

Advanced constitutive model parameter determination, optimisation and selection using a database of triaxial tests and machine learning tools

Author:
M.T. Stals

Delft University of Technology
Faculty of Civil Engineering and Geosciences
Section Geo-Engineering

The figure on the front page is from the Data Science Major at the University of Arkansas

Advanced constitutive model parameter determination, optimisation and selection using a database of triaxial tests and machine learning tools

MSc thesis

By

M.T. Stals

Student number: 5421713

A partial fulfilment of the requirements for the degree of
Master of Science in Civil Engineering at the Delft University of Technology
Defence: 22-08-2023.

Master track: Geo-Engineering

Graduation committee:

Chair/supervisor

Dr. Ir. R.B.J. Brinkgreve
Geo-Engineering, TU Delft

Supervisor

Dr. Ir. M.A. Cabrera
Geo-Engineering, TU Delft

Company supervisor

Ir. K. Siderius
Sr. Geotechnical Engineer, Fugro

External supervisor

Dr. Ir. G. Rongier
Applied Geology, TU Delft

Electronic version available at: <https://repository.tudelft.nl/>. Contents under embargo until at least 22-02-2024.



Acknowledgements

The research before you was conducted as a partial fulfilment of my Master of Science in Civil Engineering at the Delft University of Technology, specialisation: Geo-Engineering. This marks the end of a very memorable and enjoyable era that I can look back on with pride. I could not have done this without the support of countless exceptional people, a few of whom I would like to highlight.

Starting with the members of my graduation committee. My gratitude goes to Klaas Siderius at Fugro for the weekly meetings and for introducing me to the topic, which formed the foundation for this thesis. Also to my other colleagues at Fugro who welcomed me into their group and were always available for a sparring session. The knowledge in the field of Geo-Engineering and the shared enthusiasm of my supervisors R. Brinkgreve and M. Cabrera played a crucial role in carrying out this research, as did G. Rongier, who provided me guidance in the world of machine learning. The progress meetings were invaluable to me, with lots of useful feedback, tips and tricks. I am very grateful to them.

I would like to thank my fellow students with whom I spent countless hours studying. It has not been easy, especially since much of the (pre)master was online, but we have kept each other motivated with daily Skype calls. In particular, Mike Mellaard, Tim Nooteboom, Adil Ayi and Matthijs Blankendaal.

And last, but certainly not least, I would like to thank my family, friends and girlfriend who have made it possible for me to complete this journey. These have been challenging years, but they have always supported me and believed in me.

*Michael T. Stals
Heemskerk, July 2023*

Abstract

Numerical modelling in Geo-Engineering is used to solve complex problems by simulating, analysing, or predicting soil behaviour under certain loading and boundary conditions. The soil behaviour is simulated by constitutive models that describe the relationship between stresses and strains through a mathematical formulation. Model parameters are used to calibrate model behaviour to physical soil behaviour measured during in-situ testing (e.g. CPT) or laboratory testing (e.g. triaxial testing). The selection of model parameters is challenging as it needs to cope with aspects as, constitutive model limitations, laboratory test limitations, sample disturbance, soil heterogeneity and many other. This study shows how these model parameters can be determined, optimised and selected by using over 3000 triaxial test results performed on dutch soils (stored in text files) and machine learning tools.

The soil properties (γ , w , γ_d , e_0 and n) of the samples used for these tests were comparable with the literature, as were the soil parameters c' and ϕ' . The model parameters, for the constitutive model Hardening Soil with small-strain stiffness (HS small), were determined by expanding upon the soil parameters and using common correlations and default values from the literature (the traditional method). This study developed a procedure to calibrate/optimize these model parameters by matching the entire stress-strain path of the triaxial test simulation to that of the stress-strain path from the text file. The simulation is performed in the SoilTest facility from the PLAXIS software, this facility allows for simulating laboratory tests based on a single point algorithm using constitutive models. A significant improvement was observed when comparing the evaluation metric r^2 of the stress-strain paths before and after optimising the model parameters. This method proved to be particular effective for the softer soil types like clay and peat in comparison to sand and silt.

The established database consisting of soil properties (which is used as input), soil parameters and optimised model parameters (which are used as output) was investigated with several data analysis techniques. This involved obtaining an overview of (new) statistics that often corresponded to some extent with existing literature. Very high correlations were found between the soil properties, the soil parameter ϕ' and the optimised stiffness parameters (E_{50}^{ref} , E_{oed}^{ref} , E_{ur}^{ref} and G_0^{ref}). For these parameters, a linear and exponential regression analysis between two individual parameters resulted in a fit with a score of $r^2 > 0.4$. A lot of scatter in the graphs was observed which is why more advanced machine learning models were deployed to further improve the score.

The machine learning models: Artificial Neural Network, Gradient Boosting and Kernel Ridge Regression seemed to have the most potential based on an initial analysis on the data set, and a literature study (in which similar models were used). The x values were scaled by means of the Quantile Transformer, the hyperparameters were tuned with a grid search, and the results were evaluated using (group) k-fold, with project numbers as groups. The models were able to predict the same 5 parameters as the linear/exponential regression, but the r^2 showed a significant improvement, in the range of 0.05-0.28 with an average of 0.20. Furthermore, providing the machine learning models with more input parameters (soil properties), generally resulted in an increase of r^2 . Especially when adding the unit weight and water content an increase in performance was observed, only slight increases or even decreases were observed when adding the dry unit weight and the initial void ratio.

This study aims to provide guidance in the determination, selection and optimisation of model parameters for the HS small model, specifically for Dutch soils. It is important to note that these methods, results and conclusions are based on the triaxial test and have not been validated for other tests and engineering practices.

Keywords: triaxial test, parameter determination, constitutive models, machine learning

Contents

Acknowledgements	i
Abstract	ii
Contents	iv
List of Figures	vi
List of Tables	viii
Nomenclature	ix
1 Introduction	1
1.1 Problem statement	1
1.2 Solution orientation	2
1.2.1 Hypothesis	2
1.3 Research questions	2
1.4 Research method	3
1.5 Scope of work	3
1.6 Report layout	4
2 Review of literature	5
2.1 Triaxial test	5
2.1.1 Consolidated	6
2.1.2 Unconsolidated	8
2.2 Constitutive models	9
2.2.1 Linear Elastic Perfectly Plastic (LEPP) Mohr-Coulomb (MC) model	9
2.2.2 Hardening Soil model with small-strain stiffness (HS small)	10
2.2.3 Soft Soil model	12
2.3 Machine learning tools	12
2.3.1 Single regression techniques	13
2.3.2 Multiple linear regression	14
2.3.3 Tree-based methods	14
2.3.4 Artificial Neural Network	15
2.3.5 (Kernel) Ridge Regression	16
2.3.6 Evaluation metrics	17
2.3.7 Machine learning in Geotechnical engineering	18
2.4 Summary	19
3 Methodology	20
3.1 Database description	20
3.1.1 General	20
3.1.2 Locations	21
3.2 Soil properties	22
3.3 Soil parameters	23
3.3.1 Calibrating and correcting	23
3.3.2 Determination: Stiffness and power for stress dependency of stiffness	24
3.3.3 Determination: Cohesion and friction angle	25

3.4	Model parameters	26
3.4.1	Initial	26
3.4.2	Sensitivity analysis	27
3.4.3	Optimisation	29
3.5	Parameter boundaries	33
3.6	Single regression analysis	33
3.6.1	Outliers	33
3.7	Machine learning analysis	34
3.7.1	Preprocessing the data	35
3.7.2	Validating	35
3.7.3	Hyperparameter tuning	36
3.7.4	Input/output of data	37
3.7.5	Model selection	37
3.8	Exploratory data analysis	38
3.8.1	Exploratory data analysis: Statistics	39
3.8.2	Exploratory data analysis: Correlation	41
3.9	Summary	42
4	Results and discussion	43
4.1	Statistics	43
4.1.1	Soil properties	43
4.1.2	Soil parameters	44
4.1.3	Model parameters	46
4.2	Correlations	49
4.3	Single regression	50
4.4	Machine learning	51
4.4.1	Quick scan	52
4.4.2	Single output predictions	52
4.4.3	Multi output predictions	55
4.4.4	General discussion	57
4.5	Automated Parameter Determination	57
5	Conclusions	59
6	Recommendations	61
	Bibliography	62
A	Example text file	66
B	Database summary	68
C	Visualisation of triaxial results	70
D	Sensitivity analysis	71
D.1	Drained	71
D.2	Undrained	74
E	Code snippets	78
F	The five-number summary	80
G	Additional violin plots	84
H	CPT	86

List of Figures

1.1	Reading guide	4
2.1	Triaxial apparatus NEN-EN-ISO 2018 <i>b</i>	5
2.2	Cohesion and internal friction angle NEN-EN-ISO 2018 <i>b</i>	6
2.3	Stress path diagram (p'-q) Obrzud et al. 2018	7
2.4	Stress path comparison	7
2.5	Stiffness for different confining pressures	8
2.6	Unconsolidated Undrained triaxial test Verruijt and Van Baars 2007	8
2.7	Yield contour of the Mohr-Coulomb model in the principal stress space Brinkgreve 2005	9
2.8	Yield contour of the Hardening Soil model in the principal stress space Brinkgreve 2005	10
2.9	Small-strain stiffness Brinkgreve et al. 2007	11
2.10	Yield contour of the Soft Soil model in the principal stress space	12
2.11	Machine learning types with subcategories (Sarker 2021)	13
2.12	Single regression techniques	14
2.13	Decision tree (Charbuty and Abdulazeez 2021)	15
2.14	Tree based algorithms (Lev 2022)	15
2.15	Artificial Neural Network	16
2.16	Common activation functions (Conza 2020)	16
2.17	KRR vs SVR execution time (Metzen 2023)	17
2.18	RIFT method Lin et al. 2015	18
3.1	Key features from the database	21
3.2	Borehole locations	22
3.3	Determination of the E50	24
3.4	Parameter determination: cohesion and internal friction angle	26
3.5	Parameter determination: internal friction angle with zero cohesion	26
3.6	Sensitivity analysis	28
3.7	Schematic overview of the algorithm	29
3.8	Fitting the stress-strain curve	30
3.9	$E_{oed}^{ref} = E_{50}^{ref}$	30
3.10	Visualisation of the DBSCAN algorithm (Tran et al. 2013)	34
3.11	Outlier detection example with DBSCAN, ($\epsilon = 0.05$ and $samples = 5$)	34
3.12	Simplest form of machine learning	35
3.13	Cross validation Pedregosa et al. 2011	36
3.14	Hyperparameter tuning Bergstra and Bengio 2012	36
3.15	A more advanced approach on machine learning	37
3.16	Normal distribution and the influence of μ and σ_{sd}	39
3.17	Visualisation of the five-number summary, box-and-whisker plot (Dekking et al. 2005)	40
3.18	Violin plot explanation (labxchange 2021)	41
4.1	Soil properties violin plot	44
4.2	Violin plot: c' and ϕ'	45
4.3	Including additional features to subdivide ϕ'	45
4.4	Violin plot: E_{50}^{ref}	46
4.5	Convergence curve of r^2	46
4.6	Initial r^2 vs optimised r^2	47
4.7	Model parameters violin plot	48

4.8	Explanation of lower friction angles for sand (Cui et al. 2021)	48
4.9	Correlation heat map of all collected parameters	49
4.10	Best found r^2 , γ_d vs ϕ'	50
4.11	Double parameter input	51
4.12	Comparison between models, in legend order; Multiple Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, eXtreme Gradient Boosting, Artificial Neural Network, Gaussian Process Regressor, Regular Ridge Regressor, Kernel Ridge Regressor	52
4.13	3D plots of output parameter set 1 using k-fold cross validation	53
4.14	Validation of the variability of the model and the influence of k-fold	54
4.15	3D plots of output parameter set 1 using group k-fold cross validation	55
4.16	3D plots of output parameter set 2 and 3	56
4.17	Optimised model parameters, adjusted output array	56
4.18	Triaxial test simulation comparison with the APD system	58
A.1	Example text file	67
B.1	Bar charts	68
B.2	Histograms	69
C.1	Example of how to present triaxial results	70
D.1	E_{50}^{ref}	71
D.2	E_{oed}^{ref}	71
D.3	E_{ur}^{ref}	72
D.4	ν_{ur}	72
D.5	G_0^{ref}	72
D.6	$\gamma_{0.7}$	73
D.7	ϕ'	73
D.8	K_0^{nc}	73
D.9	R_f	74
D.10	E_{50}^{ref}	74
D.11	E_{oed}^{ref}	74
D.12	E_{ur}^{ref}	75
D.13	ν_{ur}	75
D.14	G_0^{ref}	75
D.15	$\gamma_{0.7}$	76
D.16	ϕ'	76
D.17	K_0^{nc}	76
D.18	R_f	77
G.1	Initial void ratio e_0	84
G.2	Optimised model parameters: E_{50}^{ref} , E_{oed}^{ref} , E_{ur}^{ref} and G_0^{ref}	85
H.1	DKM5	87
H.2	DKM32	88

List of Tables

2.1	LEPP MC parameters	10
2.2	HS small parameters	11
2.3	SS parameters	12
2.4	Coefficient of determination guidelines Latuni 2019	18
3.1	Key features in text files	20
3.2	Initial parameters	23
3.3	Initial parameter set	27
3.4	Base case for sensitivity analysis	28
3.5	Optimisation results comparison	31
3.6	Optimisation sequence	32
3.7	Parameter boundaries	33
3.8	Parameter sets	37
4.1	Relevant linear and non-linear exponential regression results	50
4.2	Relevant multiple linear regression results	51
4.3	Comparison with the APD system	57
F.1	Soil property: γ	80
F.2	Soil property: w	80
F.3	Soil property: γ_d	80
F.4	Soil property: e_0	80
F.5	Soil property: n_0	80
F.6	Soil parameter: ϕ'	81
F.7	Soil parameter: c'	81
F.8	Soil parameter: E_{50}^{ref} not optimised	81
F.9	Initial r^2	81
F.10	Optimised r^2	81
F.11	Optimised model parameter: E_{50}^{ref}	81
F.12	Optimised model parameter: E_{oed}^{ref}	82
F.13	Optimised model parameter: E_{ur}^{ref}	82
F.14	Optimised model parameter: ν_{ur}	82
F.15	Optimised model parameter: m	82
F.16	Optimised model parameter: G_0^{ref}	82
F.17	Optimised model parameter: $\gamma_{0.7}$	82
F.18	Optimised model parameter: $*c'$	83
F.19	Optimised model parameter: $*\phi'$	83
F.20	Optimised model parameter: ψ	83
F.21	Optimised model parameter: k_0^{nc}	83
F.22	Optimised model parameter: R_f	83

Nomenclature

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
APD	Automated Parameter Determination
CIDMS	Consolidated Isotropic Drained Multi Stage
CIUMS	Consolidated Isotropic Undrained Multi Stage
CPT	Cone Penetration Test
CV	Cross Validation
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DMT	Flat Dilatometer Test
DSS	Direct Simple Shear
ESP	Effective Stress Path
FEM	Finite Element Method
GBR	Gradient Boosting Regressor
GPR	Gaussian Process Regressor
HS (small)	Hardening Soil (with small-strain stiffness)
IQR	InterQuartile Range
ISO	International Standardization Organization
KRR	Kernel Ridge Regressor
LEPP	Linear Elastic Perfectly-Plastic
MAE	Mean Absolute Error
MC	Mohr-Coulomb
ML	Machine Learning
MLR	Multiple Linear Regression
NC	Normally Consolidated
NEN	Dutch standardisation of guidelines
OAT	One-At-a-Time
OC	Over-Consolidated
PSD	Particle Size Distribution

RFR	Random Forest Regressor
RIFT	Robust and Interpolation-Free Technique
RMSE	Root Mean Squared Error
RRR	Regular Ridge Regressor
SLS	Serviceability Limit State
SPT	Standard Penetration Test
SS(C)	Soft Soil (Creep)
SVR	Support Vector Regressor
TSP	Total Stress Path
ULS	Ultimate Limit State
UU	Unconsolidated Undrained
XGB	eXtreme Gradient Boosting

Roman symbols

A_{cor}	Cross-sectional area of the specimen at any point in time	m^2
B	Skempton B-parameter	–
c'	Effective cohesion	kPa
D_c	Specimen diameter at the end of loading	m
D_m	Initial internal diameter of the confining membrane	m
E	Young's modulus	kPa
e_0	Initial void ratio	–
E_m	Elastic modulus of the confining membrane	kPa
E_{50}	Drained Secant stiffness at 50% of the peak deviatoric stress	kPa
E_{50}^u	Undrained Secant stiffness at 50% of the peak deviatoric stress	kPa
f	Yield function	–
f_r	Friction ratio	–
f_s	Sleeve friction	kPa
g	Gravitational constant	m/s^2
g	Plastic potential function	–
G_s	Specific gravity	–
H	Height	m
K_{fp}	Load carried by filter paper covering a unit length of the specimen diameter	kPa
m	Power for stress-level dependency of stiffness	–
m_s	Mass of solids	kg
m_t	Total mass	kg
m_w	Mass of water	kg
n	Porosity	–

P	Vertical load reading	kPa
p'	Mean effective stress	kPa
P_{fp}	Fraction of perimeter covered by filter	—
q	Deviatoric stress	kPa
q_c	Cone resistance	kPa
q_{50}	50% of peak deviatoric stress	kPa
r_p	Pearson product-moment correlation coefficient	—
r_s	Spearman rank order correlation coefficient	—
s_u	Undrained shear strength	kPa
Sr	Degree of saturation	—
t_m	Initial thickness of the unstressed membrane	m
u	Pore water pressure	kPa
V	Volume	m^3
w	Water content	—

Greek symbols

$(\Delta\sigma_h)_m$	Correction to horizontal total stress due to membrane	kPa
$(\Delta\sigma_v)_m$	Correction to vertical total stress due to membrane	kPa
$(\Delta\sigma_v)_{fp}$	Correction to vertical total stress due to filter paper	kPa
γ	Unit weight	kN/m^3
γ_d	Dry unit weight	kN/m^3
μ	Sample mean	—
ν	Poisson's ratio	—
ν_{ur}	Poisson's ratio for unloading/reloading	—
ϕ'	Effective internal friction angle	°
ψ	Dilatancy angle	°
ρ	Density	kg/m^3
ρ_d	Dry density	kg/m^3
ρ_p	Particle density	kN/m^3
ρ_w	Density of water	kN/m^3
σ_1	Major principal stress	kPa
σ_3	Minor principal stress	kPa
σ_c	Cell pressure	kPa
σ_{sd}	Standard deviation	—
ε_{ax}	Axial strain	—
ε_{vol}	Volumetric strain	—

Chapter 1

Introduction

Numerical methods, like the finite element method (FEM), have gained popularity and an increasing importance in Geo-engineering. They are widely accepted and are now considered a standard design tool due to advances in hardware and software in recent decades. Numerical modelling has proven to be a powerful tool for solving complex problems by simulating, analysing, or predicting the soil behaviour under certain loading and boundary conditions. This is firstly because the (commercial) software has been developed to the point where it is easy to operate and secondly because of the constitutive models that describe the mechanical behaviour of soils in a continuum framework (Schweiger et al. 2019). These constitutive models describe the relationship between stresses and strains, by means of a mathematical formulation, and there are models available ranging from very simple to very advanced. Model parameters are required to quantify certain features of the soil behaviour. In general, simple models require less input parameters than more advanced models, but they may therefore lack some essential features of soil behaviour (Brinkgreve et al. 2010). Nevertheless, the right selection of model parameters is always important to make good predictions when using FEM, regardless of the model chosen.

1.1 Problem statement

Today, there is no single constitutive model that is applicable in every situation. The applicability of a model depends on the soil type, application, type of loading (Bentley 2022) and available soil parameters. In the preliminary phase of a project, relatively little soil investigation has been done and therefore one tends to opt for a more simplistic model, such as the Linear Elastic Perfect Plastic (LEPP) Mohr-Coulomb (MC) model, which requires only a limited number of parameters. This is a clear first-order approximation model and is well known for its simple stress-strain relationship (Ti et al. 2009), it can describe the (drained) failure behaviour quite well (Goldscheider 1984) and is suitable for many practical applications. However, there are limitations as this model lacks essential features of the complex soil behaviour. More advanced models, such as Hardening Soil with small-strain stiffness (HS small) or Soft Soil Creep (SSC) can be chosen for a more representative description of soil behaviour, depending on the application. The problem is that, as mentioned earlier, this usually involves an increasing number of parameters to be obtained.

There are several methods to determine or derive these model parameters. In situ tests perfectly mimic on-site stress conditions and the soil is virtually undisturbed, but control over the boundary conditions is lacking (Wroth 1984). Some examples of these in situ techniques are the cone penetration test (CPT) (Robertson 1986), the standard penetration test (SPT) and the flat plate dilatometer (DMT) (Marchetti 2015). Sampling tests, on the other hand, do control the boundary conditions, but work with disturbed samples. Some examples are the triaxial test and the oedometer test. In addition, many correlations, rules of thumb and tables have been created to provide guidance when selecting parameters. Parameter determination is a heavily debated and researched topic in the field of Geo-Engineering, due to the complexity of these heterogeneous, natural building materials and the amount of engineering judgement required. van Berkom 2020, for example, aimed to elaborate a transparent and adaptable parameter determination framework that will increase the reliability of parameters derived from in situ tests by using a graph-based approach.

1.2 Solution orientation

The triaxial test is perhaps one of the most well-known laboratory tests among Geo-Engineers, to measure the mechanical properties of the soil. Traditionally, these tests are used to determine soil parameters such as the cohesion (c'), internal friction angle (ϕ') and stiffness (E), but in doing so, a lot of information of the soil behaviour is lost in the process.

In fact, the triaxial data can also be used to optimise the parameters of a constitutive model by matching the measurements simulated by a model to the actual laboratory measurements. This way, the parameters are adjusted so that the constitutive model best mimics the soil behaviour measured during the laboratory tests. By analysing a large number of tests, (new) statistics and correlations can be discovered. Correlations are a powerful tool to determine relationships between different parameters and can provide guidance when selecting model parameters, especially when little soil investigation has been performed. Many correlations have been established in recent decades, which often consist of correlating two parameters with each other by means of (non-)linear regression techniques. When increasing the number of parameters to be correlated, simplistic regression methods, like linear regression, are less suitable. Perhaps that more advanced methods, such as machine learning tools, might be able to make better predictions of the optimised model parameters.

Today, artificial intelligence (AI) is at the heart of many technologies. Broadly speaking, AI encompasses the field of developing computers and robots capable of behaving in ways that both mimic and exceed human capabilities. Machine learning (ML), a subcategory that grew out of the field of artificial intelligence, allows machines to acquire human like intelligence without explicit programming (Das et al. 2015). This subcategory of AI uses algorithms to automatically learn insights and recognize patterns from data and apply that learning to make ever-improving decisions. These techniques can provide a more advanced approach to recognize patterns and correlations between multiple parameters.

Cone Penetration Testing (CPT) is an important (in situ) soil investigation technique, where the resistance (q_c), sleeve friction (f_s) and optionally pore water pressures (u) are measured by pushing a cone into the soil. The technique is widely used because it is relatively fast, cost-effective and gives a good first impression of the subsoil, as is the determination of soil properties such as: unit weight (γ), water content (w) and void ratio (e_0). It would be advantageous to correlate these relatively easy to obtain parameters/properties to the harder to obtain soil/model parameters.

1.2.1 Hypothesis

The 3073 triaxial tests available provide an interesting and unique opportunity to carry out a comprehensive analysis.

It is expected that the stress-strain paths measured during these tests, which describe the mechanical behaviour of the soil, could be used to optimise advanced constitutive model parameters.

After processing and optimising the data, a fully formed database consisting of soil properties, soil parameters, constitutive model parameters and CPT parameters is established. When using advanced data analysis techniques, (new) statistics and correlations might be discovered.

Machine learning models will be used to predict the soil and optimised constitutive model parameters using soil properties as input. Furthermore, it is expected that an increasing number of input parameters will improve the performance of the models.

1.3 Research questions

This study is conducted with the aim of answering the following research question:

How can a database of triaxial tests contribute to the parameter determination, optimisation and selection for advanced constitutive models?

To support answering the main research questions, some sub-questions are formulated:

- Which types of (advanced) data analysis techniques can be used to extract the most value out of the database?
- What would be an appropriate strategy for optimising advanced constitutive model parameters using the measured stress-strain path from the laboratory?

- Which different aspects of soil behaviour can be defined/grouped and which groups can be extracted from the database?
- What (new) statistics and correlations between soil properties, soil parameters, constitutive model parameters and CPT parameters can be obtained by using (advanced) data analysis techniques?

1.4 Research method

A large set of triaxial and CPT data used for this research, which was conducted across the Netherlands, has been made available for this study by Fugro. All processing and manipulating of data will be done using the programming language Python (Van Rossum and Drake 2009) in combination with the SoilTest facility from the PLAXIS software. This facility offers the possibility to simulate laboratory tests using constitutive models.

The triaxial tests were performed according to NEN-EN-ISO 2018*a* and NEN-EN-ISO 2018*b* or one of its predecessors. Each test is stored in a separate text file and the format of this file has changed over the years. Before this data can be used for further analysis, it must first be compiled into a well-structured database, this is done using the Python *Pandas* package (McKinney et al. 2010). In addition to the data collected during the test itself, information of the soil sample is also available in the text file, such as the project number, classification, depth it was taken from, specifications of the sample (e.g., mass (m) and volume (V)) etc.

The soil properties and soil parameters are determined from the triaxial tests and constitutive model parameters are optimised by fitting the triaxial measurements generated by the constitutive model to that of the measurements from the laboratory with an optimisation algorithm. Sample locations are traced back to the borehole from which they were taken and are linked to nearby CPTs. In the process, CPT parameters corresponding to the soil layer of which the sample was taken from are obtained. At this stage, a fully formed database with all required parameters is available for further analysis.

Different types of aspects of soil behaviour are defined/grouped for which parameters sets with associated statistics are established. Correlations between these different types of parameters are made with methods such as linear regression, exponential regression, and more advanced machine learning techniques. Because of the large number of parameters and existing correlations with more simplistic regression methods, such as linear regression, the focus of this study will be on the more advanced machine-learning techniques. Well-known Python packages like *Numpy* (Oliphant et al. 2006), *SciPy* (Virtanen et al. 2020) and *Scikit-learn* (Pedregosa et al. 2011) are used for these analyses.

1.5 Scope of work

This research includes the following:

- A review of literature on topics such as triaxial tests, constitutive models, parameter determination, data analysis techniques, machine learning, etc.
- Preprocessing of the available data into a well-structured database.
- Parameter determination, derivation, calculation, and optimisation using Python and the PLAXIS SoilTest facility.
- Obtaining (new) statistics and correlations by using advanced data analysis techniques such as machine learning.

This research will not include:

- Other laboratory tests than the triaxial test.
- Constitutive models other than HS small.
- An in-depth study on the mathematical background and formulation of machine learning models.

1.6 Report layout

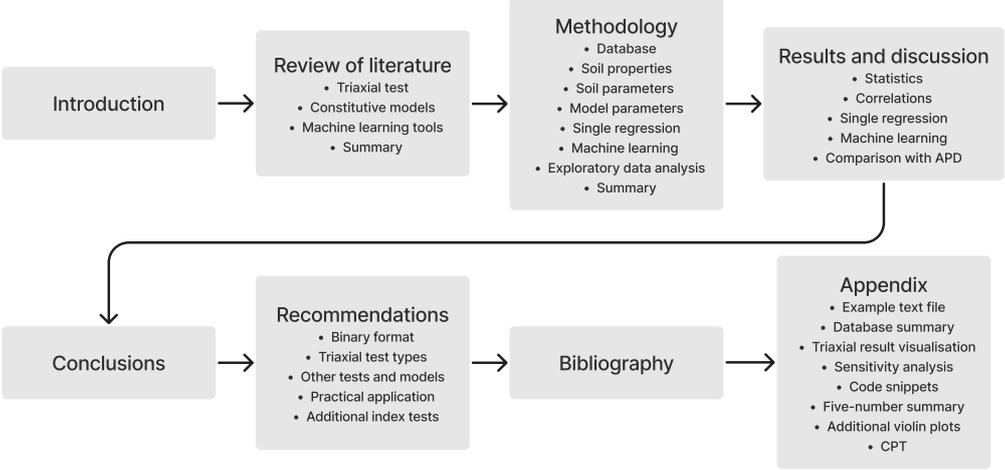


Figure 1.1: Reading guide

Chapter 2

Review of literature

This chapter provides an overview of existing literature on topics like: the triaxial test, constitutive models and machine learning tools. A short summary is provided in the end which also elaborates on how this will be applied in this research.

2.1 Triaxial test

The triaxial test is a laboratory tests which is used in Geo-Engineering to measure the mechanical properties of the soil, a schematic diagram of a typical triaxial apparatus is shown in Figure 2.1. The cylindrical soil specimen generally has a length to diameter ratio of 2 and is enclosed by a rubber membrane. The top and bottom of this membrane are connected to circular plates enclosed by o-rings to secure a water tight connection. Several devices are connected to measure, regulate or apply; cell pressure (σ_c), pore water pressure (u) and deviatoric stress (q). The cell is filled with water and is therefore applying pressure to the sample from all directions, hence a pressure equal to the cell pressure is generated in the specimen (Verruijt and Van Baars 2007). The test should involve the loading of a sample at several different cell pressures. Further steps are dependent upon the type of test that is chosen. This section elaborates on the different test types and how parameters are derived.

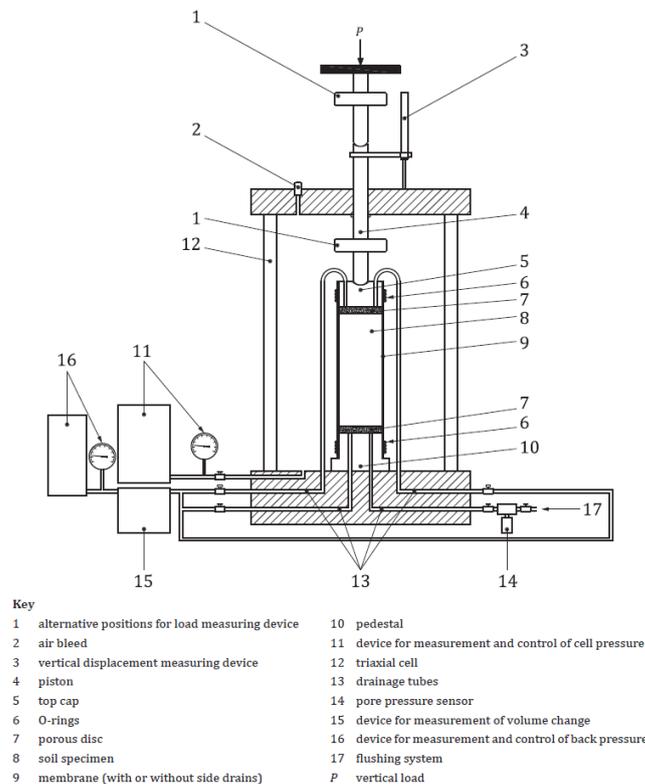


Figure 2.1: Triaxial apparatus NEN-EN-ISO 2018b

2.1.1 Consolidated

The consolidated test type allows the specimen to drain and let excess pore water pressure dissipate during the applying of the cell pressure. The valve is open and sufficient time is taken such that the excess pore pressure is reduced to zero. The back pressure is used to ensure that the sample is fully saturated. However, checks need to be performed prior to both the consolidation phase and the shearing phase. The degree of saturation can be expressed in terms of Skempton's pore pressure parameter (B) Skempton 1954. B is calculated according to Equation 2.1 and has to be ideally one for full saturation although above 0.95 is generally accepted as well.

$$B = \frac{\Delta u}{\Delta \sigma_3} \quad (2.1)$$

After saturation and consolidation the final phase starts, the shearing phase. This can be done either undrained or drained. In undrained situation the valve is closed and excess pore pressure is allowed to build up (CU).

When performing drained shearing, the valve is opened to allow excess pore pressure to dissipate (CD). Do note that excess pore pressures can still develop when the loading goes fast with respect to the consolidation time of the sample. Such a test usually takes a rather long time and it is important to determine an appropriate strain rate upfront. An elaboration on the parameters and method of determination is given in the following paragraphs.

Cohesion and friction angle

The cohesion and internal friction angle are strength parameters, which can also be described as the resistance against failure. In engineering practices they are commonly determined at either 2%, 5% axial strain (ε_{ax}) or at failure. The NEN-EN-ISO 2018b offers two ways of determining these parameters from the triaxial test results, see Figure 2.2.

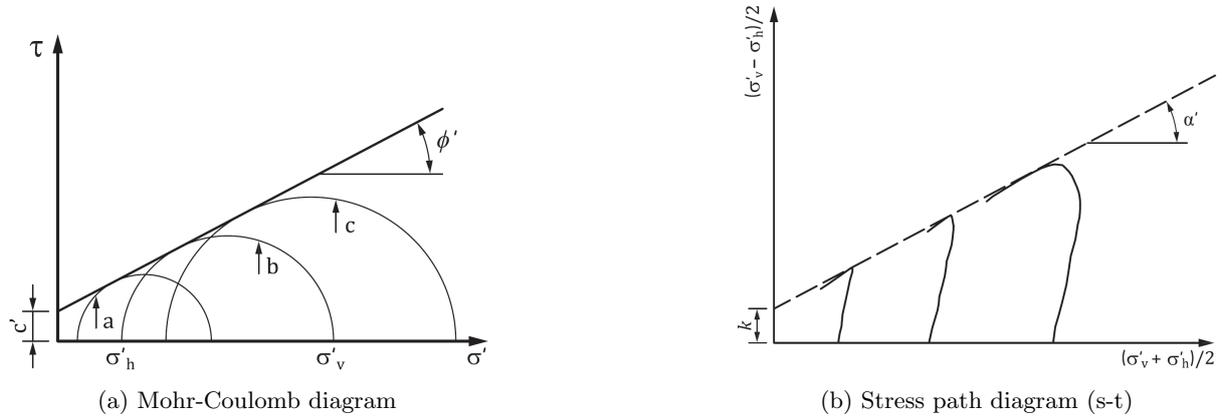


Figure 2.2: Cohesion and internal friction angle NEN-EN-ISO 2018b

The Mohr-Coulomb diagram, Figure 2.2a, can determine the parameters directly by fitting a line through the circles, these circles can be determined for different strain levels.

Figure 2.2b shows an alternative method to present the stress path based on Mohr's circle, also known as the s-t stress space. Where s and t (Equation 2.2 and 2.3) are the centre and radius of the Mohr's circle, respectively, and represent the mean stress and maximum shear stress, respectively. This method was developed by Professor T. W. Lambe of the Massachusetts Institute of Technology (Lambe 1967). Unlike the Mohr-Coulomb diagram, the parameters k and α' , first need to be converted in order to obtain c' and ϕ' . This is shown in Equations 2.4 and 2.5.

$$s = \frac{\sigma_1 + \sigma_3}{2}; \quad s' = \frac{\sigma'_1 + \sigma'_3}{2} \quad (2.2)$$

$$t = \frac{\sigma_1 - \sigma_3}{2} = \frac{\sigma'_1 - \sigma'_3}{2} \quad (2.3)$$

$$\sin \phi' = \tan \alpha' \quad (2.4)$$

$$c' = \frac{k}{\cos \phi'} \quad (2.5)$$

An alternative representation would be to use the mean of the three principal effective stresses instead of the mean of the major and minor principal stresses. This representation is known as the Cambridge stress path or the p-q stress space (Roscoe et al. 1958) and is illustrated in Figure 2.3. The mean effective stress (p') and deviatoric stress (q) can be expressed as is shown in Equation 2.6 and 2.7, the parameters ϕ' and c' are calculated with Equation 2.8 and 2.9 respectively.

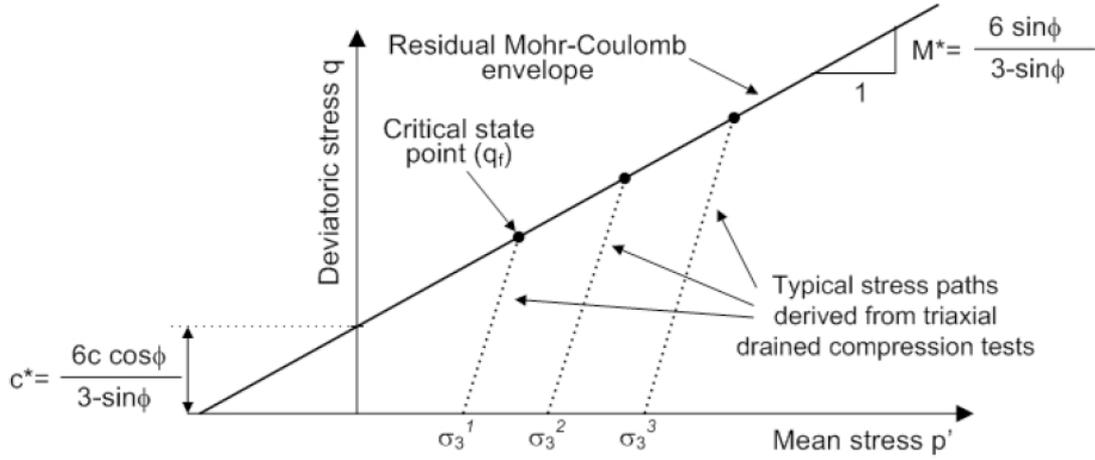


Figure 2.3: Stress path diagram (p'-q) Obrzud et al. 2018

$$p = \frac{\sigma_1 + 2\sigma_3}{3}; \quad p' = \frac{\sigma'_1 + 2\sigma'_3}{3} \quad (2.6)$$

$$q = |\sigma'_1 - \sigma'_3| = |\sigma_1 - \sigma_3| \quad (2.7)$$

$$\phi' = \sin^{-1} \left(\frac{3 * M^*}{6 + M^*} \right) \quad (2.8)$$

$$c' = c^* \frac{3 - \sin \phi'}{6 \cos \phi'} \quad (2.9)$$

A comparison between the two methods is presented in Figure 2.4, for both the Total Stress Path (TSP) and Effective Stress Path (ESP).

