Predicting Land Subsidence by Combining In Situ Testing and Remote Sensing Data

Carmen Martinez Barbosa[1], Faraz S. Tehrani[2,3], and Ana Teixeira[3]

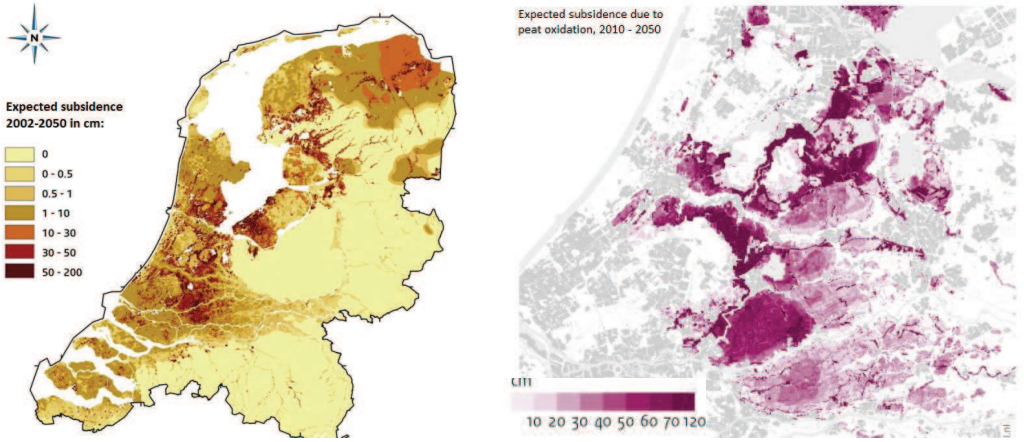
[1] Deltares Software Center unit, Delft, The Netherlands.
E-mail: carmen.martinezbarbosa@deltares.nl
[2] Faculty of Civil Engineering and Geosciences, Technical University of Delft, Delft, The Netherlands.
E-mail: faraz.tehrani@deltares.nl
[3] GEO-unit, Deltares, Delft, The Netherlands.
E-mail: ana.teixeira@deltares.nl

Abstract: Land subsidence in peat areas is a known issue in the Netherlands, and it is responsible for damages in housing and infrastructure. There is still a lot of uncertainty in predicting this phenomenon and this research aims to improve the predictability of subsidence in the Netherlands by using Machine Learning (ML) techniques. A proof-of-concept of this idea is presented in detail in this article. We collect publicly available data of cone penetration testing (CPT) in the Netherlands along with remote sensing measurements of the subsidence rate. We use three flood protection dikes for this study. The predictors of the ML models are CPT measurements, that include cone tip resistance and sleeve friction; and the variable to predict (i.e. target) is the subsidence rate. The ML techniques used to predict the subsidence rate include Ridge regression, Random Forest (RF) and Gradient Boosting Machines (GBM). Evaluating the prediction accuracy among these methods it is found that the GBM gives the lowest average prediction error. Among features used in the prediction models, it is seen that the cone tip resistance at certain depths is the parameter that contributes more at predicting the rate of subsidence. This proof-of-concept case study shows that there is a promising relationship between subsidence rate and cone penetration data.

Keywords: Subsidence; in-situ testing; machine learning; uncertainty

1 Introduction

Land subsidence in peat areas is a known issue in the Netherlands (Figure 1). It is responsible for damages in housing and infrastructure. Studied examples in the Netherlands include: the city of Gouda (Willemse 2018), but also some railways (Peduto et al. 2018) and sewage systems (Abspoel et al. 2018).



(a) (b)

Figure 1. Expected subsidence map (when no measures are taken) for the Netherlands, (a) between years 2002 and 2050 (source: Deltares), (b) between years 2010 and 2050 (source: pbl.nl).

