Stochastic Environmental Research and Risk Assessment (2025) 39:423–435
https://doi.org/10.1007/s00477-024-02861-6

ORIGINAL PAPER

Predicting saturated hydraulic conductivity from particle size distributions using machine learning

Valerie de Rijk[1] · Jelle Buma[2] · Hans Veldkamp[2] · Alraune Zech[1]

Accepted: 5 November 2024 / Published online: 19 January 2025

Abstract
Estimating saturated hydraulic conductivity K_f from particle size distributions (PSD) is very common with empirical formulas, while the use of machine learning for that purpose is not yet widely established. We evaluate the predictive power of six machine learning algorithms, including tree-based, regression-based and network-based methods in estimating K_f from the PSD solely. We use a dataset of 4600 samples from the shallow Dutch subsurface for training and testing. The extensive dataset provides not only PSD, but also measured conductivities from permeameter tests. Besides training and testing on the entire data set, we apply the six algorithms to data subsets for the soil types sand, silt and clay. We further test different feature/target-variable combinations such as reducing the input to PSD-derived grain diameters d_{10}, d_{50} and d_{60} or estimating porosity from PSD. We test feature importance and compare results to K_f estimates from a selection of empirical formulas. We find that all algorithm can estimate K_f from PSD at high accuracy (up to R^2/NSE of 0.89 for testing data and 0.98 for the entire data set) and outperform empirical formulas. Particularly, tree-based algorithms are well suited and robust. Reducing information in the feature variables to grain diameters works well for predicting K_f of sandy samples, but is less robust for silt and clay rich samples. d_{10} also shows to be the most influential feature here. An interesting, but not surprising outcome is that PSD is not a suitable predictor for porosity. Overall, our results confirm that machine learning algorithms are a powerful tool for determining K_f from PSD. This is promising for applications to e.g. deep-drilling data sets or low-effort and robust K_f-estimation of single samples.

Keywords Saturated hydraulic conductivity · Particle size distribution · Machine learning

1 Introduction

Saturated hydraulic conductivity (K_f, also known as K_{sat} or K_s) is the key parameter in groundwater flow and subsurface transport processes (Bear 1972). It determines flow velocities, and thus arrival times and distribution patterns of contaminants in soil and sediments. At shallow and medium depths, hydraulic conductivity impacts groundwater recharge and the protection of freshwater aquifers by overlying aquitards. In deeper systems, it is important for the characterization of aquifers with regard to geothermal exploration (Veldkamp et al. 2021). K_f depends on the fluid properties as well as the texture and structure of the porous medium which makes it subject to intrinsic spatial heterogeneity.

Measuring hydraulic conductivity directly in the field is timely and expensive, if possible at all. Pumping tests provide hydraulic conductivity, but only averages of up to several cubic kilometers of porous material. They are furthermore cost and labour intensive. Other methods, such as lab-based permeameter tests, require undisturbed samples. Most direct methods to determine K_f are not well suited for deep aquifers, where K_f is typically estimated from correlations to porosity or from particle size distributions.

✉ Alraune Zech
a.zech@uu.nl

Valerie de Rijk
v.derijk@uu.nl

Jelle Buma
jelle.buma@tno.nl

Hans Veldkamp
hans.veldkamp@tno.nl

[1] Department Earth Science, Utrecht University, Princetonlaan 8a, 3584CB Utrecht, Netherlands

[2] TNO, Princetonlaan 6, 3584CB Utrecht, Netherlands

The particle size distribution (PSD) is the size-sorted list of particle diameters present in a porous material sample. The PSD is classically obtained through sieve analysis, where the sample is shifted through a series of sieves with decreasing gap size while the amount of passing material is weighed. Novel techniques under use are laser diffraction which is advantageous in workload, required sample size and number of sieve sizes determined compared to classical sieving.

Traditionally, PSD information is translated to hydraulic conductivity through empirical formulas relating effective grain size diameters and total porosity to K_f. The first relation was developed by Hazen in 1892. Many more followed, e.g. Devlin (2015) compiled a spreadsheet program which calculates K_f from PSD curves using 15 different methods. However, the procedure is error prone as each empirical method was developed for and/or calibrated to a specific type of material. Multiple studies showed that the empirical relations are of low accuracy e.g. Vienken and Dietrich (2011), Cabalar and Akbulut (2016), Chandel and Shankar (2022).

In recent years, Machine Learning (ML) has been increasingly applied to geoscientific research (Tahmasebi et al. 2020). While the application of ML has become popular for identifying permeability (translatable to hydraulic conductivity) from image processing, we put it to use for physical parameter estimation from a numerical dataset of soil properties. One of the first applications of ML in this context was the case of the software Rosetta (Schaap et al. 2001) focusing on unsaturated soils where other (structural/textural) parameters impact flow behaviour compared to saturated soils. Studies, such as Jorda et al. (2015), Araya and Ghezzehei (2019), Kotlar et al. (2019), Sihag et al. (2019) showed excellent predictability of saturated and near-saturated hydraulic conductivity with ML through the use of a wide range of input parameters, such as textural properties of the solid matrix, like sand, clay and silt content, and structural properties like land use and bulk density. However, extensive textural and structural information is not always available, particularly for deeper soils. It remains unknown how good ML works with only textural information, such as the PSD of a sample.