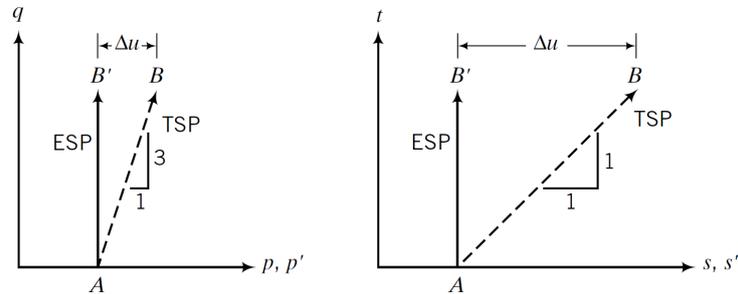


Figure 2.4: Stress path comparison

Stiffness and power for stress dependency of stiffness

Whereas strength parameters say something about resistance against failure, stiffness says something about the resistance to deformations. The stiffness (E) of sand is generally higher than that of clay, furthermore, the stiffness of sand shows less stress dependency (m). The stiffness can be determined

from a standard triaxial test by drawing a secant line in the stress-strain graph (Figure 2.5), and can be calculated with Equation 2.10. In engineering practices it is common to determine the stiffness at 50% of the peak deviatoric stress. The rate of stress dependency of stiffness can be calculated with Equation 2.11, note that at least two triaxial tests at different confining pressures are required.

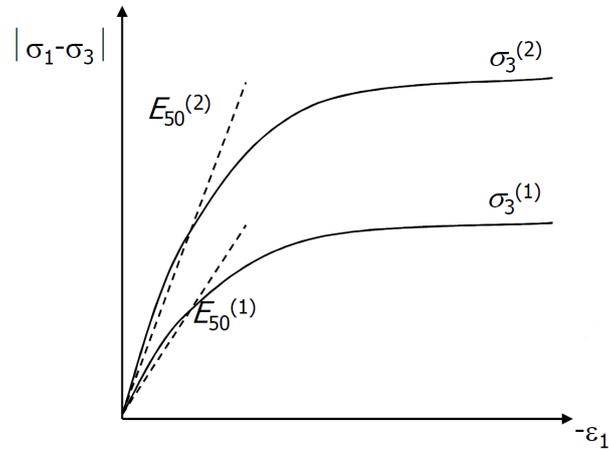


Figure 2.5: Stiffness for different confining pressures

$$E = \frac{q}{\varepsilon_{ax}} \quad (2.10)$$

$$m = \frac{\ln(E_{50}^{(1)}/E_{50}^{(2)})}{\ln(\sigma_3^{(1)}/\sigma_3^{(2)})} \quad (2.11)$$

2.1.2 Unconsolidated

Another possibility is to never allow drainage at all, but to perform it at in-situ stress conditions, meaning, the sample is directly axially loaded. This test is denoted as Unconsolidated Undrained (UU). If a UU test is done at a higher cell pressure, the difference with the first test will be that the pore pressures are higher, the effective stresses will be practically the same. This results in the same effective Mohr circles but different Mohr circles for total stresses, see also Figure 2.6. It is therefore not possible to determine effective strength parameters with this test, only the undrained shear strength (s_u) and it is calculated according to Equation 2.12. The reason that this test is conducted is because it is relatively fast in comparison to the other tests since there is no consolidation time required.

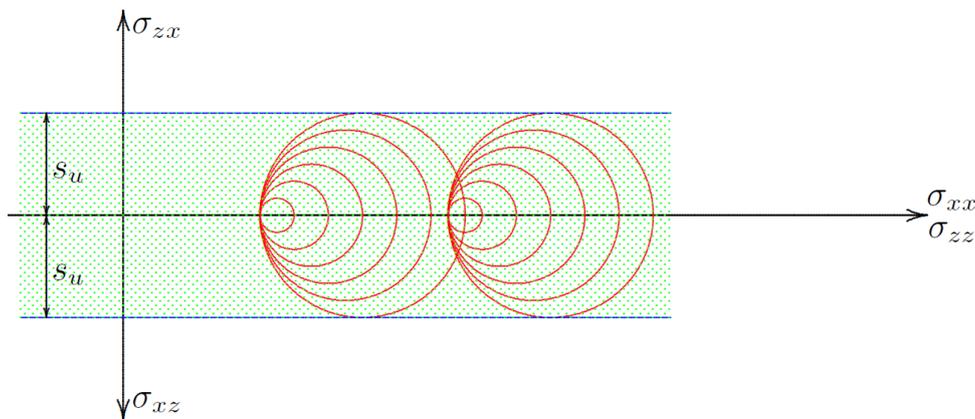


Figure 2.6: Unconsolidated Undrained triaxial test Verruijt and Van Baars 2007

$$s_u = \frac{\sigma_1 - \sigma_3}{2} \quad (2.12)$$

2.2 Constitutive models

Constitutive models define the mechanical behaviour of materials with a stress-strain relationship. One of the practical applications is in a continuum framework where they are used to simulate, analyse or predict the behaviour of materials under certain loading and boundary conditions using numerical methods. This section discusses three constitutive models in order to select a model for which the parameters will be optimised.

2.2.1 Linear Elastic Perfectly Plastic (LEPP) Mohr-Coulomb (MC) model

The Mohr-Coulomb model is a clear first order approach model and is known for its simple linear elastic perfectly plastic stress-strain relationship. This model is used as a first approximation of soil behaviour and can give a good representation of (drained) failure. The fact that the number of parameters is limited and a feature like dilatancy can be included makes it suitable for many practical applications. However, the limited amount of parameters is a limitation as well, because it therefore lacks a lot of essential soil features such as: stress-dependent stiffness, distinction between primary loading and unloading/reloading etc. These reasons not only make the model inaccurate but can also make it dangerous to use, due to all of its limitations and simplifications.

The linear elastic part is based upon Hooke's law of isotropic elasticity and the perfectly plastic part is based on the MC failure criterion, formulated in a non-associated plasticity framework (Smith et al. 2013). Irreversible strains are developed due to plasticity and the evaluation of when plasticity occurs is done by a so called yield function (f). This is a function of stresses and strains, and plastic yielding corresponds with the condition $f = 0$, the condition can be presented as a fixed surface in the principal stress space. The full MC yield condition consists of six yield functions in terms of principal stresses and is defined by the two parameters c' and ϕ' , a visual representation is shown in Figure 2.7. The direction of the plastic strain is defined by the plastic potential function (g). For non-associated plasticity, f and g are different, otherwise it will give too large plastic volumetric strains.

The model requires a total of five parameters which can be found in Table 2.1. Two elastic parameters and three plastic parameters.

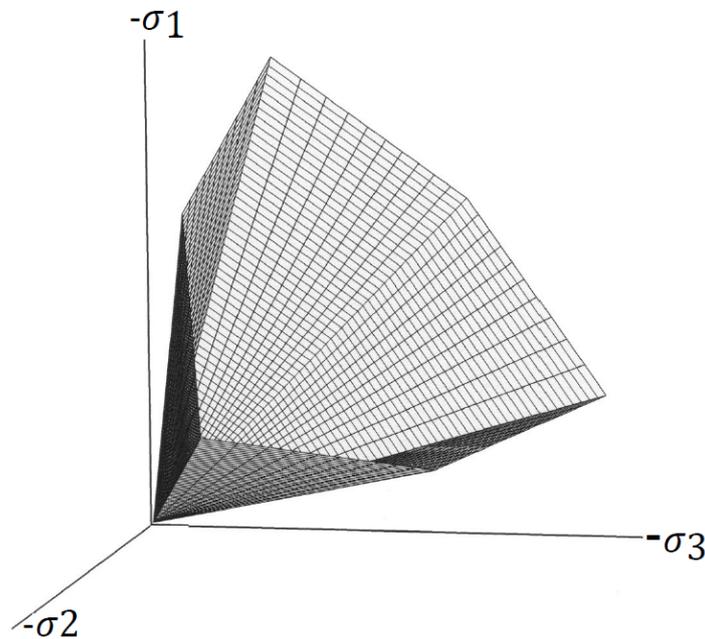


Figure 2.7: Yield contour of the Mohr-Coulomb model in the principal stress space Brinkgreve 2005

Parameter	Definition	Unit
E	Young's modulus	kN/m^2
ν	Poisson's ratio	-
c'	Cohesion	kN/m^2
ϕ'	Friction angle	$^\circ$
ψ	Dilatancy angle	$^\circ$

Table 2.1: LEPP MC parameters

2.2.2 Hardening Soil model with small-strain stiffness (HS small)

The Hardening soil model is a true second order model for soils in general, for any type of application (Brinkgreve 2005). In contrast to the LEPP Mohr Coulomb model, the yield surfaces of a hardening plasticity model are not fixed in the principal stress space, but they can expand due to plastic straining. Furthermore, it has a better non-linear formulation of soil behaviour (for both soft soils and harder types of soil), makes a distinction between primary loading and unloading/reloading, accounts for stress dependency of stiffness etc. Do note that the model still lacks features such as creep and anisotropy and that the model is unsuitable for very soft soils.

The original Hardening Soil (HS) model was based on the hyperbolic stress-strain relationship in drained triaxial loading, also know as the Duncan-Chang model (Duncan and Chang 1970). The HS model is based on hardening plasticity in contrast to the Hyperbolic model which is entirely based on elasticity theory. The HS model consists of two types of hardening, one being shear hardening, which is used to model irreversible strains due to primary deviatoric loading. While compaction hardening is used to model irreversible strains due to primary compression. Figure 2.8 visualises the Cone (shear hardening contour) and the Cap (compaction hardening) in the three dimensional stress space. The formulation and verification of this model can be found in Schanz et al. 1999.

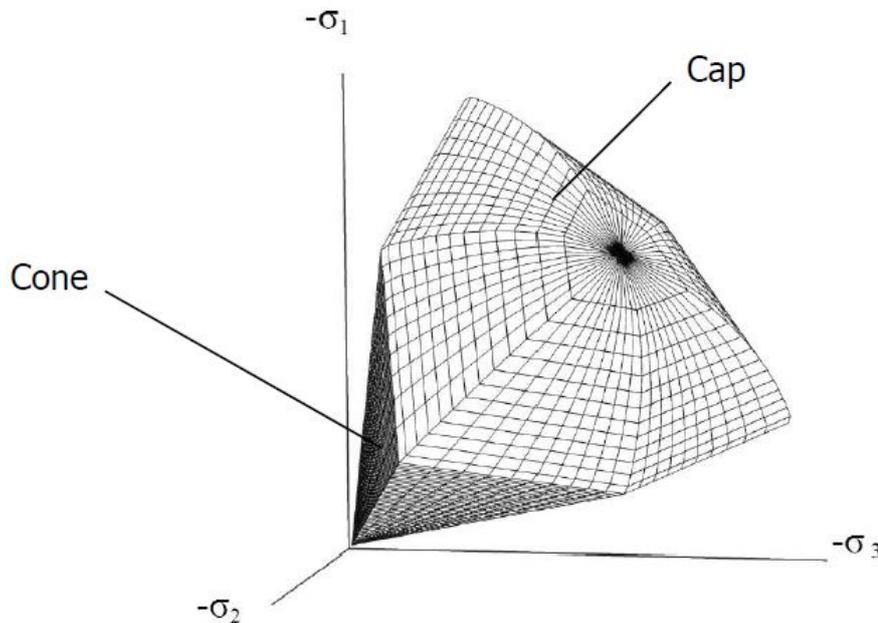


Figure 2.8: Yield contour of the Hardening Soil model in the principal stress space Brinkgreve 2005

An upgraded version for the HS model is called the Hardening soil model with small-strain stiffness (HS small) (Benz 2006). This additional feature is not based on plasticity theory, but is included in the elastic formulation as an overlay function. So now not only stress-dependency of stiffness is included, but also strain dependency of stiffness. Soil tends to behave very stiff at small strains and the stiffness decreases with the strain level. This can best be visualised in the modulus reduction curve where the secant shear modulus (G_s) is plotted as a function of shear strain (γ) on a log scale (Figure 2.9a).

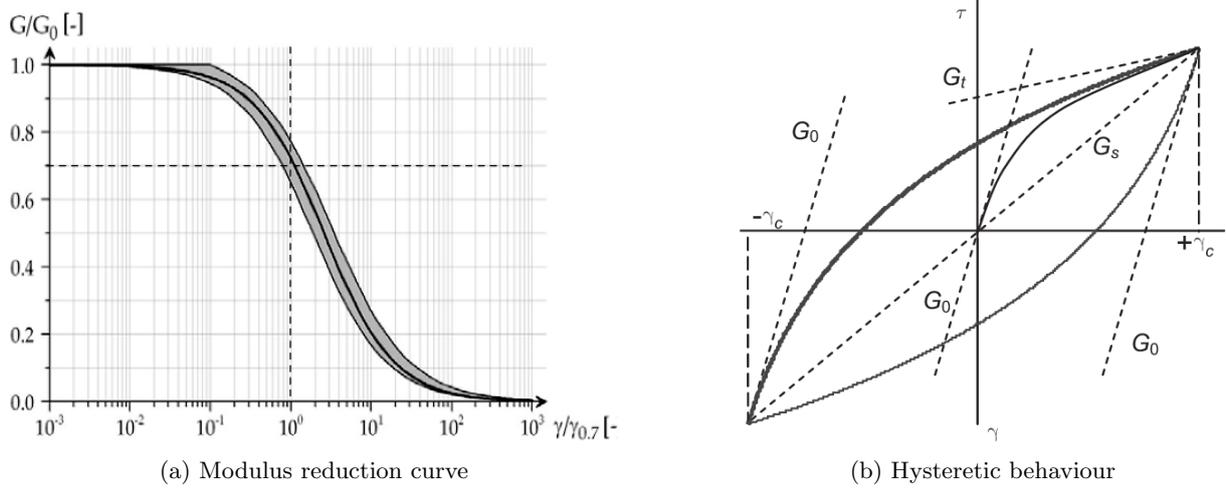


Figure 2.9: Small-strain stiffness Brinkgreve et al. 2007

The small band can best be described with the hyperbolic Equation 2.13. This equation involves two additional parameters with respect to the HS model, the small-strain shear modulus (G_0) and the shear strain at which the secant shear modulus has been reduced to approximately 70% of G_0 ($\gamma_{0.7}$). Relating stresses to strains, requires a tangent stiffness formulation (G_t), which can be done by taking the derivative and is given in Equation 2.14. The lower bound of G_t is given in Equation 2.15.

$$G_s = \frac{G_0}{1 + 0.385\gamma/\gamma_{0.7}} \quad (2.13)$$

$$G_t = \frac{G_0}{(1 + 0.385\gamma/\gamma_{0.7})^2} \geq G_{ur} \quad (2.14)$$

$$G_{ur} = \frac{E_{ur}}{2(1 + \nu_{ur})} \quad (2.15)$$

An important element in the HS small model is that upon full strain reversal, the stiffness restarts at the small-strain stiffness (G_0), hence, the model allows for hysteresis. Hysteresis is a phenomena which occurs under cyclic loading and is visualised in Figure 2.9b. The enclosed area represents the energy dissipation and depends upon the magnitude of the shear strain. In dynamic calculations, such as vibrations, the hysteresis and associated energy dissipation leads to damping. A complete overview of the parameters is given in Table 2.2.

Parameter	Definition	Unit
E_{50}^{ref}	Secant stiffness in standard drained triaxial test	kN/m^2
E_{oed}^{ref}	Tangent stiffness for primary oedometer loading	kN/m^2
E_{ur}^{ref}	Unloading / reloading stiffness	kN/m^2
G_0^{ref}	Reference shear modulus at very small strains ($\varepsilon < 10^{-6}$)	kN/m^2
$\gamma_{0.7}$	Threshold shear strain at which $G_s = 0.722G_0$	-
m	Power for stress-level dependency of stiffness	-
p^{ref}	Reference stress for stiffnesses	kN/m^2
ν_{ur}	Poisson's ratio for unloading/reloading	-
c'	Cohesion	kN/m^2
ϕ'	Friction angle	°
ψ	Dilatancy angle	°
R_f	Failure ratio q_f/q_a	-
K_0^{nc}	K_0 -value for normal consolidation	-

Table 2.2: HS small parameters

2.2.3 Soft Soil model

The Soft Soil model is between the MC model and the HS small model in terms of how advanced it is. This model is used for near-normally consolidated (NC) fine grained types of soil (clay's, clayey silts and peat), which generally have a high degree of compressibility. It shares a lot of (dis)advantages to that of the HS small model, but it is the soft soils where this model is chosen over the more advanced HS small model. This model is not suitable for: other types of soil, highly sensitive fine grained soils like quick clay or over-consolidated (OC) soils.

The total yield contour is presented in Figure 2.10 and the cap can be pushed out just like the HS model resulting in plastic straining, but the failure surface is fixed. The parameters for the Soft Soil model are given in Table 2.3

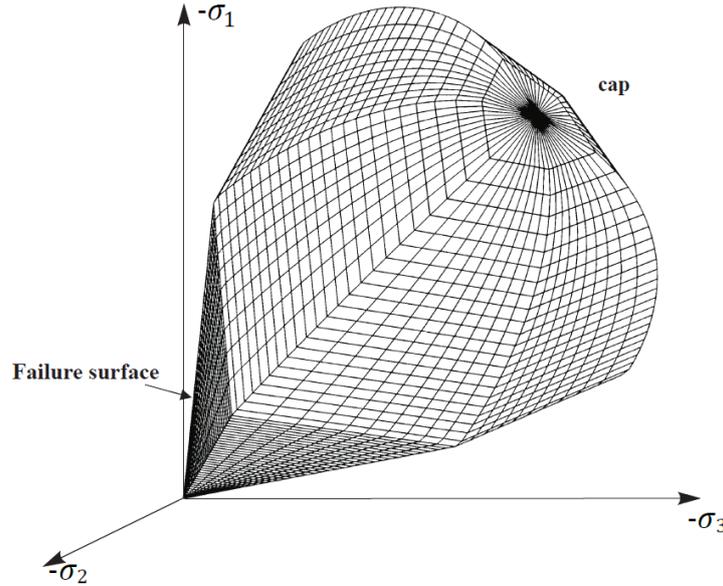


Figure 2.10: Yield contour of the Soft Soil model in the principal stress space

Parameter	Definition	Unit
λ^*	Modified compression index	-
κ^*	Modified swelling index	-
ν_{ur}	Poisson's ratio for unloading / reloading	-
c'	Cohesion	kN/m^2
ϕ'	Friction angle	$^\circ$
ψ	Dilatancy angle	$^\circ$
K_0^{nc}	K_0 -value for normal consolidation	-

Table 2.3: SS parameters

2.3 Machine learning tools

Artificial intelligence (AI) plays a central role in numerous technologies today. It encompasses the development of computers and robots that can imitate and surpass human capabilities. Within the field of AI, machine learning (ML) has emerged as a sub field that enables machines to develop their own algorithms/models without being explicitly programmed (Das et al. 2015). ML employs algorithms to automatically extract insights, identify patterns, and continuously enhance decision-making through learning by leveraging statistical and mathematical principles. These techniques offer an advanced approach to recognizing patterns and establish correlations among multiple parameters. They are used in several fields like healthcare to predict disease diagnosis, banking for fraud prevention etc. ML algorithms are organised into taxonomy, based on the desired outcome of the algorithm, some common algorithm types are listed down below (Ayodele 2010) and summarised with their associated subcategories in Figure 2.11.

- Supervised learning: where the algorithm generates a function that maps inputs to labelled outputs.
- Unsupervised learning: models a set of inputs of which no labelled outputs are available, attempts to find similarities, patterns and differences.
- Semi-supervised learning: uses both labelled and unlabelled data to generate an appropriate function or classifier.
- Reinforcement learning: this works on a feedback-based process in which an AI agent gets rewarded for each good action and penalties for each bad action.

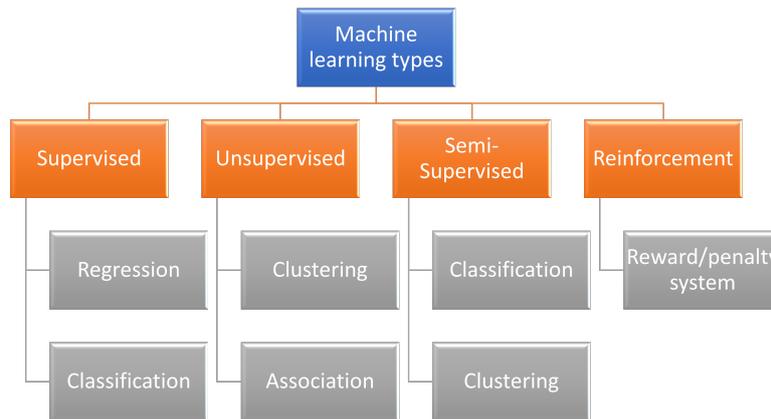


Figure 2.11: Machine learning types with subcategories (Sarker 2021)

The most relevant category for this research is supervised learning, some of the most popular methods will be further elaborated. It is a fundamental technique that enables computers to learn from labelled training data and make predictions on unlabelled data Cord and Cunningham 2008. The reason it is called supervised is because the model needs a "supervisor" to provide it with the correct target answers. The goal is to learn the relationship between input parameters and labelled output parameters. Supervised learning involves two primary techniques: regression and classification. Regression techniques predict a *continuous* numerical output variable. For example, predicting housing prices based on features like area, number of bedrooms, and location. Classification problems involve predicting a *discrete* output variable that falls into predefined classes or categories. For example, classifying emails as spam or non-spam based on their content.

2.3.1 Single regression techniques

The most common and easy to use technique is the linear regression method, it fits the most optimal straight line through the data which is illustrated in Figure 2.12a. A linear regression model expands upon the idea of linear correlation and formalizes a statistical relation between the two variables such that y is linearly related to x (Eberly 2007). The best fit is created by minimising the sum of the squared error, which therefore also minimises the root mean squared error, this is referred to as the least squares approach. Linear regression results come in the form of a function (Equation 2.16) in which two coefficients are determined, a and b (and the error e). The equation is fitted to the measured data and it is now possible to make predictions on the dependent variable y with independent x inputs.

$$y = ax + b + (e) \quad (2.16)$$

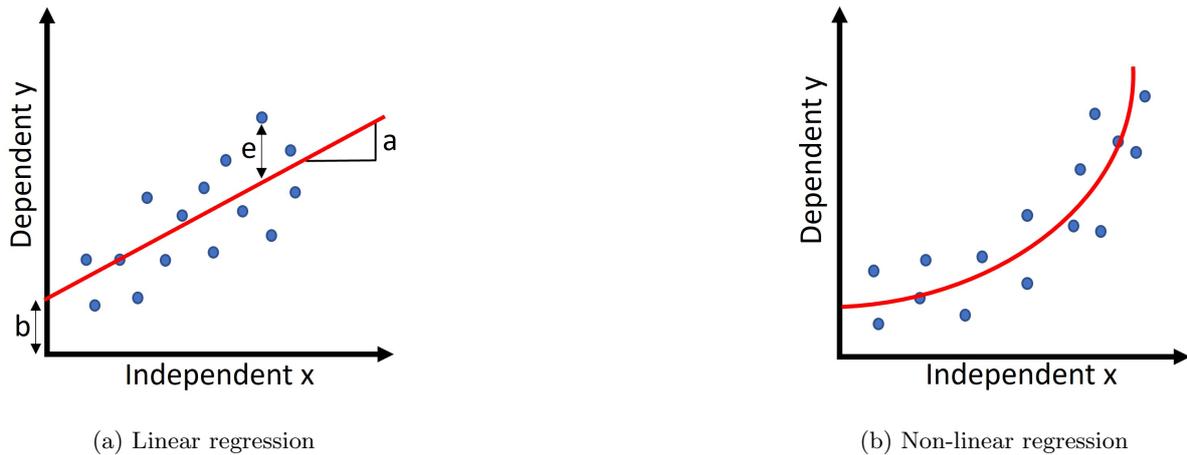


Figure 2.12: Single regression techniques

Note that it does not always make sense to fit a straight line through the data. When looking at Figure 2.12b. It would make more sense to fit a non-linear line through this data set. There are multiple non-linear fitting techniques, like increasing the order of the polynomial to be fitted (linear regression is a first order polynomial). It makes sense that the fit would be better using a higher order since it now has three parameters which can be fitted instead of two. Alternatively one can use an exponential function which is presented in Equation 2.17.

$$y = be^{ax} + (e) \quad (2.17)$$

2.3.2 Multiple linear regression

The two regression techniques discussed above both compare two parameters with each other, one dependent and one independent. Often several independent variables affect one continuous dependent variable and it is of interest to describe the combined associations of these inputs on the output. This is where a technique such as multiple linear regression can be used, which is basically an extension of the simple linear regression model (Eberly 2007). It formalises a simultaneous statistical relation between the single continuous outcome y and the predictor variables $x_k (k = 1, 2, \dots, p - 1)$ which is given in Equation 2.18.

$$y = a_1x_1 + a_2x_2 + \dots + a_{p-1}x_{p-1} + b + (e) \quad (2.18)$$

2.3.3 Tree-based methods

Decision trees are simple and understandable models that predict outcomes by repeatedly dividing the input space based on different features. A decision tree structure consists out of nodes and edges, the top node is called the root node which represents the entire data set. From this root node branches or edges extent to subsequent nodes (internal nodes), representing the decisions based on feature values. The final nodes are called the leaf nodes and these contain the predicted output. At each internal node a decision is made on how to split the data on a specific feature and a threshold value, the algorithm aims to reduce the error by gaining this additional information due to splitting the data set. This process stops when one of the following criteria is met: maximum depth of the tree is reached, having a minimum number of samples in the nodes or when further splits do not result in improving the model's performance. After the tree is built, predictions can be made by giving the tree a new data point, it then follows the path from root node to leaf.

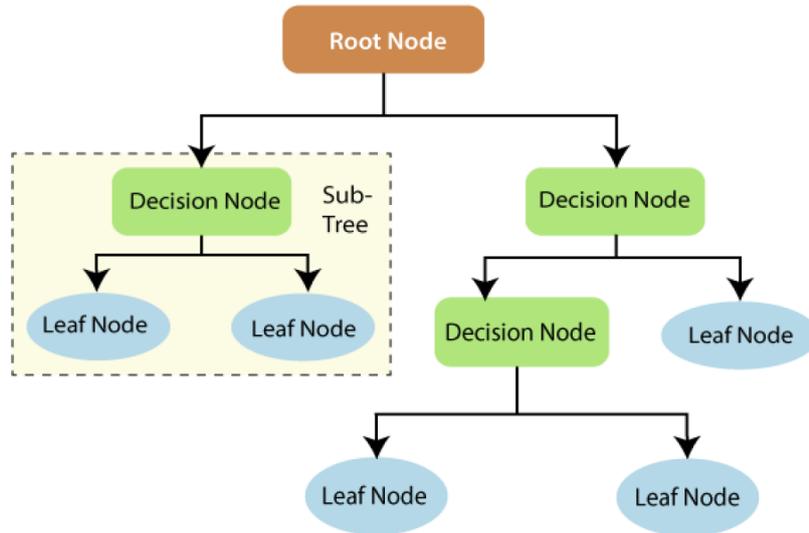
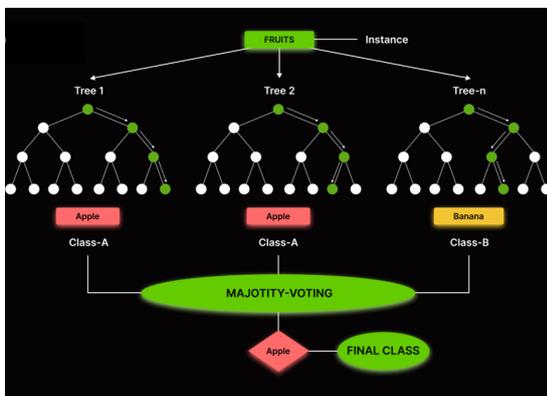
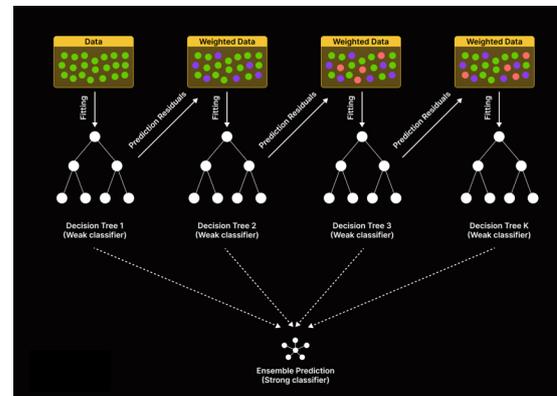


Figure 2.13: Decision tree (Charbuty and Abdulazeez 2021)

Using a single decision tree often results in relatively poor results, a powerful extension is the so-called random forest (Breiman 2001), which basically consists of multiple decision trees (Figure 2.14a). It aims to improve predictive accuracy and reduce over-fitting. Each decision tree in the random forest is trained on a random subset of the training data and a random subset of the available features, this randomness introduces diversity among trees. The original process of a random forest is parallel, while boosting methods such as Gradient Boosting built decision trees sequentially. In doing so, the algorithm focuses on correcting mistakes made by previous trees (Figure 2.14b). A newer version called eXtreme Gradient Boosting (XGBoost) (Chen and Guestrin 2016) excels in model performance and computational speed in comparison to the standard version, this algorithm has become one of the most popular algorithms in the ML branch. Note that these models are more advanced and become harder to understand in comparison to the decision tree.



(a) Random forest



(b) Gradient Boosting

Figure 2.14: Tree based algorithms (Lev 2022)

2.3.4 Artificial Neural Network

An Artificial Neural Network (ANN) is an algorithm with a completely different methodology than the previous discussed tree based methods. The ANN is inspired by the structure and functioning of the human brain. The development of this algorithm has a rich history, McCulloch and Pitts 1943 introduced the first model of an artificial neuron in 1943 which laid the foundation for further developments. At the core of a neural network are artificial neurons (also called nodes). These neurons are connected to each other and are organised in layers, the input layer, a specified amount of hidden layers and the output layer (Figure 2.15a).

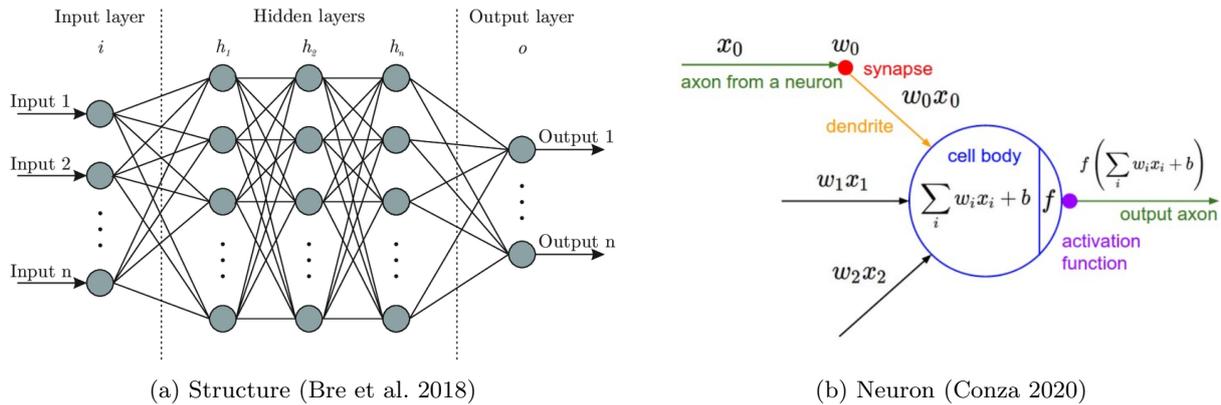


Figure 2.15: Artificial Neural Network

Figure 2.15b zooms in on a single neuron from the structure. The connection between neurons is associated with weights, representing the importance of that connection with respect to the other inputs. Subsequently, the node applies an activation function to the weighted sum of the inputs which transforms the analysis from linear to non-linear. The activation function basically decides whether a neuron should be activated or not, meaning, it checks if the contribution is important to the network. There are several activation functions available of which some of them are shown in Figure 2.16. The output is passed onto the next layers and a similar procedure is applied till the output layer. This process is called feedforward propagation and it generates predictions based on the learned parameters of the network. Subsequently, the neural network learns from training data through a process called back-propagation. Instead of starting at input layer, the algorithm starts at the output layer and compares the predicted value to the actual value and quantifying the error in the process. The training process of this model is done by iteratively adjusting the weights to minimise the error. Backward propagation calculates the gradient of the loss function with respect to the weights, allowing for weight adjustments to improve the performance of the neural network.

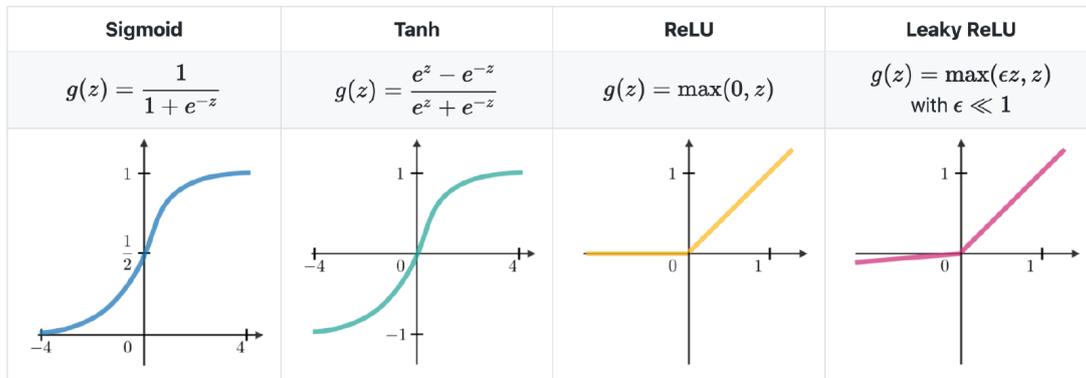


Figure 2.16: Common activation functions (Conza 2020)

2.3.5 (Kernel) Ridge Regression

Regular Ridge Regression (RRR) is a linear technique that solves the problem by introducing a penalty term to the loss function, it aims to minimise the sum of the squared error and regularization is given by the l_2 -norm. RRR works well when the data consists out of linear relationships and has a large number of features compared to the sample size. On top of that, it is computationally fairly efficient. However, it cannot capture the more complex non-linear behaviour.

The Kernel Ridge Regression (KRR) combines Ridge Regression with the kernel trick. This kernel enables the algorithm to operate in a high-dimensional feature space without explicitly computing the coordinates of the data in this new space. The choice of the kernel function, like Sigmoid or polynomial, enables for non-linear relationships to be captured. This does however make the algorithm computationally more expensive. The KRR is actually very similar to the Support Vector Regressor (SVR) which also makes use of the kernel trick, they differ in the type of loss functions which are used. The fitting

time of the KRR is typically faster for medium-sized data sets (Figure 2.17). The SVR on the other hand is faster with its predictions times since the learned model is non-sparse.

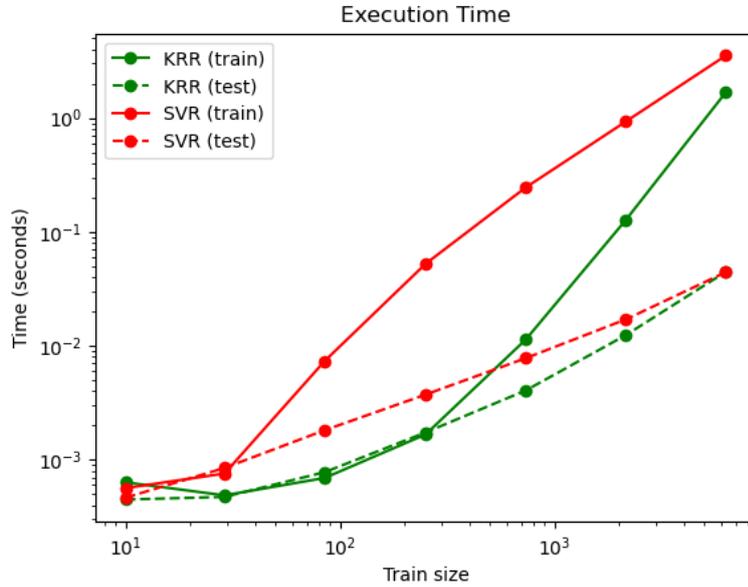


Figure 2.17: KRR vs SVR execution time (Metzen 2023)

2.3.6 Evaluation metrics

Regression is a fundamental concept in machine learning, and statistical modelling in general that involves predicting continuous numerical values based on the relationships between variables. Once a regression model is trained and applied to make predictions, it is crucial to interpret and evaluate its performance. Several different metrics are evaluated to maintain transparency over the regression results and assess the goodness of the fit/match.

The Mean Absolute Error (MAE), which is calculated as the sum of the absolute errors divided by the sample size, is shown in Equation 2.19. In which $|e_i|$ is denoted as $y_i - \hat{y}_i$, y_i being the measured value and \hat{y}_i being the fitted value. A somewhat similar method is the Root Mean Squared Error method which is calculated in Equation 2.20. Both these techniques calculate the error/residual between the actual data and the predicted data. The results can vary from 0 to infinite and are negatively orientated, meaning the lower the score, the better the fit. There is a lot of discussion on which of the two should be chosen when validating a model. Willmott and Matsuura 2005 argues that the RSME is not a good indicator of the average model performance and thus the MAE would be a better metric, while Chai and Draxler 2014 states that the RSME is more appropriate than the MAE when the error distribution is expected to be Gaussian (normal distribution).

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (2.19)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (2.20)$$

The coefficient of determination (r^2), also known as the squared multiple correlation coefficient, is another use-full index to evaluate regression results (Equation 2.21). A perfect fit would result in an r^2 of 1, this can easily be evaluated, when the actual data is the same as the predicted data, the numerator will be 0. So in comparison to the previous methods, this one does not give a score of the residual, but rather the goodness of the fit itself. Counter-intuitive, this method can give negative results as well. This may occur in case of linear regression without an intercept (Barten 1987), or when a non-linear function is fitted (Cameron and Windmeijer 1997). Latuni 2019 presented an indicative guide for the qualitative expression of the coefficient of determination which is given in Table 2.4.

$$r^2 = 1 - \frac{SS_{res}}{SS_{tot}} = \frac{\sum_{i=1}^n e_i}{\sum_{i=1}^n (y_i - \bar{y}_i)} \quad (2.21)$$

Range	Correlation
< 0.04	Very low
0.05 - 0.16	Low
0.17 - 0.49	Medium
0.50 - 0.81	Strong
> 0.82	Very Strong

Table 2.4: Coefficient of determination guidelines Latuni 2019

One of the problems with the methods described above is that the y values need to be given on the same x values. If this is not the case, interpolation techniques would need to be used in order to obtain y values which can be compared. Alternatively, one might use the Robust and interpolation-free technique (RIFT), which prevents interpolation errors and was introduced by Lin et al. 2015. RIFT considers the deviations between two data sets or two curves in terms of the area enclosed by these two curves and bounds of interest, as illustrated by Figure 2.18. For every point in the first data set a triangle is formed by using two other points from the second data set. The area of these triangles can be calculated using the coordinates, by summing all the triangles an indication of the total area between the data sets can be given.

