Interpretation of CPTu data using machine learning techniques to develop the ground model of a dam

Mauro Giuliano Sottile[1,2*], *Jodie Amberly* Crocker[1] and *Lisandro* Roldan[1,2]

[1] *SRK Consulting, Buenos Aires, Argentina*
[2] *Universidad de Buenos Aires, Argentina*
[*] *msottile@srk.com.ar*

ABSTRACT

Building a ground model through manual processes can be time consuming, as large amounts of data need to be classified to define the extent and spatial distribution of the different soil materials. This paper delves into the application of machine learning (ML) methodologies, in conjunction with in-situ geotechnical testing data, to develop the ground model for a downstream dam founded on both weak and liquefiable soils. The dam covers a linear extent of approximately 800 m and was extensively characterized by means of in-situ tests, including 206 cone penetration tests (CPTu), 37 boreholes and 35 test pits. The performance of two unsupervised ML clustering algorithms are compared: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and an extended version with a hierarchical component (HDBSCAN). The clustering uses CPTu data, which consists of the normalized cone tip resistance (Q_{tn}) and the normalized sleeve friction (F_r) varying with elevation. Nearby borehole logs are used to evaluate the results of both clustering methods for a single single CPTu sounding using different clustering parameters. Then, a global clustering including several CPTu soundings is done and results are compared with the ground model that was manually made using Leapfrog software. Both methods show very good performance, with HDBSCAN being better and more robust.

Keywords: Ground Model; CPT; Machine Learning; DBSCAN; HDBSCAN

1 Introduction

Building a ground model is an essential step for geotechnical engineering applications, such as slope stability analysis, liquefaction assessment, or foundation design. However, manual processes for defining soil layers and their properties can be time consuming, subjective, and dependent on the expertise of the person creating the model. This is particularly challenging for tailings dams due to the large amount of CPTu data typically available, combined with the heterogeneous composition of the materials resulting from its complex deposition. Due to these difficulties, many practitioners often simplify the sub-layering and define properties based on frequency analyses done on large and non-homogeneous layers, which can result in unrealistic or non-conservative estimates of properties.

Many authors have recently proposed methodologies to automate geotechnical stratigraphic profiling from CPTu data, aiming to enhance both the efficiency and accuracy of constructing ground models. To illustrate: Collico et al (2023) introduced a semiautomated tool based on probability; while Brinkgrieve et al (2022) investigated the use of machine learning (ML) algorithms for clustering.

This paper delves into the application of ML methodologies in conjunction with in-situ geotechnical testing data to develop the ground model for a downstream dam founded on both weak and liquefiable soils. It examines the grouping capabilities of two clustering algorithms: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and an extended version with a hierarchical component (HDBSCAN). First, the clustering is done individually for each CPTu sounding using the normalized cone tip resistance (Q_{tn}) and the normalized sleeve friction (F_r) varying with elevation; then, nearby borehole logs are used to assess how well both methods work with different clustering parameters. Finally, a global clustering grouping multiple CPTu soundings is conducted and compared with a manually constructed ground model.

2 Case study

2.1 Dam description

The case study is a 26 m-high downstream-raised dam. It was built in the 1980s as a water retention

