

INTERNATIONAL SOCIETY FOR SOIL MECHANICS AND GEOTECHNICAL ENGINEERING



This paper was downloaded from the Online Library of the International Society for Soil Mechanics and Geotechnical Engineering (ISSMGE). The library is available here:

<https://www.issmge.org/publications/online-library>

This is an open-access database that archives thousands of papers published under the Auspices of the ISSMGE and maintained by the Innovation and Development Committee of ISSMGE.

Detecting highly sensitive materials with CPTu in Norway using machine learning

S.M. Valsson, S.A. Degago, E. Haugen

Norwegian Public Roads Administration

Abstract

The Norwegian Public Roads Administration has, since early 2017, systematically studied CPTu data with laboratory results from road-projects across Norway. This effort has produced a set of labelled data from over 200 CPTu tests in varied field conditions, and a portion of this set contains data that can be used to study how existing models fare in detecting highly sensitive materials. This paper describes how techniques from the field of machine learning are used to create classifiers that greatly outperform today's classification diagrams in detecting highly sensitive materials in Norway. The results show that tuned classifiers fitted with selected features on samples from local sites can generate high detection rates for highly sensitive materials while at the same time keeping rates of false detections low.

1 Introduction

In slopes with highly sensitive materials, a small initial slide can potentially develop into a very large landslide. When such slides occur, they are fast and extensive with a potential to cause damage to nearby people and adjacent infrastructure. There has been several large landslides involving highly sensitive soils in Norway (e.g. Thakur et al. 2013).

A primary and crucial task in investigating the threat posed by such potential slides is to map the extent of the highly sensitive materials. Soil sampling and laboratory testing is the main method of detecting highly sensitive materials; however, methods based on interpretation of field investigation data have also been developed and used to detect such materials (e.g. NIFS 2015). Interpretation based on field investigation is interesting as it is more effective as compared to laboratory sampling and testing. However, the accuracy of detection is a challenge that needs to be improved. As an attempt to achieve this, this work resorts to another field of study, named machine learning, in order to better analyze and interpret field data.

Machine learning is a field of computer science, where computer programs are trained to make predictions based on data, without being explicitly programmed to do so. Utilizing methods from statistics and linear algebra, these types of programs

can sift through very large datasets, and find correlations that might be next to impossible for humans to derive.

The goal of this study is to see if machine learning classifiers can be used to distinguish highly sensitive materials from non-sensitive based on CPTu results.

2 Gathering data

In 2017, the Norwegian Public Roads Administration launched an internally developed spreadsheet for parameter interpretation based on CPTu data. The aim was to facilitate the interpretation of engineering parameters based on current practices and to be used in various projects across Norway.

The spreadsheet encourages users to share their data to a central register with a single click on a button, resulting in references to over 200 CPTu tests from over 50 different projects from across Norway being saved. The register can be used to collect the data into a large table, linking laboratory- and CPTu data by position and dept. For each depth with lab data, 27 CPTu features (or variables) are collected.

For this study, three subsets were filtered from the table. They are shown along with their defining criteria Tab. 1. Note that a lower bound is added to the definition of highly sensitive materials to avoid points belonging to two groups.

Tab. 1: Subsets that are selected for this study.

Subset	Class number	Criteria	Data points	Σ
Quick clays	0	$c_{ur} < 0.5 \text{ kPa}$	288	1568
Highly sensitive material	1	$0.5 \leq c_{ur} \leq 2.0 \text{ kPa}$ and $S_t > 15$	234	
Not sensitive material	2	$c_{ur} > 2.0 \text{ kPa}$	1046	

3 How are classes assigned to points

Classes are assigned to points by the use of classifiers. These are computer programs that apply algorithms to a set of training data with known outcomes in order to predict the outcome for new data. In practice the classifier generates decision boundaries where points on one side of a boundary get assigned one class, and points on the other get assigned another.

Classifiers often have tuning parameters to control their behavior, and the goal of tuning is to find the parameters that make the classifier behave in such a way, that it best generalizes to data that is not in the training set (out of sample data).

This work relies on a machine learning framework for Python called Scikit-learn, which provides a large assembly of machine learning tools. For more information see Pedregosa et al. (2011).

4 Adding new features

The original feature space contains 27 features, these include depth, cone resistance, sleeve frictional resistance, pore pressure, estimated (or measured) in-situ pore pressure, overburden pressure and other commonly known features derived from different combination of these. Before any analysis are done, it is not known if the best features for detecting highly sensitive materials are among these 27.