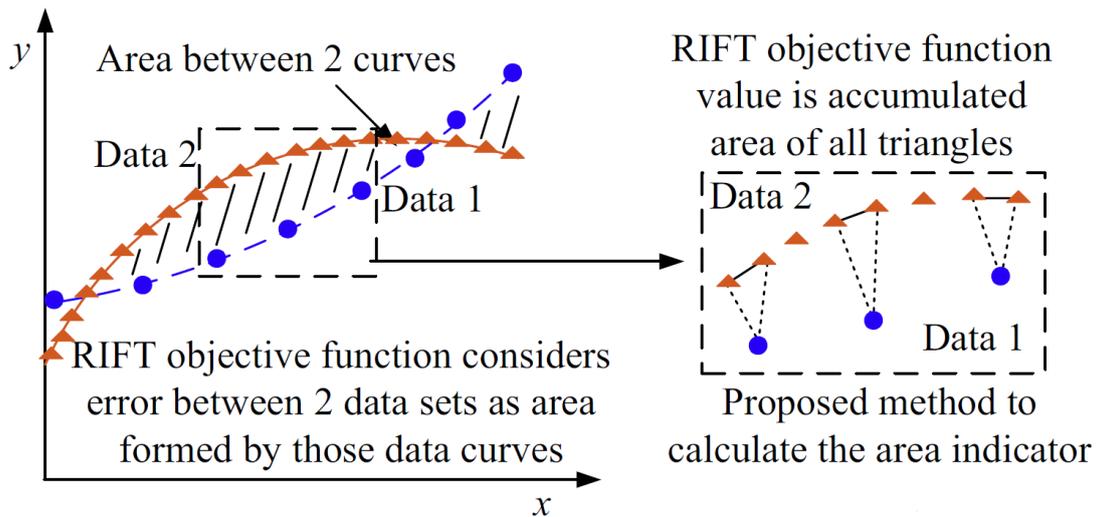


Figure 2.18: RIFT method Lin et al. 2015

2.3.7 Machine learning in Geotechnical engineering

Soil is a non-linear, inhomogeneous and complex material to work with and modeling the behaviour of such a material has proven to be complicated. Machine learning methods have gained increasing popularity in the past decade due to their predictive capabilities. Which is why in the past decade, multiple studies on Geo-engineering related topics in combination with machine learning have been conducted. For example, Yu 2022, attempted to use CPT parameters to predict the undrained shear strength. Duffy 2019 did something similar but instead aimed to predict oedometer test results. Both came to similar conclusions, that machine learning has great potential in predicting parameters. However, both faced the same problem, namely insufficient data to train the models. The more unpredictable the material is, the more data is required to make use of machine learning models. The heterogeneity of soil, results in requiring large sums of data to be able to train it.

2.4 Summary

This chapter serves as a comprehensive literature review, playing a crucial role in providing fundamental background information for this research. Broadly speaking, three main topics have been discussed: the triaxial test, constitutive models, and machine learning tools.

Section 2.1 presents a description of the triaxial apparatus, test procedure, post processing of the data, and the parameter determination. These techniques are used later on to derive the parameters from all of the available triaxial test data.

The LEPP MC model, HS small model and SS model have been elaborated in Section 2.2 by describing the model, their applicability, pros and cons, and required parameters. This was done to select a constitutive model for which the parameters will be optimised, HS small will be used.

Today, there are a variety of machine learning models available. Ranging from the simple regression techniques to the more advanced random forests and neural networks. Section 2.3 provides background information on some of these machine learning tools which was required to perform the advanced data analysis.

Chapter 3

Methodology

3.1 Database description

3.1.1 General

Over the past decades lots of triaxial tests on soil samples from across the Netherlands have been conducted by Fugro according to the NEN-EN-ISO 2018*a* and NEN-EN-ISO 2018*b* standards, or one of its predecessors. This available data offers an interesting opportunity to perform an extensive analysis on and research (new) statistics and correlations. Before any analysis is possible, the data first needs to be collected and processed into a well-structured database. On top of that a clear understanding of the database is required as well, since the size and complexity make it rather difficult to asses how the data can be put to good use.

All triaxial text files were extracted from the Fugro SharePoint and collected in a single folder. The total amount being 20831, generally speaking, three text files are required for one complete triaxial test. Some key features which can be found in the text files are presented in Table 3.1 and an example text file can be found in Appendix A.

Database name	Description	Unit
Project	Fugro's internal project number	-
Borehole	Borehole number	-
Sample	Sample number	-
Stage	Stage number	-
Test depth	Depth from which the sample was taken	<i>m</i>
Test type	The type of triaxial test	-
Sample type	Method of preparation of the sample	-
Classification code	Visual description of the sample	-
Particle density	Often assumed	<i>g/cm³</i>
Mass	Initial, final and dry mass	<i>g</i>
Volume	Initial, after consolidation and final volume	<i>ml</i>
Date and time	Measured during testing	yyyy-mm-dd hh:mm:ss
Def.	Deformation measured during testing	<i>mm</i>
Force	Force measured during testing	<i>kN</i>
Pore pr.	Pore water pressure measured during testing	<i>kPa</i>
Cell pr	Cell pressure measured during testing	<i>kPa</i>

Table 3.1: Key features in text files

The format of these files have changed over the years, although most of the time ever so slightly, a Python script has been developed to read all the files and structure them into one single database using the *Pandas* package. This database is considered the starting point and is stored in a feather file format, which is basically a more efficient and faster way of storage than the more well-known Comma-Separated Values (CSV) format. A summary of all the features from the entire database is given in Appendix B. The three most interesting features, the type of triaxial test, the type of soil and the test date are given in Figure 3.1. It shows that the Consolidated Isotropic Undrained Multi-Stage (CIUMS) test is most

commonly executed on clay samples. Also note that there are not many tests available between 2005 and 2010. This is not necessarily because they were not executed, but more likely because they were stored in a binary format. This could not be read without modifying internal software, which was not feasible within the time window of this research.

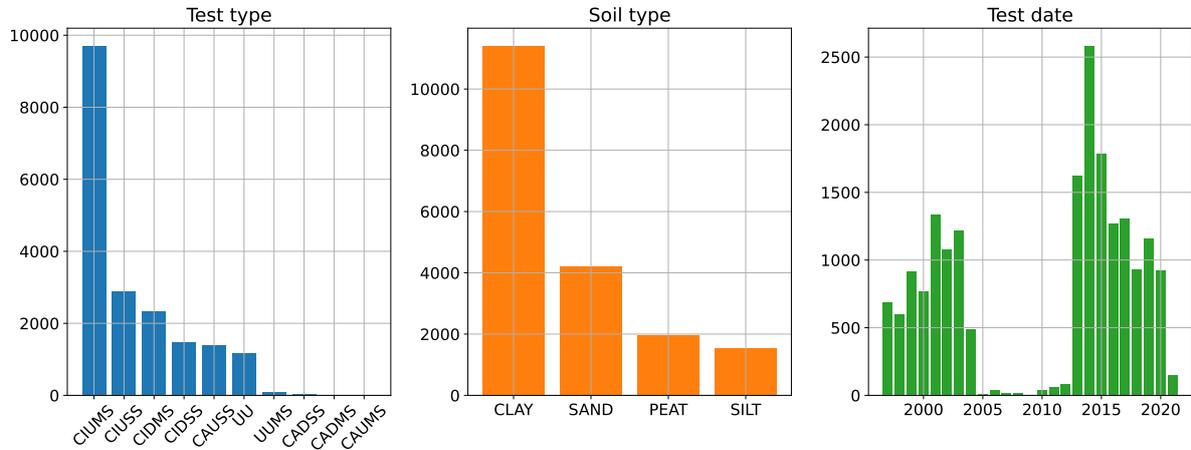


Figure 3.1: Key features from the database

3.1.2 Locations

The triaxial tests come from various places in the Netherlands, as mentioned above. Although the text files in itself did not include the coordinates of where the samples came from, it did contain the project- and borehole number. This made it possible to link files to the corresponding borehole coordinates by using the borehole archive. The reason not all coordinates could be traced back is due to the fact that not all samples were collected by Fugro, it is quite common that soil samples are delivered to the laboratory for research by clients.

The elevations were measured relative to the Amsterdam Ordnance Datum (NAP). Most of the coordinates are given in the Dutch RD (Rijks-Driehoek) coordinate system except for a few cases which are in the Belgium Lambert 72 system. The locations of the boreholes, after conversion to WGS84 coordinates, can be found in Figure 3.2. In some cases multiple samples are taken from a single borehole.



Figure 3.2: Borehole locations

3.2 Soil properties

Each text file contains a lot of information regarding the soil sample like the initial weight, dry weight, volumes etc. With this information, some initial index parameters can be determined, which already tells a lot about the sample and its properties, these results can later on be used to make correlations with other types of parameters. Table 3.2 gives an overview of the parameters and how they were obtained. Note that the particle density given in the file was almost always an estimation and was only measured in a few cases, it could not be established which cases. Generally, a value of 2.65 was assumed by a lab technician for clay and sand, and a value of 1.4 for organic material. Equation 3.1 - 3.6 further elaborate on how the calculations were performed.

Parameter	Symbol	Unit	Obtained
Gravitational constant	g	m/s^2	Assumed: 9.81
Density of water	ρ_w	g/cm^3	Assumed: 1
Degree of saturation	Sr	–	Assumed: 1
Water content	w	–	Calculated
Unit weight	γ	kN/m^3	Calculated
Dry unit weight	γ_d	kN/m^3	Calculated
Particle density	ρ_s	g/cm^3	Given
Specific gravity	G_s	–	Calculated
Initial void ratio	e_0	–	Calculated
Initial porosity	n_0	–	Calculated

Table 3.2: Initial parameters

$$\gamma = \frac{m_t g}{V} \quad (3.1)$$

$$w = \frac{m_w}{m_s} \quad (3.2)$$

$$\gamma_d = \frac{\gamma}{1 + w} \quad (3.3)$$

$$G_s = \frac{\rho_s}{\rho_w} \quad (3.4)$$

$$e_0 = \frac{w G_s}{S_r} \quad (3.5)$$

$$n_0 = \frac{e_0}{1 + e_0} \quad (3.6)$$

3.3 Soil parameters

This section describes how the raw data from the triaxial test is converted and the methods used to determine the soil parameters, additional background information can be found in Section 2.1.

3.3.1 Calibrating and correcting

Before determining the soil parameters, the data first had to be calibrated and corrected. It was often provided in an Analog/Dialog (A/D) format for which calibration factors were included, after applying them, physical units were obtained. This is the raw data and it needs to be corrected which is done according to the NEN-EN-ISO 2018*b*. Do note that some of the tests date back up till 1997, even though the NEN-EN-ISO 2018*b* did not exist at the time, it is still used.

The cross sectional area of the specimen will vary during the test, it is important to take this into account since a changing area will result in a difference in vertical stresses. Equation 3.7 corrects the specimen area by taking the difference in volume (ΔV) with respect to the initial volume (V_i) over the difference in height (ΔH) with respect to the initial height (H_i). Note that this Equation is strictly speaking not valid after the formation of shear planes.

$$A_{cor} = \frac{V_i - \Delta V}{H_i - \Delta H} \quad (3.7)$$

Corrections for the vertical stresses $(\Delta\sigma_v)_m$ imposed by the elastic membranes are determined according to Equation 3.8 and for horizontal stresses $(\Delta\sigma_h)_m$ according to Equation 3.9. In which t_m , E_m and D_m are the initial thickness of the membrane, elastic modulus and the initial internal diameter respectively. These equations assume no slippage between the membrane and specimen meaning that the membrane deforms as the specimen, this is strictly speaking not valid after the formation of shear planes.

$$(\Delta\sigma_v)_m = \frac{4t_m E_m}{D_m} \left[(\varepsilon_{ax})_m \frac{(\varepsilon_{vol})_m}{3} \right] \quad (3.8)$$

$$(\Delta\sigma_h)_m = \frac{4t_m E_m (\varepsilon_{vol})_m}{D_m} \frac{3}{3} \quad (3.9)$$

The correction for the type of filter paper $(\Delta\sigma_v)_{fp}$ which is used depends upon the strain levels. The correction for the vertical total stress during the first 2% strain is determined according to Equation 3.10 and after that with Equation 3.11. There is no correction required for the horizontal stresses.

$$(\Delta\sigma_v)_{fp} = \frac{\varepsilon_{ax} K_{fp} P_{fp}}{0.005 D_c} \quad (3.10)$$

$$(\Delta\sigma_v)_{fp} = \frac{K_{fp} P_{fp}}{0.25 D_c} \quad (3.11)$$

With all the corrections being calculated and applied, the results can be computed with Equation 3.12 till 3.16, an example of the visualised end product is presented in Appendix C.

$$q = \frac{P}{A_{corr}} - (\Delta\sigma_v)_m - (\Delta\sigma_v)_{fp} \quad (3.12)$$

$$\sigma_3 = \sigma_c - (\Delta\sigma_h)_m; \quad \sigma'_3 = \sigma_3 - u \quad (3.13)$$

$$\sigma_1 = q + \sigma_3; \quad \sigma'_1 = \sigma_1 - u \quad (3.14)$$

$$\varepsilon_{ax} = \frac{\Delta H}{H_i} \quad (3.15)$$

$$\varepsilon_{vol} = \frac{\Delta V}{V_i} \quad (3.16)$$

3.3.2 Determination: Stiffness and power for stress dependency of stiffness

The stress-strain graph can be used to determine the stiffness parameter E_{50} , which is the slope of the secant line drawn at 50% of the peak deviatoric stress (q_{50}), as is shown in Figure 3.3. Note that the exact value of q_{50} is not always available, since it is a high frequency measurement and not continuous measurement, the value closest is selected as starting point. An additional point is selected which centres q_{50} between the two, subsequently linear interpolation is used to overcome some of the deviation. Strictly speaking, E_{50} cannot be determined for the first two stages in a multi stage test, since these do not show peak strength. The first two stages tend to overestimate the stiffness which is why the E_{50} from the last stage will be used from here on.

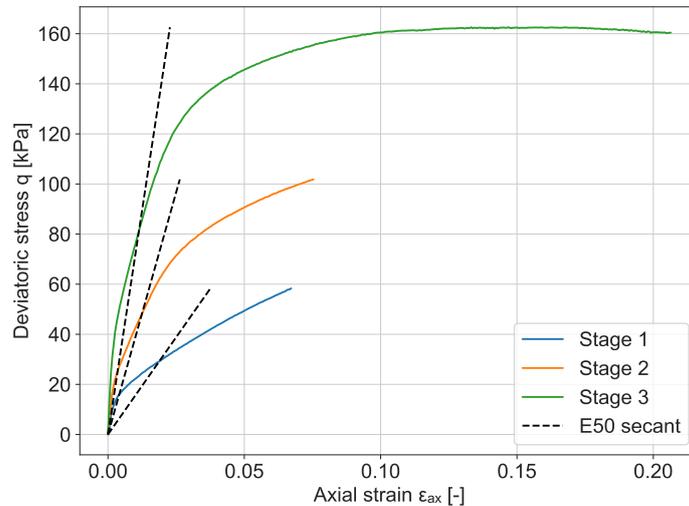


Figure 3.3: Determination of the E50

It is important to make a distinction between drained stiffness (E_{50}) and undrained stiffness (E_{50}^u). The HS small model takes effective parameters as input, meaning that all the undrained tests need to be converted. A first estimation is done with the assumption: $E_{50} = \frac{1}{2}E_{50}^u$. This can be optimised and back-calculated by fitting the simulation to the actual laboratory test.

The calculation for the power for stress-level dependency of stiffness (m) is presented in Equation 2.11. This requires two E_{50} 's at two different confining pressures, however, a triaxial test usually involves three stages. All three tests were taken into account by calculating all three possible combinations as shown in Equation 3.17 over which the average was taken with Equation 3.18.

$$m_1 = \frac{\ln(E_{50}^{(1)}/E_{50}^{(2)})}{\ln(\sigma_3^{(1)}/\sigma_3^{(2)})}, m_2 = \frac{\ln(E_{50}^{(2)}/E_{50}^{(3)})}{\ln(\sigma_3^{(2)}/\sigma_3^{(3)})}, m_3 = \frac{\ln(E_{50}^{(1)}/E_{50}^{(3)})}{\ln(\sigma_3^{(1)}/\sigma_3^{(3)})} \quad (3.17)$$

$$0.3 \leq m_{avg} = \frac{m_1 + m_2 + m_3}{3} \leq 1 \quad (3.18)$$

3.3.3 Determination: Cohesion and friction angle

The cohesion and friction angle have to be determined on a certain strain level, which is often 2%, 5% or at failure and they can be determined in either the p-q stress space or the s-t stress space. Just like determining the E_{50} , the required values are not always exactly present, hence, linear interpolation will be used once more to overcome this slight deviation. Before continuing, it is important to make some comments on the multi stage tests. Multi stage tests are performed on a single soil sample and the first two stages are loaded till near failure, hence, no actual failure is reached. This would result in more conservative results when determining the parameters at failure, since the maximum deviatoric stress might not have been reached. This can actually be seen in Figure 3.3, stage 1 and stage 2 are still increasing before the loading stopped. The last stage is loaded till failure which can also be seen when looking at the large strain level. Single stage tests on the other hand, are loaded till failure in each stage, meaning each stage requires a new sample. Although the soil samples might slightly differentiate, results are generally more representative since actual failure is reached. The more common method used to be the multi stage since this test is executed on the exact same sample and it is relatively faster, although in recent years the single stage is used more and more due to the fact that actual failure is reached. This research works with the isotropic multi stage tests since this database is significantly larger.

Two features will be calculated, the cohesion and friction angle at 2% strain and at failure according to both the s-t stress space and the p-q stress space. The index of the 2% strain level is obtained and is used to select the stress points, since neither the p-q nor the s-t stress space contain strain levels. When using the failure strain level, the maximum strain level for each stage is determined and out of those 3 levels the lowest is selected. This is done so that the parameters can be determined at the same strain levels. A fit is made through the three obtained points by using the linear regression. The interception with the y-axis and the slope of the fit can be extracted and are converted into the soil parameters as described in Section 2.1.1. All scenarios described above are visualised in an example in Figure 3.4. Note that it can clearly be seen that the failure line is on the conservative side due to the previously described comments on the multi stage tests.

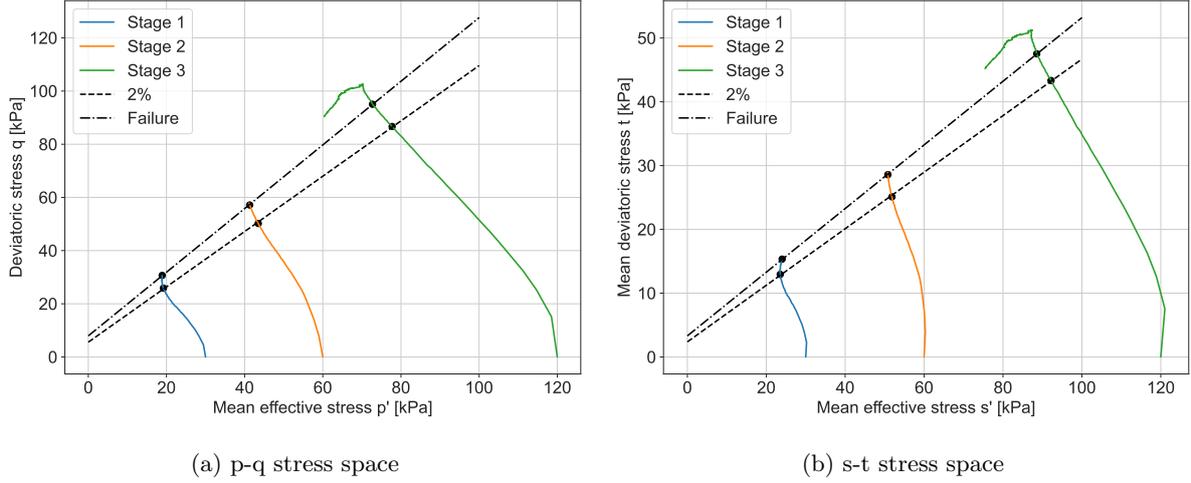


Figure 3.4: Parameter determination: cohesion and internal friction angle

The friction angle at failure is also determined when assuming zero cohesion (p-q space only), since it is preferred to work with low (to none) cohesion in constitutive models. This is done by forcing a zero-intercept with the y-axis (Figure 3.5). Since only a single additional point is required to make a fit, the maximum deviatoric stress of the final stage is chosen. Since this stage actually reaches failure it would be more representative, which would therefore result in less conservative values.

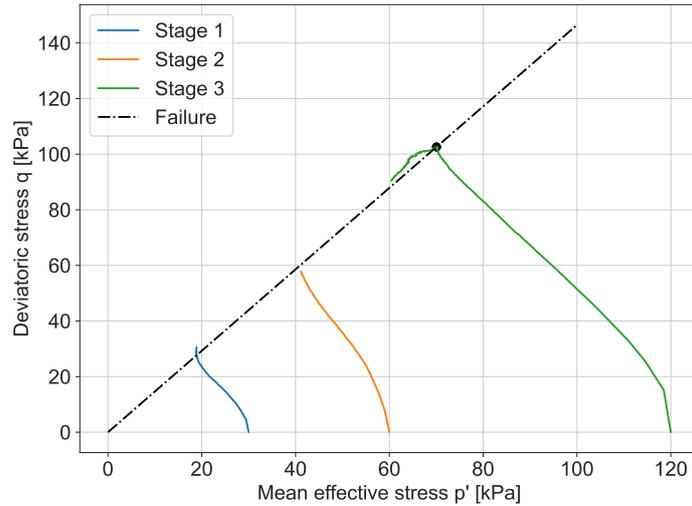


Figure 3.5: Parameter determination: internal friction angle with zero cohesion

3.4 Model parameters

This section describes how the model parameters for the HS small model were determined, estimated and optimised. The HS small model was chosen since it is a true second order model, widely used, with a variety of different applications and suitability for many soil types. The theoretical background of the HS small model and some other models for comparison is given in Section 2.2.

3.4.1 Initial

The soil parameters determined from the triaxial test were the: E_{50} , m , c' and ϕ' . Note that the E_{50} in the first 2 stages is often overestimated as described in paragraph 3.3.2, which is why the third stage is chosen as input. The next step is to further expand these into model parameters for the HS small model as an initial parameter set, which can later be used to further optimise. This is done by using

some common relationships between parameters, such as, $E_{ur}^{ref} \approx (3to5)E_{50}^{ref}$, $G_0^{ref} \approx (2.5to10)G_{ur}^{ref}$ and Jaky's equation (Jaky 1948). Also with defaults values, such as, $R_f = 0.9$ and $\gamma_{0.7} \approx (1to2) \cdot 10^{-4}$ for sands, although clay tend to have higher values. The optimisation algorithm which will be explained later on also uses parameter boundaries as input in order to prevent it from coming up with unrealistic values. The initial parameter set and its boundaries are presented in Table 3.3.

Parameter	Determined	Boundaries	Unit
* E_{50}	triaxial	$100 \leq^* E_{50} \leq 80000$	kN/m^2
E_{50}^{ref}	$E_{50}^{ref} = E_{50} \left(\frac{p^{ref}}{\sigma_c} \right)^m$	$0.25E_{oed}^{ref} \leq E_{50}^{ref} \leq 2.5E_{oed}^{ref}$	kN/m^2
E_{oed}^{ref}	$(\frac{1}{2})E_{50}^{ref}$	$0.25E_{50}^{ref} \leq E_{oed}^{ref} \leq 2.5E_{50}^{ref}$	kN/m^2
E_{ur}^{ref}	$3E_{50}^{ref}$	$3E_{50}^{ref} \leq E_{ur}^{ref} \leq 5E_{50}^{ref}$	kN/m^2
* G_{ur}^{ref}	$\frac{E_{ur}^{ref}}{2(1+\nu_{ur})}$	–	kN/m^2
G_0^{ref}	$6G_{ur}^{ref}$	$3G_{ur}^{ref} \leq G_0^{ref} \leq 9G_{ur}^{ref}$	kN/m^2
$\gamma_{0.7}$	$1.5 \cdot 10^{-4}$	$1 \cdot 10^{-4} \leq \gamma_{0.7} \leq 2 \cdot 10^{-4}$	–
m	triaxial	$0.5 \leq m \leq 1$	–
p^{ref}	100	–	kN/m^2
ν_{ur}	0.2	$0.15 \leq \nu_{ur} \leq 0.25$	–
c'	triaxial ($c=0$)	$0 \leq c' \leq 10$	kN/m^2
ϕ'	triaxial	$15 \leq \phi' \leq 50$	°
ψ	0	$-3 \leq \psi' \leq 5$	°
R_f	0.9	$0.1 \leq R_f \leq 1$	–
K_0^{nc}	$1 - \sin \phi'$	–	–

* Not a direct input parameter

Table 3.3: Initial parameter set

3.4.2 Sensitivity analysis

Sensitivity analysis is the science of quantifying the impact of variations in the values of the inputs on the output. When referring to the degree of which an input parameter effects the model the most, terms such as, 'important' and 'sensitive' are often used interchangeably Hamby 1994. The distinction made by Crick and Hill 1987 will be used, which refers to 'important' parameters as those whose uncertainty contributes substantially to the uncertainty in in the assessment of the results. While 'sensitive' parameters are defined as those which have a significant influence on the assessment of the results itself. Several types of methods of sensitivity analyses have been developed, which can fall under local or global methods.

When one is interested in performing the analysis around a point of interest, it would be called a local sensitivity analysis. Changes in the output are studied for specific values of the input. The simplest methods in this category would be the so-called one-at-a-time (OAT) methods and the procedure is rather straight forward (Borgonovo and Plischke 2016). A base case is assigned and a sensitivity case to the model inputs, which basically comes down to changing a single parameter and measure the magnitude of the rate of change compared to the base case (Equation 3.19). A nice way to visualise the results is by a Tornado diagram, it represents the measured sensitivity on a horizontal bar chart, sorted from largest to smallest and was introduced by Howard 1988.

$$\left| \frac{\partial Y}{\partial X_i} \right| \approx \left| \frac{\partial Y(x + \varepsilon_i) - Y(x)}{\varepsilon} \right| \quad (3.19)$$

The HS small model consists of a total of 13 parameters which is quite overwhelming. A local sensitivity analysis is done to obtain a better understanding of the model and what the influence of each individual parameter is. Two fictive soil samples were created, one sand sample and one clay sample

(Table 3.4). These initial simulations are the so called base cases. The simulation is executed for 10% strain in isotropic compression with a 100 simulation steps, drained for the sand sample and undrained for the clay sample.

Parameter	Sand	Clay	Unit
E_{50}^{ref}	20000	5000	kN/m^2
E_{oed}^{ref}	20000	2500	kN/m^2
E_{ur}^{ref}	80000	20000	kN/m^2
G_0^{ref}	200000	50000	kN/m^2
$\gamma_{0.7}$	$1.5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	-
m	0.5	1.0	-
p^{ref}	100	100	kN/m^2
ν_{ur}	0.2	0.2	-
c'	0	0	kN/m^2
ϕ'	35	27.5	$^\circ$
ψ	0	0	$^\circ$
R_f	0.9	0.9	-
K_0^{nc}	0.426	0.5	-

Table 3.4: Base case for sensitivity analysis

From this point onward each parameter individually is either increased or decreased by 10% except the reference stress for stiffness (p_{ref}) which is considered a constant. Changing this would also influence the meaning of all the stiffness parameters. The newly acquired stress-strain relationship is compared to the stress-strain results of the base case by determining the coefficient of determination between the two, the rate of change is then defined as $1 - r^2$ and the results are presented in Figure 3.6. Do note that the G_0^{ref} and $\gamma_{0.7}$ are mostly of influence in the low strain levels, which is why they would score lower in general with this type of analysis. Furthermore, the K_0^{nc} is a parameter that has mainly influence on the oedometer test rather than the triaxial test, it influences the (plastic) volumetric strain. A complete overview of the effect of each parameter is given in Appendix D.

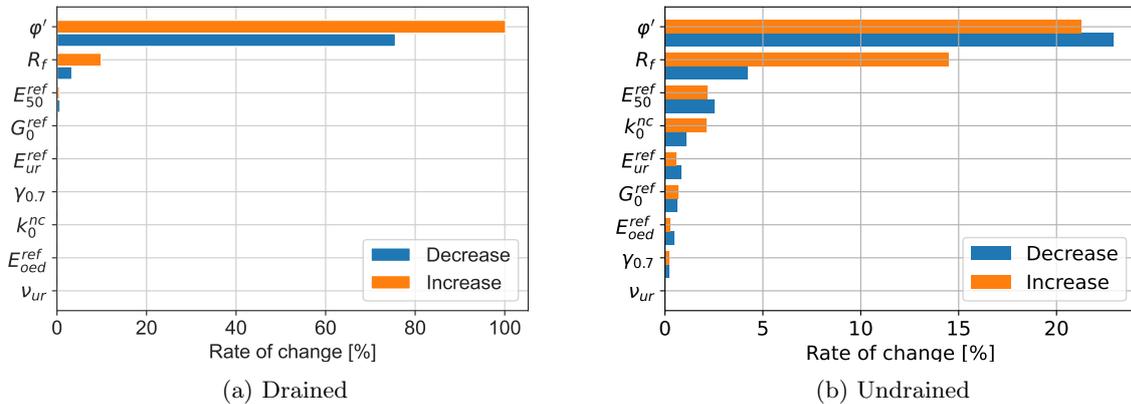


Figure 3.6: Sensitivity analysis

What immediately stands out is the difference of the amount of influence between drained and undrained tests. However, it was to be expected that undrained tests would be more sensitive due to the significant influence of the pore water pressure development. The ϕ' and E_{50}^{ref} seem to have the biggest influence which is no real surprise. What is interesting to see is that the parameter R_f is of major influence, especially in the top half of the stress-strain curve, even though it is considered a minor parameter with a default value of 0.9.

3.4.3 Optimisation

The next step is to actually optimise the model parameters. The SoilTest facility from the PLAXIS software offers the possibility to simulate laboratory tests on the basis of a single point algorithm using constitutive models and their calculation methods. This facility allows for evaluating parameters and soil behaviour by checking if it matches existing measurements and expectations. Since it is a single point, it does not account for deformations of the soil sample during the test. However, this is taken into account in the actual laboratory test, meaning, when fitting the simulation to the laboratory test the deformations are reflected onto the simulation. The remote scripting server is used to communicate with the PLAXIS SoilTest facility in order to provide input, run the simulation, and extract the output using a Python script. This makes it possible to run a large amount of simulations automatically. The idea is to fit the stress-strain path of the simulation to that of the one measured in the laboratory with an optimisation algorithm. The last stage of the multi stage triaxial test is used to make the fit since this stage reaches actual failure and the fit is made for the entire length of the stress-strain curve measured in the laboratory.

The goodness of the fit will be quantified using the coefficient of determination (r^2) and the aim is to reach a value as close to 1 as possible. The initial parameter set which is presented in Table 3.3 is used as a starting point. By varying the parameters of the HS small model an optimum is found and the general idea behind the algorithm is presented in Figure 3.7.

An example of an undrained triaxial test before and after optimisation is given in Figure 3.8. Already note one of the limitations of the HS small model in this example, it is not able to model softening behaviour (it is arguably possible with a negative dilation angle). The algorithm attempts to compensate for this by fitting a line through the middle of the top part of the stress-strain curve.

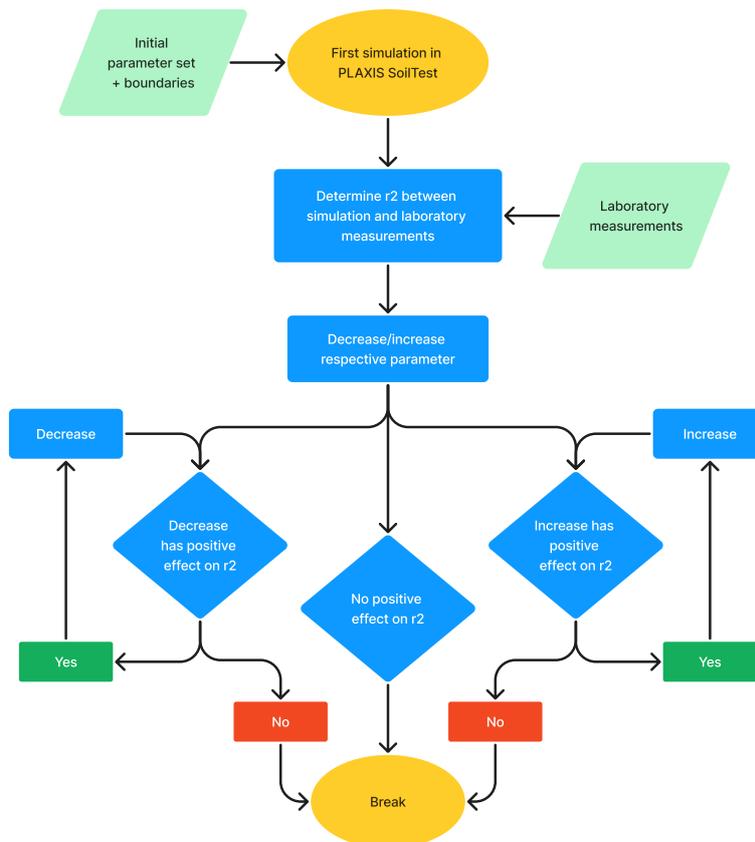


Figure 3.7: Schematic overview of the algorithm

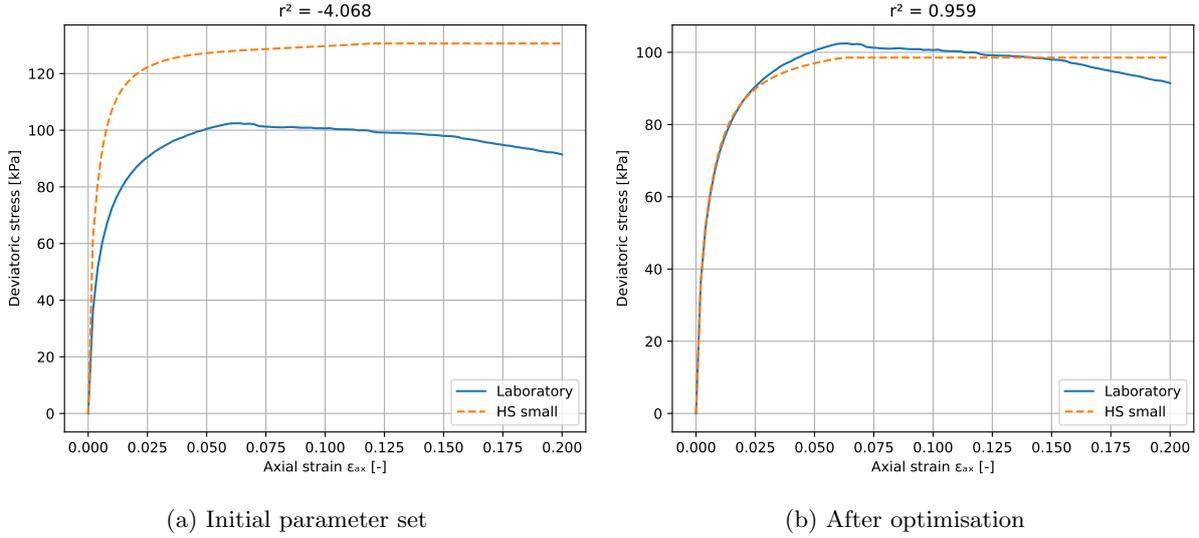


Figure 3.8: Fitting the stress-strain curve

The optimum r^2 found is a local optimum, it attempts to find the closest maximum r^2 , meaning that on a global scale there might be an even better fit. This is highly dependent on the starting point which is largely based on general assumptions and default values, Figure 3.9 illustrates this. The same optimisation algorithm is used, nearly the same r^2 is obtained and the fit looks very similar, the only difference between this optimisation and the previous one is that a ratio of $E_{oed}^{ref} = E_{50}^{ref}$ is chosen rather than the original $E_{oed}^{ref} = \frac{1}{2}E_{50}^{ref}$. At first glance it looks like the initial parameter set does not have a significant influence, but Table 3.5 proves quite the contrary. The stiffness parameters are in the order of 2 to 3 times bigger, while the friction angle is only slightly smaller.

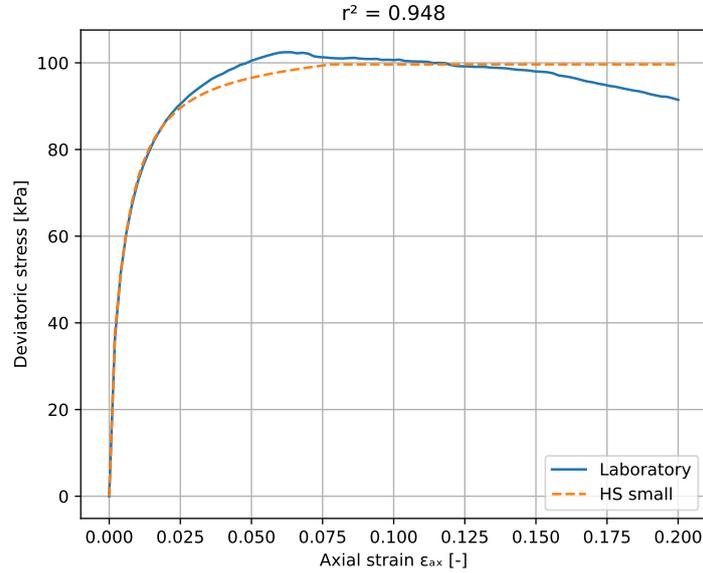


Figure 3.9: $E_{oed}^{ref} = E_{50}^{ref}$

Parameter	Original	Adjusted	Unit
E_{50}^{ref}	2751	5201	kN/m^2
E_{oed}^{ref}	1425	5601	kN/m^2
E_{ur}^{ref}	8852	16804	kN/m^2
G_0^{ref}	21380	43760	kN/m^2
$\gamma_{0.7}$	$1.7 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$	-
m	0.64	0.74	-
p^{ref}	100	100	kN/m^2
ν_{ur}	0.23	0.19	-
c'	0	0	kN/m^2
ϕ'	28.9	27.8	$^\circ$
ψ	0	0	$^\circ$
R_f	0.9	0.85	-
K_0^{nc}	0.52	0.53	-

Table 3.5: Optimisation results comparison

A possible solution to overcome this singularity (multiple solutions) would be by starting with a global optimisation algorithm, such as a particle swarm, which searches for global highs/lows. However, when looking at the minor difference in r^2 between the two optimisations and the significant difference in parameter outcomes, one might wonder if such a global search for the highest r^2 would be the solution. The r^2 is a nice way to quantify the goodness of the fit but it is not the main target, the aim is to model the soil behaviour as good as possible.

