Master's Thesis

Digital Soil Mapping based on PDFs of Cone Penetration Tests and Vibro Cores using Image Processing and Machine Learning

S.F. Ordeman

Student number:	4385969
Date of final version:	February 22, 2022
Master's programme:	Offshore & Dredging Engineering
Specialisation:	Dredging Engineering, Trenching & Deepsea Mining
Graduation Chairman:	Prof. dr. ir. C. van Rhee*
Graduation committee:	Dr. ir. A.M. Talmon*‡
	Dr. ir. J.D. Nuttall‡
	Dr. ir. M. Soleymani Shishvan*
	Dr. ir. F. Pisano*
External supervisors:	Ir. F.Y. Emha†
	Ir. C. de Rooij†

Faculty of Mechanical, Maritime and Materials Engineering

TU Delft Delft University of Technology

*Delft University of Technology
†Baggermaatschappij Boskalis BV
‡Deltares

Abstract

Digital Soil Mapping (DSM) of soil types in geotechnical project areas is a top priority. These maps are often used in decision making and can have significant consequences related to costs and risks. Usually, these maps are generated by digital soil models that interpolate soil types at known locations. In practice, conventional spatial interpolation techniques are still often used for DSM of soil types, such as inverse distance weighting and kriging. However, conventional models are not well suited for predicting or interpolating soil types because of their inability to deal with categorical data properly. Besides, the design of the conventional models does not allow for incorporating the abundance of meaningful covariate information that is available nowadays. The flexibility of machine learning algorithms vanquish both problems and has become increasingly popular for DSM of soil properties in recent years. The results of machine learning techniques for DSM of soil properties are promising and generally outperform conventional models. However, few studies have used machine learning for DSM of soil types and is therefore still a relatively unknown field. Moreover, at the time of writing, there are no studies that use sequence models for DSM of soil properties or types. Hence, the author proposes to introduce a new method for DSM of soil types, namely a Long Short-Term Memory (LSTM) network. The intuition behind this introduction is that the spatial correlation can be captured in sequences and can improve soil type prediction.

Real project data from a cable burial project is used to evaluate and compare the performance of the conventional interpolation methods triangulation and kriging, the machine learning models random forest and XGBoost, and the newly proposed deep learning model LSTM. The project data consist of 757 vibro cores (VC), 718 cone penetration test (CPT), bathymetry data and sub-bottom profilers. The geotechnical data, i.e. VCs and CPTs, is received on separate PDF pages that require to be digitized first. This thesis describes a simple yet precise manner to extract this data from the PDFs. The VCs and CPTs are provided with a soil type interpretation and can be used directly for developing the models. The data is split into a training set to develop/train the models and a test set for evaluation. Ultimately, the best performing model is used to build a 3D stratigraphic soil model for the project area with associated prediction accuracies.

All state-of-the-art techniques outperform the conventional models and especially in predicting minority classes. The best performing model is random forest with an overall accuracy of 85.44% and is comparable to the performance of XGBoost of 85.11%. LSTM network achieved a slightly lower accuracy of 84.27%. The results show that LSTM is suitable for DSM of soil types and has considerable potential for improvement as only a few possibilities of the model have been examined.

Contents

List of Figures

List of Tables

List of Symbols

β	Optimized coefficients
$\Delta\tau$	Lag distance
γ	Penalty term
$\hat{\mu}$	Estimated mean or deterministic component
$\hat{\rho}$	Estimated correlation function
$\hat{\sigma}$	Estimated standard deviation or sigmoid layer
$\hat{f}_i$	Tree structure
$\hat{y}_i$	Predicted output
λ	Optimized weights or penalty term
ω	Regularization term
ρ	Correlation function
τ	Distance
θ	Scale of Fluctuation
b	Bias vector
C_t	Cell state
e	Prediction error
g_i	First-order derivative of loss function
h_i	Second-order derivative of loss function
h_t	Current output
h_{t-1}	Previous output
k	number of variables

L	Length of CPT or loss function
n	Number of observations
Q	Input variables in general DSM formula
R	Stochastic component
S	Soil type
T	Total number of leaves
U	Weight matrix
W	Weight matrix
w	Output of a leave
x_t	Current input
x_{t-1}	Previous input
y_i	Output
Z	Dependent variable

List of Abbreviations

Adam Adaptive Moment Estimation.

ANN Artificial Neural Network.

BLUP Best Linear Unbiased Predictor.

CCE Categorical Cross Entropy.

CPT Cone Penetration Test.

CV Cross-Validation.

DSM Digital Soil Mapping.

ESBN European Soil Bureau Network.

FN False Negatives.

FP False Positives.

GT Geotechnical Test (VC or CPT).

IDW Inverse Distance Weighting.

LSTM Long Short-Term Memory.

ML Machine & Deep Learning.

MSE Mean Squared Error.

OvR One versus Rest.

PDF Portable Document Format.

RNN Recurrent Neural Network.

SBT Soil Behaviour Type.

SMOTE Synthetic Minority Oversampling Technique.

SoF Scale of Fluctuation.

TIN Triangulated Irregular Network.

TN True Negatives.

TP True Positives.

VC Vibro Core.

Terminology

bootstrapping Random sampling with replacement. Normally, the size is equal to the size of the original dataset.

categorical variable A variable based on a qualitative property rather than a quantitative property as is the case of a continuous variable. A categorical variable assigns each observation to a particular group or nominal category, such as soil types in this case.

dropout Regularization technique for neural networks. Dropout randomly omits neurons during training, forcing the model to be more general. Dropout of 0.5 means that half of all neurons are omitted in an epoch.

entropy measurement of impurities or randomness in the data points. If all elements belong to a single class, then it is termed as "Pure", and if not then the distribution is named as "Impurity". $Entropy = -\sum_{i=1}^{n} p_i \cdot Log_2(p_i)$.

epoch A training a neural network with all training data for one cycle. In an epoch, all training data is used exactly once.

feature Explanatory variable in machine learning.

Gini Index degree of probability of a specific feature that is classified incorrectly when selected randomly. $GiniIndex = 1 - \sum_{i=1}^{n} (P_i)^2$.

hyperparameter A parameter which controls the learning process of a machine learning model. The hyperparameters are set by the user before training the model and are therefore not optimized during training. Consequently, hyperparameters should be optimized by the user.

imbalanced dataset A dataset characterized by a disproportionate ratio between classes.

information gain gained information by making a split. Information gain is equal to entropy before splitting minus entropy after splitting.

label A class. In this case a soil type.

leaf An end node of a tree representing a prediction. In this case a soil type.

logistic function A sigmoid function that outputs a value between 0 and 1. For example used in a LSTM network to decide whether the cell state is totally 'forgotten' 0, totally 'remembered' 1 or anything in between 0 and 1.

node Part of the structure of a decision tree where a decision is made. A node splits the data in two based on a feature that minimizing a loss function.

outlier An observation that does not seem to fit with the rest of the data.

overfitting Fitting a model that is too closely related to the training set and therefore too specific for new data.

regularization A process that helps to prevent overfitting, which is performed by a regularizer.

sigmoid layer A mathematical function with a S-shaped curve. Examples are a logistic function and hyperbolic tangent. These functions are used in neural networks to push values in the range from 0 to 1 or in the range from -1 to 1.

Chapter 1

Introduction

1.1 Background Information

For many years people have been trying to map soil. The first known soil maps date from the 18^{th} century for agricultural purposes (Minasny & Mcbratney, 2015). Later on, soil mapping becomes essential to determine, among other things, the bearing capacity, settlement or strength of the soil. These soil characteristics can be determined using a soil profile. A soil profile is a vertical intersection of the soil with the soil layers present and is crucial for all kinds of projects where soil investigation is required. Based on soil investigation, the soil profile can be established. Commonly, these soil profiles are used to create a map or a model of an area which is known as Digital Soil Mapping (DSM).

For many onshore and offshore projects, a thorough soil investigation in combination with DSM is of great importance. Take, for example, a bottom-founded structure, more certainty about the soil profile, and thus the bearing capacity can drastically reduce the size of the foundation. Implicating that there is a trade-off between the thoroughness of the investigation and the uncertainty of the soil profile model. However, a thorough soil investigation can be costly and does not necessarily lead to a better or more reliable soil model. For this trade-off, an optimum is sought depending on the risk taken and costs. The maximum risk is often predefined by the company's regulations or law, resulting in optimum only depending on costs.

There will always be interest in an enhanced soil model, as it can significantly reduce the costs of almost all projects requiring soil investigation. More certainty can reduce the project's costs by making less conservative assumptions and calculations. In addition, a less thorough soil investigation is required to achieve the same certainty in the soil profile model.

1.2 Problem Definition

Offshore wind farm generated electricity is transported to onshore facilities via export cables. These cables are laid on the seabed, and to protect these cables from external hazards, they can be covered with rocks, mattresses or trenched into the seabed. In the case of trenching, the burial

process is usually performed by a trencher, plough or a sledge. The success of trenching/cable burial depends a lot on the availability of sufficient soil information within the shallow seabed along the cable route. Oftentimes geotechnical data is collected at a fixed interval along the cable route. Then, in combination with the geophysical data, a soil profile model is developed whereby the quality of the model depends on the proper integration of these two datasets. For the areas between the geotechnical testing locations, interpolation techniques are required.

In practice, conventional spatial interpolation techniques are still often used for DSM of soil types, such as Inverse Distance Weighting (IDW) and kriging. However, conventional models are not well suited for DSM of soil types because of their inability to deal with categorical data properly. This problem is encountered by the subsea cable department of Boskalis. This department advises which burial tool can be used best for trenching cables into the seabed. Their advice is based on a soil profile model developed in ArcGIS Pro, among others, where only conventional methods are available. Hereby, it is common that Boskalis receives the geotechnical tests (GTs), where the model is based on, in Portable Document Format (PDF). PDFs have an inconvenient data format to work with, and it takes geotechnical engineers an unnecessary amount of time to manually convert it to a workable format and analyze it. Automation is desired where the PDF is directly converted to a workable format.

1.3 Research Goal

The aim of this thesis can be divided into a practical and an academic component. The practical component is to digitize the geotechnical data from a PDF to a workable format, such as an Excel or a comma-separated value file. The geotechnical data includes Vibro Cores (VCs), Cone Penetration Tests (CPTs), both requiring digitization of the soil interpretations and the CPT graphs. Then, the digitized data is enriched with geophysical data, after which a 3D stratigraphic soil model is built.

The academic aim of this thesis is to evaluate the performance of conventional models against state-of-the-art techniques and to a newly proposed model for DSM of soil types. The soil interpretation provided with the GTs is considered ground truth and can directly be interpolated to build the soil profile model. Currently, the most popular spatial DSM techniques are kriging and deterministic interpolation methods, such as IDW and nearest neighbours. However, in recent years, the popularity of machine learning algorithms for DSM of soil properties has increased. Algorithms such as random forest and gradient boosting are nowadays often used (Sekulić et al., 2020). The results of machine learning techniques for DSM of soil properties are promising and generally outperform conventional models. HHowever, few studies have used machine learning for DSM of soil types and is therefore still a relatively unknown field. Moreover, at the time of writing, there are no studies that use sequence models for DSM of soil properties or types. For this reason, the author proposes to introduce a new method for DSM of soil types that have been proven in other fields, the Long Short-Term Memory (LSTM) network (S. Wang & Jiang, 2015; Yao & Guan, 2018; J.-H. Wang et al., 2018; Y. Wang, Zhu,

& Li, 2019; Zhang et al., 2019). This neural network is known from time series problems but is used, with success, in different fields as well, such as in natural language processing. The intuition behind this introduction is that the spatial correlation can be captured in sequences and can improve soil type prediction.

The implemented techniques that are used for DSM of soil types are the conventional interpolation methods triangulation and kriging, the machine learning models random forest and XGBoost, and the newly proposed deep learning model LSTM. The DSM models are evaluated on real project data from a cable burial project. The final product is a more accurate soil map with probabilities that gives an indication of certainty.

1.4 Relevance to Trenching

An algorithm that is consistently better at generating soil maps is of great value for many fields, including trenching. For instance, the best burial tool is selected based on the in situ soil conditions. As a consequence, an accurate soil map is vital for selecting a suitable burial tool. Mapping the soil conditions more accurately provides certainty for selecting the most suitable burial tool. In addition, with a more accurate soil model, the performance of a burial tool can be evaluated more precisely. The performance is continuously monitored and logged during burial. By assessing the logging data and relating this performance to a more accurate soil model, it becomes more apparent in which circumstances the burial tool performs well or poorly. The field experience can then be used to improve burial tool models for future projects.

Furthermore, predicting more accurately critical soil types, in this case, peat, can significantly reduce the cost of a trenching project. For instance, if all peat along the cable corridor has to be excavated, it is plausible that a substantial amount of soil is excavated without peat. Predicting where peat is located with more certainty means less abundant material has to be excavated.

Another worth mentioning contribution of this research is the contribution to the innovation of a rather traditional field. Nowadays, the current hype of becoming more data-driven has touched many research fields. It is vital that trenching keeps pace with innovations in the field of data to remain interesting for young engineers and improve the trenching activities.

The field of trenching is ideally suited to become more data-driven due to the amount of available data. The burial tools are equipped with numerous sensors and cameras which record a vast amount of data. Resulting in almost a limitless amount of research possibilities.

1.5 Research Questions

To accomplish the research goals stated in Section 1.3, research questions have been formulated. The main research question is the following:

Are conventional spatial interpolation techniques the best way for digital soil map-

ping of soil types, or does machine learning offer an opportunity for improvement?

In addition, sub-research questions have been formulated that relate to other relevant parts of this research. The research can be divided into 1) digitizing the geotechnical data; 2) predicting the locations to be excavated: 3) evaluating the performance of conventional techniques against state-of-the-art techniques and 4) analyzing the suitability of an LSTM for DSM of soil types. The following sub-research questions have been formulated:

1. How accurate is the digitizing tool in digitizing cone penetration test graphs compared to the measured data, and can this product be used in practice?
2. Can the implemented models assist in determining which sites contain peat and need to be excavated?
3. How do the machine and deep learning models perform compared to the conventional models in the digital mapping of soil types?

1.6 Approach

In order to answer the research questions, the approach shown as a flow chart in Figure 1.1 is used. The first step is digitizing the PDFs to a workable format. Subsequently, the digitized data is split into a training and a test set. The training set is used to develop and train the models, whereafter the models are evaluated on the test set. Eventually, the best performing model is used to develop a complete 3D stratigraphic soil model along the cable corridors.

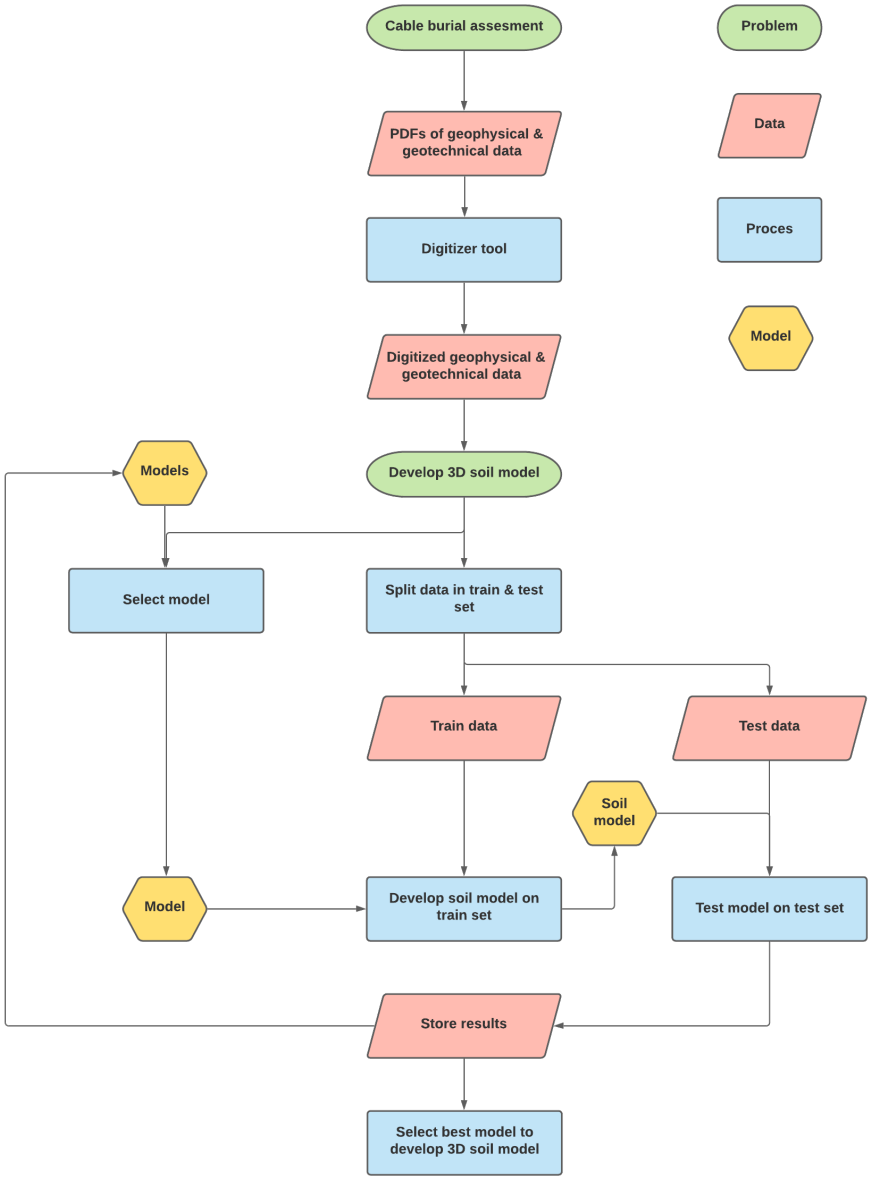


Figure 1.1: This flowchart represent the intended approach for answering the research questions. There is a loop in the diagram where in each loop a different model is trained and evaluated on the test set.

1.7 Thesis Outline

The structure of the remainder of this research is as follows:

- Chapter 2 provides the literature review. This chapter provides geotechnical background as well as the theory of the models used to generate a digital soil map. Potential explanatory variables and the problematic appearance of an imbalanced dataset are also addressed. This chapter finalizes with concluding remarks of the literature study.
- Chapter 3 provides a description of the used data. It also describes how the explanatory

variables are derived from the data and their correlation. Completing this chapter with the interpretation of the present soil types.

- ≻ Chapter 4 elaborates on the methodology used. First, a section on image processing describes the technique used to extract the data from the PDFs. This is followed by the procedure of splitting the data into a training and test set. The next section covers the data preprocessing steps. Selecting the best model by optimizing and thereupon a final evaluation is described in the section Model Selection. This chapter ends with the methodology to develop a 3D soil model.

- ≻ Chapter 5 discusses the results. The results of hyperparameter optimization are addressed first. Then, a ranking of the most important features is provided. The following section presents the results of the final models. Concluding with an analysis of the results.

- ≻ Chapter 6 finalizes with the conclusions. Starting with the findings, then answering the research questions and finalizing with recommendations for future research.

Additionally, an appendix supplements the thesis after the references.

Chapter 2

Literature Review

This chapter elaborates the theory on this subject as well as existing literature. The first section discusses the geotechnical background related to the problem. Secondly, the theory of the applied digital soil models is explained and why they were selected. The next section discusses the problematic appearance of an imbalanced dataset. This is followed by addressing the potential features and why they could be used as explanatory variables. The literature review concludes with a discussion of the findings.

2.1 Geotechnical Background

2.1.1 Cable Burial Methods

The selection of the best trencher, plough or sledge for the burial process is based on the soil conditions along the cable corridor. Different burial tools perform better in different circumstances. For instance, the top layers are critical for the stability of a burial tool. In case the burial tool is too heavy with respect to its contact surface, the possibility exists that the burial tool sinks into the seabed. Besides, the topsoil layer affects the traction of the trencher as well. The tracks of a trencher must have sufficient traction to move and should not slip away. This is only relevant for a trencher because a plough or a sledge is towed by a vessel. The soil layers in the complete soil profile are decisive for selecting a trenching methodology. Cable burial can be performed by three main trenching methods:

- Mechanical cutting: Soil is cut away mechanically by a cutting chain or cutting disc
- Ploughing: Soil is cut/opened by a passive tool forming a trench
- Jetting:
    1. Soil is fluidized by water released under medium pressure and high flow rates so that a cable can sink into the soil
    2. Soil is cut away by water released under high pressure and low flow rates

Figure 2.1 shows which trenching method should be considered depending on the soil characteristics.

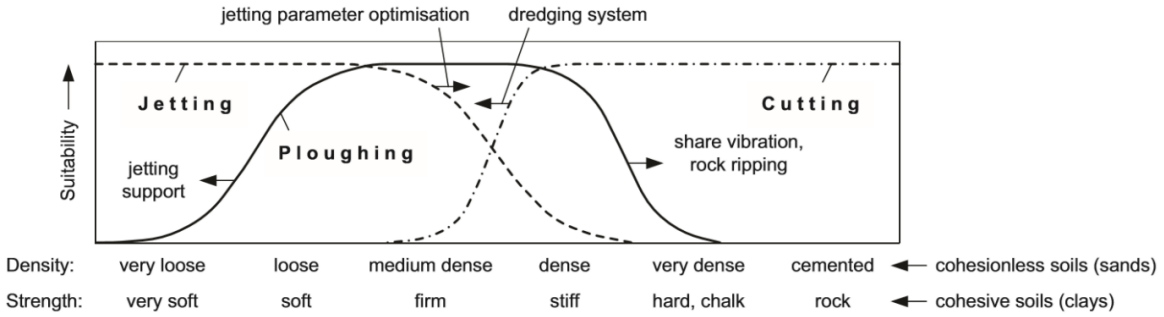


Figure 2.1: An overview to determine which tools to consider for installation in a range of soil conditions (Linnane, 2019).

2.1.2 Digital Soil Mapping

Creating a 3D soil model that shows the soil types at each location is a branch of (DSM). The European Soil Bureau Network (ESBN) defines DSM as follows: "Computer-assisted production of digital maps of soil type and soil properties. It typically implies the use of mathematical and statistical models that combine information from soil observations with the information contained in correlated environmental variables and remote sensing images" (Dobos et al., 2006). These digital maps are produced by digital soil models or DSM models that aim to predict soil types or soil properties based on soil observations and auxiliary spatial data. McBratney et al. (2003) propose a general formulation for these models through the following equation:

$$S = f(Q) + e \tag{2.1}$$

where stands S for soil type because this research concerns the prediction of soil types; Q is the input variables, and e is the prediction error. In other words, the soil type S is predicted by a function f that is performed on the input variables Q. It is essential that Q contains pedologically meaningful predictor variables defined at locations $[X, Y, Z]$. Once the DSM model is fitted at these locations, the DSM model can be used to predict new data and create a soil map. According to the ESBN, there are three great ways of building DSM models to predict soil types.

- **Data mining** - detecting unknown relationships between the predictor variables Q and the predicted variable S.
- **Geostatistical** - interpolation of soil properties from soil observations using spatial correlations between the soil observations.

- **Soil surveyor** - based on the experience and knowledge of the soil surveyors in a given region.

2.1.3 Soil Classification

Before the soil type of a sample can be predicted, the soil needs to be classified first. Soil is classified on characteristics such as colour, structure, texture, consistency and degree of acidity or alkalinity (Ritzema, 1994). According to International Standards ISO 14688-1, seven main types can be distinguished based on particle size fractions, from large to small: 1) Large boulder; 2) Boulder; 3) Cobble; 4) Gravel; 5) Sand; 6) Silt and 7) Clay. The table from ISO 14688-1 is presented in Table 2.1 including the main types, associated sub-types and their corresponding particle size fraction. In nature, soils generally have a range of different particle sizes and cannot be assigned to a single particle size fraction. Mixtures of different particle size fractions are often classified using the texture triangle of the United States Department of Agriculture shown in Appendix A. In addition, there are a few organic soil types depending on the organic content. When the organic content in soil is more than 50% of the volume, it is called peat. Mud soils have an organic content in the range of 20% to 50%, organic soils in between 15% and 20% and mineral soils less than 15% (Ritzema, 1994).

After the soil is classified, soil layers can be identified. The vertical section of the soil, through all its layers, is defined as the soil profile (Ritzema, 1994). However, an soil profile might not contain distinct boundaries (Phoon et al., 2021). Based on the soil profile, one can perform geotechnical calculations, such as determining the bearing capacity of the soil, which is important to determine the dimensions of foundations. Or in this case, one can determine the best burial tool for cable burial based on the soil conditions.

2.1.4 Site Characterization

Currently, site characterization is based on geotechnical data combined with geophysical data. Commonly, the geotechnical data consists mainly of CPTs and additionally VCs. Typically, the CPTs are classified using the Soil Behaviour Type (SBT) index of Robertson 1990 (Phoon et al., 2021). This classification is based on the normalized tip resistance and the normalized friction ratio of the CPT. In Figure 2.2 an SBT index can then be read for each measured value and assigned to a region. These regions correspond to an SBT. The VCs, if collected, are classified in the laboratory.

The challenge in site characterization is properly integrating geotechnical and geophysical data. The geotechnical data consist of sparse 1D GTs, and the geophysical data is typically 2D and occasionally 3D seismic imaging from the surveys. The geophysical data is crucial for interpolating the soil interpretation of the geotechnical data as it is the only type of data that allows regional interpretation and understanding of site conditions (Sauvin et al., 2019). Geotechnical engineers rely heavily on local knowledge to interpolate layer boundaries, and

Table 2.1: Particle size fractions ISO standard 14688-1

Soil group	**Particles size fractions (symbol)**	**Particles size fractions (mm)**
Very coarse soil	Large boulder (lBo)	>630
	Boulder (Bo)	>200 to ≤630
	Cobble (Co)	>63 to ≤200
Coarse soil	Gravel (Gr)	>2.0 to ≤63
	Coarse gravel (cGr)	>20 to ≤63
	Medium gravel (mGr)	>6.3 to ≤20
	Fine gravel (fGr)	>2.0 to ≤6.3
	Sand (Sa)	>0.063 to ≤2.0
	Coarse sand (cSa)	>0.63 to ≤2.0
	Medium sand (mSa)	>0.2 to ≤0.63
	Fine sand (fSa)	>0.063 to ≤0.2
Fine soil	Silt (Si)	>0.002 to >0.063
	Coarse silt (cSi)	>0.02 to ≤0.063
	Medium silt (mSi)	>0.0063 to ≤0.02
	Fine silt (fSi)	>0.002 to ≤0.0063
	Clay (Cl)	≤0.002

according to Phoon et al. (2021) of the layer boundaries involves a certain degree of guesswork. Consequently, the soil models are still qualitative rather than quantitative as the integration of geophysical and geotechnical data is not yet fully implemented (Sauvin et al., 2019).

This study approaches site characterization from a more data-driven perspective than the conventional approach. According to Phoon et al. (2021) "data-driven site characterization" refers to site characterization that relies solely on measured data. In this study, this is not the case because the soil interpretations of the companies were used. However, using machine learning with as little human intervention as possible to deliver work more consistently and efficiently is a more data-driven approach than usual. In current data-driven site characterization studies the CPT sounding is the commonly used data source because it is the only near-continuous record that is commonly available (Phoon et al., 2021). Robertson 1990 is not suitable for classifying these CPTs as it does not allow for using other data. Recent studies use more advanced ways the classify the measured data from the CPT, such as machine learning methods and convolutional neural networks. These models obtained extremely high accuracies, even up to 99% per cent with a random forest (Rauter & Tschuchnigg, 2021). These results are promising and show that machine learning can classify soil based on measured data.

Table 2.2: Soil behaviour type index (Robertson, 1990))

Soil behaviour type index, I_c	**Zone**	**Soil behaviour type (SBT)**
-	9	Very stiff fine-grained
-	8	Very stiff sand to clayey sands
$I_c < 1.31$	7	Gravelly sand to dense sand
$1.31 < I_c < 2.05$	6	Sands: clean sand to silty sand
$2.05 < I_c < 2.60$	5	Sand mixtures: silty sand to sandy silt
$2.60 < I_c < 2.95$	4	Silt mixtures: clayey silt to silty clay
$2.95 < I_c < 3.60$	3	Clays: silty clay to clay
$I_c > 3.60$	2	Organic soils: peats
-	1	Sensitive fine-grained

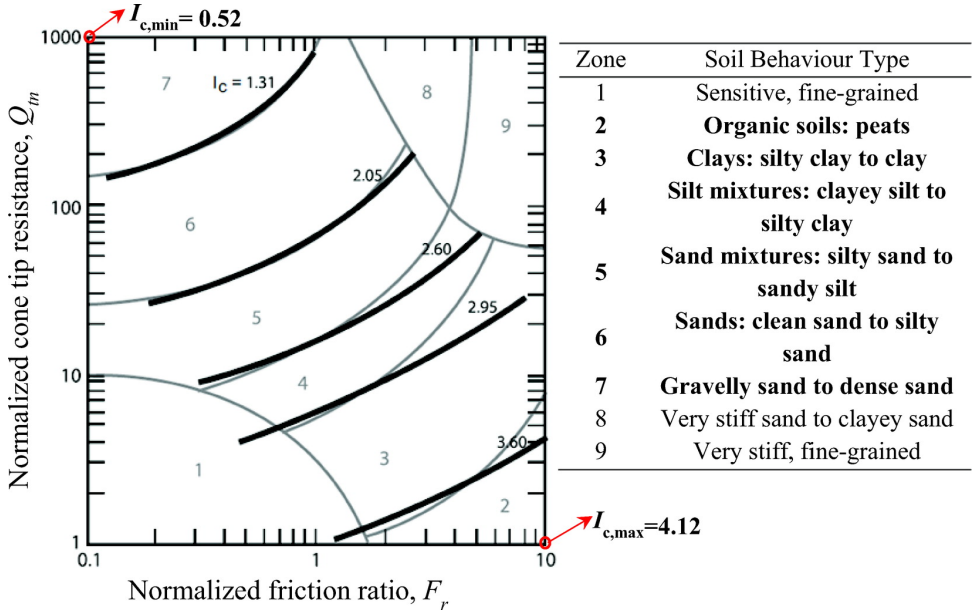


Figure 2.2: Figure to determine the soil behaviour type of measured CPT values with Robertson 1990 (Cao et al., 2019).

2.1.5 Spatial Variability

Soil properties vary in space, and are rarely uniform or homogeneous with depth. Often it varies in horizontal direction as well. Homogeneity in soil characteristics is the exception rather than the rule (Ritzema, 1994). Even within layers of the same soil type, soil properties can show considerable variation from one location to another. Geology and the conditions during soil deposition are associated with this variability (Yan & Guo, 2015). Barrette (2011) stated that the grain size of the soil is a function of water energy. Transporting large grain sizes requires more energy than finer sediment. Therefore, less coarse material remains longer in suspension. Moving further offshore, currents, waves, and tides become less powerful. For this reason, the

larger grain sizes, gravel and sand, are often found in rivers, sand and silt in delta, and silt and clay further offshore and in deeper environments.

Phoon & Kulhawy (1999) argues that there are three primary sources of uncertainty observed in soils, i.e., spatial variability, measurement error and transformation of laboratory measurements into soil properties. Herein, spatial variability is one of the primary sources of uncertainty in stochastic soil models (Lloret-Cabot et al., 2014). The variability is generally characterized by the Scale of Fluctuation (SoF). SoF describes the correlation of parameters of soil in relation to the distance. A large SoF indicates a more homogeneous soil, whereas a smaller SoF indicates a more heterogeneous soil.

2.1.6 Scale of Fluctuation

Many studies demonstrated the necessity of considering spatial variability in geotechnical applications (Cami et al., 2020). In particular, SoF plays a key role in describing soil variability at a site. To obtain more realistic results, it is crucial to estimate accurate values of the vertical and horizontal scales of fluctuation when using advanced probabilistic approaches (Lloret-Cabot et al., 2014).

In literature, reported scales of fluctuation generally indicated that the horizontal SoF is larger than the vertical SoF (Phoon & Kulhawy, 1999). The cause of this difference can be explained by deposition processes (Gast et al., 2018). One can imagine that soil characteristics in horizontal direction gradually changes during deposition, while more abrupt changes are manifested in vertical direction. The ratio between the two scales of fluctuation is known as anisotropy. When the scales of fluctuation in horizontal and vertical direction are equal, then the soil is called isotropic. As stated in Section 2.1.5, this is hardly ever the case.

Determining the horizontal SoF based on site-specific investigation data can be very challenging. Most site investigation methods, such as CPTs, are performed vertically. As a consequence, only the vertical variation of soil is explored. Because a single test can obtain more than sufficient information in vertical direction, but not in horizontal direction (Ching et al., 2017).

Estimation based on Tip Resistance

For estimating the SoF, various methods are available. Lloret-Cabot et al. (2014) use the tip resistance, q_c, of CPTs to estimate the horizontal and vertical SoF. The theoretical correlation model is then best fitted to the available data. The theoretical correlation model is as follows:

$$\rho(\tau) = \exp\left(\frac{-2|\tau|}{\theta}\right) \tag{2.2}$$

and the estimated correlation function:

$$\hat{\rho}(\tau_j) = \frac{1}{\hat{\sigma}^2(n-j)} \sum_{i=1}^{n-j+1} (X_i - \hat{\mu})(X_{i+j} - \hat{\mu}) \tag{2.3}$$

where $\hat{\mu}$ is the estimated mean; $\hat{\sigma}$ is the estimated standard deviation; n is the number of observations; $\tau_j = j\Delta\tau$ with $j = 1, 2, .., n$ and $\Delta\tau$ is the lag distance, i.e., the distance separating two points. Resulting in a correlation function that describes the correlation between points for a given separation distance. The rule of thumb is to keep lag distance shorter than the SoF and shorter than $\frac{L}{4}$, where L is the length of the CPT (Cami et al., 2020). In horizontal direction, this is not practicable because the available CPTs in this study are several hundred meters apart, which is already way larger than the horizontal SoF. Moreover, the CPTs are not equidistant from each other. For that reason, $\Delta\tau$ is not constant.

Estimation based on Detected Soil Layers

Although tip resistance is closely related to soil type, it does not directly imply a change in the soil layer. Therefore, estimating the SoF based on the tip resistance is not the most obvious choice. Hence, the detected soil layers by the VCs and the CPTs are used for estimation. Both companies provided for the VCs and CPTs an interpretation of the soil. The vertical SoF is estimated by a simple calculation of the average number of detected soil layers of all GTs in the surroundings. The horizontal SoF is estimated, in the same surroundings, by calculating how many soil layers are detected at each depth beneath the seafloor. Meaning that there is a single vertical SoF at location [x, y] for all depths and an estimated horizontal SoF for each depth on location [x, y].

2.1.7 Geostatistics

Geostatistics originated in the mining industry and was first developed in the early 1950s. Minerals such as ore are often found in highly concentrated veins and are not evenly distributed over an area (Ecker, 2021). This implies that the concentration of ore exhibits spatial correlation. Tobler described this phenomenon in 1970 in his first law of geography, "everything is related to everything else, but near things are more related than distant things." (Tobler, 1970). Ignoring his first law in computations and by using only point statistics of soil parameters, such as the mean and standard deviation, leads typically to over-estimation and over-conservative design. Including the spatial correlation, i.e. SoF, a more accurate representation can be achieved (Gast et al., 2018).

Hence, classical statistics were found unsuitable for estimating data with autocorrelation or spatial correlation. In fact, many statistical analyses treat the data as an Independent and Identically Distributed (IID), rather than one spatially correlated dataset of observations (Ecker, 2021). Matheron, one of the founders of geostatistics and kriging, recognized that prediction is not the only element in geostatistical analysis. He concluded that it consists of two fundamental elements, namely the correlation structure and prediction of conditions at unsampled locations. The first step is to explain the clustering mechanism, i.e., develop a model for the correlation structure. Secondly, the proposed model is used to predict the target variable at unsampled locations or areal units (Ecker, 2021).

2.2 Digital Soil Mapping Models

This section describes the applied models to map the soil types. In this study, two conventional models, two machine learning models and a deep learning model were implemented. Machine and deep learning models are adaptive and require a training set to optimize. Since deep learning is a subset of machine learning, the overarching abbreviation ML is used in the remainder of this report when referring to both groups. Otherwise, the full designation is used. The theory of the models is explained in the following sections. The following models are used:

1. Triangulated Irregular Network (conventional model)
2. Universal Kriging (conventional model)
3. Random Forest (machine learning model)
4. Extreme Gradient Boosting (machine learning model)
5. Long Short-Term Memory network (deep learning model)

Figure 2.3 shows an overview of the implemented models. The two conventional models were selected because they were used in this specific project. The objective is to compare these conventional models to state-of-the-art models. The other three models mentioned have been selected based on their characteristics that match the provided data and problem. Random forest is selected based on its hierarchical structure, which is beneficial for class imbalances and its predictive accuracy (Li et al., 2011). XGBoost has been selected for its high performance in many different fields (Bentéjac et al., 2019). The last model and most captivating one is the LSTM network. Originally, this is a time series model, which is translated into a spatial model, and that is a first. The subsequent subsections also include the pros and cons of the models.