Machine learning is built on data, and if classifiers are fitted to good data and relevant features they may produce good results. In order to increase the chances of producing good results, the feature space is expanded by adding new features from the existing ones.

By replacing features in the formulas used to define the existing features and by applying log-transformations to selected ones, the total number of features is increased from 27 to 1989.

5 Feature selection

Defining classifiers from a large number of features is not necessarily positive, as this can slow down the training time considerably. In addition, irrelevant or redundant features can cause classifiers to produce poor results.

Two feature selection methods have been implemented, i.e. recursive feature selection (RFS) and Recursive feature elimination (RFE). The number of features used to fine-tune each classifier is reduced to 20. For more information a reference is made to Kotsiantis, Kanellopoulos & Pintelas (2006).

6 Test observations and score metrics

In order to consequently evaluate performances, a score metric needs to be selected. The confusion matrix proposed to describe each predicted result is presented in Tab. 2. Highly sensitive materials and quick clay are treated alike.

Tab. 2: Confusion matrix used to describe predicted results.

		Observed outcome		
		Not sensitive	Highly sensitive	Quick clay
Predicted outcome	Not sensitive	True negative (TN)	False negative (FN)	False negative (FN)
	Highly sensitive	False positive (FP)	True positive (TP)	True positive (TP)
	Quick clay	False positive (FP)	True positive (TP)	True positive (TP)

The true positive rate and false positive rate are defined as from Tab. 2 as

$$TPR = \frac{\sum TP}{\sum TP + \sum FN} \quad (1)$$

$$FPR = \frac{\sum FP}{\sum TN + \sum FP} \quad (2)$$

The classification accuracy and a custom metric score are defined as

$$Accuracy = \frac{\sum TP + \sum TN}{\sum TP + \sum FN + \sum TN + \sum FP} \quad (3)$$

$$custom\ metric = TPR - FPR \quad (4)$$

Both eq. (1) and (2) are defined using points from either sensitive or not sensitive subsets, but eq. (3) and (4) are defined using points from the entire set.

The custom metric introduces a bias if the subsets are of different sizes, as percentages from uneven class sizes are directly related to one another.

6.1 Applying score metrics to existing models

Soil behavior type classification diagrams are widely used to interpret CPTu data and many different diagrams have been proposed over the years. For more details on the models presented here, reference is made to Fellenius et al. (2000), Lunne et al. (1997), NIFS 2015, Schneider et al (2012) and Valsson (2016).

In order to apply the confusion matrix to existing diagrams, a system for classifying their different areas is given in Tab. 3. Points outside model definitions are counted as not sensitive. The results for each model can be seen in the classifier summary in Fig. 8.

Tab. 3: Proposed system to implement the confusion matrix on existing models.

Model	Highly sensitive or quick	Not sensitive
Robertson 1990 (B_q-Q_t)	1. Sensitive, fine grained	Everything else
Robertson 1990 (F_r-Q_t)	1. Sensitive, fine grained	Everything else
Robertson et al. 1986 (B_q-q_t)	1. Sensitive, fine grained	Everything else
Robertson et al. 1986 (R_f-q_t)	1. Sensitive, fine grained	Everything else
Eslami et al. 2000 (f_s-q_e)	1. Sensitive and Collapsible Clay and/or Silt	Everything else
Senneset et al. 1989 (B_q-q_t)	1. Very soft	Everything else
Schneider et al. 2012 ($\Delta u_2/\sigma'_{v0}-Q_t$)	1c. Sensitive Clays	Everything else
Schneider et al. 2012 (F_r-Q_t)	1c/3. Sensitive Clays/Transitional soils	Everything else
NIFS 2015 ($B_{q1}-N_{mc}$)	1. Sannsynlig kvikkleire 2. Mulig sprøbruddmateriale	Everything else
NIFS 2015 ($R_{fu}-N_{mc}$)	1. Mulig sprøbruddmateriale	Everything else
Valsson 2017 ($B_q-f_{sn}-q_{tn}$)	1. Quick clay	Everything else

7 Validation

The goal of the validation process is to provide an estimate on how well the classifier will generalize to out of sample data, i.e. how accurate will a prediction be for data from new projects.

The main idea is thus to split the samples into a training- and testing set. A classifier is fitted to the training set, and then used to predict the outcomes for the test set. The score is calculated by comparing the prediction to the known results. There are many different validation methods, but they differ only in how the points are split into the train- and test sets.