Alternatively, an additional feature measured during the triaxial test could be used to match the data. The pore pressure development in the undrained tests is of significant importance to properly describe the soil behaviour and is highly influenced by the stiffness and strength parameters. This does not apply to drained triaxial tests, but in this case the volumetric strain could be used as an additional fit to prevent a singularity from occurring. This theory is used to further develop the algorithm which attempts to properly describe the soil behaviour, starting of with the initial estimation as presented in Algorithm 1.

Algorithm 1 Base case

- 1: Obtain initial model parameters and test conditions
 - 2: Run PLAXIS SoilTest
 - 3: Obtain laboratory $q-\varepsilon_{ax}$ paths (y and x values respectively) ▷ Fit 1
 - 4: **if** Test type is drained **then**
 - 5: Obtain laboratory $\varepsilon_{vol}-\varepsilon_{ax}$ paths ▷ Fit 2
 - 6: **else if** Test type is undrained **then**
 - 7: Obtain laboratory $u_{excess}-\varepsilon_{ax}$ paths ▷ Fit 2
 - 8: **end if**
 - 9: Perform interpolation on laboratory data to obtain y-values at the same x-values as the simulation
 - 10: Determine the r_0^2 between the two y-values for both fit 1 and 2
 - 11: Assign weights to both fits
 - 12: Determine the average weighted r_0^2
-

The sensitivity analysis in Section 3.4.2 showed that ϕ and E_{50}^{ref} are the two most sensitive parameters in the HS small model, when developing an order for the optimisation algorithm it would make sense to prioritize these. However, it is also important to note the parameters that are perhaps a bit less sensitive but are solely based on general assumptions such as the E_{oed}^{ref} and R_f . Table 3.6 presents the order of the optimisation algorithm in a nutshell. Two optimisation runs are performed, the first run is different depending on the drainage conditions. It attempts fit two graphs at the same time to prevent a singularity from occurring and making it less sensitive from the chosen initial parameter set. The second run is the same for both and only the stress-strain curve is fitted. The parameters in the drained tests are overall less sensitive although the dilatancy angle turned out to play an important role due to the significant effect on the volumetric strains. The weight ratio's are defined as the weight assigned to each of the fit, such that a weighted r^2 is obtained, see Equation 3.20. These weights were determined by running a large number of simulations and checking if desired outcomes were obtained, meaning, getting the same results independent of the initial $E_{50}^{ref}/E_{oed}^{ref}$ ratio and obtaining the best r^2 .

$$r_{weighted}^2 = \frac{w_1 r_{Fit1}^2 + w_2 r_{Fit2}^2}{w_1 + w_2} \quad (3.20)$$

Type	Drained	Undrained
Fit 1	$q-\varepsilon_{ax}$	$q-\varepsilon_{ax}$
Fit 2	$\varepsilon_{vol}-\varepsilon_{ax}$	$u_{excess}-\varepsilon_{ax}$
Optimisation run 1		
Fit weight ratio	1:1	4:1
Sequence	ϕ, ψ, E_{50}^{ref}	$\phi, E_{oed}^{ref}, E_{ur}^{ref}, E_{50}^{ref}, R_f, c'$
Optimisation run 2		
Fit weight ratio		1:0
Same sequence	$\phi, E_{oed}^{ref}, E_{ur}^{ref}, G_0^{ref}, E_{50}^{ref}, R_f, m, \psi, c', \gamma_{0.7}, \nu_{ur}$	

Table 3.6: Optimisation sequence

One could argue that it is perhaps more suitable to use the p-q stress space as the second fit for the undrained tests. However, the algorithm requires at least one ascending axis in order to compare the fits and this is not the case for the p-q plot. It makes sense to use the $u_{excess}-\varepsilon_{ax}$ graph since the ε_{ax} is always ascending and also because the u_{excess} is being reflected onto the p-q stress space.

Furthermore, it is questionable if the parameter m should be optimised since it is determined directly from the triaxial test and has an influence on the stress dependency when the initial effective cell pressure is different than a 100 kPa. The parameter m is determined based on at least two $E50$'s at two different confining pressures. Section 3.3.2 elaborates that the first two stages of a Multi Stage do not actually show peak strength since no actual failure is reached, therefore $E50$ and m can strictly speaking not be determined. That is the reason it was decided to still optimise parameter m , although relatively late in the sequence.

Algorithm 2 determines if parameter should be increased or decreased by evaluating if the r^2 increases as a result of the applied changes. This function is called upon by Algorithm 3 which performs the actual optimisation.

Algorithm 2 Determine direction of the optimisation

- 1: Increase/Decrease the respective parameter
 - 2: Run simulation and determine r_-^2 and r_+^2 \triangleright Plus and Minus representing increasing and decreasing the parameter respectively
 - 3: Select maximum r_{max}^2 out of the three: r_0^2, r_-^2 and r_+^2
 - 4: **if** $r_{max}^2 == r_0^2$ **then**
 - 5: break \triangleright No further action required, parameter was already optimal
 - 6: **else if** $r_{max}^2 == r_-^2$ **then**
 - 7: Direction is negative \triangleright Parameter should be decreased to optimise
 - 8: **else if** $r_{max}^2 == r_+^2$ **then**
 - 9: Direction is positive \triangleright Parameter should be increased to optimise
 - 10: **end if**
-

Algorithm 3 Parameter optimisation

- 1: Apply Algorithm 2 to determine the direction of the optimisation
 - 2: **while** $r^2[i+1] > r^2[i]$ **do**
 - 3: parameter +/- direction
 - 4: Run simulation
 - 5: Determine weighted r^2
 - 6: **end while**
 - 7: Return maximum r^2 and associated optimised parameter
-

3.5 Parameter boundaries

With such a large data set it is to be expected that there will be some invalid and/or unrealistic results. All data outside of the boundaries given in Table 3.7 has been deleted to prevent it from having an influence on the rest of the data and its statistical analysis later on. The boundaries were inspired by the well-known table 2b from the NEN 2017 (Eurocode 7) although more conservative values were selected. Note that this is only done for the index parameters (soil properties) and the soil parameters, but not for HS small model parameters. This is because the optimisation algorithm was already given a set of boundaries which can be found in Table 3.3.

Parameter	Boundary	Unit
Unit weight	$0 \leq \gamma \leq 25$	kN/m^3
Water content	$0 \leq w \leq 10$	–
Dry unit weight	$0 \leq \gamma_d \leq 25$	kN/m^3
Specific gravity	$0.4 \leq G_s \leq 2.7$	–
Initial void ratio	$0 \leq e_0 \leq 30$	–
Initial porosity	$0 \leq n \leq 1$	–
Cohesion	$0 \leq c' \leq 30$	–
Internal friction angle	$10 \leq \phi \leq 75$	–

Table 3.7: Parameter boundaries

3.6 Single regression analysis

Regression analysis is a statistical tool to investigate the relation between an independent variable and one or more dependent variables. By fitting a regression model to a data set, one can estimate the dependent variable with independent variables where no actual data is available. These techniques are widely used in different fields, such as engineering, finance and machine learning to uncover patterns and make predictions based on those patterns. Note that these regression techniques are already considered machine learning techniques.

This study will perform the two kinds of single regression techniques: simple linear regression and non-linear regression using the exponential fit. The single (non-)linear regression will be performed between every parameter, additional background information on these techniques is provided in Section 2.3.1.

3.6.1 Outliers

Such a large database will undoubtedly contain outliers, values which deviate from the typical expected pattern in the data set. When observations are inconsistent with the majority of the set it could indicate errors and/or anomalies, these outliers can occur due to various reasons like measurement errors, rare events etc. Even though they only are a small part of the data set, it can still have quite the influence on regression techniques. Therefore, before using the above mentioned regression techniques, outliers will be removed using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al. 1996). This is an unsupervised machine learning clustering algorithm provided in the *Scikit-learn* package. The algorithm groups together data points and requires two input parameters, epsilon which is the radius of the core object and the minimal amount of points required within the radius. The algorithm is visualised in Figure 3.10. Do note that one has to be careful to let DBSCAN determine the outliers, because it also depends on the accuracy parameter eps. In this simple regression analysis a visualisation could be made to check whether or not it has the desired outcome, but this is rather difficult in the more advanced machine learning techniques, especially with multiple input parameters. That's why only in this relatively easy regression analysis the DBSCAN is used to boost the r^2 scores of the fits, the score without removing of the outliers will also be presented.

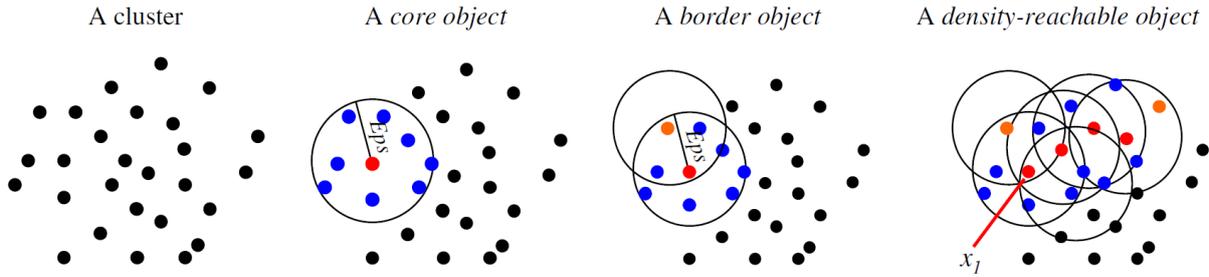


Figure 3.10: Visualisation of the DBSCAN algorithm (Tran et al. 2013)

Before this is possible, the data first needs to be normalised such that the scale of both parameters is equalised. This is done with the *MinMaxScaler* from the preprocessing module of *Scikit-learn* which returns the data set in values between 0 and 1. Figure 3.11 is an example of how the algorithm would like in action including a linear and exponential fit (although exponential is the obvious choice in this case).

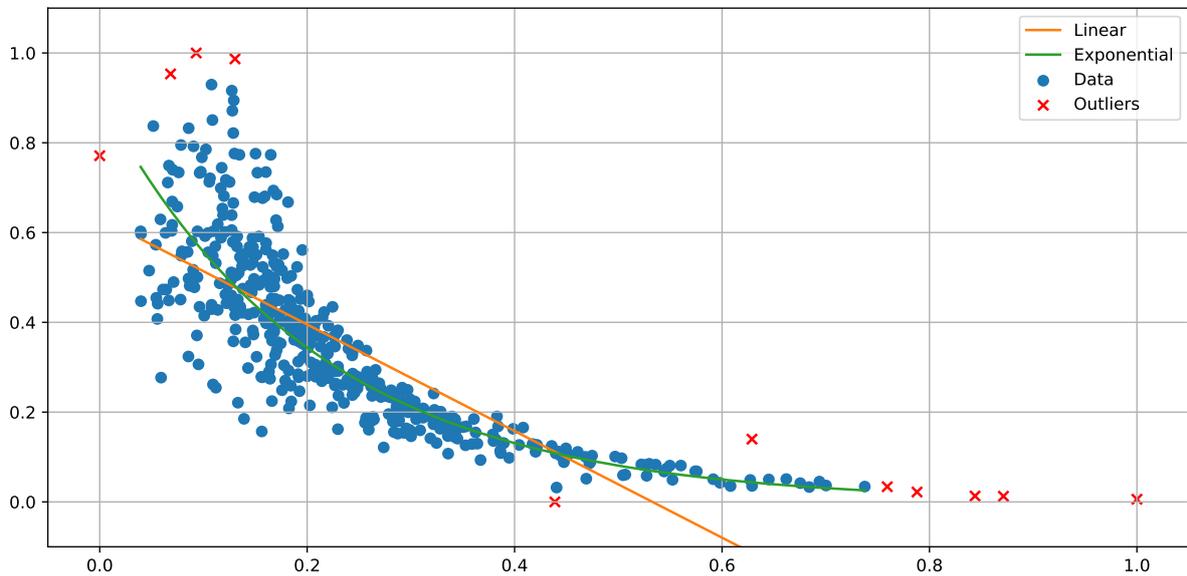


Figure 3.11: Outlier detection example with DBSCAN, ($\epsilon = 0.05$ and $samples = 5$)

3.7 Machine learning analysis

Scikit-learn has made using machine learning models very accessible for the public. It has a wide variety of different models and tools to facilitate an extensive and advanced analysis, the background on some of these more commonly used models is elaborated in Section 2.3. With that being said, it's essential to exercise caution and consider certain factors. Let's expand a bit on how a model can be properly used and evaluated. A supervised machine learning analysis in its simplest form looks something like Figure 3.12. There are independent x values and dependent labelled y values. These are presented to the machine learning model and it attempts to find patterns, similarities, differences etc. (this depends on the type of algorithm). Now that the model is trained, new x values can be presented to which the model predicts the y values.

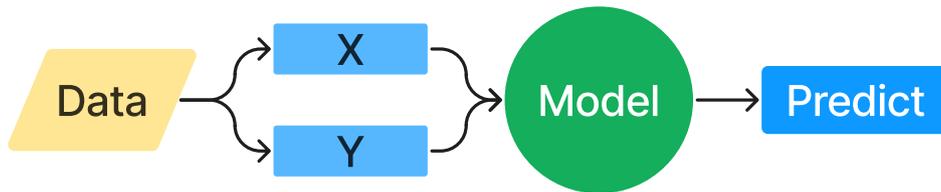


Figure 3.12: Simplest form of machine learning

3.7.1 Preprocessing the data

One of the things to take into account is that some of the independent x variables are presented on a very different scale, these scales can influence the behaviour and performance of ML models. It is therefore of importance, for certain models, to preprocess the data by scaling the x values. The combination of the ML model and prescaled data is also called the pipeline. There are quite a few methods available in the *sklearn* library, the most basic one being the *StandardScaler*. However, this one behaves quite poorly when the distribution is not more or less Gaussian (normal). This research has tested a variety of different scaling techniques like: *MinMaxScaler*, *PowerTransformer*, *Normalizer*, *QuantileTransformer* and *MaxAbsScaler*. The *QuantileTransformer* seemed to have most potential and was used for the rest of the research.

The complete input parameter set consist of the: γ , w , γ_d , e_0 and the soil type. The idea is that by providing more information on the soil sample to the ML model, that better predictions can be made which reduces the bandwidth of the outcomes. Different variations will be attempted, because sometimes an additional feature will introduce more uncertainty in the model. Meaning that simply inserting all of them does not necessarily mean a higher score. One additional preprocessing step which needs to be performed is changing the soil type from categorical data (e.g. Clay, Sand, Peat or Silt) to numerical values (e.g. [0,0,1,0]). One-hot encoding involves representing a vector wherein all elements are set to 0, except for a single element with a value of 1. This special "1" denotes the presence of a specific category for the element, expressed as a Boolean value. Without doing this, the ML model is not able to interpret the data, this is done using the *OneHotEncoder*.

3.7.2 Validating

Scaling and preprocessing the data is already a significant improvement, but how doe one evaluate such a model. It would make sense to simply present the model with an x , let it predict \bar{y} and compare it to the actual y . The problem with this is, if the presented x and y values were also used to train the model, it has a certain bias and tends to overestimate the performance of the model. One of the most commonly used methods to overcome this, is a Cross-Validation (CV) strategy called *k-fold* (Figure 3.13a). The data set is split into k number of folds. Each k fold is used as a test set for validation once, while each $k - 1$ fold is used as a training set. The r^2 is determined for each fold and the average for all folds combined is used as the evaluation of the model. The number of folds is usually chosen to be 5 or 10, although there is no formal rule. The *group k-fold* is a strategy that ensures that the same group is not represented in both the training and the testing data set (Figure 3.13b). This approach closely replicates real-world scenarios in which the model is evaluated using entirely unseen data. To put it simple, *k-Fold* can be regarded as an ideal CV strategy and is widely used in a ML context since the training and testing data share the same distribution. This means in essence that the simulated scenario tends to be milder and therefore lead to relatively good results. In contrast, the *group k-fold* simulates a more complex scenario in which the testing distribution can be very different from the training distribution. This brings a harder challenge to the ML algorithm, which leads to relatively poor results if the groups are of significant influence. When implementing both, one conservative and one radical evaluation is done.

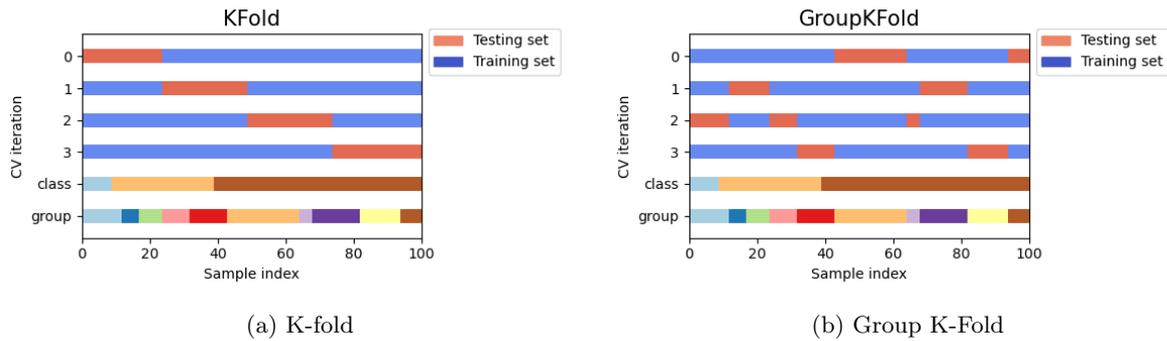


Figure 3.13: Cross validation Pedregosa et al. 2011

3.7.3 Hyperparameter tuning

It is possible to further optimise the model by tuning the hyperparameters. This is an essential aspect of machine learning and a crucial part of model development. Hyperparameters are external settings which cannot be directly learned from the data. A suboptimal hyperparameter set leads to over- or underfitting, resulting in a poor model performance. The tuning process involves the systematically exploring of the hyperparameter space to find the combinations which results in the highest performance. A grid search is a powerful tool to explore this hyperparameter space (Figure 3.14a). The *GridSearchCV* function is used to explore the hyperparameter space and obtain the best combination. For each combination it implements a (*group*) *k*-fold cross validation, note that the *GridSearchCV* is computationally an expensive method. An alternative solution, *RandomizedSearchCV*, was explored (Figure 3.14b). In contrast to the grid search, not all unique parameters combinations are simulated, rather a fixed number of settings is sampled from the distributions. It has been demonstrated that it can discover models of comparable or superior quality using only a fraction of the computation time, Bergstra and Bengio 2012 argued that random grid search CV especially performs better when using artificial neural networks. It is however decided to still choose the regular grid search since it is the more popular option, and time and computing power were not an issue.

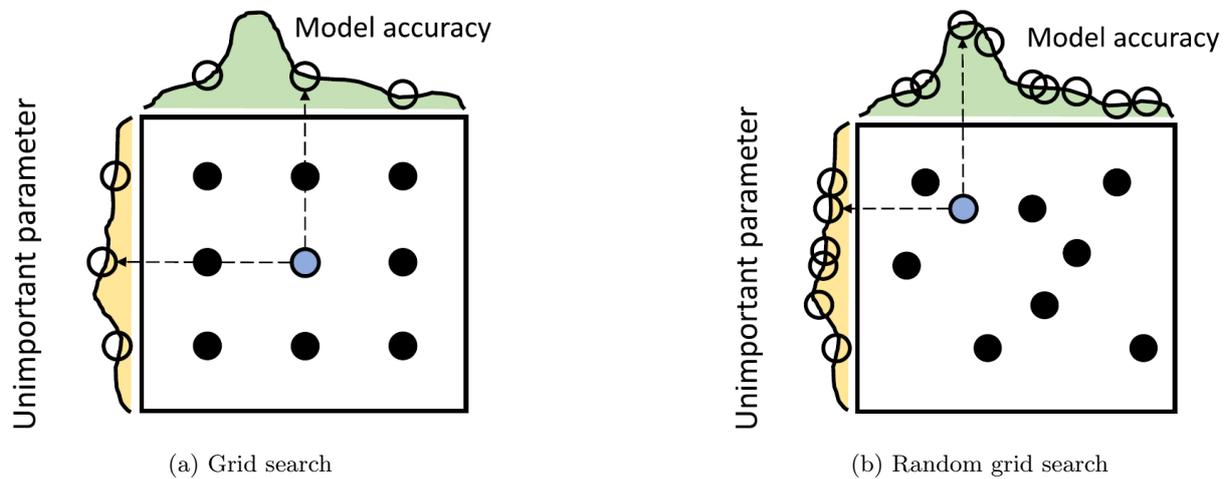


Figure 3.14: Hyperparameter tuning Bergstra and Bengio 2012

Combining all of the above results into a fully formed ML model which is optimised and can evaluate its performance. When looking at Figure 3.15 note the features which are involved when comparing it to the basic model in Figure 3.12.

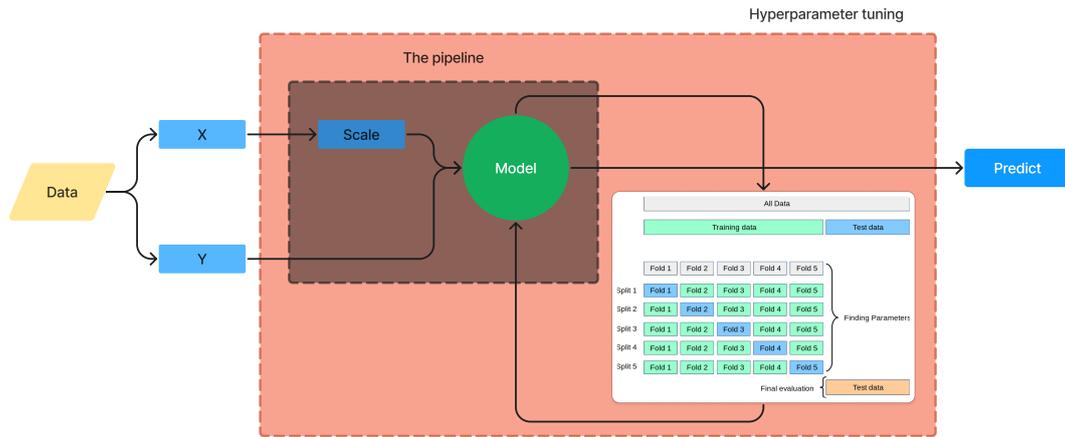


Figure 3.15: A more advanced approach on machine learning

3.7.4 Input/output of data

So now that there is a general understanding of what such a machine learning model looks like, let's proceed on what data will be used as input, and what data is used as output. At this stage a database consisting of soil properties, soil parameters, and optimised constitutive model parameters for the HS small model is established. The idea is to use these relatively easy to obtain soil properties to try and predict the soil parameters and optimised model parameters. Additionally, it is expected that by including more soil properties as input, that the performance of the models increases and the bandwidth of the predictions will become narrower. More parameters does not necessarily increase the performance, since additional inputs might introduce additional uncertainty. The idea will be tested by formulating multiple input parameter sets with an increasing number of soil properties, this is presented in Table 3.8, the soil type is the description which was given in the laboratory. The output parameters will first be predicted independently, and afterwards the complete array of either soil parameters or model parameters will be predicted. It makes sense to also attempt to predict the entire array since the model parameters affect each other.

Type	Combination	Parameters
Input	1	Soil type
	2	Soil type, γ
	3	Soil type, γ , w
	4	Soil type, γ , w , γ_d
	5	Soil type, γ , w , γ_d , e_0
Output	1	Every parameter individually
	2	Array of soil parameters
	3	Array of optimised model parameters

Table 3.8: Parameter sets

3.7.5 Model selection

The wide selection of machine learning models in the *Scikit-learn* library is quite staggering. These models have different methods and applicability's and can therefore perform very different from one another depending on the type of data. Duffy 2019 for example made use of methods such as (eXtreme) Gradient Boosting and Artificial Neural Networks, in the end Gradient Boosting seemed to perform the best. Yu 2022 used these techniques as well and expanded by using Support Vector Machines and the Gaussian Process Regressor. The eXtreme Gradient Boosting came out on top and the second best performing model was the Gaussian Process Regressor, this ML model quantifies uncertainty as well.

Besides existing literature *Scikit-learn* also provide guidance in selecting the right model by means of a map, which suggested (Kernel) Ridge Regression. The multiple linear regression was selected as a baseline as it is considered machine learning in it's simplest form. All of the above named techniques are summed up in the following list:

- Multiple Linear Regression (MLR)
- Tree-based
 - Random Forest Regressor (RFR)
 - Gradient Boosting Regressor (GBR)
 - eXtreme Gradient Boosting (XGB)
- Artificial Neural Network (ANN)
- Gaussian Process Regressor (GPR)
- Support Vector Regressor (SVR)
- Regular Ridge Regressor (RRR)
- Kernel Ridge Regressor (KRR)

This is quite an extensive list of different models, and performing an full analysis on all of them would be abundant. A quick scan will be performed to see which top three models show the most potential, these will be used for further research purposes. The quick scan is done by using combination 5 from the input parameter sets in Table 3.8 and predicting individual output parameters. The snippet down below provides a general overview of how the code is structured, in this example the ANN was used. To maintain transparency code snippets of the pipe and hyperparameter grids have been added in Appendix E. Note that in case of tree-based methods no feature scaling is required. The split feature is not influenced by other features which is why they are invariant to the scale of the feature.

```

1 # Random state parameter which allows for reproducible results
2 rs = 1
3
4 # Creating the pipeline
5 pipe = Pipeline([
6     ("scale", QuantileTransformer(random_state=rs)),
7     ("model", MLPRegressor(random_state=rs, max_iter=2000, activation='relu'))
8 ])
9
10 # Setting up a hyperparameter grid for the grid search
11 paramgrid = {'model__hidden_layer_sizes': [(100,), (15,50,15), (16, 16)],
12             'model__alpha': [0.0001, 0.05, 0.01],
13             'model__solver': ['lbfgs', 'adam']}
14
15 # Specifying the type of Cross Valiation, either k-fold or group k-fold
16 if crossval == 'k-fold':
17     cv = KFold(shuffle=True, n_splits=5, random_state=rs)
18 elif crossval == 'group_k-fold':
19     cv = list(GroupKFold(n_splits=5).split(x,y,groups))
20
21 # Setting up the model
22 model = GridSearchCV(estimator=pipe, param_grid=paramgrid, cv=cv, scoring='r2')
23
24 # Fitting the model
25 model.fit(x, y)
26
27 # Extracting results
28 resultsCV = pd.DataFrame(model.cv_results_)

```

Listing 3.1: Artificial Neural Network code snippet

3.8 Exploratory data analysis

A this stage, a large database of a variety of different properties and parameters is obtained. This section will elaborate on different types of methods to describe and visualise the results in order to extract the most amount of information from the database.

3.8.1 Exploratory data analysis: Statistics

The classical method to describe important features of a data set is to give several numerical features, such as the sample mean (μ) which is easily calculated with Equation 3.21. The standard deviation (σ_{sd}) is a measure of how dispersed the data is in relation to the mean and is calculated with Equation 3.22, a high standard deviation indicates that the data is more spread out. These two can be used to determine the probability density function of a normal distribution (Equation 3.23), which is a mathematical function that defines the likelihood of different outcomes or values of a variable. Figure 3.16 illustrates the effects of μ and σ_{sd} on the normal distribution. There are several types of distributions but the normal distribution is most commonly used and plays a central role in probability theory and statistics (Dekking et al. 2005).

$$\mu = \frac{x_1 + x_2 + \dots + x_n}{n} \quad (3.21)$$

$$\sigma_{sd} = \frac{\sum |x - \mu|}{n} \quad (3.22)$$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{for } -\infty < x < \infty \quad (3.23)$$

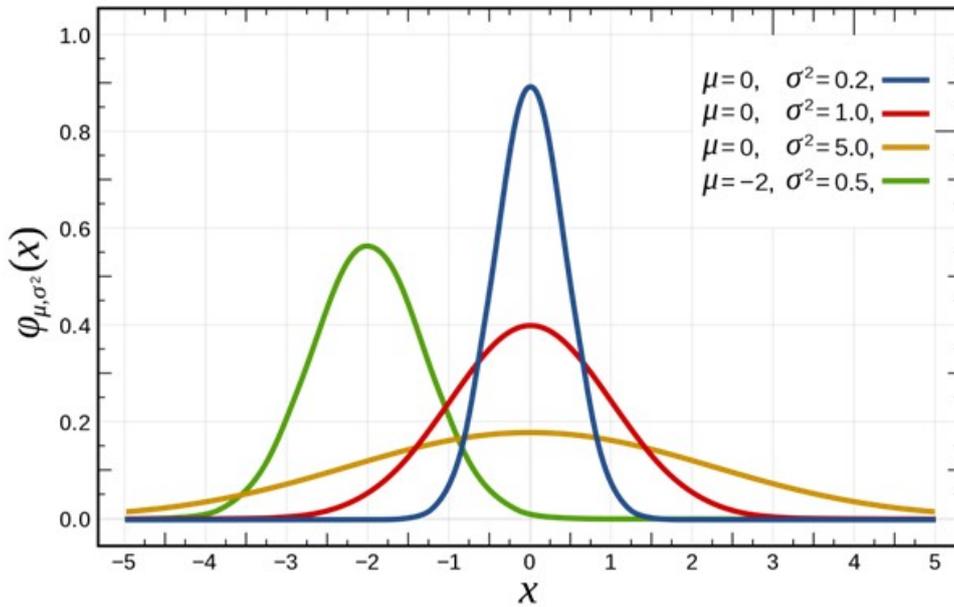


Figure 3.16: Normal distribution and the influence of μ and σ_{sd}

The normal distribution does not necessarily always fits the data well, alternatively one might use the Kernel Density Estimate (KDE), a technique that creates a smooth curve from a set of data. This might be useful when trying to visualise the shape of the data in a continuous form rather than a discrete histogram. The kernel estimator is given by Parzen 1962 and is presented in Equation 3.24, where K is the kernel. An important parameter of the KDE is the bandwidth (h) which describes how smooth the line is, the higher the number, the smoother the curve. The bandwidth selection in this research is done by Scott's rule Scott 2015.

$$f(x) = \frac{1}{nh} \sum_{i=1}^n K \frac{x - x_i}{h} \quad (3.24)$$

Continuing on the idea of how to describe a data set, the mean is the natural analogy for a data set of what the expectation is for a probability distribution. However, it is very prone to outliers, outliers are observations in the data that deviate a lot from the bulk data. Another way to identify the centre of the data set is by means of the sample median also known as the 50th percentile. A percentile (p^{th}) divides the data set in two at a certain percentage and specifies a portion p^{th} which is less than that number and a portion which is larger than this number ($1 - p^{th}$). The median is defined as the middle

element in a data set when it is put in ascending order, in case the data set is even, the average of the middle two elements is taken. The median is more robust in the sense that it is less affected by outliers. Tukey et al. 1977 suggested to provide a five-number summary instead of only identifying the median, which helps to give a more complete overview of the data set. This five-number summary can nicely be visualised by a so-called box-and-whisker plot, which makes it possible to get a general understanding of the data set in an instant (Figure 3.17). The distance between the upper quartile and the lower quartile is also known as the interquartile range (IQR) and specifies the range of the middle half of the data set.

- Minimum
- Lower quartile: 25th percentile
- Median: 50th percentile
- Upper quartile: 75th percentile
- Maximum

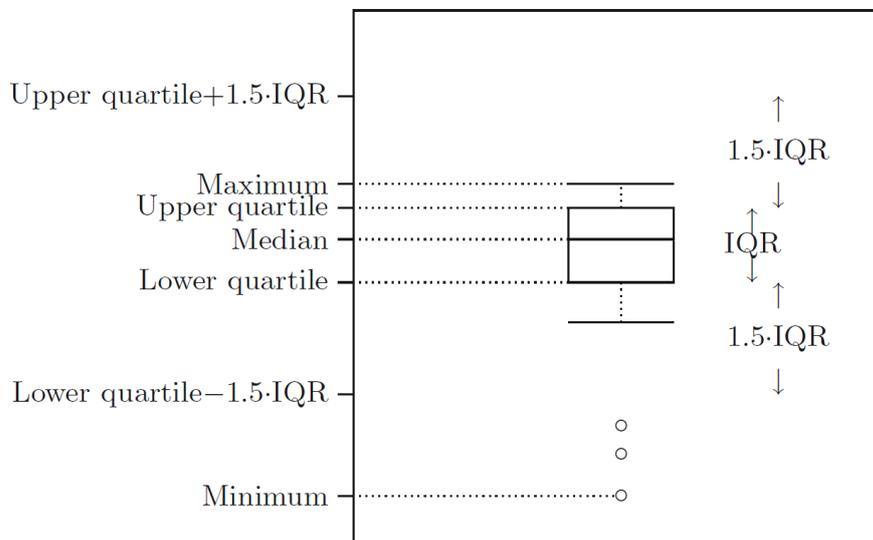


Figure 3.17: Visualisation of the five-number summary, box-and-whisker plot (Dekking et al. 2005)

The probability distribution can be visualised with the KDE as a continuous function and the five-number summary can be visualised with the box-and-whisker plot. These two provide a lot of information regarding the database and can be combined in a so-called violin plot. A summary of the features in such a plot is presented in Figure 3.18.

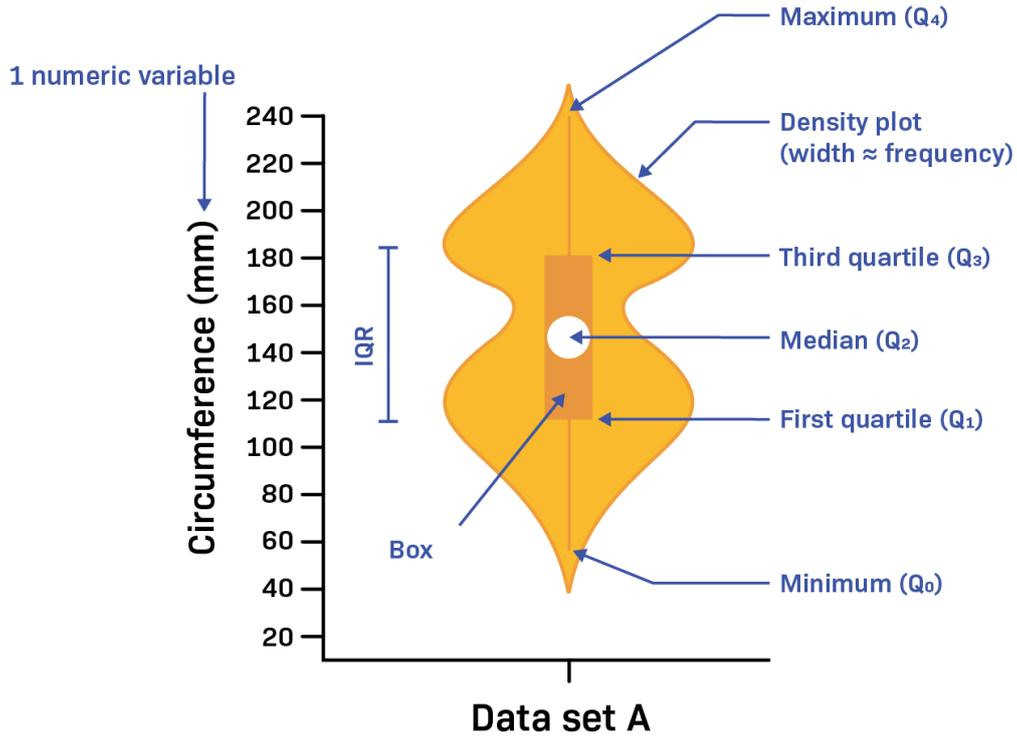


Figure 3.18: Violin plot explanation (labxchange 2021)

3.8.2 Exploratory data analysis: Correlation

It is common in statistical analysis to explore and summarize the strength of association between two continuously measured features, for which there are a numerous of different methods available. The Pearson product-moment correlation coefficient (r_p) is arguably one of the most frequently used, it is a dimensionless index which is invariant to linear transformations of either variable (Pearson 1896). Equation 3.25 demonstrates how to calculate the coefficient and the range of outcomes lies between -1 and + 1. A value of -1 meaning a perfect negative correlation and a value of +1 a perfect positive correlation.

Alternatively, the Spearman rank order correlation coefficient (r_s) could be used (Spearman 1987). The calculation is equivalent to that of Pearson, but is performed after both x and y have been rank transformed to values between 1 and N . When calculating r_s fractional ranking is used, which means that the rank is assigned in case of ties (De Winter et al. 2016). For example, the two smallest values of x are the same, then they will both be ranked 1.5.

$$r_p = \frac{\sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(\sum_{i=1}^N (x_i - \bar{x})^2 \sum_{i=1}^N (y_i - \bar{y})^2)}} \quad (3.25)$$

Both these coefficients describe the correlation between two parameters, but measure different types of association. Pearson's coefficient only measures linear association whereas the Spearman coefficient measures a broader class of association. A high absolute value of this coefficient indicates that there is a monotonic, but not necessarily linear relationship between two variables (Puth et al. 2015). This research uses the Spearman's coefficient since soil behaviour is highly non-linear, making Pearson's coefficient less suitable. The visualisation of correlating parameters against one another can be done by means of a heat map.

3.9 Summary

This chapter, the methodology, provides a structured overview of how the research was conducted and which steps and procedures were taken in order to obtain the results. Section 3.1 started of with an overview of the database, this was required to see how this data can be best put to use. Additionally, the project number and borehole number were traced back to make a map of where the samples came from.