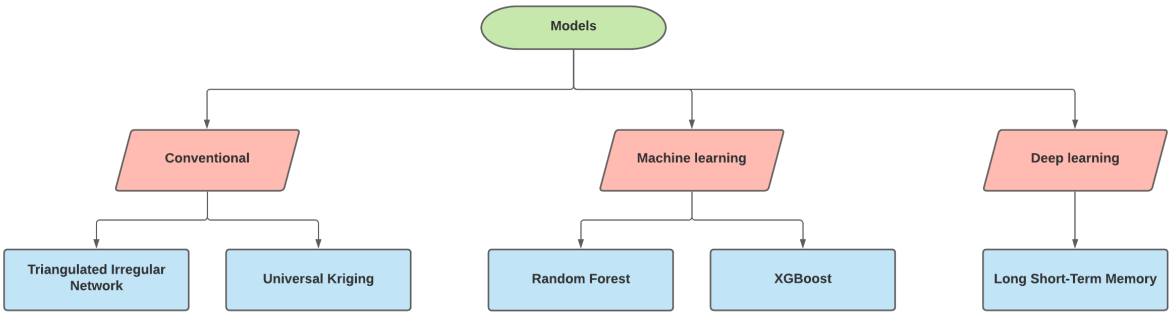


Figure 2.3: An overview of the implemented models used for DSM.

2.2.1 Conventional Models

Classical spatial interpolation techniques can be roughly classified into three categories 1) deterministic or non-geostatistical interpolators, such as IDW; 2) stochastic or geostatistical interpolators, such as ordinary kriging and 3) combined methods, such as regression kriging (Li,

2008). The performance of these methods is often affected by many factors, such as sampling density, sample spatial distribution, sample clustering, surface type, data variance, data normality, quality of secondary information, stratification and grid size resolution (Li & Heap, 2011). On top of that, there may be interaction between different factors, which makes it even more challenging to select an appropriate model for any given dataset. Relatively simple interpolation techniques, e.g. IDW, are used frequently because of their simplicity and availability. However, these models are often associated with significant prediction errors (Li et al., 2011). For this specific problem, the conventional models differ from the ML models because it interpolates single values while the ML models predict categorical variables*, i.e. soil types. Eventually, all models are tested on the test set, after which the performance is compared.

2.2.1.1 Triangulated Irregular Network

Triangulated Irregular Network (TIN) is a simple interpolation technique that creates a network formed by triangles of nearest neighbour points. In this case, Delaunay triangulation is used to interpolate and form the triangles. This method satisfies the Delaunay criterion, which ensures that in the resulting network, no point lies within the interior of any circumcircles of the triangles. This maximizes the minimum interior angle of all triangles, which avoids that long thin triangles being formed as much as possible (ArcGIS, n.d.-b). Subsequently, the points associated with a circumcircle are connected to each other. This establishes the final network. Figure 2.4 shows an example of a triangulated network.

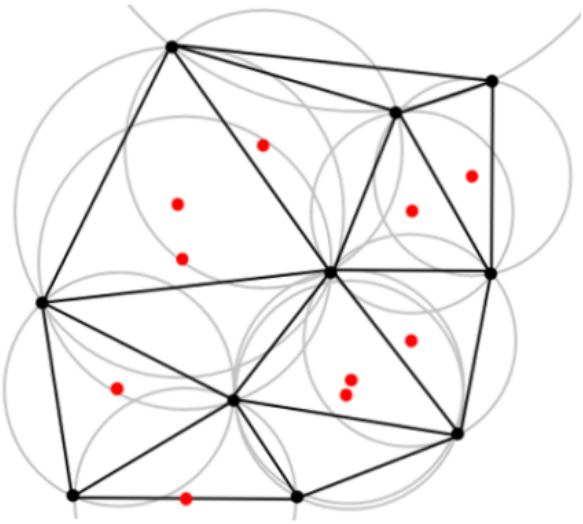


Figure 2.4: A triangular network created with TIN (QGIS, 2020). The black dots are the locations of known samples. Through these locations circumcircles are drawn in a way that no known location is in the interior of any circumcircle. Subsequently points associated to the same circumcircle are connected to each other.

*A categorical variable is based on a qualitative property rather than a quantitative property as is the case with a continuous variable. A categorical variable assigns each observation to a particular group or nominal category, in this case, soil types.

Pros and Cons

The simplicity of TIN is a big advantage. It is easy to explain and has low computational costs. On the other hand, simplicity is also a disadvantage. In a 1D world, it connects the soil layers with a linear line which is often not a good approximation as soil layers are often horizontal. Furthermore, TIN is a spatial interpolation technique for single values and not for categorical variables such as soil types. Figure 2.5 shows an example where it is not straightforward how to interpolate the soil types with TIN. This example has two possible ways to interpolate. The first way is to connect the mud layers. The other option is to connect the sand layers. TIN is the most simplistic model implemented and therefore deployed as a benchmark.

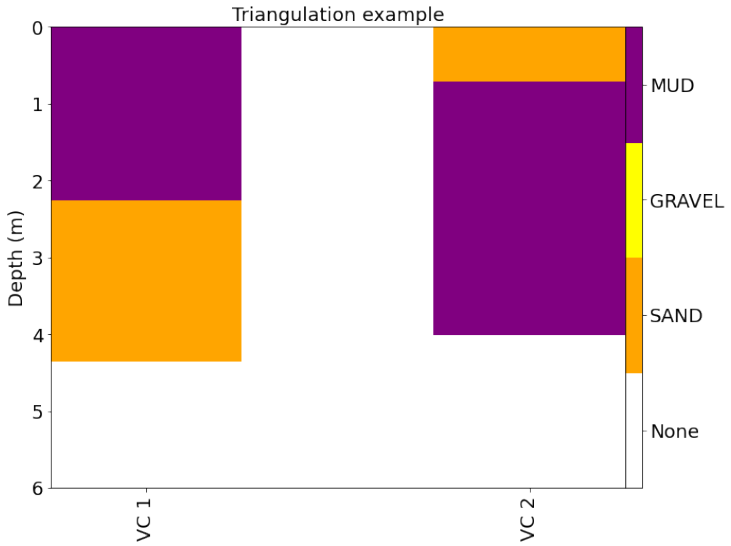


Figure 2.5: An example where it is not straightforward how to interpolate the soil types.

2.2.1.2 Kriging

One of the most commonly used spatial interpolation techniques is kriging. This method is similar to IDW, where closer points have more weight than points farther away. This fundamental concept is enshrined in Tobler's first law, mentioned in section 2.1.7, which forms the basis for IDW and kriging. Kriging obtains the average value at unknown locations by interpolating known points and uses a weighted average of the covariance function between them. The variable Z is assumed to be equal to the sum of a deterministic component $\mu(x)$ and a stochastic component $R(x)$ as shown in equation 2.4. The expected value of the stochastic component is 0, hence the expected value of variable $Z(x)$ is equal to the deterministic component. The value z_0 at an unknown location x_0 is estimated with equation 2.6.

$$Z(x) = \mu(x) + R(x) \tag{2.4}$$

$$E(R(x)) = 0 \quad \Longrightarrow \quad E(Z(x)) = \mu(x) \tag{2.5}$$

$$\hat{z}(x_0) = \sum_{i=1}^{n} \lambda_i z(x_i) \quad with \sum_{i=1}^{n} \lambda_i = 1 \tag{2.6}$$

where $z(x_i)$ are known values at location x_i; n is the sample size and λ_i are the optimized weights for x_i. In 1951 D. Krige described kriging for the first time in his Master's Thesis. Nowadays, kriging is a generic name adopted by geostatisticians for a family of generalized least-squares regression algorithms (Goovaerts, 1997). It is an extended family with many variations such as ordinary kriging, simple kriging, universal kriging, indicator kriging, probability kriging, disjunctive kriging, co-kriging, kriging with external drift and even more methods. All flavours of kriging share the same objective of minimizing the variance $\sigma^2(x)$ under the constraint of unbiasedness of the estimator (Goovaerts, 1997). The variance is defined by equation 2.7 which is minimized under the constraint in equation 2.8.

$$\sigma^2(x) = Var\big[\hat{Z}(x) - Z(x)\big] \tag{2.7}$$

$$E\big[\hat{Z}(x) - Z(x)\big] = 0 \tag{2.8}$$

Accordingly, it has a sound theoretical basis in the form of minimizing the Mean Squared Error (MSE) and is known as the Best Linear Unbiased Predictor (BLUP) for spatial data (Sekulić et al., 2020; Cressie, 1990). In other words, the predictor minimizes the MSE over all linear unbiased predictors. The name predictor is preferably used instead of estimator since it concerns unknown values.

This research uses a kriging model, which is available in ArcGIS Pro. ArcGIS Pro offers two options for kriging, namely ordinary kriging and universal kriging. Hereof ordinary kriging is the most general method. Ordinary kriging assumes that there is a constant mean which is unknown. On the other hand, universal kriging assumes there is a trend in the data and that it can be modelled by a deterministic function, defined as:

$$\mu(x) = \sum_{l=1}^{k} \beta_l x_l(x) \tag{2.9}$$

where β are the optimized coefficients; k is the number variables and x are the variables. With three variables, longitude x_1, latitude x_2 and depth x_3, the formula can be rewritten as:

$$\mu(x) = \beta_0 + \beta_1 x_1(x) + \beta_2 x_2(x) + \beta_3 x_3(x) \tag{2.10}$$

It is a shortcoming that ArcGIS Pro does not have the ability to pass additional variables to kriging. Consequently, kriging bases its prediction on spatial position only, without any other explanatory variables available in the data.

The trend from equation 2.10 is subtracted from the original data, and the autocorrelation is modelled from the random errors, $R(x)$. After fitting the model to the random errors and before making a prediction, the trend is added back to the predictions to produce the final

results. It is only useful to use universal kriging when it is known that there is a trend in the data (ArcGIS, n.d.-a). In this case, it is assumed that there is a trend in the data due to historical events such as glaciers in the ice age and sediment deposits over time.

Pros and Cons

When kriging was introduced in the 1980s, it was a significant improvement over existing interpolation techniques. The power of kriging arises from using the spatial correlation structure in the data. Furthermore, kriging minimizes the MSE under the constraint of unbiasedness, which makes it the BLUP. A wide variety of models makes it applicable to various data configurations. On the other hand, kriging also has disadvantages. It makes many assumptions, can be computationally demanding, and can be challenging to select the best model for the data. Additionally, kriging is not well designed for incorporating all covariate information which is available nowadays (Sekulić et al., 2020). On top of that, the available kriging methods in ArcGIS Pro do not provide this capability at all. Kriging is therefore based on the location only. Finally, kriging is a regression technique that minimizes the MSE and therefore is not well designed for a classification problem.

2.2.2 Machine & Deep Learning Models

ML models are adaptive models, which require a training set to 'train' the model. This training set contains input variables and the corresponding output, also known as labels. While training, the parameters of an adaptive model are tuned on the input variables with respect to their labels (Bishop, 2006). This process is called learning and optimizes an inferred function $y(x)$. Once the model is trained, $y(x)$ can take new input value(s) $x(i)$ and generate new predicted label(s). A test set is usually withheld of the data to assess the model's performance. Then, in the evaluation phase, the generated labels are compared to the 'groundtruth' labels.

Conventional interpolation techniques such as kriging, IDW and triangulation presumes that the predictions are a linear combination of the available data (Schloeder et al., 2001). One of the strengths of ML is that it is very flexible and not restricted to linear relations (Sekulić et al., 2020).

2.2.2.1 Random Forest

Decision Tree

A random forest is an extension of the decision tree classifier. It is a so-called "ensemble" learning technique, where multiple learning algorithms are combined to obtain a better performance. It grows a fixed number of decision trees on bootstrapped samples*. A positive aspect of a decision tree is its high interpretability. Thus, the structure of a decision tree consists of

*Bootstrapping is random sampling with replacement. Normally, the size is equal to the size of the original dataset.

internal nodes and leaves (James et al., 2017). The leaves are end nodes of the tree, representing a label. Prior to arriving at a leaf, decisions are made at internal nodes. Figure 2.6 presents an example of the structure of a decision tree.

An important characteristic of a decision tree is that it maximizes the "quality" of a split at every decision node. The quality is measured by a function which is either the information gain or the Gini Index. Information gain is based on entropy and information theory. The model considers every possible split and selects only the one with the highest quality. This is a greedy algorithm that is prone to local optima. It is greedy because it maximizes the quality at every split without taking into account future splits. Starting with the entire training set, a threshold is made for the most powerful feature, splitting the training set into two subsets. A subsequent threshold is made for both subsets, which maximizes the quality of a split for the associated subset. This process is repeated until each observation ends up in a different leaf or the process is early stopped by a regularizer[†]. A visualization of this process is presented in Appendix B. In case the tree is not regularized, the tree will very likely overfit[‡] the training set due to a tree that is too complex for new data or a test set. Using a regularizer prevents the model from developing a fully grown tree, forcing it to be more general. An example of a regularizer is a maximum number of decisions or a minimum number of samples on a node to split further.

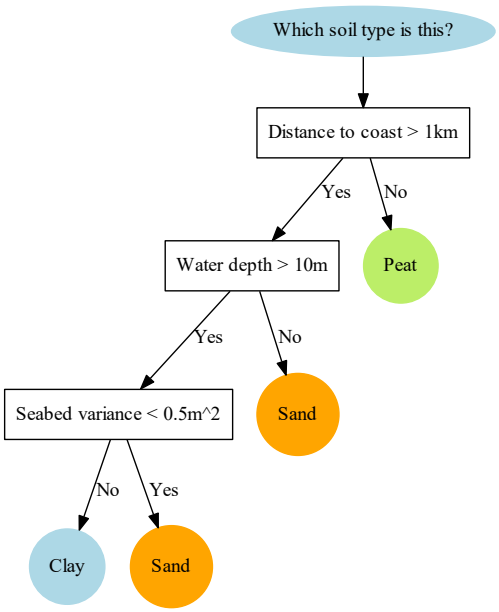


Figure 2.6: An example of the structure of a decision tree. At the top, the classification task is. The decisions are made in the squares which are the nodes of the tree. The leaves of the tree are the end nodes of the tree which represent a soil type.

[†]Regularization is a process that helps to prevent overfitting, which is performed by a regularizer.

[‡]Overfitting is training a model that is too specified on the training data and therefore too specific for new data.

Randomness

The word random in the classifiers name comes from the random feature selection at each decision node. This tweak helps the trees to decorrelate (James et al., 2017). This is very effective when there are one or more dominant features. For instance, when every tree is allowed to choose from all available features to minimize the objective function, most or all trees will use the strong predictor in the first split depending on the bootstrapped training sample. Resulting in very similar trees and highly correlated predictions. Averaging highly correlated predictions does not lead to a considerable variance reduction. Using random feature selection gives other features more chance, and therefore more diverse trees will be grown. Resulting in a more stable and reliable random forest. In classification, it is standard that $\sqrt{p}$ features are available at each node when the total number of features is p.

Bagging

A random forest grows a multitude of these trees, a forest. Each tree is built on a different bootstrapped training sample. As a consequence, diverse trees are constructed, which are averaged in the end. This procedure is called *Bagging* (Bagging=**B**ootstrap **Agg**regat**ing**). Bagging reduces the variance of a statistical learning method like a random forest (van Giersbergen, 2018). For each tree, a bootstrapped sample is drawn from the entire training set. Each sample is constructed by drawing with replacement from the entire training set. Therefore the probability that an observation is not in the bootstrapped sample is equal to $\left(1-\frac{1}{n}\right)^n \approx e^{-1} \approx \frac{1}{2.72} \approx 37\%$. This also implies that each bagged tree is trained on approximately $\frac{2}{3}$ of the observations. The remaining are out-of-bag (OOB) observations which can be used as test observations.

When a random forest predicts a new input sample, the model follows for every tree the corresponding path. Ending with the same number of predictions as the number of trees constructed. The final prediction is the majority vote of all trees. Figure 2.7 shows the architecture of a random forest. Due to the random nature of each bootstrapped sample, it can be seen as an independent set. Hence, each prediction of a tree is independent too. Averaging a set of independent predictions reduces the variance. Given a set of n independent predictions $z_1, ..., z_n$ each with variance σ^2, the variance of the mean $\bar{z}$ of the predictions is given by $\frac{\sigma^2}{n}$. Meaning that the predictions of a random forest converges with an increasing number of trees and does not overfit. Due to averaging independent predictions, there is no need to do cross-validation for a random forest.

Pros and Cons

In general, random forests are robust against overfitting and are not sensitive to excessively noisy data (Wyk et al., 2018). Due to the hierarchical structure of decision trees, a random forest is less prone to data imbalance. Another benefit of a random forest is that it is capable of providing an estimate of the feature importances because each split is based on one feature,

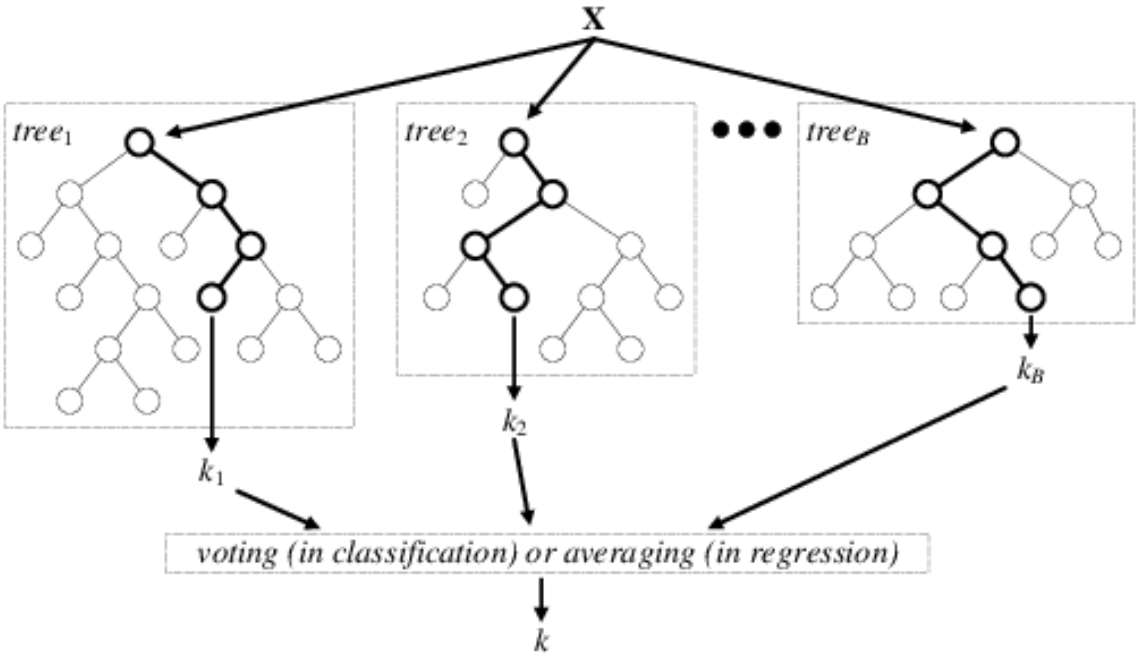


Figure 2.7: An example of an architecture of a trained random forest with B trees (Verikas et al., 2016). If the sample x is predicted, it follows for each tree the corresponding path resulting in a prediction k_i for tree$_i$. The final prediction k is the majority vote of all trees.

which maximizes the quality of a split. Averaging the quality per feature gives an indication of which feature has the most predictive power. Furthermore, a random forest is a non-linear algorithm that is not restricted to linear relations like kriging. Kirkwood et al. (2016) concluded that a random forest was capable of entirely capturing the spatial autocorrelation of the target variable. The random forest in their article also obtained more accurate results than ordinary kriging. The downsides of a random forest are low interpretability and high computationally cost, especially when many trees with large depth[§] are grown. Both do not influence the results, and therefore it is expected to outperform kriging.

2.2.2.2 XGBoost

E**X**treme **G**radient **Boost**ing (XGBoost) is another ensemble machine learning technique and somewhat similar to a random forest. In contrast to a random forest, XGBoost grows trees sequentially instead of independent. Each subsequent grown tree attempts to minimize the classification error of the previously constructed trees. Incorrect predicted samples are given a higher weight and force the new trees to focus on those hard-to-learn samples (Taghizadeh et al., 2020). Figure 2.8 presents the architecture of XGBoost. The objective function that the algorithm minimizes is expressed as:

[§]The depth of a decision tree is equal to the number of layers. For example, the decision tree in Figure 2.6 has a depth of three.

$$obj(t) = \sum_{i=1}^{n} L(y_i, \widehat{y_i}^t) + \Omega(f_t) \tag{2.11}$$

where L is the loss function; Ω is the regularization term for quantifying the model complexity; t is the t^{th} iteration for generating the t^{th} tree; n is the total number of observations; f_i is the tree structure of i^{th} tree. The model complexity is modelled as follows:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{2.12}$$

where γ and λ are penalty terms for regularization to avoid overfitting; T is the total number of leaves in each tree and w is the output of a leaf. When t^{th} tree is created to fit the residual error of the former tree, the prediction of the new tree can be expressed as:

$$\widehat{y_i}^t = \widehat{y_i}^{t-1} + f_t(x_i) \tag{2.13}$$

Substituting this into the objective function:

$$obj(t) = \sum_{i=1}^{n} L(y_i, \widehat{y_i}^{t-1} + f_t(x_i)) + \Omega(f_t) \tag{2.14}$$

Expanding f_t with Taylor polynomial, the approximation of objective function with second-order accuracy can be expressed as:

$$obj(t) = \sum_{i=1}^{n} \left[L(y_i, \widehat{y_i}^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{2.15}$$

in which g_i denotes the first derivative:

$$g_i = \frac{\partial L(y_i, \widehat{y_i}^{t-1})}{\partial \widehat{y_i}^{t-1}} \tag{2.16}$$

and h_i denotes the second-order derivative as:

$$h_i = \frac{\partial^2 L(y_i, \widehat{y_i}^{t-1})}{\partial \left(\widehat{y_i}^{t-1} \right)^2} \tag{2.17}$$

The value of the objective function depends only on g_i and h_i, allowing for a customized loss function based on the residual errors of former trees. Appendix C shows an example how XGBoost minimizes MSE as a loss function. The MSE has a friendly derivative with a first and second-order term and does not need a Taylor polynomial. Finally, the derivative of the loss function can be set equal to zero to find the minimum.

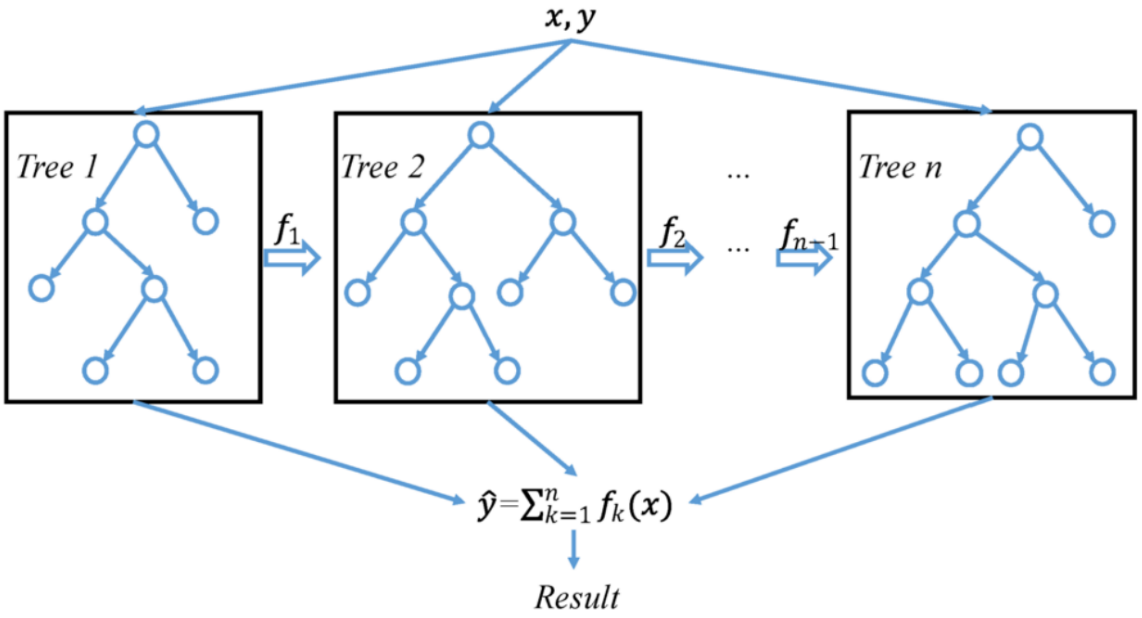


Figure 2.8: An example of an architecture of a trained XGBoost model with n trees (Y. Wang, Pan, et al., 2019). The algorithm starts with growing the 1^{st} tree which passes its tree structure f_1 on to the next tree. Then, the algorithm constructs the 2^{nd} tree on the classification error of 1^{st} tree. The 2^{nd} tree passes its tree structure on to the next tree which is constructed on the classification error of 2^{nd} tree. This process is repeated until n trees are grown. A sample x is predicted by following the paths for all trees and adding up all predictions.

Pros and Cons

XGBoost is a high-performance model, and its impact is widely recognized in machine learning. For example, Kaggle is an online community of data scientists and machine learning practitioners. Kaggle hosts machine learning competitions in which thousands of people participate. In 2019, XGBoost won 17 out of 29 Kaggle competitions.

Not surprisingly, XGBoost has some similar characteristics as a random forest due to their related structure. That is why it is also less sensitive to data imbalance and can provide an estimate of feature importance. XGBoost can handle imbalanced data even better as it gives more weight to misclassified samples, thereby increasing its ability to predict the minority class. Logically, there are also elements that are different, such as that XGBoost can overfit the training data by growing too many trees. Another characteristic is that XGBoost has much fewer hyperparameters to tune, which can be an advantage and a disadvantage.

2.2.2.3 Long Short-Term Memory Network

Originally, LSTM network is a time series model, but today it is used in, among other things, sequence prediction problems such as natural language processing. In the latter case, which is an emerging field of research, it is often used to predict the next word in a sentence. Although, most

of its fame it acquired with predicting time series. This research is about spatial dependence instead of temporal dependence. In both cases, there is a correlation between (near) data points. In the same way that Tobler's first law applies to spatial correlation, it also applies to time series. This means that closer timestamps are more related than timestamps that are further apart, which seems intuitively plausible. For the reason that the problems are related in this way, the author proposes LSTM network for the first time in a spatial context. The next paragraphs provide an introduction to the LSTM network.

Recurrent Neural Network

LSTM network introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997 is a special kind of a Recurrent Neural Network (RNN). And a RNN is on its turn a special kind of an Artificial Neural Network (ANN). RNNs are capable of storing previous inputs in a "memory" to persist in the network internal state. This is accomplished by a loop in the cell. While an ordinary ANN neuron performs a transformation on the input x_t and passes the result h_t on to the next neuron. Does a RNN neuron make a loop where it passes the result on to itself, "remembering" while considering the next observation (x_{t+1}). Figure 2.9 shows a schematic representation of a loop in a LSTM cell. The three green blocks indicate the same cell that passes information on to itself by the horizontal black lines while considering the next observation. With this ability, a RNN is capable of computing output based on the entire history of preceding inputs (Graves, 2012). Theoretically, a RNN can learn an entire sequence, but in practice, this is rather limited. Given that the input either increases or decreases, the output leads to a vanishing or exploding gradient when it cycles around the recurring network connections. This is addressed in literature as the "vanishing gradient problem" by Hochreiter et al. (2001). RRNs embodies "short-term memory", characterized by fast changing weights and cannot deal with slowly changing weights. "Long-term memory", characterized by slowly changing weights, is potentially essential in many applications, such as in time series and spatial data.

LSTM Cell

The remedy for this problem is a "long short-term memory" network. This recurrent network is efficient in remembering over 1000 steps without loss of short-term capabilities (Hochreiter & Schmidhuber, 1997). Four interacting layers in the structure of a LSTM cell makes it very efficient in long-term problems. The essence of LSTM is the ability to add and/or remove information to/from the cell state. This "updating" or "forgetting" is regularized by gates. Each LSTM cell contains in total three gates, namely a forget gate f_t, an update gate i_t and an output gate o_t. These gates can take values between 0 and 1. Based on the new input x_t and the previous output h_{t-1} a sigmoid layer decides whether the cell state C_{t-1} is totally "forgotten" ($f_t = 0$), totally "remembered" ($f_t = 1$) or anything in between ($0 < f_t < 1$). This can be written as:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.18}$$

where σ is a sigmoid layer, which can be seen as a logistic regression function that outputs a value between 0 and 1. The input arguments are the new input x_t, the previous output h_{t-1}, the weight matrices W_f and U_f and a bias vector b_f. The weight matrices and the bias vector are learned during training. Next, another sigmoid layer decides what to update as well based on x_t and h_{t-1} and outputs a value between 0 and 1.

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.19}$$

Then a hyperbolic tangent (tanh) layer constructs potential new values, $\widetilde{C}_t$, that could be added to C_{t-1}.

$$\widetilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{2.20}$$

$\widetilde{C}_t$ is multiplied with the output of the update gate, i_t, what is added to C_{t-1} after multiplication with f_t. The updated cell state is the cell state C_t for the next iteration. This is captured in the following formula:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \widetilde{C}_t \tag{2.21}$$

where $\otimes$ is the Hadamard product, which means element-wise multiplication. The last and third sigmoid layer, again based on x_t and h_{t-1}, decides which parts of the cell state is going to be the output, o_t.

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.22}$$

Then C_t is put into a tanh function to push the values between 0 and 1 and multiplied by o_t to output only the particular parts that was decided in the previous step. The output becomes h_t what is considered in the next iteration together with the new input x_{t+1}, starting the whole procedure from the beginning.

$$h_t = o_t \otimes \tanh(C_t) \tag{2.23}$$

For each time step the entire process is completed and iteration stops when all time steps have been processed. Figure 2.9 offers an overview of a LSTM cell with three iterations.

LSTM Network

Now that the theory of an LSTM cell has been explained, an LSTM network can be clarified. The architecture of an LSTM network consists of layers and a certain number of neurons per layer. This study only uses LSTM layers without any combination with other layers except a dropout layer and a fully-connected layer. The fully-connected layer is used for classification

to push the output between 0 and 1. For regularization, a dropout layer is used. This reduces overfitting by ignoring neurons in a layer and forces the network to use other connections every epoch. An ordinary LSTM layer works only in forward direction and can therefore only learn from the past. A bidirectional LSTM layer has the advantage that it also can learn from the current information.

Pros and Cons

An LSTM network is very flexible and has the big advantage of remembering historical data. Only, it needs a lot of data to train. Combined with the high class imbalance, this can lead to problems as there is a shortage of minority classes. Besides, the computational costs are very high and the interpretability very low. It is a so-called "black box" model where it is almost impossible to determine how the output is generated.

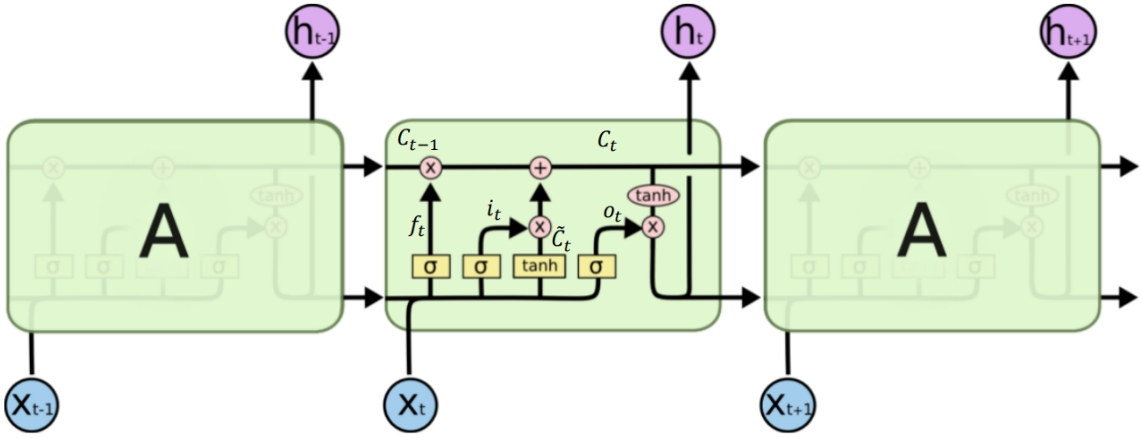


Figure 2.9: Schematic representation of one LSTM cell (Palagi et al., 2018). Three green blocks that each indicates an iteration of the loop in a LSTM cell. The left block considers the previous input, the middle block the current input and the right block the next input, i.e., X_{t-1}, X_t and X_{t+1} respectively. After X_{t-1} has been treated, the cell passes its output, h_{t-1}, onto itself, i.e. the middle block. The middle block shows the applied transformations to the previous output plus the current input, i.e. h_{t-1} and X_t respectively. Information is transferred by the black lines, the yellow blocks represent a neural network layer and the light pink circles indicate operations between the intersecting information.

2.3 Class Imbalance

An imbalanced dataset is characterized by a disproportionate ratio between classes. When machine learning algorithms encounter an imbalanced dataset, they tend to have a bias towards the majority class. Because all data is treated equally and therefore the misclassifications too. False accuracy estimates, misclassification, or complete ignoring minority classes are often the consequence. In machine learning, this is known as the class imbalance problem (Taghizadeh et

al., 2020). In soil classification problems, this phenomenon can cause some minority soil types to get omitted in the results. This is undesired when the minority soil types are crucial and can lead to unreliable soil maps (Sharififar et al., 2019). In this study, peat is a minority class that is of great importance because of its low thermal conductivity. For that reason, all peat had to be excavated from the seabed.

Imbalanced datasets are a known issue in soil classification where often a highly skewed class distribution is faced (Taghizadeh et al., 2020). While imbalanced classification is a recognized problem in the machine learning discipline for categorical data modelling, this issue has not been well addressed in soil mapping (Sharififar et al., 2019).

Machine learning literature offers many methods to overcome this challenge, from oversampling and undersampling techniques to class weights to specially designed metrics and deep learning sample generators. Oversampling tackles the imbalance by generating new samples of the minority classes and undersampling does this by removing samples from the majority classes. To compensate for an imbalance that is present in the data, both oversampling and undersampling involve introducing a bias to select more samples from one class than from another. The introduction of a bias is inevitable, and this is at the expense of the performance of the majority classes.

Oversampling can be done by randomly duplicating existing samples of the minority class or by a more popular method Synthetic Minority Oversampling Technique (SMOTE). Randomly duplicating the majority class does not contribute to improve the classifier's performance, while SMOTE does (Chawla et al., 2002). Randomly undersampling, randomly deletes samples of the majority class, which unavoidably lead to loss of information. Despite the loss of information, randomly undersampling leads to a better discriminating ability of the minority classes according to Chawla et al. (2002). Furthermore, SMOTE was shown superior to other oversampling methods in classifying soil types by Taghizadeh et al. (2020). For these reasons, a combination of SMOTE and randomly undersampling was adopted in this thesis. The next paragraph elaborates on the theory of SMOTE.

Synthetic Minority Oversamplig Technique

SMOTE generates synthetic minority observations between existing observations by differing one or more features of the existing observations. The method considers the five nearest neighbours of each minority sample and selects randomly n samples from the nearest neighbours, depending on the amount of oversampling. When the amount of oversampling is 100%, 200% or 300%, n is 1, 2 or 3, respectively. Subsequently, a random number between 0 and 1 is generated for each selected nearest neighbour, which is multiplied with the difference in feature vector, i.e. the distance between the samples, and added to feature values of the sample under consideration. This results in a synthetic minority sample at a random point along the line segment between two minority samples. The decision region of the minority classes is effectively forced to become more general (Chawla et al., 2002). Figure 2.10 visualizes an example of SMOTE

with two features and an oversampling rate of 100%. Therefore only one nearest neighbour is selected.

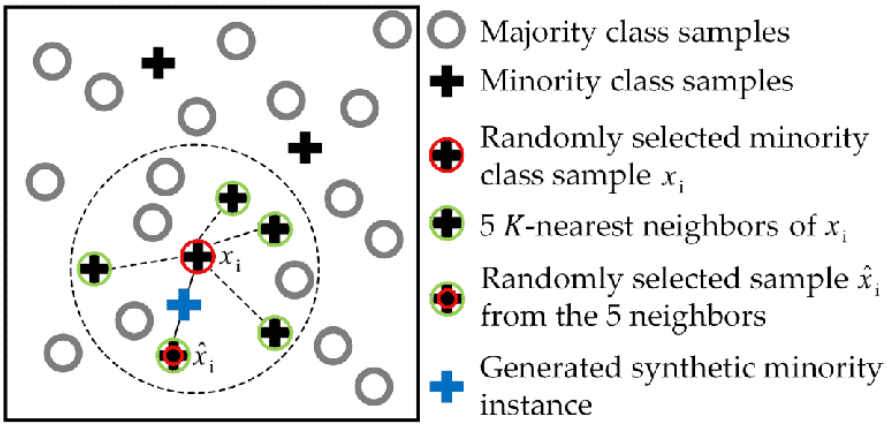


Figure 2.10: A visualization of SMOTE. A randomly selected minority sample (red outer circle) selects random a neighbor (green outercircle and red innercircle) from the 5 nearest neighbors (green outercircle). Then at a random distance between the two selected samples a synthetic minority sample is created (blue cross) (Kunert, 2020).