This study implements the LOGOS (Leave One Group Out) method, which defines the training set as all points belonging to a specific group. The groups are defined as the projects in the CPTu register (50), and so with each validation a classifier is tested with data from a site that is not present in the training set. The validation is repeated for all the projects and results are summed up. The reason for this is to avoid testing with data which has specific correlated data in the training set.

8 Overview of tested classifiers

A brief description of each classifier is presented below. The best results were often found when using a large number of features (>10), but a 2D figure showing a (very) simplified decision boundary for each classifier with CPTu data is given to provide an insight on how it works.

A separate feature selection process is implemented for each classifier presented in this section, and the best 2 features will therefore vary from case to case.

The aim was to identify the optimal number and combination of features for each classifier. The custom metric (eq. 4) is used to judge each classifiers ability to separate highly sensitive soils from not sensitive soils.

8.1 Decision Tree Classifier

The Decision Tree classifier (DTC) works by recursively splitting the dataset into groups (or leafs) by a value of one feature at a time. Fig. 1 shows decision boundaries of a DTC that has been trained with two features.

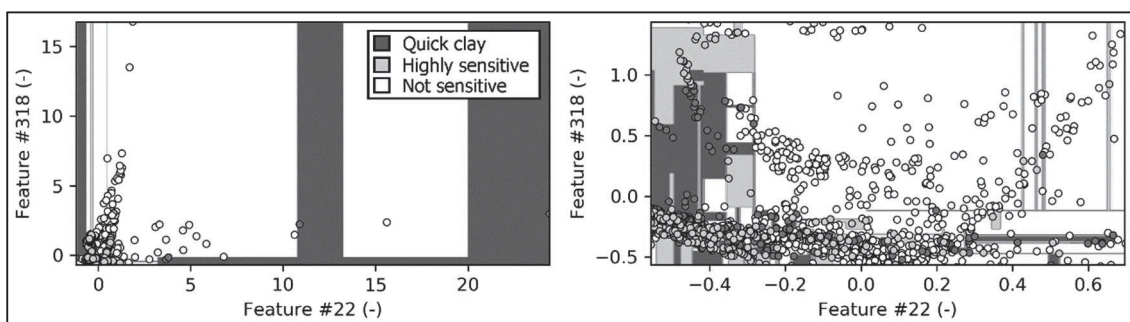


Fig. 1: Decision boundary of a 2D DTC. The figure on the right is a close up, showing about 90% of the points in the figure on the left.

8.2 k Nearest Neighbors Classifier

The k Nearest Neighbors classifier (kNNC) finds the k nearest neighbors of a given point, and assigns it the same class as the one with the highest count.

The kNNC can also produce a reliability estimate by returning the percentage out of the k belonging to the returned class. Fig. 2 shows decision boundaries of a kNNC that has been trained with two features.

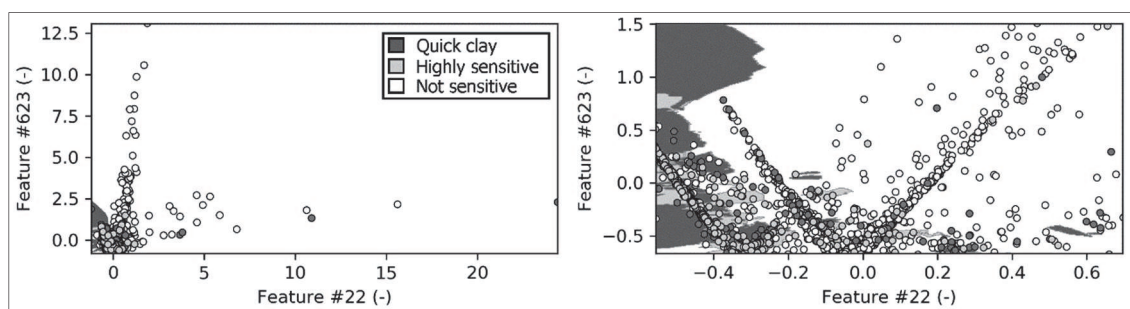


Fig. 2: Decision boundary of a kNNC in 2D (7 neighbors). The figure on the right is a close up, showing about 90% of the points in the figure on the left.

8.3 Nearest Centroid Classifier

The Nearest Centroid classifier (NCC) calculates the centroid (average) for each class in the training set. Each sample test point is given the class of its nearest centroid. Fig. 3 shows decision boundaries of a NCC that has been trained with two features.