The soil properties are calculated in Section 3.2, the idea is to link these properties to other types of parameters using different types of machine learning tools. Classic correlations compare one independent parameter to a dependent parameter, this often comes with a lot of scatter and spread. It is expected that the spread of predictions can be reduced by including multiple of these independent soil properties.

Section 3.3 explains how the classic parameters c' , ϕ' , E_{50} (and m) were determined from the triaxial test. Before deriving the parameters, the data was first calibrated and corrected according to the NEN-EN-ISO 2018*a* and NEN-EN-ISO 2018*b*.

The model parameters for the HS small model are determined in Section 3.4, starting of with the initial estimation of the parameters. A sensitivity analysis was performed to see how the different parameters affect the soil behaviour in a triaxial test. This was used as input to create an optimisation algorithm which matches the triaxial simulation to the laboratory measurements. Section 3.5 provides some boundaries for which tests are considered invalid.

Section 3.6 performs the single regression analysis: linear and exponential. While Section 3.7 provides the analysis of the more advanced machine learning tools among which: multiple linear regression, tree based methods and artificial neural networks.

At this stage all the data has been processed, parameters are determined and optimised, and a machine learning analysis is conducted. The trick is now to properly visualise and present the obtained results, Section 3.8: exploratory data analysis, provides the methods which were used to do so.

Chapter 4

Results and discussion

This chapter gives an overview of the obtained statistics, correlations, regressions and machine learning results while elaborating on them. The remaining database, after scraping invalid results and test types like the UU or anisotropic tests, consisted out of 3073 CIUMS and CIDMS which can be subdivided in soil type categories:

- Clay: 1707
- Sand: 718
- Peat: 374
- Silt: 274

4.1 Statistics

This section elaborates and discusses the statistics found in the database, background information regarding the presentation format of the results can be found in Section 3.8.1.

4.1.1 Soil properties

The summary of the soil properties is given in Figure 4.1 and additional information regarding the calculations of these properties can be found in Section 3.2. The *violinplot* function from the Python package *Seaborn* was used to make the visualisation. This was chosen since it can present both the distribution of the data using the KDE and present the five-number summary using the box-and-whisker plot. The KDE normalised the data such that the total area under the continuous function fitted through the bins is equal to 1 which describes the likelihood of different outcomes, somewhat similar to a normal distribution. It is important to note that the soil label given does not mean that it only consists out of this specific soil. Descriptions in the text file are often quite long (and subjective), E.g. "CLAY= slightly silty= slightly organic grey". In this example, the sample is labelled as clay since this is the main component, this also explains the relatively wide spread in the graphs. Remarkably, the unit weight of some peat samples is lower than that of water, which seems unlikely. A possible explanation is the presence of organic matter with a specific gravity lower than water, or gasses that cause an overall unit weight lower than water. The corresponding numerical values of the five-number summary can be found in Appendix F. The graphs in general show results which were to be expected, like fairly low (dry) unit weights for peat and relatively high for sands which corresponds with existing literature. Similarly for the initial void ratio and porosity, especially note the amount of spread on the peat samples, which highly depends on the amount of organic content. The void ratio graph is rather hard to read due to this large spread, which is why a re-scaled version with peat in a separate graph can be found in Appendix G.

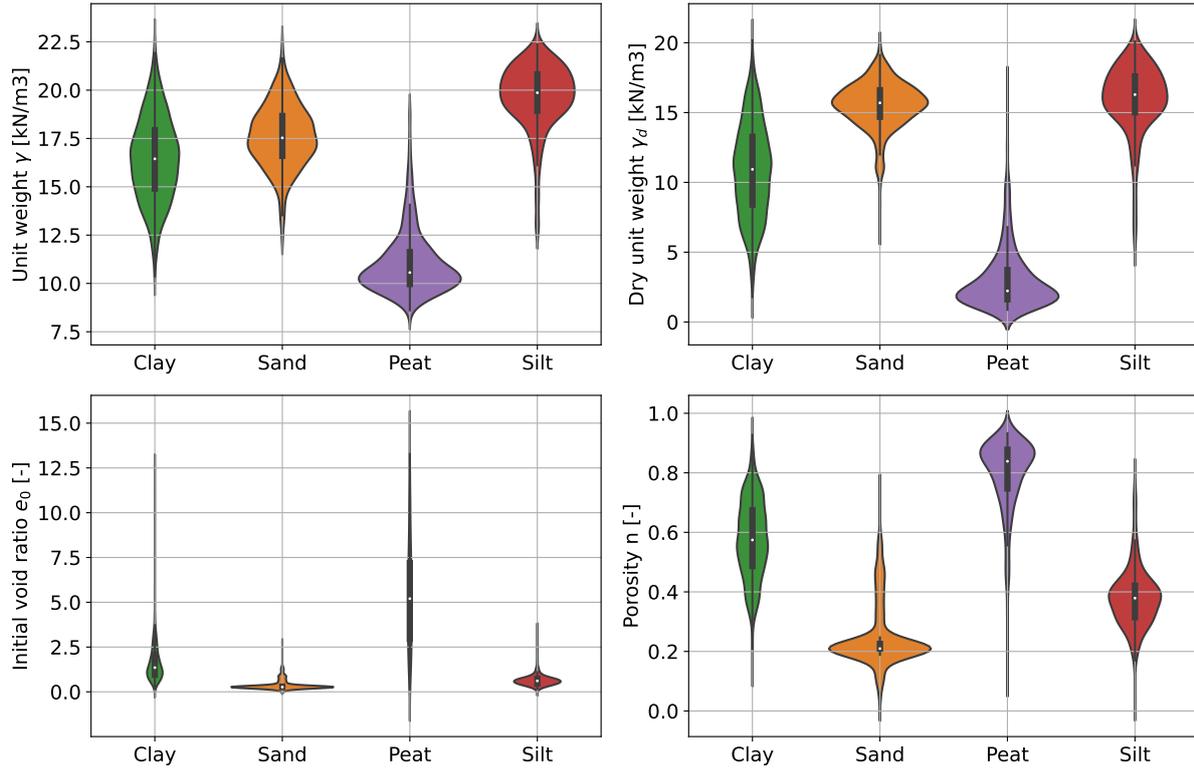


Figure 4.1: Soil properties violin plot

4.1.2 Soil parameters

Similar to the previous Section, violin plots have been made for the soil parameters which are presented in Figure 4.2. The parameters were determined at 2% strain with the p-q stress space which is common in engineering practices, additional information regarding the determination of the parameters can be found in Section 3.3.3. As expected, soils of which the main component is sand have the highest friction angle and peat the lowest, furthermore, they are in the same order of magnitude as table 2b from the NEN 2017. As discussed earlier, the wide spread can be explained due to the classification of the samples. It also explains why even though sand has the lowest mean cohesion, it isn't zero, since it can contain other soil types like clay as well. The KDE gives the impression that the cohesion can be lower than zero which is not the case and this can be seen when taking a close look at the box-and-whisker plot part of the graph. The entire numerical five-number summary of the box-and-whisker plot be found in Appendix F.

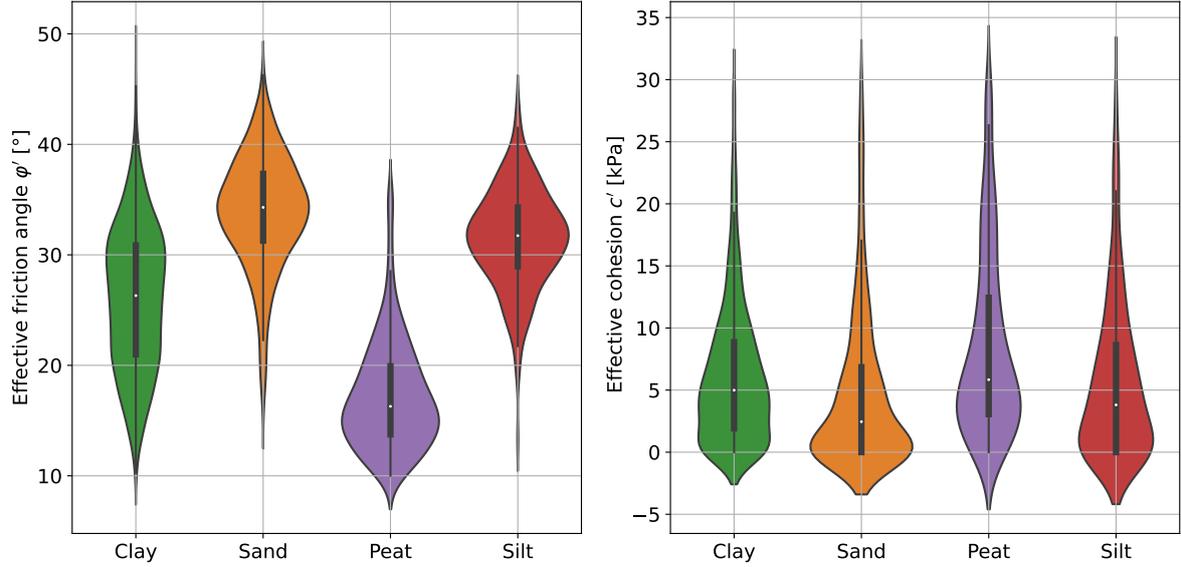


Figure 4.2: Violin plot: c' and ϕ'

The large spread in these graphs is partly due to the type of classification, the large number of samples, and the complex heterogeneous material itself. When including an additional objective feature, like the initial void ratio or unit weight, the spread in the graph is expected to be reduced. Clay has a particular large spread which is why it was selected as an example. The data is split up in three bins based on the initial void ratio, the first bin consists of all the data in the lower percentile (25th), the second bin of everything above the upper percentile (75th) and the third bin is everything in between (IQR). The same has been done for the unit weight and the results are presented in Figure 4.3, the black line is the KDE containing all the clay samples. It's interesting to see that besides the decreasing spread, that indeed samples with a lower void ratio show an overall higher friction angle and vice versa. While this is the exact opposite when looking at the unit weight, meaning, that the friction angle is positively correlated with the unit weight and negatively correlated with the initial void ratio. This is a simple example of how an extra feature can reduce the spread and increase the accuracy of predicting the friction angle, and this is also the reason why later on advanced machine learning techniques will be explored. What if instead of just one additional feature, multiple features can be provided as input, the idea is to even further reduce the spread and improving predictive capability's.

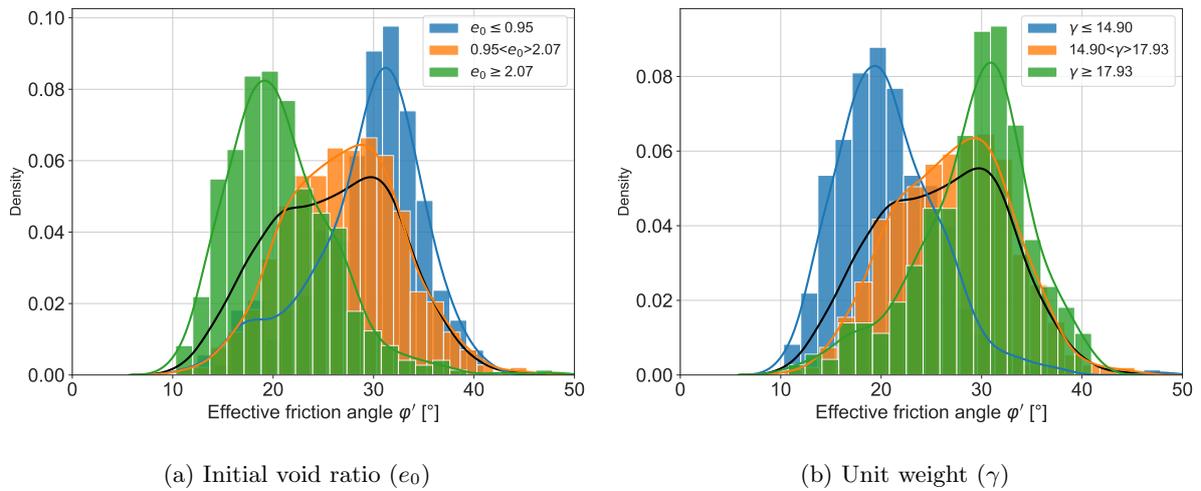


Figure 4.3: Including additional features to subdivide ϕ'

Figure 4.4 presents the violin plot of the E_{50}^{ref} at a reference stiffness of 100 kPa. This parameter is officially considered a model parameter, but comparing the E_{50} determined from the triaxial test would be

unsuitable since stiffness is stress dependent. Both the drained and undrained tests are presented in the plot, although the undrained stiffness was first converted to a drained stiffness by a general assumption as explained in Section 3.3.2. As expected, peat has the lowest stiffness, followed by clay, silt and sand, although the wide spread of the sand samples is quite remarkable. This is possibly due to the significant influence of the amount of clay in the samples. A highly clayey sand sample is expected to have a lower stiffness than a coarse grained, densely packed sand. The entire numerical five-number summary can be found in Appendix F.

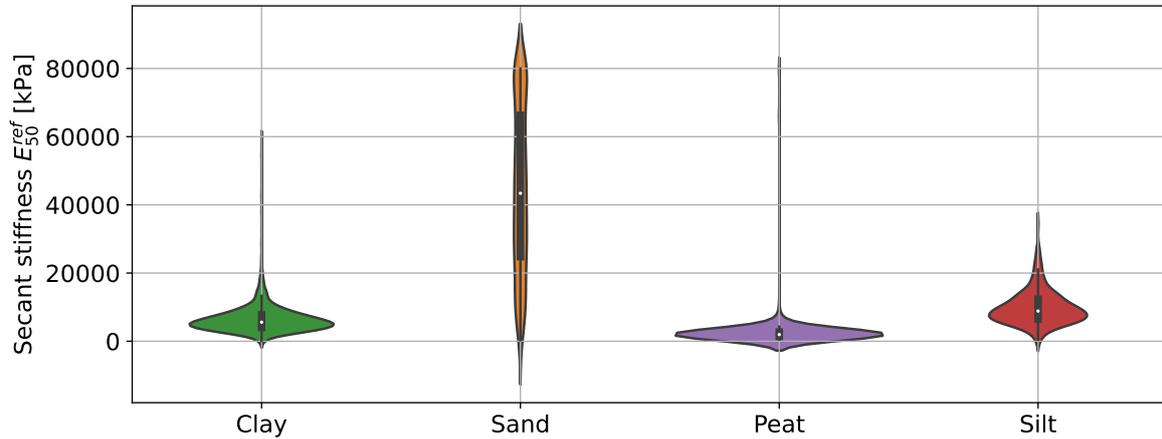


Figure 4.4: Violin plot: E_{50}^{ref}

4.1.3 Model parameters

The model parameters were first estimated using results from the triaxial test, general assumptions and correlations. Subsequently, these model parameters were optimised using the algorithm which is elaborated in Section 3.4.3. The goodness of the match is measured with the r^2 and it attempts to find a better match with every iteration by adjusting the model parameters. An example of this iterative process is presented in the convergence curve in Figure 4.5. The algorithm explores the direction it needs to take in order to increase r^2 which is why it also decreases sometimes, ideally it eventually converges to 1. This process is actually similar to the Kalman filter (Welch 2020), it describe the process and estimations of the past in a way that minimizes the mean of the squared error (although the coefficient of determination is used in this case).

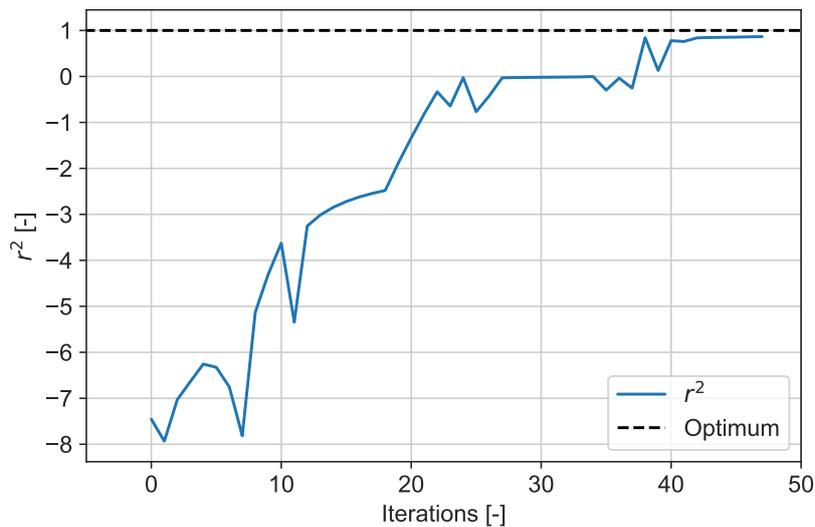


Figure 4.5: Convergence curve of r^2

This is done for every triaxial test and the results are presented in Figure 4.6, the left graph shows the initial guess and the right graph shows the optimised r^2 . The box-and-whisker plot was chosen here rather than the violin plot, since the KDE gives the impression that values higher than 1 can be achieved, which is not possible for the r^2 . It can be seen that overall every initial guess was quite poor at mimicking the laboratory results and that after optimisation it performs quite a lot better. Especially interesting how well it seems to model the softer soils like clay and peat, while having a relatively bad score for silt. Sand is somewhere in between in terms of how well the match is, this can be explained due to some of the limitations of the HS small model. Sand sometimes shows a high peak strength and low residual strength, it is exactly that softening behaviour which cannot be captured by the HS small model (it is arguably possible with a negative dilation angle). The five-number summary in Appendix F present the accompanying numerical values.

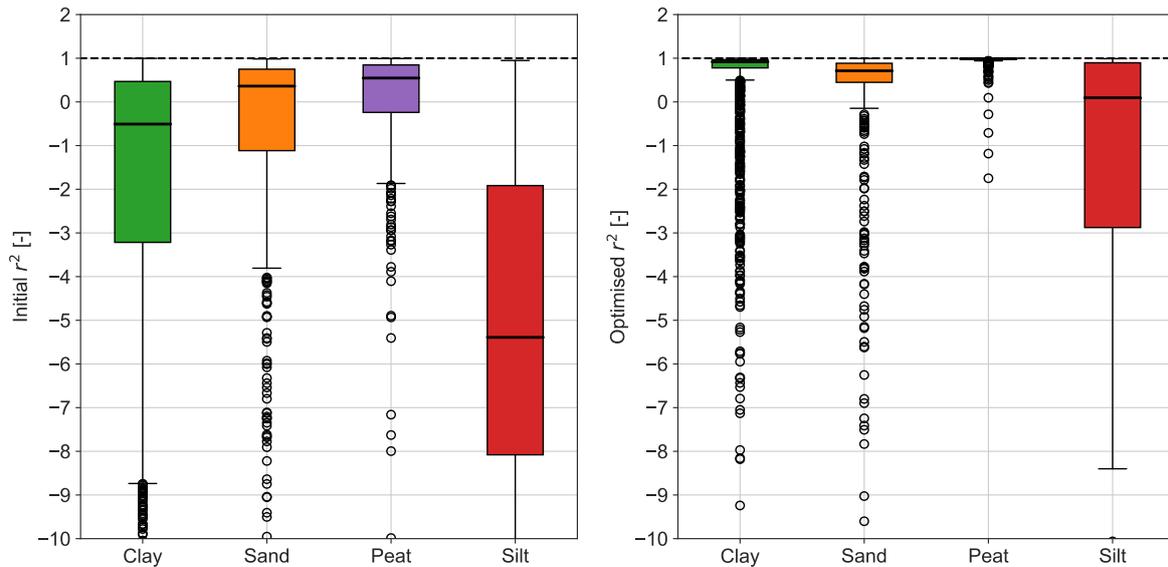


Figure 4.6: Initial r^2 vs optimised r^2

The violin plot of the optimised model parameters is presented in Figure 4.7 and the entire numerical five-number summary can be found in Appendix F. To prevent confusion, c' and ϕ' are the soil parameters determined at 2% strain with the p-q stress space and $*c'$ and $*\phi'$ are the optimised model parameters for the HS small model. A couple of interesting things can be noted when looking at the graphs below. One being that all the stiffness related parameters are somewhat as expected with sand having the highest values and peat the lowest values. The stiffness parameter graphs are rather hard to read due to the large spread of the sand samples, which is why a re-scaled version with sand in a separate graph can be found in Appendix G.

The internal friction angle on the other hand seems quite contradictory with sand being on the lower end of the spectrum. This is probably because it is influenced by the long residual strength branch in the stress strain curve, and the fact that the HS small model cannot account for softening behaviour. The optimisation algorithm attempts to find a compromise between the peak strength and the residual strength (critical state) which is illustrated in Figure 4.8 with the dashed orange line. This is where the question arises of what one is trying model, this research made the match for the entire stress-strain curve measured in the laboratory, however, perhaps only the first 3% strain is of interest. Alternatively, a distinction can be made between Serviceability Limit State (SLS) or Ultimate Limit State (ULS) parameters. For which the SLS should be matched more on the peak strength and the ULS on the residual strength, by e.g. applying weights to different parts of the curve. Another remarkable feature is the high friction angles of peat, literature such as the NEN 2017 often suggests friction angles of $\approx 15^\circ$ while undrained triaxial tests on peat samples result in significantly higher values. Do keep in mind that this is not the traditional ϕ' but one that is optimised for the HS small model using the triaxial laboratory results.

The dilatancy angle for sand on the other hand is more realistic, when recalling the compromise between peak and residual strength, but it does not correspond to the commonly used rule of $\psi = \phi' - 30$ (Bolton 1987). Dilatancy occurs due to volumetric straining which theoretically cannot occur in undrained

tests. There are a handful of clay, peat and silt samples which show some dilatancy, this probably manifested due to phenomena like anisotropy and pore water pressure development.

There are also a couple of parameters which show no real clear pattern when you look at the spread of the results, such as the power for stress dependency m or the Poisson ratio for unloading/reloading ν_{ur} .

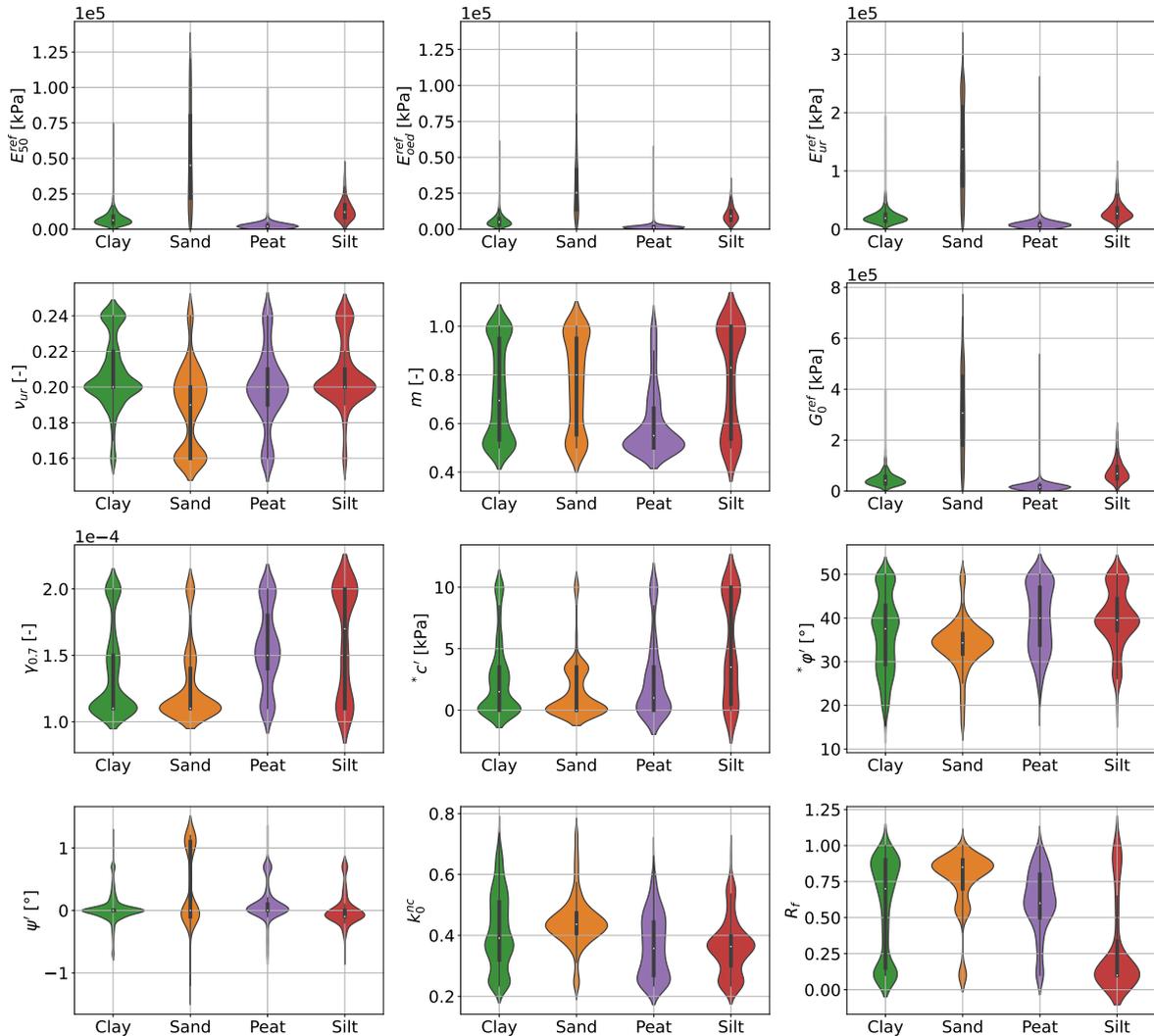


Figure 4.7: Model parameters violin plot

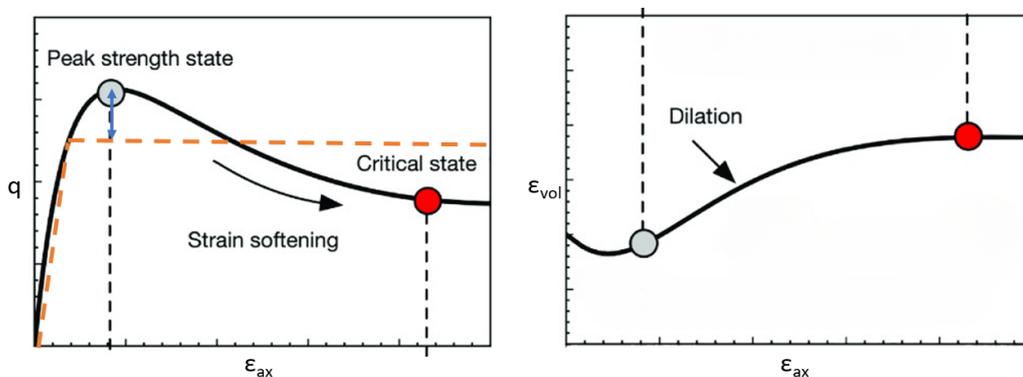


Figure 4.8: Explanation of lower friction angles for sand (Cui et al. 2021)

4.2 Correlations

Correlations describe the strength of association between continuously measured parameters. There are multiple different methods of quantifying this association although this research makes use of the Spearman 1987 method, additional information is provided in Section 3.8.2. Each parameter obtained during this research is compared to each other and is visualised using a heat map which is presented in Figure 4.9. Starting in the bottom left corner of the x-axis with 5 soil properties, followed by 2 soil parameters and 12 optimised model parameters, the same list can be found on the y-axis. The dashed black lines indicate the different parameter groups.

A score of 1 means a perfect positive correlation, this is relatively easy to see when looking at the diagonal where a parameter is compared to itself. A score of -1 means a perfect negative correlation which can be seen when comparing the optimised $*\phi'$ against k_0^{nc} since the relationship between the two was kept constant ($K_0^{nc} = 1 - *\phi'$) and is negatively orientated.

What can be seen straight away is that the soil properties among themselves have a very strong correlation with each other, this was to be expected since some of them were used to calculate one another, but this was not really the point of interest. What is interesting to see however, is the strength of association between soil properties and the soil parameter ϕ' , and the optimised stiffness model parameters. For example, soils with a large unit weight γ were expected to have a higher E_{50}^{ref} which is reflected in the high positive correlation. Furthermore, when increasing the water content, the stiffness tends to decrease which is shown with a strong negative correlation.

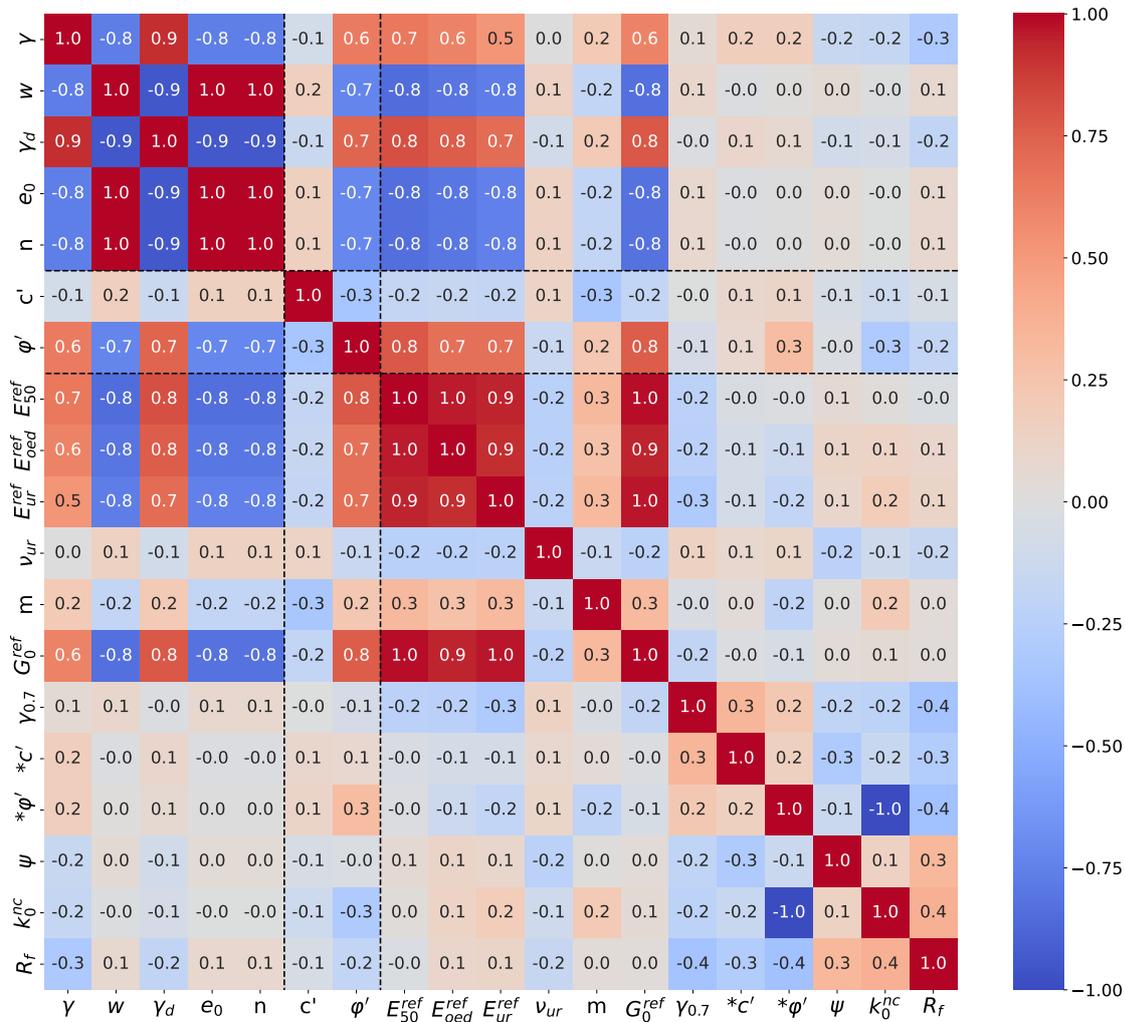


Figure 4.9: Correlation heat map of all collected parameters

4.3 Single regression

The previous section showed the amount of correlation between different parameters and the parameters seemed to have some form of association to one another in different degrees of strength. Although this is a nice piece of information, it can't really be used in more practical applications. Which is why in addition two different single regression analysis have been performed: simple linear regression and non-linear exponential regression, additional information regarding these techniques is provided in Section 3.6. The relevant results between parameters from different groups with $r^2 \geq 0.4$, are presented in Table 4.1. The best found fit was between the soil property γ_d and the soil parameter ϕ' , Figure 4.10 presents how such a fit would look like. The outliers were nicely removed by the DBSCAN algorithm to increase the accuracy of the fit although a lot of scatter is still visible, as expected. The r_{ref}^2 has been added to the table which is the score of the (non)-linear regression without outlier removal. This can be considered the true baseline since the r^2 was already slightly boosted by the more advanced clustering algorithm DBSCAN, the slight increase is in the range of 0.01-0.04. This is one of the motives why this research will look into more advanced machine learning techniques, to attempt obtaining even better scores.

Independent x	Dependent y	Linear: $y = ax + b$				Exponential: $y = be^{ax}$			
		Coefficients		r^2	r_{ref}^2	Coefficients		r^2	r_{ref}^2
		a	b	[-]	[-]	a	b	[-]	[-]
γ	ϕ'	1.81	-2.34	0.48	0.46	0.076	7.48	0.44	0.42
w	ϕ'	-3.65	30.29	0.36	0.33	-0.16	29.78	0.40	0.37
γ_d	ϕ'	1.23	13.15	0.59	0.55	0.05	14.53	0.57	0.54
e_0	ϕ'	-2.74	31.87	0.44	0.41	-0.12	31.78	0.48	0.45
n	ϕ'	-26.43	40.74	0.53	0.51	-1.08	45.16	0.49	0.47
n	E_{50}^{ref}	-77349	56934	0.42	0.41	-4.51	86407	0.43	0.41
n	E_{oed}^{ref}	-43684	33431	0.42	0.41	-4.18	52143	0.45	0.43
n	E_{ur}^{ref}	-200272	149831	0.45	0.44	-3.95	193664	0.46	0.44
n	G_0^{ref}	-452334	335518	0.49	0.47	-4.29	500447	0.53	0.51

Table 4.1: Relevant linear and non-linear exponential regression results

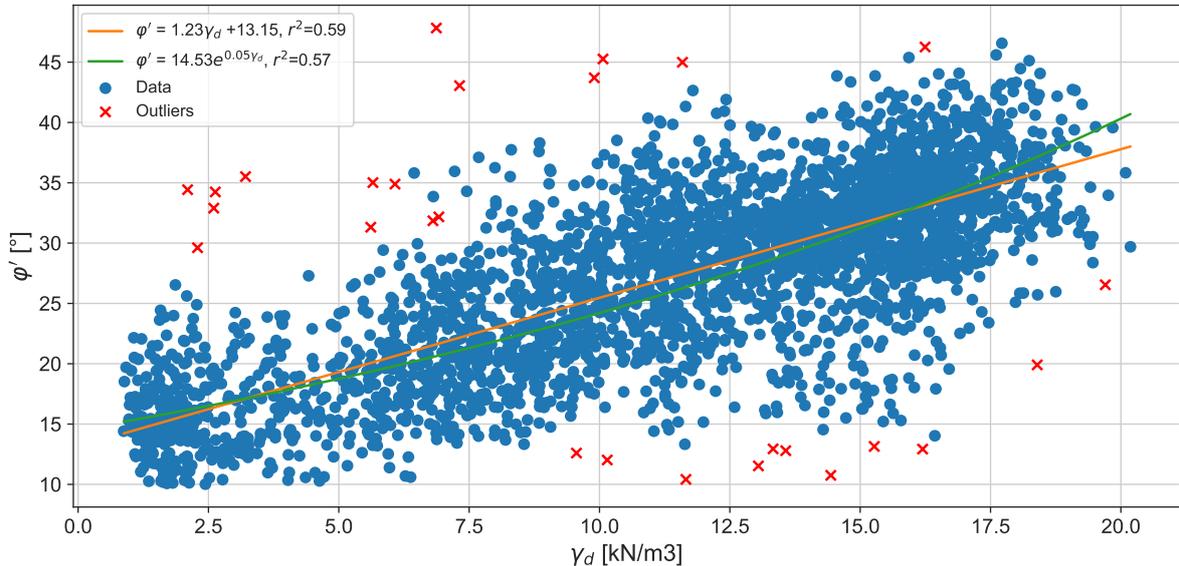


Figure 4.10: Best found r^2 , γ_d vs ϕ'

The above analysis compared correlations between two separate parameters, it is expected that when multiple input parameters are used, the predictions improve. The soil properties were selected for the multiple linear analysis as is shown in Equation 4.1 and the results are presented in Table 4.2. It was able to make reasonable predictions for the same parameters as the simple linear- and exponential regression

methods. Which also matches which the expectation upfront, when looking at the correlation heat map, this showed strong correlations between these parameters. No outliers were removed for this analysis, so they need to be compared to the r_{ref}^2 from the previous analysis. Including more parameters shows an increase in the range of 0.01-0.09, which is not a significant improvement.

$$y = a_1\gamma + a_2w + a_3\gamma_d + a_4e_0 + b \quad (4.1)$$

Dependent y	Multiple linear regression					r^2 [-]
	a_1	a_2	a_3	a_4	b	
ϕ'	-0.63	0.15	1.55	-0.17	20.06	0.56
E_{50}^{ref}	-11800	-808	10591	3063	85042	0.46
E_{oed}^{ref}	-7145	-1094	6040	1578	57220	0.47
E_{ur}^{ref}	-35660	-6527	29287	8866	286430	0.53
G_0^{ref}	-77806	-9104	65504	17584	607851	0.56

Table 4.2: Relevant multiple linear regression results

The r^2 scores are in the range of 0.5, which is still considered a strong correlation according to Table 2.4, although using linear techniques on such a non-linear material seems inappropriate. To further elaborate the relatively small increase, another multiple linear regression is performed using only the γ_d and e_0 as input, and ϕ' as output. This way, the results can be visualised in a 3d plot as is done in Figure 4.11. The red plane is the result of the multiple linear regression as is Equation 4.2. A negligible small improvement in r^2 was obtained compared to using only γ_d as input, most likely due to the strong correlation between γ_d and e_0 which seems to be exponential.

