minimum installation depth of 5 meters relative to NAP are excluded from the analysis.
- **Pile Installation Data Relation:**
  It is assumed that there is a relationship between Delta Blow Count and Energy Level, as contractors adjust energy levels to maintain a consistent blow count throughout the installation process. While this relationship is acknowledged, its quantification is beyond the scope of this research.

## 4. Results and Discussion

This chapter presents the results of predictive modeling for cone resistance using ML methods based on pile installation data. The outcomes are evaluated following the experimental framework outlined in Chapter 3. Key findings include baseline performance derived from two independent data sources and feature importance insights at both global and local levels. Figure 6 compares measured and predicted CPTs in a 3D grid, demonstrating the model's capability to capture subsurface variations effectively. This visualization highlights not only the model's predictive accuracy but also its potential to generate a more refined and representative soil profile post-installation compared to the original CPTs themselves. By leveraging pile installation records as post-installation CPTs, the model enhances the understanding of subsurface conditions, addressing limitations in the original CPT data and providing valuable insights for real-world geotechnical applications.

### 4.1 Performance on Different Depth Resolutions

Table 1 compares the $R^2$ performance metric of multi-regression models RF, XGB, and TabNet across different depth resolutions (10 cm, 30 cm, and 50 cm) and hyperparameter configurations. Results, derived from a 5-Fold Cross-Validation (CV), identify the optimal resolution and hyperparameters for accurately predicting $q_c$.

| Model Configuration | $R^2$ (10 cm) | $R^2$ (30 cm) | $R^2$ (50 cm) |
|---|---|---|---|
| **Random Forest (RF)** | | | |
| Default Parameters | 75.40% | **87.12%** | 86.10% |
| Training Time (s) | 119 | 40.66 | 23.74 |
| Best Parameters | 73.67% | 84.76% | 83.92% |
| Training Time (s) | 19 | **6.41** | 3.67 |
| **XGBoost (XGB)** | | | |
| Default Parameters | 71.33% | 83.34% | 83.07% |
| Training Time (s) | 1.41 | 0.72 | 0.5 |
| Best Parameters | 74.29% | **86.57%** | 85.74% |
| Training Time (s) | 5.21 | **2.33** | 2.67 |
| **TabNet** | | | |
| Default Parameters | 65.75% | 74.14% | 72.38% |
| Training Time (s) | 2056 | **576** | 269 |
| Best Parameters | 69.26% | **80.37%** | 77.04% |
| Training Time (s) | 1751 | 586 | 302 |

**Table 1:** $R^2$ under different resolutions, default and best (hyper)parameter configuration for predicting $q_c$ (5-Fold CV)

From Table 1, we observe that 30 cm resolution consistently provides the best balance between model accuracy and computational efficiency. Although RF achieves the highest $R^2$ with default parameters (87.12%), its training time is significantly longer (40.66 s) compared to XGB. When optimized hyperparameters are considered, XGB delivers the best overall performance, achieving an $R^2$ of 86.57% with a training time of just 2.33 seconds, making it the most practical choice for real-time applications requiring quick updates and predictions.

A 30 cm depth resolution represents an optimal trade-off between granularity and practical applicability. Resolutions finer than 30 cm, such as 10 cm, can introduce noise and fail to reliably capture distinct soil layers, while coarser resolutions, like 50 cm, risk overlooking critical subsurface variations. The 30 cm resolution balances capturing essential subsurface details with manageable data volumes and computational demands. While TabNet performs the worst among the models, this is likely due to the limited dataset size, which can hinder the performance of neural network-based approaches. TabNet's computational complexity and data-hungry nature make it less suited for this dataset compared to tree-based methods like XGB and RF.

Based on these findings, the remainder of the results section will focus on the XGB model using its best hyperparameters and a 30 cm resolution. This configuration provides the best combination of accuracy and computation time, aligning with the objectives of this research. The optimized hyperparameters for XGB, determined through grid search

optimization, can be seen in Excerpt 10). For a detailed comparison of the results of different models on the remaining target variables ($f_s$ and $R_f$) and at all resolutions, including their performance with the best hyperparameters, see Appendix C.1.

### 4.2 Minimum Number of Piles Required for Training

A $R^2$ score of approximately 80% is considered acceptable by geotechnical engineers, as it provides a more accurate estimate of $q_c$ than the empirical relations typically used in the initial pile design phase in soil mechanics. This higher accuracy is likely achieved because the estimates are derived from installation data.

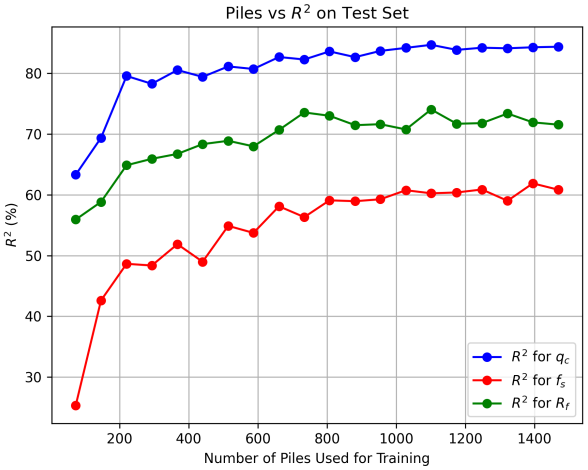



**Figure 7:** Performance of XGBoost trained on increasing numbers of piles, demonstrating the trade-off between the amount of data and prediction accuracy.

As shown in Figure 7, around 200 piles are required to achieve an $R^2$ score close to 80% for cone resistance ($q_c$). Each point on the plot represents a 5% increase in the number of piles used for training, with the smallest subset including 74 piles (5% of the dataset) and the largest encompassing 1469 piles (100% of the dataset). Stratified sampling ensures that samples from all segments of the basin, as defined by geotechnical engineers, are included in the training process. This approach guarantees the training data effectively represents the entire basin's variability.

The primary focus is on $q_c$, the figure also demonstrates the model's predictive performance for $f_s$ and $R_f$, highlighting its versatility in capturing additional target variables. These findings show that even smaller subsets of training data can provide acceptable performance, particularly in localized areas of the basin. See Appendix C.4 for the results of RF and TabNet.

### 4.3 Performance on Location Perturbed Data

Under location perturbations, we focus on the X and Y coordinates of the piles. As discussed in Chapter 3.3, two main perturbations are performed. Each perturbation tests the model's robustness to changes in spatial orientation and its reliance on consistent coordinate axes.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) |
|---|---|---|---|
| No Perturbation | 84.88% | 2.73 | 4.14 |
| Flipped X and Y Coordinates | 84.88% | 2.73 | 4.14 |
| **Rotated X and Y (45°)** | **84.94%** | **2.73** | **4.13** |

**Table 2:** XGBoost performance metrics ($R^2$, MAE, and RMSE) for predicting $q_c$ at a 30 cm resolution under various perturbations of the X and Y coordinates.

Table 2 presents the model performance under each perturbation scenario. The high $R^2$ values (above 84%) and low errors (MAE and RMSE) in both the *No Perturbation* and the *Flipped X and Y Coordinates* settings suggest that simply swapping axes does not adversely affect the model. This indicates that the model effectively learns the underlying spatial relationships (numerically) without strict dependence on a particular axis labeling. Notably, *Rotated X and Y (45°)* shows a marginal improvement in $R^2$ (84.94%) and a slightly lower RMSE (4.13 MPa) compared to the baseline. This outcome implies that introducing a consistent rotation can sometimes yield a more uniform numerical representation of the site, potentially helping the model isolate relevant patterns. In other words, a fixed-angle rotation may reorient the data in a way that reduces collinearity or otherwise improves how the model captures spatial variation.

### 4.4 Impact of Feature Removal on Model Performance

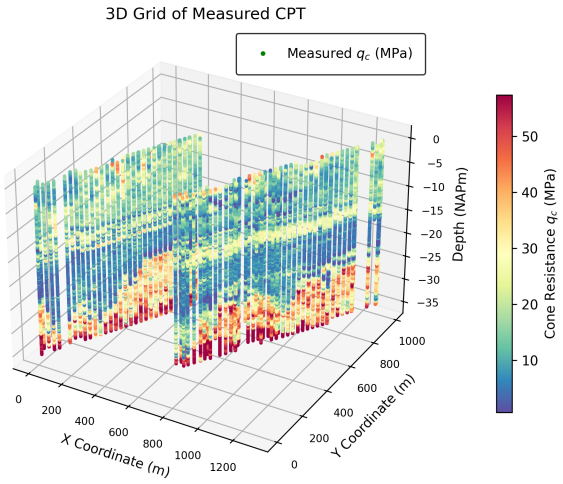
This section presents the results of the feature sensitivity analysis, focusing on the impact of excluding key features from the dataset. Depth, expected to be the most informative feature, serves as the baseline to assess how the model compensates with other features.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) |
|---|---|---|---|
| **All Features** | **84.89%** | **2.74** | **4.14** |
| Keep only Depth | 48.52% | 5.48 | 7.64 |
| Remove Depth | 58.30% | 5.15 | 6.88 |
| Remove X and Y | 69.27% | 4.17 | 5.90 |
| Remove X, Y and Depth | 57.24% | 5.32 | 6.97 |
| Remove Delta Blow Count | 84.59% | 2.78 | 4.18 |
| Remove Energy Level | 84.75% | 2.73 | 4.16 |
| Remove Energy Level and Delta Blow Count | 84.46% | 2.78 | 4.20 |

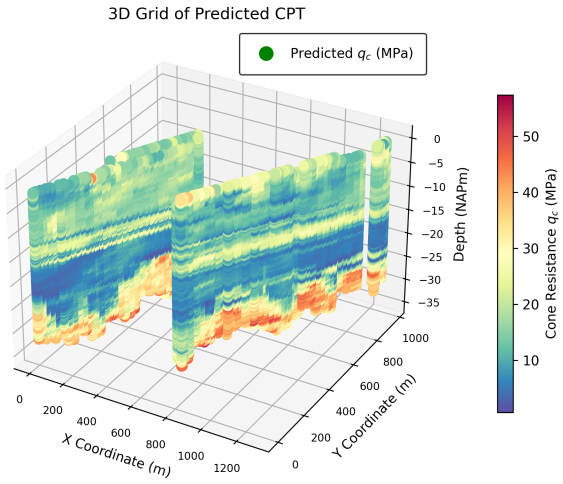
**Table 3:** XGBoost performance metrics ($R^2$, MAE, and RMSE) for predicting $q_c$ at a 30 cm resolution with different feature configurations.

**Key Observations from Feature Removal:**

- **Depth as the Most Critical Feature:** Depth is confirmed as the most important feature for predicting $q_c$, reflecting its strong correlation with soil variability. Removing depth results in a substantial drop in performance ($R^2$ = 58.30%, MAE = 5.15 MPa, RMSE = 6.88 MPa), indicating that depth provides foundational information about subsurface characteristics.




**(a)** 3D grid of measured CPTs that are compared to the final prediction of the XGB model.




**(b)** 3D grid of predicted CPTs by the XGB model.

**Figure 6:** Comparison of measured and predicted CPTs visualized in 3D grids. The left subfigure shows the measured CPTs, while the right subfigure presents the CPTs predicted by the XGB model.

However, the *Keep only Depth* configuration performs even worse ($R^2$ = 48.52%, MAE = 5.48 MPa, RMSE = 7.64 MPa), suggesting that while depth is essential, it cannot alone explain the complex interactions governing $q_c$. This highlights the complementary nature of depth and other features in capturing intricate subsurface behaviors.

- **Spatial Features (X, Y, and Depth):** The X and Y coordinates provide crucial spatial context for $q_c$ predictions, as seen from the moderate reduction in performance when they are removed ($R^2$ = 69.27%, MAE = 4.17 MPa, RMSE = 5.90 MPa). This reduction illustrates that spatial information helps contextualize depth-based variability by accounting for lateral soil heterogeneity. When X, Y, and depth are all excluded, the model's performance significantly declines ($R^2$ = 57.24%, MAE = 5.32 MPa, RMSE = 6.97 MPa). This indicates that while machine-related features (e.g., Delta Blow Count and Energy Level) contribute to predictions, they cannot fully replace the spatial and depth-related information necessary for robust modeling.
- **Machine-Related Features (Delta Blow Count and Energy Level):** Removing Delta Blow Count or Energy Level individually results in minimal reductions in model accuracy ($R^2$ = 84.59% and $R^2$ = 84.75%, respectively). This suggests a high degree of interdependence between these features, as contractors adjust energy levels to maintain consistent blow counts during pile installation. Even when both are removed together, the model achieves a comparable performance $R^2$ = 84.46%. The model's ability to compensate for their absence highlights its effectiveness in leveraging the remaining inputs.
- **Model Robustness:** The model's resilience, as shown by maintaining $R^2 > 84\%$ despite the removal of machine-related features, underscores its robustness and adaptability. However, this robustness may partly stem from preprocessing techniques, such as IDW interpolation, which preserve critical patterns but may also introduce biases. To validate these findings, testing alternative interpolation methods (e.g., Kriging or Radial Basis Functions (RBF)) is recommended, particularly for geotechnical applications where data availability or reliability may vary.
- **Error Analysis through MAE and RMSE:** The MAE and RMSE metrics provide complementary insights into model performance. The difference between MAE and RMSE across configurations (e.g., RMSE = 4.14 MPa and MAE = 2.74 MPa for all features) indicates that most errors are minor, with few significant outliers. Even with critical features like depth removed (RMSE = 6.88 MPa, MAE = 5.15 MPa), the model primarily struggles with capturing fine-grained variability rather than gross inaccuracies. The higher RMSE values confirm that losing key features like depth disproportionately impacts the model's ability to capture larger deviations in $q_c$ predictions.
- **Complementary Role of Features:** The performance contrast between the *Keep only Depth* ($R^2$ = 48.52%) and *All Features* ($R^2$ = 84.89%) configurations highlights the critical interplay among features. While depth is foundational for capturing vertical variability in soil properties, it alone cannot explain the complex subsurface interactions. Remaining spatial features (X and Y) provide lateral context, accounting for soil heterogeneity across locations, as evidenced by the performance decline ($R^2$ = 69.27%) when they are excluded. Similarly, machine-related features like Delta Blow Count and Energy Level contribute nuanced information about soil resistance during pile installation, likely allowing the model to better interpret subsurface conditions. The results suggest that the combination of spatial, and machine-related features enriches the model's understanding of subsurface behavior, with each feature set compensating

for the limitations of others. This complementary relationship ensures robust predictions even when some features are unavailable, although the influence of IDW interpolation warrants further exploration.

| Feature | Built-In Importance (%) | SHAP Importance (Value) |
|---|---|---|
| Delta Blow Count | **58.0** | 0.9 |
| Depth | 20.0 | **4.0** |
| X | 8.0 | 0.4 |
| Energy Level | 8.0 | 1.3 |
| Y | 6.0 | 3.2 |

**Table 4:** Comparison of Built-In Feature Importance and SHAP Feature Importance for all Features on the entire combined dataset.

Additionally, Table 4 illustrates the differences between the global feature importance (GFI) derived from the built-in method and SHAP-derived global importance. While the built-in method assigns 58% of predictive capability to Delta Blow Count, SHAP ranks Depth (4.0) as the most influential feature, followed by Y (3.2), Energy Level (1.3), and Delta Blow Count (0.9) and X (0.4). SHAP-derived importance highlights localized contributions and feature interactions, providing a more comprehensive assessment that aligns with the findings from the sensitivity analysis.

Key differences between GFI and SHAP importance reflect their underlying methodologies. GFI overemphasizes features that frequently split high in the tree structure, such as Delta Blow Count, which appears as the most important feature in GFI but ranks lower in SHAP. In addition, its redundancy with Energy Level, as observed in the sensitivty analysis, and by the similarity in their SHAP values, indicates that the model detects their interdependence. SHAP captures the marginal contributions and interactions of features, examining local SHAP values, rather than solely focusing on their global impact, reveals the contextual importance of machine-related features, discussed in Section 4.5.

The elevated SHAP importance of Y (3.2) compared to its GFI (6%) highlights the value of spatial context in specific regions of the dataset. This complements sensitivity analysis, where the removal of X and Y caused a notable performance drop ($R^2$ = 69.27%). Similarly, SHAP reveals nuanced contributions of Energy Level (1.3) and Delta Blow Count (0.9), underscoring their interplay in modeling soil resistance during pile installation. These localized insights, absent from GFI, align with the observed robustness of the model in compensating for missing features.