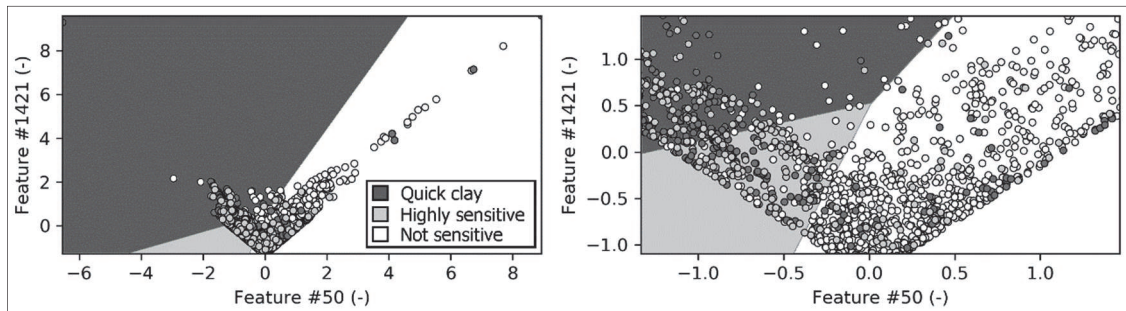


Fig. 3: Decision boundary of a 2D NCC. The figure on the right is a close up, showing about 90% of the points in the figure on the left.

8.4 Support Vector Machines Classifier

The Support Vector Machines classifier (SVC) creates a decision boundary that separates the classes with the greatest possible margin to both sides. This approach often tends to generalize well to out of sample data. Fig. 4 shows decision boundaries of a SVC classifier that has been trained with two features.

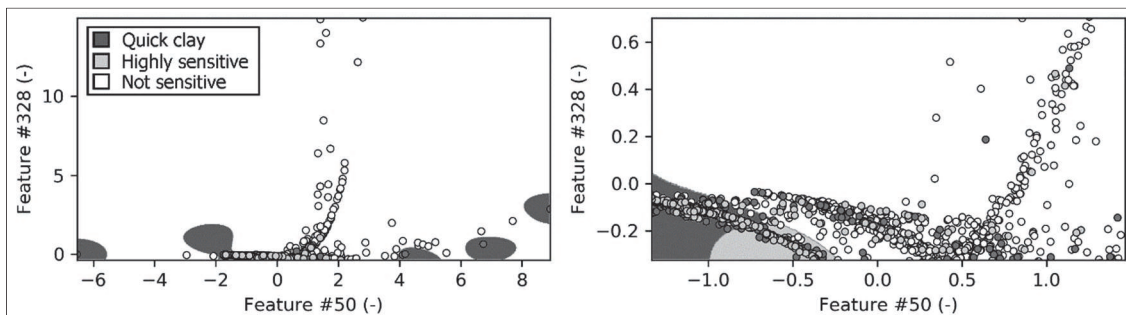


Fig. 4: Decision boundary of a SVC in 2D. The figure on the right is a close up, showing about 90% of the points in the figure on the left.

8.5 Neural Network Classifier

The Neural Network classifier (NNC) used is a multi-layered perceptron (MLP). It works by creating m hidden layers with n perceptrons between the input layer (features) and the output layer (labels). Fig. 5 shows decision boundaries of a NNC classifier that has been trained with two features.

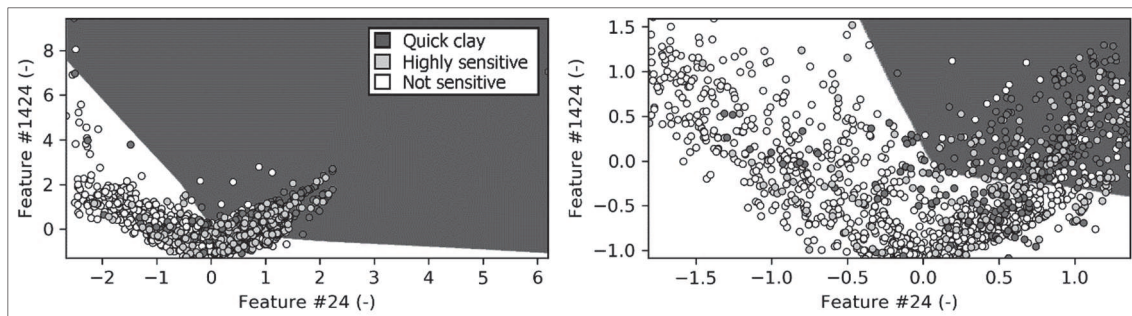


Fig. 5: Decision boundary of a NNC (MLP) in 2D. The figure on the right is a close up, showing about 90% of the points in the figure on the left.