2.4 Potential Features

Machine learning algorithms can have a very good performance on specific problems based on a lot of data, but without good data quality, performance is also poor. The data quality for a model is and remains the most critical part. This still applies to state-of-the-art models. This means that the quality of the features for a machine learning model is of utmost importance. These features comprise all the information the model gets to "know". Based on this information, it tries to learn specific patterns and make predictions on new information. When the features do not contain valuable information, the algorithms will also be useless. Therefore it is essential to thoroughly investigate which features are available and which features have predictive power.

For example, for soil type prediction, the genesis of the soil is relevant. Geological events, such as sediment deposits, earthquakes and glacial periods, can contain a lot of valuable information about the current soil composition. McBratney et al. (2003) identified 7 factors for soil spatial prediction:

S: Soil, other properties at location of prediction;

C: Climate, climate properties at location of prediction;

O: Organisms, vegetation, fauna or human activity;

R: Relief or topography, landscape attributes;

P: Parent material, lithology;

A: Age, time factor;

N: Space, spatial position.

These are the so-called SCORPAN factors. The features that are used in this study are based on these factors. By substituting the SCORPAN factors into the general equation 2.1 results in equation 2.24. This formula can be seen as a general function used by the implemented models. The ML models base their predictions on the input variables but use a different function f for the prediction.

$$S = f(s, c, o, r, p, a, n) + e \tag{2.24}$$

Soil, other properties at location of prediction

For the first variable, soil or other properties at location, the soil types and other soil properties of the nearest VCs and CPTs are included. The soil type at the location of prediction is, of course, not known, but some soil properties at this location are known and identified by the geophysical survey. The soil layers that reflect the signals from the equipment and are then captured reveal certain soil properties of these layers.

Climate, climate properties at location of prediction

Climate properties at the location of prediction are not included. Obviously, in this case, it is about the climate subsea. The available data do not provide information about the climate subsea. Although it is plausible that there are differences in the area, such as current velocity.

Organisms, vegetation, fauna or human activity

Organisms, vegetation, fauna or human activity, are not present in the data and therefore not included. Nevertheless, it is known that dredging, which is part of human activities, is done close to the fairway. However, the exact locations of the dredging, when it took place, and the amount is unknown.

Relief or topography, landscape attributes

Relief or topography, landscape attributes are included. The variance of the seafloor height and the relative height of the seafloor to the surroundings have been added to the features.

Parent material, lithology

Parent material, lithology, are included through the beginning and the end of the layers of the nearest GTs at equal vertical positions beneath the seafloor. In addition, the detected soil layering by the geophysical survey at the location of prediction is included.

Age, time factor

Age, time factor is not included. McBratney et al. (2003) indicates, as an example, that an estimate of the age of the soil surface may be useful. This is unknown and will probably not differ much in this project area. It is essential that a feature varies between the locations in the area, otherwise no distinction can be made and the feature is useless.

Space, spatial position

Space, spatial position is included. Intuitively, the location of a sample contains valuable information about its soil type. Therefore easting, northing and the depth beneath the seafloor of the sample are included as features. Besides, water depth and the distance to coast also give an indication of the spatial position and are included too. According to Li et al. (2011) it is likely that distance to coast has some influence on the transportation of sediment from onshore sources and can therefore also be related to climate properties.

Derived Features

The derived features are listed in Table 2.3. The first column indicates from which dataset the feature is derived. Herein, the spatial position is unique as these features can be relocated anywhere in the area. This is essential because predictions have to be made everywhere in the area to develop a 3D model. After determining the spatial position, all other features are computed for this position. For training and testing, it is required that the output is known. Therefore, during training and testing, the spatial position is bounded to the locations of the GTs because at these locations, the output, i.e. soil type, is known. The second column lists the derived features, and the third the associated unit. The features are scaled before training, therefore the units do not actually matter.

2.5 Literature Review Conclusion

DSM is a challenging task that involves a lot of uncertainty (Phoon et al., 2021; H. Wang et al., 2020; Samui & Thallak, 2010). Unfortunately, not much literature has been written about predicting or interpolating soil types, making it a relatively undiscovered field of research. The vast majority of current literature deals with predicting soil properties rather than soil types. However, this is a much simpler task since one value has to be predicted for each location instead of a soil type. Simply put, a soil property can be measured for each sample in the area and therefore can be directly interpolated even if the soil does not have this property, i.e. with zero values. In soil type prediction, a model has to deal with a classification task of categorical variables. The conventional modes, TIN and kriging, are designed to deal with single values and not with categorical variables.

Table 2.3: Derived Features

Source data	Feature	Unit
Spatial position	Easting	m
	Northing	m
	Depth beneath seafloor	m
	Distance to coast	m
Nearest VCs	Distance to VC	m
	Primary soil type at equal vertical position	-
	Secondary soil type at equal vertical position	-
	Start of that layer below seabed	m
	End of that layer below seabed	m
	Water depth	m
Nearest CPTs	Distance to CPT	m
	Primary soil type at equal vertical position	-
	Tip resistance at equal vertical position	MPa
	Sleeve friction at equal vertical position	MPa
	Pore water pressure at equal vertical position	MPa
	Water depth	m
Nearest VCs & CPTs	Vertical SoF	-
	Horizontal SoF	-
Sub-bottom profilers	Detected soil type at equal vertical position	-
	Start of that layer below seabed	m
	End of that layer below seabed	m
	Detected soil type above that layer	-
	Detected soil type beneath that layer	-
Bathymetry	Water depth	m
	Water depth variance	m^2
	Quantile of water depth	-

Concluding that the conventional models are not well suited for classifying soil types. Therefore, the broad applicability of machine learning offers potential for improvement in this field. The articles of Sekulić et al. (2020); Kirkwood et al. (2016); Taghizadeh et al. (2020) have tested this hypothesis and show a good performance of machine learning in predicting soil properties. This confirms that machine learning can be used for spatial problems and makes it promising for soil type classification. Machine learning models are ideally suited for predicting categorical variables due to their ability to predict probabilities for each soil type. These probabilities add up to 1, and the final prediction is simply the type with the highest probability. Another major benefit is that the probability also gives an indication of how confident the model is. This can be very useful in practice if the model can indicate where the uncertainties are. In this way,

measures can then be taken to determine whether or not to obtain more certainty about this area.

Despite the differences between soil property and soil type prediction, both problems have many similarities. For example, spatial correlation plays a crucial role in both problems, which can be exploited by the field of geostatistics. Moreover, with both problems, it is fundamental to acquire as much information about the area as possible. Therefore, the literature on soil property prediction has been involved in determining which features can be used and how to include the spatial correlation.

One of the main topics of this study is the introduction of the LSTM network in DSM, which has not been used before in this field. Based on the theory of the LSTM network, the model suits the problem well. Especially because of its capability to process sequential data and the presence of spatial correlation in the problem.

Another shortcoming of the applied conventional models is that they do not utilize all available information present in the data. The results are only based on how soil layers develop in space. Meanwhile, the applied state-of-the-art models utilize all available information provided. The challenging task here is that the implementer extracts as much information as possible from the data, preprocesses it as well as possible, and presents it to the model in the best way. If these steps are done properly, it is credible that the ML models outperform the conventional models. Since Kirkwood et al. (2016) stated that a random forest can fully capture the spatial correlation, it can at least be assumed that ML with more information performs at least as well as the conventional models.

Chapter 3

Data Description

This chapter provides a decription of the used data. The first section describes the available data. The next section discusses which features are used and how they are extracted. This is followed by the correlation between these features. This chapter finalizes with a section about soil interpretation.

3.1 Available Data

The data made available for this study consists of geotechnical and geophysical data. The geotechnical part is conducted by two companies and consists of CPTs and VCs. In total, 757 VCs were collected along the cable corridor and 718 CPTs. The total length of all GTs together is 5,435.31 meters. The soil type distribution of the GTs is presented in Figure 3.1. Additionally, on some of these VCs, laboratory tests were performed to acquire more knowledge about the soil behaviour and its characteristics. In Appendix D two PDFs are presented that Boskalis received from one of the companies. The location and other sensitive data have been made unrecognizable for confidentiality reasons. The data is received in PDF format and is therefore not immediately ready for use.

The geophysical data was provided by a third company and acquired using bathymetric scans, side-scan sonar, magnetometer scans and sub-bottom profilers, and contains every 2 meters along each cable corridors a record. A total of 131,513 depth measurements and 110,935 records of the sub-bottom profilers are provided. Finally, the received data is enriched with public data including coordinates of the German coastline. The data is provided by European Environment Agency (*EEA coastline for analysis*, 2015).

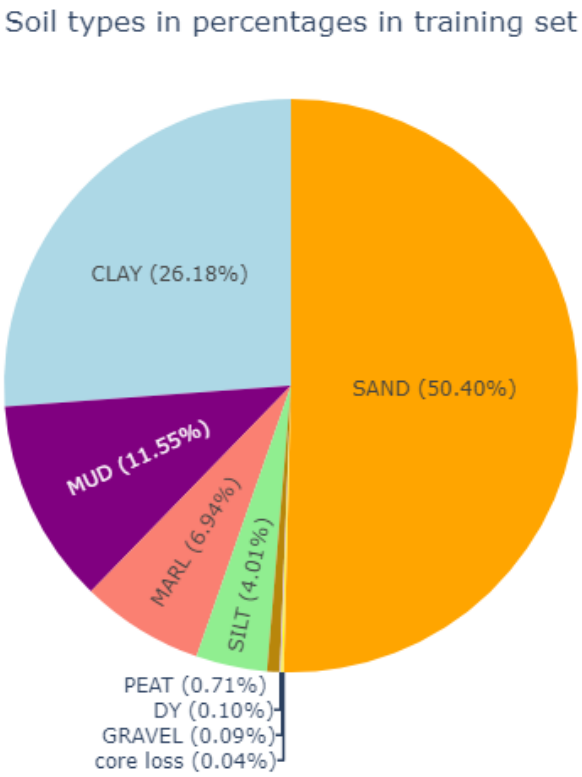


Figure 3.1: Soil types in percentages in training data.

Vibro Core

A VC is a soil sample of a few meters in length and a diameter of approximately 10 centimeters. The sample is collected at the vessel and sent to the laboratory for detailed logging and testing. On some of the VCs, additional tests were performed 1) Grain Size Distribution; 2) Natural Moisture Content; 3) Organic Content; 4) Atterberg-Limits; 5) Undrained Shear Strength; 5) Chemical Analysis and 6) Thermal Resistivity. Figure A.3 shows a typical PDF page of available VC data.

Cone Penetration Tests

A CPT is a well-known method to determine the geotechnical properties of soils. When performing a test, an instrumented cone is hydraulically pushed into the soil at a constant speed, in this case, 2 cm/s. While pushing the cone into the soil, several geotechnical soil properties are measured. The most important one is the total cone resistance (q_t) which is measured in MPa. The sleeve friction (f_s) and pore water pressure (u_2) are also measured in MPa. The friction ratio and the pore water pressure ratio are computed by dividing f_s and u_2 by q_t, respectively. Figure A.4 shows a typical PDF page of the available CPT data. It contains graphs of the five properties discussed above with a resolution of 2 centimeters.

Geophysical Survey

The geophysical survey is performed using bathymetric scans, side-scan sonar, magnetometer scans and sub-bottom profilers. On the basis of these surveys, the seabed profile along the three cable corridors is determined. These techniques can also penetrate the seafloor and detect soil layers beneath it. Although, soil layers were not detected everywhere.

3.2 Feature Engineering

This study aims to develop a 3D soil model along the export cable corridors of the wind farms. The ML models require features for predictions and building a soil model. The next paragraphs clarify how the features from Table 2.3 are derived.

Spatial Position

For the final 3D model, the locations are selected by taking an adequate distance between the predicted locations based on the soil variability. The locations along the cable corridors can be obtained from the bathymetry data. For training and testing, the locations are restricted to the locations of the GTs because a known output is required. These locations are obtained from the test itself, but the water depth is retrieved from the nearest bathymetry datapoint. Obviously, the locations along the corridor do not have water depth available from a GT. Therefore, the water depth for a VC and CPT is also retrieved from the bathymetry data. In this way, the spatial position is determined consistently because easting and northing are the exact* locations from the data and water depth from the geophysical survey. Subsequently, at each easting and northing position, every centimeter below the seafloor to a depth of six meters is predicted. Resulting in 600 samples at each easting and northing location. Finally, the distance from the coast is calculated as the geodesic distance, i.e. the shortest path between two points on a curved surface.

Vibro Core Features

A soil sample is expected to be correlated with nearby VCs. Therefore characteristics from the four nearest VCs are included as input variables. The distance to each of those VCs is computed as the geodesic distance. Additionally, the primary and secondary soil type at equal depth, the start of that layer and the end of that layer are all retrieved from the VC PDF. Finally, the water depth is retrieved from the bathymetry data as stated in the previous paragraph.

Cone Penetration Test Features

For consistency, the features of the CPTs have also been determined for the four nearest CPTs. Similar to the distance of the VCs, the distance to each of those CPTs is computed as the

*Exact here means that it is the exact coordinates from the data. The water depth of the GTs is not exact as they are retrieved from the bathymetry data.

geodesic distance. The companies that conducted the soil investigation also provided a soil type interpretation of the CPTs. This interpretation is used for the primary soil type at equal vertical position of the sample. Additionally, the tip resistance, sleeve friction and pore water pressure are included too. These three values are calculated as the statistical medians from 10cm above to 10cm below the depth of the sample. In this case, the median is preferred because the mean is sensitive to outliers*, especially with only a few observations in this range. Again, the water depth is retrieved from the bathymetry data.

Nearest VCs and CPTs Features

The SoFs are determined based on both VCs and CPTs in a radius of 500 meters of the sample. The vertical SoF is calculated by taking the average number of detected layers of all GTs in this radius. The horizontal SoF is calculated by determining how many different primary soil types are detected at the same vertical position beneath the seafloor in this radius.

Sub-bottom Profilers Features

Sub-bottom profilers transmit signals and capture reflections from soil layers. In this way, they are able to reveal soil properties at the location of the sample. A great benefit of a geophysical survey is that it collects data on every location along the corridor. The company that conducted the survey also provided a soil type interpretation. This is used to include the soil type at equal vertical position, the soil type above and beneath this layer. Additionally, the boundaries of the layer in which the sample is located are also included.

Bathymetry

The water depth variance of a location is calculated over all geophysical records within a radius of 50 meters. Each cable corridor has every 2 meters a data record of the depth, but not all GTs are performed perfectly on the route of the cable corridor. Fo this reason, is opted for a radius of 50 meters because then there are at least 20 records for each sample. When a larger area is chosen, it contains less information about the location in question. In addition, the values will differ less from each other, reducing the power of discrimination for this feature. For instance, if the water depth variance is taken for the entire area, there will be one value for all samples, which is useless as the model cannot distinguish between different samples. The water depth variance in a specific area contains information about whether or not the seabed is even or uneven. It is expected that this is correlated to the soil types or the variance of the soil types. For instance, one could think of sand dunes where the water depth variance would be high. Subsequently, the quantile of the water depth is calculated. This gives an indication of whether the seabed is low or high with respect to the surrounding area. For consistency, a

*An outlier is an observation that does not seem to fit with the rest of the data. This often occurs in CPTs due to measurement errors, resulting in 0 values.

radius of 50 meters has also been used here. A quantile value of 1 corresponds to the deepest point in the area, while a value of 0 corresponds to the most elevated point.

3.3 Feature Correlation

It is vital that the explanatory variables are correlated with the variable in question. If this is not the case, the model has to guess randomly, or it finds patterns that happen to be there, which in all probability will not be present in new data. On the contrary, it is favourable that the explanatory variables do not correlate with each other. It is evident that highly correlated features contain largely the same information. This directly means that the variables together do not have much more predictive power than just one of the variables. Therefore, it is desirable that the correlation between variables is close to zero.

Appendix E presents the correlation between the features and the actual soil type (y (nom)) in a heatmap[†]. All continuous features have (con) after their names, and all categorical features have (nom), from nominal, after their names. Since the continuous features are not Gaussian distributed, Spearman's correlation is used for the correlation between continuous variables. For the correlation between categorical variables and continuous variables, the correlation ratio is used, and the correlation between two categorical variables is computed with Cramer's V. The hue of the squares indicates the magnitude of the correlation, and the colour indicates whether the correlation is negative (blue) or positive (red). Not surprisingly, the features that define the position, i.e. Easting, Northing, Water depth, exhibit high correlation with each other and with inferred features from the position, such as Distance to coast. It can be noticed that the water depth features also show a high correlation.

Figure 3.2 shows the 25 most correlated features with the soil type. The categorical features (nom) and continuous features (con) cannot be compared one to one because they are calculated with a different association measure. Nevertheless, continuous and categorical variables can be mutually compared. Therefore, this graph gives an indication of which characteristic is most correlated with the soil type. Since soil type is a categorical variable, all correlations are positive. The features that determine the spatial position and the water depths are highly correlated and correlated to the soil type as well. It is expected that the features that are correlated most to the soil type are also the most important features in the tree-based models.

3.4 Soil Type Interpretation

Although there are quantitative, measurable characteristics to classify soil, as described in Subsection 2.1.3, there is still room for interpretation. Because not every soil sample is brought to the laboratory for examination and GTs do not provide all information to classify the soil samples. Due to the room for interpretation and the fact that the data is provided by two different companies, it occasionally occurs that there is a discrepancy in soil type interpretation

[†]A heatmap is a graphical representation of data that uses colour-coding to represent different values.

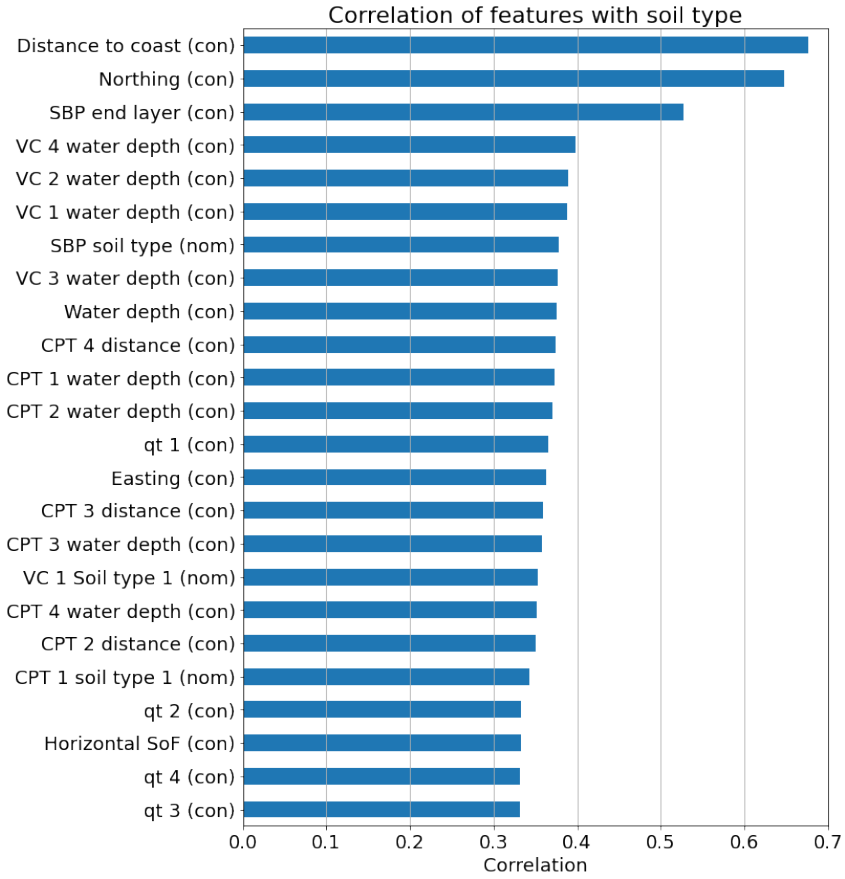


Figure 3.2: The correlation of the features with the soil type of the sample.

in the same area. Figure 3.3 shows a section of one of these areas along a cable corridor. Here, it is apparent that the GTs of company 1 are largely classified as mud while the GTs of company 2 contain multiple different layers. Besides, a difference in soil interpretations can also occur between different test types, which is shown in Figure 3.4. In this part of the corridor, the VCs are interpreted as mud and the CPTs as clay.

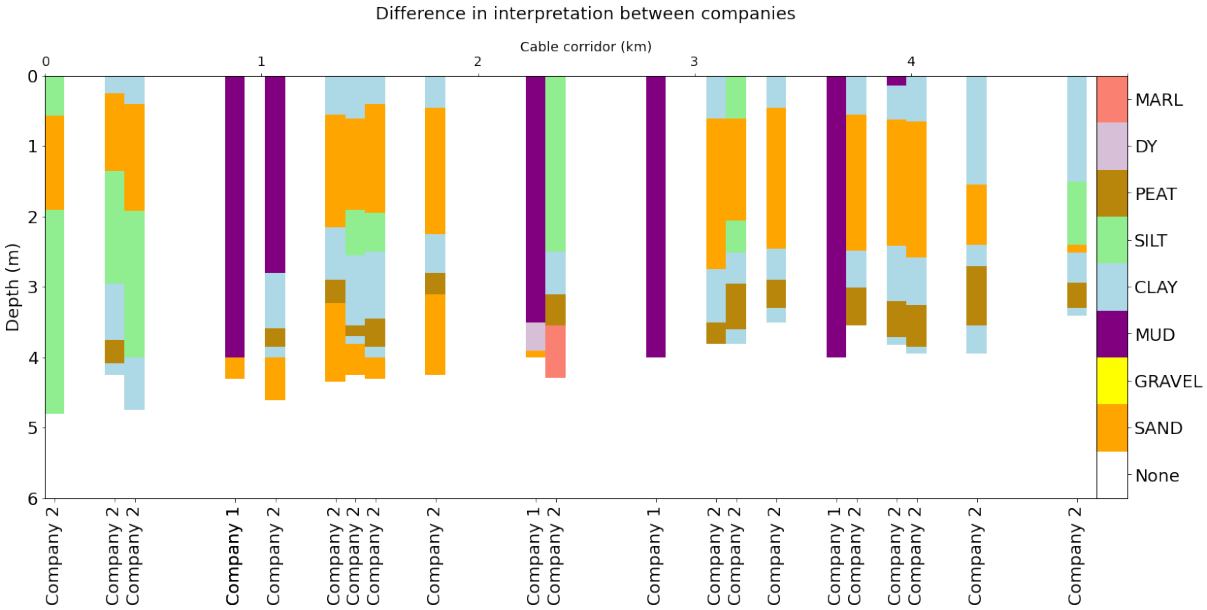


Figure 3.3: This figure visualizes a section of a cable corridor where the soil type interpretation between the companies differs. All GTs of company 1 are largely mud while the GTs of company 2 show a completely different soil layering

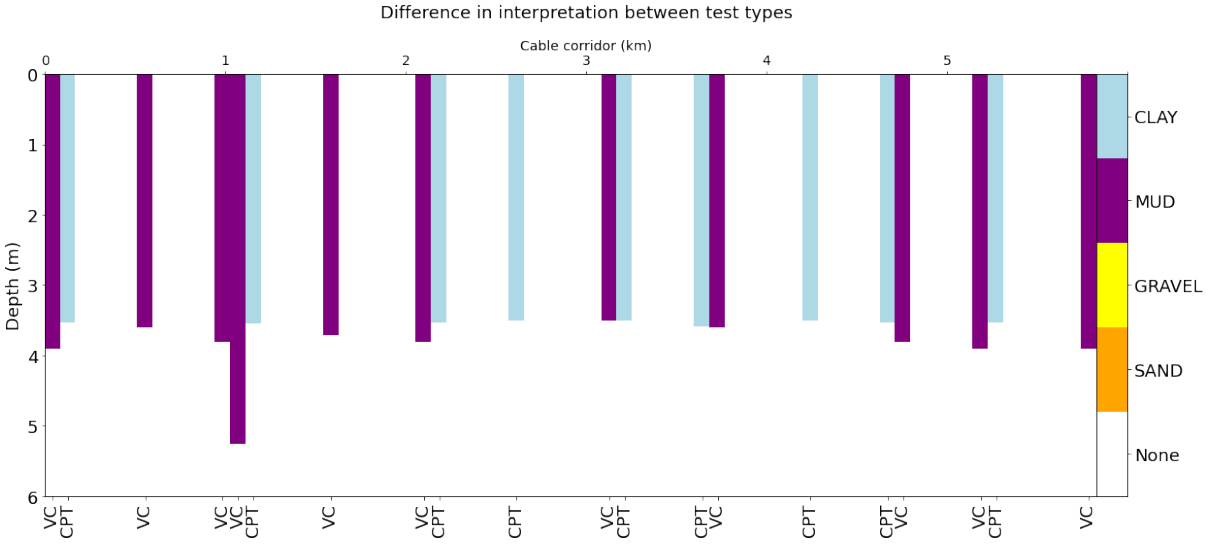


Figure 3.4: This figure visualizes a section of a cable corridor where the soil type interpretation between the test types differs. All VCs are interpreted as mud while all CPTs are interpreted as clay.

Chapter 4

Methodology

This chapter describes the methodology. The first section addresses the data extraction from PDF files through image processing. Secondly, after the data is extracted, it is split into a training and a test set. The conventional models can use this data directly to develop a model without any pre-processing steps. On the other hand, the ML models require pre-processing of the data before training and prediction, covered in the third section. The last section, model selection, clarifies how the best model is selected for the final 3D stratigraphic soil model. A schematic representation of the complete approach for the cable burial assessment is shown in Figure 1.1.

4.1 The Peat Classifier

Since peat is critical in this project, it is essential to predict the occurrence of peat samples optimally. For instance, if many peat samples are not predicted by the model, these locations will not be excavated and will impact the performance and life of the cable. On the other hand, if a lot of peat is predicted, then an unnecessary amount of soil has to be excavated. On beforehand, the criterion for excavation is known, and that is, if there are more than 20 centimeters of peat in the top 3 meters of soil, the soil has to be excavated.

In terms of DSM, the correct prediction of peat is not more important than the correct prediction of other soil types. Achieving as many correctly predicted samples as possible has priority. Accordingly, it is especially important to correctly predict the locations to be excavated and, secondly, the soil types. Due to this difference in priority, a separate peat classifier is designed for the first problem. The only task of this classifier is to predict the locations to be excavated as good as possible. To accomplish this, the peat classifier uses resampling techniques and extra weights for peat samples, as described in 4.4.4. After classifying the peat separately with the peat classifier, a normal classification is performed for all soil types without resampling or extra weights for minority classes. Meaning that the soil types are treated equally. For the peat classifier, a random forest is used, and therefore it was decided to use it only in combination with ML models. In order to retain separation between the conventional models and the ML

models. In Appendix G, a flowchart is presented with the peat classifier.

4.2 Train-Test Split

All available data is split into a training and a test set for developing and evaluating the models. ML models require a training set for training the model. Based on this training set, the algorithm can learn patterns in the features with respect to the outcomes. Therefore the outcomes of the training set must be known. The test set is used to estimate the model's performance on new data that also requires known outcomes. Otherwise, it is impossible to validate the results of the model. Obviously, conventional models also require data to develop a model, but it is less common to withhold a test set. For comparison between the models, all models are developed/trained on the same training set and tested on the same test set. Since the outcomes have to be known, only locations can be used where the soil profile is known, i.e. locations of the GTs. Both VCs and CPTs are used to train and test the models.

Usually, the train-test split is done randomly in machine learning. In this case, this is not intuitive since clusters can then arise with many or few observations. This has as a consequence that there are GTs in the test set where nearby GTs are also in the test set. This reduces the available information of the surroundings in the training set. It is more intuitive to take every n^{th} GT of each export cable as a test set. In practice, it is also more intuitive that the GT density will be less across the entire cable corridor than that there are no observations at all on certain spots. It is common to use 80% of the data for training and 20% for testing, which is also used here. This means that every 5^{th} GT in each cable corridor is in the test set and the remaining tests in the training set. The test set for each corridor is shifted to optimise the information coverage for the test set. The training set comprises 1180 unique GTs, and the test set 294. Because it is known when peat has to be excavated, the amount of GTs to be excavated can be determined in both sets. In the training and test set are 35 and 13 GTs to be excavated, respectively. Figure 4.1 visualizes the train-test split and Table 4.1 shows amount of GTs related to the train-test split. It can be noted that the total number of CPTs is smaller than stated in Section 3.1. This is because there is one CPT whose location is unknown and therefore unusable.

Table 4.1: Train-test split

	Training set	**Test set**	**Total**
VCs	604	153	**757**
CPTs	576	141	**717**
Total	**1180**	**294**	**1474**
Excavated	35	13	48

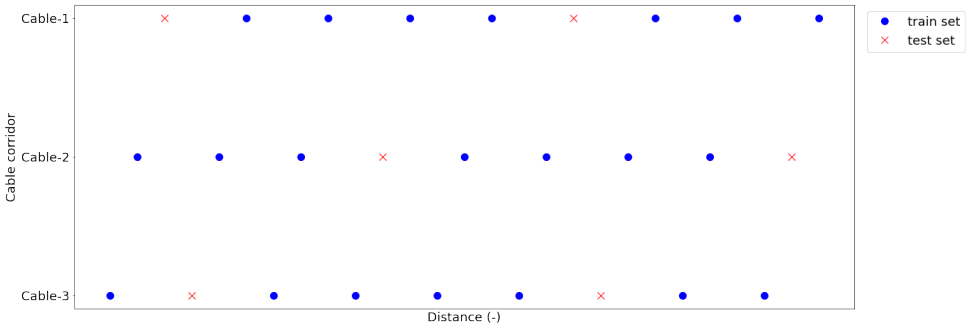


Figure 4.1: Training & test set distribution along the cable corridors. Every fifth GT is in the test set and the test set is shifted for each cable corridor to achieve optimal information coverage.

4.3 Data Structure

The training data consist of 1180 unique GTs with a total length of 4,349.27 meters. Since every centimeter is predicted, this results in 434,927 observations in the training set. The test set consists of 294 unique tests with a total length of 1,086.04 meters, resulting in a test set of 108,604 observations. Each observation has a unique 3D spatial position. The spatial position is the only data provided to the conventional models. The dimensions of the training set are therefore 434,927 x 3. The rows represent unique observations, and the columns represent the features easting, northing and depth. The data provided to the machine learning models have a separate column for each feature in Table 2.3. For both VCs and CPTs, the four nearest tests are included, making 62 columns. This results in a training set with dimensions 434,927 x 62.

On the other hand, the data structure provided to the LSTM is different because it has a third dimension. Usually, this dimension represents the time but, in this case, the vertical position. The same features as the machine learning models are extracted for the LSTM. However, the features of the LSTM data is based on only the two nearest VCs and CPTs instead of four. This is because additional columns are added with sequences, and otherwise, the dimensionality will be too large for the model. The added sequences contain the primary soil types of the nearest GTs from seafloor to a depth of 6 meters. The sequence has a length of 12, and therefore the majority soil type every 50 centimeters is included. Additionally, from the nearest CPTs, the median and variance of tip resistance, sleeve friction and pore water pressure of every 50 centimeters are included. The data provided to the LSTM has for each observation a 2D matrix with a size 12 x 54 instead of a vector.

4.4 Data Processing

The applied data pre-processing steps consist of data cleaning, scaling, handling missing data, and resampling. Then, a final post-processing step is applied after training which is data aggregation.

4.4.1 Data Cleaning

Data cleansing is a vital process in modelling and even more so when using machine learning models, as these models rely more on data quality. Many data cleaning is already being done by the companies that provided the data. This includes noise filtering of the sub-bottom profilers, bathymetry and CPT data. The steps taken in this study is removing a single CPT without a spatial position. Additionally, parts of some VCs were lost, leading to missing values. These missing values were also removed so that the ML models do not learn and predict these core losses.

4.4.2 Data Scaling

The two most common types of data scaling are standardisation and normalisation. With standardisation, a Gaussian distribution is assumed for the scaled variable. Since the data is not Gaussian distributed, normalisation is used to scale the data. Normalisation is done by the MinMax scaler, which is defined in equation 4.1. Each feature is separately scaled by the minimum and maximum values of the training set. After scaling all the values will be in the range of [0,1]. It is possible that new data, for example, the test set, contains values outside the bounds of the minimum and maximum. Resulting in a scaled value that is not in the range of [0,1]. To handle these out-of-bounds values, they can be removed from the dataset or can be limited to a predefined minimum and maximum values. In this research, the values are limited to 0 and 1.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.1}$$

4.4.3 Missing Data

The next pre-processing step is handling missing data. Except for the location variables and the distances to nearby GTs, all features have missing data for some samples. For instance, the nearest VC for a sample at 3 meters depth can have a length of 2 meters. Therefore the soil type at an equal vertical position is not known and missing in the data. These missing values need to be handled because most ML models cannot deal with missing data. XGBoost is an exception to this, as it has an algorithm that finds the best split at each decision node for missing data. The loss is calculated twice for each decision node, one for each direction the missing values can take. Naturally, the direction with the lowest loss is chosen. However, the other models need a different way of dealing with missing values. The missing soil types are assigned to a new and unique category so that they are not added to another soil type. Missing values of continuous variables have been replaced by 0, which is a common approach.

4.4.4 Data Resampling

Data resampling is only performed for the peat classifier and is performed after handling missing values. SMOTE is used to oversample the peat samples, and random undersampling is used to undersample sand, clay, mud and marl. The oversampling and undersampling factors are optimized because it is not known in advance to what extent oversampling and undersampling are required.

4.4.5 Data Aggregation

The last step is post-processing the data by aggregating the predictions. Figure 3.1 shows the proportions of the soil types in percentages in the training data. It can be noticed that the data is highly imbalanced. Over 50% of the soil types is sand, while dy, gravel and peat together make up less than 1% of the types. With these ratios, it is extremely difficult for an ML model to learn the minority classes. For this reason, dy and gravel are aggregated to other soil types. Dy is added to peat because of its high organic content, and gravel is added to sand. Since peat plays a crucial role, it is not aggregated to another soil type.

Besides, clay and mud are also aggregated together because of the difference in soil type interpretation between test types as shown in Figure 3.4. The aggregation is done after prediction to preserve the variation in the data. For instance, if aggregation is done before training, the feature soil type at equal vertical position of the nearest VC will contain less variation and information because three soil types are already aggregated.

4.5 Model Selection

This section clarifies the strategy to select the best model. First, the classification metrics to measure the performance of the models are explained. Then a section cross-validation describing the method to optimize the hyperparameters of the ML models, which is naturally the next section. Finalizing with a subsection about the model evaluation.

4.5.1 Classification Metrics

Classification metrics are used to measure and compare the performance of models. Numerous metrics are available, all (partly) based on the four possible outcomes in classification: 1) True Positives (TP); 2) True Negatives (TN); 3) False Positives (FP) and 4) False Negatives (FN). An overview of the possible outcomes is given in a so-called confusion matrix in Table 4.2. Typically, the performance of machine learning models is evaluated by a confusion matrix (Chawla et al., 2002). This study is confronted with a multi-class classification problem. Therefore, each class has its own confusion matrix. For instance, in the case of sand, TP: the model predicts sand, and the soil actually is sand; TN: the model predicts not sand, and it is not sand; FP: the model predicts sand, and it is not sand; FN: the model predicts not sand, and it is sand. In medical applications, the latter is known as a type II error which is a very bad outcome. A person who

is actually sick is then labelled as non-sick with all its consequences. On the other hand, a FP, type I error, is much less dramatic. In this case, a type II error can be associated with the misclassification of peat because the consequences of a misclassification of peat are significantly worse than the misclassification of sand.

Accordingly, metrics with equally weighted errors, such as accuracy, do not suit the problem of the peat classifier. On the other hand, peat is only important when excavation is necessary. Accordingly, the peat classifier is optimized on how many of the GTs should be excavated according to the model and how many of these correspond to the number of GTs that should actually be excavated. One can think of this in the following manner. Obviously, the peat classifier predicts the entire test set. Then, it can be determined in how many predicted GTs more than 20 centimeters of peat in the top 3 meters is predicted. For evaluation, these predicted GTs to be excavated can be compared to the actual GTs to be excavated.

Table 4.2: Confusion matrix

	Predicted negative	**Predicted positive**
Actual negative	True negative (TN)	False positive (FP)
Actual positive	False negative (FN)	True positive (TP)

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{4.2}$$

4.5.2 k-Fold Cross-Validation

Cross-validation (CV) is a renowned method that is used to find the optimal hyperparameters* of a model. At first, multiple hyperparameter combinations are defined. Subsequently, CV is performed on the training set for each combination. The parameter k determines the number of folds into which the training set is divided. A fold is simply $\frac{1}{k^{th}}$ part of the training set, and each fold is used once as a validation set. The other k-1 folds are then used for training, after which the model is tested on the validation set. This procedure results in k CV results for each hyperparameter combination.