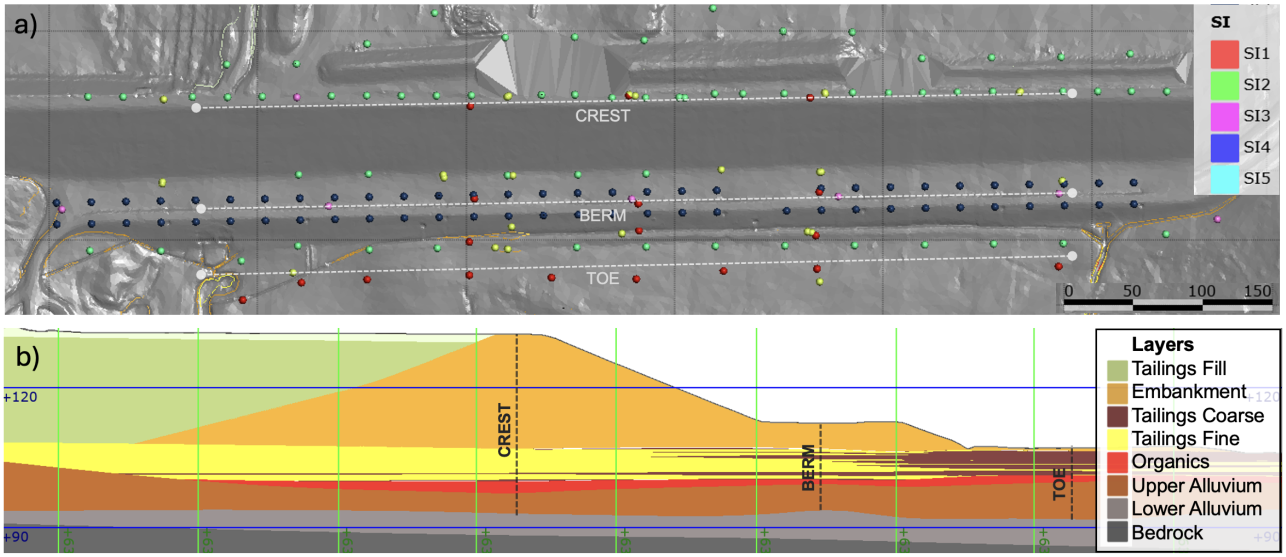

Figure 1. Dam considered for this case study. a) shows the plan view of the site with CPTu locations and b) shows a typical cross-section at the center of the valley. The dam is divided into three general parts: the crest, berm, and toe.

structure over an 800 m-long shallow valley that had been previously covered with 6-8 m of hydraulically deposited tailings. Below this unit, there is a very soft clay layer with organic content and two natural alluvial layers: an upper unit that is mostly clayey and soft, and a lower unit that is harder and has some gravel mixed within the fine soil matrix. Figure 1a shows a plan view of the site, while Figure 1b shows a typical cross-section at the center of the valley. Note that the dam is divided into three general parts: the crest, berm, and toe.

The site has been extensively studied with in-situ tests, including borehole logs (BH), test pits (TP), cone penetration tests with pore pressure measurements (CPTu), ball penetrometer tests (BPT), seismic dilatometer tests (SDMT) and multi-channel analysis of surface waves (MASW) tests. In Figure 1a, all CPTu sounding locations are illustrated, which are the main inputs of this study.

2.2 Ground model

The in-situ testing data was used to manually define a geotechnical stratigraphic profiling and produce a 3D ground model using Leapfrog software; full details are presented in Rola et al. (2024). The layers were defined using CPTu data and nearby BH logs; Figure 2 shows an example of a CPT-BH pair located at the center of the dam's crest. A description of the main geotechnical units are summarized as follows:

- Embankment Fill: a compacted clayey material that forms the main embankment. It has a q_t around 5 MPa, f_s between 150 and 350 kPa, B_q near 0 and I_c larger than 2.7.

- Tailings: located below the embankment and extends laterally across the dam area. It has q_t ranging between 5 MPa and 10 MPa, f_s between 0 and 300 kPa and uniform B_q near 0. It is subdivided into coarse ($I_c < 2.6$) and fine ($I_c > 2.6$) subunits, as the coarser portion shows a drop in the sleeve friction measurements (e.g., RL100-101m in the sounding shown in Figure 2).

- Organics: a natural soft clayey layer located below the Tailings; the borehole log depicts the presence of roots and organic content. In CPTu logs, it shows very low q_t and f_s values, along with a noticeable spike in B_q.

- Upper Alluvium: a natural clay layer below the Organics layer. It has a low strength matrix depicted by low q_t and f_s values along with some spikes in B_q; however, there are sections of higher strength, probably due to the presence of gravels.

- Lower Alluvium: a gravelly clay layer below the upper alluvium. Identified by a sudden increase in tip resistance and negative/null B_q values. CPT refusal occurs in this layer.