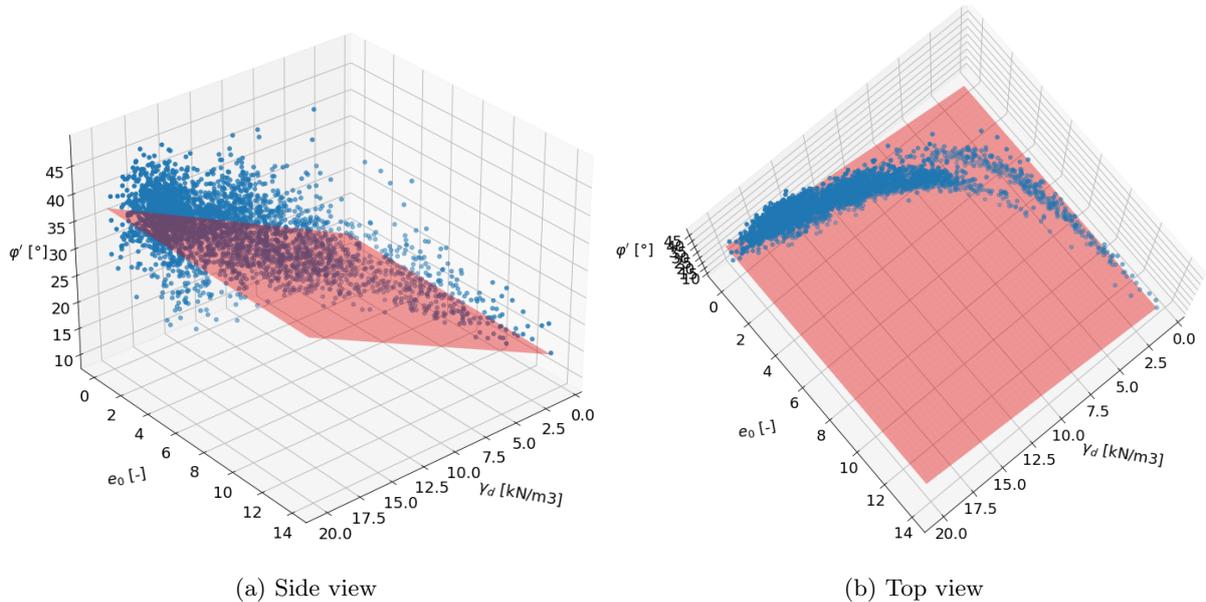


Figure 4.11: Double parameter input

$$\phi' = 1.189\gamma_d - 0.047e_0 + 13.74, \quad r^2 = 0.55 \quad (4.2)$$

4.4 Machine learning

This section presents the results and discussion of the machine learning analysis, additional information regarding how the analysis was performed can be found in Section 3.7.

4.4.1 Quick scan

Before the actual analysis was conducted, a quick and rough scan on the different machine learning models was performed by using parameter input combination 5, meaning all the soil properties were used as input (the different parameter input and output combinations are presented in Table 3.8). Subsequently all of the individual output parameters were predicted (output set 1) and every output parameter with a score $r^2 \geq 0.4$ has been selected. Not surprisingly, the same parameters which showed a strong correlation in the heat map, and were found in the single regression analysis were found here as well, see Figure 4.12. The Multiple Linear Regression is used as a baseline and is plotted as the dashed black line, the other ML models have been normalised against the r^2 score of the MLR. The Support Vector Machine was dropped from the graph due to poor performance, besides that, most of the other models tend to have an overall higher r^2 score which was to be expected.

It is quite interesting to see the Artificial Neural Network (ANN) perform so well when recalling it did relatively poor in the studies from Duffy 2019 (286 data entries) and Yu 2022 (526 data entries). The most likely cause for this is that this study had considerably more data available (3073 data entries), large amounts of data are required to properly train this type of model. The Kernel Ridge Regressor (KRR) seemed to have the best performance and the Gradient Boosting Regressor (GBR) scored the best out of the tree-based algorithms. These three models are selected for further research purposes in combination with the baseline Multiple Linear Regressor (MLR).

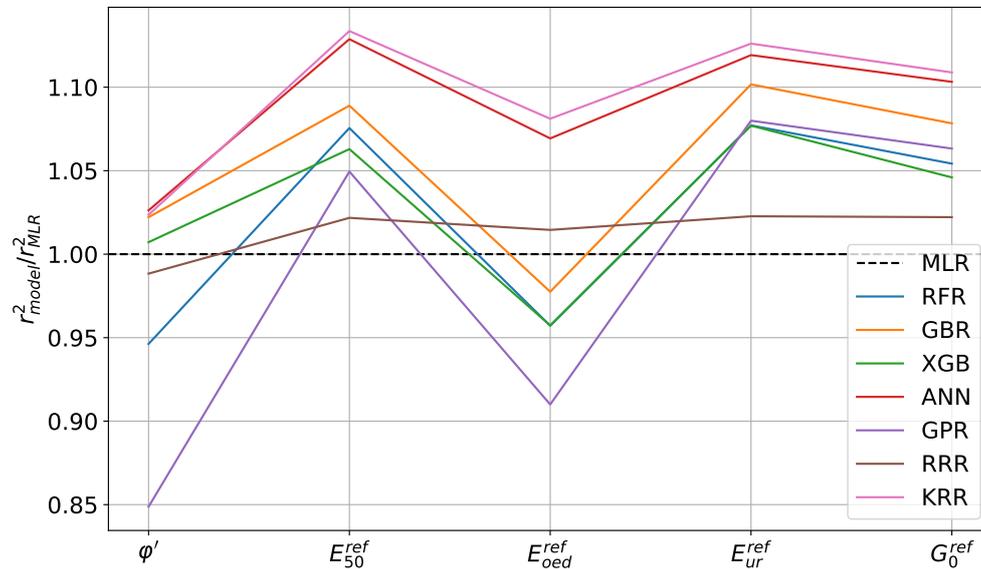


Figure 4.12: Comparison between models, in legend order; Multiple Linear Regression, Random Forest Regressor, Gradient Boosting Regressor, eXtreme Gradient Boosting, Artificial Neural Network, Gaussian Process Regressor, Regular Ridge Regressor, Kernel Ridge Regressor

4.4.2 Single output predictions

Part of the hypothesis was that if multiple input parameters are provided, the machine learning models would start to perform better, since they possess more information regarding the soil samples used to obtain the output results. Figure 4.3 showed, as a first check, to see if the bandwidth would become narrower when looking at the range of friction angles for samples which were classified as clay. When including the features like the initial void ratio or the unit weight, the bandwidth of the outcome indeed became narrower and corresponded with initial expectations, e.g. lower void ratio resulted in higher friction angles. This was the incentive for exploring machine learning models, to attempt and make better predictions of the soil parameters and optimised model parameters by including more soil properties as input. This was done by varying the input parameter sets which were presented to the model by adding an additional parameter each consecutive input combination.

The first output set consisted out of predicting each soil parameter and optimised model parameter individually, the parameters with a score of $r^2 \geq 0.4$ are presented in Figure 4.13. The different input parameters sets are displayed on the y-axis, the machine learning models on the x-axis and the coefficient

of determination on the z-axis. The letter s on the y-axis represents the soil type as classified in the laboratory and the best score has been displayed in red. The first thing which can be noticed is that overall the KRR produces the best results with ANN and GBR not far behind. MLR preforms the poorest which was to be expected as it is the most simplistic model, besides, it was used as a baseline to compare/evaluate the performance of the more advanced models. A clear trend upwards can be seen when increasing the number of input parameters, especially adding the unit weight and water content causes significant increases in the performance of the model. In general, parameter input combination 4 seems to obtain the best score which is classified in the range of strong correlation when looking at Table 2.4. Combination 5, which adds the initial void ratio, might be introducing some uncertainty to the model which decreased the performance ever so slightly.

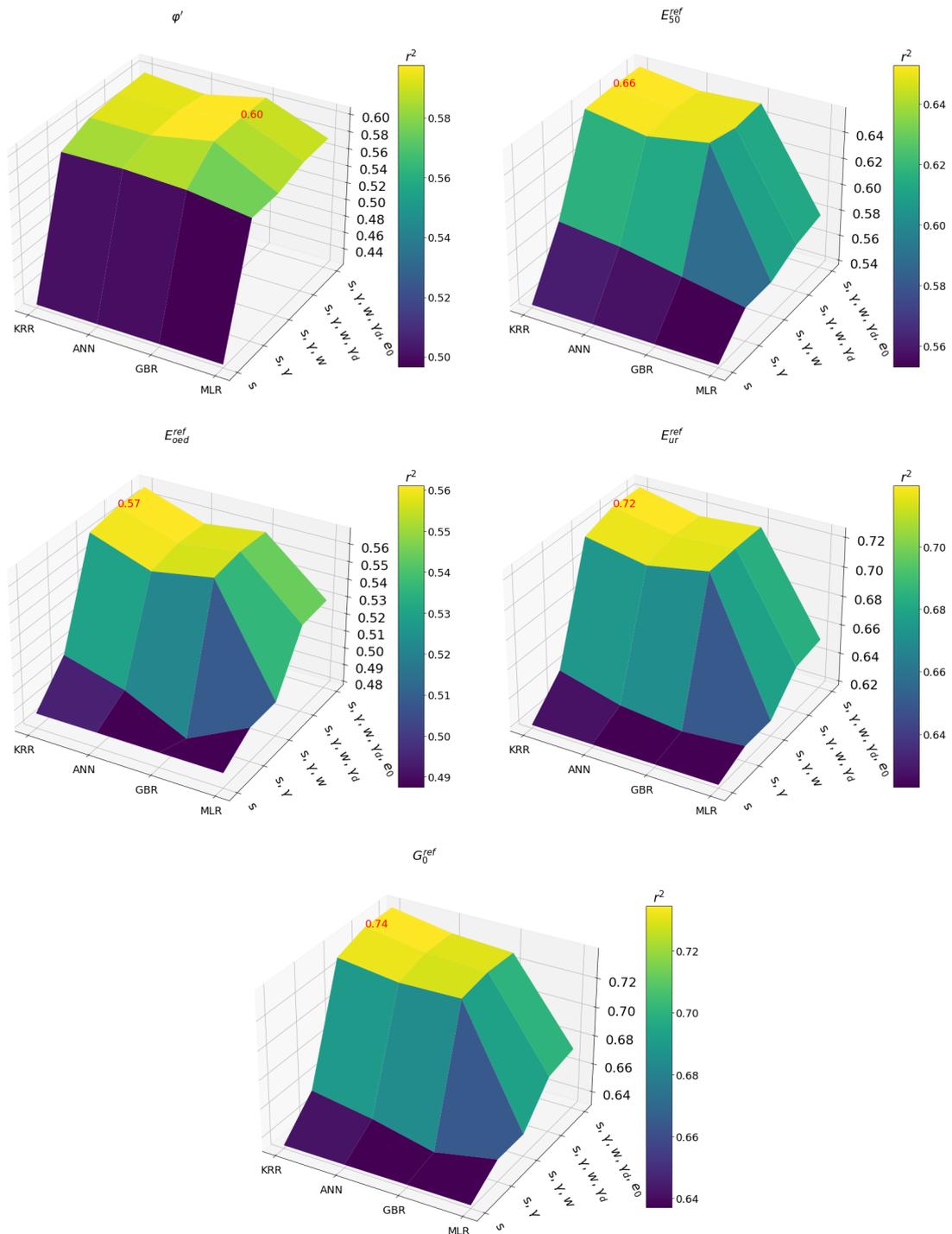


Figure 4.13: 3D plots of output parameter set 1 using k-fold cross validation

This analysis was done using the Quantile Transformer to scale the data, grid search CV to tune the hyperparameters (see the code snippets in Appendix E) and the k-fold for the actual Cross Validation (CV) part rather than the group k-fold. The k-fold strategy is used most commonly, but a big downside can be that the training data can come from the same place/group which is used to test the data, making the predictions potentially more favourable. The group k-fold ensures that the same group is not included in both training and testing data. The groups in the group k-fold are supposed to match how the predictions would be applied in practice. For example, when data is available from multiple wells, and the practical application is for the ML model to make predictions on a new well, than the wells would be the groups in this case. Since both training and testing on the same well would generally result in better CV scores. The groups in this research are not as obvious as the previous example which was given, since each measurement was performed on a different sample. If there are no clear groups, it is also possible to include a spatial component. Not all samples contained coordinates, but each sample did contain a project number. The project numbers could be used as the groups in this situation since these are typically linked to the same location or at least fairly close to each other. Furthermore, it makes sense to choose the project number, since the practical application for these models would be to make predictions for new projects.

Before performing an additional analysis using the group k-fold, first an analysis was done on the variability of the outcomes from the regular k-fold CV. K-fold splits the data into k number of folds for which each k fold is used as a testing set and $k - 1$ is used as training data (see also Section 3.7.2). These splits, when turning on the shuffle, are different every time. The random state parameter can be used to make the results reproducible. The variability of the outcomes depending on the way the data is split by using several random seeds can be mapped, and therefore also the overall sensitivity of the model. This validation is done by running a 100 simulations with different seeds, while predicting E_{50}^{ref} in combination with input parameter set 4 and KRR, the results are presented in Figure 4.14.

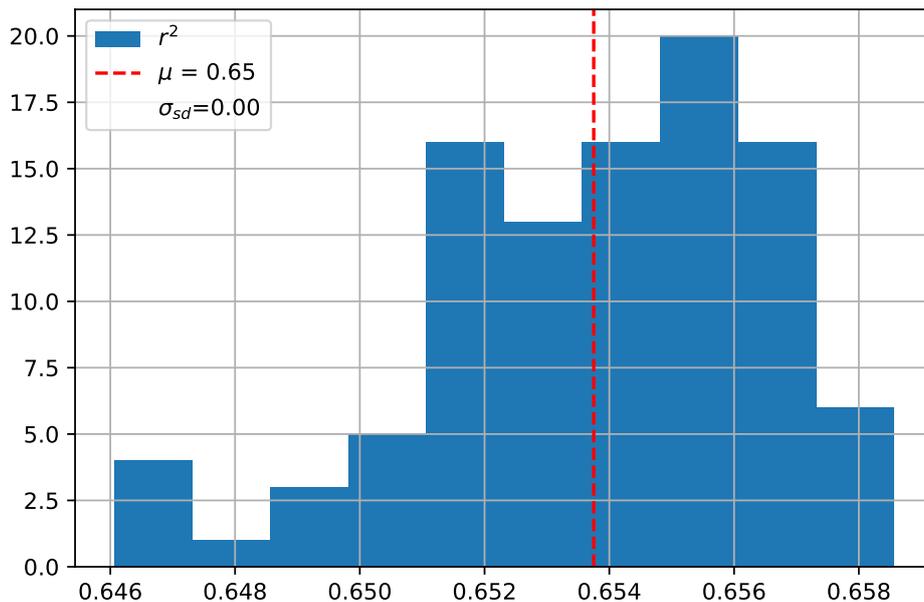


Figure 4.14: Validation of the variability of the model and the influence of k-fold

At first glance it looks like there is quite a spread on the outcome of r^2 , but when looking at the scale of the x-axis and the standard deviation in the legend it becomes clear that the shuffle has no real influence on the performance of the model, which is a favourable attribute. To further investigate the performance and reliability of the models, the group k-fold analysis is performed. As mentioned above, the project numbers will be used as the groups, since that is where the model can be used for in practice, attempting to predict the parameters for a new project. The results are presented in Figure 4.15 and it can be seen that results are very similar to the regular k-fold strategy, arguably slightly lower. This means that these models could be used to make predictions on an entire new project without gathering data upfront.

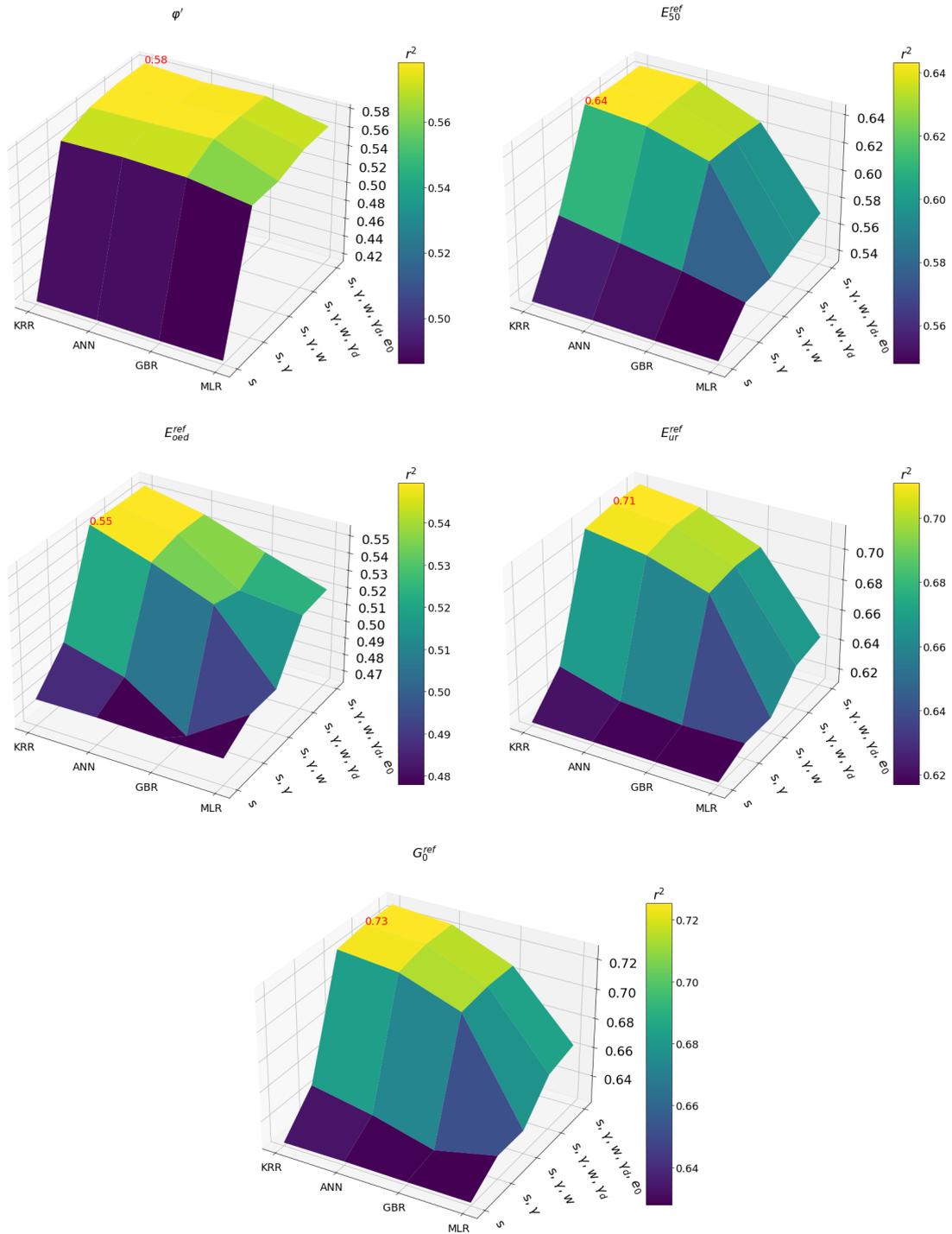


Figure 4.15: 3D plots of output parameter set 1 using group k-fold cross validation

4.4.3 Multi output predictions

The analysis above was to predict individual output parameters (output set 1), now the entire array of soil parameters (output set 2) and the entire array of optimised model parameters (output set 3) will be used. Predicting multiple outputs was done by using the *MultiOutputRegressor*, which is basically a wrap around the machine learning model. Only regular k-fold cross validation was performed since the previous analysis showed that group k-fold obtained similar results.

Figure 4.16a present the results of the soil parameter array which consisted out of the cohesion and internal friction angle. The best results are obtained using the KRR in combination with all of the soil properties as input. Note that most of the performance increase came from introducing the unit weight

and every additional parameter only resulted in small increments. On top of that, the MLR actually performs quite well when comparing it to the more advanced models. The best r^2 score is 0.33 which still shows medium correlation, while ϕ' individually scored an r^2 of 0.60 when looking back at Figure 4.13. This drop is most likely due to the low correlation between the cohesion and the input parameters. This was also seen in the heat map and this is also why there was no individual graph for the cohesion presented in the first place.

Figure 4.16b present the results of the optimised model parameter array which consisted out all of the HS small model parameters. Note that the ANN is not present, this is due to extremely poor performance which made the graph unreadable. The best result $r^2 = 0.35$ was obtained with the KRR and all of the soil properties as input, this is quite significantly lower than some of the model parameters which were predicted individually. The build up to this score was more gradually than with the soil parameters when including additional parameters.

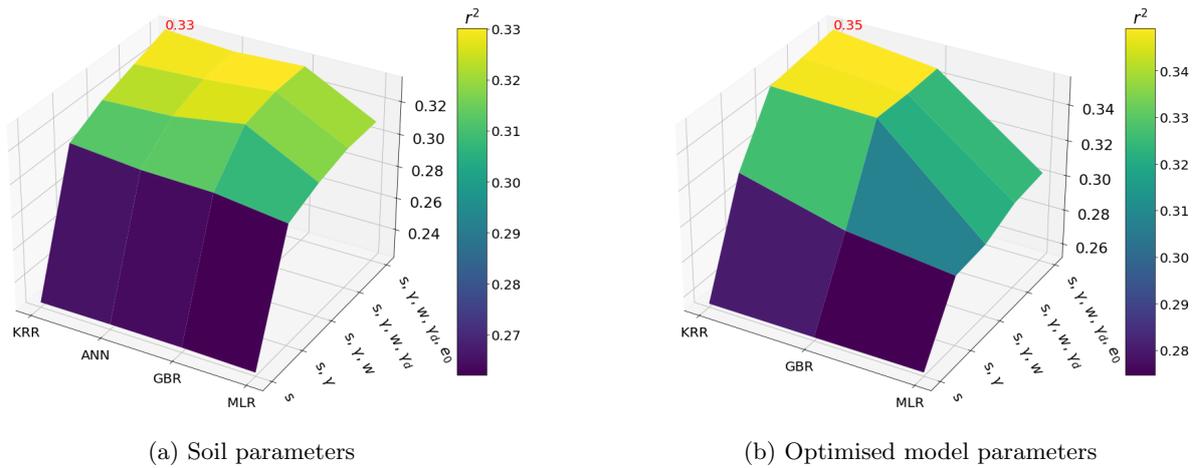


Figure 4.16: 3D plots of output parameter set 2 and 3

There are quite a view parameters in the array of optimised model parameters which showed little to no correlation. By excluding some parameters with very low correlation and/or a low sensitivity, a higher score might be obtained. For example when looking at the heat map in Figure 4.9 and the sensitivity analysis in Figure 3.6, it would make sens to drop some parameters like: ν_{ur} , $\gamma_{0.7}$ and k_0^{nc} which was kept constant relative to ϕ' anyways. Therefore a new output array has been created with the parameters which showed either a lot of correlation or were very sensitive/important: $[E_{50}^{ref}, E_{oed}^{ref}, E_{ur}^{ref}, G_0^{ref}, *c', * \phi', R_f]$. The results are presented in Figure 4.17 and removing some of parameters did indeed result in an increase in performance.

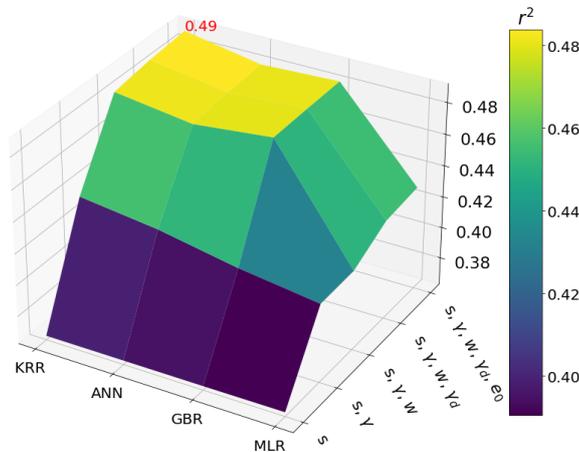


Figure 4.17: Optimised model parameters, adjusted output array

4.4.4 General discussion

A common trend which can be seen in basically all of the graphs is that increasing the number of parameters does indeed increase the performance of the models. With that being said, most of the increase occurred when introducing the unit weight and the water content. It was to be expected that adding the dry volumetric weight would not result in a large increase since the model already has the information required to calculate it. A significant improvement was however expected upon introducing the initial void ratio, but this was not the case. Either small increments appeared and it sometimes even slightly decreased in terms of performance. At this point one might wonder if adding even more parameters could be beneficial, or that this is a ceiling through which it cannot further improve. It's important to remember that inaccuracy's, measurement errors and sample disturbances take place in the laboratory as well.

4.5 Automated Parameter Determination

Selecting constitutive model parameters have proven to remain a challenge which was both the reason for this research and why another ongoing research project aims to create an Automated Parameter Determination (APD) system. This system derives the model parameters for the HS small model from in situ tests (CPT) and is based on graph theory, which makes the parameter determination transparent and adaptable, it was introduced by van Berkom 2020. Some follow up papers have been published as a part of the ongoing research project such as, Marzouk et al. 2023 which aimed on adding the DMT to the framework and Brinkgreve and Brasile 2022 which applied the framework in a practical example. This research will not perform an entire study on the graph based method, for that, the above papers are referenced. What will be done is a comparison between the model parameters, which were found in this research, and the parameters determined by the graph based method.

The framework requires CPT parameters which first had to be obtained in order to determine the model parameters. The locations of the soil samples used in the triaxial test were already traced back to the borehole location they came from (see Figure 3.2) and all nearby CPT's, within a radius of 3 meters, were added to the database. Out of the original 3073 triaxial tests, 105 samples could be linked to a nearby CPT. The significant reduction in database is also the reason that no machine learning models could be trained to find patterns between the CPT parameters and the optimised model parameters which were determined in this research.

Two CPT's were selected and can be found in Appendix H, the yellow bar shows where the samples were obtained, these will be referred to as DKM5 and DKM32. The optimised model parameters (Opt.) determined in this study and the parameters determined by the APD are presented in Table 4.3. The APD is more conservative when looking at the stiffness parameters and the friction angle, the friction was already expected to be on the lower end. Furthermore, the APD makes use of a negative dilatancy angle to simulate the 'softening' behaviour.

Parameter	DKM5		DKM32		Unit
	Opt.	APD	Opt.	APD	
E_{50}^{ref}	5939	3472	4778	3985	kN/m^2
E_{oed}^{ref}	3563	2083	3584	2391	kN/m^2
E_{ur}^{ref}	24052	14089	19353	15346	kN/m^2
G_0^{ref}	40384	46563	32493	39070	kN/m^2
$\gamma_{0.7}$	$1.2 \cdot 10^{-4}$	$2.55 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$	$2.14 \cdot 10^{-4}$	-
m	0.92	1.0	0.95	1.0	-
p^{ref}	100	100	100	100	kN/m^2
ν_{ur}	0.16	0.2	0.19	0.2	-
c'	0	0	0	0	kN/m^2
ϕ'	30.8	25.6	27.3	26.8	°
ψ	0	-5.2	0.1	-5.4	°
R_f	0.65	0.9	0.8	0.9	-
K_0^{nc}	0.49	0.54	0.54	0.52	-

Table 4.3: Comparison with the APD system

Figure 4.18 shows the simulations of the triaxial test using these parameter sets and the actual

measurements from the laboratory, the r^2 score for each simulation is added to the legend. This graph confirms that the strength in particular is modelled more conservatively with the APD than the optimised model parameters from this study. This is also related to the fact that performing the triaxial test on softer soils often results in (too) high friction angles, raising the question of whether these high friction angles are real. These high friction angles are often not used in engineering practices, in that regard the parameters from the APD are more focused on what is commonly used in practice. The APD system scored lower on the r^2 , but keep in mind that the parameters determined in this research could make use of the actual laboratory test, while the APD system only saw a nearby CPT. When looking at DKM5 it can be seen that the softening in the top part of the curve is actually modelled quite well by the APD system, by using this negative dilatancy. This research tried to make as little use as possible of negative dilatancy, the parameter R_f first appeared in the optimisation algorithm, which often made a compromise between peak- and residual strength. This left little room for optimising the dilatancy angle.

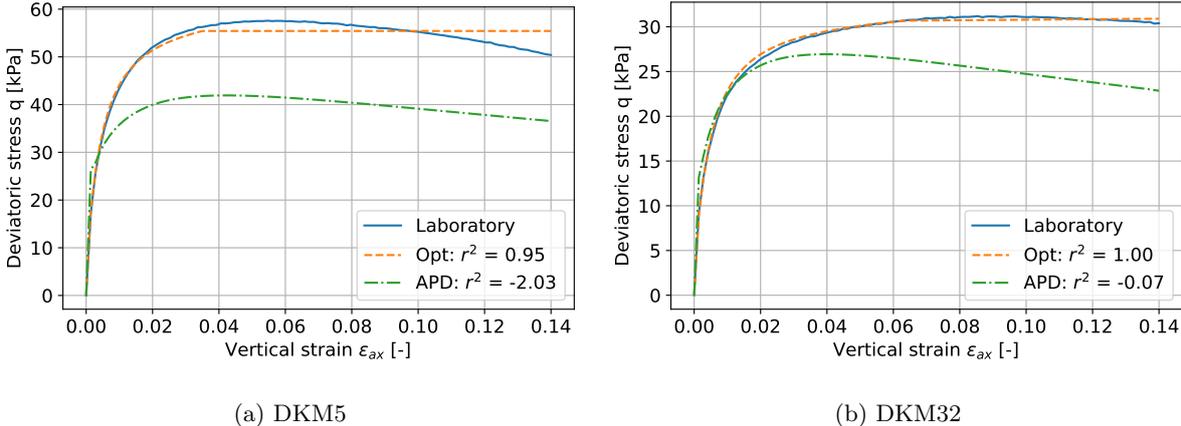


Figure 4.18: Triaxial test simulation comparison with the APD system

Chapter 5

Conclusions

The incentive for this study was to provide guidance in the parameter determination for constitutive models. This stemmed from the 3073 Consolidated Isotropic Drained/Undrained multi stage triaxial test results (stored in text files) which were made available by Fugro for this study. A three-part hypothesis was formulated. Firstly, it was expected that the entire measured stress-strain path from the text files could be used to optimise constitutive model parameters, compared to using soil parameters, basic correlations and default values from the literature (the traditional method). Secondly, the database consisting out of several different types of parameters could be used to discover (new) statistics and correlations. Thirdly, it was presumed that advanced machine learning models could be used to predict soil parameters and optimised model parameters using soil properties as input. Furthermore, it is expected that an increasing number of input parameters will improve the performance of the machine learning models. Several sub-questions were formulated to assist answering the following main research question:

How can a database of triaxial tests contribute to the parameter determination, optimisation and selection for advanced constitutive models?

- The Hardening Soil with small-strain stiffness (HS small) was selected as constitutive model to describe the stress-strain relationship, which requires 13 model parameters. An initial parameters set was derived by expanding upon the soil parameters, using common correlations and default values from the literature (the traditional method). These soil parameters, c' and ϕ' , were determined at 2% strain using the stress paths from the text files (NEN-EN-ISO 2018b). A simulation of the triaxial test was performed using the SoilTest facility in the PLAXIS software package in combination with the HS Small model and this initial parameter set. The stress-strain path of the simulation was compared to the one in the text file using the r^2 evaluation metric, this resulted in a relatively low median r^2 of -0.61 for all of the text files. The initial parameter set can be determined by expanding upon the soil parameters as described in this research, also referred to as the traditional method, and could be used as a first estimation.
- An algorithm was developed which attempted to match the simulated/calculated stress-strain path of the triaxial test to the stress-strain path available in the text files by varying the HS small model parameters in a predetermined sequence. The predetermined sequence was based on the impact of a change of an HS Small parameter on the calculated/simulated triaxial test result. Only matching the stress-strain curve from the triaxial test resulted in a singularity, the optimised model parameters could be very differently depending on the initial parameter set. To overcome this problem, an additional feature was matched simultaneously, for drained triaxial tests the $\varepsilon_{vol}-\varepsilon_{ax}$ and for undrained triaxial tests the $u_{excess}-\varepsilon_{ax}$. The algorithm was deployed for all the text files and a significant improvement in r^2 was observed, the median r^2 increased from -0.61 to 0.81. It was quite remarkable that the softer soil types clay and peat scored overall higher on the r^2 than sand and silt. This was investigated for a number of results and it is most likely due to the fact that HS small cannot account for softening behaviour. The algorithm tried to find a compromise between the peak strength and residual strength, this is also why sandy soils had an overall lower friction angle compared to softer soil types, due to the influence of the long residual tail in the stress-strain curve. The developed algorithm is capable of optimising the HS small parameters, which were initially determined using the traditional method, by matching the simulated/calculated stress-strain path to the entire stress-strain path from the text files.

- The soil properties (γ , w , γ_d , e_0 and n) showed a strong correlation with 5 parameters, the soil parameter ϕ' and the optimised model parameters E_{50}^{ref} , E_{oed}^{ref} , E_{ur}^{ref} and G_0^{ref} . The single linear/exponential regression managed to find a fit with an $r^2 > 0.4$ for these parameters. Still, a lot of scatter in the graphs was noticed which is why more advanced machine learning methods were explored. The machine learning models: Artificial Neural Network, Gradient Boosting and Kernel Ridge Regression were selected based on an initial analysis on the data set, and a literature study (in which similar models were used). Different input sets were presented to the machine learning model, input set 1 only consisted out of the soil type as classified in the laboratory, every successive input set added a soil property in the same order as motioned above. The soil parameters and optimised model parameters are the output. After training, the machine learning models were able to make significant better predictions for the same 5 parameters, with an increase of r^2 in the range of 0.05-0.27 (average of 0.2), compared to the single linear/exponential regression methods. Providing the machine learning models with more soil properties generally resulted in an increase in performance, especially when adding the unit weight and water content. Introducing the dry unit weight and initial void ratio only resulted in changes of $r^2 < 0.01$, probably due to the strong correlations between the input parameters themselves. All the results were evaluated using regular k-fold and group k-fold, with project numbers as groups. The trained machine learning models, which are made available with this study, could be used to select the soil parameter ϕ' and the optimised HS small model parameters E_{50}^{ref} , E_{oed}^{ref} , E_{ur}^{ref} and G_0^{ref} with an r^2 in the range of 0.6-0.74 by using 1 to 5 input parameters (soil properties). The linear/exponential regression results come in the form of equations that can also be used, and they may even be easier to apply, but these score significantly lower, with an r^2 in the range of 0.41-0.56.

At the end of each bullet point above, a recommendation is given on how this research could be applied. It is however important to note that these methods, results and conclusions are based on the triaxial test and have not been validated for other tests and engineering practices.

Chapter 6

Recommendations

The conclusion already included recommendations on how the results of this research could be used. This chapter aims to provide recommendations for future research based on the outcomes and findings of this study.

After the data was collected into a well-structured database, some initial key features could be visualised. Features like the test type, soil type and test date are given in Figure 3.1. When looking at the test date plot, it appears that during the period of 2005-2010 little to no tests have been conducted, but this is actually not the case. Most files in this period were stored in a binary format that could not be read without modifying internal software, which was not feasible within the time window of this research. In machine learning, the more training data available, the better the model becomes. It would be beneficial to be able to read the data from that period as well and increase the size of the database. Moreover, this study only used data from the Netherlands, while data from other countries are also available.

Continuing on the features of the database, this research only made use of the CIUMS and CIDMS tests, when in fact other test types are available as well. It would be interesting to extract even more from the current database by looking at the other test types. The single stage tests for example, reach failure in every stage in comparison to the multi stage tests. Even though different samples are required and will therefore slightly differentiate, results are generally more representative. The reason multi stage testing was chosen in this study is that this used to be the common method and therefore a significantly larger database is available. In addition, UU tests and anisotropic tests are available as well. Especially the anisotropic tests in combination with constitutive models which can account for anisotropy like (Creep) S-Clay1S or PM4Sand would be interesting to further investigate.

This research aimed to properly describe the soil behaviour by using triaxial test measurements from the laboratory to optimise the model parameters for the HS small model. Several laboratory tests are available, such as the Oedometer or the Direct Simple Shear (DSS) test, which measure different characteristics of soil behaviour. The soil behaviour can be described even better by including more than one type of laboratory test. Besides looking at different laboratory tests, different constitutive models can also be explored. One of the problems encountered during this study was that HS small cannot account for softening behaviour. Models such as Soft Soil (Creep) and Modified Cam-Clay do take this into account, and when the Oedometer is included, the creep feature can also be modelled.

The current optimisation algorithm made the match for the entire stress-strain curve measured in the laboratory, which sometimes included a long residual tail which had a lot of influence. The part of the curve which is attempted to be matched can be adjusted for practical applications. For example, a distinction could be made between Ultimate Limit State and Serviceability Limit State calculations. For which the ULS should be matched with the residual strength and the SLS with the peak strength, by e.g. applying weights for different parts of the curve.

It was shown that in general, more input parameters resulted in better performance of the machine learning models. Especially when including the unit weight and water content, including more features either resulted in very small increases or sometimes even slight decreases, most likely because additional uncertainty was introduced or due to correlations between soil properties. However, just like adding tests such as the DSS or Oedometer to improve the description of the soil behaviour, one could use additional index tests as input to try and increase the performance of the machine learning model. Tests like the Particle Size Distribution (PSD) or the consistency limits provide a lot of information on the soil samples, which could potentially help to make better predictions.