SHAP's ability to uncover localized impacts and interactions also provides validation for the model's robustness under perturbation scenarios. Although GFI highlights Depth's importance, SHAP further confirms its crucial role across predictions, even in scenarios where complementary features are absent. This insight aligns with the performance drop when Depth is removed ($R^2$ = 58.30%) and its inability to independently explain all variability ($R^2$ = 48.52% for *Keep only Depth*). By integrating SHAP, the nuanced contributions of features like Y, Energy Level, and Delta Blow Count become evident. These findings highlight the need for complementary interpretability methods, as SHAP augments the global perspective provided by GFI with a localized understanding of feature interactions and contributions. For further analysis of feature importance in the different scenarios, see Appendix C.5.

### 4.5 Zone-Specific Performance and Explainability

After analyzing the most important features of the dataset and their influence on the prediction capabilities of the XGB model, further tests were conducted to evaluate the specific training data. This involved splitting the basin into distinct zones: blue, red, sequential, and random regions. These subsets were extracted from the entire dataset (see Figure 5b). The prediction results for each region are summarized in Table 5, which also includes the baseline performance from the previous section for comparison. This allows us to identify performance increases and gaps across the zones.

| Zone | $R^2$ | MAE (MPa) | RMSE (MPa) |
|---|---|---|---|
| Baseline | 84.89% | 2.74 | 4.14 |
| Blue Region | 77.69% | 2.85 | 4.56 |
| **Red Region** | **86.34%** | **2.65** | **3.98** |
| Sequential Region | 54.59% | 4.39 | 6.23 |
| Random Region | 85.86% | 2.72 | 4.16 |

**Table 5:** XGBoost performance metrics ($R^2$, MAE, and RMSE) for predicting $q_c$ at a 30 cm resolution for different training zones in comparison to the baseline.

From Table 5, the following key takeaways can be drawn:

- **Sensitivity to Data Distribution:** The significant performance variation across regions, particularly the blue and sequential regions, highlights the model's sensitivity to data distribution and variability. This underscores the importance of ensuring representative training datasets to achieve robust predictions across all zones.
- **Limitations in Weak Zones:** The seemingly poor performance in the blue region ($R^2$ = 77.69%) indicates that the model struggles with regions characterized by high variability and lower $q_c$ values. This suggests a need for targeted feature engineering or additional data collection in weak zones to better capture their unique characteristics.
- **Improved Performance in Homogeneous Zones:** The superior performance in the red region ($R^2$ = 86.34%) suggests that the model excels in more homogeneous datasets with higher $q_c$ values. This finding supports the use of stratified data preprocessing or region-specific models to maximize predictive accuracy in areas with consistent conditions.
- **Sequential Splitting Challenges:** The sequential region's poor performance ($R^2$ = 54.59%) reveals the limitations of sequential data splitting strategies in capturing basin-wide variability. This suggests that

random splitting or cross-validation methods should be preferred to ensure adequate exposure to the variability in the dataset.

- **General Applicability of the Model:** The comparable performance of the random region to the baseline indicates that the model has good generalization capabilities when trained on diverse and representative data. This reinforces its potential applicability to other regions, provided that the training data adequately captures the variability of the target area.
- **Recommendations for Model Deployment:** The results emphasize the importance of understanding the underlying dataset and tailoring the model application to specific regions. In practice, regional calibration or hybrid modeling approaches, such as kriging or radial basis function (RBF) interpolation, could be explored to address performance gaps in heterogeneous or weak zones. Furthermore, leveraging multi-modal input features should be a priority for future deployment. This approach could incorporate additional data sources, such as results from Finite Element Method (FEM) simulations, to enrich the model's understanding of subsurface characteristics and improve its predictive performance in diverse geotechnical contexts.
- **Geotechnical Applications:** These findings highlight the potential for ML to improve predictions in geotechnical engineering. Beyond predicting $q_c$, the model's predictions can inform various geotechnical assessments, such as:

    1. **Refining Design Assumptions:** Improved $q_c$ predictions can help verify or update existing design assumptions. For instance, if the strong (Pleistocene) sand layer is shallower than assumed, embankment stability or pile capacity may be higher than anticipated.
    2. **Identifying Local Weak Layers:** Detecting weak soil zones near the pile tip locally, can prompt a re-evaluation of pile bearing capacity and potentially also lead to design modifications or ground improvement measures.
    3. **Assessing Ground Investigation Coverage:** ML-based insights can reveal whether ground investigations are sufficient, guiding decisions on additional data collection in critical areas or layers.
    4. **Reverse Engineering:** Reverse engineering through the use of pile installation records as post-installation CPTs enables updating the soil profile itself, enhancing its accuracy for downstream tasks. Examples include refining load-bearing capacity calculations, and informing real-time adjustments to pile driving strategies. Additionally, this approach supports the development of digital twins, facilitating better decision-making in construction and long-term maintenance.

The red region is the region of the basin that excludes the blue region shown in Figure 5b. From the red region, we will zoom into a specific set of piles that were part of the test set and compare the ground-truth to the predictions, comparing the local SHAP values in order to explain what the model is actually learning from on a local level. In Figure 8a we zoom in on three piles and see their ground-truth. When comparing this image to the predicted values in Figure 8b , we can clearly see a change in the subsoil layers of VIP-0493 when comparing it to its ground-truth layers. We can conclude the potential detection of the compaction effect, the installation order and the effect the installation procedure has on the subsoil structure in comparison to the subsoil structure prior to installation.

| Feature | First Prediction | Middle Prediction | Last Prediction |
|---|---|---|---|
| **Pile ID: VIP-0495** | | | |
| Delta Blow Count | **-1.70** | -0.61 | +1.96 |
| Y | -1.46 | -3.59 | +10.37 |
| Energy Level | -0.64 | -0.24 | -1.28 |
| X | -0.53 | -0.16 | +0.49 |
| Depth | +0.43 | **-4.90** | **+10.52** |
| **Pile ID: VIP-0491** | | | |
| Delta Blow Count | **-1.70** | -0.85 | +1.56 |
| Y | -1.46 | -3.19 | **+14.34** |
| Energy Level | -0.64 | -0.50 | -1.40 |
| X | -0.53 | -0.92 | -2.11 |
| Depth | +0.43 | **-5.57** | +13.29 |
| **Pile ID: VIP-0493** | | | |
| Delta Blow Count | **-1.70** | -0.74 | +1.81 |
| Y | -1.46 | -3.29 | **+15.37** |
| Energy Level | -0.64 | -0.49 | -0.61 |
| X | -0.53 | -0.76 | +1.81 |
| Depth | +0.43 | **-5.37** | +12.66 |

**Table 6:** SHAP Contributions for First, Middle, and Last Predictions Across Specified Piles.

Using the SHAP local explainability values, we can identify which features predominantly influence or marginally impact the model's predictions. To achieve this, we examine three key instances: the first, middle, and last predictions for each specified pile. These correspond to the first, middle, and last predictions made along the depth of a single pile. Since the piles VIP-0495, VIP-0491, and VIP-0493 have been installed to the same depth and are in close proximity, they are excellent candidates for evaluating local feature importance using SHAP. The local SHAP values for these piles are summarized in Table 6. Notably, the piles are presented in the order of installation: VIP-0495 (the first pile installed), VIP-0491 (the second pile installed), and VIP-0493 (the last pile installed). From this analysis, we observe that for the first prediction of each pile, the contributions of individual features are identical. This consistency is expected for piles located in close proximity, as they penetrate similar soil conditions.

However, when examining the middle predictions, while the order of feature importance remains consistent, incremental changes in the absolute SHAP values are evident:

- For pile ID **VIP-0495**, the feature with the largest magnitude change in the SHAP value is depth (-4.90). This substantial influence indicates significant soil variability encountered at intermediate depths.
- Similarly, in **VIP-0491**, the depth (-5.57) contribution changes, reinforcing the observed trend of decreasing SHAP values. This may signify compaction, more specifically a change in resistance of the subsoil layer.
- For **VIP-0493**, depth (-5.37) continues to be the predominant feature, indicating consistent change of soil behavior across nearby piles, with marginal differences between piles.

In contrast, the last predictions reveal a shift in feature importance:

- For **VIP-0495** depth (+10.52) emerges as the most influential feature of this prediction, indicating that soil resistance drastically increases near the end of pile installation.
- For **VIP-0491**, Y (+14.34) overtakes depth (+13.29) in significance, signaling lateral spatial variability in soil stiffness or stress redistribution effects.
- Finally, for **VIP-0493** Y (+15.37) solidifies its dominance with a slight decrease in depth (+12.66), underscoring the progression of installation-induced soil interactions led by the spatial variation through the Y feature.

These trends highlight the ability of the model to detect sequential dependencies and the impact of installation procedures on subsoil behavior. Additionally, the SHAP values for machine-related features Delta Blow Count and Energy Level, reveal their consistent but nuanced contributions throughout the installation process. This suggests that these features not only capture variations in soil conditions but also provide actionable insights for real-time adjustments during pile installation, potentially improving efficiency and accuracy in response to subsurface conditions.

## 5. Conclusion and Future Work

This section synthesizes the study's key findings and outlines potential avenues for future research, focusing on improving predictive modeling of cone resistance from pile installation data.

### 5.1 Conclusion

This research demonstrated the capability of ML, particularly the XGB model, to accurately predict CPT measurements (specifically $q_c$) from DCIS pile installation data. By leveraging a well-prepared dataset and emphasizing representativeness, the model achieved $R^2$ values exceeding 80%, affirming its utility for geotechnical applications. Depth emerged as the most critical feature, reflecting the fundamental influence of vertical soil layering on CPT predictions. Spatial coordinates (specifically Y) also played a significant role, highlighting the importance of accurate geolocation in capturing lateral soil heterogeneity. While Energy Level and Delta Blow Count were less critical globally, SHAP values revealed their localized importance in capturing variations in soil resistance during pile installation. However, in regions with significant soil variability ("weak" areas), where there are many changing values and higher measurement errors, this is reflected in the SHAP values for these features being almost negligble (approximately 0) in the middle prediction. This indicates that their contributions are diminished when the model prioritizes features such as depth and spatial coordinates to account for the dominant variability in the subsurface conditions. These features provided insights into the interaction between machine-related features and subsurface conditions, aiding the model in refining predictions where soil characteristics deviated from homogeneity.

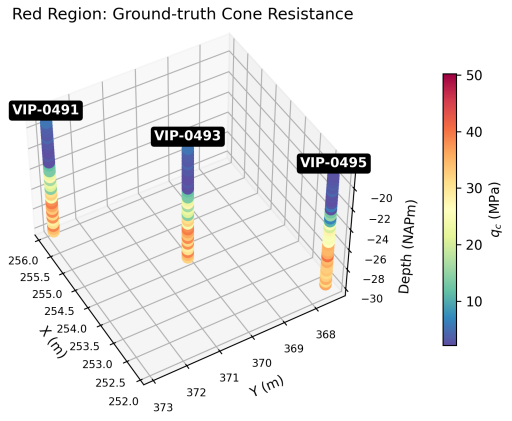
The model performed effectively in areas with consistent subsurface conditions (e.g., the red region) and demonstrated robustness even with reduced datasets, provided the data was representative. However, predictions in regions with lower cone resistance values (e.g., the blue region) or higher variability were less reliable, emphasizing the challenges of modeling in such conditions. These results suggest that strategic data collection, such as focusing on uncertain regions during preliminary pile installations, can significantly improve model accuracy and generalizability for subsequent predictions. This approach ensures that critical regions with higher uncertainty are addressed early, enabling more accurate soil profile updates and improved predictions for nearby piles yet to be installed.

The study's findings indicate feasible applications for this methodology in geotechnical engineering, including improving subsurface characterization, enhancing pile design processes, and potentially facilitating adaptive site-specific decision-making. While digital twins and real-time design updates remain aspirational goals, the results validate ML as a complementary tool in geotechnical workflows. An example use case of the developed model, showcasing its application in updating the subsoil model based on $q_c$ predictions is provided in Appendix D.

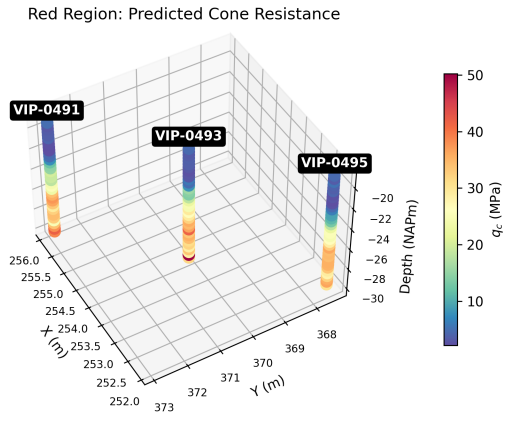
### 5.2 Future Work

While this study demonstrates the viability of ML models for predicting CPT measurements using pile installation data, several avenues offer potential for further enhancement and broader applicability:

- **Incorporation of Additional Features:** Future research could incorporate pore water pressure measurements (from CPTu) and other geotechnical parameters. Embedding advanced soil mechanics or thermodynamic principles and leveraging higher-quality sensor data from installation equipment may further improve predictive accuracy and model robustness.
- **Improved Ground-Truth CPT Interpolation:** Improving the interpolation methods used for ground-truth CPT data with Kriging, RBF, or other advanced geostatistical techniques to subsequently improve the model.




**(a)** Ground-truth Cone Resistance zoomed in at arbitrary piles within the red region of the basin.




**(b)** Predicted Cone Resistance zoomed in at arbitrary piles within the red region of the basin.

**Figure 8:** Comparison of Ground-Truth and Predicted Cone Resistance within the red region of the basin. The left subfigure shows the ground-truth values, while the right subfigure presents the corresponding predictions by the XGB model.

- **Physics-Informed Models:** Integrating domain knowledge and physical laws into ML architectures, such as Thermodynamics-based Artificial Neural Networks (TANNs) or Physics-Informed Neural Networks (PINNs) based on the wave equation, can increase transparency and ensure predictions adhere to fundamental geotechnical principles.
- **Transferability Across Sites:** Extending the current approach to multiple project locations with diverse soil types, pile geometries, installation techniques, and environmental conditions can help establish more versatile and generalizable predictive models, broadening their practical applicability.
- **Uncertainty Quantification:** Adopting Bayesian ML methods or quantile regression techniques would provide uncertainty estimates for each prediction. These confidence intervals can guide engineers in making more informed design decisions under variable subsurface conditions.
- **Integration with Digital Twins:** Incorporating ML-based CPT predictions into digital twins of geotechnical structures can enable real-time model updates and predictive maintenance, contributing to enhanced resilience and adaptability in infrastructure systems. Achieving robust predictive capabilities paves the way towards automated construction processes, eventually enabling robotic pile-driving machines to operate autonomously with human oversight limited to supervisory roles.

By pursuing these directions, future research can continue refining ML-driven geotechnical analyses, fostering the adoption of data-centric geotechnics, and harmonizing advanced AI methodologies with traditional engineering expertise.

## References

[1] W. Zhang, H. Li, Y. Li, H. Liu, Y. Chen, and X. Ding, "Application of deep learning algorithms in geotechnical engineering: A short critical review," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 5633–5673, Dec. 1, 2021, ISSN: 1573-7462. DOI: 10.1007/s10462-021-09967-1. [Online]. Available: https://doi.org/10.1007/s10462-021-09967-1 (visited on 02/07/2024).

[2] W. Shao, W. Yue, Y. Zhang, *et al.*, "The application of machine learning techniques in geotechnical engineering: A review and comparison," *Mathematics*, vol. 11, no. 18, p. 3976, Jan. 2023, Number: 18 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2227-7390. DOI: 10.3390/math11183976. [Online]. Available: https://www.mdpi.com/2227-7390/11/18/3976 (visited on 02/07/2024).

[3] K.-K. Phoon and W. Zhang, "Future of machine learning in geotechnics," *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, vol. 17, no. 1, pp. 7–22, Jan. 2, 2023, Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/17499518.2022.2087884, ISSN: 1749-9518. DOI: 10.1080/17499518.2022.2087884. [Online]. Available: https://doi.org/10.1080/17499518.2022.2087884 (visited on 09/30/2024).