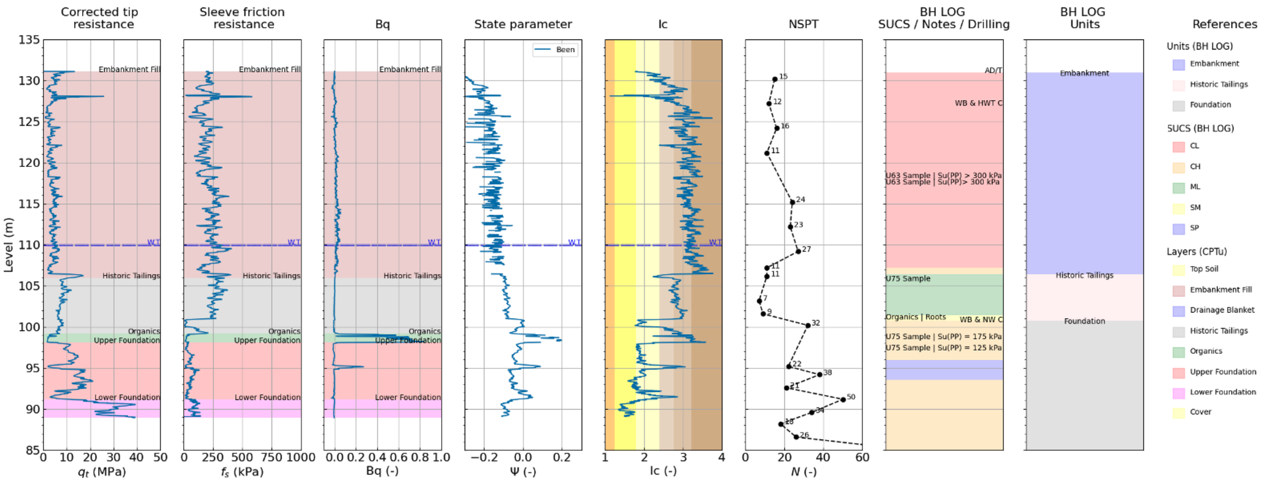

Figure 2. Example of layer identification using CPTu and BH logs located at the dam crest. The interpretations for this sounding (first three columns) were manually done by considering the CPTu data, CPTu correlations, and the BH log.

3 Clustering methods

3.1 DBSCAN

DBSCAN is a machine learning clustering algorithm that defines clusters of data based on the density, or closeness, of the data (Ester et al., 1996; Schubert et al., 2017). This allows the algorithm to focus on dense regions in the dataspace, while data outside of these dense regions are treated as outliers or noise. Two user-defined parameters are required for the algorithm: *epsilon* (*eps*) and *minPts*. For a given datapoint, *eps* defines the radius that is used to search for its neighboring datapoints. The number of points within this radius is then compared to *minPts*, which determines if a new cluster is created. Using these two parameters, the method proceeds through a dataset and labels points as either core, border, or outlier points. A core point exists if it has at least *minPts* number of points within the *eps* distance surrounding the sample. Border points are neighbors of core points that are within *eps* distance of the sample but have fewer points than *minPts*. Points that either have fewer neighboring points than *minPts* or are not within *eps* of the sample are labeled outliers. An example of these points is shown in Figure 3a, where a core, border, and outlier point are each illustrated using a sample dataset with $eps = 0.4$ and $minPts = 5$. Points are then connected if they are in the same neighborhood as one another or if they are mutually connected through another point, which forms clusters as shown in Figure 3a. Note that this sample dataset was created using four random clusters of data, but only three clusters are provided by DBSCAN.

In general, DBSCAN is well suited for data with non-uniform trends and noise, as it can create non-spherical clusters that do not include noise or outliers (see Figure 3a). Additionally, a priori information is not required to use DBSCAN, although data analysis may aid in choosing appropriate values for *eps* and *minPts*. Although DBSCAN has many advantages, it is important to note that its results are sensitive to the choice of *eps* and *minPts*. In general, small values of *eps* result in many clusters and outliers, while large values of *eps* may lead to a single cluster with most of the data included. However, this is also influenced by *minPts*, as particularly noisy data may be difficult to capture with a poorly chosen *minPts*. Therefore, when using DBSCAN for highly variable data such as the dataset shown in this study, it is necessary to cluster using a suite of *eps* and *minPts* values to find the best clustering results.

3.2 HDBSCAN

HDBSCAN is an extension of the original DBSCAN method that converts DBSCAN into a hierarchical clustering algorithm (Campello et al., 2013; 2015). Due to its clustering method, DBSCAN struggles to capture clusters of different densities. For example, choosing a large *eps* may result in small clusters being merged to form one larger cluster. HDBSCAN circumvents this by clustering over all possible values of *eps* and selecting the most persistent clusters from all possible clusters. Similar to DBSCAN, HDBSCAN begins by finding core points, or points that contain a minimum number of neighboring points within a core distance. This core distance changes for each core point, such that points in areas of low density will have larger core distances compared to points in areas of high density. These core distances are then used to compute a "mutual reachability distance" (MRD), which is the maximum of three distances: the core distance of point A, the core distance of point B, and the distance between points A and B. The purpose of this calculation is to bias the clusters towards regions of higher density, causing datapoints in less dense regions to be considered as outliers.

After finding the MRD between all points, a mutual reachability graph is created. In this graph, every datapoint is a vertex connected to one another by edges whose weights are equal to the points' MRDs. HDBSCAN then builds a minimum spanning tree (MST) (i.e., a graph where all vertices are connected with minimum possible edge weight), and a single cluster is created and labeled for the MST. The edge with the highest weight is then removed from the MST, and cluster labels are assigned to the remaining connected components that contain an end point from the removed edge. If any remaining group has fewer points than a user-defined minimum cluster size, those points are labeled as outliers and the algorithm continues splitting the remaining components. This process is repeated until there are no more connected components.

An advantage of HDBSCAN is that the only required parameter is the minimum cluster size (*mcs*). This controls the number of points required for a cluster to form. In general, this is more intuitive than the *eps* and *minPts* required by DBSCAN, as it is directly related to the minimum size a cluster can be. Thus, while HDBSCAN may be more computationally complex than DBSCAN, it is easier to use, and its insensitivity to the chosen *mcs* value means few iterations are required to achieve good results. An example of HDBSCAN clustering is shown in Figure 3c. Note that the sample dataset contains four clusters, which HDBSCAN identifies, compared to the three clusters provided by DBSCAN (Figure 3b).