Bibliography

- Ayodele, T. O. 2010, ‘Types of machine learning algorithms’, *New advances in machine learning* **3**, 19–48.
- Barten, A. P. 1987, The coefficient of determination for regression without a constant term, *in* ‘The Practice of Econometrics: Studies on Demand, Forecasting, Money and Income’, Vol. 15, Springer, pp. 181–189.
- Bentley 2022, *Plaxis CONNECT Edition V22.02 Material Models Manual*.
- Benz, T. 2006, *Small-strain stiffness of soils and its numerical consequences*, PhD. Thesis. Institut für Geotechnik, Universität Stuttgart.
- Bergstra, J. and Bengio, Y. 2012, ‘Random search for hyper-parameter optimization’, *Journal of Machine Learning Research* **13**, 281–305.
- Bolton, M. 1987, ‘Discussion: The strength and dilatancy of sands’, *Géotechnique* **37**(2), 219–226.
- Borgonovo, E. and Plischke, E. 2016, ‘Sensitivity analysis: A review of recent advances’, *European Journal of Operational Research* **248**(3), 869–887.
- Bre, F., Gimenez, J. M. and Fachinotti, V. D. 2018, ‘Prediction of wind pressure coefficients on building surfaces using artificial neural networks’, *Energy and Buildings* **158**, 1429–1441.
- Breiman, L. 2001, Random forests, *in* ‘Machine learning’, Vol. 45, Springer, pp. 5–32.
- Brinkgreve, R. 2005, Selection of soil models and parameters for geotechnical engineering application, *in* ‘Congres Austin, Texas’, American Society of Civil Engineers (ASCE), pp. 69–98.
- Brinkgreve, R. B. and Brasile, S. 2022, Automatic finite element modelling and parameter determination for geotechnical design, *in* ‘Workshop Numerische Methoden in der Geotechnik’, Vol. 53, Grabe, J. TUHH, pp. 83–97.
- Brinkgreve, R., Engin, E. and Engin, H. K. 2010, Validation of empirical formulas to derive model parameters for sands, *in* ‘Numerical methods in geotechnical engineering’, Vol. 137, CRC Press/Balkema, Rotterdam, the Netherlands, p. 142.
- Brinkgreve, R., Kappert, M. and Bonnier, P. 2007, ‘Hysteretic damping in a small-strain stiffness model’, *Proc. of Num. Mod. in Geomech., NUMOG X, Rhodes* pp. 737–742.
- Cameron, A. C. and Windmeijer, F. A. 1997, ‘An r-squared measure of goodness of fit for some common nonlinear regression models’, *Journal of econometrics* **77**(2), 329–342.
- Chai, T. and Draxler, R. R. 2014, ‘Root mean square error (rmse) or mean absolute error (mae)’, *Geoscientific model development discussions* **7**(1), 1525–1534.
- Charbuty, B. and Abdulazeez, A. 2021, ‘Classification based on decision tree algorithm for machine learning’, *Journal of Applied Science and Technology Trends* **2**(01), 20–28.
- Chen, T. and Guestrin, C. 2016, Xgboost: A scalable tree boosting system, *in* ‘Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining’, pp. 785–794.
- Conza, K. A. L. 2020, ‘Deep learning for decision support in dermatology’, *MSc thesis, Technical University of Denmark*.

- Cord, M. and Cunningham, P. 2008, *Machine learning techniques for multimedia: case studies on organization and retrieval*, Springer.
- Crick, M. and Hill, M. 1987, The role of sensitivity analysis in assessing uncertainty, in 'Uncertainty analysis for performance assessments of radioactive waste disposal systems', pp. 116–130.
- Cui, K., Hu, B., Cui, A., Li, J., Wei, E. and Zhang, Z. 2021, 'An extended super/subloading surface model for soft rock considering structure degradation', *PLoS One* **16**(10), e0258813.
- Das, S., Dey, A., Pal, A. and Roy, N. 2015, 'Applications of artificial intelligence in machine learning: review and prospect', *International Journal of Computer Applications* **115**(9).
- De Winter, J. C., Gosling, S. D. and Potter, J. 2016, 'Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data.', *Psychological methods* **21**(3), 273.
- Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P. and Meester, L. E. 2005, *A Modern Introduction to Probability and Statistics: Understanding why and how*, Vol. 488, Springer.
- Duffy, K. 2019, 'Assessment of a soil compressibility index using cone penetration testing and machine learning tools', *MSc thesis, University College Dublin*.
- Duncan, J. M. and Chang, C.-Y. 1970, 'Nonlinear analysis of stress and strain in soils', *Journal of the soil mechanics and foundations division* **96**(5), 1629–1653.
- Eberly, L. E. 2007, 'Multiple linear regression', *Topics in Biostatistics* pp. 165–187.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al. 1996, A density-based algorithm for discovering clusters in large spatial databases with noise., in 'kdd', Vol. 96, pp. 226–231.
- Goldscheider, M. 1984, True triaxial tests on dense sand, in 'International workshop on constitutive relations for soils', Balkema, Rotterdam, pp. 11–54.
- Hamby, D. M. 1994, 'A review of techniques for parameter sensitivity analysis of environmental models', *Environmental monitoring and assessment* **32**, 135–154.
- Howard, R. A. 1988, Uncertainty about probability: A decision analysis perspective, in 'Risk Analysis', Vol. 8, Wiley Online Library, pp. 91–98.
- Jaky, J. 1948, Pressure in silos, in 'Proc. 2nd International Conference on Soil Mechanics and Foundation Engineering', Vol. 1, Rotterdam, pp. 103–107.
- labxchange 2021, 'How to interpret violin plots'.
URL: <https://www.labxchange.org/library/items/lb:LabXchange:46f64d7a:html:1>
- Lambe, W. T. 1967, Stress path method, in 'Journal of the soil mechanics and foundations division', Vol. 93, American Society of Civil Engineers, pp. 309–331.
- Latuni, F. 2019, 'Development of road and bridge infrastructure to enhance economic growth in the coastal communities of tuminting district in manado city', *International Journal of Multicultural and Multireligious Understanding* **6**(5), 780–791.
- Lev, A. 2022, 'Xgboost versus random forest'.
URL: <https://www.quak.com/post/xgboost-versus-random-forest>
- Lin, H.-D., Dang, P. H. and Hsieh, Y.-M. 2015, 'Rift: Robust and interpolation-free technique for objective functions in geotechnical inverse analysis', *Computers and Geotechnics* **64**, 96–104.
- Marchetti, S. 2015, Some 2015 updates to the tc16 dmt report 2001, in 'The 3rd International Conference on the Flat Dilatometer', pp. 43–65.
- Marzouk, I., Tschuchnigg, F. and Brinkgreve, R. 2023, Expansion of an automated system for determining soil parameters using in-situ tests, in 'NUMGE 2023', European Conference on Numerical Methods in Geotechnical Engineering (NUMGE), p. 70.

- McCulloch, W. S. and Pitts, W. 1943, ‘A logical calculus of the ideas immanent in nervous activity’, *The bulletin of mathematical biophysics* **5**, 115–133.
- McKinney, W. et al. 2010, Data structures for statistical computing in python, in ‘Proceedings of the 9th Python in Science Conference’, Vol. 445, Austin, TX, pp. 51–56.
- Metzen, J. H. 2023, ‘Comparison of kernel ridge regression and svr’.
URL: https://scikit-learn.org/stable/auto_examples/miscellaneous/plot_kernel_ridge_regression.html
- NEN 2017, ‘NEN 9997-1 +C2 17892-8:2018: Geotechnical design of structures - part 1: General rules’.
- NEN-EN-ISO 2018a, ‘NEN-EN-ISO 17892-8:2018: Geotechnical investigation and testing - laboratory testing of soil - part 8: Unconsolidated undrained triaxial test’.
- NEN-EN-ISO 2018b, ‘NEN-EN-ISO 17892-9:2018: Geotechnical investigation and testing - laboratory testing of soil - part 9: Consolidated triaxial compression tests on water saturated soils’.
- Obrzud, R., Truty, A., Podles, K., Commend, S. and Zimmermann, T. 2018, ‘Z soil. pc 120201 report revised 8.04. 2018’.
- Oliphant, T. E. et al. 2006, *A guide to NumPy*, Vol. 1, Trelgol Publishing USA.
- Parzen, E. 1962, ‘On estimation of a probability density function and mode’, *The annals of mathematical statistics* **33**(3), 1065–1076.
- Pearson, K. 1896, ‘VII. mathematical contributions to the theory of evolution.—III. regression, heredity, and panmixia’, *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character* **1**(187), 253–318.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. et al. 2011, ‘Scikit-learn: Machine learning in python’, *the Journal of machine Learning research* **12**, 2825–2830.
- Puth, M.-T., Neuhäuser, M. and Ruxton, G. D. 2015, ‘Effective use of spearman’s and kendall’s correlation coefficients for association between two measured traits’, *Animal Behaviour* **102**, 77–84.
- Robertson, P. K. 1986, ‘In situ testing and its application to foundation engineering’, *Canadian Geotechnical Journal* **23**(4), 573–594.
- Roscoe, K. H., Schofield, A. and Wroth, a. P. 1958, ‘On the yielding of soils’, *Geotechnique* **8**(1), 22–53.
- Sarker, I. H. 2021, ‘Machine learning: Algorithms, real-world applications and research directions’, *SN computer science* **2**(3), 160.
- Schanz, T., Vermeer, P. and Bonnier, P. G. 1999, The hardening soil model: formulation and verification, in ‘Beyond 2000 in computational geotechnics’, Balkema, Rotterdam, pp. 281–296.
- Schweiger, H., Fabris, C., Ausweger, G. and Hauser, L. 2019, ‘Examples of successful numerical modelling of complex geotechnical problems’, *Innovative Infrastructure Solutions* **4**(1), 1–10.
- Scott, D. W. 2015, *Multivariate density estimation: theory, practice, and visualization*, John Wiley & Sons.
- Skempton, A. 1954, The pore-pressure coefficients a and b, in ‘Geotechnique’, Vol. 4, Thomas Telford Ltd, pp. 143–147.
- Smith, I. M., Griffiths, D. V. and Margetts, L. 2013, *Programming the finite element method*, John Wiley & Sons.
- Spearman, C. 1987, ‘The proof and measurement of association between two things’, *The American journal of psychology* **100**(3/4), 441–471.
- Ti, K. S., Huat, B. B., Noorzaei, J., Jaafar, M. S. and Sew, G. S. 2009, ‘A review of basic soil constitutive models for geotechnical application’, *Electronic Journal of Geotechnical Engineering* **14**, 1–18.

- Tran, T. N., Drab, K. and Daszykowski, M. 2013, 'Revised dbscan algorithm to cluster data with dense adjacent clusters', *Chemometrics and Intelligent Laboratory Systems* **120**, 92–96.
- Tukey, J. W. et al. 1977, *Exploratory data analysis*, Vol. 2, Reading, MA.
- van Berkom, I. 2020, 'An automated system to determine constitutive model parameters from in situ tests', *MSc thesis, Delft University of Technology* .
- Van Rossum, G. and Drake, F. L. 2009, *Python 3 Reference Manual*, CreateSpace, Scotts Valley, CA.
- Verruijt, A. and Van Baars, S. 2007, *Soil mechanics*, VSSD Delft, the Netherlands.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J. et al. 2020, 'Scipy 1.0: fundamental algorithms for scientific computing in python', *Nature methods* **17**(3), 261–272.
- Welch, G. F. 2020, Kalman filter, in 'Computer Vision: A Reference Guide', Springer, pp. 1–3.
- Willmott, C. J. and Matsuura, K. 2005, 'Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance', *Climate research* **30**(1), 79–82.
- Wroth, C. 1984, The interpretation of in situ soil tests, in 'Geotechnique', Vol. 34, Thomas Telford Ltd, pp. 449–489.
- Yu, B. 2022, 'Machine learning for prediction of undrained shear strength from cone penetration test data', *MSc thesis, Delft University of Technology* .

Appendix A

Example text file

```
File triaxial test FGS B.V. Version 3.15 ASCII
Project Number      ,1016-0784-000
BoreHole Number    ,B1
Sample Number      ,St5
Stage Number       , 1
Test type          ,5) CUMS Meertraps isotroop
Sample type        ,0) Ongeroerd
Test Date          ,05-dec-2016 11:41:09
Lab technician     ,Censored
Classification Code ,KLEI, sterk siltig, sterk humeus grijs
Sample Class       , 1
Test depth (m)     , 4.6
Cel Number         , 13
Eps.f              , 5
T100               , 49.47
Regressie A B      , -7.844209E-02, .1023954
Max Strain r.(mm/min), 0
Act Strain r.(mm/min), .049
Drainage conditions ,4) De omtrek plus 1 eind van het proefstuk
Eindplaten         ,2) Ruw
Membrane Em(kPa)   , 1650
Membrane Tm(mm)    , .3
Filter paper Kfp    , .04
Filter paper Pfp    , .5
Particle dens.     ,0) Geschat
Particle dens. (t/m3), 2.65
Failure shape       ,1) Opgestuikt
Back pressure(kPa) , 300
Piston 100%        (mm), 133
Height of dummy    (mm), 100
Piston initial     (mm), 133
Piston after sat.(mm), 133
Piston a cons.    (mm), 129.6
Specimen diam.    (mm), 50.15
Vol.change a sat.(mm), 0
Vol.change a cons(mm), 4.5
Vol.change at end(mm), 18.7
B-factor init      , .9
B-factor after sat. , .96
Mass initial(g)    , 307.22
Mass final(g)      , 290.84
Mass dry(g)        , 181.39
Remark             ,
Remark             ,
Remark             ,
Report as 1 Stage  ,NO
Use 15% value      , 0
Shear stress TV    , 0
Shear stress PP    , 0
number of channels ,0 7
Amplification *n   1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
Calibration *n     1.4937E-03 2.2535E-01 7.5378E-03 -1.5446E-03 6.5085E-02 1.0000E+00 1.0000E+00
Datum tijd         Def.      Force      Pore pr     Volume     Cell pr     Temp      Power
[s]                A/D       A/D       A/D         A/D        A/D        A/D       A/D
-99,ZERO
2016-12-05 12:55:36, 2048.604, .000, .000, .000, .000, .000, .000
2016-12-05 12:55:36, .000, .000, .000, .000, 460.936, .000, .000
-99,START
2016-12-05 11:41:09, 2048.604, -74.773, 39429.277, 11504.597, 5181.378, .000,9984617.000
-99,START CV
2016-12-05 11:41:09, .000, .000, .000, .000, .000, .000, .000
2016-12-05 11:41:16, .000, .000, .000, -64.742, .000, .000, .000
2016-12-05 11:41:23, .000, .000, .000, -129.483, .000, .000, .000
2016-12-05 11:41:29, .000, .000, .000, -161.854, .000, .000, .000
2016-12-05 11:41:37, .000, .000, .000, -226.596, .000, .000, .000
.....
```

Figure A.1: Example text file

Appendix B

Database summary

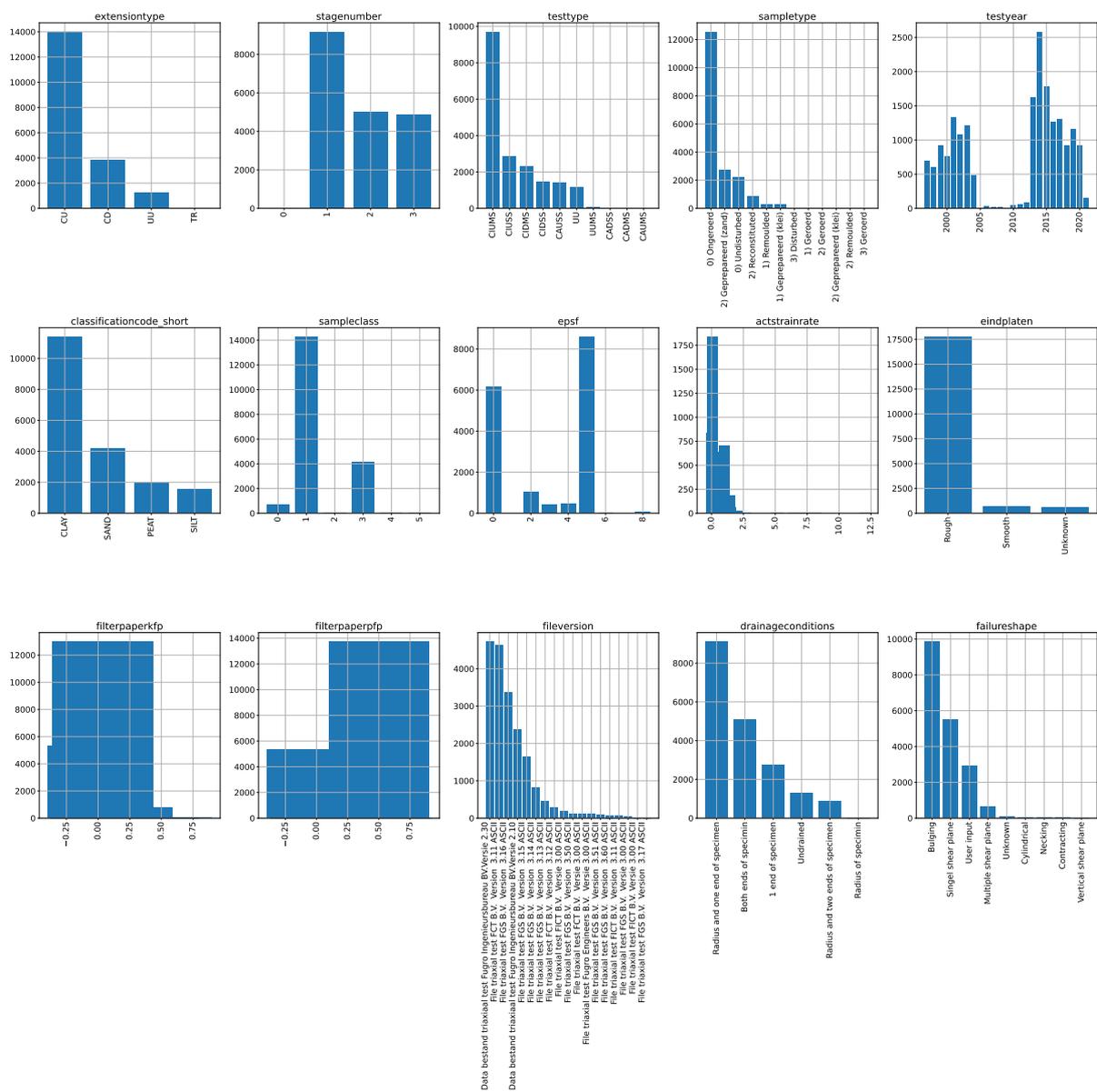


Figure B.1: Bar charts

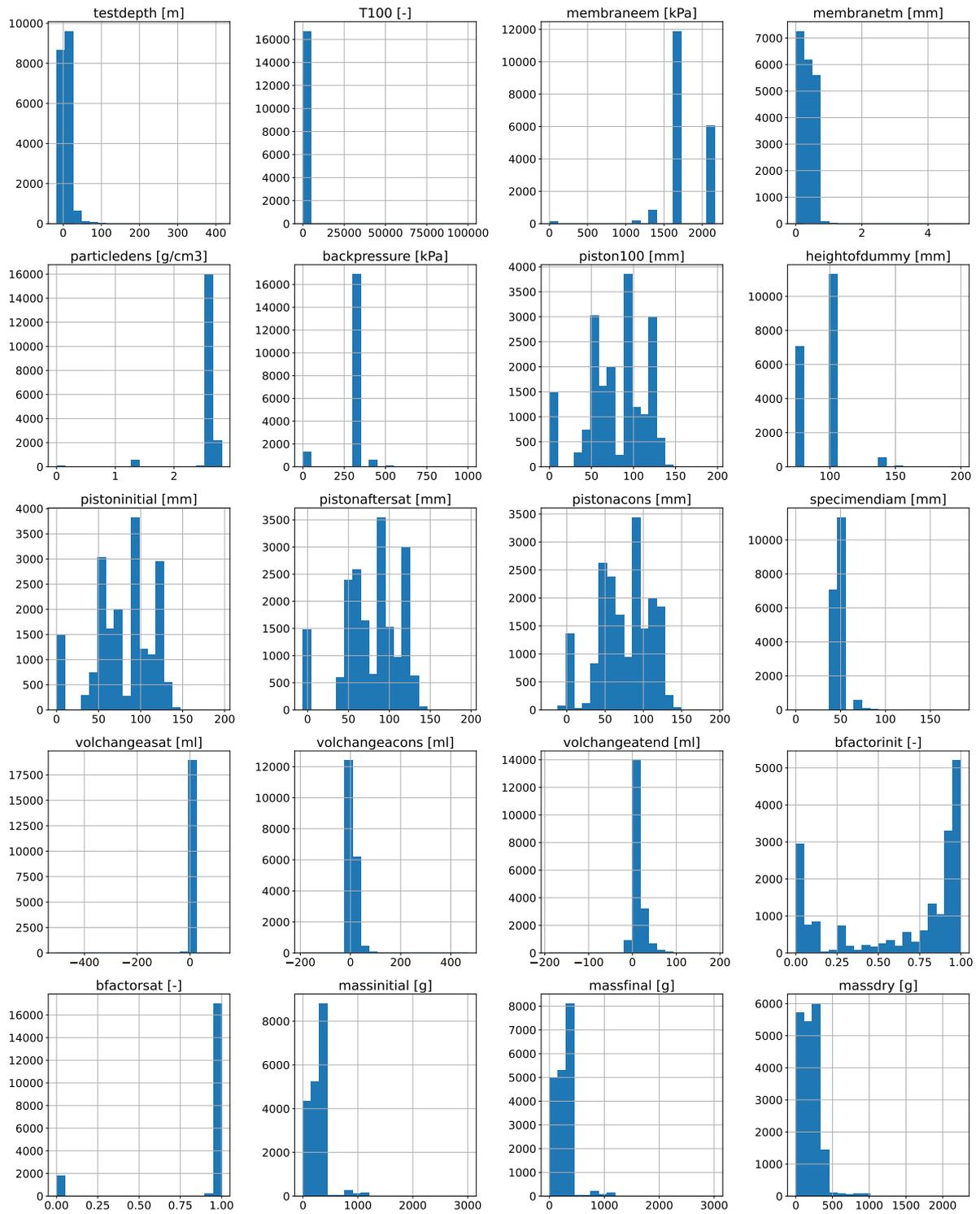


Figure B.2: Histograms

Appendix C

Visualisation of triaxial results

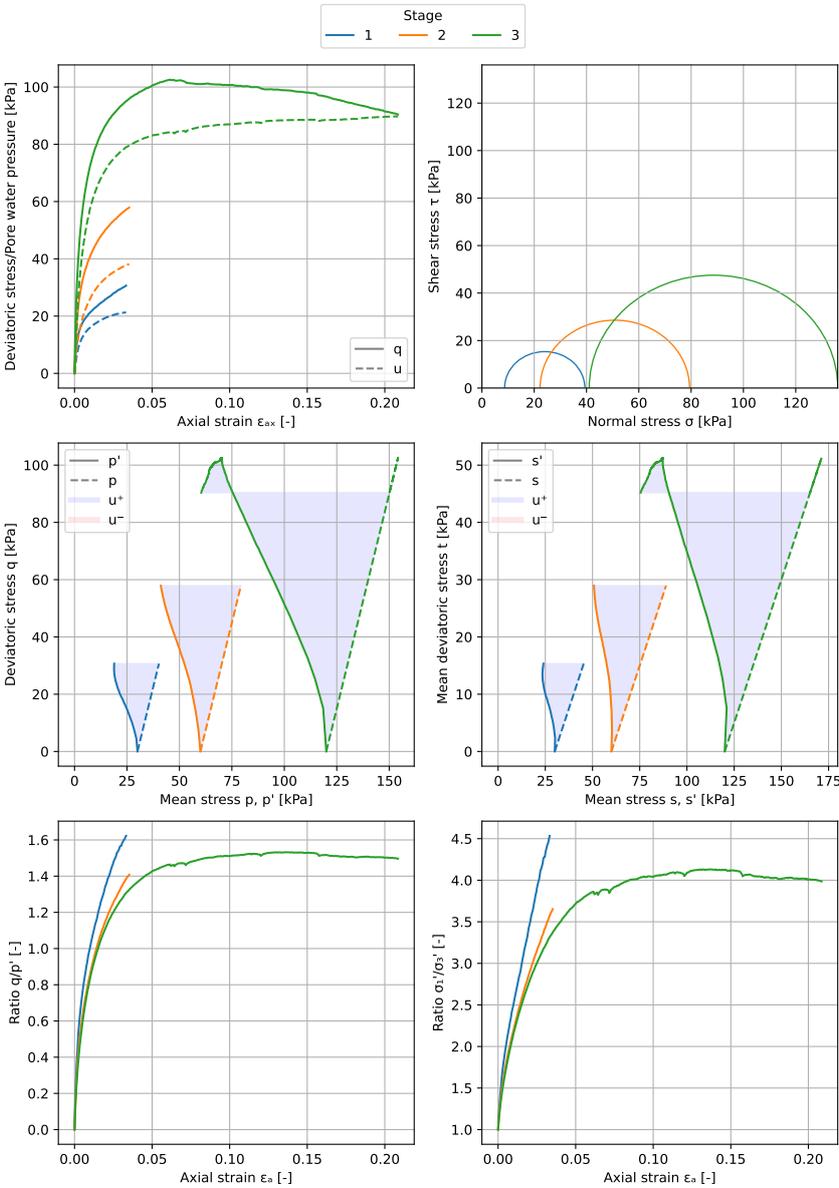


Figure C.1: Example of how to present triaxial results

Appendix D

Sensitivity analysis

D.1 Drained

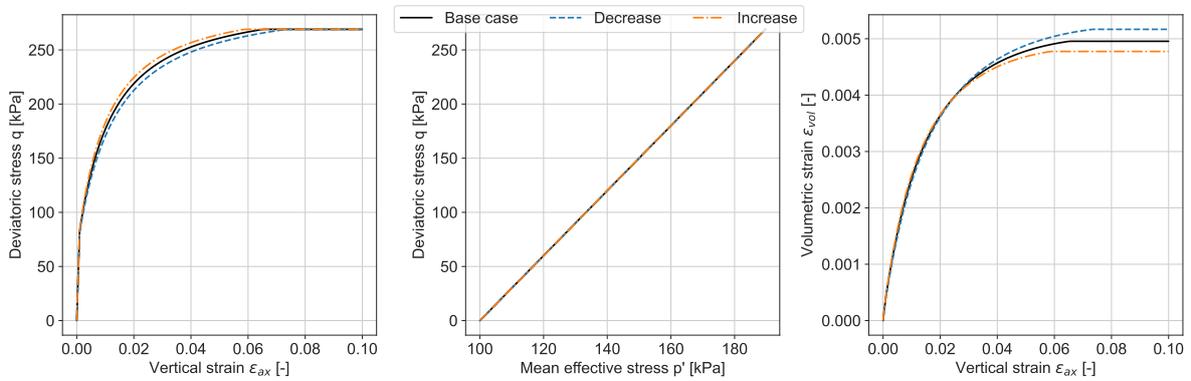


Figure D.1: E_{50}^{ref}

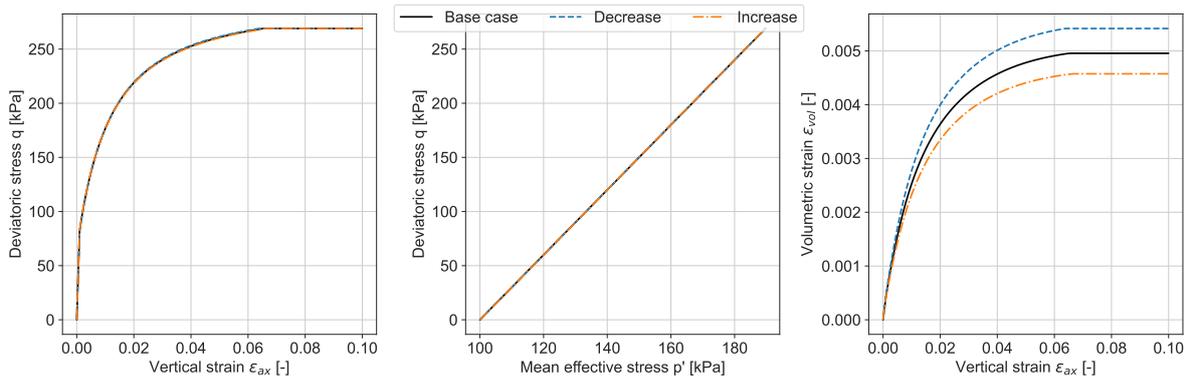


Figure D.2: E_{oed}^{ref}

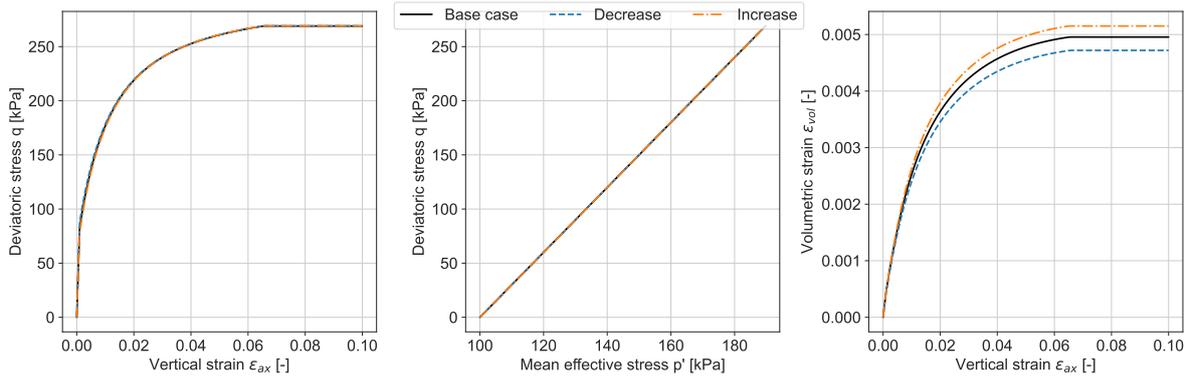


Figure D.3: E_{ur}^{ref}

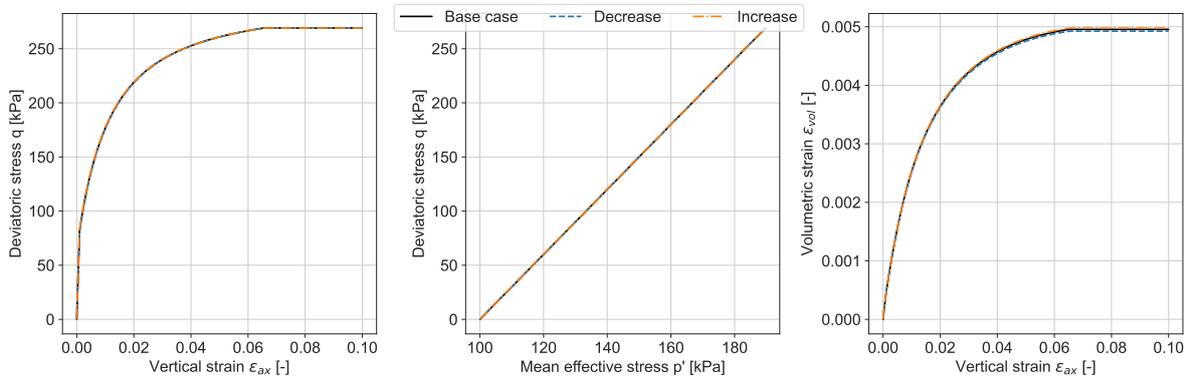


Figure D.4: ν_{ur}

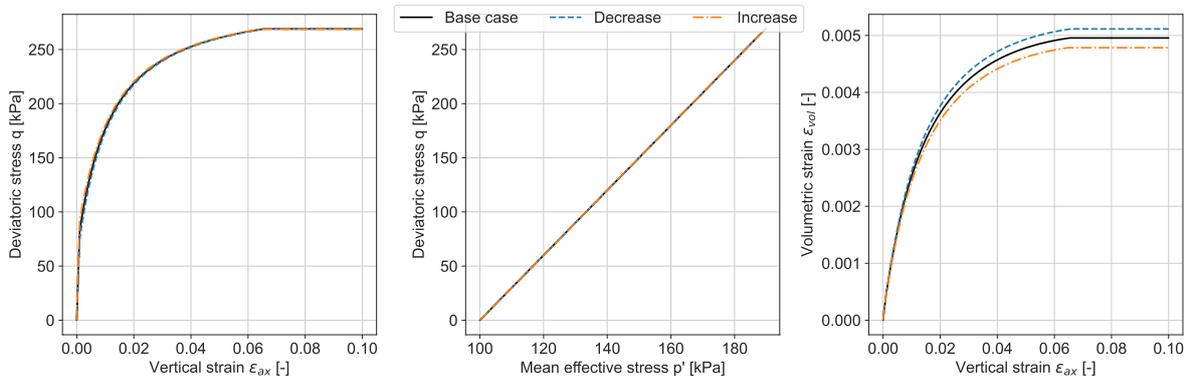


Figure D.5: G_0^{ref}

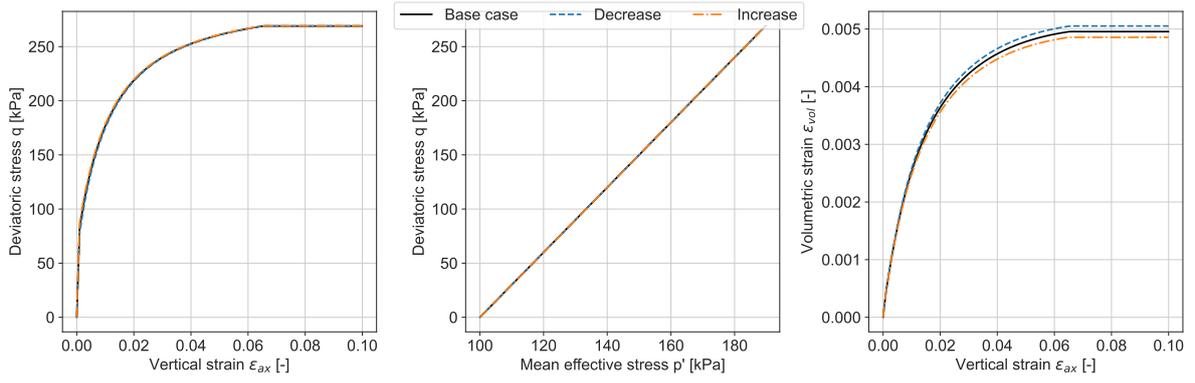


Figure D.6: $\gamma_{0.7}$

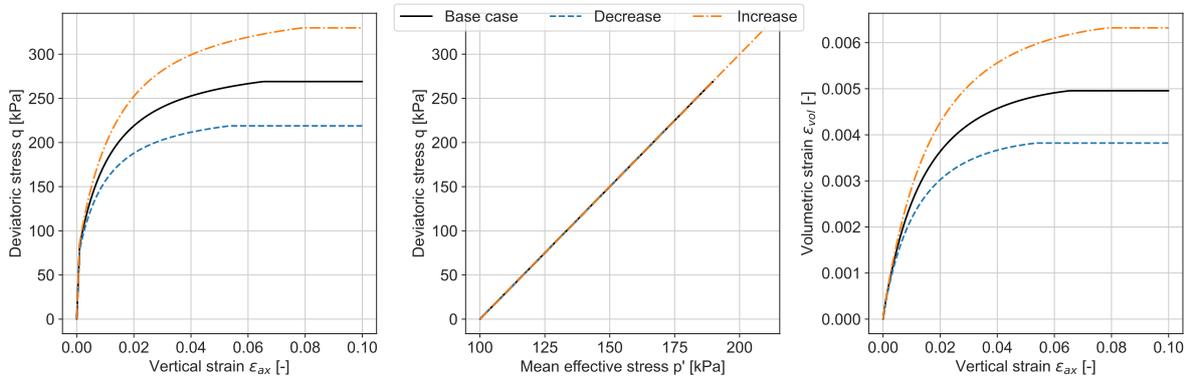


Figure D.7: ϕ'