[4] M. Fatehnia, V. Mahmoudabadi, and S. Amiri, "Utilizing machine learning for cone penetration test-based soil classification," *Transportation Research Record*, p. 03 611 981 241 245 679, May 16, 2024, Publisher: SAGE Publications Inc, ISSN: 0361-1981. DOI: 10.1177/03611981241245679. [Online]. Available: https://doi.org/10.1177/03611981241245679 (visited on 09/27/2024).

[5] A. Baghbani, T. Choudhury, S. Costa, and J. Reiner, "Application of artificial intelligence in geotechnical engineering: A state-of-the-art review," *Earth-Science*

*Reviews*, vol. 228, p. 103 991, May 1, 2022, ISSN: 0012-8252. DOI: 10.1016/j.earscirev.2022.103991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0012825222000757 (visited on 10/07/2024).

[6] M. A. Shahin, "State-of-the-art review of some artificial intelligence applications in pile foundations," *Geoscience Frontiers*, Special Issue: Progress of Machine Learning in Geosciences, vol. 7, no. 1, pp. 33–44, Jan. 1, 2016, ISSN: 1674-9871. DOI: 10.1016/j.gsf.2014.10.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1674987114001327 (visited on 09/27/2023).

[7] F. Masi and I. Stefanou, *Thermodynamics-based artificial neural networks (TANN) for multiscale modeling of materials with inelastic microstructure*, version: 2, Sep. 1, 2021. DOI: 10.48550/arXiv.2108.13137. arXiv: 2108.13137[cond-mat]. [Online]. Available: http://arxiv.org/abs/2108.13137 (visited on 12/09/2024).

[8] M. Sabah, M. Talebkeikhah, D. Wood, R. Khosravanian, M. Anemangely, and A. Younesi, "A machine learning approach to predict drilling rate using petrophysical and mud logging data," *Earth Science Informatics*, 2019. [Online]. Available: https://www.semanticscholar.org/paper/A-machine-learning-approach-to-predict-drilling-and-Sabah-Talebkeikhah/62fdd634edf407131cb139b785e46fb8b4efcca5 (visited on 02/26/2024).

[9] O. Ahmed, A. Adeniran, and A. Samsuri, "Computational intelligence based prediction of drilling rate of penetration: A comparative study," *Journal of Petroleum Science and Engineering*, vol. 172, pp. 1–12, Oct. 6, 2018. DOI: 10.1016/j.petrol.2018.09.027.

[10] D. J. Armaghani, E. T. Mohamad, M. S. Narayanasamy, N. Narita, and S. Yagiz, "Development of hybrid intelligent models for predicting TBM penetration rate in hard rock condition," *Tunnelling and Underground Space Technology*, vol. 63, pp. 29–43, Mar. 1, 2017, ISSN: 0886-7798. DOI: 10.1016/j.tust.2016.12.009. [Online]. Available: http://www.scopus.com/inward/record.url?scp=85006990022&partnerID=8YFLogxK (visited on 02/27/2024).

[11] A. Abouelmaty, A. Colaço, and P. Alves Costa, *Machine Learning Application for Predicting Ground-Borne Vibrations Induced by Impact Pile Driving*. Jun. 16, 2023.

[12] G. Singh and B. S. Walia, "Performance evaluation of nature-inspired algorithms for the design of bored pile foundation by artificial neural networks," *Neural Computing and Applications*, vol. 28, no. 1, pp. 289–298, Dec. 1, 2017, ISSN: 1433-3058. DOI: 10.1007/s00521-016-2345-1. [Online]. Available: https://doi.org/10.1007/s00521-016-2345-1 (visited on 02/09/2024).

[13] W. Gao, "The application of machine learning in geotechnical engineering," *Applied Sciences*, vol. 14, no. 11, p. 4712, Jan. 2024, Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417. DOI: 10.3390/app14114712. [Online]. Available: https://www.mdpi.com/2076-3417/14/11/4712 (visited on 10/01/2024).

[14] Keller, *Driven cast-in-situ piles*, Accessed: 2024-10-25, 2024. [Online]. Available: https://www.keller.com/expertise/techniques/driven-cast-in-situ-piles.

[15] P. K. Robertson, "Soil behaviour type from the CPT: An update," 2010.

[16] T. Lunne, P. Robertson, and J. Powell, "Cone penetration testing in geotechnical practice," *Soil Mechanics and Foundation Engineering*, vol. 46, Jan. 1, 1997. DOI: 10.1007/s11204-010-9072-x.

[17] P. Kurup and E. Griffin, "Prediction of soil composition from CPT data using general regression neural network," *Journal of Computing in Civil Engineering - J COMPUT CIVIL ENG*, vol. 20, Jul. 1, 2006. DOI: 10.1061/(ASCE)0887-3801(2006)20:4(281).

[18] P. Robertson, "Soil classification using the cone penetration test," *Canadian Geotechnical Journal - CAN GEOTECH J*, vol. 27, pp. 151–158, Feb. 1, 1990. DOI: 10.1139/t90-014.

[19] F. Campos Montero, "Deep learning for geotechnical engineering: The effectiveness of generative adversarial networks in subsoil schematization," 2023. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3Ac18cb6cf-3574-484d-aacc-dabd882341de (visited on 11/07/2023).

[20] M. Shahin, "Intelligent computing for modeling axial capacity of pile foundations," *Canadian Geotechnical Journal*, vol. 47, pp. 230–243, Feb. 10, 2010. DOI: 10.1139/T09-094.

[21] M. Bagińska and P. E. Srokosz, "The optimal ANN model for predicting bearing capacity of shallow foundations trained on scarce data," *KSCE Journal of Civil Engineering*, vol. 23, no. 1, pp. 130–137, Jan. 1, 2019, ISSN: 1976-3808. DOI: 10.1007/s12205-018-2636-4. [Online]. Available: https://doi.org/10.1007/s12205-018-2636-4 (visited on 10/07/2024).

[22] F. Pooya Nejad and M. B. Jaksa, "Load-settlement behavior modeling of single piles using artificial neural networks and CPT data," *Computers and Geotechnics*, vol. 89, pp. 9–21, Sep. 2017, ISSN: 0266352X. DOI: 10.1016/j.compgeo.2017.04.003. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0266352X17300915 (visited on 01/31/2024).

[23] A. S. Vesić, *Design of pile foundations* (Synthesis of highway practice 42), in collab. with National Research Council (U.S.) Washington: Transportation Research Board, National Research Council, 1977, 68 pp., ISBN: 978-0-309-02544-7.

[24] I. V. Ofrikhter and A. B. Ponomarev, "Estimation of load-set behavior of driven concrete piles using artificial neural network and cone penetration test," *Journal of Physics: Conference Series*, vol. 1928, no. 1, p. 012 055, Jun. 1, 2021, ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1928/1/012055. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1928/1/012055 (visited on 01/31/2024).

[25] Inpieq. "Inpieq Data Logging System (IDLS)." Accessed: 2024-10-25. (2024), [Online]. Available: https://inpieq.nl/.

[26] SBE Engineering. "SBEGEO Standard." Accessed: 2024-03-11. (2024), [Online]. Available: https://www.sbe-engineering.com/.

[27] Basisregistratie Ondergrond (BRO). "BROloket: Geotechnical Data." Accessed: 20232-11-08. (2024), [Online]. Available: https://www.broloket.nl/ondergrondgegevens.

[28] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Foundations and Trends® in Computer Graphics and Vision*, vol. 7, no. 2, pp. 81–227, Mar. 28, 2012, Publisher: Now Publishers, Inc., ISSN: 1572-2740, 1572-2759. DOI: 10.1561/0600000035. [Online]. Available: https://www.nowpublishers.com/article/Details/CGV-035 (visited on 11/12/2024).

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 1, 2001, ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324 (visited on 11/12/2024).

[30] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas, "Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines," *Ore Geology Reviews*, vol. 71, pp. 804–818, Dec. 1, 2015, ISSN: 0169-1368. DOI: 10.1016/j.oregeorev.2015.01.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169136815000037 (visited on 11/12/2024).

[31] M. H. Rahman, M. Y. Abu-Farsakh, and N. Jafari, "Generation and evaluation of synthetic cone penetration test (CPT) data using various spatial interpolation techniques," *Canadian Geotechnical Journal*, vol. 58, no. 2, pp. 224–237, Feb. 2021, Publisher: NRC Research Press, ISSN: 0008-3674. DOI: 10.1139/cgj-2019-0745. [Online]. Available: https://cdnsciencepub.com/doi/10.1139/cgj-2019-0745 (visited on 07/30/2024).

[32] J. Liu, J. Liu, Z. Li, X. Hou, and G. Dai, "Estimating CPT parameters at unsampled locations based on kriging interpolation method," *Applied Sciences*, vol. 11, no. 23, p. 11 264, Jan. 2021, Number: 23 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417. DOI: 10.3390/app112311264. [Online]. Available: https://www.mdpi.com/2076-3417/11/23/11264 (visited on 08/01/2024).

[33] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1, 1996, ISSN: 1573-0565. DOI: 10.1007/BF00058655. [Online]. Available: https://doi.org/10.1007/BF00058655 (visited on 11/12/2024).

[34] W. Wang, G. Chakraborty, and B. Chakraborty, "Predicting the risk of chronic kidney disease (CKD) using machine learning algorithm," *Applied Sciences*, vol. 11, p. 202, Dec. 28, 2020. DOI: 10.3390/app11010202.

[35] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, *A comparative analysis of XGBoost*, Nov. 5, 2019. DOI: 10.48550/arXiv.1911.01914. arXiv: 1911.01914. [Online]. Available: http://arxiv.org/abs/1911.01914 (visited on 11/12/2024).

[36] S. O. Arik and T. Pfister, *TabNet: Attentive interpretable tabular learning*, Dec. 9, 2020. arXiv: 1908.07442. [Online]. Available: http://arxiv.org/abs/1908.07442 (visited on 11/05/2024).

[37] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*, Second edition. Beijing Boston Farnham Sebastopol Tokyo: O'Reilly, 2019, 1 p., ISBN: 978-1-4920-3264-9.

[38] J. Tan, J. Yang, S. Wu, G. Chen, and J. Zhao, *A critical look at the current train/test split in machine learning*, Jun. 8, 2021. DOI: 10.48550/arXiv.2106.04525. arXiv: 2106.04525. [Online]. Available: http://arxiv.org/abs/2106.04525 (visited on 11/16/2024).

[39] G. James, D. Witten, T. Hastie, and R. Tibshirani, "Resampling methods," in *An Introduction to Statistical Learning: with Applications in R*, G. James, D. Witten, T. Hastie, and R. Tibshirani, Eds., New York, NY: Springer US, 2021, pp. 197–223, ISBN: 978-1-07-161418-1. DOI: 10.1007/978-1-0716-1418-1_5. [Online]. Available: https://doi.org/10.1007/978-1-0716-1418-1_5 (visited on 11/16/2024).

[40] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination r-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, e623, Jul. 5, 2021. DOI: 10.7717/peerj-cs.623. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8279135/ (visited on 10/20/2024).

[41] T. Chai and R. Draxler, "Root mean square error (RMSE) or mean absolute error (MAE)?– arguments against avoiding RMSE in the literature," *Geoscientific Model Development*, vol. 7, pp. 1247–1250, Jun. 30, 2014. DOI: 10.5194/gmd-7-1247-2014.

[42] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Research*, vol. 30, p. 79, Dec. 19, 2005. DOI: 10.3354/cr030079.

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, *Scikit-learn: Machine learning in python*, Jun. 5, 2018. DOI: 10.48550/arXiv.1201.0490. arXiv: 1201.0490. [Online]. Available: http://arxiv.org/abs/1201.0490 (visited on 11/16/2024).

[44] L. Buitinck, G. Louppe, M. Blondel, *et al.*, *API design for machine learning software: Experiences from the scikit-learn project*, Sep. 1, 2013. DOI: 10.48550/arXiv.1309.0238. arXiv: 1309.0238. [Online]. Available: http://arxiv.org/abs/1309.0238 (visited on 11/16/2024).

[45] S. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, Nov. 25, 2017. DOI: 10.48550/arXiv.1705.07874. arXiv: 1705.07874. [Online]. Available: http://arxiv.org/abs/1705.07874 (visited on 10/20/2024).

[46] M. T. Ribeiro, S. Singh, and C. Guestrin, *"why should i trust you?": Explaining the predictions of any classifier*, Aug. 9, 2016. DOI: 10.48550/arXiv.1602.04938. arXiv: 1602.04938[cs]. [Online]. Available: http://arxiv.org/abs/1602.04938 (visited on 01/10/2025).

[47] H. M. Hammouri, R. T. Sabo, R. Alsaadawi, and K. A. Kheirallah, "Handling skewed data: A comparison of two popular methods," *Applied Sciences*, vol. 10, no. 18, p. 6247, Jan. 2020, Number: 18 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2076-3417. DOI: 10.3390/app10186247. [Online]. Available: https://www.mdpi.com/2076-3417/10/18/6247 (visited on 11/19/2024).

# Appendix A. Data Preprocessing Explained

Figure 9 and Figure 10 provide an overview of the data preprocessing pipelines used in this study. Pipeline 1 processes the Cone Penetration Test (CPT) data, while Pipeline 2 handles the pile installation data. These pipelines ensure that both datasets are cleaned from missing values, aligned on depth, dowmsampled, and combined for ML tasks.

## A.1 Description of Data Pipeline 1 for CPT Data:

During the parsing process, the elevation $z$ (see # ZID in Excerpt 1) with respect to the "Normaal Amsterdams Peil" (NAP) was also taken into account to ensure that all measurements were correctly aligned across the different datasets. The following preprocessing steps were applied:

1. **Reading and Parsing GEF Files:** Viewing the data available in the GEF files and extracting the useful information within the metadata and the raw measurements to a structured CSV file format for further data manipulation.
2. **Handling Missing Values:** No missing values were found in the CPT dataset. The data was complete and ready for further processing.
3. **Aggregation of Data Points:** CPT data, originally recorded at fine depth intervals of every 2 cm, was aggregated into coarser intervals of 10 cm, 30 cm, and 50 cm to align with the resolution of the pile installation data. This aggregation was performed using linear interpolation to ensure that the recorded target variables were available at specific depths, facilitating direct comparisons with the corresponding data from the pile installation records.
4. **Training feature** A key feature was selected from the available CPT parameters, with the aid of domain knowledge, namely:
    - **Elevation ($z$):** Required for calculating and aligning the data w.r.t NAP for comparison with other CPTs.
5. **Target Variables:** From the CPT parameters, we also extract the target variables required for our prediction, namely:
    - **Cone Resistance ($q_c$):** The resistance acting on the cone in megapascals (MPa).
    - **Sleeve Friction ($f_s$):** The resistance acting on the shaft of the cone in MPa.
    - **Friction Ratio ($R_f$):** The ratio, expressed as a percentage, of $f_s$ to $q_c$, both measured at the same depth as shown in Equation 4.
6. **Look-up table:** Create a mapping of the **five** CPTs closest (from the sparse grid of 355 GEF files) to each pile that has been installed, see Figure 2. By using the five closest CPTs we allow for the addition of random noise in the interpolation step that follows. Providing insights into how the models can handle such uncertainties.
7. **Inverse Distance Weighted (IDW) interpolation:** Generating synthetic CPT data to map to every pile location on the grid, see Figure 2.