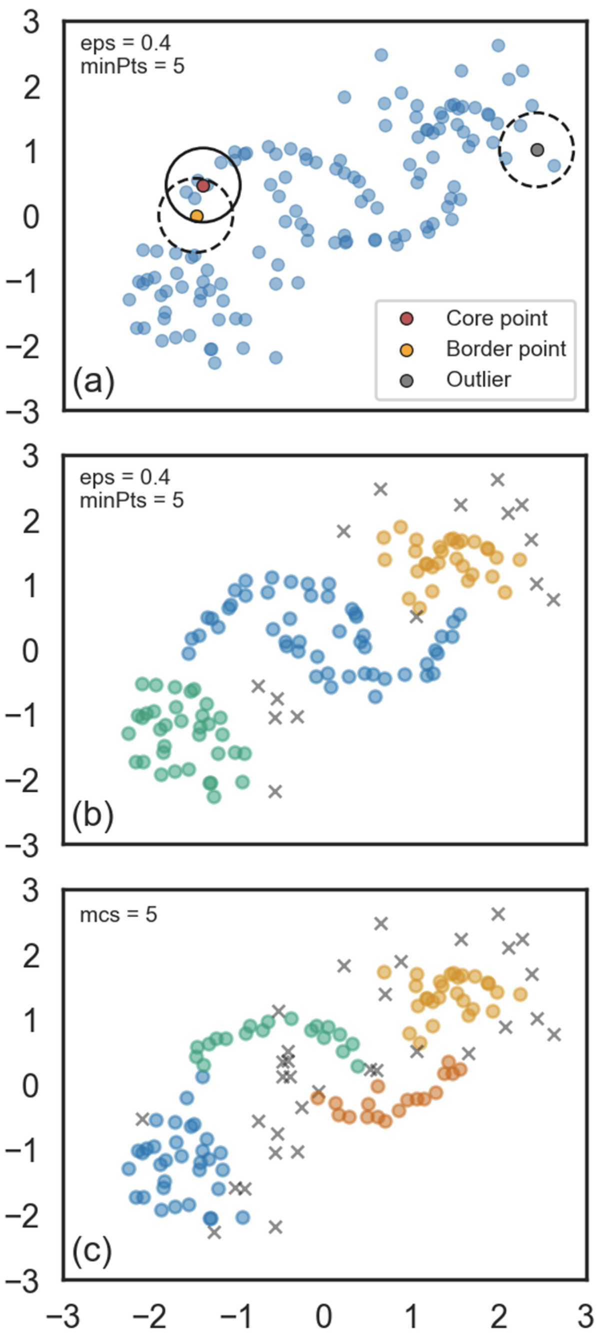

Figure 3. A sample dataset containing four unique clusters with additional noise. In (a), a core point, border point, and outlier are shown for the DBSCAN parameters *eps* = 0.04 and *minPts* = 5. The resulting DBSCAN clusters are shown in (b), while HDBSCAN is used with *mcs* = 5 to form the clusters shown in (c). Note that HDBSCAN finds the true number of clusters in the sample dataset compared to DBSCAN at the cost of more noise.

4 Individual CPTu clustering

The CPTu data for this case study includes 206 soundings collected over 800 meters. As such, manually interpreting these soundings and cross-referencing them to their nearest boreholes is time consuming.

A possible solution is to use DBSCAN and HDBSCAN to cluster the CPTu data and form layers automatically. However, it is first necessary to determine how well these clustering algorithms perform when applied to this dataset, as many locations in this case study have high spatial variability (i.e., thin soil layers interspersed between larger layers).

To begin the analysis, several CPTu soundings are selected to represent various locations across the site. Specifically, a CPTu sounding is selected at the crest of the dam (CPT-Crest), along the berm (CPT-Berm), and at the toe (CPT-Toe). Three variables are used to perform 3D clustering: elevation, the logarithm of the normalized cone resistance ($\log(Q_{tn})$) and the logarithm of the normalized friction ratio ($\log(Fr)$). The latter two values, $\log(Q_{tn})$ and $\log(F_r)$, are selected in accordance with the updated CPT-based SBTn chart proposed by Robertson (2009). Elevation was chosen as the third variable so the clustering would consider layer deposition. To prepare the dataset for clustering, each of these variables are normalized to a range of (0,1) to prevent biased clusters. Finally, the DBSCAN and HDBSCAN implementations from the Scikit-learn open-source library (Pedregosa et al., 2011) are used for clustering.