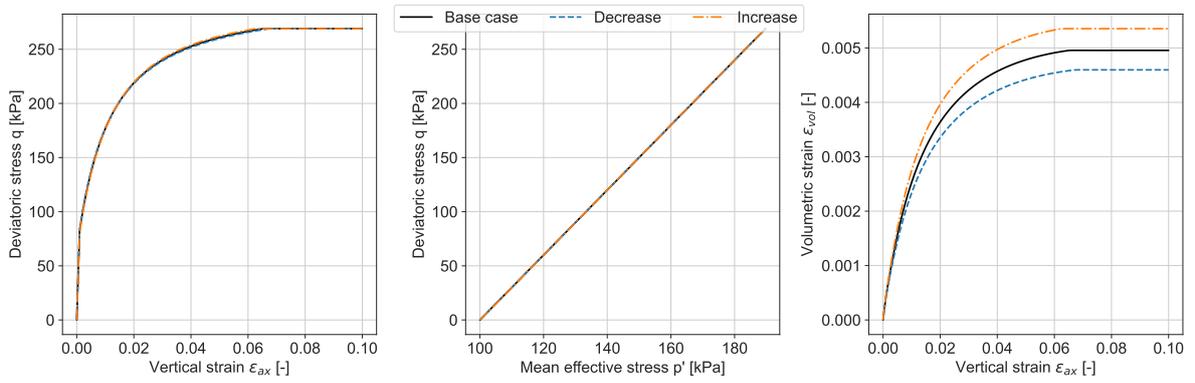


Figure D.8: K_0^{nc}

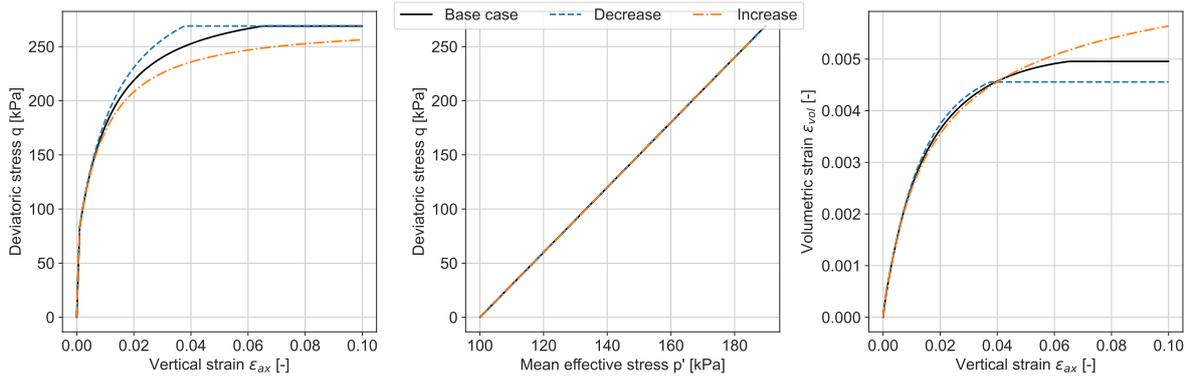


Figure D.9: R_f

D.2 Undrained

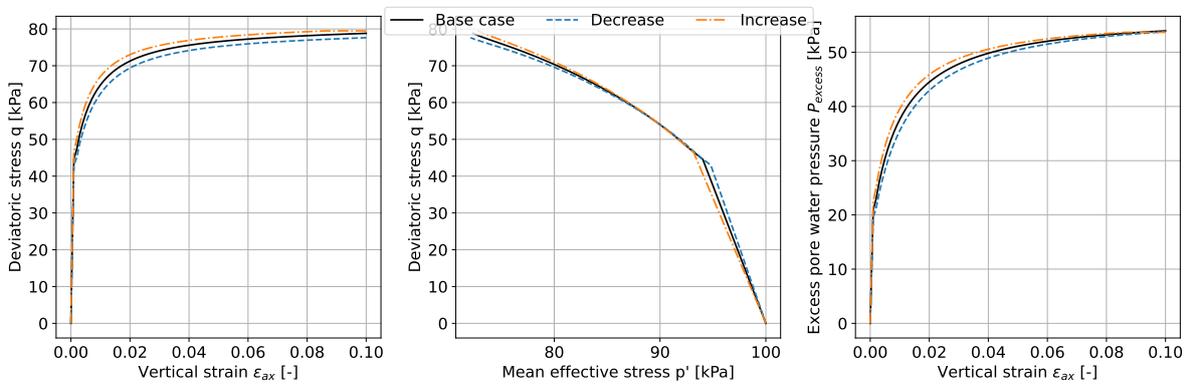


Figure D.10: E_{50}^{ref}

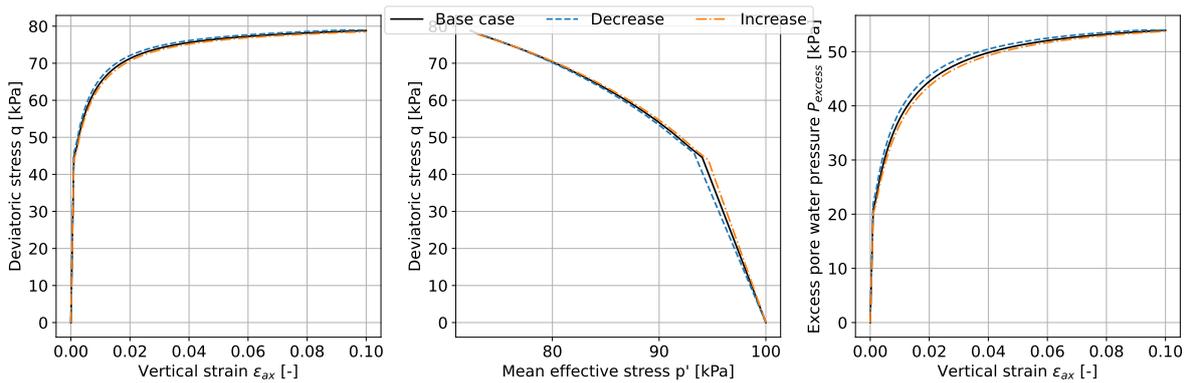


Figure D.11: E_{oed}^{ref}

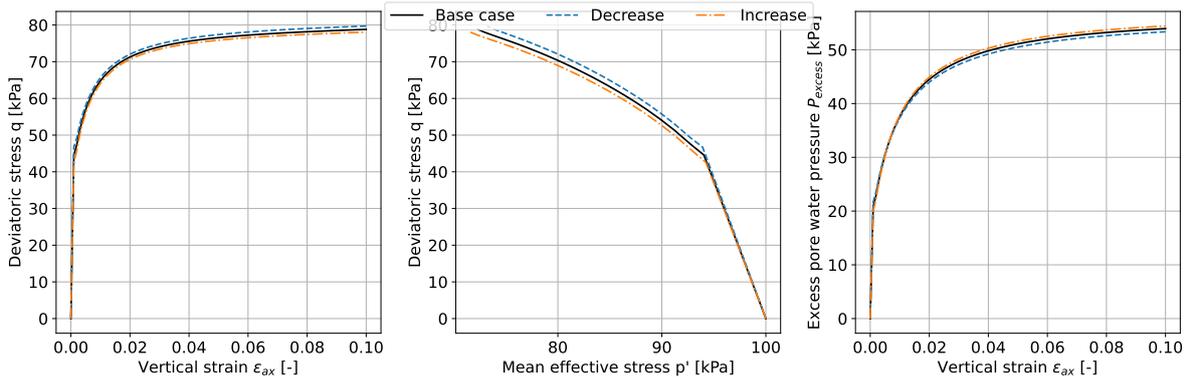


Figure D.12: E_{ur}^{ref}

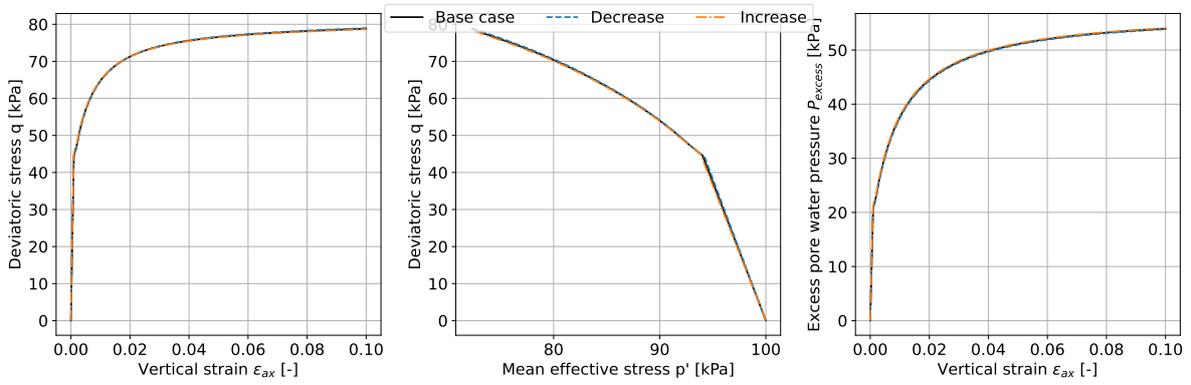


Figure D.13: ν_{ur}

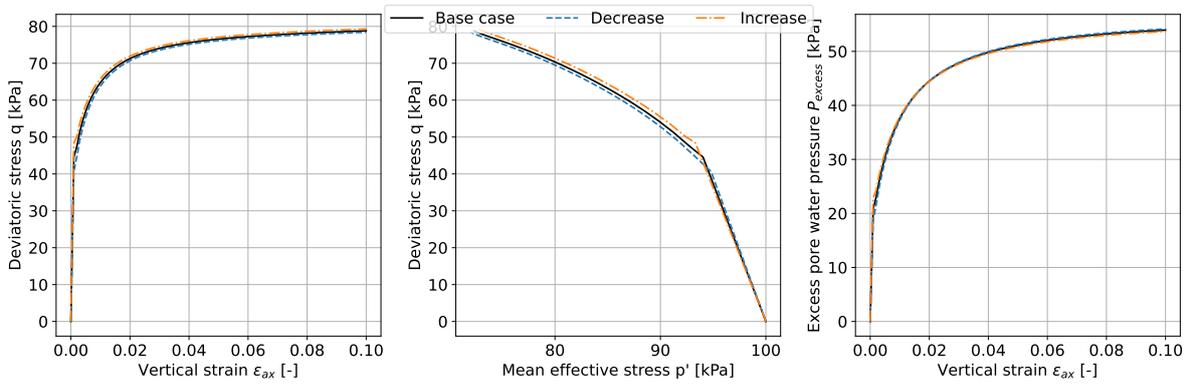


Figure D.14: G_0^{ref}

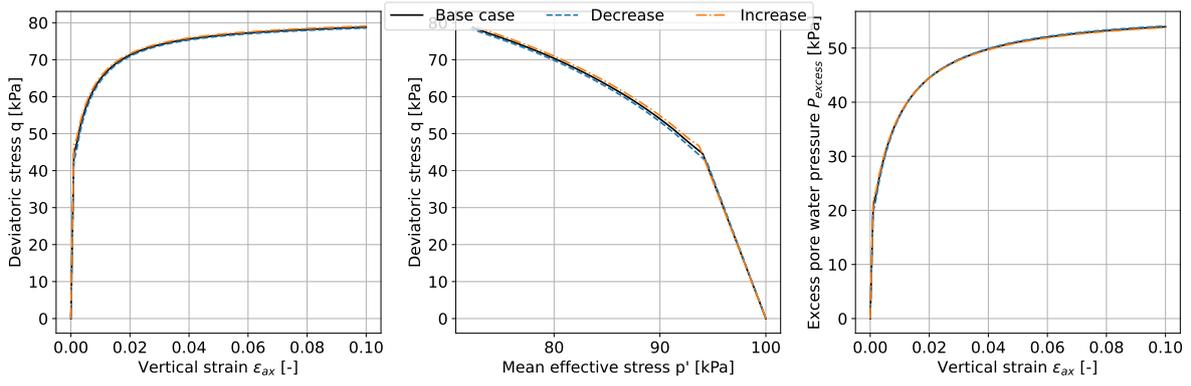


Figure D.15: $\gamma_{0.7}$

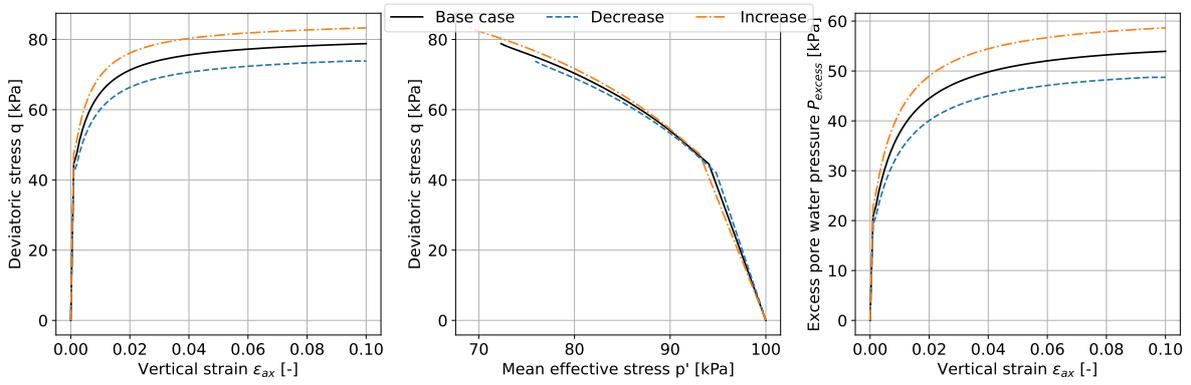


Figure D.16: ϕ'

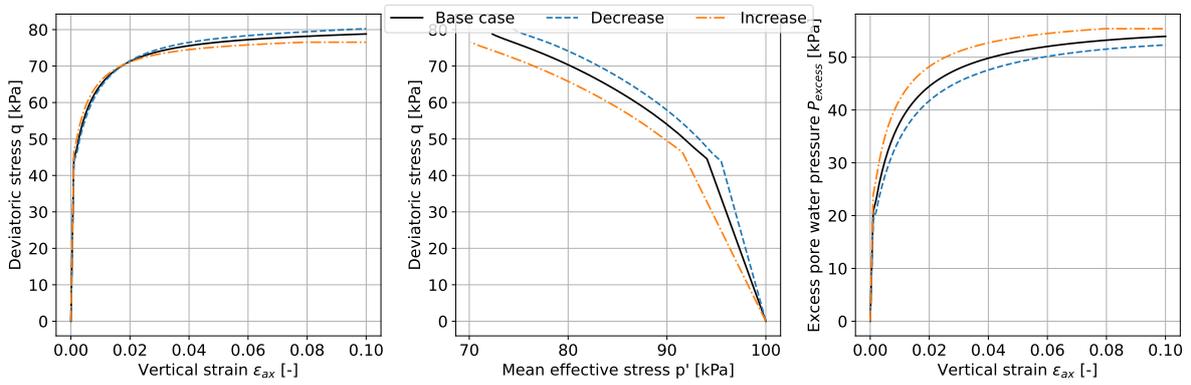


Figure D.17: K_0^{nc}

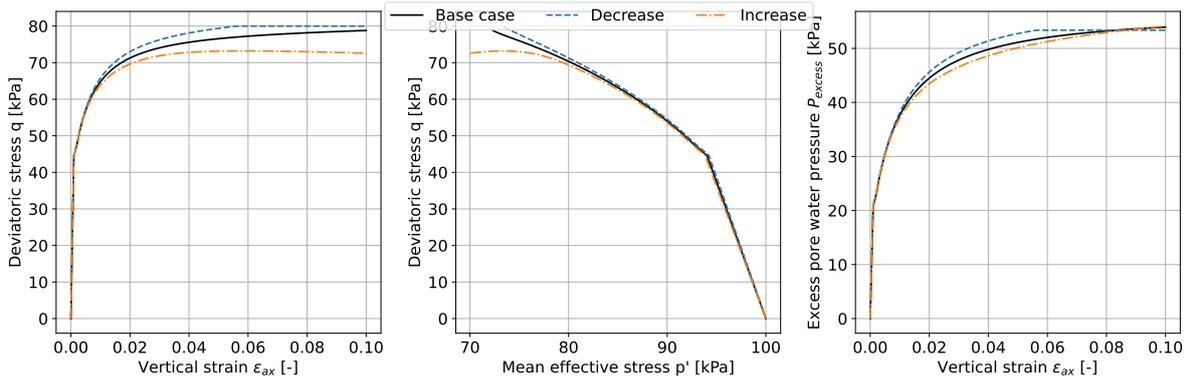


Figure D.18: R_f

Appendix E

Code snippets

```
1 pipe = Pipeline([
2     ("model", RandomForestRegressor(random_state=rs))
3 ])
4
5 paramgrid=[{'model__n_estimators': [100, 500],
6             'model__min_samples_split': [1, 2, 5],
7             'model__min_samples_leaf': [1, 3],
8             }]
```

Listing E.1: Random Forest Regressor

```
1 pipe = Pipeline([
2     ("model", GradientBoostingRegressor(random_state=rs))
3 ])
4
5 paramgrid=[{'model__n_estimators': [100, 500],
6             'model__min_samples_split': [1, 2, 5],
7             'model__min_samples_leaf': [1, 3],
8             }]
```

Listing E.2: Gradient Boosting Regressor

```
1 pipe = Pipeline([
2     ("model", XGBRegressor(random_state=rs))
3 ])
4
5 paramgrid=[{'model__n_estimators': [10, 100],
6             'model__max_depth': [6, 10],
7             'model__num_parallel_tree': [1, 3],
8             'model__booster': ['gbtree', 'gblinear', 'dart']
9             }]
```

Listing E.3: eXtreme Gradient Boosting

```
1 pipe = Pipeline([
2     ("scale", QuantileTransformer(random_state=rs)),
3     ("model", MLPRegressor(random_state=rs, max_iter=2000, activation='relu'))
4 ])
5
6 paramgrid = {'model__hidden_layer_sizes': [(100,), (15,50,15), (16, 16)],
7             'model__alpha': [0.0001, 0.05, 0.01],
8             'model__solver': ['lbfgs', 'adam']}
```

Listing E.4: Artificial Neural Network

```
1 pipe = Pipeline([
2     ("scale", QuantileTransformer(random_state=rs)),
3     ("model", GaussianProcessRegressor(random_state=rs))
4 ])
5
6 paramgrid=[{'model__kernel': [None],
7             'model__alpha': [1e-10, 1e-5]}
```

Listing E.5: Gaussian Process Regressor

```

1 pipe = Pipeline([
2     ("scale", QuantileTransformer(random_state=rs)),
3     ("model", SVR())
4 ])
5
6 paramgrid=[{'model__kernel': ['rbf', 'poly'],
7             'model__degree': [3,5],
8             'model__gamma':['scale', 'auto'],
9             'model__epsilon':[0.1, 0.5]}]

```

Listing E.6: Support Vector Regressor

```

1 pipe = Pipeline([
2     ("scale", QuantileTransformer(random_state=rs)),
3     ("model", Ridge(random_state=rs))
4 ])
5
6 paramgrid=[{'model__alpha': [0.5, 1, 2],
7             }]

```

Listing E.7: Regular Ridge Regressor

```

1 pipe = Pipeline([
2     ("scale", QuantileTransformer(random_state=rs)),
3     ("model", KernelRidge())
4 ])
5
6 paramgrid=[{'model__alpha': [0.5, 1, 2],
7             'model__kernel': ['poly'],
8             'model__degree': [3,4,5,6],
9             'model__coef0': [0,1,2]
10            }]

```

Listing E.8: Kernel Ridge Regressor

Appendix F

The five-number summary

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	16.4	2.2	10.4	14.9	16.4	17.9	22.7
Sand	718	17.6	1.6	12.4	16.6	17.5	18.7	22.4
Peat	374	11.0	1.6	8.6	10.0	10.6	11.6	18.8
Silt	274	19.6	1.7	12.9	18.9	19.9	20.8	22.4

Table F.1: Soil property: γ

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.6	0.4	0.1	0.4	0.5	0.8	4.8
Sand	718	0.1	0.1	0.0	0.1	0.1	0.1	1.1
Peat	374	3.8	2.1	0.1	2.1	3.7	5.1	9.9
Silt	274	0.3	0.2	0.0	0.2	0.2	0.3	1.3

Table F.2: Soil property: w

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	10.8	3.3	1.8	8.4	10.9	13.3	20.2
Sand	718	15.5	1.7	6.4	14.7	15.7	16.6	19.8
Peat	374	3.1	2.3	0.9	1.6	2.2	3.7	16.9
Silt	274	15.9	2.5	5.6	15.0	16.3	17.6	20.1

Table F.3: Soil property: γ_d

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	1.7	1.1	0.2	0.9	1.3	2.1	12.8
Sand	718	0.4	0.3	0.0	0.3	0.3	0.3	2.8
Peat	374	5.3	2.9	0.1	3.0	5.2	7.2	13.9
Silt	274	0.7	0.4	0.0	0.5	0.6	0.7	3.6

Table F.4: Soil property: e_0

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.6	0.1	0.1	0.5	0.6	0.7	0.9
Sand	718	0.3	0.1	0.0	0.2	0.2	0.2	0.7
Peat	374	0.8	0.1	0.1	0.7	0.8	0.9	0.9
Silt	274	0.4	0.1	0.0	0.3	0.4	0.4	0.8

Table F.5: Soil property: n_0

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	26.0	6.4	10.3	21.0	26.3	30.9	47.8
Sand	718	33.9	5.1	15.2	31.3	34.3	37.3	46.6
Peat	374	17.2	5.0	10.0	13.8	16.3	19.9	35.6
Silt	274	31.5	4.4	13.3	29.0	31.7	34.3	43.4

Table F.6: Soil parameter: ϕ'

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	6.2	5.7	0.0	1.9	5.0	8.9	29.9
Sand	718	4.7	6.3	0.0	0.0	2.5	6.9	29.8
Peat	374	8.6	7.5	0.0	3.1	5.8	12.4	29.7
Silt	274	5.7	6.4	0.0	0.0	3.8	8.6	29.3

Table F.7: Soil parameter: c'

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	6519	4526	100	4020	5597	7738	59689
Sand	718	44531	24349	340	24805	43396	66273	80000
Peat	374	2570	5324	531	1434	1955	2668	80000
Silt	274	9880	5077	432	6477	8857	12306	34397

Table F.8: Soil parameter: E_{50}^{ref} not optimised

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	-1.96	3.42	-23.11	-3.22	-0.51	0.47	1.0
Sand	718	-0.84	2.97	-19.81	-1.12	0.36	0.75	0.98
Peat	374	-0.09	1.85	-16.71	-0.24	0.55	0.85	0.99
Silt	274	-5.54	4.53	-22.45	-8.08	-5.39	-1.92	0.95

Table F.9: Initial r^2

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.52	1.3	-13.99	0.78	0.92	0.97	1.0
Sand	718	0.16	2.01	-16.37	0.45	0.71	0.88	1.0
Peat	374	0.94	0.23	-1.75	0.97	0.99	0.99	1.0
Silt	274	-1.23	2.76	-11.66	-2.88	0.09	0.9	0.99

Table F.10: Optimised r^2

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	7595	5685	114	4231	6396	9193	72796
Sand	718	52267	34743	252	22044	45130	79972	120000
Peat	374	2741	6007	531	1463	2018	2759	96000
Silt	274	13685	7565	540	8320	12105	17267	42996

Table F.11: Optimised model parameter: E_{50}^{ref}

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	5993	4440	52	3235	5088	7478	59689
Sand	718	29511	20269	135	13632	25449	41300	126201
Peat	374	1759	3110	398	1002	1292	1780	56000
Silt	274	10224	5549	318	6332	9159	12923	32042

Table F.12: Optimised model parameter: E_{oed}^{ref}

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	21824	14177	312	14099	19120	25667	188020
Sand	718	139342	77137	1142	74643	137065	209562	296000
Peat	374	8994	16437	2151	5281	7106	9496	252000
Silt	274	30063	15584	1296	19856	26883	36918	107490

Table F.13: Optimised model parameter: E_{ur}^{ref}

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.21	0.02	0.16	0.2	0.2	0.22	0.24
Sand	718	0.19	0.02	0.16	0.16	0.19	0.2	0.24
Peat	374	0.2	0.02	0.16	0.19	0.2	0.21	0.24
Silt	274	0.21	0.02	0.16	0.2	0.2	0.21	0.24

Table F.14: Optimised model parameter: ν_{ur}

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.7	0.2	0.5	0.5	0.7	0.9	1.0
Sand	718	0.8	0.2	0.5	0.6	0.8	1.0	1.0
Peat	374	0.6	0.1	0.5	0.5	0.6	0.7	1.0
Silt	274	0.8	0.2	0.5	0.5	0.8	1.0	1.0

Table F.15: Optimised model parameter: m

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	47257	31860	838	28791	40977	56709	384993
Sand	718	311676	166983	2789	180593	306046	450249	684000
Peat	374	19100	36780	4036	10956	14885	20086	516000
Silt	274	78276	42487	3239	47871	68011	95297	240778

Table F.16: Optimised model parameter: G_0^{ref}

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.00014	3e-05	0.00011	0.00011	0.00011	0.00015	0.0002
Sand	718	0.00013	3e-05	0.00011	0.00011	0.00011	0.00014	0.0002
Peat	374	0.00016	3e-05	0.00011	0.00014	0.00015	0.00018	0.0002
Silt	274	0.00016	4e-05	0.00011	0.00011	0.00017	0.0002	0.0002

Table F.17: Optimised model parameter: $\gamma_{0.7}$

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	2.6	3.2	0.0	0.0	1.5	3.5	10.0
Sand	718	1.6	2.3	0.0	0.0	0.0	3.5	10.0
Peat	374	2.6	3.2	0.0	0.0	1.0	3.5	10.0
Silt	274	5.1	4.1	0.0	0.5	3.5	10.0	10.0

Table F.18: Optimised model parameter: $*c'$

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	36.4	8.8	15.3	29.4	37.5	42.9	50.0
Sand	718	33.9	5.8	15.1	31.8	34.3	36.4	50.0
Peat	374	39.9	7.5	19.9	33.8	40.0	47.0	50.0
Silt	274	39.9	6.8	19.3	37.1	39.5	44.4	50.0

Table F.19: Optimised model parameter: $*\phi'$

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.0	0.2	-0.7	0.0	0.0	0.0	1.2
Sand	718	0.4	0.6	-1.2	-0.1	0.0	1.1	1.2
Peat	374	0.1	0.3	-0.7	0.0	0.0	0.1	1.2
Silt	274	0.0	0.3	-0.7	-0.1	-0.1	0.0	0.7

Table F.20: Optimised model parameter: ψ

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.41	0.12	0.23	0.32	0.39	0.51	0.74
Sand	718	0.44	0.08	0.23	0.41	0.44	0.47	0.74
Peat	374	0.36	0.1	0.23	0.27	0.36	0.44	0.66
Silt	274	0.36	0.09	0.23	0.3	0.36	0.4	0.67

Table F.21: Optimised model parameter: k_0^{nc}

Soil	Size	μ	σ	Min.	Lower quartile	Median	Upper quartile	Max.
Clay	1707	0.6	0.3	0.1	0.2	0.7	0.9	1.0
Sand	718	0.8	0.2	0.1	0.7	0.8	0.9	1.0
Peat	374	0.6	0.2	0.1	0.5	0.6	0.8	1.0
Silt	274	0.3	0.3	0.1	0.1	0.1	0.3	1.0

Table F.22: Optimised model parameter: R_f

Appendix G

Additional violin plots

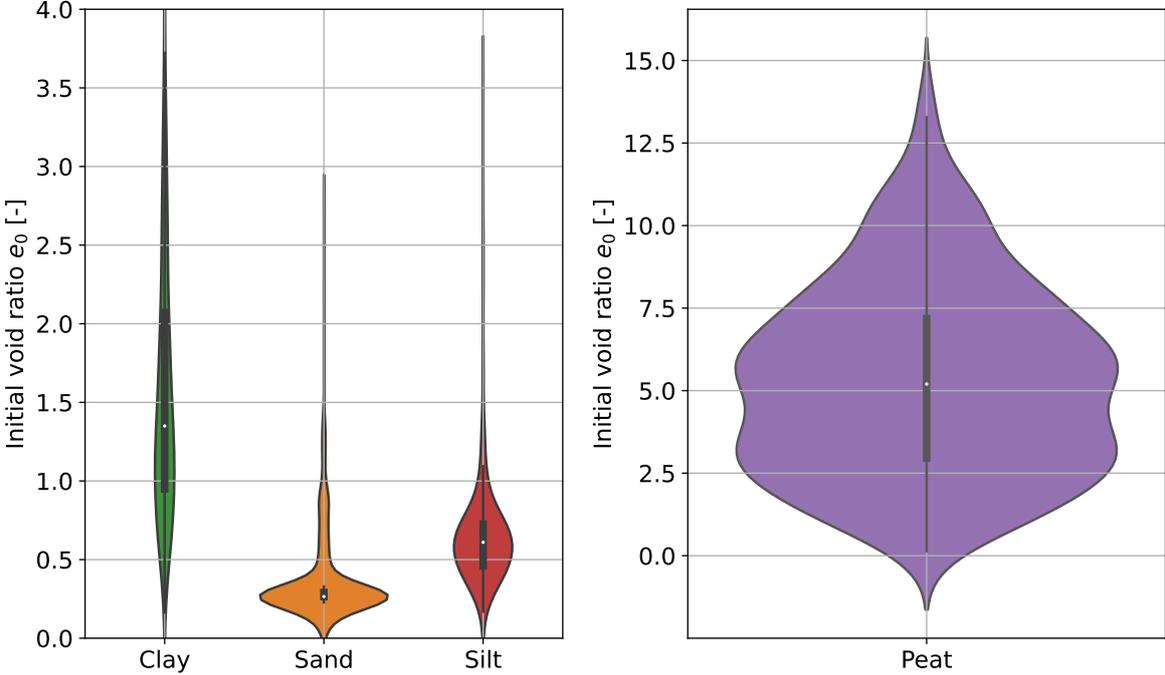


Figure G.1: Initial void ratio e_0

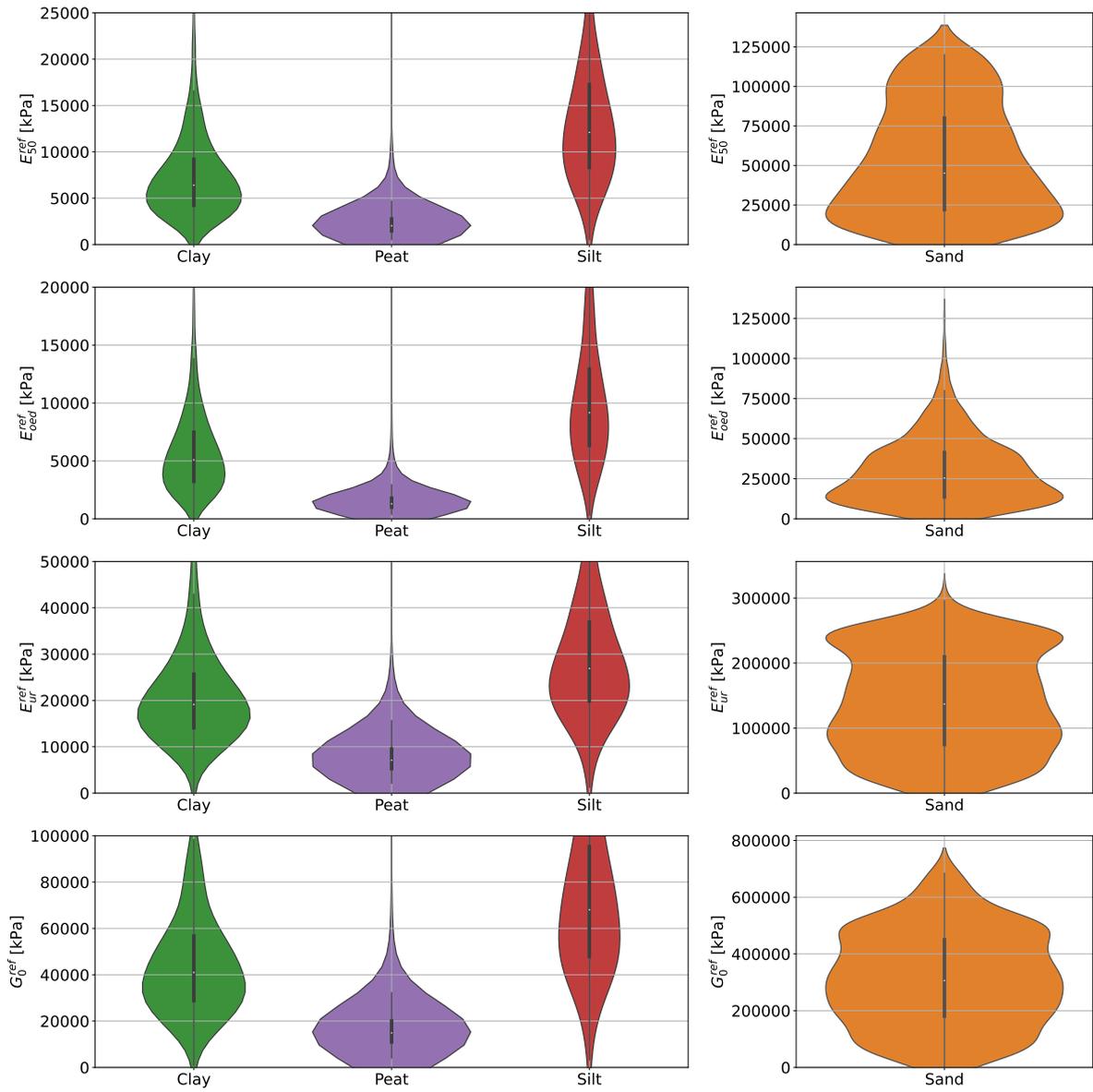


Figure G.2: Optimised model parameters: E_{50}^{ref} , E_{oed}^{ref} , E_{ur}^{ref} and G_0^{ref}

Appendix H

CPT

1211-0082-001_DKM5

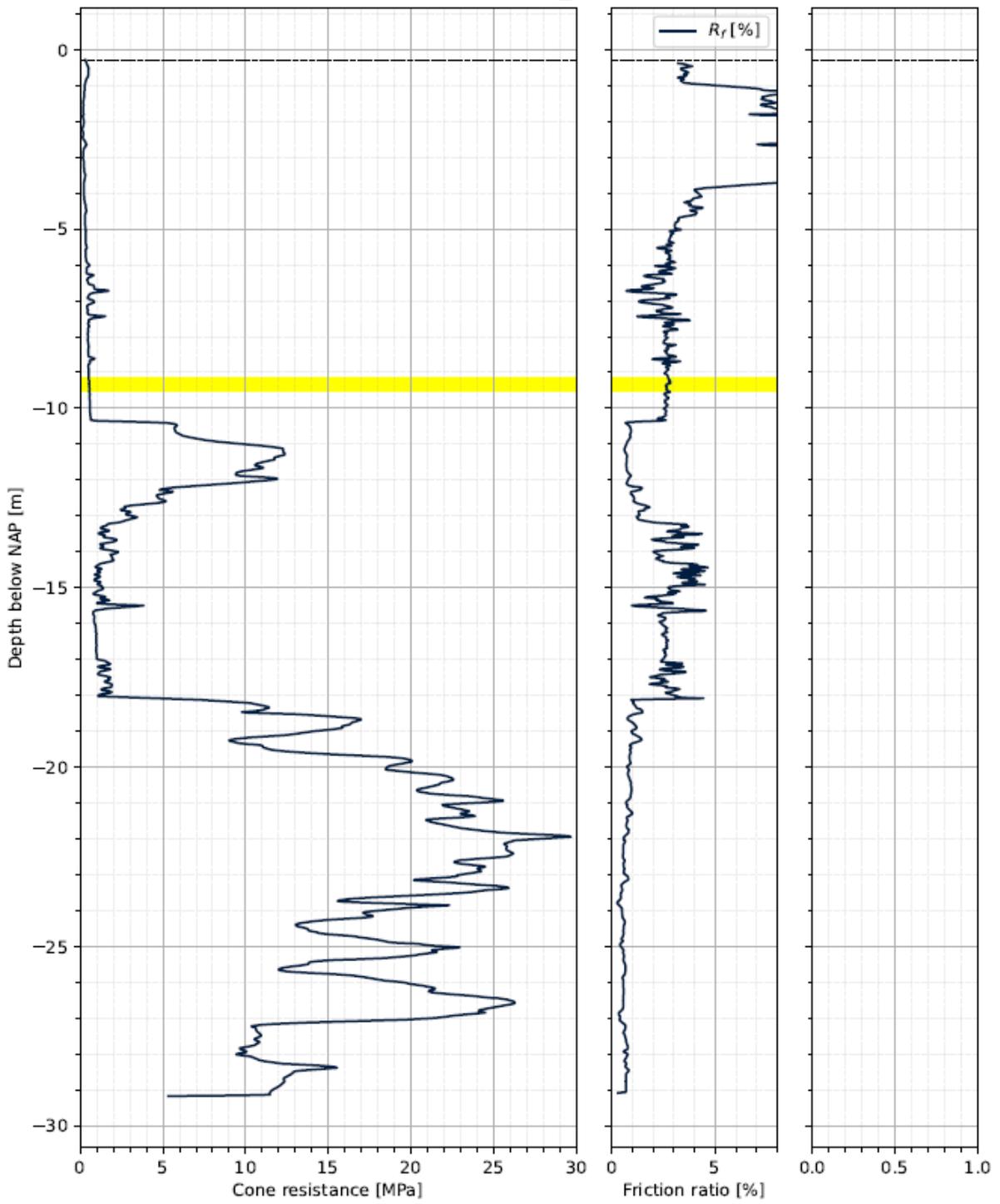


Figure H.1: DKM5

1211-0082-001_DKM32

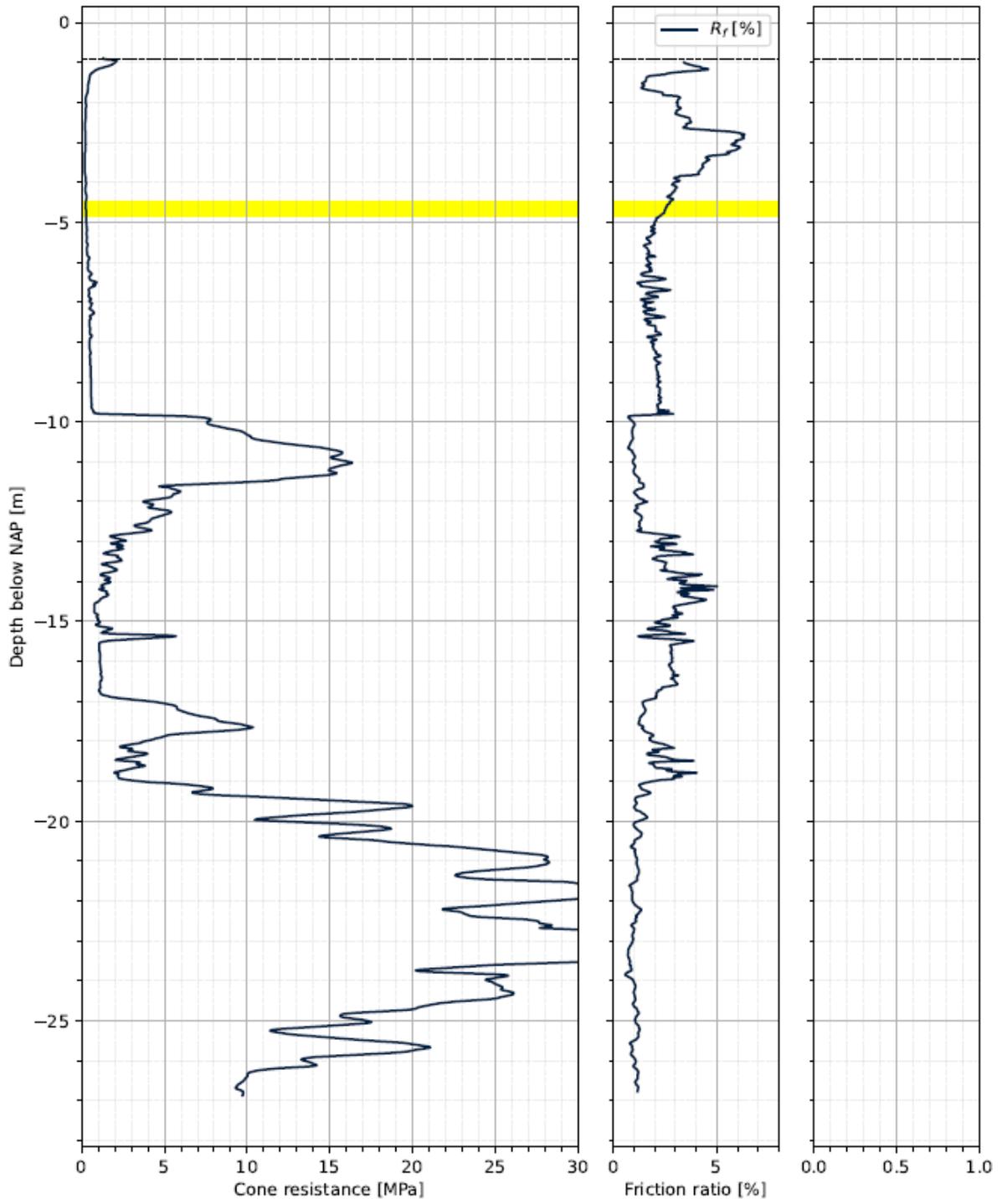


Figure H.2: DKM32