**Example of Metadata and Measurements from Raw GEF File**

The following is an excerpt of the metadata and the first ten rows of recorded data from one of the 355 Geotechnical Exchange Format (GEF) files used in this study:

```
# GEFID= 1, 1, 0
# FILEOWNER= SW11
# FILEDATE= 2017, 11, 22
# PROJECTID= CPT, 1702693
# PROJECTNAME= Geotechnisch Bodemonderzoek RWG
# TESTID= 1
# STARTTIME= 07, 48, 51
# STARTDATE= 2017, 09, 11
# COLUMN= 8
# COLUMNINFO= 1, m, Sondeerlengte, 1
# COLUMNINFO= 2, MPa, Conusweerstand, 2
# COLUMNINFO= 3, MPa, Plaatselijke wrijving, 3
# COLUMNINFO= 4, %, Wrijvingsgetal, 4
# COLUMNINFO= 5, Graden, Helling, 8
# COLUMNINFO= 6, Graden, Helling N-Z, 9
# COLUMNINFO= 7, Graden, Helling O-W, 10
# COLUMNINFO= 8, m, Gecorrigeerde diepte, 11
# COMPANYID= REDACTED
# COMMENT= ================================
# COMMENT= Geconverteerde sondering uit MRSV
# COMMENT= REDACTED
# COMMENT= Datum: 22-nov-2017 - 16:23
# COMMENT= ================================
# COLUMNVOID= 2, -999999
# COLUMNVOID= 3, -999999
# COLUMNVOID= 4, -999999
# COLUMNVOID= 5, -999999
# COLUMNVOID= 6, -999999
# COLUMNVOID= 7, -999999
# COLUMNVOID= 8, -999999
# COLUMNSEPARATOR= ;
# RECORDSEPARATOR= !
```

```
# LASTSCAN= 2013
# XYID= 31000, 58236.37, 440128.03, 0.02, 0.02
# ZID= 31000, 5.91, 0.05
# MEASUREMENTTEXT= 1, Havenbedrijf Rotterdam N.V., opdrachtgever
# MEASUREMENTTEXT= 4, S15-CFII.951, conus type en serienummer
# MEASUREMENTTEXT= 5, Sondeer track-truck 11; 23000 kg; geen ankers, sondeerequipment
# MEASUREMENTTEXT= 6, NEN-EN-ISO22476-1 / klasse 2 / TE1, gehanteerde norm en klasse en type sondering
# MEASUREMENTTEXT= 9, maaiveld, vast horizontaal vlak
# MEASUREMENTTEXT= 101, Bronhouder, 52766179, 31
# MEASUREMENTVAR= 1, 1500, mm2, nom. oppervlak conuspunt
# MEASUREMENTVAR= 5, 80, mm, afstand tussen conuspunt en hart kleefmantel
# MEASUREMENTVAR= 12, 0, -, elektrische sondering
# MEASUREMENTVAR= 17, 0, -, Stopcriterium: Einddiepte bereikt
# REPORTCODE= GEF-CPT-Report, 1, 1, 2, gefcr112.pdf
# REPORTTEXT= 201, REDACTED
# REPORTTEXT= 202, Geotechnisch Bodemonderzoek RWG te Rotterdam - Maasvlakte
# REPORTTEXT= 203, Sondering 1
# DATAFORMAT= ASCII
# EOH=
00.00;-999999;-999999;-999999;-999999;-999999;-999999;00.000;!
00.01;0.0288;0.0070;2.9789;0.5747;0.0741;0.5699;00.010;!
00.03;0.1568;0.0070;2.2615;0.7822;0.1765;0.7620;00.030;!
00.05;0.2817;0.0069;1.6817;0.5956;0.1730;0.5699;00.050;!
00.07;0.4801;0.0070;1.3520;0.6867;0.1800;0.6626;00.070;!
00.09;0.6178;0.0085;1.1758;0.5966;0.1765;0.5699;00.090;!
00.11;0.8963;0.0101;1.0988;0.8127;0.2824;0.7620;00.110;!
00.13;1.1780;0.0097;0.8508;0.9375;0.3848;0.8548;00.130;!
00.15;1.4501;0.0108;0.7645;1.0245;0.3813;0.9509;00.150;!
00.17;1.5845;0.0108;0.6346;0.8761;0.1765;0.8581;00.170;!
```

**Excerpt 1:** Metadata from GEF File

**Example of Parsed CSV file from Raw GEF File**

The excerpt below presents the first 30 lines in a parsed CSV file, including the columns Conusweerstand, Plaatselijke wrijving, Wrijvingsgetal, X, Y, and depth_NAPm. Specifically, the first column represents the Cone Resistance, the second column corresponds to the Sleeve Friction, and the third column denotes the Friction Ratio. The X and Y columns indicate spatial coordinates. The depth_NAPm column was calculated using the elevation of the probe at the start of the CPT procedure, ensuring that all CPTs are aligned to the same zero level for consistency.

```
Conusweerstand,Plaatselijke wrijving,Wrijvingsgetal,X,Y,depth_NAPm
0.0288,0.007,2.9789,297.82,376.08,5.9
0.8963,0.0101,1.0988,297.82,376.08,5.8
2.3656,0.013,0.5578,297.82,376.08,5.7
3.8797,0.0138,0.3549,297.82,376.08,5.6
4.8369,0.022,0.4534,297.82,376.08,5.5
5.7908,0.0287,0.4848,297.82,376.08,5.4
8.4893,0.0432,0.4928,297.82,376.08,5.3
13.028,0.06,0.446,297.82,376.08,5.2
19.017,0.0871,0.4682,297.82,376.08,5.1
21.473,0.1115,0.5192,297.82,376.08,5.0
23.083,0.1367,0.6155,297.82,376.08,4.9
20.247,0.1598,0.7826,297.82,376.08,4.8
20.986,0.1479,0.6956,297.82,376.08,4.7
25.689,0.1363,0.5364,297.82,376.08,4.6
28.826,0.1685,0.5855,297.82,376.08,4.5
30.317,0.202,0.6657,297.82,376.08,4.4
31.479,0.2089,0.6649,297.82,376.08,4.3
31.675,0.2078,0.6541,297.82,376.08,4.2
31.636,0.2114,0.6663,297.82,376.08,4.1
31.489,0.2114,0.6719,297.82,376.08,4.0
30.49,0.2224,0.7288,297.82,376.08,3.9
30.116,0.2516,0.8373,297.82,376.08,3.8
30.676,0.2518,0.825,297.82,376.08,3.7
30.804,0.208,0.6784,297.82,376.08,3.6
28.877,0.1482,0.5126,297.82,376.08,3.5
25.24,0.1181,0.4621,297.82,376.08,3.4
21.415,0.1162,0.5346,297.82,376.08,3.3
19.84,0.1498,0.7548,297.82,376.08,3.2
19.315,0.1451,0.7455,297.82,376.08,3.1
17.977,0.1758,0.9734,297.82,376.08,3.0
```

**Excerpt 2:** First 30 rows of parsed CSV file

This pipeline processed the CPT data and prepared it for combination with the pile installation dataset in pipeline 2.

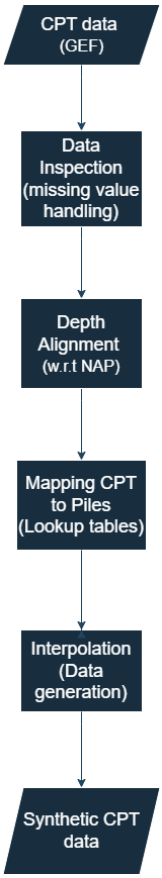



**Figure 9:** Data Pipeline 1.

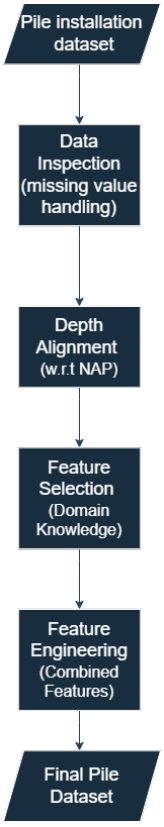



**Figure 10:** Data Pipeline 2

### A.2 Description of Data Pipeline 2 for Installation Data:

1. **Reading and Parsing the SBEGEO standard (JSON format):** From the available installation records, the information is extracted and similarly converted to CSV format for further data manipulation. However the JSON files came in two flavors, a headers file and a data file (see Excerpt 3 and Excerpt 4. These had to be parsed and combined to create the installation data CSV files that are used to later combine both data pipelines.
2. **Handling Missing Values:** As mentioned the pile records came in two parts, a headers file for all piles and a pile installation record per pile installed. Header files contain 10 key-value pairs with metadata on the pile to be installed. All the columns in these files were analyzed for missing values and if they can be used in the ML task.
3. **Relevant information addition:** From alternative sources (e.g. proprietary Geographic Information System (GIS) platform and contractor records) more information was gathered on each pile, such as information on the diameter of the piles, the baseplate used during installation and its dimensions, the installation order, the X and Y coordinates of each pile. Also an attempt was made at some feature engineering, however these features were
4. **Feature Selection:** From the available columns using domain knowledge, a selection of features was made of the most relevant features for the ML regression task. Additionally the distribution of the different available columns was checked, this resulted in the eventual elimination of certain columns as relevant features as there was little variation and would not inform the ML models for the prediction task. The selected features are summarized below:

    - **Depth NAP($m$):** This is the depth at which the measurements are taken w.r.t the NAP in meters. This is particularly useful when aligning the CPT data with the pile records.
    - **Energy level ($kNm$):** This is the energy setting of the diesel hammer recorded at every 10 cm (30 or 50 cm) increment during pile installation.
    - **Coordinate X ($m$):** The local relative x position of each pile.
    - **Coordinate Y ($m$):** The local relative y position of each pile.
    - **Delta Blow Count:** The number of blows to the pile, by the pile hammer, that is required for the pile to travel 10 cm (30 or 50 cm) through the subsoil.

**Example JSON Files of Installation Data**

As previously mentioned the installation data came in two parts a headers JSON and a data JSON, further separated in files for the DCIS piles and Steel Tubular Piles. In addition that had another subset where the files were even further separated

into installation methods of impact driving or vibratory driving. Thus, we only provide excerpts in the following of the headers and installation data of the DCIS piles installed with a diesel impact hammer.


```
[
    {
        "version": "sbegeo6.10",
        "element_type": "vibropalen",
        "element_id": "VIP-1434",
        "segment_id": "MA81",
        "type": "drive",
        "equipment": "s150",
        "operator": "Kees",
        "contractor": REDACTED,
        "rigname": "thw1100",
        "monitoringname": "DLS_205_V3_0_C8"
    },
    {
        "version": "sbegeo6.10",
        "element_type": "vibropalen",
        "element_id": "VIP-2567",
        "segment_id": "178",
        "type": "drive",
        "equipment": "D100_FVE50_PE400",
        "operator": "ROLAND",
        "contractor": REDACTED,
        "rigname": "6100XLR",
        "monitoringname": "TomTech III"
    }

    ...
]
```


**Excerpt 3:** First and last entries from the headers (metadata) JSON file.


```
[
    {
        "length_m": "0.00",
        "depth_NAPm": "1.50",
        "datetime": "NONE",
        "drive_energy_level_kNm": "NONE",
        "drive_blows_cumul": "NONE",
        "pile_id": "VIP-002"
    },
    {
    "length_m": "21.70",
    "depth_NAPm": "-19.55",
    "datetime": "2022-03-28 10:22:49",
    "drive_energy_level_kNm": "210",
    "drive_blows_cumul": "699",
    "pile_id": "VIP-002"
    },
    {
        "length_m": "0.00",
        "depth_NAPm": "-1.41",
        "datetime": "NONE",
        "drive_energy_level_kNm": "NONE",
        "drive_blows_cumul": "NONE",
        "pile_id": "VIP-046"
    },
    {
        "length_m": "31.80",
        "depth_NAPm": "-32.30",
        "datetime": "2022-05-06 10:07:00",
        "drive_energy_level_kNm": "210",
        "drive_blows_cumul": "2225",
        "pile_id": "VIP-046"
    }

    ...
]
```


**Excerpt 4:** First and last entries from the VIP-002 and VIP-046 in the data JSON file.

### Example of parsed CSV file from Installation data

```
length_m,depth_NAPm,datetime,drive_energy_level_kNm,drive_blows_cumul,pile_id
0.0,1.5,NaN,NaN,NaN,VIP-002
0.1,1.4,NaN,NaN,NaN,VIP-002
0.2,1.31,NaN,NaN,NaN,VIP-002
0.3,1.21,NaN,NaN,NaN,VIP-002
0.4,1.11,NaN,NaN,NaN,VIP-002
0.5,1.01,NaN,NaN,NaN,VIP-002
0.6,0.92,NaN,NaN,NaN,VIP-002
0.7,0.82,2022-03-28 10:02:25,NaN,1.0,VIP-002
0.8,0.72,2022-03-28 10:02:30,NaN,3.0,VIP-002
0.9,0.63,NaN,NaN,3.0,VIP-002
1.0,0.53,2022-03-28 10:02:56,NaN,4.0,VIP-002
1.1,0.43,2022-03-28 10:03:00,NaN,6.0,VIP-002
1.2,0.34,NaN,NaN,6.0,VIP-002
1.3,0.24,2022-03-28 10:03:31,NaN,9.0,VIP-002
1.4,0.14,2022-03-28 10:03:56,NaN,10.0,VIP-002
1.5,0.04,2022-03-28 10:03:58,NaN,11.0,VIP-002
1.6,-0.05,2022-03-28 10:04:01,NaN,12.0,VIP-002
1.7,-0.15,2022-03-28 10:04:29,NaN,14.0,VIP-002
1.8,-0.25,2022-03-28 10:04:33,NaN,15.0,VIP-002
1.9,-0.34,2022-03-28 10:04:59,NaN,16.0,VIP-002
2.0,-0.44,2022-03-28 10:05:04,NaN,19.0,VIP-002
2.1,-0.54,2022-03-28 10:05:29,NaN,20.0,VIP-002
2.2,-0.63,2022-03-28 10:05:30,NaN,21.0,VIP-002
2.3,-0.73,2022-03-28 10:05:36,NaN,24.0,VIP-002
2.4,-0.83,2022-03-28 10:06:02,140.0,25.0,VIP-002
2.5,-0.93,2022-03-28 10:06:03,140.0,26.0,VIP-002
2.6,-1.02,2022-03-28 10:06:05,140.0,28.0,VIP-002
2.7,-1.12,2022-03-28 10:06:08,140.0,30.0,VIP-002
2.8,-1.22,2022-03-28 10:06:10,140.0,32.0,VIP-002
2.9,-1.31,2022-03-28 10:06:13,140.0,34.0,VIP-002
3.0,-1.41,2022-03-28 10:06:15,140.0,36.0,VIP-002
```

**Excerpt 5:** First 31 rows of a parsed CSV file with missing values represented as NaN

### Example of Final Combined CSV files from GEF and Installation Data

```
length_m,depth_NAPm,datetime,drive_energy_level_kNm,drive_blows_cumul,delta_blow_count,cum_drive_energy,delta_drive_energy_10cm,pile_id,
      seconds,total_installation_time,equipment,segment_id,voorgeboord,type_deksel,installation_order,aux_tube_inner_diameter,
      aux_tube_outer_diameter,base_plate_diameter,base_plate_thickness,X,Y,q_c_MPa,f_s_MPa,R_f
1.6,-0.05,2022-03-28 10:04:01,140.0,12.0,1.0,1680.0,140.0,VIP-002,103,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,14.713,0.0782,0.5255
1.7,-0.15,2022-03-28 10:04:29,140.0,14.0,2.0,1960.0,280.0,VIP-002,131,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,14.713,0.0782,0.5255
1.8,-0.25,2022-03-28 10:04:33,140.0,15.0,1.0,2100.0,140.0,VIP-002,135,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,31.6629,0.2135,0.678
1.9,-0.34,2022-03-28 10:04:59,140.0,16.0,1.0,2240.0,140.0,VIP-002,161,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,22.6123,0.1129,0.503
2.0,-0.44,2022-03-28 10:05:04,140.0,19.0,3.0,2660.0,420.0,VIP-002,166,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,24.0783,0.139,0.584
2.1,-0.54,2022-03-28 10:05:29,140.0,20.0,1.0,2800.0,140.0,VIP-002,191,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,22.7885,0.1566,0.688
2.2,-0.63,2022-03-28 10:05:30,140.0,21.0,1.0,2940.0,140.0,VIP-002,192,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,20.1963,0.1569,0.768
2.3,-0.73,2022-03-28 10:05:36,140.0,24.0,3.0,3360.0,420.0,VIP-002,198,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,14.927,0.0917,0.615
2.4,-0.83,2022-03-28 10:06:02,140.0,25.0,1.0,3500.0,140.0,VIP-002,224,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,22.7916,0.1617,0.709
2.5,-0.93,2022-03-28 10:06:03,140.0,26.0,1.0,3640.0,140.0,VIP-002,225,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,22.3732,0.1595,0.721
2.6,-1.02,2022-03-28 10:06:05,140.0,28.0,2.0,3920.0,280.0,VIP-002,227,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,15.4845,0.1347,0.8635
2.7,-1.12,2022-03-28 10:06:08,140.0,30.0,2.0,4200.0,280.0,VIP-002,230,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,18.168,0.1045,0.5807
2.8,-1.22,2022-03-28 10:06:10,140.0,32.0,2.0,4480.0,280.0,VIP-002,232,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,18.471,0.1121,0.6017
2.9,-1.31,2022-03-28 10:06:13,140.0,34.0,2.0,4760.0,280.0,VIP-002,235,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,19.552,0.1168,0.6017
3.0,-1.41,2022-03-28 10:06:15,140.0,36.0,2.0,5040.0,280.0,VIP-002,237,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,20.933,0.1095,0.5298
3.1,-1.51,2022-03-28 10:06:18,140.0,38.0,2.0,5320.0,280.0,VIP-002,240,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,22.105,0.116,0.5261
3.2,-1.6,2022-03-28 10:06:20,140.0,40.0,2.0,5600.0,280.0,VIP-002,242,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,18.719,0.1199,0.659
3.3,-1.7,2022-03-28 10:06:23,140.0,42.0,2.0,5880.0,280.0,VIP-002,245,3315,Diesel,51,Nee,M
      ,203,0.549,0.609,0.68,0.05,707.894,1033.497,18.982,0.1161,0.617
3.4,-1.8,2022-03-28 10:06:27,140.0,45.0,3.0,6300.0,420.0,VIP-002,249,3315,Diesel,51,Nee,M,203,0.549,0.609,0.68,0.05,707
...
```

**Excerpt 6:** Rows from VIP-002 CSV File, these contain no NAN values which have been preprocessed out of the CSV File. Furthermore all avaialble features can be seen, but only the final subset of features were used as stated in the main text of this paper.