Because DBSCAN is particularly sensitive to the choice of *eps* and *minPts*, a suite of values are tested on one CPTu sounding (CPT-Toe) to determine the best pairing of *eps* and *minPts* that could be used as starting values for the remainder of the dataset. The clustering results for several *eps* and *minPts* pairings are shown as Figure 4b-g with the "true", or manual, interpretation shown as Figure 4a. Note that the colors in Figure 4b-g do not represent any specific soil type but are instead used to distinguish clustered data, while outliers are plotted in white. Although many combinations were tested, only combinations of eps = (0.04, 0.06, 0.08) and $minPts$ = (5, 10) are shown. When reviewing the clusters, it seems that using a small value of $eps = 0.04$ results in many small clusters (Figure 4b) or, when combined with a larger *minPts*, many outliers (Figure 4c). As *eps* increases, more layers are clustered together, with many small layers merging into larger ones (Figure 4d-e). Finally, when choosing a rather large *eps*, the results are less sensitive to *minPts*, as Figure 4f-g show that eps = 0.08 and $minPts$ = 5 or 10 both yield decent results compared to the "true" interpretation. After reviewing the clustered layers for a variety of *eps* and *minPts* values, it was determined that $eps = 0.08$ and $minPts$ = 10 are the best values to use for individual CPTu clustering.

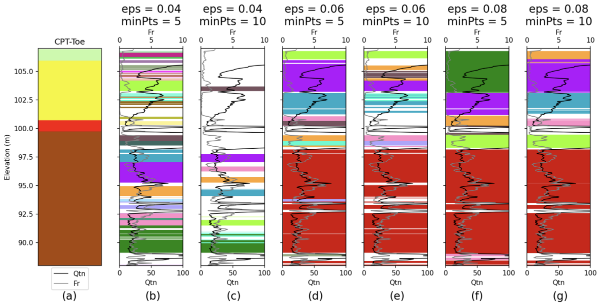

Figure 4. A sample CPTu dataset (CPT-Toe) is clustered using DBSCAN. The "true" interpretation of the CPTu is shown in (a), while pairings of eps = 0.04, 0.06, and 0.08 and $minPts$ = 5 and 10 are used with DBSCAN to obtain the results shown in (b-g). The parameters Q_{tn} and F_r are plotted in (b-g) to help identify cluster trends.

Although HDBSCAN is not particularly sensitive to the choice of *mcs*, the process of testing multiple parameter values is repeated for demonstration. Values of mcs = (5, 10, 15, 20, 25, 30) are all used to perform clustering, and the results are shown as Figure 5. In Figure 5b, a recommended starting value of $mcs = 5$ results in many thin layers/clusters as expected. This is likely due to the large amount of data collected for each CPTu sounding; thus, the best clustering results occur when using a slightly larger mcs = 10-15 (Figure 5b-c). Beyond this value, the results are largely the same, although very large values of *mcs* result in more outliers. It should be noted that only three iterations are required to find the best *mcs* value here (mcs = 5, 10, and 15), and therefore it is less difficult for the user to tune HDBSCAN's parameter compared to the DBSCAN parameters.

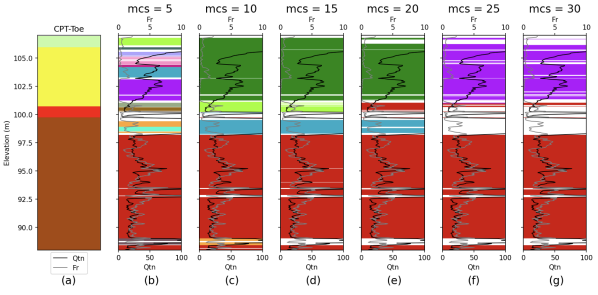

Figure 5. A sample CPTu dataset (CPT-Toe) is clustered using HDBSCAN. The "true" interpretation of the CPTu is shown in (a), while different values of mcs = 5, 10, 15, 20, 25, and 30 are used with HDBSCAN to obtain the results shown in (b-g). The parameters Q_{tn} and F_r are plotted in (b-g) to help identify cluster trends.

Following the selection of *eps* and *minPts* for DBSCAN and *mcs* for HDBSCAN, clustering is performed on each CPTu sounding. Figure 6 shows a comparison with nearby BH logs for CPTu soundings located at the crest (CPT-Crest), berm (CPT-Berm) and toe of the dam (CPT-Toe). The first column in each row shows the nearest BH data to the CPTu, while the manual interpretation of the CPTu is shown in the second column. The third column shows the DBSCAN clustering results when using $eps = 0.08$ and $minPts = 10$. The final column shows the HDBSCAN clustering results when using $mcs = 10$. Note that the soil layers are colored similarly in each borehole and CPTu interpretation. For the borehole and manual interpretations, these colors represent distinct soil types as shown in Figure 1. For the DBSCAN and HDBSCAN results, these colors do not necessarily represent the same soil types, as the clustering algorithms do not consider soil type/SBTn classification during clustering. Instead, similar colors are used to demonstrate the similarity between the layers resulting from clustering and the layers determined by manual interpretation. Additionally, outliers are plotted in white.