Subsidence is the downward motion of the earth surface and it involves primary settlement and the secondary settlement of the sub-soil, also known as creep. In the Netherlands, the main causes of subsidence are as follows: (i) extraction of natural gas, (ii) groundwater changes, due to e.g. water extraction and (iii) loading weak grounds with e.g. landfill. Other causes are underground excavation, mining, or tectonic motion. In

Proceedings of the 7th International Symposium on Geotechnical Safety and Risk (ISGSR)
Editors: Jianye Ching, Dian-Qing Li and Jie Zhang

Published by Research Publishing, Singapore.
ISBN: 978-981-11-2725-0; doi:10.3850/978-981-11-2725-0_MS2-5-cd

addition, peat oxidation due to climatic variables such as high temperature and drought can cause land subsidence.

In the Netherlands, land subsidence, increased number of storms and sea level rise force further upgrades to the flood control and water management infrastructure. Given the remarkable uncertainty in estimating the subsidence rate due to complex behavior of sub-surface materials, and due to importance of dikes in the Netherlands, as the main flood protection elements, the research presented in this article aims to improve the predictability of subsidence of dikes. To reach that end, we study the usability of Machine Learning (ML) techniques.

2 Problem Definition

We aim to provide answer to this question that how in situ testing data and remote sensing data can be used together to better predict land subsidence. The motivation behind this effort comes from the following drivers:

1. Subsidence is a problem that needs more attention, in the Netherlands but also in other countries;
2. Standardized geotechnical databases are readily available in the Netherlands and will be even more complete in the near future.
3. Subsidence phenomena are not yet totally understood, leading to current modelling be based on many (expert-based) assumptions.

Given these drivers, we look forward to establishing a link between the standardized geotechnical data and the subsidence data (obtained from remote sensing methods), by using Machine Learning (ML) algorithms, where the geotechnical data is the model predictor and the target is the subsidence rate.

3 Dataset

3.1 Source of the data

In this study, we aim to predict the rate of subsidence of dikes located at three locations in the provinces of North and South Holland (in the Netherlands). The *in situ* data are cone penetration tests (CPT), associated with each dike and obtained from the DinoLoket web portal (www.dinoloket.nl – standardized geo database). The selected CPT's are taken at the crest of the dikes only. This results in 393 CPT's in total. The spacing between CPT's ranges between 50 and 150 meters for two of the segments (14-1 and 13-8) and 600 meters for the remaining dike segment (8-1,8-2). The subsidence map was acquired from the web portal of SkyGEO (bodemdalingskaart.nl). The resolution of this map/information is 2 km × 2 km, which is not ideal but serves the purpose of this proof-of-concept study. It is understood that a better resolution of the subsidence map will lead to a better dataset with more variance in the predicting value or target (rate of subsidence), which in turn might lead to a better prediction model. Figure 2 shows the selected locations (dike segments), and the measured rate of subsidence of the Netherlands (color map).

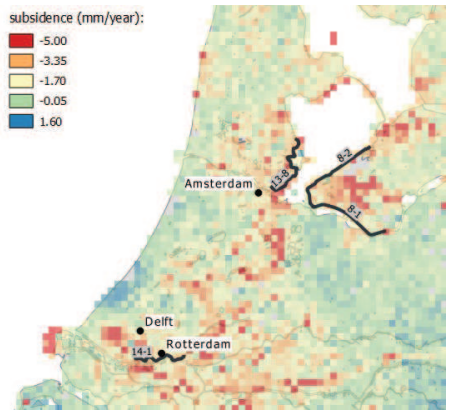


Figure 2. Current rates of subsidence in the Netherlands (source: bodemdalingskaart.nl).

3.2 Pre-processing

Preprocessing of the CPT data and the rate of subsidence was necessary to prepare the chosen dataset for machine learning (ML). Figure 3 shows the preprocessing workflow. As shown in Figure 3, first, the raw CPT data was converted into shapefile format, which is a format for storing the geometric location and attribute

information of geographic features. This dataset and the subsidence rate data (u_z) were then linked, and a csv file was created. This csv file contains the subsidence rate associated with the location of each CPT. Finally, the dataset including features extracted from the raw CPT data and the rate of subsidence, was created. The CPT features used in this study include cone tip resistance (q_c), sleeve friction (f_s), friction ratio ($R_f = f_s/q_c$) and penetration length (z). Given the presence of R_f in the features and its dependence on f_s and q_c, either q_c, f_s or R_f can be removed, but we have decided to keep them for completeness.