## Appendix B. Exploratory Data Analysis

To provide context on the input features and target variables used in this study, the distributions of each feature are shown in the histograms below (Figures 11 to 18). The exploratory data analysis (EDA) revealed several key characteristics of the input features and target variable distributions, guiding our preprocessing decisions.

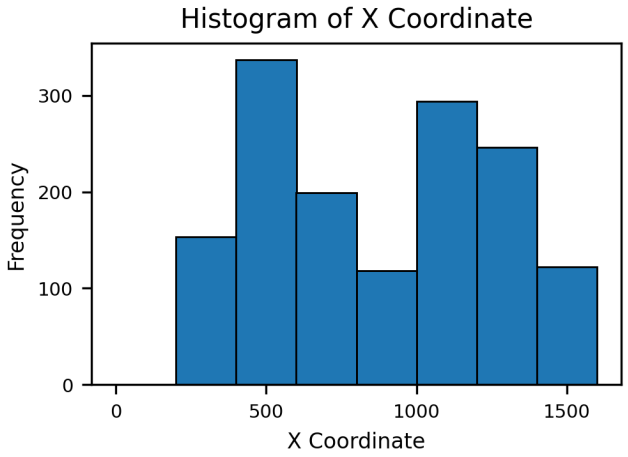



**Figure 11:** Histogram of X Coordinate across combined dataset

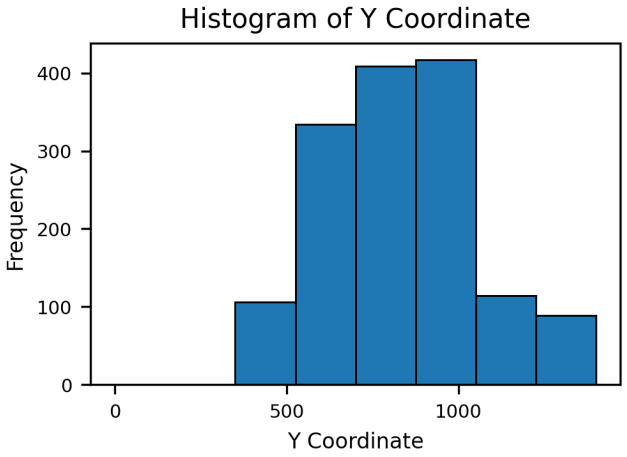



**Figure 12:** Histogram of Y Coordinate across combined dataset

The X and Y coordinate features exhibit roughly symmetric distributions, requiring minimal preprocessing apart from Min-Max scaling [43]. This method is chosen to address the large differences in absolute values between the features and target variables, ensuring proportional contributions during model training and improving numerical stability.

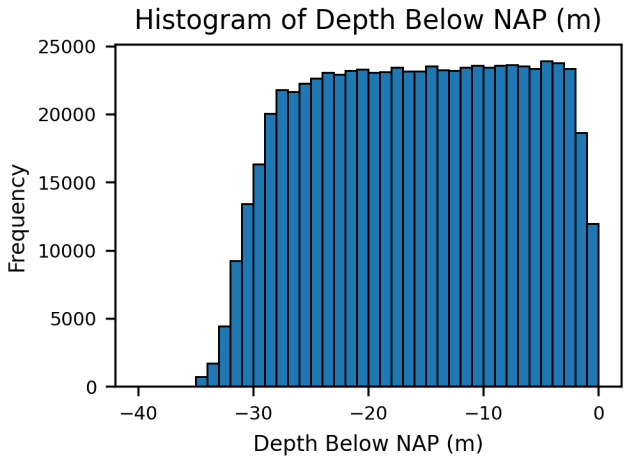



**Figure 13:** Histogram of Depth Below NAP (m)

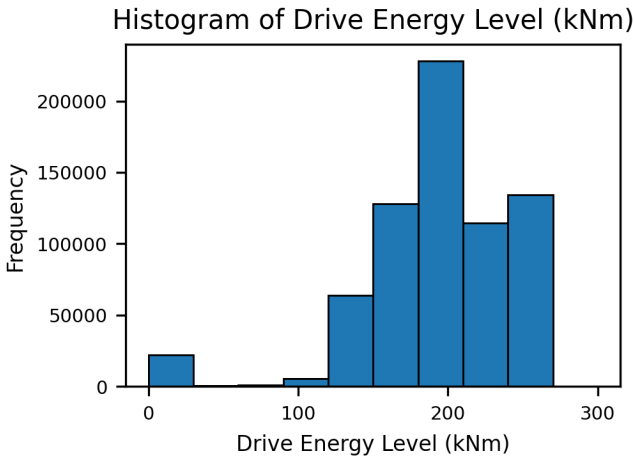



**Figure 14:** Histogram of Drive Energy Level (kNm)

Depth below NAP shows a near-uniform distribution, indicating consistent variability across its range and the drive energy level displays a mild skew, which could benefit from standardization or transformation for improved model sensitivity.

The Delta Blow Count and Cone Resistance are highly skewed, with long tails suggesting the need for log transformations to stabilize variance and mitigate the impact of extreme values [47].

Similarly, the Sleeve Friction and Friction Ratio exhibit highly skewed distributions with long tails, suggesting the use of log transformation to improve stability and handle extreme values. In Figures 19 and 20, log transformation and Min-Max scaling are applied to the features and target variables, and the resulting correlation matrices are analyzed to assess the linear dependence of target variables on input features and the effects of these transformations.

The correlation matrices reveal the effects of preprocessing on feature-target relationships. Before transformation, Delta Blow Count shows positive correlations with Cone Resistance (0.63) and Sleeve Friction (0.48), and a negative correlation

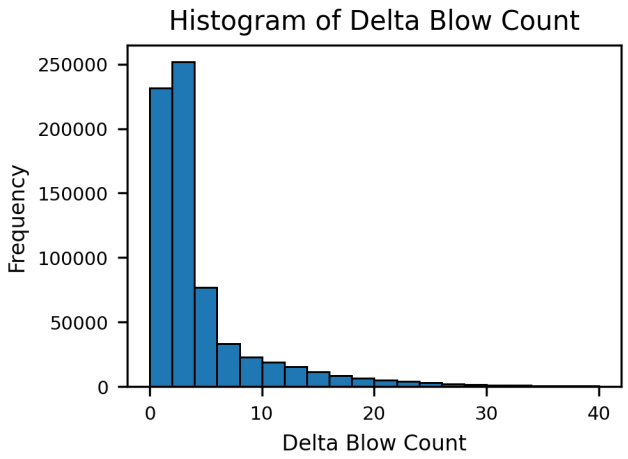



**Figure 15:** Histogram of Delta Blow Count

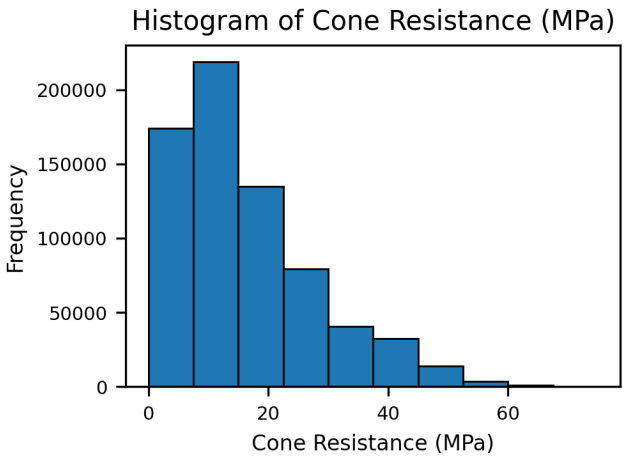



**Figure 16:** Histogram of Cone Resistance $q_c$ (MPa)

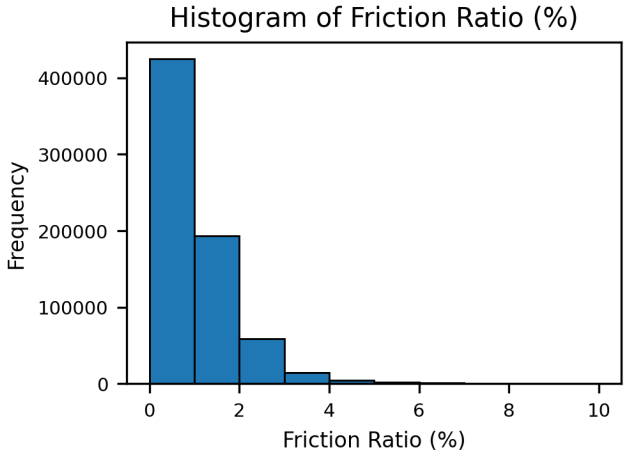



**Figure 17:** Histogram of Friction Ratio $R_f$ (%)

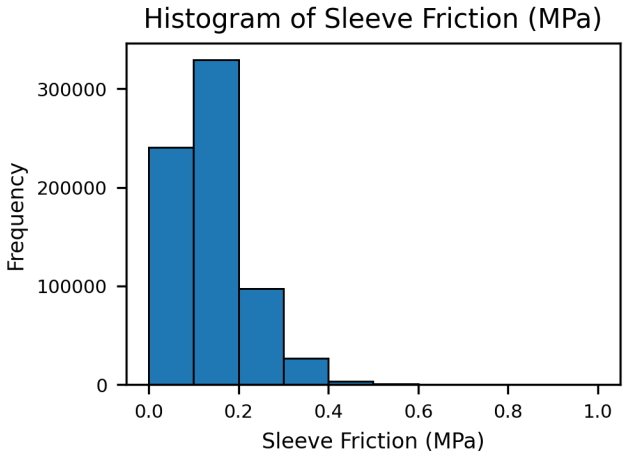



**Figure 18:** Histogram of Sleeve Friction $f_s$ (MPa)

with Friction Ratio (-0.28). After log transformation and Min-Max scaling, correlations with Cone Resistance (0.59) slightly decrease, while those with Sleeve Friction (0.49) and Friction Ratio improve (-0.36). Notably, the correlation between Depth and Cone Resistance weakens significantly (-0.31 to -0.16), which is unexpected, as Depth is considered the most relevant feature according to geotechnical experts. These findings suggest that non-linear models are the most suitable approach for this task, as they can capture complex relationships without relying heavily on feature transformations. This brings us to the next section, where we compare the previously specified tree-based models, which inherently handle non-linearities and reduce the need for extensive preprocessing.

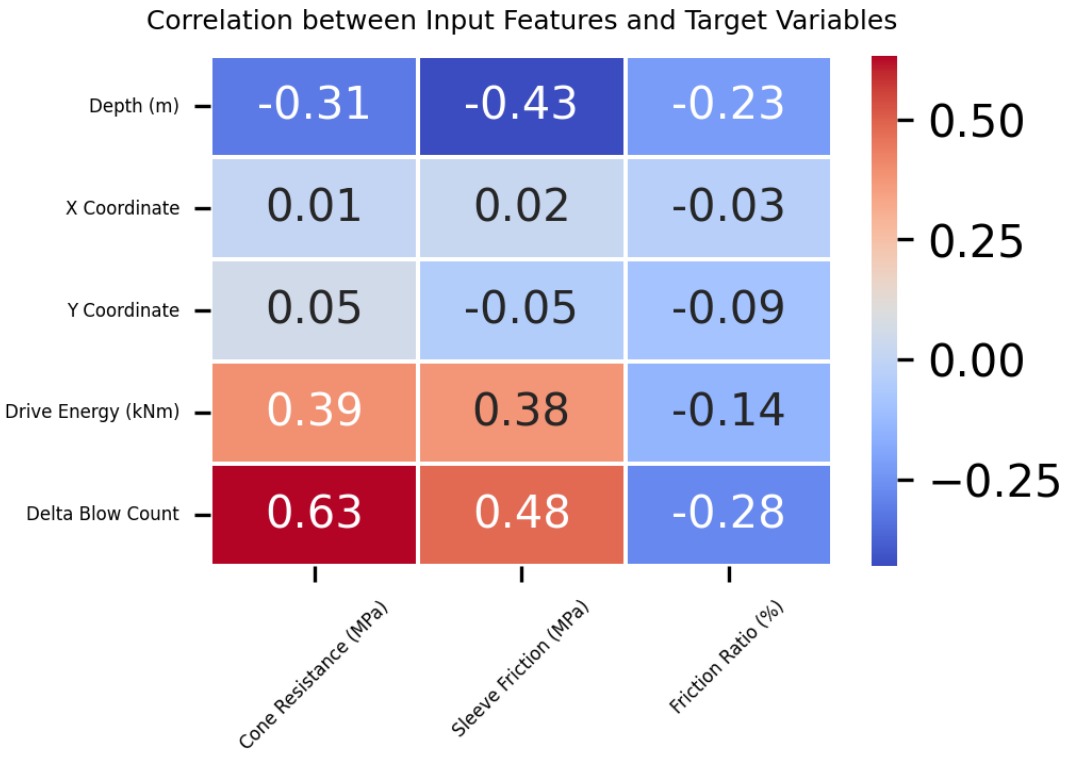



**Figure 19:** Correlation matrix showing linear relationships between input features and target variables.

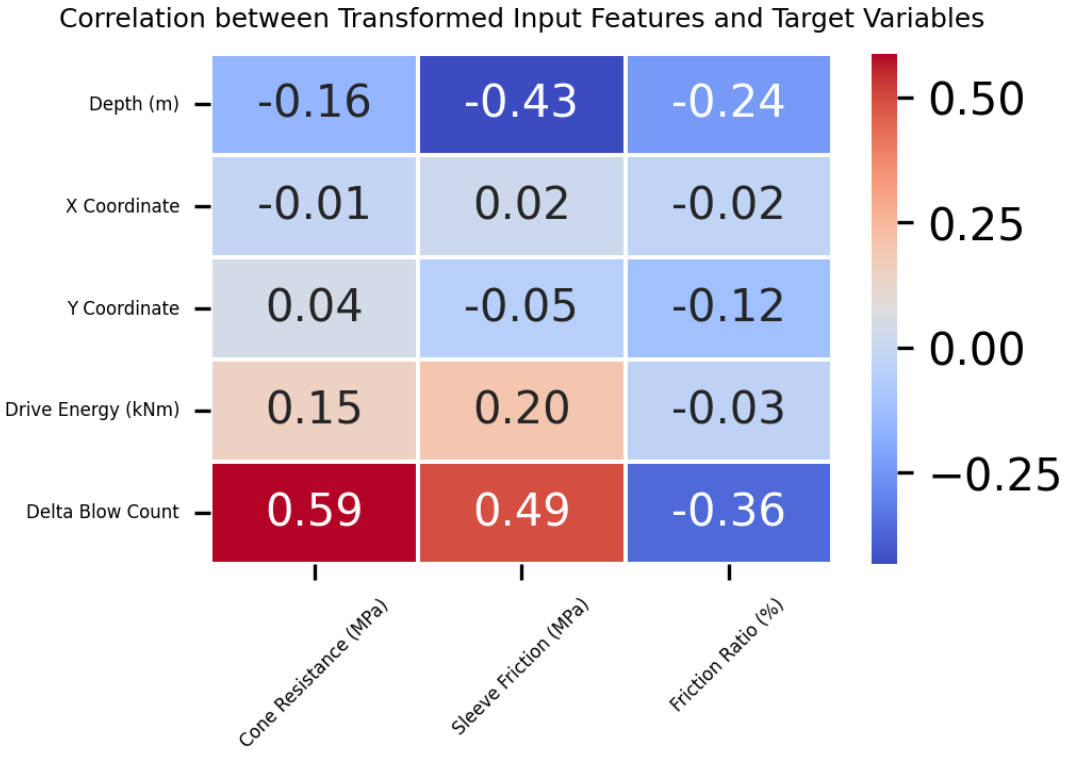



**Figure 20:** Correlation matrix showing linear relationships between transformed input features and target variables.

## Appendix C. Additional Results

This section presents the comprehensive performance results of the machine learning models tested on additional regression trends, specifically $q_c$, $f_s$, and $R_f$. Metrics such as $R^2$, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and training time were analyzed at resolutions of 10 cm, 30 cm, and 50 cm. The results highlight the performance differences across Random Forest, XGBoost, and TabNet models under default and optimized hyperparameters. For RF, the trade-off between the number of training piles and prediction accuracy is illustrated in Figure 24, showing that an increase in training data improves accuracy but at the cost of higher computational time. Similarly, TabNet's performance, shown in Figure 25, reveals that its data-intensive nature results in slower improvements with limited datasets. Optimized hyperparameters, determined through GridSearch for tree-based models and recommended configurations for TabNet, further demonstrate the importance of tuning in achieving better performance. These findings reinforce the earlier conclusion that XGBoost, with its balance of accuracy, computational efficiency, and suitability for smaller datasets, is the preferred model for this study, particularly at a resolution of 30 cm.