Across the CPTu soundings used here, the results show that both clustering methods provide good results. DBSCAN is sometimes able to capture thinner layers (such as the uppermost soil layer in the bottom row), but at the cost of increased outliers. HDBSCAN results in fewer outliers but tends to miss the thinner layers. Therefore, while both methods show promise as alternatives to traditional CPTu interpretation, HDBSCAN has the advantage of being more intuitive in its choice of parameter.

5 Grouped CPTu clustering

Although both clustering methods show promise in providing individual CPTu interpretations, this method still requires some level of interpolation to create a 2D or 3D ground model. Thus, this section illustrates the application of DBSCAN and HDBSCAN algorithms to cluster grouped data to directly create a 2D ground model. Both algorithms are applied to 24 CPTu soundings located along the dam's crest with the aim of evaluating how the methods perform when using a larger and spatially-dependent data set. The results are then compared with the ground model that was manually defined in Rola et al. (2024).

An analogous approach as per the individual CPTu sounding clustering is followed. The same three variables are used: elevation, $\log(Q_{tn})$ and $\log(F_r)$; they are normalized between 0 and 1, and then grouped

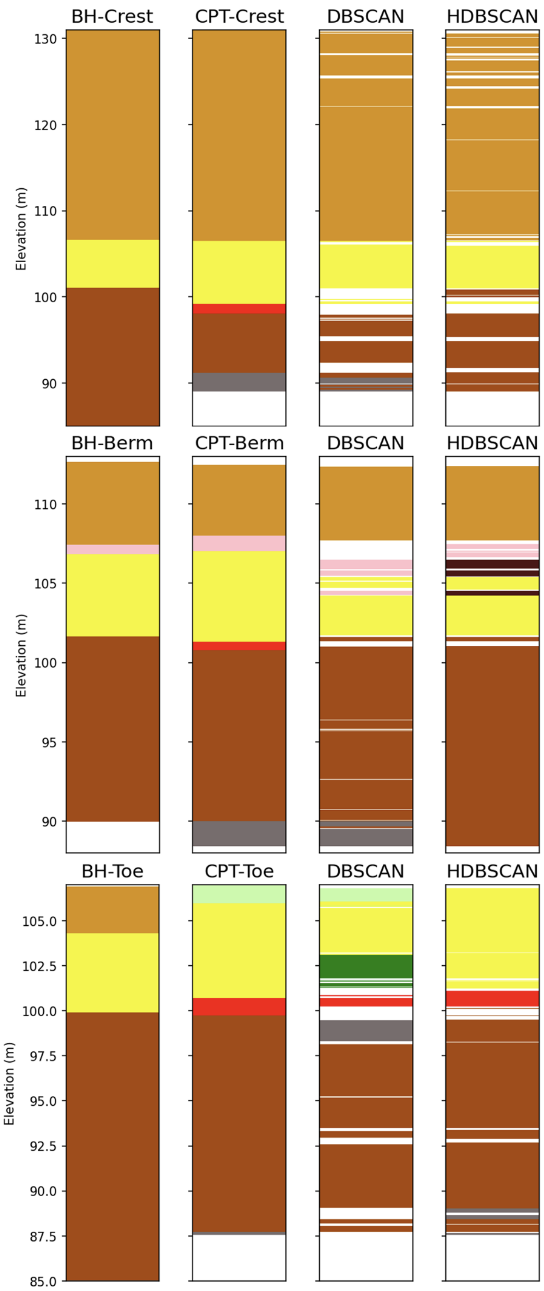

Figure 6. DBSCAN and HDBSCAN are used to cluster three CPTu soundings: CPT-Crest (top), CPT-Berm (middle), and CPT-Toe (bottom). The nearest borehole to each CPTu is shown in the first column and the manual interpretation of each CPTu is shown in the second column. The DBSCAN results using $eps = 0.08$ and $minPts = 10$ are shown in the third column. The HDBSCAN results using $mcs = 10$ are shown in the fourth column. These results show that DBSCAN is able to cluster thinner layers compared to HDBSCAN, but at the cost of more noise.