8.6 Random Forest Classifier (ensemble)

The Random Forest classifier (RFC) is an ensemble of DTCs. The number of samples used to grow each tree is the same as the full number of samples available, but the samples are picked at random with replacement, i.e. each tree will have some double points.

The number of features can be less than the total number of features (default is the square root of the total number of features), and in those cases the features are picked at random. Every tree gets an equal vote for the predicted value.

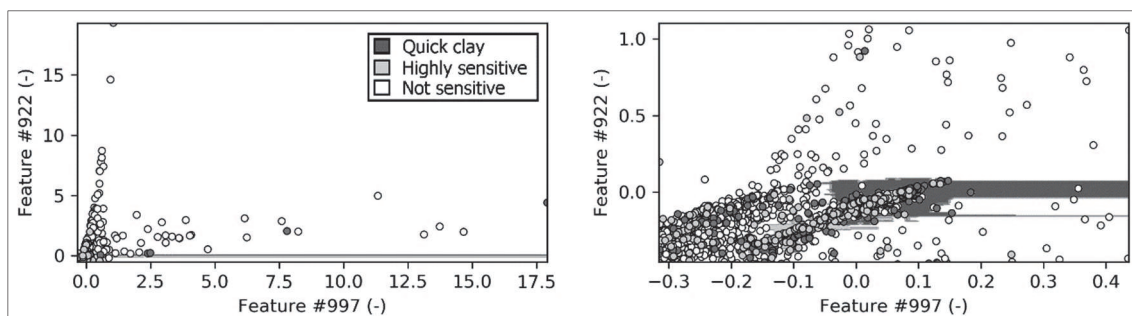


Fig. 6: Decision boundary of a RFC in 2D. The figure on the right is a close up, showing about 90% of the points in the figure on the left.

8.7 Adaptive boosting classifier (ensemble)

Adaptive boosting classifier (AdaBoost) is an ensemble classifier that uses weighted voting to create a strong classifier from weak learners, e.g. DTCs with a single branch. Classifiers are added iteratively and their performance is analyzed during each iteration. Misclassified points are given larger weights, so that they are more likely to be picked in the next iteration. More focus is therefore placed on points that are difficult to classify.

The training performance of each classifier governs the weight they are given in the final vote. The decision boundary for this classifier when trained on 2 features is shown in Fig. 7.

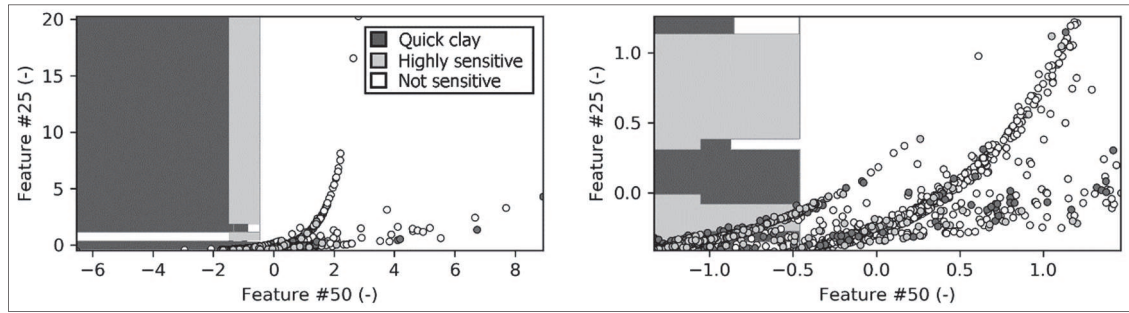


Fig. 7: Decision boundary for the AdaBoost classifier in 2D. The figure on the right is a close up, showing about 90% of the points in the figure on the left.

9 Classifier summary

Fig. 8 shows a comparison of the final results for each model and classifier presented in this study sorted by increasing custom metric score.