The CPT data is measured every ~2 cm; however, the total depth of the observations is different per CPT; some are greater than others. We set the depth of CPT's to 30 m. With this threshold, the number of observations per physical quantity is 1500 and each of these observations is taken as a predictor of the model. For instance, let us take the cone tip resistance q_c:$q_{c,0}$ is referred to as the cone's tip resistance at the surface and $q_{c,\ 1500}$ is the cone's tip resistance at 30 m depth from the surface. This structure applies also to f_s, R_f and z. Therefore, the preprocessed dataset has 6000 predictors in total, where each row corresponds to one CPT. To solve the issue of the difference in depth in the CPT data, we oversample those CPT's shallower than 30 m. The oversampling is carried out by increasing the frequency of data points (of those CPT's shallower than 30 m), so that every CPT has the same amount of data points (necessary for the machine learning). This is achieved by linear interpolation of every consecutive CPT points to acquire more samples in between. Conversely, the CPT profiles deeper than 30m were truncated at the depth 30 m. At the end, the post-processed dataset has 393rows (number of CPT's) and 6001 columns (predictors + target).

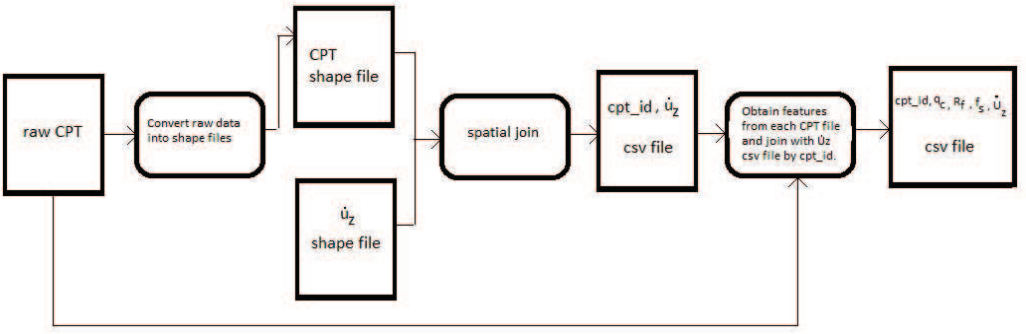


Figure 3. Data preprocessing workflow.

3.3 Feature reduction

Given the number of features, it was decided to experiment with certain feature reduction methods to study the effect of raw data (denoted as **No feature reduction**) and engineered features on the final results. Feature reduction increases prediction performance of the ML models. The methods we apply in this study are the following:

1. **F-test:** A univariate linear regression test was applied to raw features for testing the individual effect of each of many regressors. The null hypothesis is independent between target and feature X. The number of features selected by this method is 5469. Though, this number of features is not significantly reduced when compared with the total number of 6000 features,
2. **Shallow trees**: The decision trees always use the most important features to create the first nodes in the tree. Therefore, shallow decision trees can be a powerful technique to reduce the number of features in a dataset. To account for uncertainties in different algorithms, we use different ensemble of trees such as random forest, light Gradient Boost Machine (GBM), eXtreme Gradient Boost (XGBoost) and Categorical Boost (CatBoost) for this purpose. We run each of these ML algorithms on the post processed dataset, during several times. This allows us to obtain a distribution of importance per feature. From each distribution we take the average value. The features selected for modelling are those whose average importance is higher than a limit. That limit is defined as the product of a threshold and the mean of all the features' importance. In this study, we set the threshold as 1.0. A higher threshold leads to the selection of less features. The number of features selected by this method is 717.