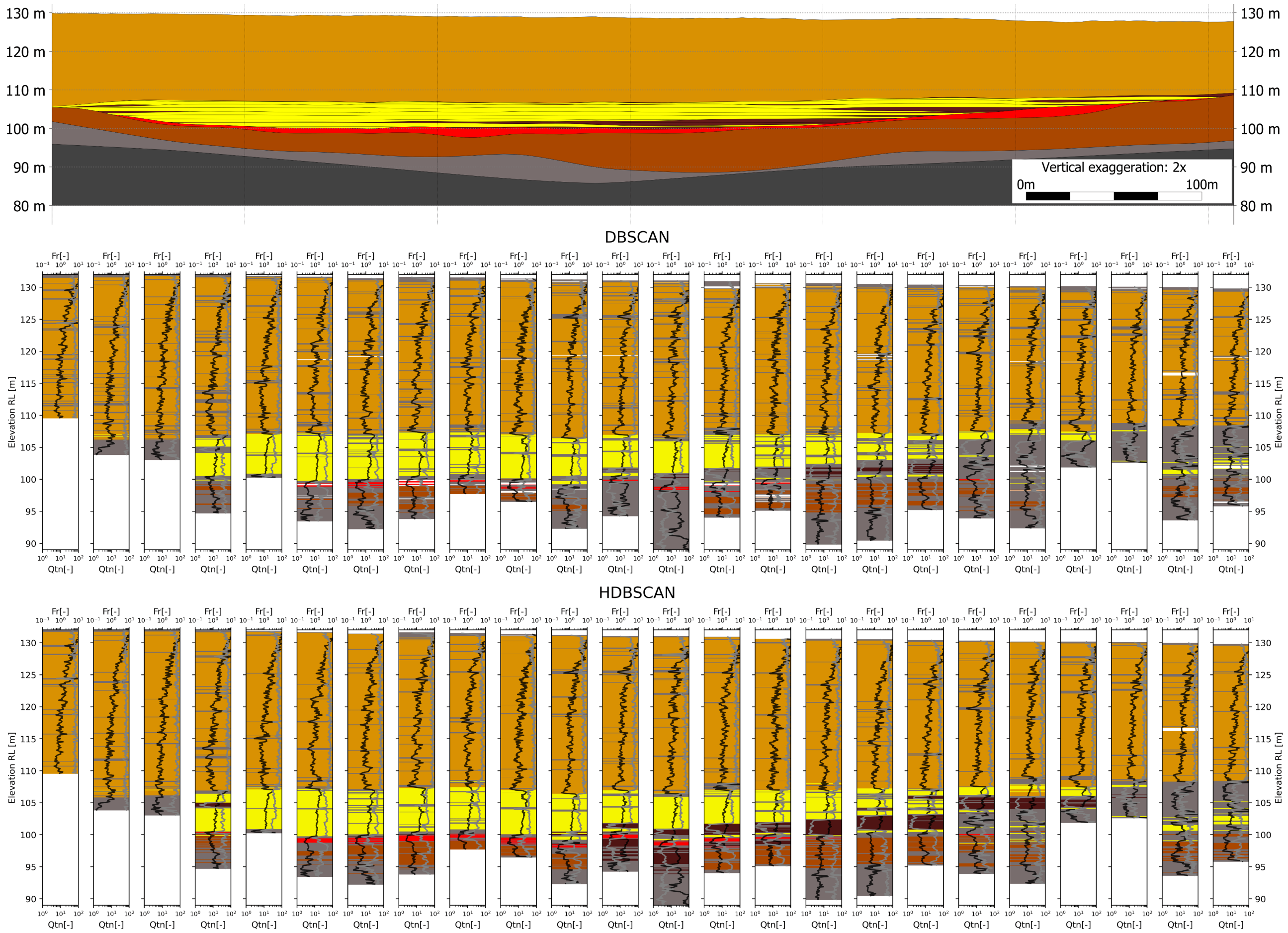

Figure 7. Grouped clustering for CPTu soundings located along the dam's crest. Comparison of the resulting longitudinal cross-section between: a) the manually-defined ground model (i.e., "true interpretation"), b) DBSCAN clustering with *eps* = 0.02 and *minPts* = 10 and c) HDBSCAN clustering with *mcs* = 20. The parameters Q_{tn} and F_r are plotted for each CPTu sounding to help identify cluster trends.

for all the CPTu soundings located along the crest. The best clustering parameters are found by repeating the clustering process until a good balance of clustered versus outlier points is reached, along with reasonable outputs on the soil profiles compared to the "true" interpretations. For the DBSCAN case, the best outcomes are obtained using *eps* = 0.02 and *minPts* = 10, giving 74% clustered points, 26% outliers and 10 clusters. For the HDBSCAN, the best outcome is obtained with *mcs* = 20, giving 83% clustered points, 17% outliers and 5 clusters.

Figure 7 shows a comparison of the manually-defined ground model (top) to the grouped clustering using DBSCAN (middle) and HDBSCAN (bottom). Overall, it is observed that both methods are very good at grouping the main geotechnical units at the dam's crest. For the case of DBSCAN, it is clear that it has some difficulties recognizing the Organics layer and further puts the coarse Tailings and the Lower Alluvium units within the same group (e.g., soundings from the center to the right area). Additionally, more iterations are required to find an appropriate value for *eps* compared to the individual clustering analysis. On the other hand, HDBSCAN has an excellent clustering capability, being able to detect the thin Organic layer and a clear separation between the Upper and Lower Alluvium units. Furthermore, the optimal clustering is achieved using *mcs* = 20, which is easily found by scaling the *mcs* used during individual clustering to this larger dataset.