The value k value should be chosen carefully because when the value is poorly chosen, it can result in not representative results of the CV. With a possible consequence that non-optimal hyperparameters are selected. It is essential that each training and validation group is large enough to be statistically representative for the entire dataset. This means that the value of k should not be too large because increasing the value of k decreases the fold size.

There is a bias-variance trade-off associated with the value of k. Typically, one performs k-fold CV with k=5 or k=10. These values have been shown empirically to yield test error rate

*A hyperparameter is a parameter which controls the learning process of a machine learning model. The hyperparameters are set by the user before training the model and are therefore not optimized during training. Consequently, hyperparameters should be optimized by the user.

estimates that suffer neither from excessively high bias nor from very high variance (James et al., 2017). In this study is opted for k=5. Normally, after the choice of k is made, the training data is shuffled randomly and split into k groups of equal size. Then one group is taken as a validation set and the remaining groups as a training set. As with the train-test split, this is not intuitive, and therefore the folds are created in the same way as the train-test split. The first fold contains GT 1, 6, etcetera and the second fold 2, 7, etcetera. The formula is shown below.

$$fold_i = \{i + j \cdot k\} \tag{4.3}$$

where $j = 1, 2, ..., \frac{n}{k}$ and n is the total number of GTs in the training set. Each iteration in the CV, the validation set shifts to the consecutive GTs. After five iterations, all GTs in the training set have been included once in the validation set.

Data pre-processing procedures are performed only on the training set to avoid data leakage. For example, if scaling is performed on all folds, the model already has information of the validation set, which can lead to overestimation of the performance of the model. After data pre-processing, the model is fit on the training set and evaluated on the validation set. This procedure is done for each fold resulting in 5 evaluation scores for the model. Figure 4.2 shows a schematization of 5-fold CV.

CV is then performed on each parameter set in the parameter grid. The best average score can be seen as the optimal hyperparameter set. When the two best scores are close to each other, but the best model has a higher variance, one could opt for the second-best score due to more stable results.

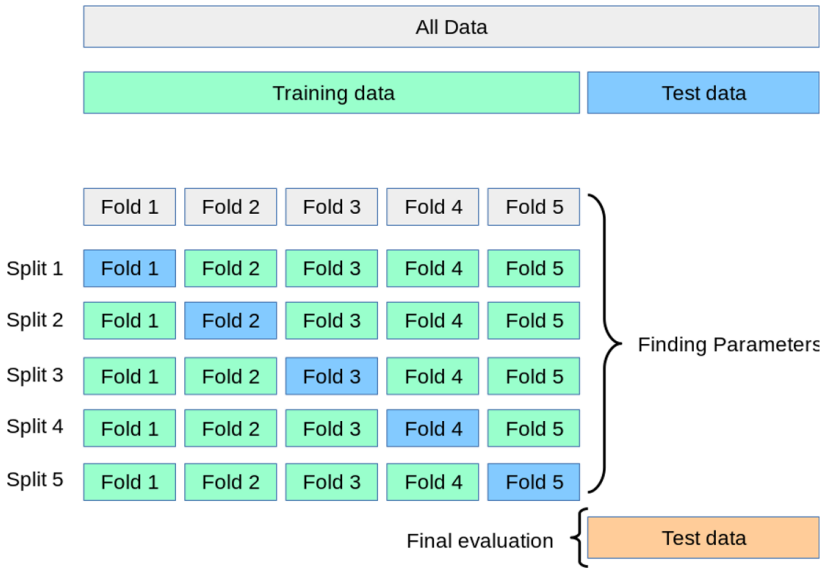


Figure 4.2: Schematization of 5-fold CV (scikit learn, n.d.). First, the training set is divided into 5 equal sized folds. Then, each fold is used once in the validation while training the model on the other four folds. This results in five CV results which can be used to determine the best hyperparameters.

4.5.3 Hyperparameter Optimization

Hyperparameter optimization is essential in ML because it controls the learning process. When the model's hyperparameters are not adjusted to the data, then the model will probably perform very poorly. The hyperparameters are set performing a grid search with CV as described in Section 4.5.2. A preliminary random grid search is performed to pre-select and find good combinations of hyperparameters. This approach is preferred because many different hyperparameter combinations can be efficiently explored. It is often a challenge to come up with well-founded arguments to use specific hyperparameters as it is often trial and error to find the optimal hyperparameters. Figure 4.3 visualizes how random grid search effectively explores the hyperparameter space with respect to exhaustive grid search. Selecting good combinations with random grid search reduces the hyperparameter space, after which an exhaustive grid search is performed to find the optimal performing combination. The best combinations from the random grid search are included in the following sections, which is the final hyperparameter grid for an exhaustive grid search. The complete optimization procedure is visualized in Appendix H.

All models are implemented in python 3.8.8. Random forest and XGBoost uses scikit-learn 0.24.1, and the LSTM-network is implemented using keras 2.7.0.

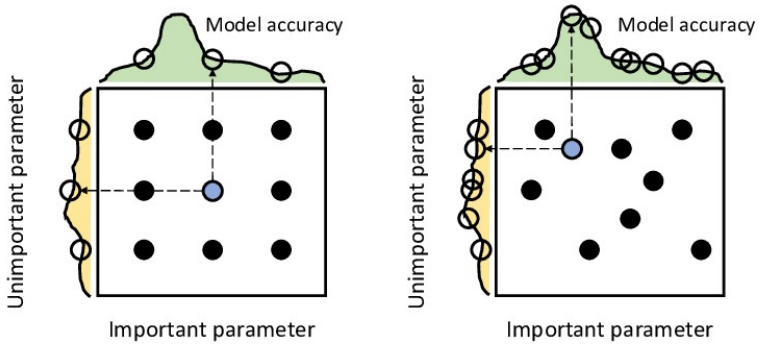


Figure 4.3: A visualization of exhaustive grid search (left) compared to random grid search (right) (Pilario et al., 2020). The square represents the hyperparameter space of two hyeperparameters. The performance of these hyperparameters are shown as graphs at the side (yellow graph) and at the top (green graph). The right figure shows, by cross-validating 9 hyperparameter combinations (black dots), that a lot more values are explored with random grid search than with exhaustive grid search. Therefore, random grid search is preferred when it is unknown which hyperparameters perform well.

Moving probability threshold

Since machine learning models can compute probabilities for the classes, it is possible to deviate from the usual threshold. Usually, the class with the largest probability is classified as 1, which is the predicted soil type, and others as 0. To achieve the highest score, it is imperative that the model can distinguish the classes as much as possible. When the model minimizes the loss function, it is highly likely that a different threshold can provide a higher score. Consequently,

performing a grid search with the usual largest probability, potentially good models can be overlooked. For instance, due to the low number of peat samples in the data and the crucial role it plays, it can be decided to classify a sample as peat if the probability is greater than 20%, while it does not have the largest probability. Figure 4.4 shows the real profile of a VC, the predicted profile and the probability profile. In this example, the soil type with the largest probability is the predicted soil type.

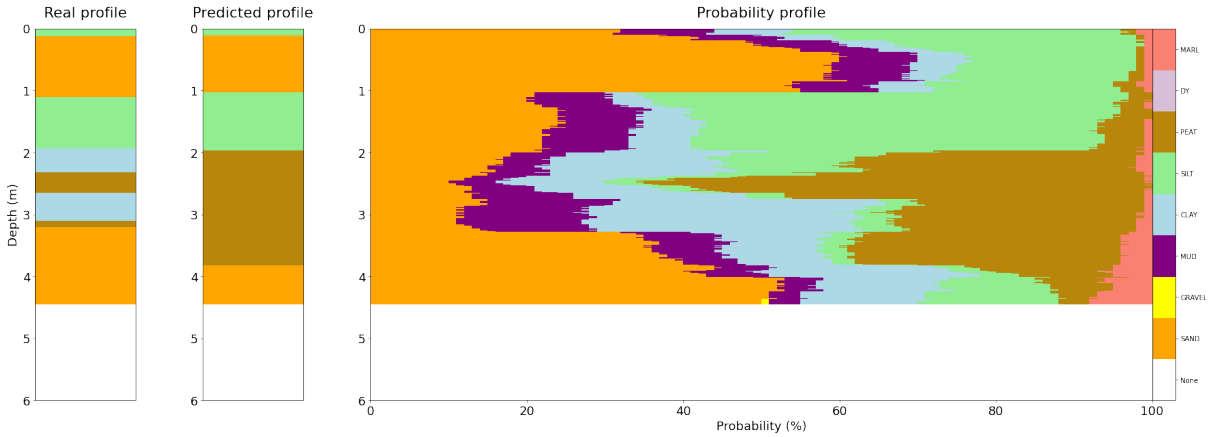


Figure 4.4: An example of a predicted probability profile for a GT. On the right the real profile is shown, in the middle the predicted profile and on the left the probability profile. The probability profile should be read from left to right for each depth. It visualizes the probability assigned to each soil type for the corresponding depth.

Peat Classifier

For the peat classifier, a random forest is used. The hyperparameters optimized for this classifier are the oversampling and undersampling factors for resampling the training data. The movable probability threshold is used too to accomplish the task it is designed for, i.e. classifying the locations of soil to be excavated as well as possible. This works in the following way, the peat classifier is trained on the resampled training data and computes the probabilities for the classes. Shifting the probability threshold makes it possible to determine how many samples are classified as peat and, therefore, the number of tests to be excavated. It is known beforehand how many GTs need to be excavated in each validation set. Then, the user can set how many of the actual excavated GTs in the validation set should be correctly predicted. Logically, the more GTs that need to be correctly predicted come at the expense of accuracy. For instance, if the threshold is zero, all samples will be classified as peat, and all GTs that have to be excavated are correctly classified, but the accuracy will be very low. The most difficult to learn samples have the lowest probability and come at the cost of more accuracy than easier to learn samples. Therefore it is not desired to correctly predict 100% of the excavated GTs in the validation sets and is opted for at least 80%. This criterion is used while optimizing the model and gives an indication of the percentage of correctly classified GTs to be excavated in new data.

By following this procedure, a probability threshold is obtained. Only the samples with a probability higher than the threshold and were at least 20 samples in the GT have a higher probability than the threshold are classified as peat. Samples in a GT with less than 20 samples with a higher probability than the threshold are normally classified whether or not their probability is higher than the threshold. In this way, the classifier focuses only on the GTs to be excavated and not on all peat samples, which improves the accuracy. Five thresholds are obtained after 5-fold CV, where the minimum threshold is used for the final model. Additionally, predicted peat samples farther than 500 meters from the nearest peat sample in the training set are ignored. By analyzing the five validation sets, it became clear that the maximum distance of a peat sample in the validation set was no farther than 500 meters from the nearest peat sample in the training set.

SMOTE is used to oversample the peat samples, and random undersampling is used for sand, mud, clay and marl. The oversampling factor is multiplied by the original amount of samples. Thus, a training set with 100 samples and an oversampling factor of 3 means that the resampled training set contains 300 peat samples, of which 100 real samples and 200 synthetic samples. An oversampling factor of 1 means that no synthetic samples are generated. The undersampling factor works the other way around, but this factor is used in the formulas below for different soil types. The presence of sand is much higher than the other soil types. Therefore, a larger undersampling factor is desired.

1. SMOTE oversampling factor (1, 3, 5, 10, 15, 20)
2. Random undersampling factor (0, 3, 5, 7)

The formulas for the undersampling factors are shown below. The actual undersampling factors can be computed by filling in the four undersampling factors shown in the formulas below. Additionally, it is possible to use class weights to give certain classes more attention than others. The hyperparameters are optimized with class weights that restore the class imbalance. This means that if sand has twice as many samples after resampling as peat, the misclassifications of peat weigh twice as much as those of sand.

- sand: $(undersamplingfactor * 3) + 1$ (1, 10, 16, 22)
- clay: $(undersamplingfactor * 2) + 1$ (1, 7, 11, 15)
- mud & marl: $undersamplingfactor + 1$ (1, 4, 6, 8)

Random forest

Random forest has many hyperparameters that can be tuned. The four below followed by their selected values will be tested to find the optimal combination:

1. Maximum depth of a tree (5, 10, 20, 40)

2. Minimum number of observations for a node to split (2, 6, 10, 20)

3. Minimum number of observations at a leaf (1, 2, 5, 10)

4. which criterion to use for a split (Gini Index, entropy)

The number of trees grown by a random forest does not need to be optimized because the results converge with an increasing number of trees. Increasing the number of trees will not overfit the data because each tree can be considered independent due to the random nature of the model.

The first three hyperparameters are regularizers and determine when the trees stop growing. Accordingly, these three hyperparameters, and also the fourth, are simultaneously set. By tuning multiple hyperparameters simultaneously, the chance to get stuck in local optima decreases.

The maximum depth of a tree determines how many decisions can be made sequentially. After the first split, there are two leaves corresponding to a depth of one and two different groups. The second layer also divides both subsets into two groups, making a total of 4 groups. This means that there are $2^{\text{depth of the tree}}$ groups. For a depth of 20, there are already 1,048,576 possible groups, but this will probably be regularized by one of the other regularizers.

The second hyperparameter, the minimum number of observations for a node to split, is also a regulator for the complexity of the trees. When a node contains fewer observations than the threshold, the node is not split. This also affects the maximum depth of a branch. A tree with a depth of 20 whereby at each split, one subset stops splitting has a total number of groups of 21, which is considerably less than the maximum mentioned above. Figure 2.6 shows an example for a tree with a depth of three whereby each split one subset stops splitting and has a total of four groups. Accordingly, the minimum number of groups is the depth of the tree plus one. The third hyperparameter prevents a node from splitting if at least one subset is less than the minimum number of observations at a node.

XGBoost

XGBoost knows a lot less hyperparameters compared to a random forest. The four hyperparameters below followed by their selected values will be tested to find the optimal combination:

1. Number of trees that will be grown (5, 10, 15, 20, 30, 50)

2. Maximum depth of a tree (4, 6, 8, 12, 20)

3. Learning rate (0.01, 0.05, 0.2, 0.6)

4. Minimum loss reduction for a split (0, 0.3, 0.8)

Unlike random forest, XGBoost grows trees sequentially and can therefore overfit the data by increasing the number of trees. Thus, this hyperparameter needs to be optimized. The

maximum depth of a tree is similar to the hyperparameter in a random forest. The learning rate is an essential hyperparameter for ML models and determines how fast weights in the model change, i.e. how fast it adapts to new data. Finding an appropriate learning rate for the model is crucial because high learning rates pass by global optima due to its large steps. Besides, too low learning rates do not reach the global optima. Figure 4.5 shows the effect of different learning rates. Finally, the last hyperparameter is comparable to the number of observations at a leaf because it considers a split and only makes the split if it meets at least the minimum loss reduction. At first glance, it might seem more similar to the minimum number of observations for a node to split, but this hyperparameter does not consider a split if there are too few observations.

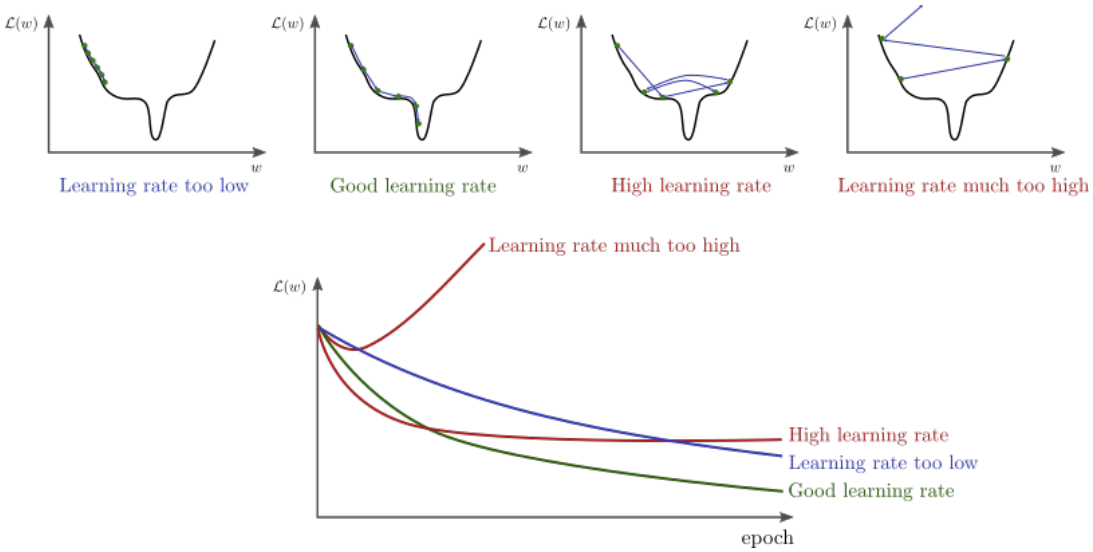


Figure 4.5: An example what the effect of the learning rate is (Hammel, 2019). The upper graphs shows the loss function with a obvious minimum. When the learning rate is to small the minimum is not reached (left figure). On the other hand, if the learning rate is too large it will pass by the minimum (two most right figures).

Long short-term memory network

The degrees of freedom for an ANN reaches infinity when the architecture also is taken into account. The architecture is formed by the layers in a model and the number of units, i.e. the dimension of the layer(s). To limit the number of options, 4 architectures were designed. It concerns the following four:

- Model 1: 1 layer LSTM unidirectional
- Model 2: 1 layer LSTM bidirectional
- Model 3: 2 layer LSTM unidirectional
- Model 4: 2 layer LSTM bidirectional

An ANN with a single hidden layer is referred to as "vanilla", and a double layer is referred to as "stacked". All architectures end with a dropout layer followed by a fully-connected layer to map the output between 0 and 1. Further, dropout is used between both LSTM layers for the latter two models. For dropout a value of 0.4 is used. This means that in each epoch 40% of the neurons are excluded from training the model, which generalizes the model. The number of neurons in the layers and the batch size is optimized. The following values are used:

1. Number of neurons in each layer (5, 10, 25, 50, 100)
2. Batch size (32, 64, 128, 256, 512)

Now that the architectures of the models are defined, the remaining specifications can be determined. First, the fully-connected end layer requires an activation function. Here the softmax function is chosen, widely used as the final layer of neural networks for classification, because it ensures that the output sums up to 1. Therefore the result can be interpreted as probabilities for the classes. Also, a loss function is required, which will be minimized by the model. An intuitive choice for the loss function is the categorical cross entropy (CCE). CCE penalizes the difference between the true value and the predicted probability of the class. The penalty increases exponentially for higher differences. It also allows adding weights to particular classes. When class weights are provided, CCE can be written as:

$$CCE = -\sum_{i=1}^{N} W_i y_i \cdot \log\ P_i \tag{4.4}$$

where N is the number of classes; W_i represents the class weight of the i^{th} class; y_i is the true value $\in \{0, 1\}^{\dagger}$ and P_i is the computed probability $\in$ [0,1]‡. To minimize the loss function, an optimizer is required to update the weights of the model. Nowadays, the most used optimizer in neural networks is Adaptive Moment Estimation (Adam). Adam is also used in the performed LSTM networks.

4.5.4 Model Evaluation

After hyperparameter optimization, the machine and deep learning models are trained on the entire training set with the optimal hyperparameters. Then, all models are evaluated on the test set and compared with the classification metrics from Section 4.5.1. Appendix I provides an overview of the final model evaluation.

4.6 3D Soil Model

The best performing model is used to develop a 3D stratigraphic soil model along the cable corridors. Since the distance between the GTs is large, the mean distance is approximately 500

†Braces are a mathematical expression of a set. Meaning that y_i is either 0 or 1.

‡Brackets are a mathematical expression of a range. Meaning that P_i is a value in the closed interval from 0 to 1.

meters. It is insufficient only to use these locations to develop a model for the entire length of the cable corridor. This does not hold for triangulation because this technique interpolates between the GTs and instantly creates a model for the entire corridor. However, machine learning models do not interpolate but predict the soil profile at designated locations instead. Based on a given location and engineered features, a machine learning model can make a prediction of the soil profile. As a consequence certain locations along the corridor need to be selected for prediction. The space between these locations must be so small that there is hardly any soil profile variation between them. Otherwise, there is a risk of missing important soil layers. After selecting an appropriate separation distance, the predicted soil profiles at the designated locations can be placed side by side to form a near-continuous soil model, which is constrained at the locations of the GTs.

Chapter 5

Results

The chapter discusses the obtained results. The first section describes the results of the proposed methodology from section 4.5.3 for optimizing the hyperparameters. The following section discusses the estimated feature importances. Then, the results of the final models with the best hyperparameters are discussed in section 5.3. Finally, the obtained results are analyzed.

5.1 Hyperparameter Optimization

This section provides the complete optimization procedure where only the ML models are optimized. The sections follows the same order as in the report starting with the peat classifier, then random forest and XGBoost and finally the LSTM. It should be noted that the peat predictions of random forest, XGBoost and LSTM have already been corrected by the peat classifier. This applies to the optimization procedure but also to the results of the final models.

Peat Classifier

For the peat classifier, the oversampling and undersampling factors were optimized. These results are shown in a heatmap in Figure 5.1, on the x-axis the oversampling factor and the y-axis the undersampling factor. Each resampling combination has a square mentioning the total number of predicted GTs to be excavated in all five validation sets. Several values are not a whole number because it is the average number of three times CV. The CV results were quite unstable, and therefore was decided to perform CV multiple times. The unstable behaviour is due to the large undersampling factors up to 22 for sand. Undersampling is done randomly and depends on which samples are removed and which are not. An undersampling factor of 22 means that only $\frac{1}{22^{nd}}$ of the sand samples persist and that each round of undersampling most likely has an entirely different set of samples.

The results satisfy the criterion of predicting at least 80% of the GTs that should actually be excavated correctly in each validation set. The validation sets from 1 to 5 contain 5, 6, 8, 10 and 6 GTs to be excavated, respectively. This means that at least 29 GTs* from the

*At least 29 as the GTs to be excavated in each validation set must be rounded up.

mentioned number are correct. The second number in the squares is the standard deviation of the residuals[†]. The colour of the squares indicate the number of GTs to be excavated. The lower, the better. As can be seen, there is not a clear optimum. This is also reflected in the graphs of Figure 5.2 when the results are averaged per factor. Here, it can be noticed that the oversampling factor does not have much influence on the number of predicted GTs to be excavated. It is plausible that oversampling has almost no effect due to the use of class weights[‡]. Appendix J presents the effect of the oversampling factor without class weights. These results clearly show an effect and optimum of the oversampling factor, although the results are considerably worse than with class weights. Of course, class weights are used in the final model.

On the other hand, the undersampling appears to have an effect. Thus, the optimum depends only on the undersampling factor regardless of the oversampling factor. This can also be noticed in the colour-coded map where the columns' optima are concentrated around the undersampling factor of 5. However, an undersampling factor of 7 in combination with an oversampling factor of 1 has the best performance. This combination also has the smallest standard deviation and is therefore used in the final model. Figure 5.3 shows the change in soil type ratios due to resampling. The original training set contains 434,927 samples, and the resampled set contains 48,940, which is just over 10% of the original set.

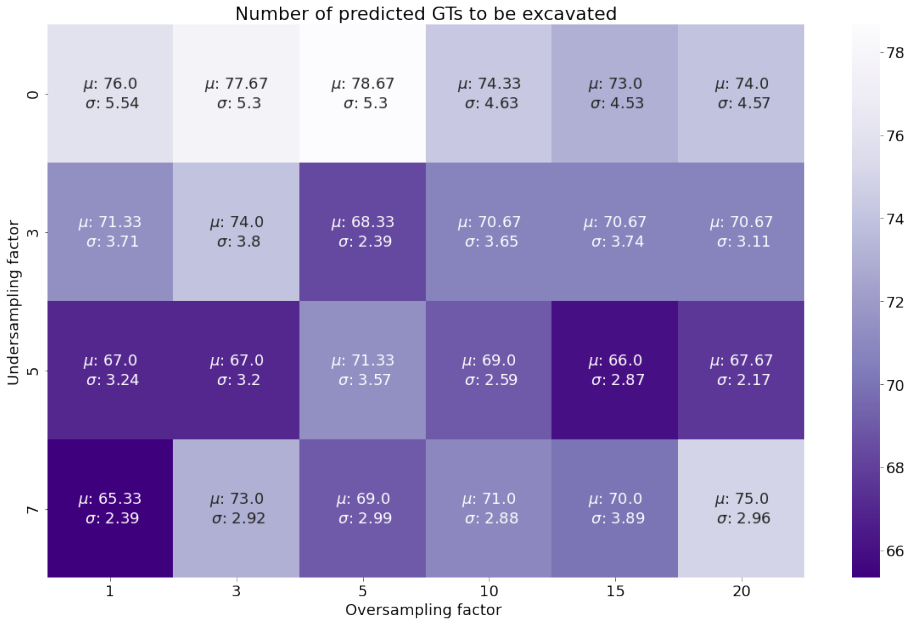


Figure 5.1: CV results of peat classifier. The average number of predicted GTs to be excavated and its standard deviation is shown for each resampling combination. The results satisfy the criterion of at least 80% of the GTs to be excavated in the validation set are correctly predicted. This means that for all results at least 28 GTs are correct.

[†]Number of predicted GTs to be excavated minus the actual number of GTs to be excavated in the validation set.

[‡]explained in Section 4.5.3 under the heading Peat Classifier.

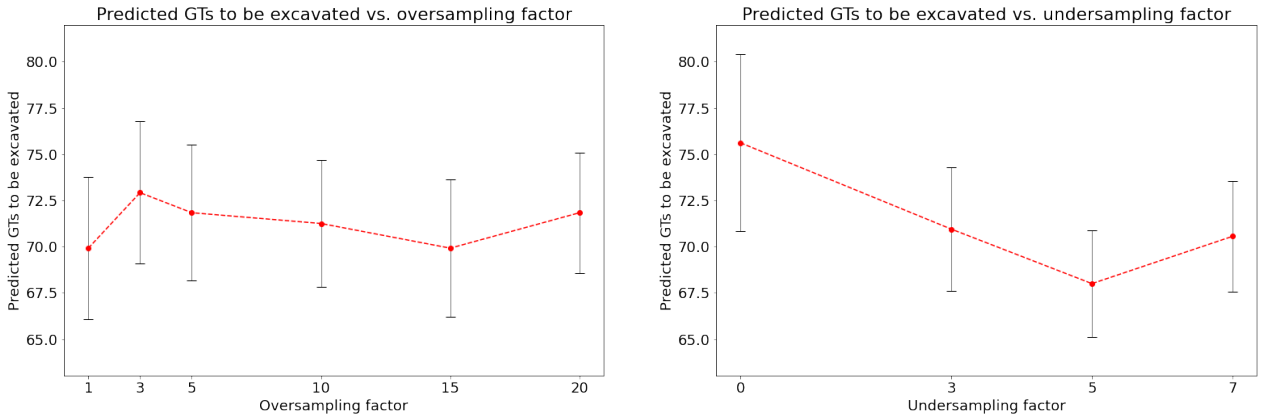


Figure 5.2: The left figure shows the effect of the oversampling factor and on the right figure the effect of the undersampling factor. The red dots are the predicted number of GTs to be excavated in all validation sets, where the actual number is 35. The bars represent the standard deviation of the residual at each validation set.

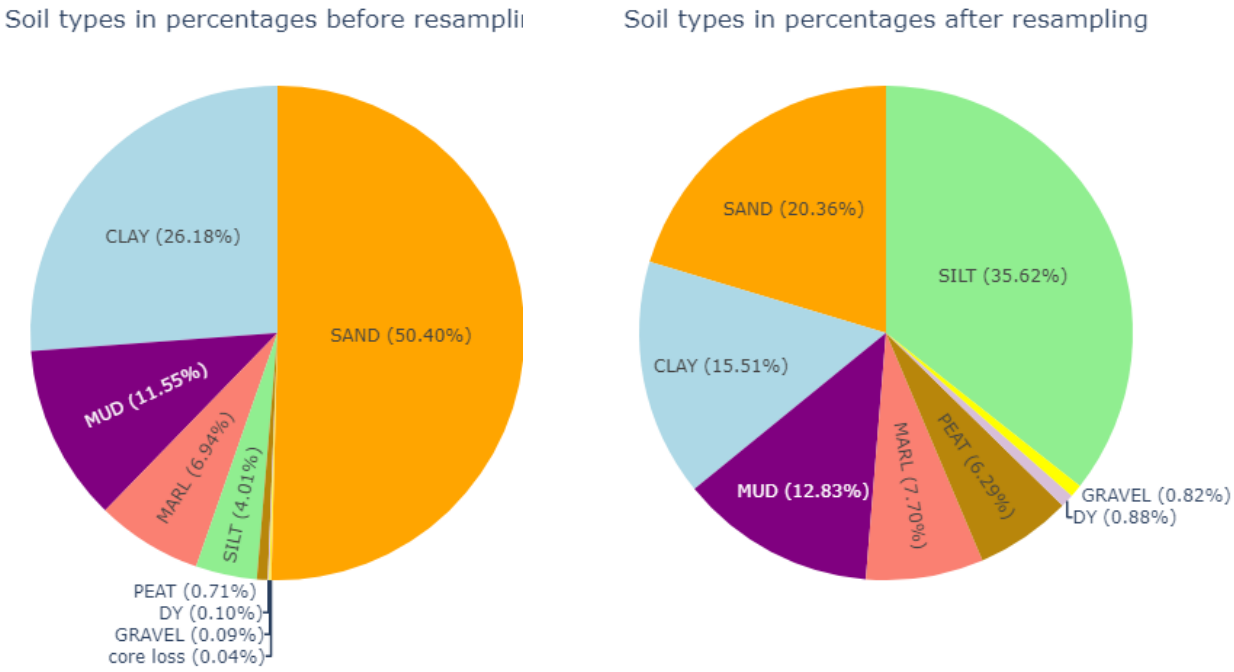


Figure 5.3: On the left the soil types in percentages of the original data (only training set) and on the right the percentages after resampling without oversampling and an undersampling factor of 7. It can be observed that peat ratio is increased.

It is evident that resampling the training set can improve the model's discriminating ability of peat, as all combinations with undersampling perform better than without resampling. This can also be seen in Figure 5.4. The left figure shows the probabilities of peat for all peat samples in a validation set before and after resampling. Nearly all probabilities are higher after resampling, which implicates that the model becomes more confident about peat. The complete

probability profile can be found in Appendix K. In addition, it is also informative whether the probability of peat increases for all other soil types. This is not the case, as can be seen in the right figure. This figure shows the mean probability of peat in each soil type before and after resampling. The mean probability of peat increases significantly for peat and dy, and in all other soil types, it remains approximately the same. Meaning that resampling improves the discriminating ability of peat and does not affect the discriminating ability of other soil types. Ultimately, dy is aggregated to peat due to its high organic content, and therefore the increase for dy is not alarming. The high organic content explains why the probability of peat also increases with dy.

It can be observed that not a One versus Rest (OvR) method is used because all soil types are presented in Figure 5.4. An OvR method was also implemented, but the results were considerably worse. The best performing OvR method with an undersampling factor of 20 and without oversampling resulted in 79 GTs to be excavated. The reason for this difference is the variation in the output. Oftentimes peat is found next to silt and clay in the project area. When using an OvR method, all other soil types are aggregated, therefore it is not clear to the model next to which soil types peat often occurs.

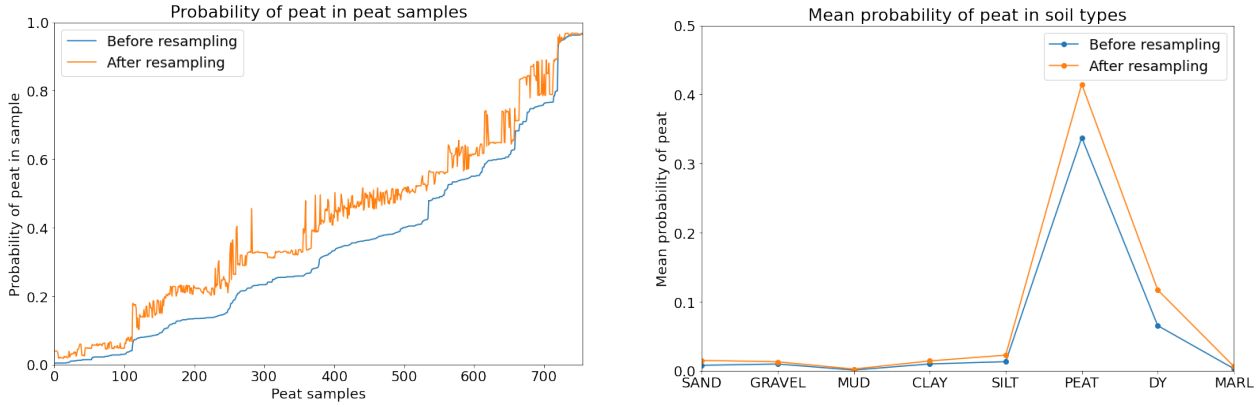


Figure 5.4: The left figure is sorted from low to high probability before resampling. The peat classifier assigns to nearly all peat samples a higher probability of peat after resampling. The right figure shows that the probability of peat for the other soil types do not increase.

Random Forest

The best CV results of the random forest optimization with entropy as the criterion and a maximum tree depth of 20 are shown in Figure 5.5. The other two optimized hyperparameters are on the x-axis and the y-axis. Each hyperparameter set has a square mentioning its average CV accuracy at the top and its standard deviation underneath. The colour of the squares indicates the level of accuracy. The rows represent the minimum number of samples for a node to split and the columns represent the minimum number of samples at a leaf. The impact of those two hyperparameters is minor compared to the maximum depth of a tree or the criterion used. The results for the Gini Index are largely worse than those of entropy and are presented in Appendix L.1 together with all entropy results. The best CV score of 84.32% is obtained by

the hyperparameter set with entropy as the criterion, a maximum tree depth of 40, a minimum of 6 samples at a node to split and a minimum of 1 sample at a leaf. However, this is 0.01% better than the accuracy's in the last column of Figure 5.5. On the other hand, the standard deviation of these accuracy's is 0.13 lower. Therefore, are these hyperparameter combinations preferred. Four combinations have exactly the same results. Since the computational costs for this model are in the order of a few minutes, it is not of interest to Boskalis. Therefore the hyperparameters that are closest to the default settings are used in the final model. The final hyperparameter set is the following: entropy as the criterion, a maximum tree depth of 20, a minimum of 2 samples at a node to split and a minimum of 10 samples at a leaf.

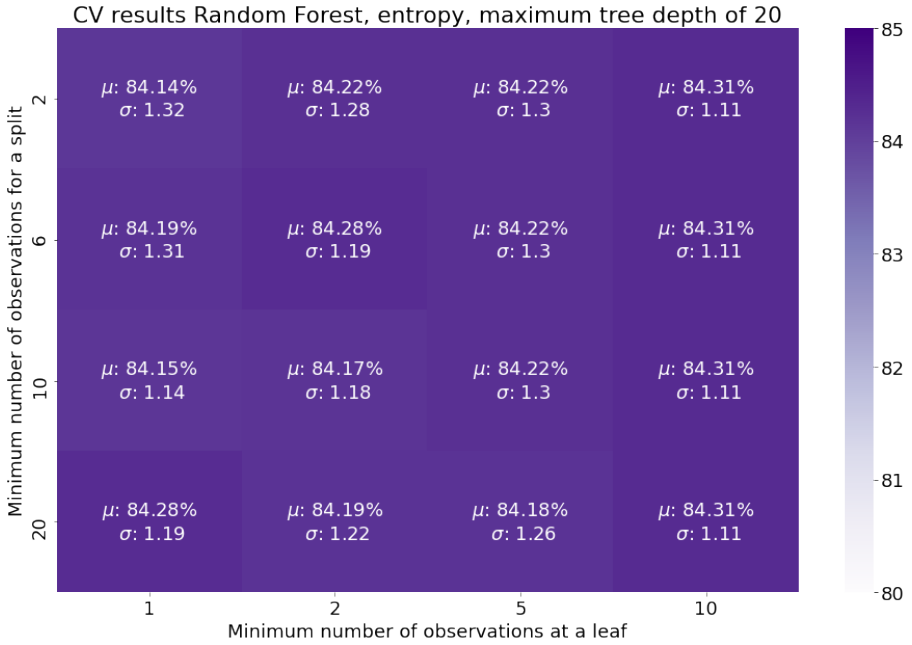


Figure 5.5: Best CV results of random forest with entropy and a maximum tree depth of 20. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

XGBoost

The hyperparameter optimization results of XGBoost with gamma is 0.8 are shown in Figure 5.6. The complete optimization results are presented in Appendix L.2. It can be observed that gamma and the learning rate have a minor impact on the results. Cause of this, there are multiple hyperparameter sets with the highest accuracy. All are obtained by 20 trees, a maximum depth of 8, a gamma of 0.8, and multiple learning rates. Again, the computational cost is in the order of a few minutes. Consequently, 0.2 is selected for the learning rate because this is closest to the learning rate of 0.3, which is the default learning rate of XGBoost. This results in the best hyperparameter set with 20 trees, maximum depth of 8, a learning rate of 0.2 and a gamma 0.8. The best accuracy is 84.62%.