4 Machine Learning Algorithms

4.1 Ridge linear regression

One of the main assumptions of the ordinary least squares (OLS) method is the independence of the predictors. When the features are correlated, the OLS estimate becomes highly sensitive to random errors in the prediction

which produces a large variance in the outcome. Therefore, the model is not accurate, nor reliable. One way to solve this problem is by imposing a penalty on the size of coefficients. This approach, which is called Ridge regression (Hoerl and Kennard 1970), makes a linear model to be robust to collinearity among features in the dataset. Due to this characteristic, Ridge regression is a more suitable approach for datasets that have hundreds of features. The formula used in Ridge regression that minimizes the residual sum of squares between the true and predicted values of the target follows:

$$\min_w \left|\mathbf{w}^T\mathbf{X}\text{-}\mathbf{y}\right|^2 + \alpha\left|\mathbf{w}\right|^2 \quad (1)$$

where $\mathbf{w} = (w_1, w_2, .., w_n)$ is the vector of model coefficients; $\mathbf{X}$ is the matrix of predictors; $\mathbf{y}$ is the target vector; and $\alpha \geq 0$ is a coefficient that controls the degree of overfitting. The larger this quantity, the coefficients become more robust to collinearity.

4.2 Random forest

To explain the concept of random forest (Breiman 2001), it is necessary to first introduce the decision trees. A decision tree is an algorithm where a set of conditions are imposed from the data. The target variable is then predicted based on those rules. In Figure 4, a decision tree built from our dataset is illustrated.

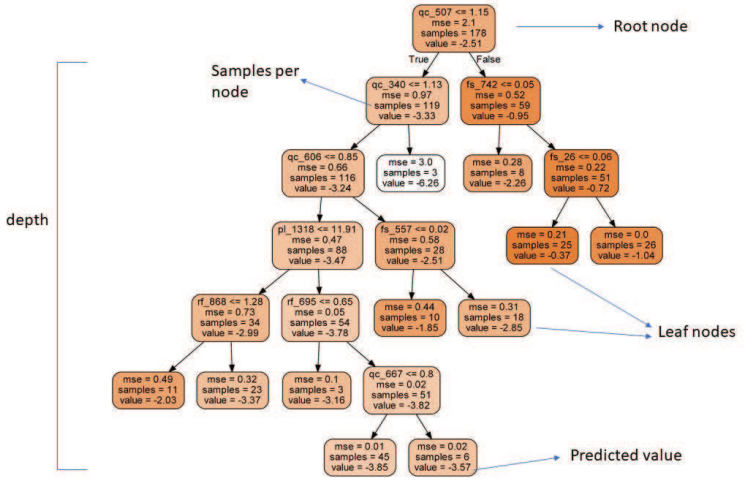


Figure 4. Structure of a decision tree. The predicted values in each node is the average of the rate of subsidence of the points that belong to each of the rules shown.

In regression problems, a decision tree splits the data into subsets always minimizing the standard deviation. Therefore, a feature is selected if it provides a high decrement of variance in the observed data (i.e. not too much noise). Because of this, a single decision tree is prone to overfitting. Additionally, a model based on a single decision tree is sensitive to different datasets. A solution to overcome these disadvantages is to have an ensemble of decision trees. The final prediction is then an average of the outcome of each tree. This is the aim of the random forest. First it creates *n* bootstrapped samples of the train dataset, then for each of these samples it creates one decision tree using random features (it always selects those that minimize the variance). In the end, the final prediction is the average of the target values that belong to the same set of rules.

4.3 Gradient boost machine

The boosting methods (Friedman 2001; Chen and Guestrin 2016) use the same principle as the random forest explained above. However, the Gradient Boost Machine goes a step forward, because instead of only making an average of values to create the prediction, they also include the prediction error at every iteration. This way we can improve the prediction. The Gradient Boost Machine is therefore, a very powerful non-parametric method that can deal with non-linear relationships in the data.