6 Conclusions

Building a ground model is a crucial step for performing geotechnical engineering analyses. However, creating a ground model is typically time-consuming and difficult, as it requires an experienced engineer to review large amounts of in-situ data. Additionally, these interpretations are highly subjective, leading to uncertainty in the final ground model. Thus, alternative methods, such as machine learning clustering algorithms, are becoming increasingly popular due to their efficiency in providing ground models.

This paper illustrated the ground model development of a real dam by clustering CPTu data using unsupervised machine learning methodologies. The grouping capabilities of two algorithms were assessed: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and an extended version with a hierarchical component (HDBSCAN).

First, the clustering was done for individual CPTu soundings located at the dam's crest, berm and toe. A sensitivity analysis was performed on the clustering parameters to find the best values that reflected the "true" interpretations; these correspond to a manual layer definition using CPTu data and nearby borehole logs, as presented in Rola et al. (2024). For DBSCAN, it was found that the outcomes are highly sensitive to the user-defined parameters: small values of *eps* combined with low values of *minPts* yield too many clusters, and when using larger values of *minPts*, can result in too many outliers. After many iterations, the optimal combination was found to be $eps = 0.08$ and $minPts = 10$. For the case of HDBSCAN, results were much less sensitive to the *mcs* values, and few iterations were required to find optimal values in the range of 10 to 20. When using these clustering parameters, both DBSCAN and HDBSCAN performed well and were able to provide quick interpretations, although HDBSCAN is more user-friendly due to its intuitive parameter *mcs*.

Subsequently, an analogous approach was followed using grouped CPTu data from soundings located along the crest, and the results were compared with the manually-defined ground model. Overall, both methods showed very good grouping capabilities. However, DBSCAN had difficulties recognizing the Organics layer and put coarse Tailings and the Lower Alluvium units within the same group; moreover, a much lower value of *epsilon* was needed to reproduce the layering observed in the manually-defined ground model compared to the *eps* used during individual clustering. On the other hand, HDBSCAN had an excellent clustering capability, being able to detect the thin Organic layer and a clear separation between the upper and lower alluvium units; moreover, the optimal clustering was achieved using $mcs = 20$, which was easily found by scaling the *mcs* used during individual clustering to this larger dataset.

Acknowledgments

The authors are grateful for the support provided by SRK Consulting Argentina.

References

Brinkgreve, R.B.J, Tschuchnigg, A., Laera, S. and Brasile, S. 2023. *Automated CPT interpretation and modelling in a BIM/Digital Twin environment.* 10th European Conference on Numerical Methods in Geotechnical Engineering. London. http://doi.org/10.53243/NUMGE2023-111

Campello, R.J.G.B., Moulavi, D., Sander, J. 2013. *Density-Based Clustering Based on Hierarchical Density Estimates.* Advances in Knowledge Discovery and Data Mining. PAKDD 2013. Lecture Notes in Computer Science(), vol 7819. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37456-2_14

Campello, R.J.G.B., Moulavi, D., Zimek, A., Sander, J. 2015. *Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection.* ACM Trans. Knowl. Discov. Data 10, 1, Article 5 (July 2015), 51 pages. https://doi.org/10.1145/2733381

Collico, S., Arroyo, M., Devicenzi, M. 2024. *A simple approach to probabilistic CPTu-based geotechnical stratigraphic profiling.* Computers and Geotechnics. https://doi.org/10.1016/j.compgeo.2023.105905

Ester, M., Kriegel, H., Sander, J., Xu, X. 1996. *A density-based algorithm for discovering clusters in large spatial databases with noise.* Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96). AAAI Press, 226–231.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). *Scikit-learn: Machine learning in Python.* The Journal of Machine Learning Research, 12, 2825-2830. https://doi.org/10.1145/2786984.2786995

Robertson, P. K. (2009). *Interpretation of cone penetration tests—a unified approach.* Canadian Geotechnical Journal, 46(11), 1337-1355. https://doi.org/10.1139/T09-065

Rola, J., Sottile, M.G., Rivas, N.A., Roldan, L., Sfriso, A. *Development of a 3D ground model to design the stabilisation of a dam founded on weak liquefiable ground.* 7 International Conference on Geotechnical and Geophysical Site Characterization. Barcelona.

Schubert, E., Sander, J., Ester, M., Kriegel, H., Xu, X. 2017. *Why and How You Should (Still) Use DBSCAN.*. ACM Trans. Database Syst. 42, 3, Article 19. https://doi.org/10.1145/3068335