## C.1 Additional Performance Results of ML Models

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **Random Forest (10 cm)** | | | | |
| Default Parameters | 75.40% | 3.69 | 5.75 | 119 |
| Best Parameters | **73.67%** | **4.09** | **5.95** | **19** |
| **Random Forest (30 cm)** | | | | |
| Default Parameters | 87.12% | 2.49 | 3.82 | 40.66 |
| Best Parameters | **84.76%** | **2.85** | **4.15** | **6.41** |
| **Random Forest (50 cm)** | | | | |
| Default Parameters | 86.10% | 2.58 | 3.97 | 23.74 |
| Best Parameters | **83.92%** | **2.92** | **4.27** | **3.67** |

Random Forest Performance metrics for $q_c$.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **Random Forest (10 cm)** | | | | |
| Default Parameters | 69.77% | 0.03 | 0.04 | 119 |
| Best Parameters | **66.93%** | **0.03** | **0.04** | **19** |
| **Random Forest (30 cm)** | | | | |
| Default Parameters | 83.43% | 0.02 | 0.03 | 40.66 |
| Best Parameters | **81.02%** | **0.02** | **0.03** | **6.41** |
| **Random Forest (50 cm)** | | | | |
| Default Parameters | 82.01% | 0.02 | 0.03 | 23.74 |
| Best Parameters | **79.79%** | **0.02** | **0.03** | **3.67** |

Random Forest Performance metrics for $f_s$.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **Random Forest (10 cm)** | | | | |
| Default Parameters | 62.74% | 0.25 | 0.46 | 119 |
| Best Parameters | **58.22%** | **0.28** | **0.49** | **19** |
| **Random Forest (30 cm)** | | | | |
| Default Parameters | 77.55% | 0.17 | 0.31 | 40.66 |
| Best Parameters | **73.80%** | **0.19** | **0.33** | **6.41** |
| **Random Forest (50 cm)** | | | | |
| Default Parameters | 76.55% | 0.18 | 0.31 | 23.74 |
| Best Parameters | **73.17%** | **0.20** | **0.34** | **3.67** |

Random Forest Performance metrics for $R_f$.

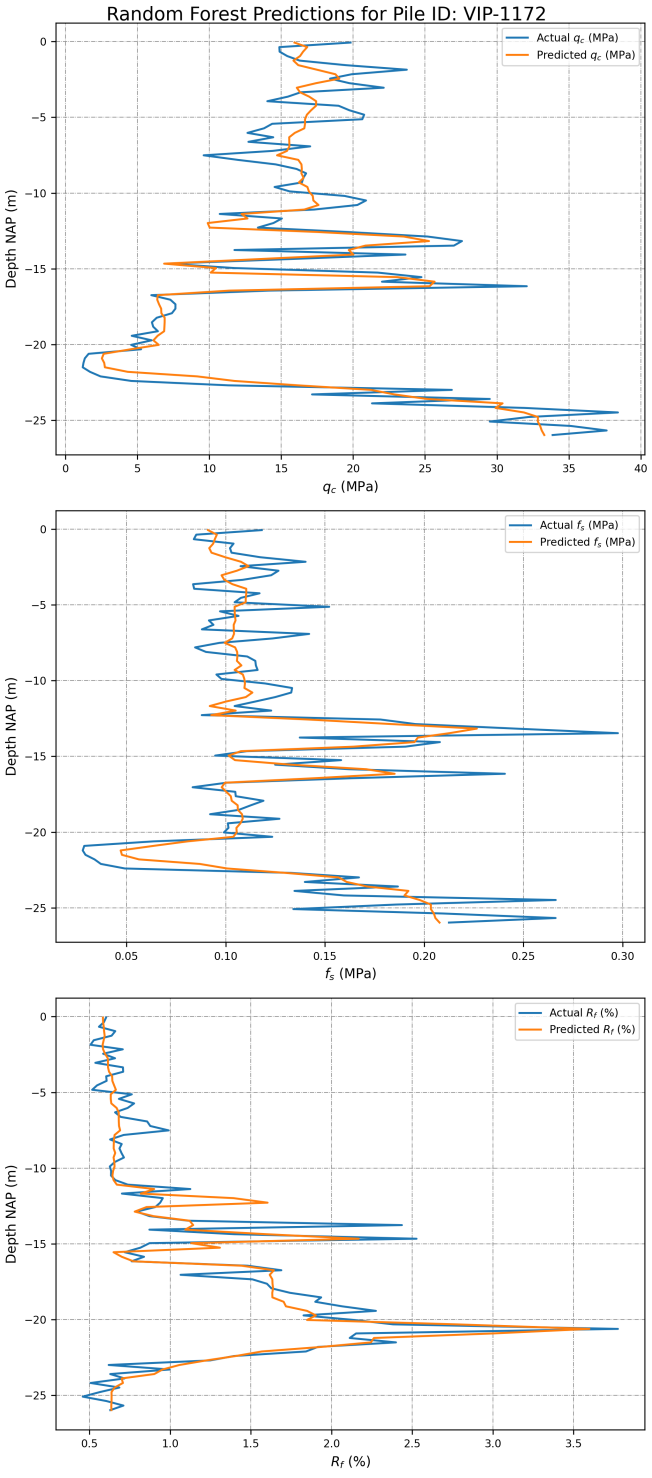



**Figure 21:** Random Forest multi-output predictions for Pile ID: VIP-1172 at a resolution of 30 cm.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **XGBoost (10 cm)** | | | | |
| Default Parameters | 71.33% | 4.33 | 6.21 | 1.41 |
| Best Parameters | **74.29%** | **3.93** | **5.88** | **5.21** |
| **XGBoost (30 cm)** | | | | |
| Default Parameters | 83.34% | 3.04 | 4.34 | 0.72 |
| Best Parameters | **86.57%** | **2.60** | **3.90** | **2.33** |
| **XGBoost (50 cm)** | | | | |
| Default Parameters | 83.07% | 3.05 | 4.38 | 0.5 |
| Best Parameters | **85.74%** | **2.67** | **4.02** | **1.56** |

XGBoost Performance metrics for $q_c$.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **XGBoost (10 cm)** | | | | |
| Default Parameters | 63.24% | 0.03 | 0.05 | 1.41 |
| Best Parameters | **54.62%** | **0.04** | **0.05** | **5.21** |
| **XGBoost (30 cm)** | | | | |
| Default Parameters | 77.89% | 0.02 | 0.03 | 0.72 |
| Best Parameters | **62.07%** | **0.03** | **0.04** | **2.33** |
| **XGBoost (50 cm)** | | | | |
| Default Parameters | 77.63% | 0.02 | 0.03 | 0.5 |
| Best Parameters | **58.71%** | **0.03** | **0.04** | **1.56** |

XGBoost Performance metrics for $f_s$.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **XGBoost (10 cm)** | | | | |
| Default Parameters | 53.74% | 0.30 | 0.52 | 1.41 |
| Best Parameters | **57.81%** | **0.28** | **0.49** | **5.21** |
| **XGBoost (30 cm)** | | | | |
| Default Parameters | 71.45% | 0.21 | 0.35 | 0.72 |
| Best Parameters | **73.42%** | **0.20** | **0.33** | **2.33** |
| **XGBoost (50 cm)** | | | | |
| Default Parameters | 72.00% | 0.21 | 0.34 | 0.5 |
| Best Parameters | **72.79%** | **0.21** | **0.34** | **1.56** |

XGBoost Performance metrics for $R_f$.

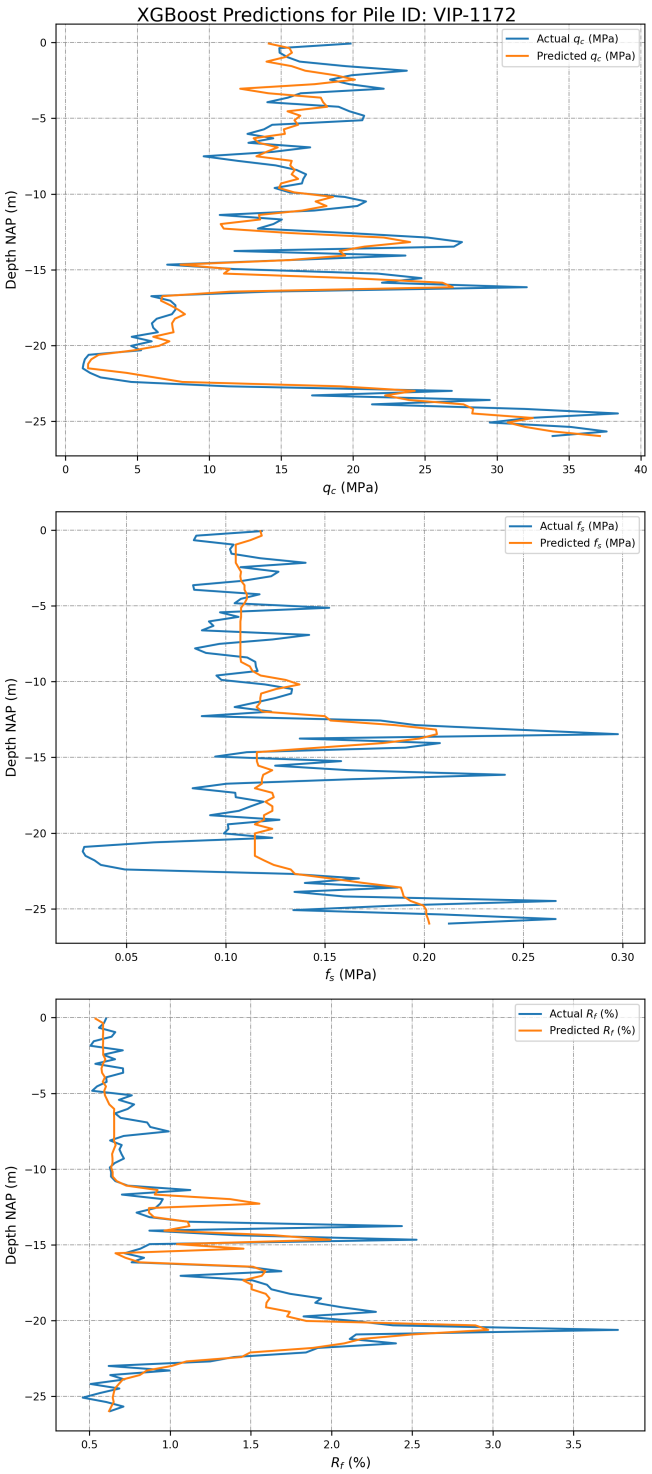



**Figure 22:** XGBoost multi-output predictions for Pile ID: VIP-1172 at a resolution of 30 cm.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **TabNet (10 cm)** | | | | |
| Default Parameters | 65.75% | 4.78 | 6.78 | 2056 |
| Best Parameters | **69.26%** | **4.50** | **6.42** | **1751** |
| **TabNet (30 cm)** | | | | |
| Default Parameters | 74.14% | 3.83 | 5.42 | 576 |
| Best Parameters | **80.37%** | **3.29** | **4.72** | **586** |
| **TabNet (50 cm)** | | | | |
| Default Parameters | 72.38% | 3.95 | 5.60 | 269 |
| Best Parameters | **77.04%** | **3.61** | **5.11** | **302** |

TabNet Performance metrics for $q_c$.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **TabNet (10 cm)** | | | | |
| Default Parameters | 47.12% | 0.04 | 0.05 | 2056 |
| Best Parameters | **51.96%** | **0.04** | **0.05** | **1751** |
| **TabNet (30 cm)** | | | | |
| Default Parameters | 45.01% | 0.03 | 0.05 | 576 |
| Best Parameters | **60.70%** | **0.03** | **0.04** | **586** |
| **TabNet (50 cm)** | | | | |
| Default Parameters | 53.76% | 0.03 | 0.05 | 269 |
| Best Parameters | **58.33%** | **0.03** | **0.04** | **302** |

TabNet Performance metrics for $f_s$.

| Configuration | $R^2$ | MAE (MPa) | RMSE (MPa) | Training Time (s) |
|---|---|---|---|---|
| **TabNet (10 cm)** | | | | |
| Default Parameters | 37.64% | 0.35 | 0.60 | 2056 |
| Best Parameters | **41.92%** | **0.35** | **0.58** | **1751** |
| **TabNet (30 cm)** | | | | |
| Default Parameters | 47.82% | 0.29 | 0.48 | 576 |
| Best Parameters | **54.81%** | **0.28** | **0.44** | **586** |
| **TabNet (50 cm)** | | | | |
| Default Parameters | 45.71% | 0.31 | 0.48 | 269 |
| Best Parameters | **52.55%** | **0.28** | **0.45** | **302** |

TabNet Performance metrics for $R_f$.