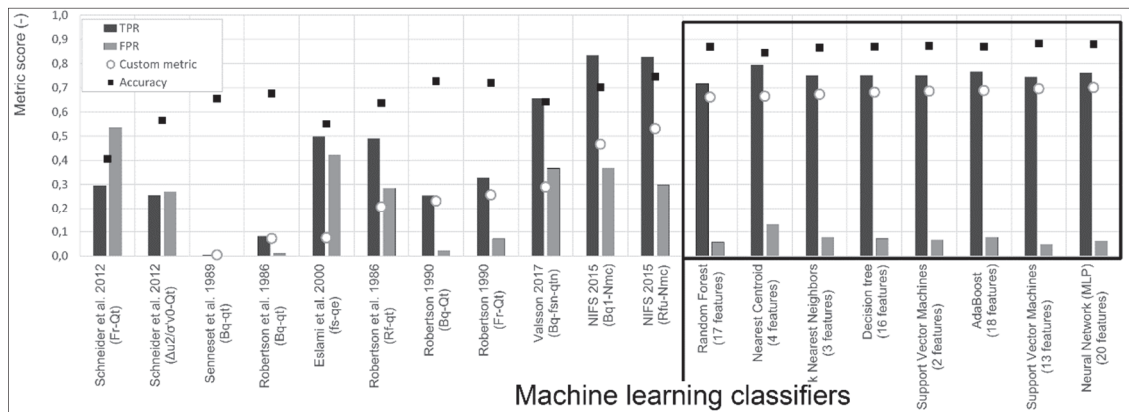


Fig. 8: Final results for all models in study.

Fig. 8 shows that all the machine learning classifiers had a true positive rate between 70 and 80% and in most cases a false positive rate below 10%. The Neural Network classifier produced the highest custom metric score, and the Support Vector Machines classifier produced the lowest false positive rate.

10 Final remarks

This work has shown that machine learning methods can generate a custom metric score of about 70% for this dataset when detecting highly sensitive materials, which translates to an overall classification accuracy of about 90%. These results are better than those found for any of the existing models, and suggest that using tuned classifiers with local training data gives an advantage when mapping highly sensitive materials. Machine learning is shown to give a very promising result, and there is a huge potential to extend it into interpretation of other engineering parameters.

Some scatter is expected in soil investigation results, and a custom metric of about 70% is therefore acceptable. In order to further improve the detection method, a reliability evaluation could be included in the classification. This would give an indication of where predictions are more uncertain.

Machine learning provides powerful tools that can help create insights into large sets of data, but there is no magic involved. Classifiers do not produce better results than the data they are provided with. It is therefore the opinion of the authors that a stronger focus on controlling the quality of the data in the database, and on the preprocessing procedures prior to classification are more likely to produce large improvements in the results than fine-tuning any of these classifiers. This aspect will further be explored as this is an ongoing research project at the Norwegian Public Roads Administration.

11 Literature

- Fellenius B.H, Eslami A. (2000)
Soil profile interpreted from CPTu data. “Year 2000 Geotechnics” Geotechnical Engineering Conference, Asian Institute of Technology, Bangkok, Thailand, November 27-30 2000, Vol. 1, pp 163-171.
- Kotsiantis S.B., Kanellopoulos D., Pintelas P.E. (2006)
Data preprocessing for supervised learning. International Journal of Computational science, Vol. 1- Nr. 1, pp 111-117.
- Lunne T., Robertson P.K., Powell J.J.M. (1997)
Cone penetration testing in geotechnical practice. Publisher: Spon Press, London, pp 53-54.
- Naturfare, Inrastruktur, Flom, Skred (NIFS) (2015)
Detektering av kvikkleire – Sluttrapport. Rapport nr. 126/2015, pp 36-40.
- Pedregosa F., Varoquaux G., Gramfort A., Michael V., Thirion B., Grisel O., Blondel M. Prettenhofer P., Weiss R., Dubourg V. Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M., Duchesnay É. (2011)
Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830.
- Schneider J.A. Hotstrem J.N. Mayne P.W. Randolph M.F. (2012)
Comparing CPTU Q-F and $Q-\Delta u_2/\sigma'_{v0}$ soil classification charts. Géotechnique Letters 2 (Oct. 2012), pp 209–215.
- Thakur V., Degago S. A., Oset F., Aabøe R., Dolva B.K., Aunaas K., Nyheim T., Lyche E., Jensen O.A., Sæter M.B., Robsrud A., Viklund M., Nigussie D. (2014)
Characterization of post-failure movements of landslides in soft sensitive clays. 1st IWLSC. Ch. 24, Publisher: Springer, pp 91–103.
- Valsson S.M. (2016):
Detecting quick clay with CPTu. 17th NGM, pp143-152.