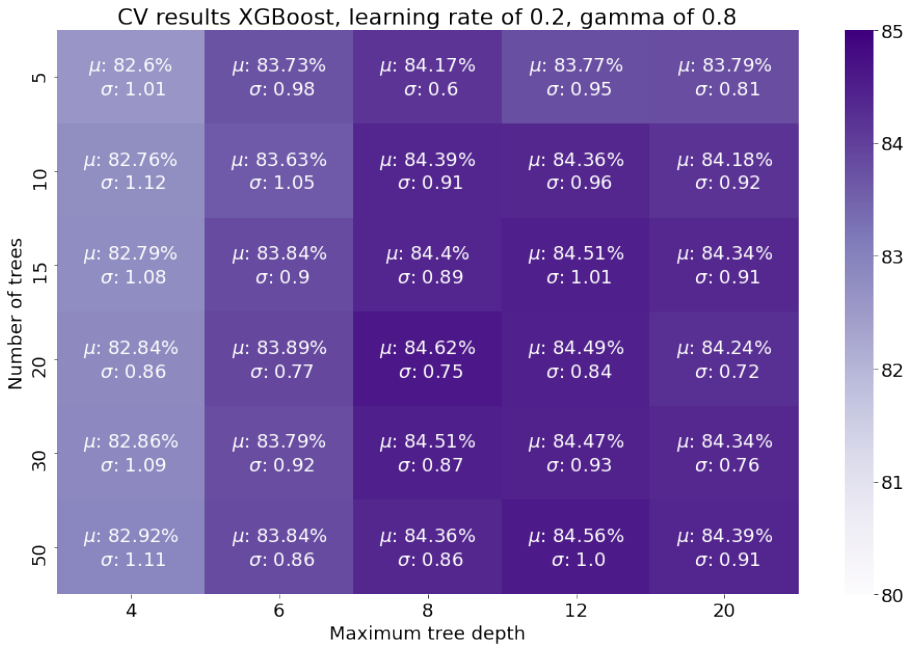


Figure 5.6: CV results of XGBoost. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

Long Short-Term Memory Network

Figure 5.7 presents the optimization results of the bidirectional stacked LSTM. The complete optimization results are presented in Appendix L.3. It can be observed that the best results for all architectures are concentrated in the lower-left corner, i.e., a high number of neurons and a small batch size. The results between different architectures are quite similar. However, bidirectional stacked, the most complex architecture, in combination with 100 neurons and a batch size of 32, obtained the highest accuracy of 83.57%. The combination of 100 neurons and a batch size of 32 is the best performing combination for each architecture, except for vanilla.

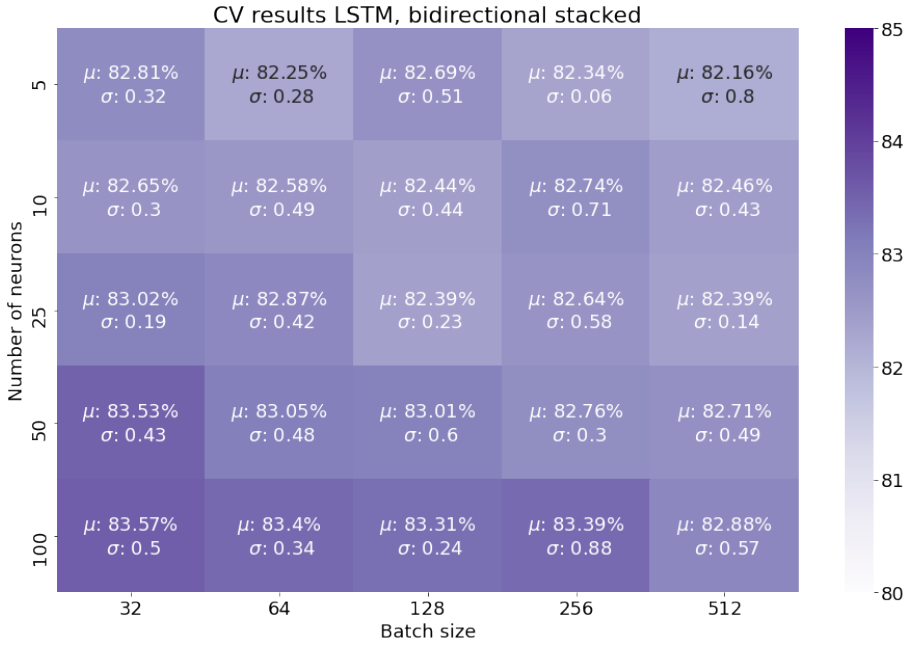


Figure 5.7: CV results of LSTM. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

5.2 Feature Importance

Figure 5.8 shows the 15 most important features in both tree-based models. The x-axis quantifies the information gain, adding op to 100% for all variables. To illustrate how the information gain is determined is in Appendix M the tree structure of the first tree of the final random forest presented. Only the first three layers of the tree are shown in two different visualizations. In the first figure, the data distribution is shown for the feature selected to split the data. The second figure provides more information, including the information gain of the split. The information gain is reported as entropy and can be summed for each feature. These values can be divided by the information gain of the entire tree, resulting in the percentages presented in Figure 5.8.

It can be observed that XGBoost uses more the same powerful features than random forest. Logically, because a random forest does not have all variables available at each split and can therefore not always use the most powerful features. That both models give more importance to more or less the same features, endorses that these features contain the most predictive power. Twelve of the top fifteen are the same for both models. As expected, the most correlated features to the soil type from Figure 3.2 are the most powerful features too.

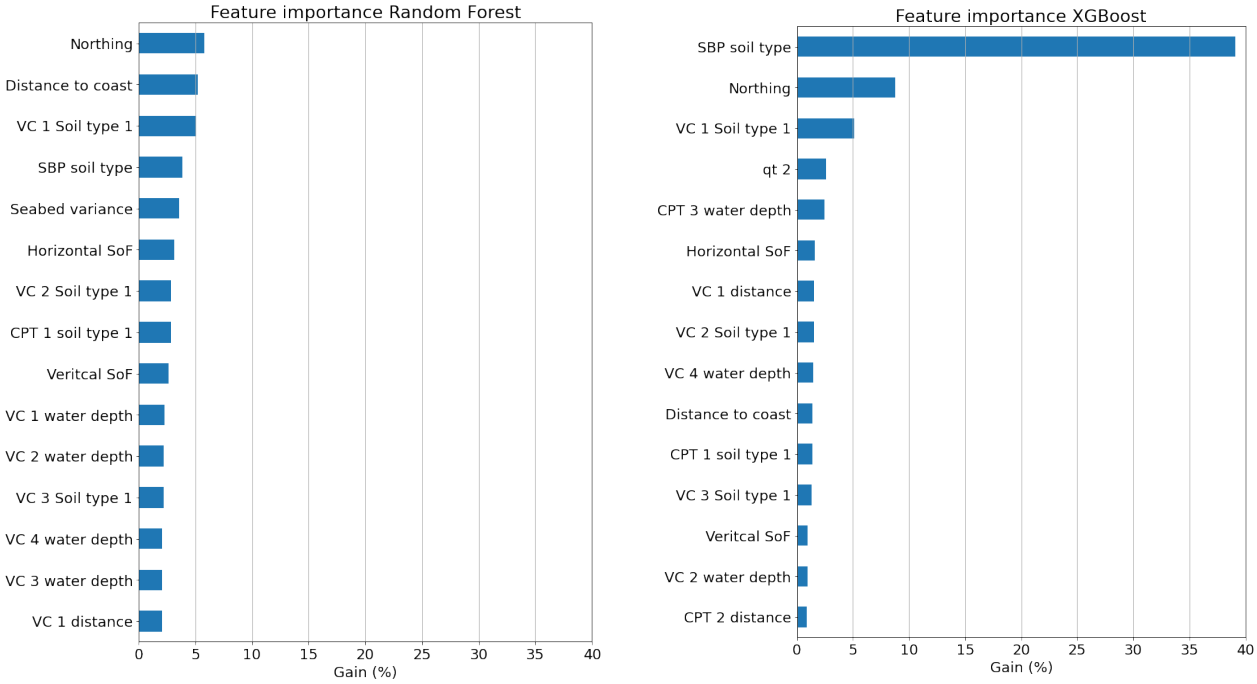


Figure 5.8: Feature importances of both tree-based models. Only the top 15 most important features are depicted.

5.3 Final Models

A test set has been withheld at the beginning of the research to test the robustness of the final selected models. Testing the models on this test set gives an indication of how well the models will perform on new data. If all preprocessing steps have been performed correctly, the results are expected to be approximately the same as the CV results. After all, the ML models have already been tested five times on unseen data with CV.

Triangulated Irregular Network

After creating the network of triangles between the GT locations of the training set, a soil model can be developed by interpolating the soil types between the locations. A section of the network is shown in Figure 5.9. Herein, the red dots represent the test set, and the green dots represent the training set. The network is created on the basis of the training set, and it can be observed that most of the red dots are mainly on the borders of the triangles. Hence, a GT from the test set can be interpolated by the two endpoints of the associated line. The left-hand side of Figure 5.10 shows an example of how triangulation is performed, i.e., the soil types are linearly connected. The right-hand side shows another example where it is not straightforward how to interpolate the soil types with TIN. There are two possible ways to interpolate 1) connect the mud layers and 2) connect the sand layers. In this case, an imaginary line is drawn in the middle, and everything left of the line is set equal to the left VC, and everything right of the line is set equal to the right VC. In other words, a GT from the test set is set equal to the

closest VC.

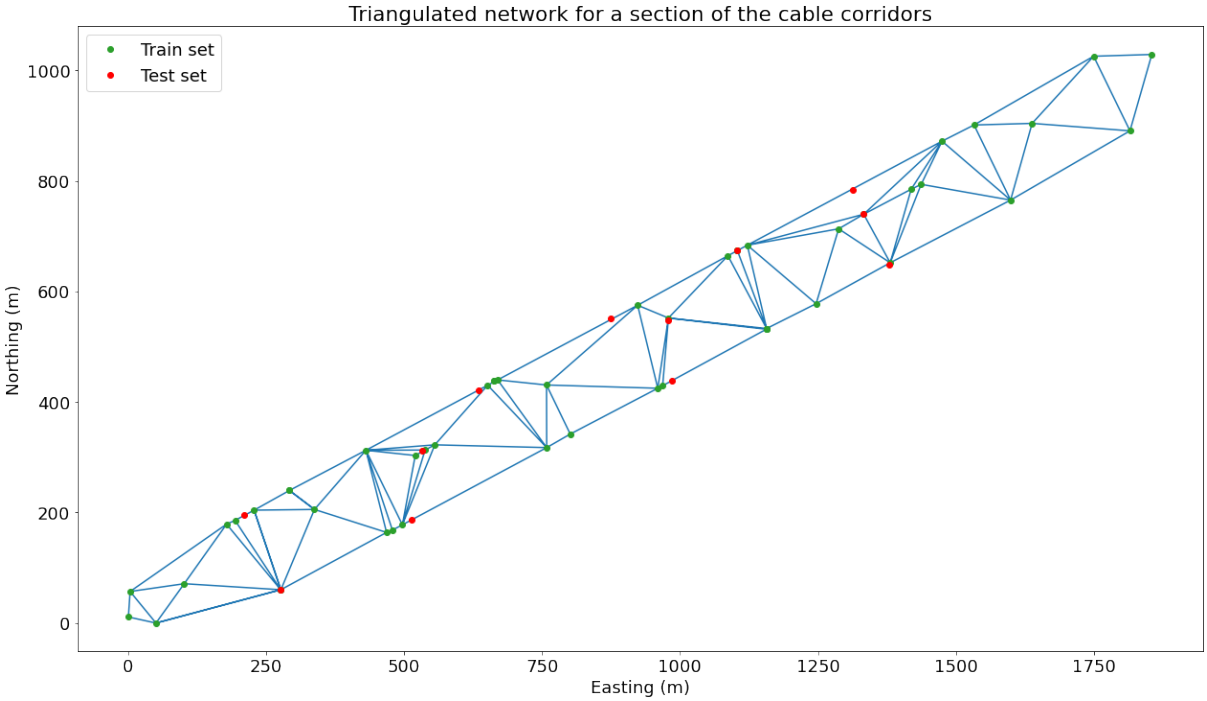


Figure 5.9: Triangulated network based on training. After the network is created the test set is plotted on the figure. It can be observed that the test set is mainly on the borders of the triangles. This simplifies interpolation because only the two endpoints of the line have to be considered instead of all three points of the triangle.

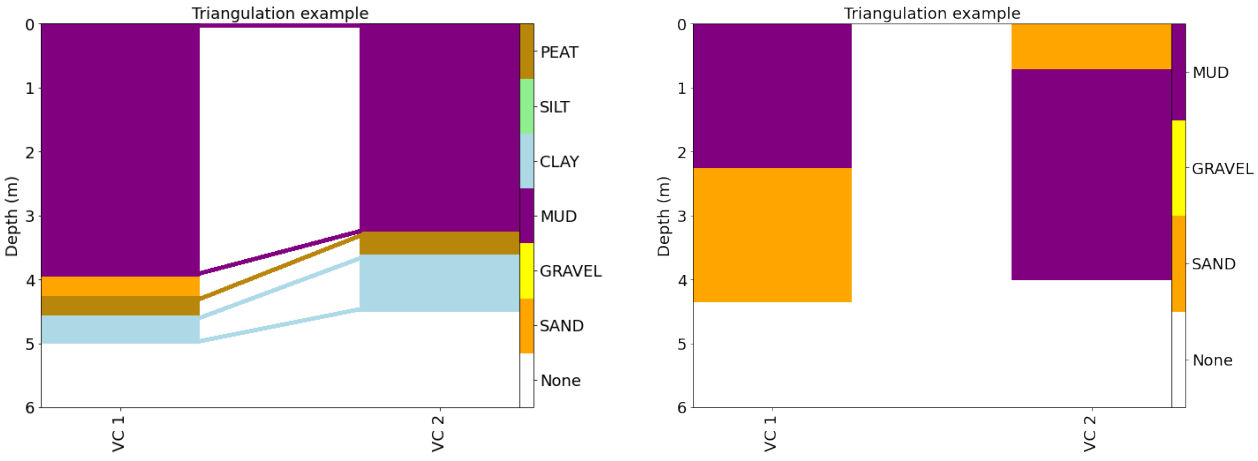


Figure 5.10: Interpolation examples of triangulation. The left-hand side shows a simple example where the soil layers can easily be connected. The right-hand side, however, it is not straightforward how to interpolate. In this case, everything left of the center is set equal to the left VC and everything right of the center is set equal to the right VC.

The results of triangulation are shown in a confusion matrix in Figure 5.11. The rows represent the actual samples and the columns the predicted samples. Consequently, all values on the diagonal are correctly classified, and the predictions that are off the diagonal are misclassified.

The top values indicate the number of samples, and the number below is the percentage of total actual samples, i.e. the rows. Naturally, the percentages in the rows add up to 100% with an occasional rounding error. The overall accuracy is calculated with equation 4.2 where the samples on the diagonal are divided by the total number of samples, which results in an accuracy of 66.63%

It seems that the model suffers from a difference in soil type interpretation between VCs and CPTs and between the two companies. Especially between mud and clay, a lot of misclassifications are made. There are 6,255 mud samples classified as clay and 6,680 clay samples classified as mud. The cause of this has all to do with the section of the cable corridors where the interpretation of the consecutive GTs is alternate from mud to clay, as was shown in Figure 3.4. In the CPTs, the soil is classified as clay and in the VCs as mud. Triangulation cannot take into account differences in interpretations, meaning that the model will perform poorly in areas where this becomes apparent.

As stated in Section 4.4.5, the soil types sand and gravel; mud and clay; and peat and dy can be aggregated. This means that misclassifications between those types can be considered as correctly classified. The concerning misclassifications are indicated by the red squares in the confusion matrix. Aggregation of these soil types improves the accuracy substantially to 78.73%, which is the benchmark for the more complex models. The new confusion matrix is shown in Figure 5.12. Despite the fairly high accuracy, only four GTs are predicted to be excavated, of which only two are correct. A rather poor performance in classifying peat. TIN is a deterministic model with no (hyper)parameters that can be tuned. Therefore it is not possible to predict more peat.

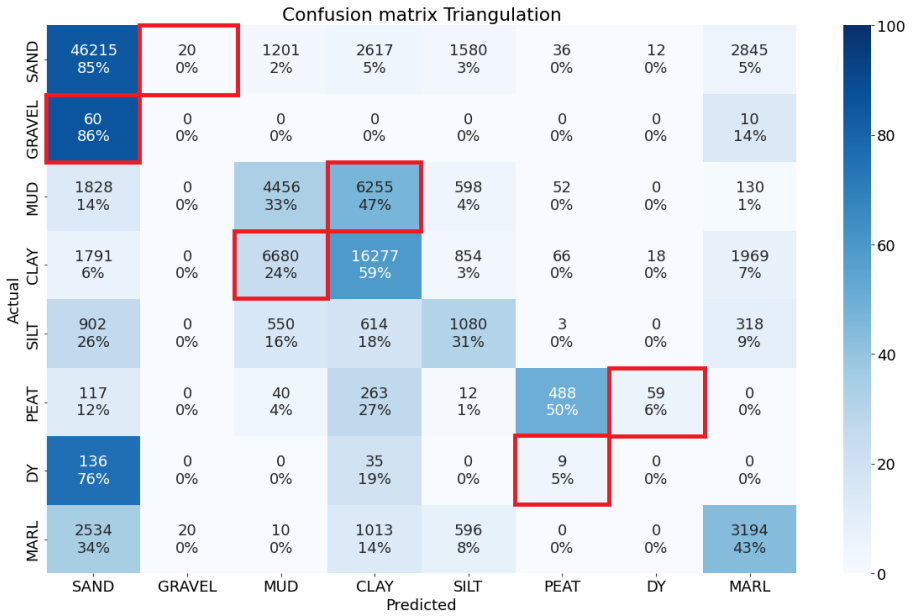


Figure 5.11: Confusion matrix of triangulation before aggregation. The top values indicate the number of samples. The number below is the percentage of total actual samples, i.e. the rows. It can be observed that the model suffers from a difference in interpretation between the test types. The red squares indicate the misclassifications that are correctly classified after aggregation of the soil types.

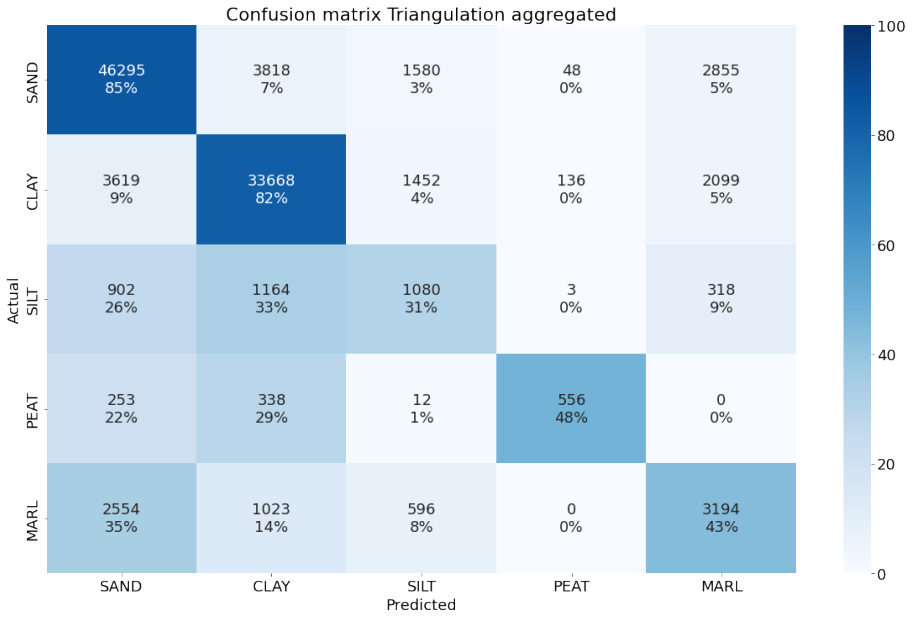


Figure 5.12: Confusion matrix of triangulation after soil type aggregation. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

Kriging

Since kriging cannot handle categorical variables well, the output must be transformed so that kriging can do a regression. For each soil type, a separate regression is performed. For example,

if a regression is performed for sand, the input is the location [x, y, z], and if there is sand on that location, the output is 1 and otherwise 0. In that way, a regression is performed for each soil type. When predicting the test set, it results in eight predictions for each location. These predictions are not bounded to 0 and 1 and can also exceed those values. The largest value is taken as the final prediction, which can be thought of as a type of indicator kriging.

The confusion matrix of kriging is presented in Figure 5.13 with an accuracy of 81.71%. The minority classes are barely predicted. This can be explained by the applied approach because a lot of majority classes often surrounds the minority classes. For example, when predicting an actual peat sample where a majority class is much more dominant in that area, a higher prediction value is assigned to that majority class. Correspondingly, kriging performs poorly at locating peat and even predicts no peat at all.

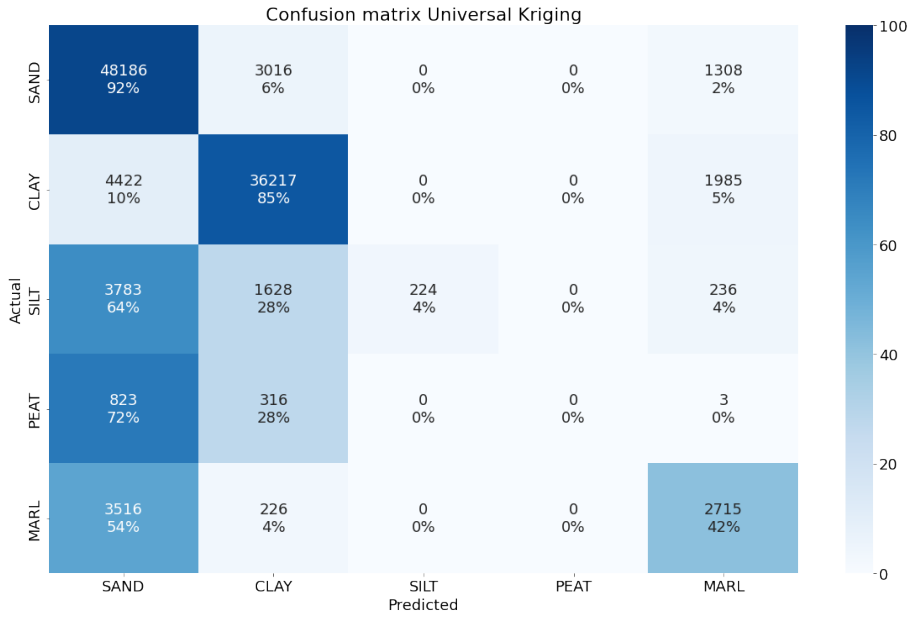


Figure 5.13: Confusion matrix of kriging. The top values indicate the number of samples. The number below is the percentage of total actual samples, i.e. the rows.

Peat Classifier

The peat classifier predicted 17 GTs to be excavated, of which 10 actually need to be excavated. The 7 GTs that are predicted to be excavated and should not be excavated are shown in Figure 5.14. The actual profiles are shown on the left, and on the right are the predicted profiles. It can be noticed that 4 out of 7 incorrect predicted GTs contain peat. However, this peat is lower than 3 meters or too little to excavate. The third and fourth GT contain very organic clay, and the last GT does not contain organic content. Moreover, the 10 out of 13 correctly predicted GTs to be excavated is approximately the 80% which was used as criterion during CV. In Appendix N the correctly and missed soil profiles to be excavated are presented.

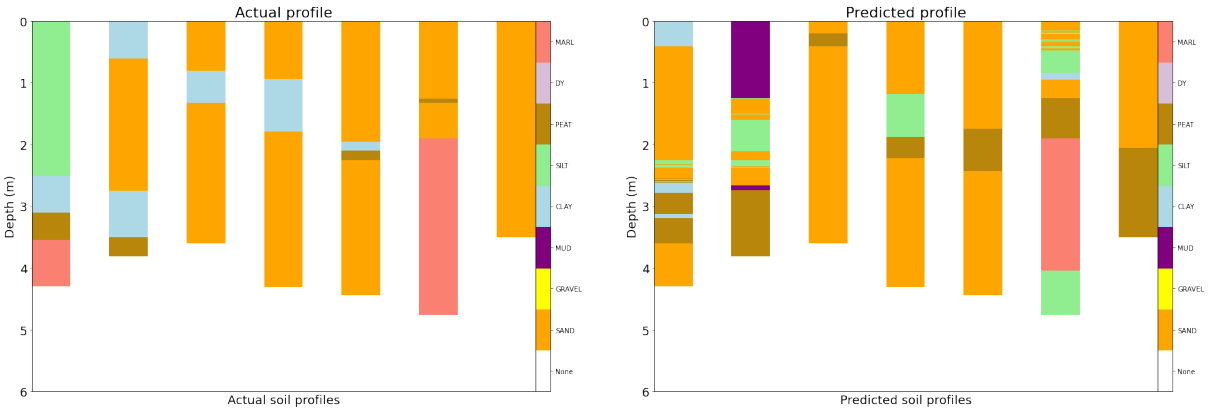


Figure 5.14: Seven incorrect predicted soil profiles to be excavated. The left figures shows the actual soil profiles and the right figure the predicted soil profiles. Four of those GTs contain peat below 3 meter or too little to excavate. The third and fourth GT contain very organic clay.

Random Forest

The final prediction is shown in the confusion matrix of Figure 5.15 with a total score of 85.44%. It can be noticed that the model has a bias towards the majority classes because these are predicted the best with a correct classification of 93% and 90% for sand and clay, respectively.

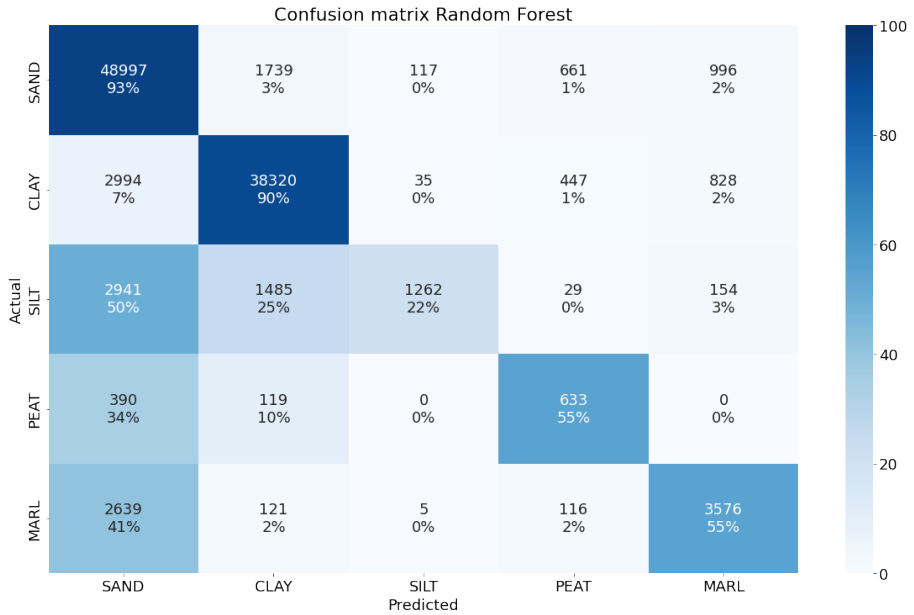


Figure 5.15: Confusion matrix random forest. The top values indicate the number of samples. The number below is the percentage of total actual samples, i.e. the rows.

XGBoost

The final prediction of XGBoost is shown in the confusion matrix of Figure 5.16, where the final score is 85.11%. Comparing the confusion matrix with random forest, it can be seen

that random forest only has a higher accuracy for sand and marl, two of the majority classes. Recalling the theory of the models, it can be explained that XGBoost gives more weight to the samples that are difficult to learn. Consequently, it can better predict these classes, which comes at the expense of the majority class.

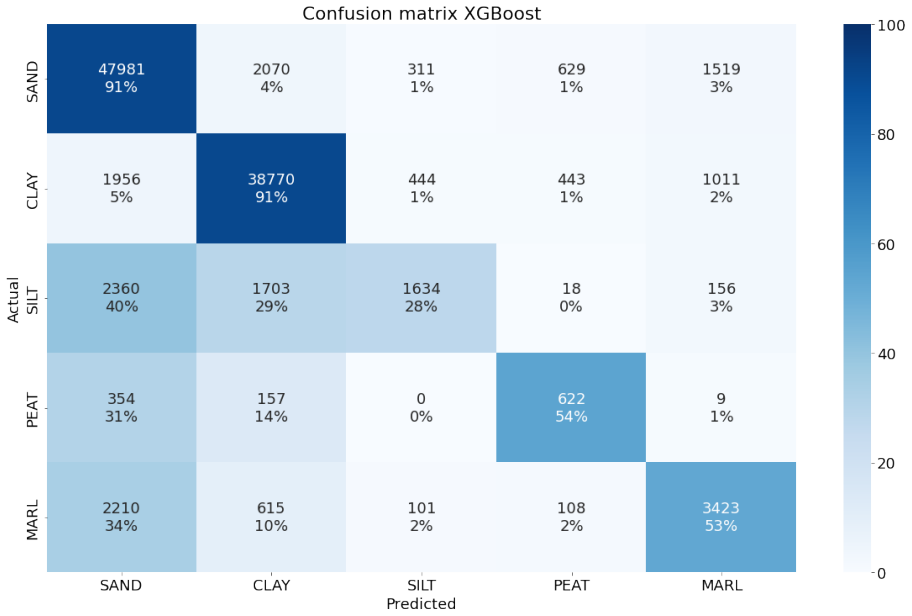


Figure 5.16: Confusion matrix XGBoost. The top values indicate the number of samples. The number below is the percentage of total actual samples, i.e. the rows.

Long Short-Term Memory Network

Figure 5.17 shows the training history of the final model. It can be noticed that accuracy did not improve much after one epoch. Due to "early stopping", the model stopped after six epochs because the loss did not decrease in the last five epochs. It is common to use the loss for early stopping because the loss quantifies how confident the model is about its predictions. The final predictions are presented in the confusion matrix of Figure 5.18. LSTM needs a lot of data and is therefore very sensitive to data imbalance. However, the results show the opposite because the model predicts silt, peat and marl the best. It should be noted that peat has already been corrected by the peat classifier.

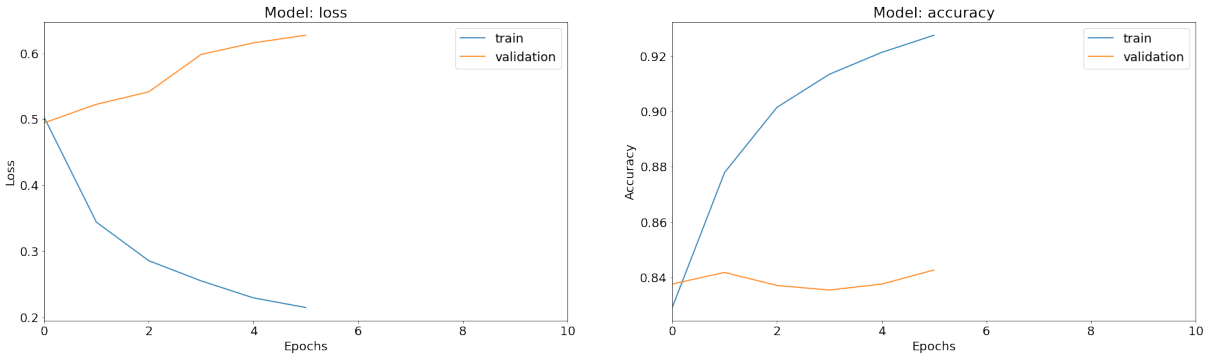


Figure 5.17: Training history LSTM. It can be observed that the model directly starts overfitting after the first epoch. The training is early stopped due to an increasing loss in the last five epochs.

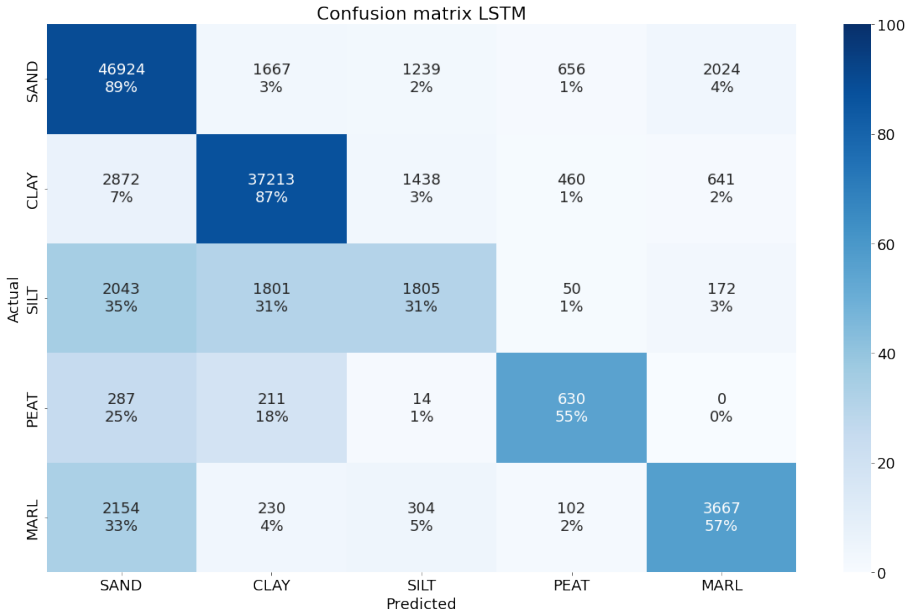


Figure 5.18: Confusion matrix LSTM. The top values indicate the number of samples. The number below is the percentage of total actual samples, i.e. the rows.

Final Results

The results of the final models on the test set are presented in Table 5.1. All complex models have a better overall performance as triangulation which was used as a benchmark. Although XGBoost has the best CV performance, random forest has the best result on the test set. The difference is small, and if tested again, it could be the other way around due to the randomness. Obviously, a seed can be used to generate the same output, but the essence is how the model performs on new data, which could be the other way around. It can also be observed that the performance for all three ML models is better on the test set than during CV. This can be explained by the fact that during CV, the models are trained on only four folds and with the final evaluation on all training data. Implying that there is a better information coverage. To

verify whether this assumption is correct, the best model is trained on 90% of the data and evaluated on the remaining 10%. The train-test split is made in the same way as in Section 4.2 but then with every 10^{th} GT in the test set instead of every 5^{th}. The result endorses the assumption because the accuracy increased to 87.10%. The peat classifier is used for all ML models, and therefore they have the same results in the last three columns.

3D Stratigraphic Soil Model

The best performance on the test set is obtained by random forest. This model is therefore used for the 3D stratigraphic soil model. Appendix P presents a section of the 3D model. Because it is difficult to display a 3D model in a report, the model is shown as 3 separate cable corridors. This is convenient because the cables are parallel. On purpose, a section with high variability is displayed. The model predicted every 5 meters along each cable a soil profile up to 4 meters depth. The real soil profiles show VC or CPT at the top, and the other soil profiles are predicted by the model. The real profiles also stand out because they have different lengths. The entire profile is rather smooth and has a realistic appearance. The difference in interpretation between the companies is also visible here, i.e. the mud VCs.

A remarkable part in cable-3, around 240 to 260 meters, shows marl as soil type. However, there is no marl at all in cable-3 nor detected by the sub-bottom profilers. The prediction of marl is most likely related to the VC in cable-2, which is 100 meters north of cable-3. The probabilities of these samples are approximately 30%, which is low and almost equal to the probability of sand. This indicates that the model is not confident about this area.

Table 5.1: Final results

	CV results (%)	**Accuracy (%)**	**Predicted GTs to be excavated**	**Correct GTs to be excavated**	**Missed GTs to be excavated**
Triangulation	-	78.73	4	2	11
Kriging	-	81.71	**0**	0	13
Random Forest	84.31	**85.44**	17	**10**	**3**
XGBoost	**84.62**	85.11	17	**10**	**3**
LSTM	83.57	84.27	17	**10**	**3**

5.4 Result Analysis

This section analyzes the results of the best model. The best performing model is random forest, although the differences between the models are quite small. In Table 5.1 is the accuracy calculated over all data, but for this project, only the top 3 meters are of interest. In Figure 5.19, the accuracy of the model against the cumulative depth is plotted. Cumulative depth

means that everything is included up to a certain depth, i.e. at 3 meters, the top 3 meters are included, and at 4 meters, the top 4 meters are included. Here it is visible that the accuracy decreases with depth. Given that only the first 3 meters are of interest, the accuracy increases to 87.26%. The GTs to be excavated can also be excluded from the evaluation, as the peat classifier edits them and, moreover, are not important for accuracy because they are excavated and backfilled with clean sand. By excluding these tests, the accuracy increases to 88.47%.