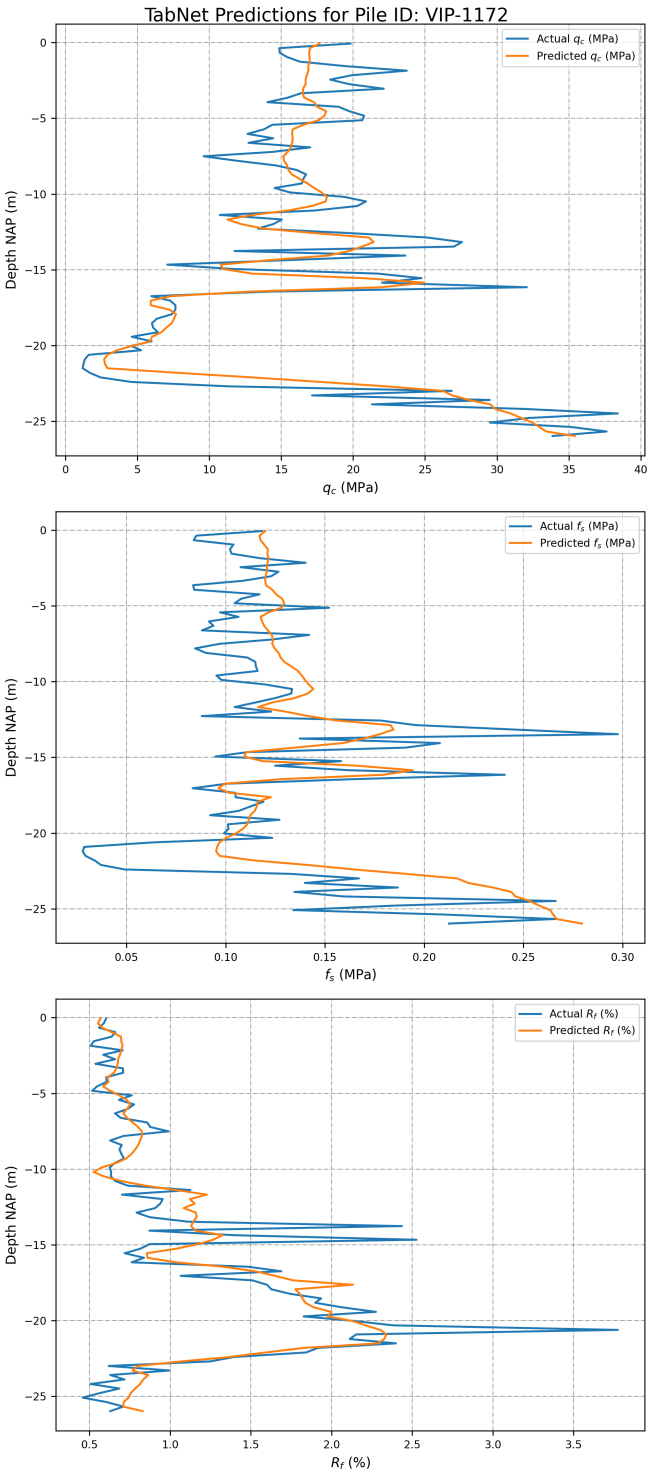



**Figure 23:** TabNet multi-output predictions for Pile ID: VIP-1172 at a resolution of 30 cm.

## C.2 Hyperparameters used for the GridSearch of Tree-based Models

```
# All RandomForestRegressor parameters considered
rf_params = {
    'random_state': 42, # For reproducibility
    'n_estimators': [50, 100, 200], # Number of trees in the forest
    'max_depth': [None, 5, 10, 15], # Maximum depth of each tree
    'min_samples_split': [10, 20, 40], # Minimum samples required to split an internal node
    'min_samples_leaf': [1, 4, 10], # The minimum number of samples required to be at a leaf node.
    'bootstrap': [True, False], # Whether bootstrap samples are used when building trees. True = use sampling with replacement.
    'oob_score': [True, False], # Whether to use out-of-bag samples to estimate the generalization score. Only works if bootstrap=True.
    'n_jobs': 12
}
```

**Excerpt 7:** Random Forest Regressor parameters used for GridSearch

```
# All XGboost parameters considered
xgb_params = {
    'random_state': 42, # For reproducibility
    'n_estimators': [50, 100, 200], # Number of trees in the forest
    'max_depth': [None, 5, 10, 15], # Maximum depth of each tree
    'learning_rate': [0.1, 0.2, 0.3], # Step size shrinkage to prevent overfitting and improve model generalization
    'subsample': [0.5, 0.6, 0.8, 1.0], # Subsample ratio of the training instances
    'colsample_bytree': [0.5, 0.8], # Subsample ratio of columns when constructing each tree
    'gamma': [0.1, 0.5, 1], # Min loss reduction required to make a further partition on a leaf node of the tree
    'reg_alpha': [0, 0.5, 1], # L1 regularization term on weights
    'reg_lambda': [1, 2, 5], # L2 regularization term on weights
    'n_jobs': 12
}
```

**Excerpt 8:** XGBoost Regressor parameters used for GridSearch

## C.3 Best Hyperparameters Found For Tree-based Models and TabNet

```
# Best RandomForestRegressor parameters
rf_params = {
    'random_state': 42,
    'n_estimators': 200,
    'max_depth': 15,
    'min_samples_split': 40,
    'min_samples_leaf': 4,
    'bootstrap': True,
    'oob_score': True,
    'n_jobs': 12
}
```

**Excerpt 9:** Best Random Forest Regressor parameters found using GridSearch

```
# Best XGBoost parameters
xgb_params = {
    'random_state': 42,
    'n_estimators': 200,
    'max_depth': 15,
    'learning_rate': 0.1,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'gamma': 1,
    'reg_alpha': 1,
    'reg_lambda': 5,
    'min_child_weight': [1, 5, 10],
    'n_jobs': 12
}
```

**Excerpt 10:** Best XGBoost parameters found using GridSearch

```
# TabNet Initialization Parameters
model = TabNetRegressor(
    optimizer_params=dict(lr=2e-2), # Learning rate
    scheduler_fn=torch.optim.lr_scheduler.ReduceLROnPlateau, # Learning rate scheduler
    scheduler_params={
        "mode": "min", # Minimize validation loss
        "patience": 5, # Number of epochs to wait before reducing learning rate
        "min_lr": 1e-4, # Minimum learning rate
        "factor": 0.5 # Factor to reduce learning rate
    },
    verbose=1 # Verbosity level
)

# model.fit Hyperparameters
fit_params = {
    'X_train': X_train, # Training features
    'y_train': y_train, # Training target
    'eval_set': [(X_train, y_train), (X_validation, y_validation)], # Evaluation sets
```

```
    'eval_name': ['train', 'valid'], # Evaluation set names
    'eval_metric': ['rmse', 'mae'], # Evaluation metrics
    'batch_size': 1024, # Batch size
    'virtual_batch_size': 512, # Virtual batch size
    'callbacks': [capture_metrics], # Custom callback for capturing metrics
    'compute_importance': True # Compute feature importance
}
```

**Excerpt 11:** Best TabNet parameters found using TabNet paper recommendations

## C.4 Minimum number of piles required for Training (RF and TabNet)

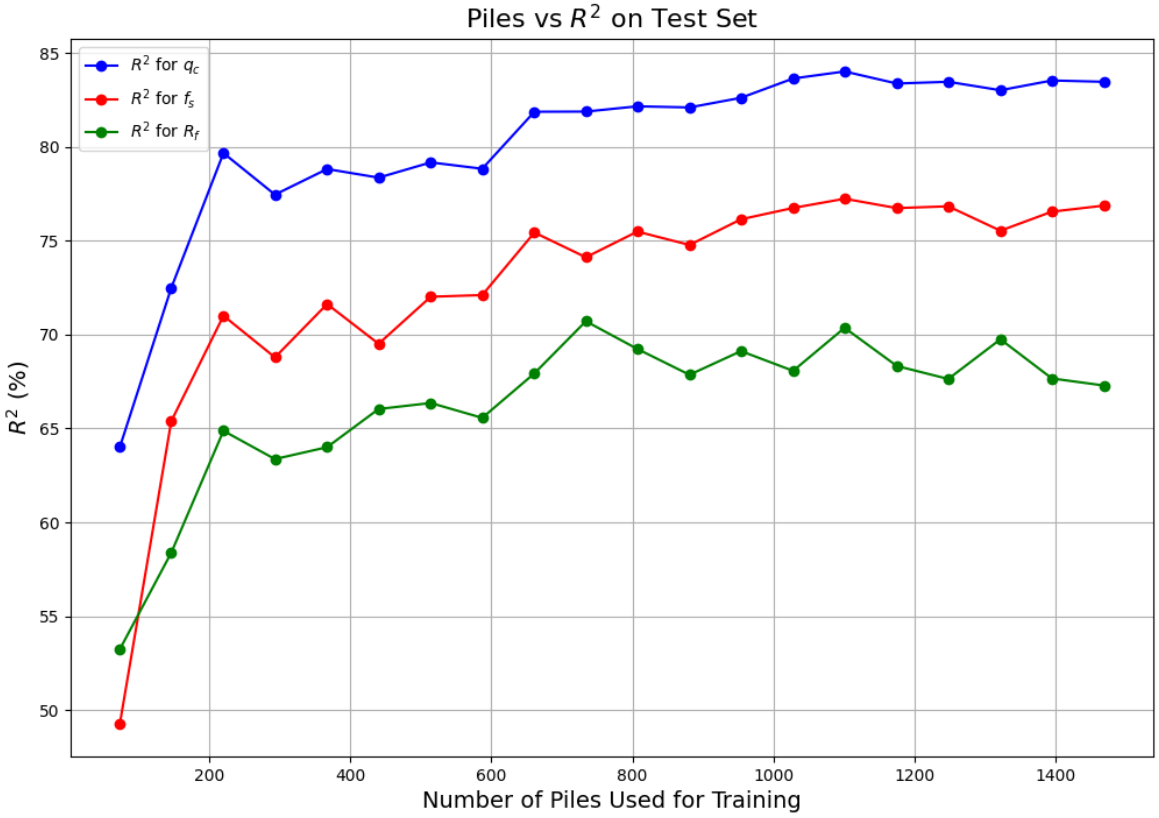



**Figure 24:** Performance of Random Forest trained on increasing numbers of piles, demonstrating the trade-off between the amount of data and prediction accuracy.

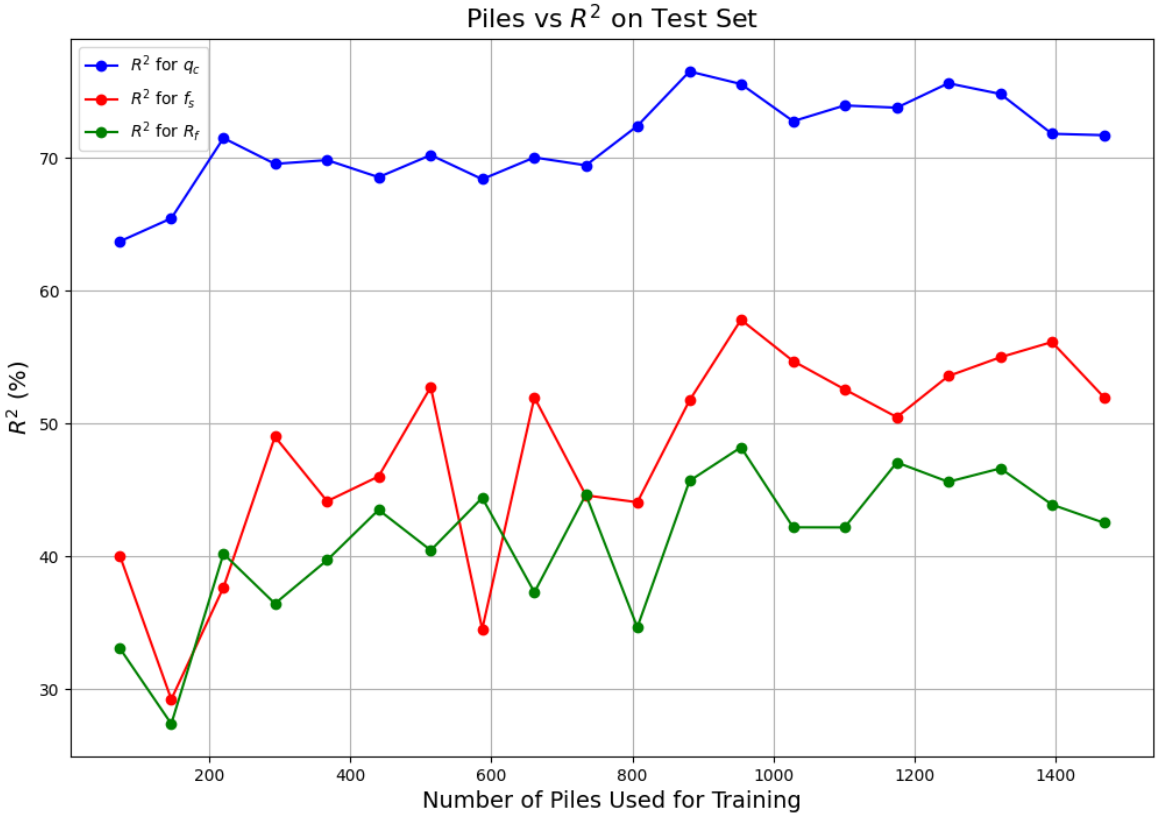
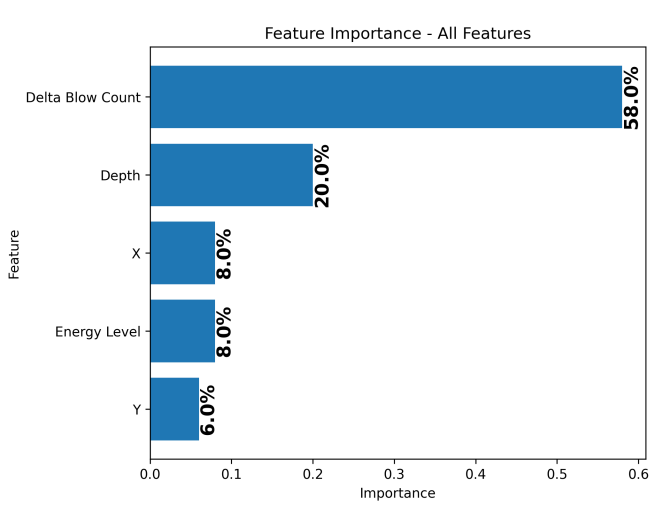
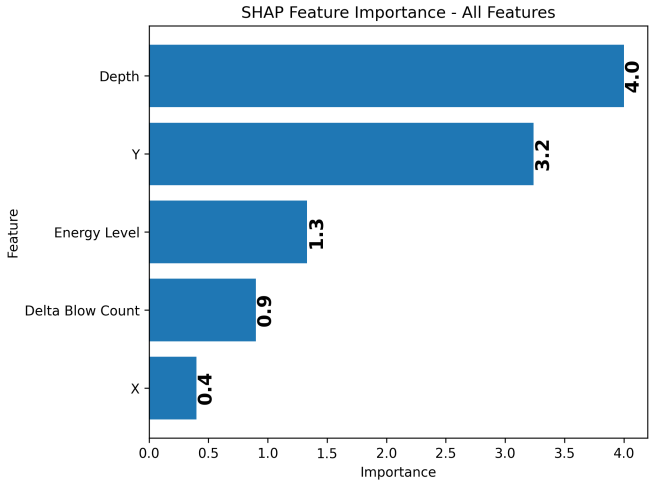



**Figure 25:** Performance of TabNet trained on increasing numbers of piles, demonstrating the trade-off between the amount of data and prediction accuracy.

## C.5 Impact of Feature Removal of Global Feature Importance Plots




**(a)** Built-in Global Feature Importance (GFI) for the model using all features. Importance values range between 0 and 1 and sum to 1, emphasizing features based on their contribution to the overall model structure.




**(b)** SHAP Global Feature Importance (GFI) for the model using all features. The largest SHAP value indicates the most influential feature, highlighting marginal contributions to predictions and capturing interactions often overlooked by built-in methods.

**Figure 26:** Comparison of Built-in Global Feature Importance (GFI) and SHAP Global Feature Importance (GFI). The left subfigure presents the built-in feature importance, while the right subfigure shows the SHAP-based feature importance, providing deeper insights into feature contributions and interactions.

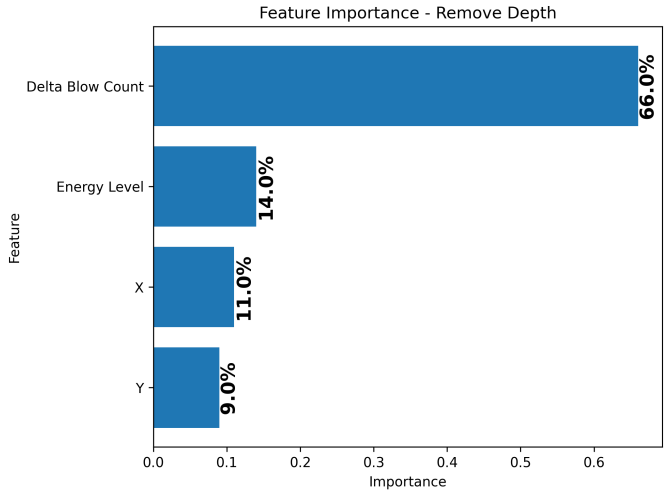



**Figure 27:** Built-in Global Feature Importance (GFI) for the model with depth removed. Importance values range between 0 and 1 and sum to 1, emphasizing features based on their contribution to the overall model structure.