5 Results and Discussion

Before building the predictive models, the dataset is split into train and test subsets. Instead of selecting the observations at random, we make sure that 70% of each dike's CPT's randomly fall in the training set. Following

this, we avoid having more data for a specific dike segment which can make the code more biased towards that information. We also fixed the random seed so that the results can be reproducible. In all these analyses, we used Python 3.7.1 and the Scikit-learn library (Pedregosa et al. 2011) which contains all the ML methods explained here. Random forest and GBM can capture non-linear relationships in the data and they can be interpreted more easily than complicated "black-box" models such as deep learning models. However, as mentioned earlier, we also build a Ridge model to compare the results between a linear regression approach and ensemble of trees methods.

In Table 1, the error metrics of the experiments that were carried out in this study are shown (i.e. 3 feature reduction scenarios and 3 ML models). MdAPE stands for Median Absolute Percentage Error; the RMSE is the root mean squared error; R^2 is the explained variance of the model and Pearson correlation is the correlation of the residuals. Both RMSE and MdAPE measure the accuracy of a model; however, RMSE is used more to compare models built from the same dataset while MdAPE is used to compare models from different datasets, where the units of the target variable are different or when the predictors used are also different. When the explained variance R^2 of a model is negative, the model does not fit the data properly and another ML algorithm needs to be applied. In Table 1, we see that this is the case for the Ridge regression ML model. In general, Ridge regression is easy to interpret; however, the dataset needs to be linearized by using many mathematical transformations to obtain a good model. One can claim that the easier an ML method is to interpret; the more feature engineering needs to be done on the train data.

Table 1 shows that the best models (i.e. better error metrics compared to others) are obtained using gradient boosting methods (GBM). As explained in the previous section, this result is expected, since boosting methods improve the prediction in the leave nodes at every iteration, while random forest only averages them. If we compare the error metrics among the GBM models listed in Table 1, we note that the scenario where shallow trees are used as feature reduction technique leads to the best model. However, the other models built with GBM have a similar prediction performance, thus we can select any of these to further analyze the causes of the rate of subsidence in the dikes in the Netherlands. We select the scenario where no feature selection is applied to the data. Figure 5 shows the (a) top 10 most important features selected by the GBM model (with no feature reduction) together with (b) the residuals plot.

The feature importance is a percentage and it accounts for the change of the model's accuracy when a variable is excluded. The higher the change in accuracy, the more important the variable is. According to Figure 5(a), the top 10 most important drivers at predicting the rate of subsidence in the Dutch dikes studied herein is the cone tip resistance (q_c) at different depths. It was found that features as R_f or f_s have an importance close to zero.

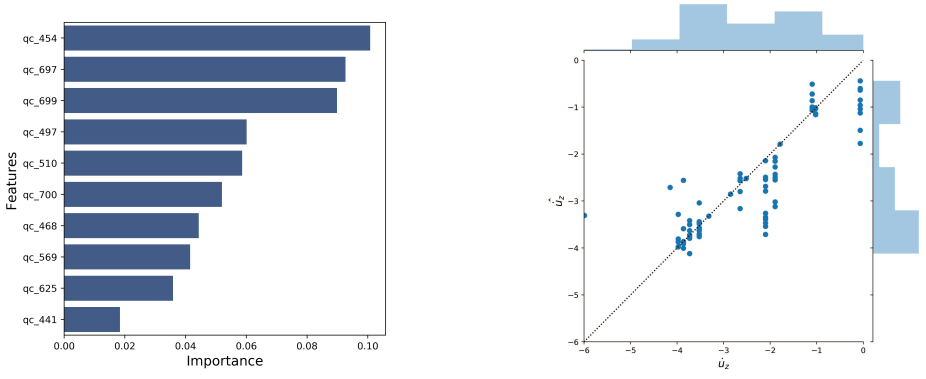


Figure 5. (a) 10 most important features in the selected model. (b) Residuals plot showing the distributions of the real and predicted rate of subsidence in the test set.