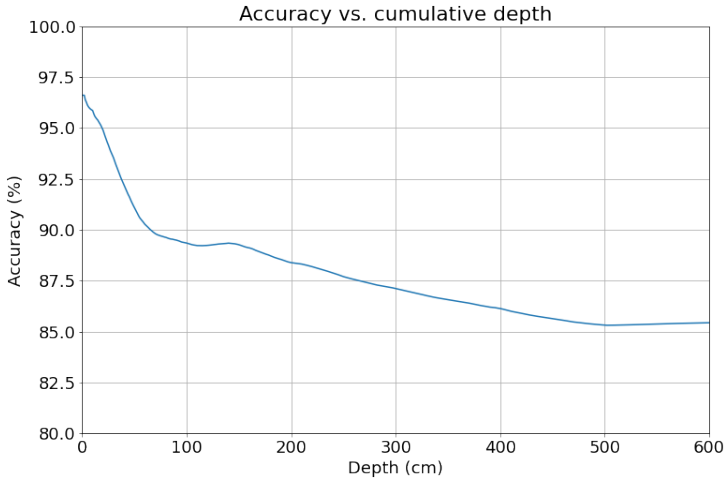


Figure 5.19: Accuracy vs. cumulative depth.This figure shows the accuracy in predicting to a certain depth. So is the prediction accuracy for the top 3 meter 87.10%.

The dependency of accuracy on different variables is presented in Figure 5.20. As mentioned above, the accuracy decreases with depth, but it sharply increases in the range of 5 meters to 6 meters. The reason for that, only 12 GTs are reaching this depth and have a low variability at large depth. Figure A.22 in Appendix O shows the low variability of these GTs at large depth, including the predicted profiles. Figure 5.20b shows that the accuracy increases with an increasing mean distance to nearby GTs. This may seem counter-intuitive, but it is not because few GTs have been performed in areas with low variability, i.e. high degree of certainty, thus achieving a high degree of accuracy. The reverse also applies, many GTs have been performed in areas with high variability and high uncertainty, resulting in a lower accuracy. Figure 5.20c shows a similar pattern as Figure 5.20a, a decreasing accuracy with an increasing vertical SoF whereafter it increases again. The reason is also quite similar as there are only a few GTs, and therefore the results fluctuate and can increase. There are only 5 GTs with a vertical SoF greater than or equal to 6. These GTs are largely well predicted due to a dominant soil type in the GTs. This is shown in Figure A.23 in Appendix O. Figure 5.20d exhibits not this pattern because the horizontal SoF can change with depth in a GT and the vertical SoF not. Consequently, there is no dominant soil type with a high horizontal SoF, which negatively affects accuracy.

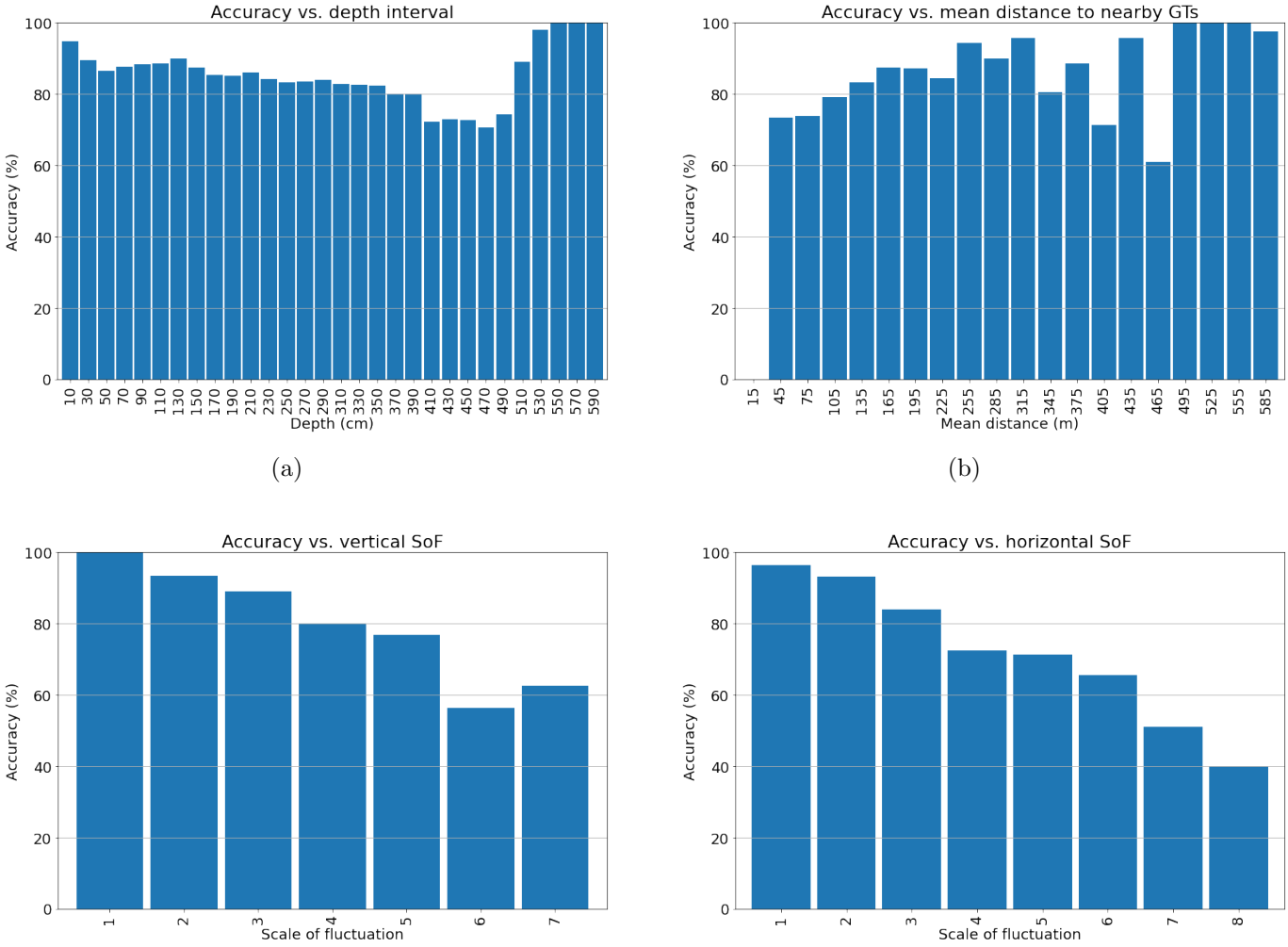


Figure 5.20: These graphs show how the accuracy is related to the features.

Certainty

A major advantage of machine learning over conventional models is that it assigns probabilities to its predictions. These probabilities give an indication of how confident the model is about its prediction. In practice, this can be very useful when a certain level of risk has to be met. For instance, geotechnical testing can be performed on locations where the model has little certainty to ensure a constant certainty. In Figure 5.21, the accuracy is plotted against the probability of the outcomes of the model. Here it can be observed that when the model assigns a probability greater than 50% to a single class, the prediction is correct 9 times out of 10. The graph also tells that it is always correct when the model assigns a probability of 100% to a class. However, this does not always have to be the case on new data.

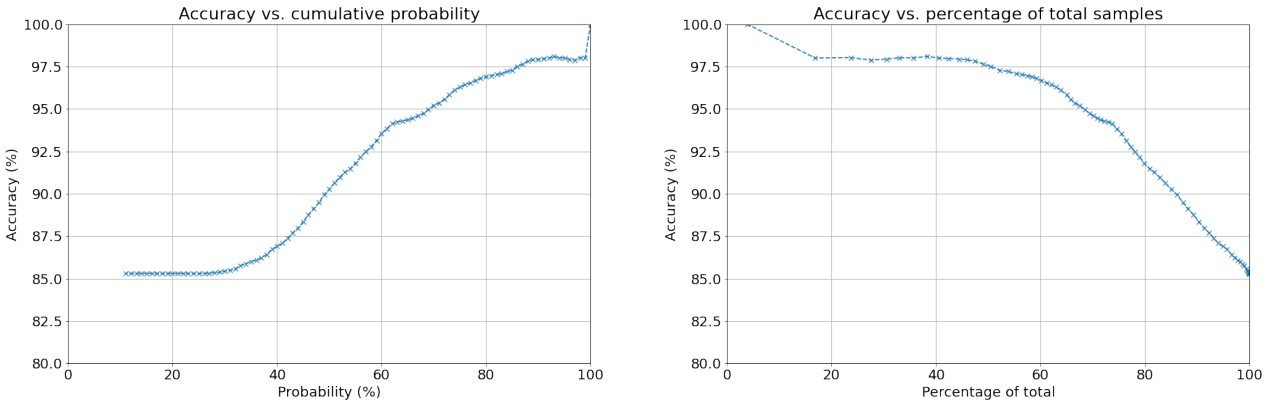


Figure 5.21: The left figure shows the accuracy if only the predictions above a certain probability are included. A prediction need at least a probability of 12.5% to be the final prediction if there are 8 soil types. Therefore the graph starts at a probability of 12.5%. The right figure shows the accuracy versus the percentage of the total data. Here can be read which percentage of the data can be predicted with which accuracy.

The right figure presents the accuracy versus the percentage of all test observations. This graph shows that 67.44% of the data can be predicted with an accuracy of 95% and 86.15% with an accuracy of 90%. These values are also shown in Table 5.2. This table shows what percentage of the data can be correctly predicted if a certain accuracy is met. This is shown for all data, data without the excavated GTs, data for the top 3 meters and for the top 3 meters without excavated GTs.

Table 5.2: Percentage of data vs. accuracy

	Percentage of total with predicted accuracy >= 90%	**Percentage of total with predicted accuracy** >= 95%
All data	86.15	67.44
Without excavated GTs	89.63	74.97
Top 3 meter	89.94	70.98
Top 3 meter & without excavated GTs	95.17	79.12

Chapter 6

Discussion & Conclusion

This final chapter discusses the results, answers the research questions and gives recommendations for future research.

6.1 Discussion

In this section the interpretation of the results is given, with extra attention to the introduced model. Additionally, the implications and limitations of this study are addressed.

Interpretations

The academic aim of this thesis was to evaluate the performance of conventional models against state-of-the-art techniques and to a newly proposed model in DSM of soil types. Considering the results of the models, it can be concluded that all ML models outperform the conventional models. The overall performance of the ML models is significantly better than that of the conventional models, but the difference is even more apparent in the classification of peat. The conventional models performed poorly in the classification of peat, and kriging did not predict peat at all. The peat classifier, with a random forest as the underlying model, correctly predicted 77% of the GTs to be excavated, which is close to the 80% for which it was optimized.

Regarding the overall performance, it can be debated whether the final accuracy of the models are the same as the accuracy of the final 3D model. This is because the GTs are performed at a planned distance from the each other. Therefore, it is likely that the GTs in the test set are the furthest locations from the GTs in the training set, which decreases the available information and certainty on those locations. Presumably, the locations close to the GTs in the training set can be predicted better than the locations further away. However, this cannot be quantified and remains a conjecture.

On the other hand, what can be assumed is that the model performs better with more data, which explains the better result on the test set compared to the CV results. This was verified by training the model on 90% of the data and testing it on 10%, whereby the accuracy increased to 87.26%. For typical machine learning problems, this is not necessarily the case.

However, for this problem, it makes sense because it is a closed area, and by training on more GTs, the model has 100% knowledge on the locations of the GTs, and more importantly, better information coverage over the entire area as the distances to uncertain areas decreases. Where in the limit a GT has been performed at each location in the area, and the model has 100% knowledge on each location when trained on this. The random forest that produced the final 3D digital soil map presented in Appendix P was trained on the training and test set. Thus, on reasonable grounds, the overall accuracy is believed to be greater than 87.26%. One way to find out the accuracy of the final model is to calibrate the models. This procedure ensures that all predictions with 90% probability also have an accuracy of 90% and that all predictions with 80% probability have an accuracy of 80%. After calibration, the accuracy of the final model can be estimated by calculating the mean probability of the predictions, which can be related to the accuracy. Calibrating the model comes at the expense of the model's performance, hence the model was not calibrated.

LSTM interpretation

LSTM, the introduced model, shows a good performance in relation to the other methods. Although it is not the best performing model, there are still plenty of opportunities for this model to improve its performance. Even for this project, there is expected to be quite some room for improvement. Especially how the data is presented to the model. First, the data provided to the LSTM is based on only the two nearest VCs and CPTs, while the machine learning models base their features on the nearest four. Finding a way to present the information to the model without increasing the dimensionality too much could improve the results. Low dimensionality is vital for the model to recognize patterns in the data. Another option that might substantially affect the performance is the sequence length. This study used a sequence length of 12, meaning averaging every 50 centimeters. This can be modified to any length. Longer sequences, i.e. smaller averaging distances, contain more information than shorter sequences. It is expected that there is an optimal sequence length for this problem as a longer sequence will also contain more noise.

Secondly, the current model directly overfits the training data, shown in Figure 5.17. Regularizing this in a better way could improve the performance significantly. Furthermore, the architecture of the LSTM can be modified in multiple ways. For instance, the number of neurons can be increased and additional layers can be added such as a dense, convolutional or LSTM layer. Since there is little literature about when which architecture works well, it requires a lot of experience from the implementer to arrive at a good architecture.

Besides, it is also possible to modify the method of the LSTM by defining the sequences in a horizontal direction instead of vertical. However, this is a challenging task and similar to determining the horizontal SoF. Therefore the same problem arises that the GTs provide enough data in the vertical direction but not in the horizontal direction. A way to implement this is to perform predictions along a single cable corridor so that the GTs are organized in a line.

Then, at each depth, the soil properties of the GTs can be used as a sequence. Additionally, the distance between the GTs should be included because it is not equidistant as the model assumes.

Finally, the design of LSTM can be modified to predict entire sequences instead of a single value. These sequences can then represent an entire soil profile in the vertical direction or a soil layer in the horizontal direction. Considering the accuracy already obtained, which is slightly lower than the best model, and the potential for improvements, it can be concluded that LSTM is suitable for DSM and has potential to improving DSM of soil types.

Limitations

The presented results show that the ML models outperform the conventional models. This does not imply that the conventional models are from now on superfluous as this has only been tested on a single project. Besides, triangulation is implemented without knowledge of the area and uses only two GTs for interpolation following specified rules. The triangulation method applied by Boskalis was performed by a geotechnical expert who did not adhere to these rules but used experience and knowledge of the area for interpolation. Therefore, the presented triangulation methodology cannot be compared one to one with the triangulation methodology used in the project. The same applies to kriging, where the geotechnical expert first identified the soil layers before interpolation.

Furthermore, different approaches could have been used for kriging. Naturally, kriging is not a classification model but a regression model. This study uses a combination of universal kriging and indicator kriging, whereby the largest outcome is used as end prediction. A better option might be to do a regression on the soil behaviour type index from Robertson (1990) shown in Table 2.2. Then a single value can be regressed, which is more convenient for kriging. Another option is to interpolate the depths of the soil layers. Then, a single value can also be regressed in space. Although, this option encounters the same problem as with triangulation, where it might occur that it is not straightforward how to interpolate as in Figure 5.10. Applying this option means that the geotechnical engineer must first determine the soil layers before interpolation.

Lastly, other types of kriging can be used that might perform better. It is unfortunate that not more kriging methods are available in ArcGIS Pro, such as regression kriging, which is capable of using additional features besides the spatial position. The prediction is now based only on the spatial position, which is expected to be the main shortcoming of the used kriging method.

Implications

As indicated in Limitations, the conventional models are not exactly the same as the models applied by Boskalis. The models applied by Boskalis were developed with the interference of a geotechnical expert, while the models in this study were not. Nevertheless, this comparison indicates the basic performance of the models without a geotechnical expert. This can be used

to determine the value of the models with a geotechnical expert's interference. The value of a model should not only depend on its accuracy, but its interpretability and the possibility of interference should also be taken into account.

Besides, the best model can assist the geotechnical engineer in determining the soil stratification in the project area. Generally, in a cable burial project, DSM of soil types takes a lot of time by cleaning the data, identifying the soil layers and then interpolating these layers. The presented approach can facilitate this process because the steps taken are easy to automate, and the developed model provides certainty indications. Possible applications of the model are to verify the existing models, assist in identifying soil layers, locating peat or identify locations in the project area with high uncertainty. In addition, the ML models can be used to satisfy a certain degree of risk by using only predictions with a probability corresponding to the degree of risk or indicating where additional soil investigation* is needed to increase the certainty.

Currently, the GTs are performed at a fixed interval along the cable route. This is obviously not optimal, as it is desirable to perform more GTs in areas with a high variability than in areas with low variability. Adapting the ML models to optimally predict where to perform the GTs, given a maximum uncertainty in the model, can reduce the total number of GTs. As it can be predicted where the variable areas are and what the certainty of the model is with the GTs already performed.

It is accurate to say that with the current data developments, machine learning will become increasingly important in this field and might eventually take over this work entirely without human intervention. Thereby, it is important to make more use of machine learning to deliver more consistent work and make it more efficient. So is human intervention susceptible to subjectivity and human error, which is not the case with machines.

6.2 Answer to Research Questions

This section provides answers to the formulated research questions in Section 1.5.

How do the machine and deep learning models perform compared to the conventional models in the digital mapping of soil types?

This question has already been mainly answered in Interpretations and can therefore be answered briefly. All state-of-the-art models outperform the conventional methods. The main reason is that the conventional models are not suitable for predicting categorical variables and cannot incorporate additional features for prediction besides the spatial position.

*Soil investigation does not necessarily mean collecting additional data in the field, but can also be a desk study.

Can the implemented models assist in determining which sites contain peat and need to be excavated?

A shortcoming of conventional models is that they are deterministic and cannot give more weight to minority classes. The flexibility of ML models can provide a solution to predict the peat samples more accurate. A separate peat classifier was designed for this task. The peat classifier is optimized with the criterion of correctly predicting at least 80% of the GTs that should be excavated. The final outcome of the model correctly predicted 10 out of 13 GTs that should be excavated, resulting in a score of 77%, which is close to the 80% for which it was optimized. The model predicted seven other GTs to be excavated that, in reality, should not have been excavated. Four of those seven contained peat but too little or too deep so that it did not have to be excavated. Two other GTs contained very organic clay, and the last one did not contain organic material. This result underscores that the peat classifier performs well at locating organic soil but still has it difficult to distinguish when to excavate or not.

The difference with the conventional models is significant. The conventional models performed poorly in the classification of peat, and kriging did not predict peat at all. Triangulation predicted 4 GTs to be excavated, of which 2 were correct. It can be stated that the peat classifier performed properly by being close to the criterion where it was optimized for. Whether the model also performs well with a different criterion should be checked. For example, at least 90% or perhaps even higher, but in practice, this percentage should be determined by the industry. Even if the desired percentage is not achieved, the peat classifier can assist in determining which locations contain peat and need to be excavated.

Are conventional spatial interpolation techniques the best way for digital soil mapping of soil types, or does machine learning offer an opportunity for improvement

The ML models performed well in this study and better than the conventional models. Especially, predicting the locations to be excavated is difficult for the conventional models. The flexibility of machine learning by using resampling techniques and designing a separate classifier for this task resulted in a good performance.

The main reason is that the conventional models are not suitable for predicting categorical variables and cannot incorporate additional features for prediction besides the spatial position. On the other hand, the models are interpreted here without an expert, which is the case in practice for the conventional models. Therefore it cannot be compared one to one. Nevertheless, the difference between the conventional and the ML models is significant, highlighting the potential of ML in DSM. Herein, the true power of the ML models is that they predict probabilities. These probabilities also showed to be crucial in peat classification and can be used for practical purposes as described in Implications.

6.3 Future Research

The performance of the ML models mainly depends on how the data is presented to the model. It is up to the engineer to extract, pre-process and structure the data in the best way. The expectation is that there are still many opportunities for improvement. For instance, not all available information is included due to the abundance of data, meaning there is potential for improvement when providing more information to the model. So were the laboratory tests not included yet, which contain a lot more detailed information. However, incorporating this can be challenging because it is not performed on every VC. What presumably contains even more predictive power is the text describing the soil type interpretations of the GTs. This research included only the primary and secondary soil types in the features, but it contains a lot more information. The description includes information such as colour, stiffness, if the soil is calciferous and loose or dense bedded and many more characteristics.

The approach used in this study really defines the soil types as distinct classes. While the reality is that it is more gradual. Here, sandy clay is entirely different from clay while it can have practically almost the same soil composition according to texture triangle in Appendix A. Obviously, a classification has to be made, but now it is one soil type or the other. By defining soil types as data points with two numerical values or more, it can be indicated that some soil types have more in common than others. A way to implement this is by means of an embedding layer in a neural network. This layer can transform categorical variables into vectors or latent variables. These vectors can be very similar or different from each other, which should reflect the similarity between soil types.

That soil types are not completely different from each other brings up the next topic, soil type interpretation. As was shown in Figures 3.3 and 3.4, the same soil layer is interpreted differently, which is difficult for the model. Therefore it is expected that the performance can be improved by data cleaning, handling soil type interpretation by different companies and different tests, and providing more information to the model. Also, different projects can be beneficial to allow the model to get a better understanding of the problem.

References

ArcGIS. (n.d.-a). *How kriging works.* Retrieved from https://pro.arcgis.com/en/pro-app/latest/tool-reference/3d-analyst/how-kriging-works.htm

ArcGIS. (n.d.-b). *What is a tin surface?* Retrieved from https://desktop.arcgis.com/en/arcmap/10.3/manage-data/tin/fundamentals-of-tin-surfaces.htm

Barrette, P. (2011, 10). Offshore pipeline protection against seabed gouging by ice: An overview. *Cold Regions Science and Technology - COLD REG SCI TECHNOL*, *69*. doi: 10.1016/j.coldregions.2011.06.007

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2019, 11). A comparative analysis of xgboost.

Bishop, C. (2006, 01). Pattern recognition and machine learning. In (Vol. 16, p. 140-155). doi: 10.1117/1.2819119

Cami, B., Javankhoshdel, S., Phoon, K.-K., & Ching, J. (2020, 07). Scale of fluctuation for spatially varying soils: Estimation methods and values. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems Part A Civil Engineering*. doi: 10.1061/AJRUA6.0001083

Cao, Z.-J., Zheng, S., Li, D.-Q., & Phoon, K.-K. (2019). Bayesian identification of soil stratigraphy based on soil behaviour type index. *Canadian Geotechnical Journal*, *56*(4), 570-586. Retrieved from https://doi.org/10.1139/cgj-2017-0714 doi: 10.1139/cgj-2017-0714

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002, 06). Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, *16*, 321-357. doi: 10.1613/jair.953

Ching, J., Wu, T.-J., Stuedlein, A., & Bong, T. (2017, 12). Estimating horizontal scale of fluctuation with limited cpt soundings. *Geoscience Frontiers*, *9*. doi: 10.1016/j.gsf.2017.11.008

Cressie, N. (1990). The origins of kriging. *Mathematical Geology*, *22*, 239-252.

Dobos, E., Carré, F., Hengl, T., Reuter, H., & Tóth, G. (2006). Digital soil mapping as a support to production of functional maps..

Ecker, M. D. (2021). Geostatistics: Past, present and future. *Environmetrics*.

Eea coastline for analysis. (2015). Retrieved from `https://www.eea.europa.eu/data-and-maps/data/eea-coastline-for-analysis-1`

Ferrat, L. A., Goodfellow, M., & Terry, J. R. (2018). Classifying dynamic transitions in high dimensional neural mass models: A random forest approach. *PLoS Computational Biology*, *14*.

Gast, T., Vardon, P., & Hicks, M. (2018, 06). Detection of soil variability using cpts..

Goovaerts, P. (1997, 01). Geostatistics for natural resource evaluation. In (Vol. 42).

Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks* (Vol. 385). doi: 10.1007/978-3-642-24797-2

Hammel, B. D. (2019). *What learning rate should i use?* Retrieved from `http://www.bdhammel.com/learning-rates/`

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies..

Hochreiter, S., & Schmidhuber, J. (1997, 12). Long short-term memory. *Neural computation*, *9*, 1735-80. doi: 10.1162/neco.1997.9.8.1735

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning with applications in r.* Springer.

Kirkwood, C., Cave, M., Beamish, D., Grebby, S., & Ferreira, A. (2016, 05). A machine learning approach to geochemical mapping. *Journal of Geochemical Exploration*, *167*. doi: 10.1016/j.gexplo.2016.05.003

Kunert, R. (2020). *Smote explained for noobs – synthetic minority over-sampling technique line by line.* Retrieved from `https://rikunert.com/smote_explained`

Li, J. (2008). *A review of spatial interpolation methods for environmental scientists.*

Li, J., & Heap, A. (2011, 07). A review of comparative studies of spatial interpolation methods in environmental sciences: Performance and impact factors. *Ecological Informatics*, *6*, 228-241. doi: 10.1016/j.ecoinf.2010.12.003

Li, J., Heap, A., Potter, A., & Daniell, J. (2011, 12). Application of machine learning methods to spatial interpolation of environmental variables. *Environmental Modelling and Software*, *26*. doi: 10.1016/j.envsoft.2011.07.004

Linnane, K. (2019). Review of cable installation, protection, mitigation and habitat recoverability.

Lloret-Cabot, M., Fenton, G., & Hicks, M. (2014, 08). On the estimation of scale of fluctuation in geostatistics. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, *8*. doi: 10.1080/17499518.2013.871189

McBratney, A., Mendonça Santos, M., & Minasny, B. (2003, 11). On digital soil mapping. *Geoderma*, *117*, 3-52. doi: 10.1016/S0016-7061(03)00223-4

Minasny, B., & Mcbratney, A. (2015, 08). Digital soil mapping: A brief history and some lessons. *Geoderma*, *264*. doi: 10.1016/j.geoderma.2015.07.017

Palagi, L., Pesyridis, A., Enrico, S., & Tocci, L. (2018, 10). Machine learning for the prediction of the dynamic behavior of a small scale orc system. *Energy*, *166*. doi: 10.1016/j.energy.2018.10.059

Phoon, K.-K., Ching, J., & Shuku, T. (2021, 03). Challenges in data-driven site characterization. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, 1-13. doi: 10.1080/17499518.2021.1896005

Phoon, K.-K., & Kulhawy, F. H. (1999). Characterization of geotechnical variability. *Canadian Geotechnical Journal*, *36*(4), 612-624. Retrieved from `https://doi.org/10.1139/t99-038` doi: 10.1139/t99-038

Pilario, K. E., Cao, Y., & Shafiee, M. (2020, 05). A kernel design approach to improve kernel subspace identification. *IEEE Transactions on Industrial Electronics*, *PP*, 1-1. doi: 10.1109/TIE.2020.2996142

QGIS. (2020). *Ruimtelijke analyse (interpolatie).* Retrieved from `https://docs.qgis.org/3.4/nl/docs/gentle_gis_introduction/spatial_analysis_interpolation.html`

Rauter, S., & Tschuchnigg, F. (2021, 06). Cpt data interpretation employing different machine learning techniques. *Geosciences (Switzerland)*, *11*, 265. doi: 10.3390/geosciences11070265

Ritzema, H. (1994). *Drainage principles and applications.* International Institute for Land reclamation and Improvement, Wageningen, The Netherlands.

Robertson, P. (1990, 02). Soil classification using the cone penetration test. *Canadian Geotechnical Journal - CAN GEOTECH J*, *27*, 151-158. doi: 10.1139/t90-014

Samui, P., & Thallak, S. (2010, 10). Site characterization model using artificial neural network and kriging. *International Journal of Geomechanics*, *10*, 171-180. doi: 10.1061/(ASCE)1532-3641(2010)10:5(171)

Sauvin, G., Vanneste, M., Vardy, M., Klinkvort, R. T., & Forsberg, C. (2019, 04). Machine learning and quantitative ground models for improving offshore wind site characterization.. doi: 10.4043/29351-MS

Schloeder, C., Zimmermann, N., & Jacobs, M. (2001, 03). Comparison of methods for interpolating soil properties using limited data. *Soil Science Society of America Journal*, *65*. doi: 10.2136/sssaj2001.652470x

scikit learn. (n.d.). *Cross-validation: evaluating estimator performance.* Retrieved from `https://scikit-learn.org/stable/modules/cross_validation.html`

Sekulić, A., Kilibarda, M., Heuvelink, G., Nikolić, M., & Bajat, B. (2020, 05). Random forest spatial interpolation. *Remote Sensing*, *12*. doi: 10.3390/rs12101687

Sharififar, A., Sarmadian, F., Malone, B., & Minasny, B. (2019, 09). Addressing the issue of digital mapping of soil classes with imbalanced class observations. *Geoderma*, *350*, 84-92. doi: 10.1016/j.geoderma.2019.05.016

Taghizadeh, R., Schmidt, K., Eftekhari, K., Behrens, T., Jamshidi, M., Davatgaar, N., ... Scholten, T. (2020, 01). Synthetic resampling strategies and machine learning for digital soil mapping in iran. *European Journal of Soil Science*. doi: 10.1111/ejss.12893

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, *46*(sup1), 234-240. Retrieved from `https://www.tandfonline.com/doi/abs/10.2307/143141` doi: 10.2307/143141

USDA. (n.d.). *United states department of agriculture.* Retrieved from `https://www.wikiwand.com/en/Loam`

van Giersbergen, N. P. A. (2018). *Machine learning for econometrics.* (Lecture)

Verikas, A., Vaiciukynas, E., Gelzinis, A., Parker, J., & Olsson, M. C. (2016, 04). Electromyographic patterns during golf swing: Activation sequence profiling and prediction of shot effectiveness. *Sensors*, *16*, 592. doi: 10.3390/s16040592

Wang, H., Wang, X., & Liang, R. (2020, 02). Study of ai based methods for characterization of geotechnical site investigation data.

Wang, J.-H., Liu, T.-W., Luo, X., & Wang, L. (2018). An lstm approach to short text sentiment classification with word embeddings. In *Rocling/ijclclp.*

Wang, S., & Jiang, J. (2015, 12). Learning natural language inference with lstm.

Wang, Y., Pan, Z., Zheng, J., Qian, L., & Mingtao, L. (2019, 08). A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science*, *364*. doi: 10.1007/s10509-019-3602-4

Wang, Y., Zhu, S., & Li, C. (2019). Research on multistep time series prediction based on lstm. In *2019 3rd international conference on electronic information technology and computer engineering (eitce)* (p. 1155-1159). doi: 10.1109/EITCE47263.2019.9095044

Wyk, F., Khojandi, A., Mohammed, A., Begoli, E., Davis, R., & Kamaleswaran, R. (2018, 12). A minimal set of physiomarkers in high frequency real-time physiological data streams predict adult sepsis onset earlier. *International Journal of Medical Informatics*, *122*. doi: 10.1016/j.ijmedinf.2018.12.002

Yan, S., & Guo, L. (2015, 01). Calculation of scale of fluctuation and variance reduction function. *Transactions of Tianjin University*, *21*, 41-49. doi: 10.1007/s12209-015-2298-y

Yao, L., & Guan, Y. (2018, 12). An improved lstm structure for natural language processing. In (p. 565-569). doi: 10.1109/IICSPI.2018.8690387

Zhang, X., Liang, X., Li, A., Zhang, S., Xu, R., & Wu, B. (2019, 08). At-lstm: An attention-based lstm model for financial time series prediction. *IOP Conference Series: Materials Science and Engineering*, *569*, 052037. doi: 10.1088/1757-899X/569/5/052037

Appendices

A Texture Triangle

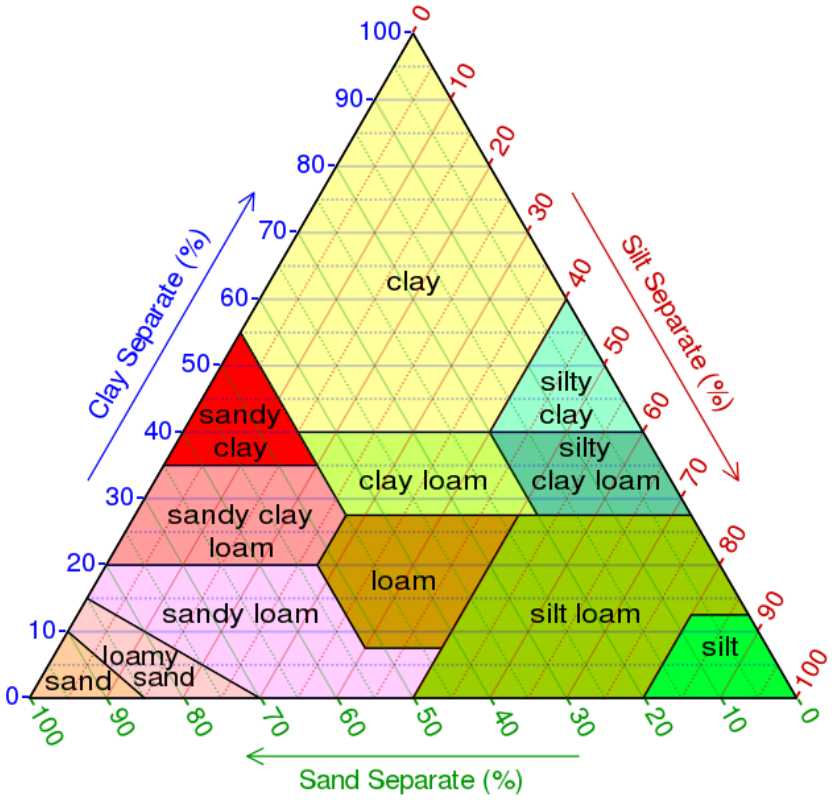


Figure A.1: Soil types by clay, silt and sand composition as used by the United States Department of Agriculture (USDA, n.d.)

B Decision Tree Split

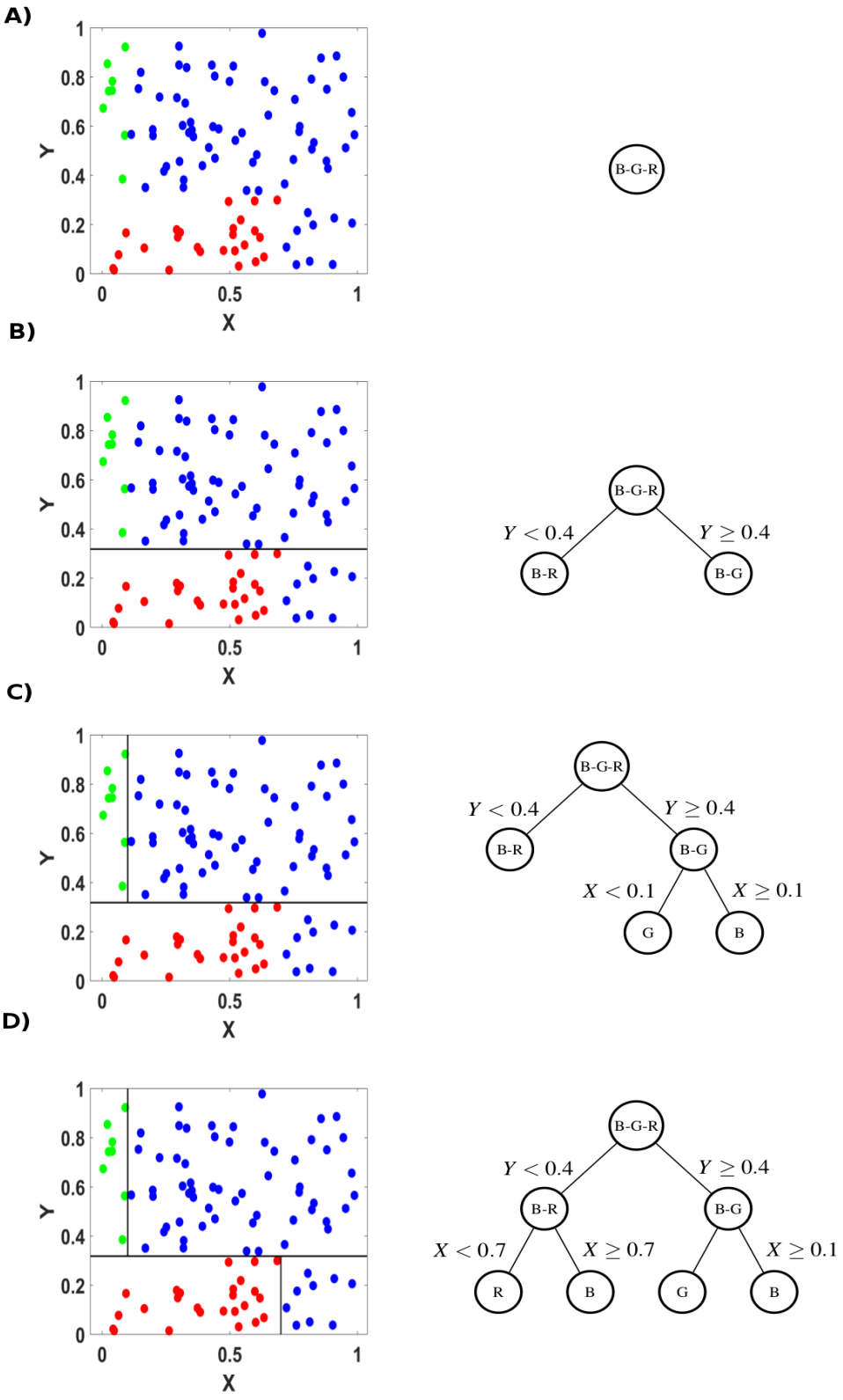


Figure A.2: This figure visualizes the splitting mechanism of a decision tree with a depth of two (Ferrat et al., 2018). A is the starting position without any split, B is the first split and C & D each split a subset from the first split. Without regularizing the tree will grow further until each datapoint is in a seperate leaf. In other words, until each datapoint is isolated.