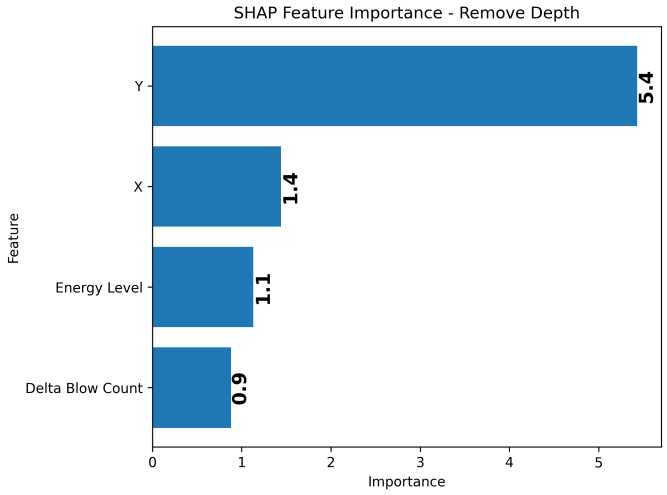



**Figure 28:** SHAP Global Feature Importance (GFI) for the model with depth removed. The largest SHAP value indicates the most influential feature, highlighting marginal contributions to predictions and capturing interactions often overlooked by built-in methods.

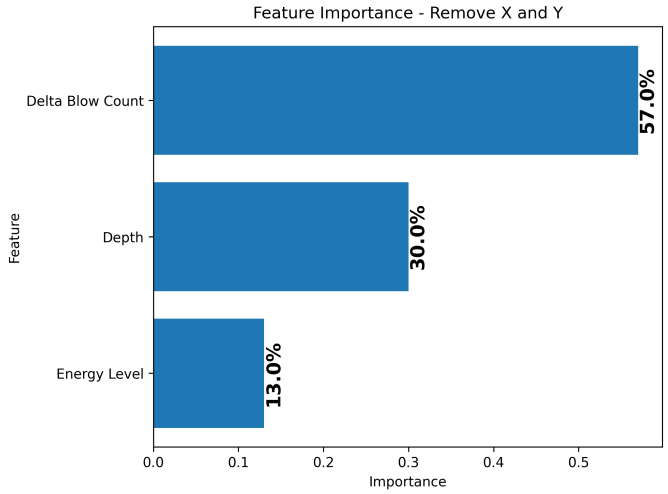



**Figure 29:** Built-in Global Feature Importance (GFI) for the model with X and Y removed. Importance values range between 0 and 1 and sum to 1, emphasizing features based on their contribution to the overall model structure.

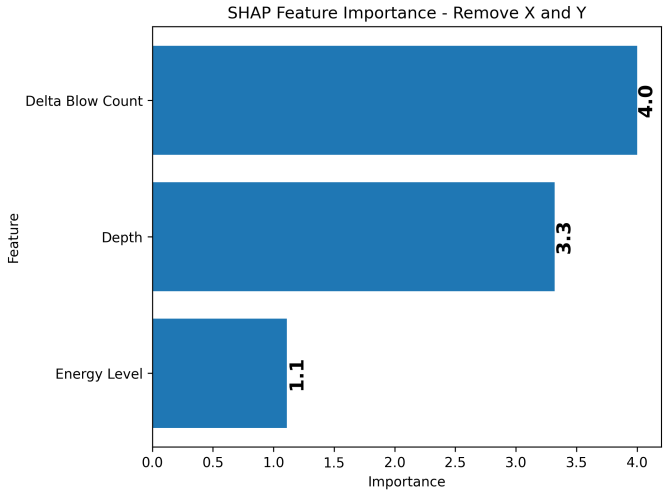



**Figure 30:** SHAP Global Feature Importance (GFI) for the model with X and Y removed. The largest SHAP value indicates the most influential feature, highlighting marginal contributions to predictions and capturing interactions often overlooked by built-in methods.

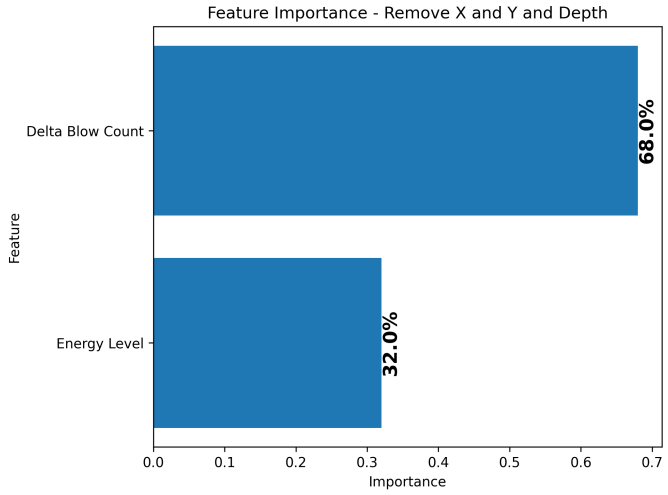



**Figure 31:** Built-in Global Feature Importance (GFI) for the model with X, Y, and depth removed. Importance values range between 0 and 1 and sum to 1, emphasizing features based on their contribution to the overall model structure.

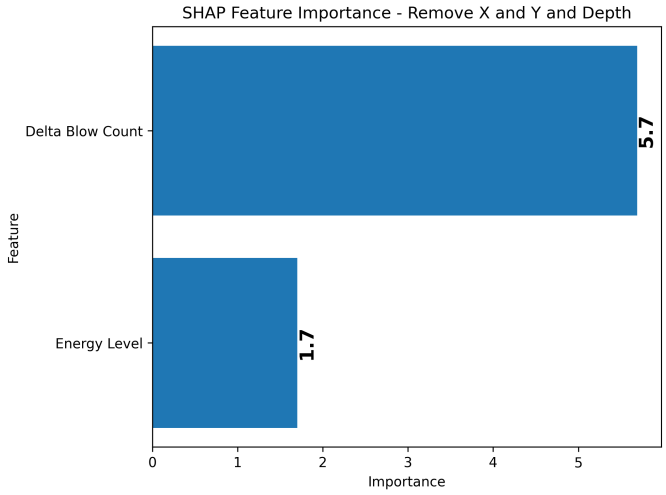



**Figure 32:** SHAP Global Feature Importance (GFI) for the model with X, Y, and depth removed. The largest SHAP value indicates the most influential feature, highlighting marginal contributions to predictions and capturing interactions often overlooked by built-in methods.

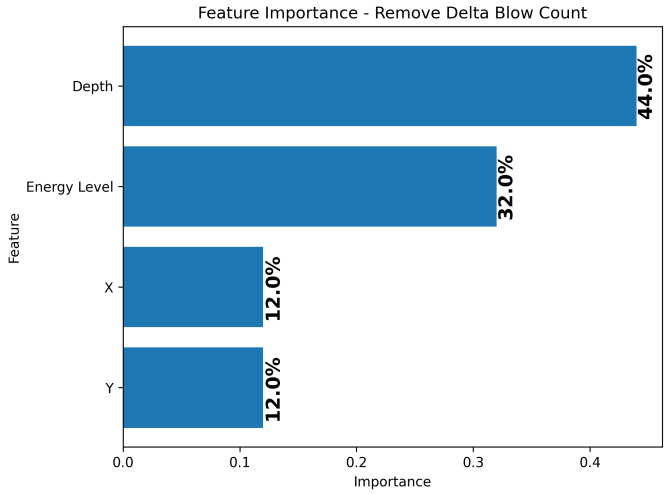



**Figure 33:** Built-in Global Feature Importance (GFI) for the model with Delta Blow Count removed. Importance values range between 0 and 1 and sum to 1, emphasizing features based on their contribution to the overall model structure.

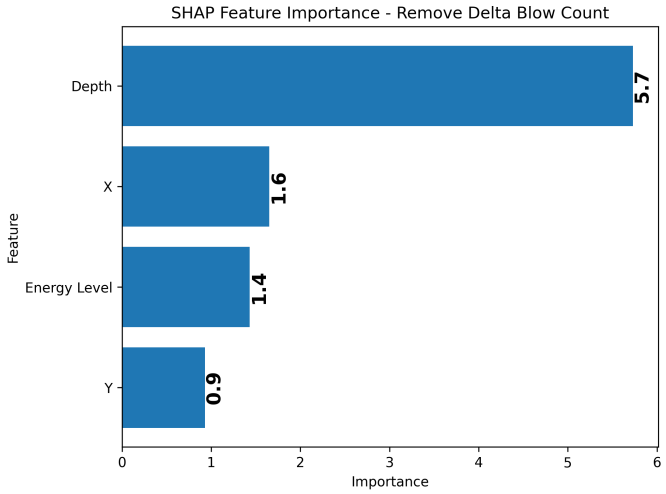



**Figure 34:** SHAP Global Feature Importance (GFI) for the model with Delta Blow Count removed. The largest SHAP value indicates the most influential feature, highlighting marginal contributions to predictions and capturing interactions often overlooked by built-in methods.

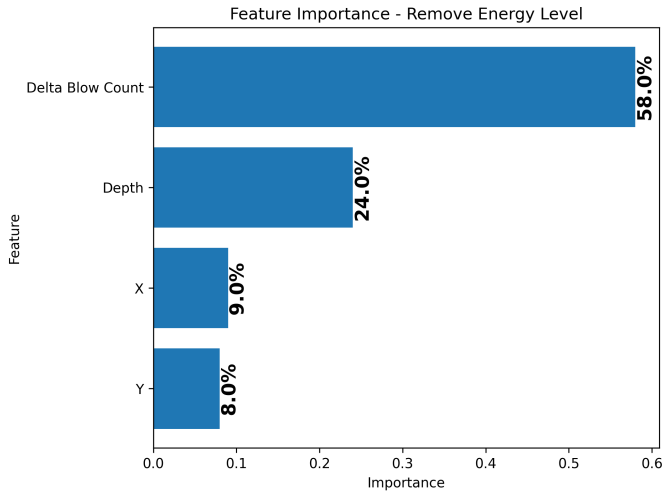



**Figure 35:** Built-in Global Feature Importance (GFI) for the model with Energy Level removed. Importance values range between 0 and 1 and sum to 1, emphasizing features based on their contribution to the overall model structure.

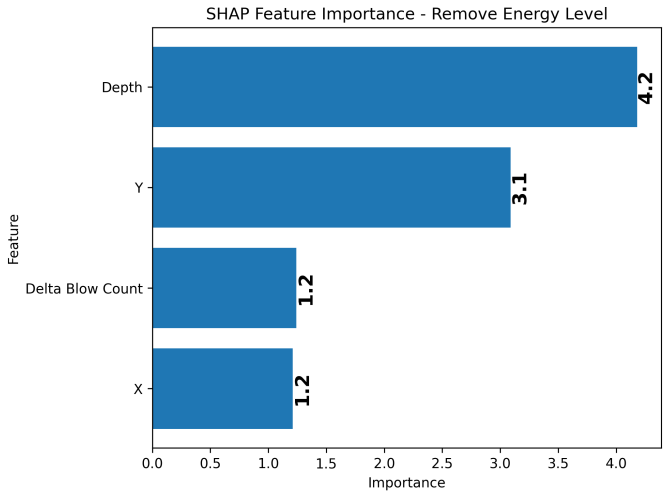



**Figure 36:** SHAP Global Feature Importance (GFI) for the model with Energy Level removed. The largest SHAP value indicates the most influential feature, highlighting marginal contributions to predictions and capturing interactions often overlooked by built-in methods.

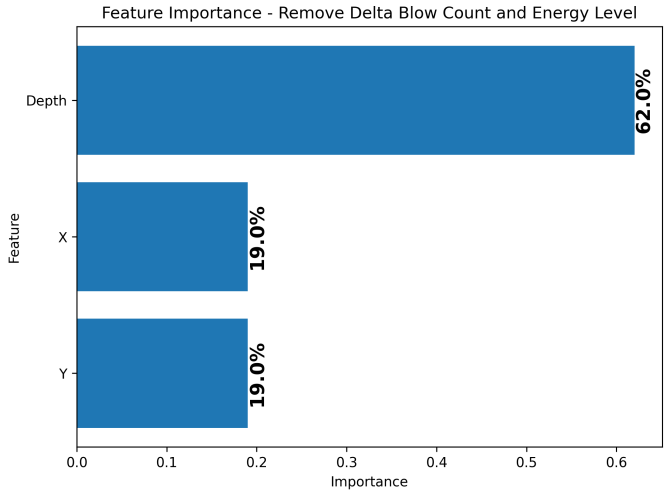



**Figure 37:** Built-in Global Feature Importance (GFI) for the model with Energy Level and Delta Blow Count removed. Importance values range between 0 and 1 and sum to 1, emphasizing features based on their contribution to the overall model structure.

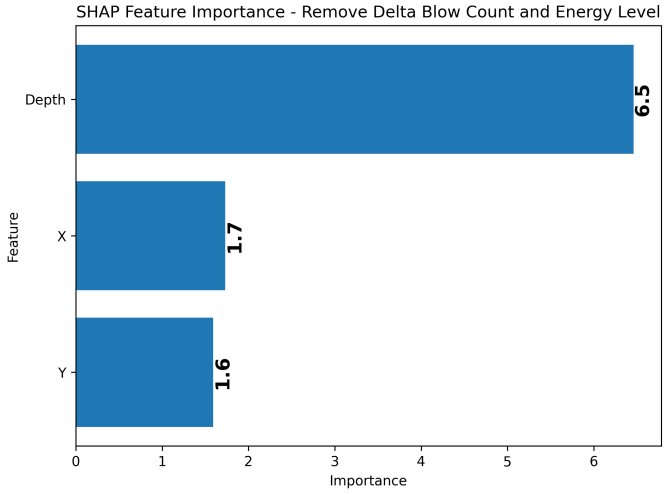



**Figure 38:** SHAP Global Feature Importance (GFI) for the model with Energy Level and Delta Blow Count removed. The largest SHAP value indicates the most influential feature, highlighting marginal contributions to predictions and capturing interactions often overlooked by built-in methods.

## Appendix D. Using the Saved ML Model for Predictions

This section provides a detailed, step-by-step guide for using the saved ML model to perform predictions on new examples. These steps ensure a systematic approach to applying the findings to real-world geotechnical engineering scenarios, as illustrated in Figure 6, which demonstrates the ground-truth and updated soil model. Follow the instructions below to replicate the process.

**Step 1: Clone the Repository**
Clone the repository to your local machine and navigate to the repository directory:

```
git clone <repository_url>
cd <repository_directory>
```

**Step 2: Install Dependencies**
Install the required Python libraries listed in the `requirements.txt` file:

```
pip install -r requirements.txt
```

Alternatively, install the required libraries individually:

```
pip install numpy pandas scikit-learn joblib matplotlib ...
```

**Step 3: Preprocess Raw Data**
Before obtaining the input CSV file, preprocess the raw data as outlined in Subsection 3.1. This includes steps such as cleaning, downsampling, feature engineering to prepare the data for model input. Ensure the final preprocessed dataset contains the required features and save it as a CSV file in your data directory.

**Step 4: Load the Input Data**
Once the input CSV file is ready, load it for predictions:

```
input_data = pd.read_csv("data/unseen_data.csv")
# Apply any required scaling or transformations here
```

**Step 5: Load the Model**
Load the trained ML model from the models directory:

```
model = joblib.load("models/cone_resistance_model.pkl")
```

**Step 6: Perform Predictions**
Use the loaded model to predict the target variables:

```
# Predict target variables
predictions = model.predict(input_data)

# Save the predictions
input_data["Predicted_qc"] = predictions[:, 0]  # Example for q_c
input_data.to_csv("results/predicted_results.csv", index=False)
```

**Step 7: Visualize Results**
If needed, visualize the predictions:

```
import matplotlib.pyplot as plt

# Example visualization
plt.plot(input_data["depth_NAPm"], input_data["Predicted_qc"], label="Predicted qc")
plt.xlabel("Depth (NAPm)")
plt.ylabel("Predicted Cone Resistance (qc)")
plt.legend()
plt.grid(True)
plt.savefig("results/prediction_plot.png")
plt.show()
```

**Step 8: Automate with a Script**
Create a script `scripts/predict.py` to automate the pipeline and run it as follows:

```
python scripts/predict.py
```

Ensure all paths are correctly set in the script for seamless execution.

**Notes**

- Validate the input data format to match the training data structure.
- Review logs in the results directory for performance metrics and debugging.
- Update the repository URL and paths as needed for your local environment.