Table 1. Error metrics in the test set of the experiments that were carried out. Highlighted are the models with best prediction performance.1.

Scenario	ML model	MdAPE (%)	RMSE (-)	R^2 (-)	Pearson corr. (-)
No feature reduction	Ridge regression	34.8	4.0	-7.4	0.3
F test	Ridge regression	49.7	5.6	-15.0	0.3
Shallow trees	Ridge regression	38.1	4.6	-9.5	0.4
No feature reduction	Random Forest	12.0	0.9	0.6	0.8
F test	Random Forest	11.8	0.9	0.6	0.8
Shallow trees	Random Forest	10.8	0.8	0.67	0.8
No feature reduction	GBM	4.3	0.8	0.7	0.8
F test	GBM	4.7	0.8	0.7	0.8
Shallow trees	GBM	3.9	0.7	0.8	0.9

6 Conclusions

A proof-of-concept was shown in this article, where we explored the link between in situ testing and remote sensing data to predict subsidence rate based on CPT measurements. We collected publicly available data and defined a case study (three dike segments). After pre-processing the datasets, three ML techniques were used to predict the subsidence rate. The results showed that the Gradient Boosting Machines gave the lowest average prediction error of around 4%. In the analyzed case study, it was observed that the cone tip resistance at different depths was the parameter that contributed the most at predicting the rate of subsidence. On the other hand, it was found that friction ratio or sleeve friction did not play an important role at predicting the rate of subsidence. The results of this study showed that it is possible to link CPT data to subsidence rate obtained from satellite remote sensing. Furthermore, it can be hypothesized that the link between CPT's and remote sensing data has the potential to also be extended to primary consolidation/settlement or even inversely, to predict a CPT/soil layering based on measured settlement. This preliminary study has certainly identified crucial points to a successful prediction of subsidence based on ML: data quality and feature engineering. A reliable subsidence rate measurement source, with enough resolution is, of course, of high importance; other data such as loading will be of relevance in next research steps, especially if one aims to consider primary consolidation. As such, further improvements can be carried out in both the data preprocessing and in the modelling. These steps will be developed in our future studies.

Acknowledgments

We thank Deltares research programs "Future-proof dikes" and "Enabling Technologies", for allowing this study to kick-off, in particular we thank Frank den Heijer, Meindert Van, and Arjan Venmans for their support.

References

Abspoel, L., Courage, W., Dabekaussen, W., de Bruijn, R., Kruse, H., Wiersma, A.P., Hijma, M.P., van den Heuvel, F., and van den Broeck, W. (2018). Risk-based asset management: automated structural reliability assessment of geographically distributed pipeline networks for gas and water in the Netherlands. *Structure and Infrastructure Engineering*, 14(7), 928-940. DOI: 10.1080/15732479.2018.1437641.

Breiman, L. (2001). Random forest. *Machine Learning*, 45, 5-32. DOI: 10.1023/A: 1010933404324.

Chen, T and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794. Available at: http://doi.acm.org/10.1145/ 2939672.2939785

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189-1232.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55-67. DOI: 10.1080/00401706.1970.10488634.

Pedregosa et al. (2011). Scikit-learn: Machine learning in Python, *JMLR*, 12, 2825-2830.

Peduto, D., Huber, M., Speranza, G., van Ruijven, J., and Cascini, L. (2017). DInSAR data assimilation for settlement prediction: case study of a railway embankment in the Netherlands. *Canadian Geotechnical Journal*, 54(4), 502-517. Available at: https://doi.org/10.1139/cgj-2016-0425.

Willemse, N. W. (2018). *Stedelijke Ontwikkeling en Bodemdaling in en Rondom Gouda - Een synthese van Drie Onderzoeken naar de Relatie Tussen (stedelijke) Ontwikkelingen en Bodemdaling*, Rapport Gemeente Gouda, ISBN 978-90-5372-116-2. (in Dutch)