C Loss Function XGBoost

An example how XGBoost minimizes MSE as a loss function. Equation A.2 can be solved for solved for the tree structure f_t.

$$obj(t) = \sum_{i=1}^{n}(y_i - (\widehat{y_i}^{t-1} + f_t(x_i)))^2 + \sum_{i=1}^{t}\Omega(f_i) \tag{A.1}$$

$$\frac{\partial obj(t)}{\partial \widehat{y_i}^{t-1}} = \sum_{i=1}^{n}\left[2(\widehat{y_i}^{t-1} - y_i)f_t(x_i) + f_t(x_i)^2\right] + \Omega(f_t) + c = 0 \tag{A.2}$$

D Raw Data

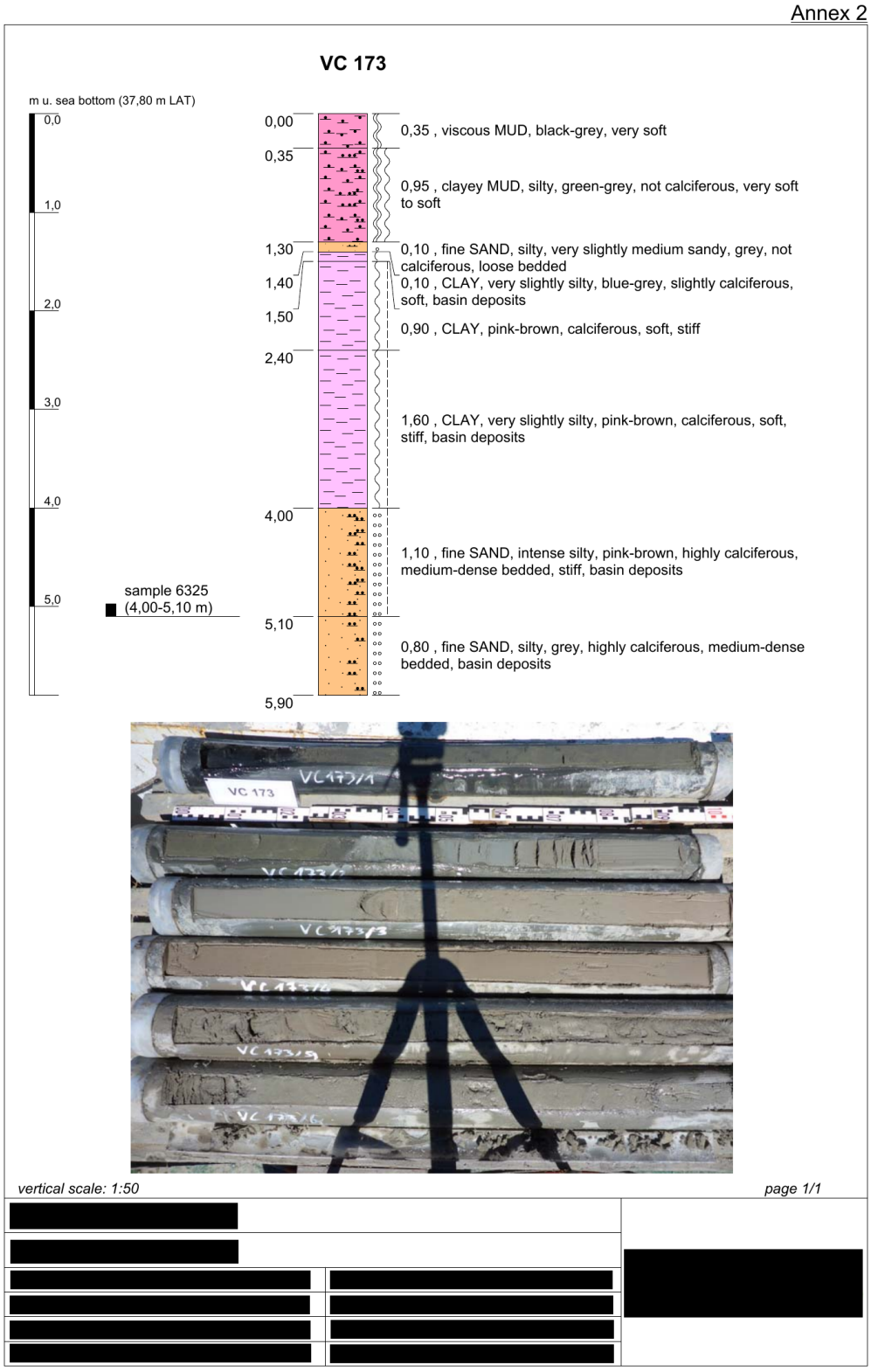


Figure A.3: A typical PDF page of the provided VC data.

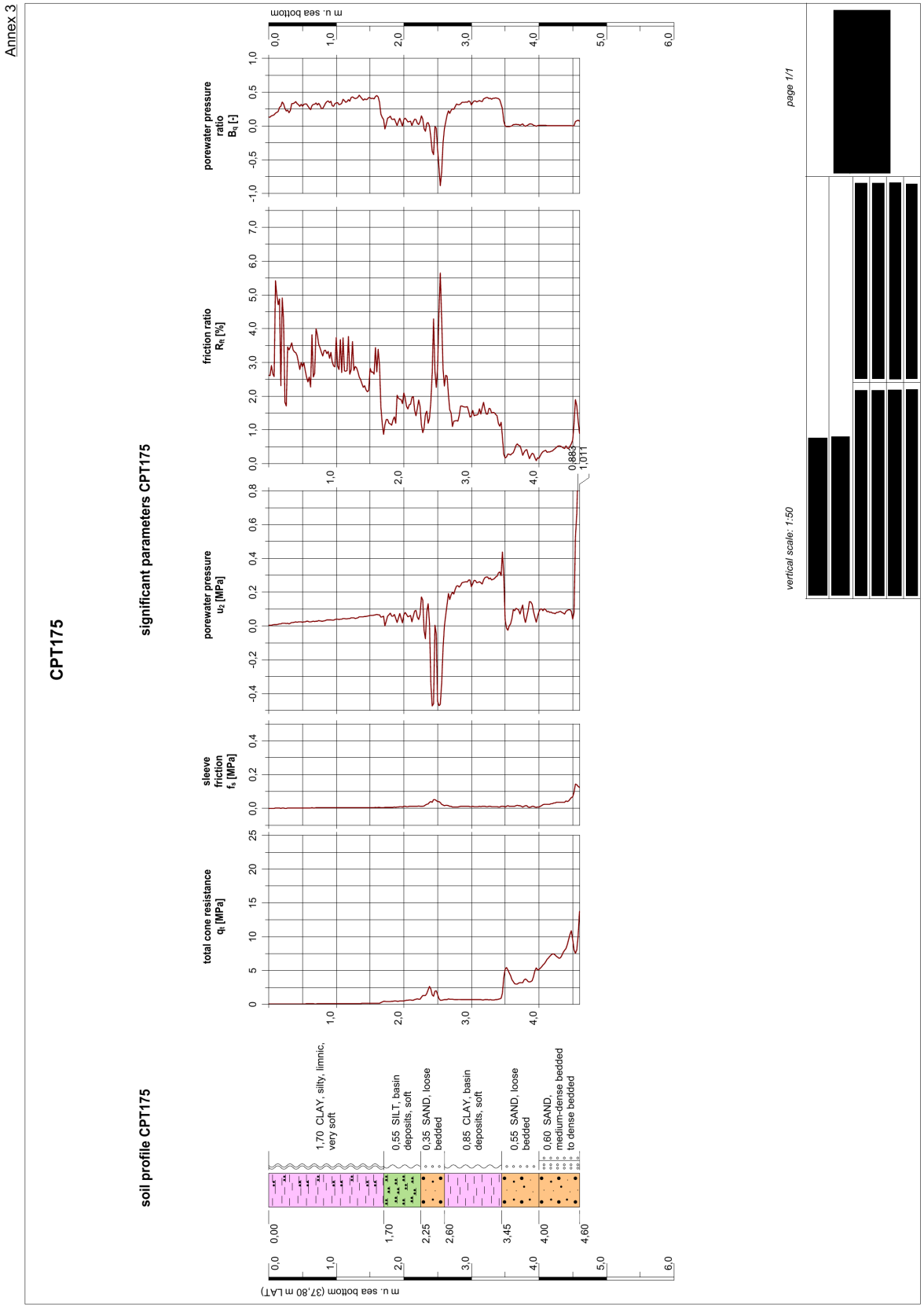


Figure A.4: A typical PDF page of the provided CPT data.

E Feature Correlation

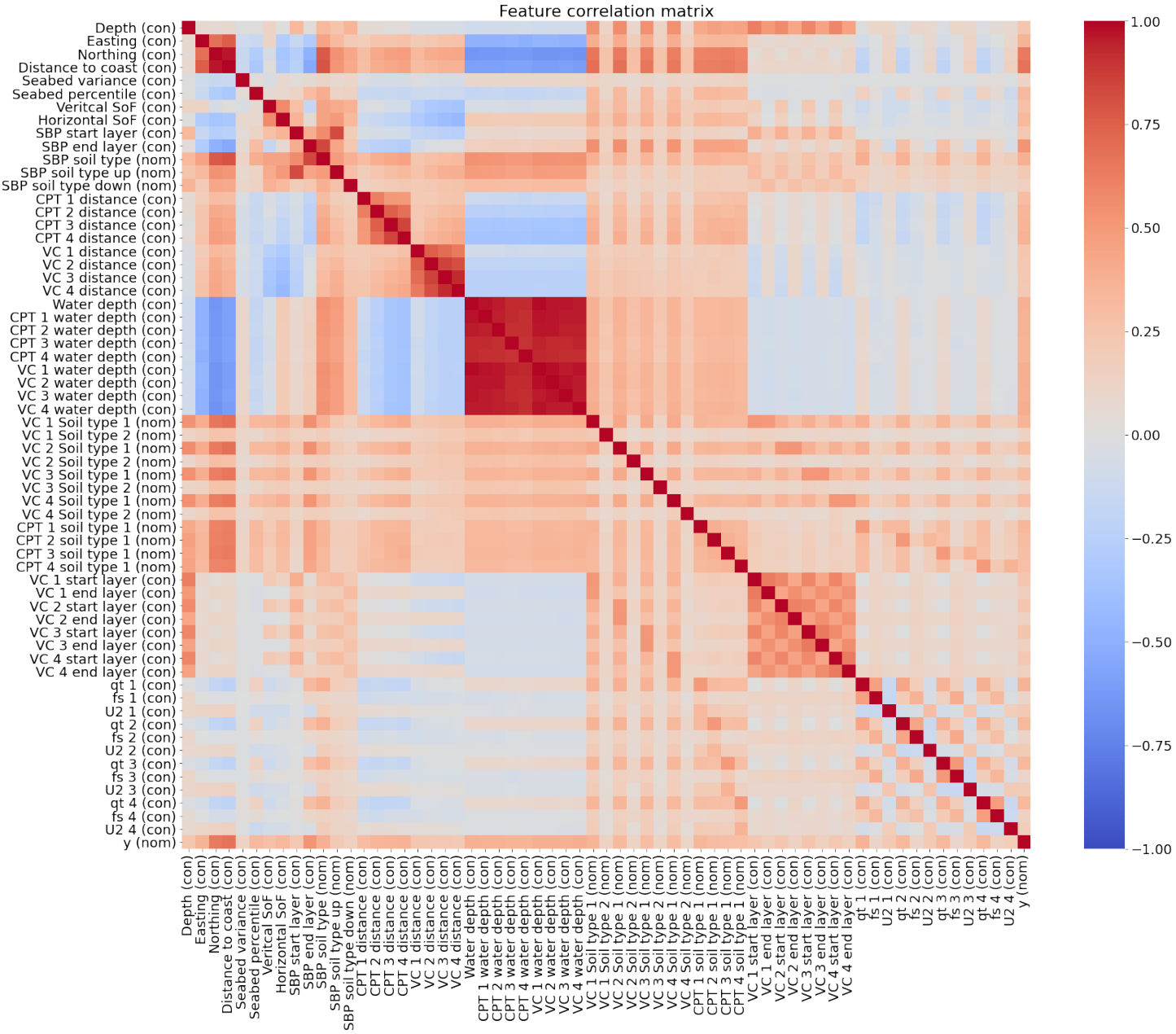


Figure A.5: Feature correlation matrix. The hue of indicates the magnitude of correlation and the colour indicates whether the correlation is negative (blue) or positive (red).

F Digitized Data

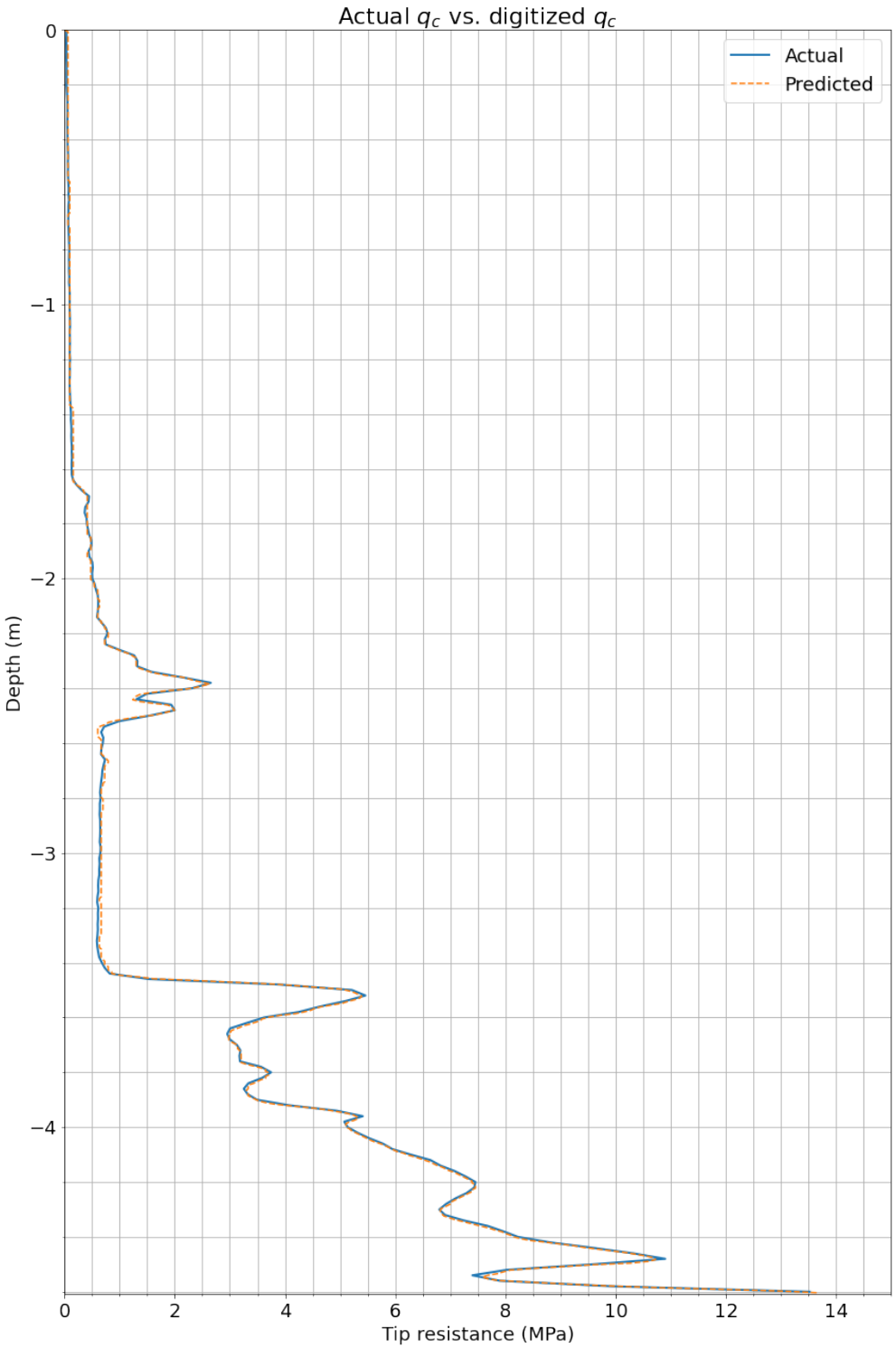


Figure A.6: An enlarged CPT graph with the digitized graph. The digitized graph matches with the real data.

G Classification Models

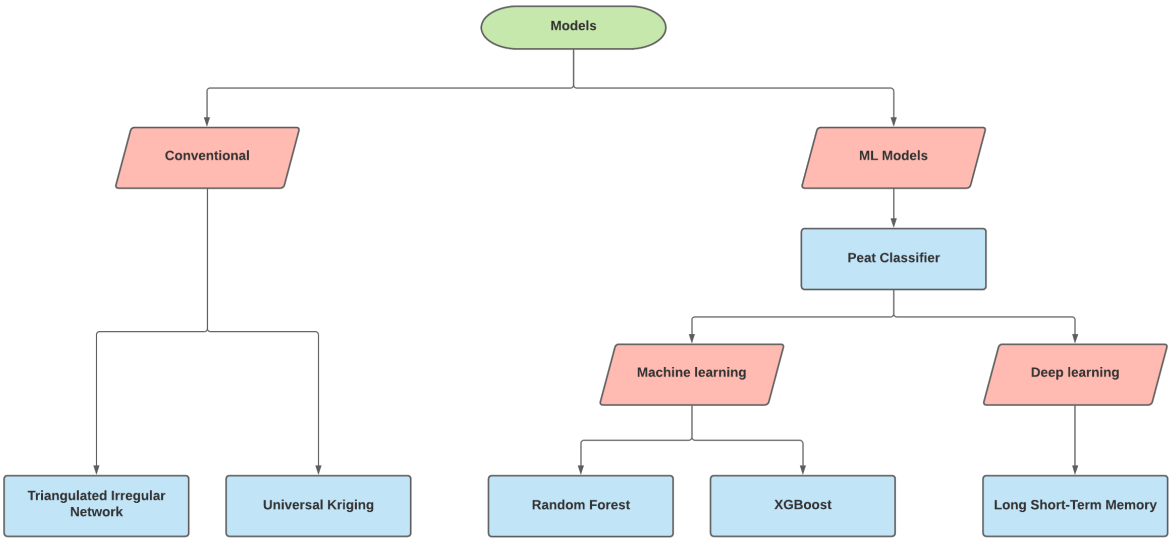


Figure A.7: Overview of the implemented models with the peat classifier.

H Hyperparameter Optimization Approach

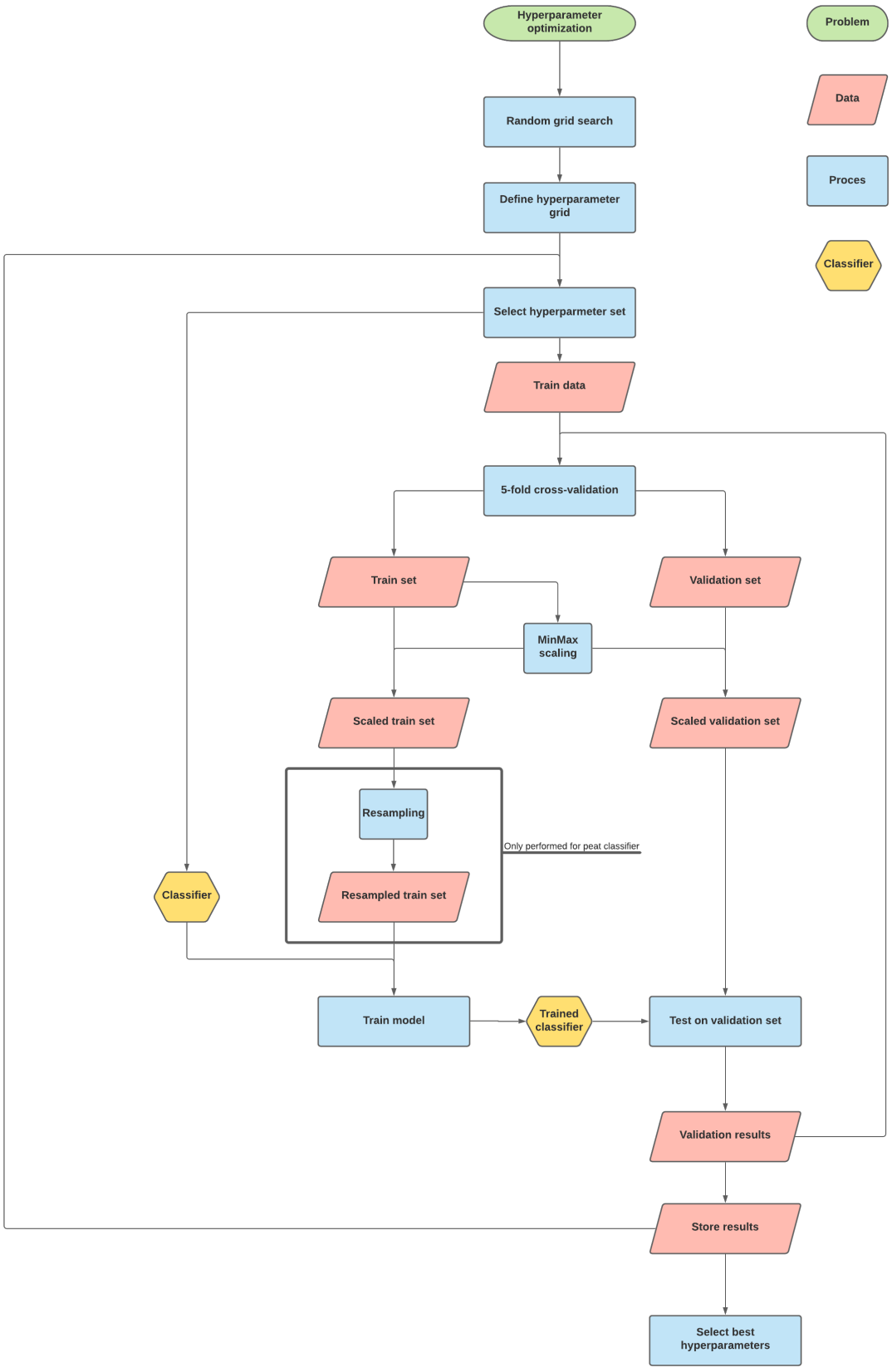


Figure A.8: A flow chart of the complete hyperparameter optimization procedure.

I Model Evaluation Approach

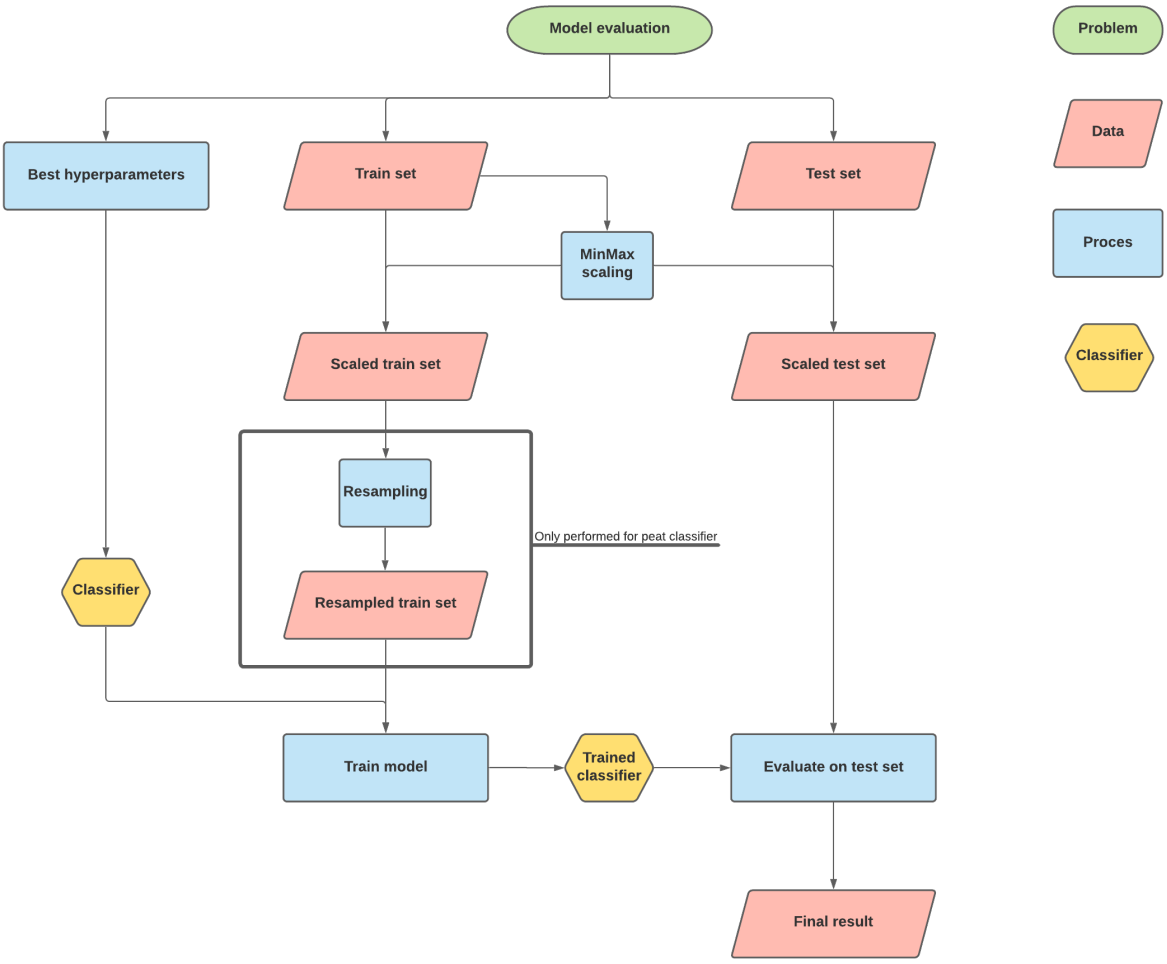


Figure A.9: Flow chart of the model evaluation.

J Oversampling Effect without Class Weights

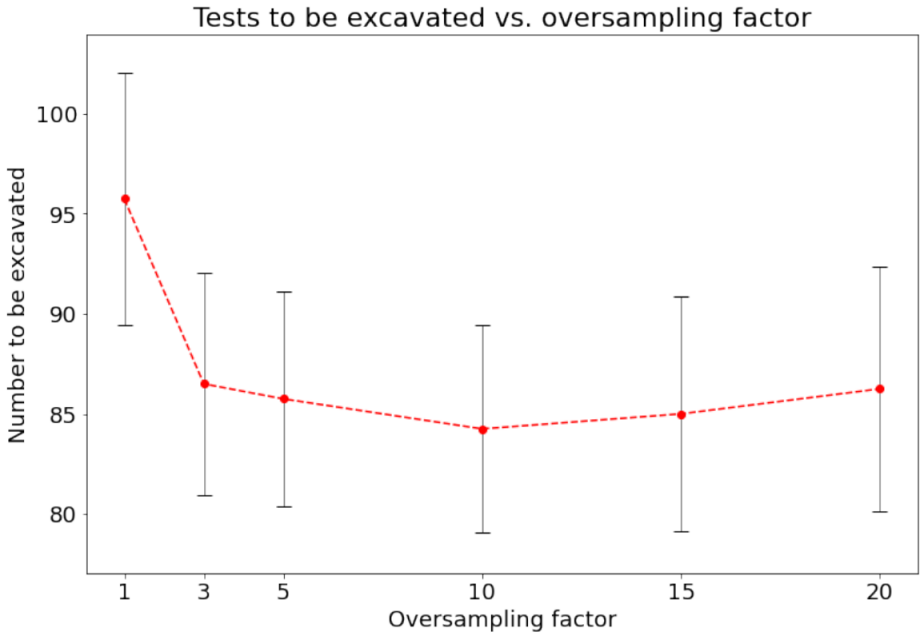


Figure A.10: Effect of oversampling factor without the use of class weights. In contrast to the use of class weights, the oversampling factor here has a clear optimum.

K Impact of Resampling on Probability of Peat

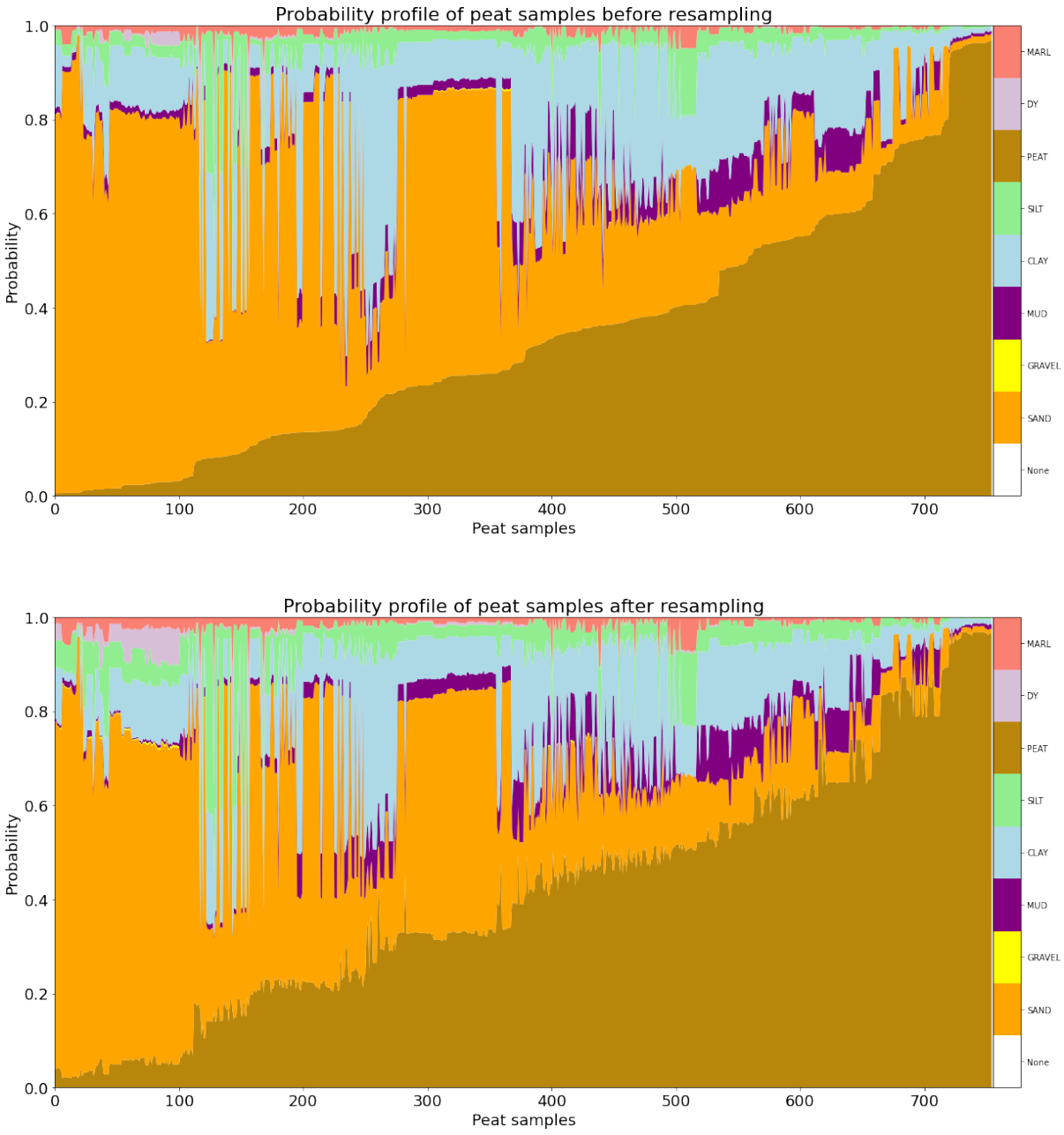


Figure A.11: Effect of resampling on the probability profiles of peat samples. It can be observed that nearly all peat samples have an higher probability of peat after resampling.

L Hyperparameter Optimization Results

L.1 Random Forest

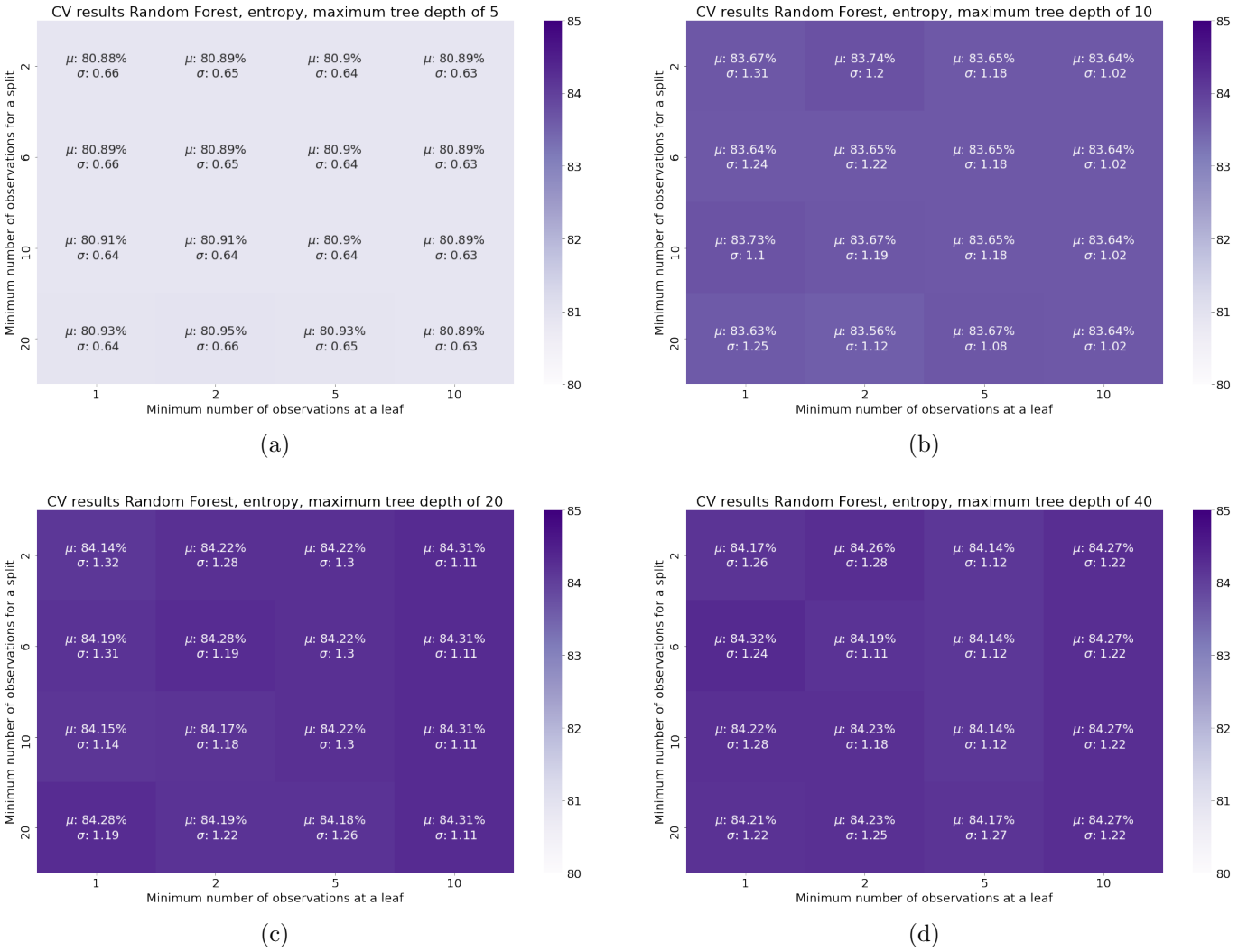


Figure A.12: CV results of random forest with entropy. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

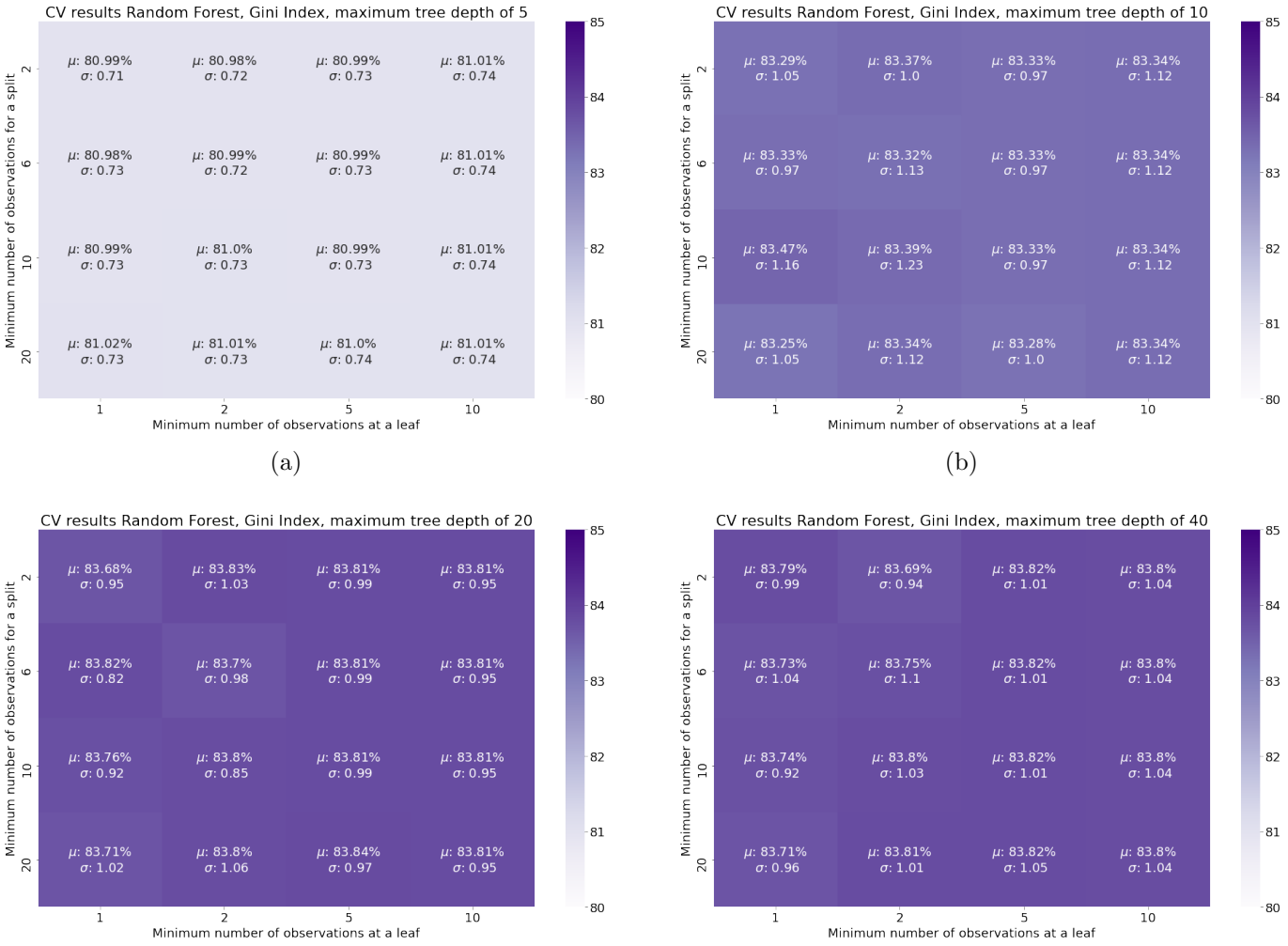


Figure A.13: CV results of random forest with entropy. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

L.2 XGBoost

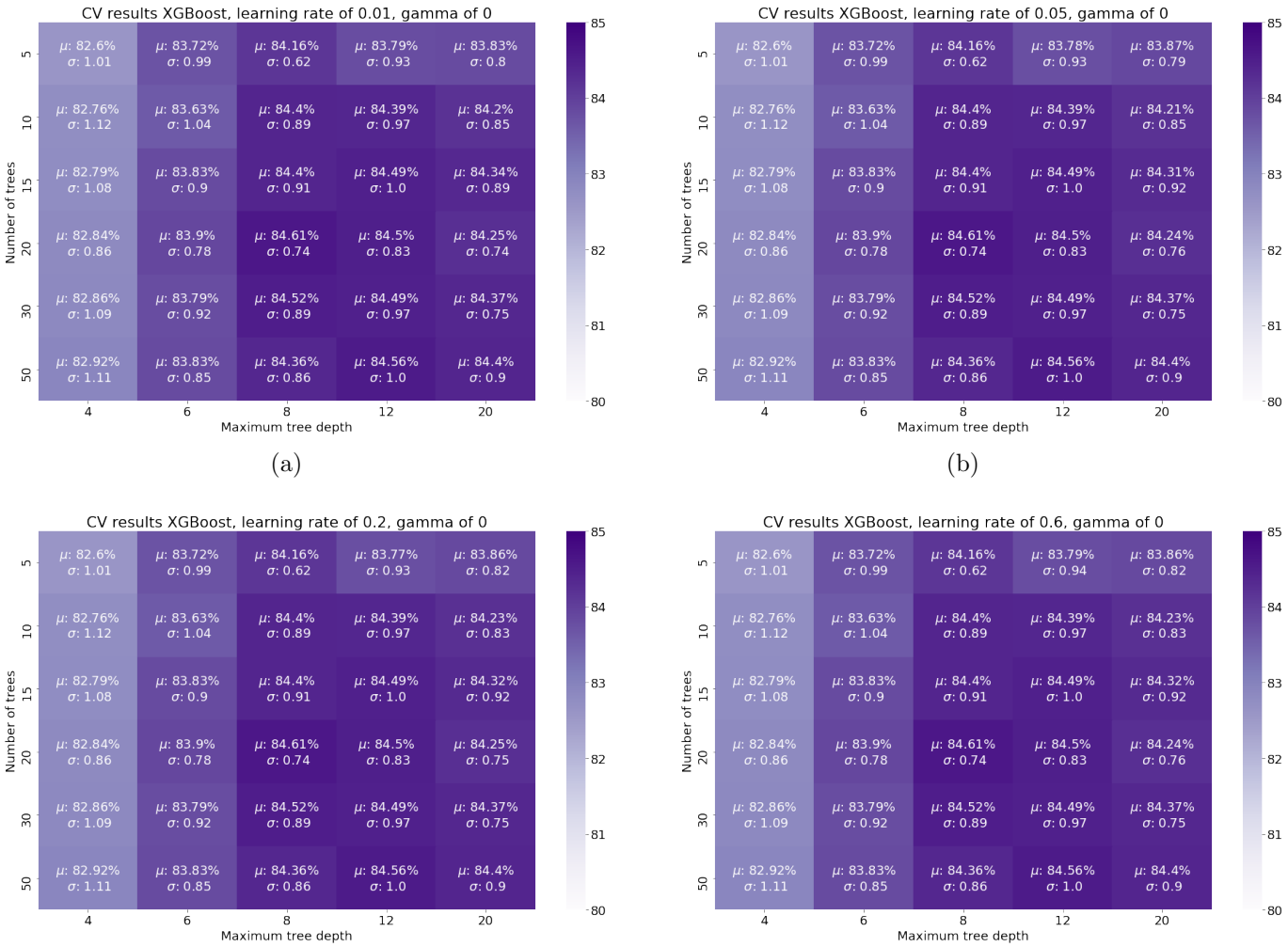


Figure A.14: CV results of XGBoost with a gamma of 0. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

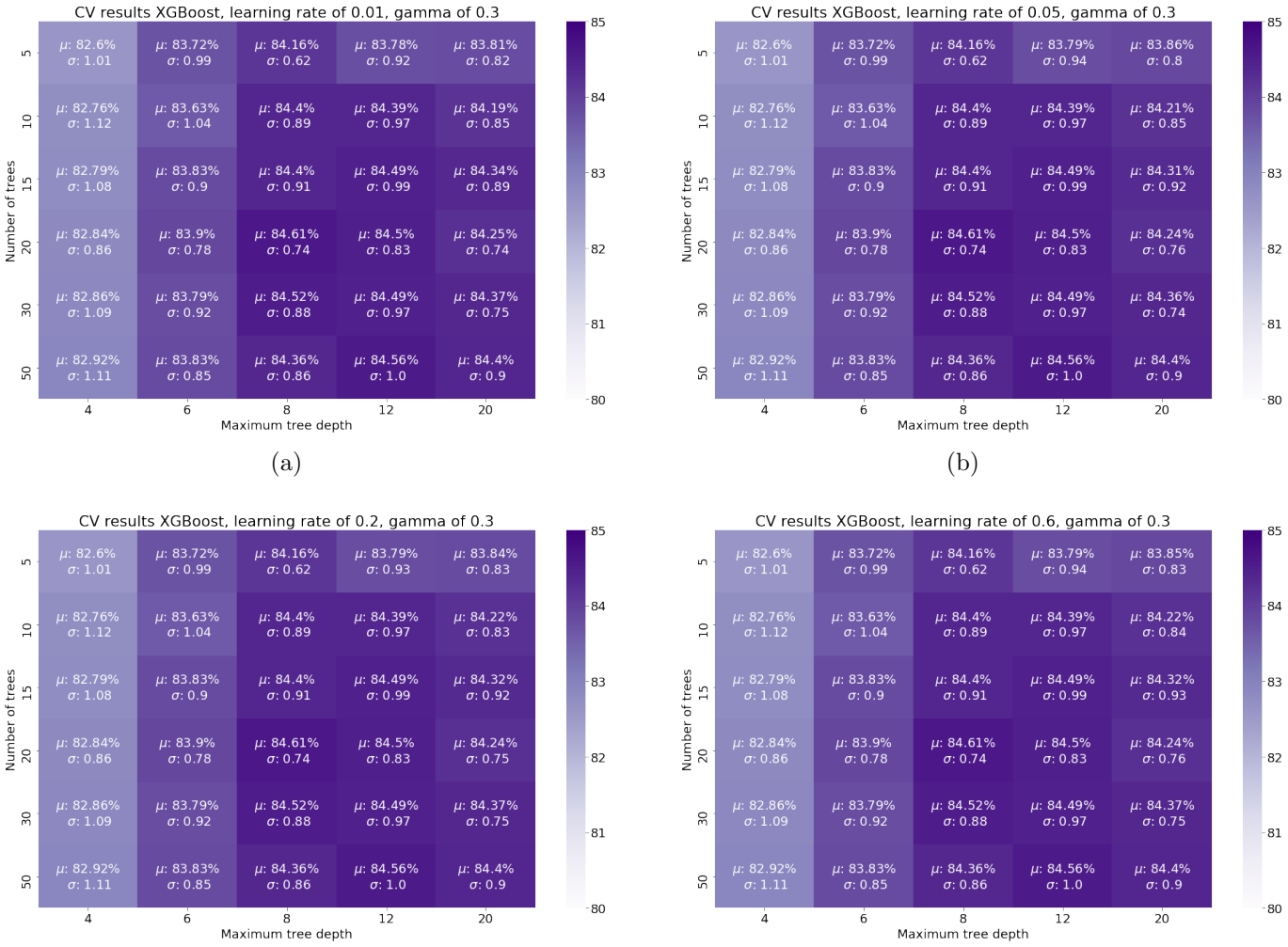
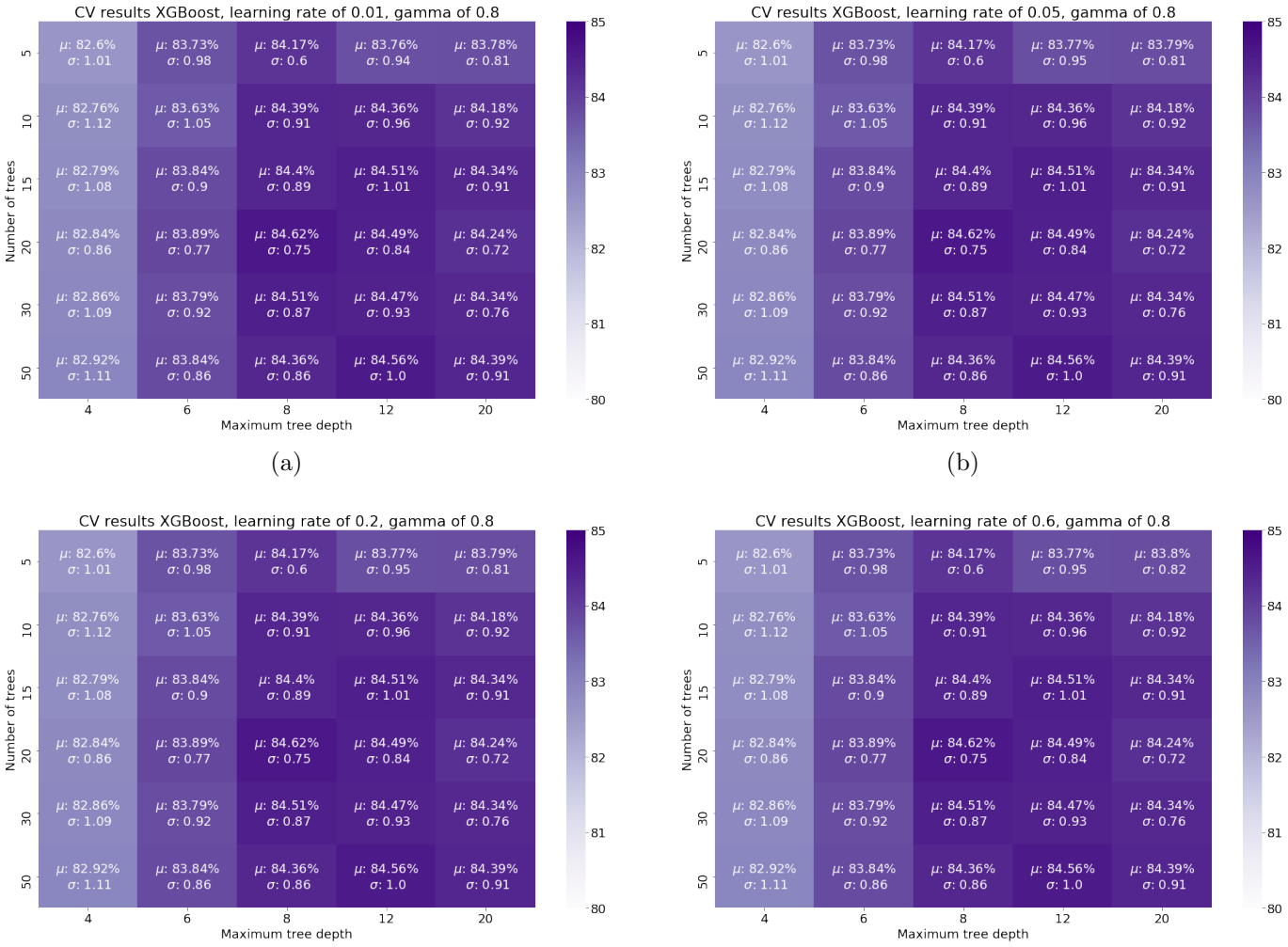


Figure A.15: CV results of XGBoost with a gamma of 0.3. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.



(a) (b) (c) (d)

Figure A.16: CV results of XGBoost with a gamma of 1. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

L.3 LSTM

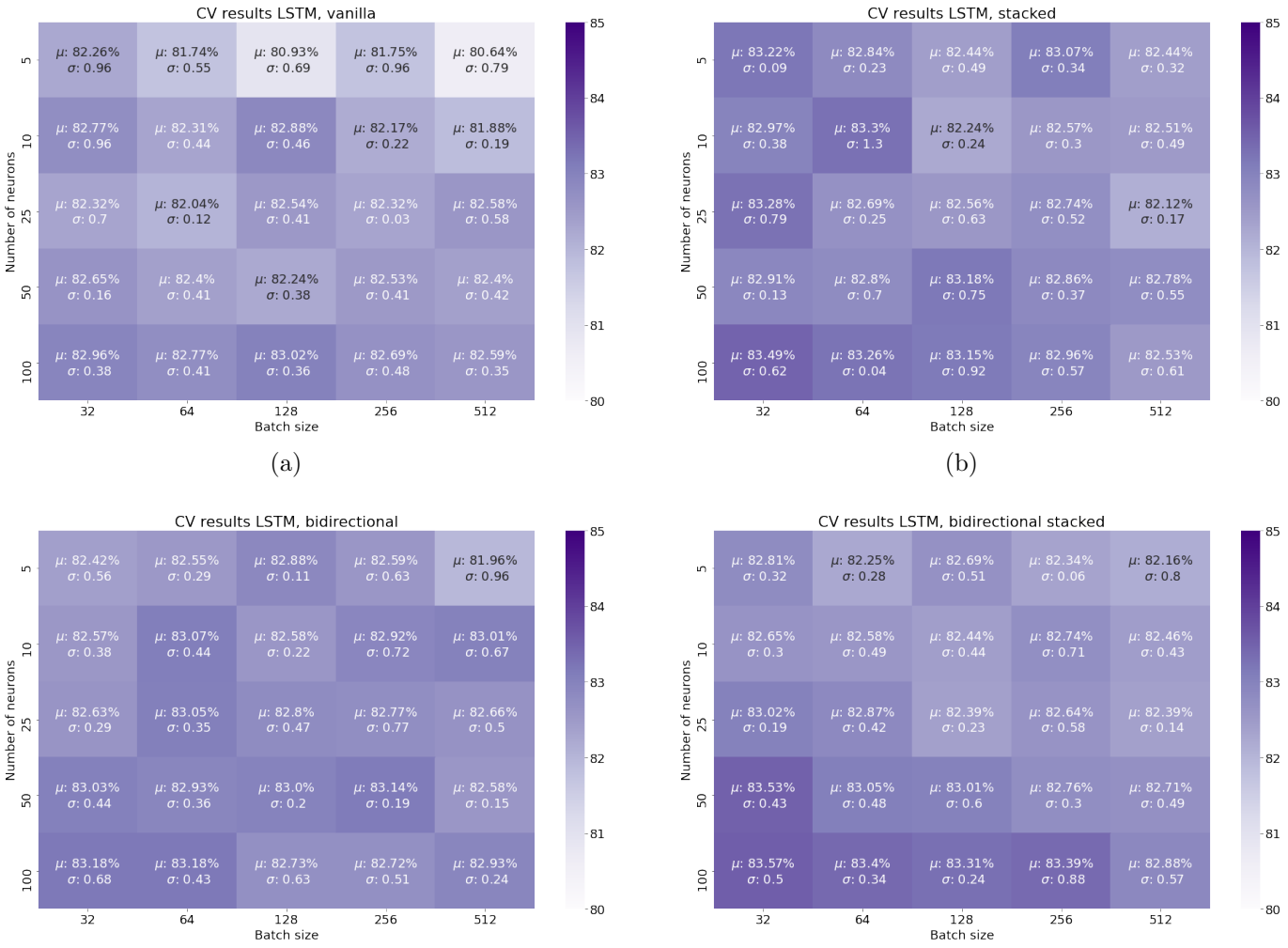


Figure A.17: CV results of LSTM. The top value represent the average accuracy obtained during CV and is accompanied with its standard deviation.

M Tree Structure Random Forest

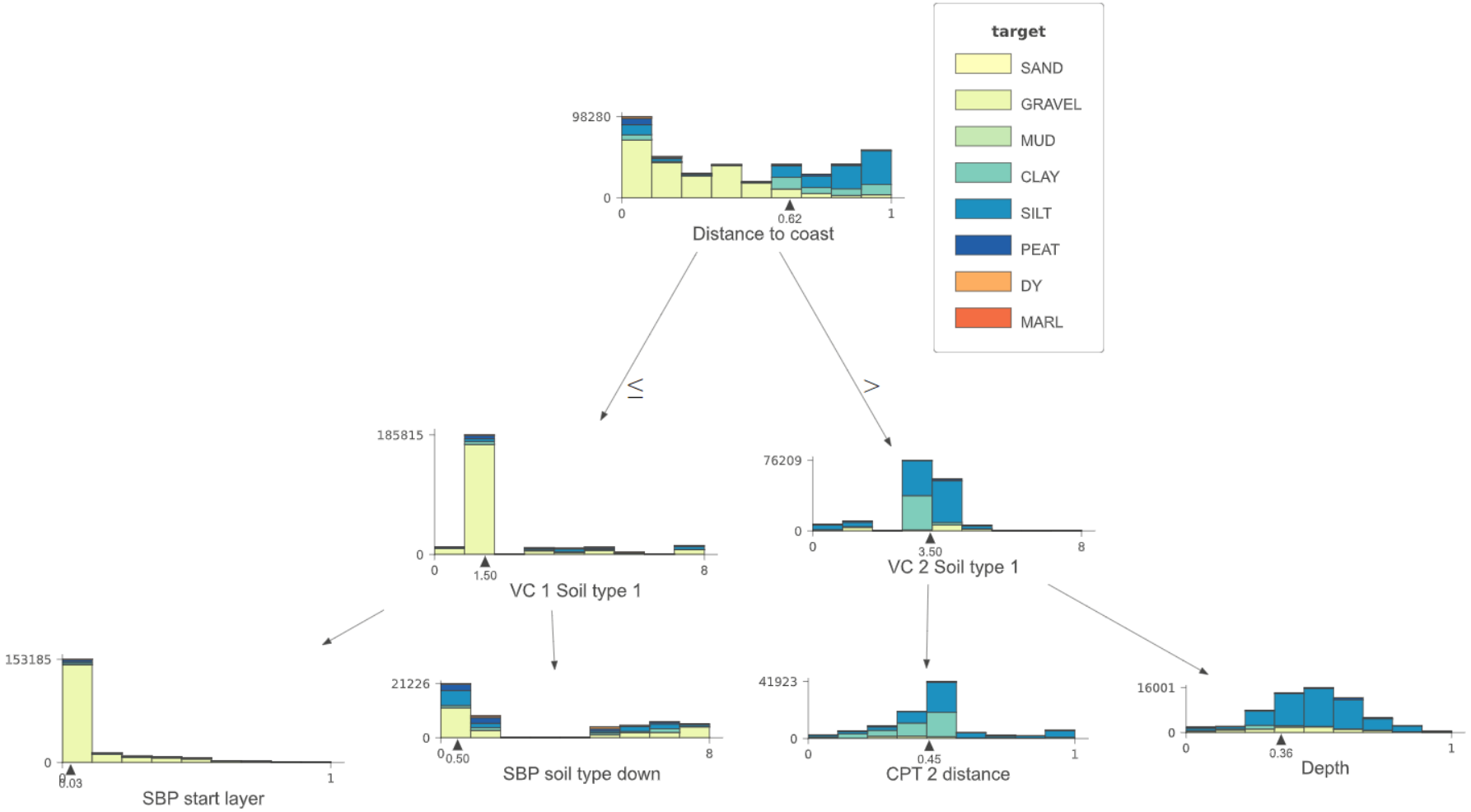


Figure A.18: First tree of the final model of random forest. Only the top three layers are presented. Each split shows the distribution of data with respect to the feature that is selected for the split.

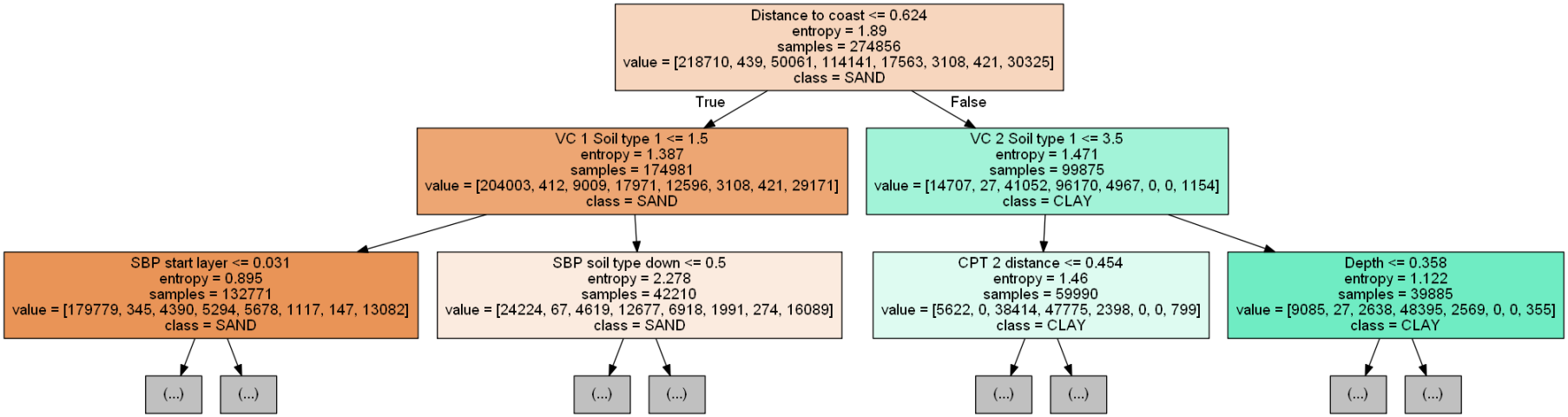


Figure A.19: First tree of the final model of random forest. This is a different illustration of the same tree as in Figure A.18. The figure indicates, among others, the information gained at each split. This is denoted as entropy.

N Final Model Results

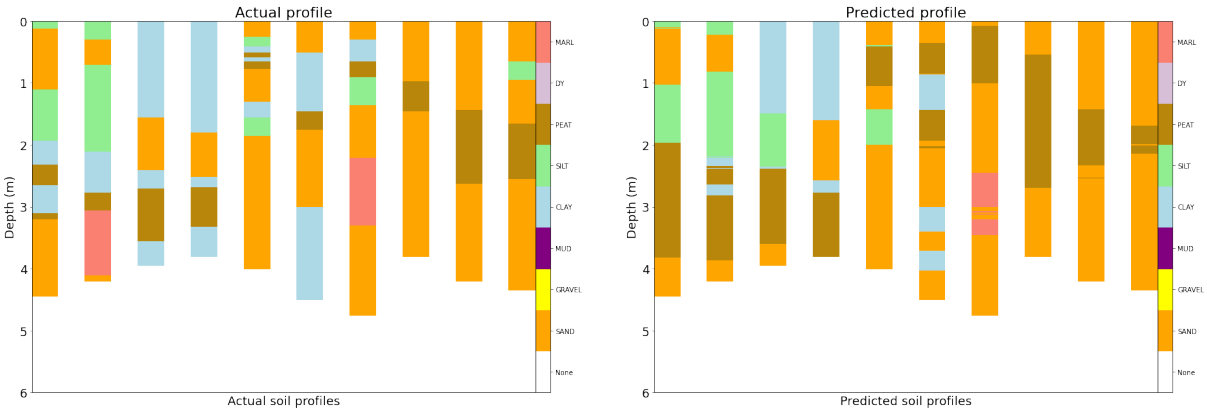


Figure A.20: Correctly predicted soil profiles to be excavated

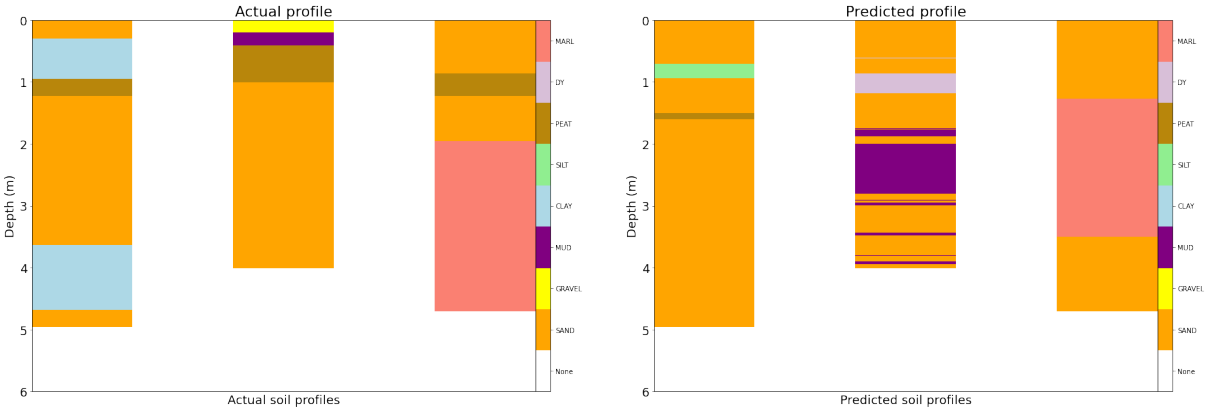


Figure A.21: Missed soil profiles to be excavated

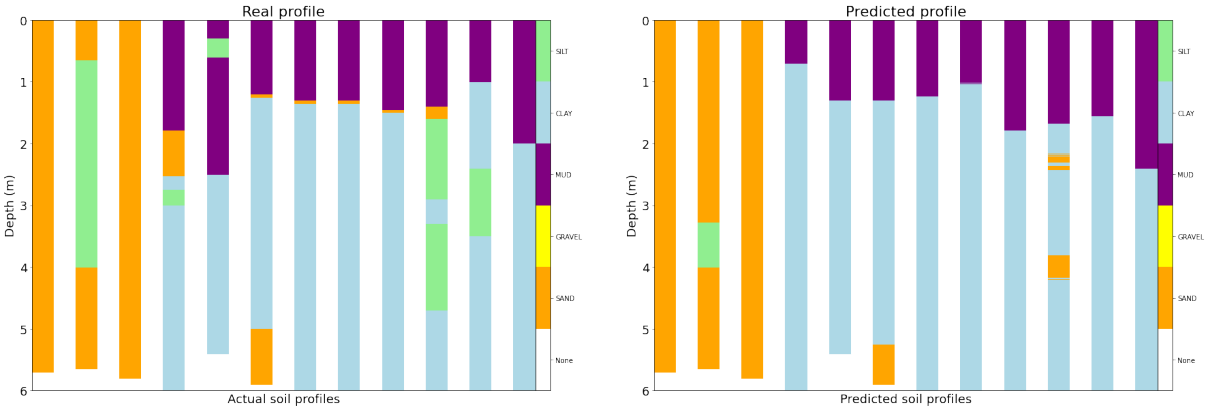


Figure A.22: Soil profiles with a low variability at large depth generated with final model, i.e. random forest.

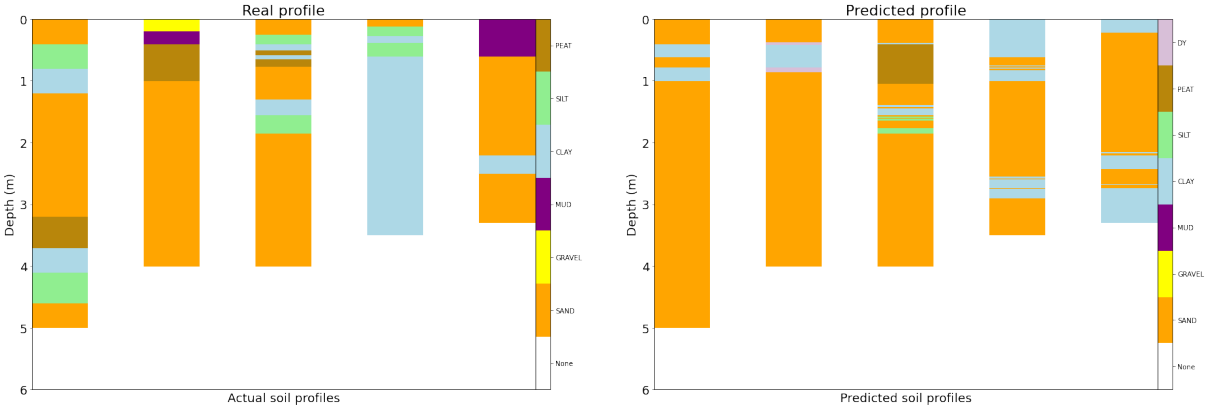


Figure A.23: Soil profiles with an high vertical SoF generated with final model, i.e. random forest.

P 3D Stratigraphic Model

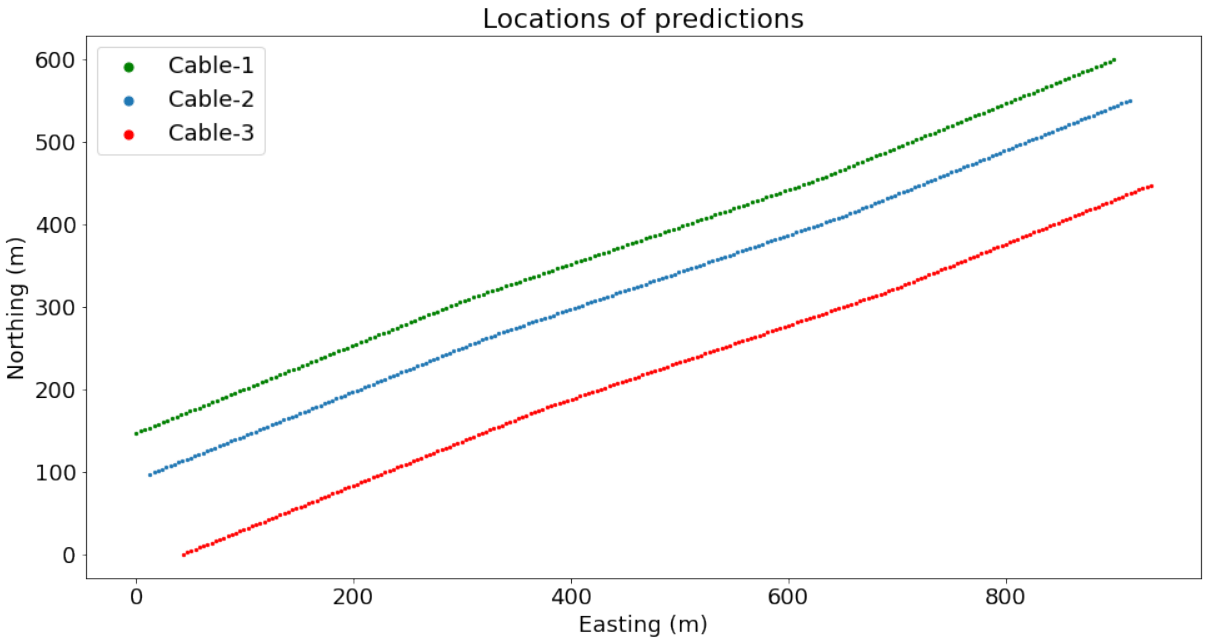


Figure A.24: Locations of the predictions in the 3D stratigraphic model.

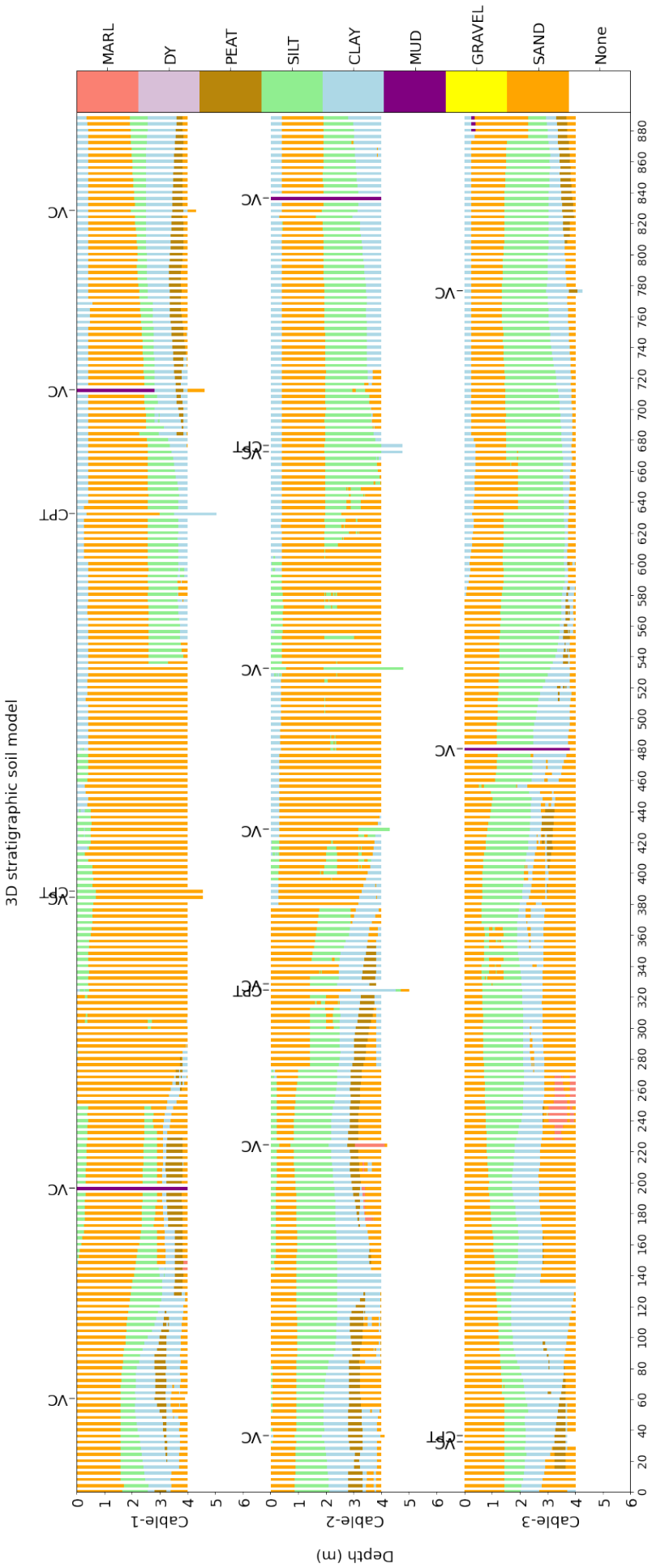


Figure A.25: Section of the final 3D stratigraphic model. It can be noticed that the difference in soil interpretation between the companies is also present.