Various algorithms have been applied to predict (saturated) hydraulic conductivity, such as support vector regression (Mady and Shein 2018), neural network algorithms (Williams and Ojuri 2021), tree-based regression (Granata et al. 2022; van Leer et al. 2023) or boosted regression (Jorda et al. 2015). Most research train one (Williams and Ojuri 2021; van Leer et al. 2023; Jorda et al. 2015) or explore two types of algorithms (Araya and Ghezzehei 2019; Granata et al. 2022). However, which type of algorithm is generally best suited remained unstudied.

We address these gaps by studying the following research questions: (i) Can we predict saturated hydraulic conductivity with PSD data only? (ii) Which ML algorithms are best suited? (iii) How do results depend on the data set size and composition? And (iv) how do results compare to empirical formulas?

Our general approach is to apply the PSD as set of feature variables to six machine learning algorithms for predicting saturated hydraulic conductivity as target variable. The tested algorithms are: (i) Decision tree (DT), (ii) Random forest (RF), (iii) Linear Regression (LR), (iv) Support vector regression (SVR), (v) XGboost (XG) and (vi) Artificial neural network (ANN). The first five algorithms are ML based while ANN is a deep learning algorithm. We compare the ML estimate performance against five selected empirical formulas. Additionally, we evaluate the effectiveness of using PSD-derived variables: particle diameters d_{10}, d_{50} and d_{60} (which represent the maximum particle diameter of 10%, 50% and 60% material passing) as predictors for K_f to possibly minimize measurement requirements. Finally, we assess if porosity can be predicted through PSD measurements.

We train all ML algorithms on an extensive dataset with almost 4600 samples, originating from the *TopIntegraal* drilling and sampling program of the shallow ($30-50$ m depth) Dutch subsurface (Buma et al. 2024). We compare algorithm performance for the entire data set as well as for soil-type data subsets. Most studies investigating the application of ML for predicting K_f of aquifers (rather than of soil) have a few hundred samples available, either from a specific location or a collection of literature data. In contrast, we can gain novel insights by using this large, consistent data set which originates from many locations all over the Netherlands while all samples have been analyzed identically, thus guaranteeing consistency.

2 Methodology

2.1 Data

The dataset under study originates from the *TopIntegraal* Program (Buma et al. 2024). As the program is ongoing with data being added on availability, we used version 1.0 of the dataset, dated April 29th 2024. The published version can be downloaded from the groundwater portal of TNO Geological Survey of the Netherlands, but does not contain the associated PSD data. We make use of 4593 out of the 4621 samples from the dataset which have PSD data available. This dataset can be found in the Github Repository (Zech 2024). All samples were taken from the shallow subsurface (< 50 m) over a 15 year period and cover about 60% of the Netherlands. Although they stem from a variety of

Table 1 Lithoclasses, sorted by the categorization into main lithology

Z - sand	L - silt	K - clay
Z_{s1} - sand, weak silty	L_{z1} - loam, weak sandy	K_{s1} - clay, weak silty
Z_{s2} - sand, moderate silty	L_{z3} - loam, strong sandy	K_{s2} - clay, moderate silty
Z_{s3} - sand, strong silty	K_{s4} - clay, extreme silty	K_{s3} - clay, strong silty
Z_{s4} - sand, extreme silty		K_{z1} - clay, weak sandy
Z_k - sand, clayey		K_{z2} - clay, moderate sandy
		K_{z3} - clay, extreme sandy
		p - peat

lithostratigraphical units, the majority of samples belong to the sandy category due to the nature of Dutch soil.

The PSDs were obtained through the *malvern* laser diffraction method following the protocol of Baars (2004). The gravel fraction ($>$ 2mm) was excluded. The samples were undisturbed and non-mixed on arrival at the laboratory. The identification of the particle size distribution was performed after removal of the organic fraction and of the carbonate fraction. Distribution of particle sizes were determined for 35 sieve sizes ranging from $0.01 - 0.1$ µm to $1680 - 2000$ µm.

Each sample is classified for its lithoclass according to the NEN5104 standard for Dutch soils (NEN2019 2019). The NEN5104 classification is based on PSD and is completely determined by the weight of the sand, silt, clay and organic matter content. All lithoclasses are listed in Table 1. The function we used can be found in the collection of python-scripts at Github.

We assign all samples to one of three main soil types based on their lithoclass: sand *Z*, silt *L* or clay *K*. Table 1 specifies the distribution of lithoclasses to the categories. These categories form three sub-data sets *Top-sand*, *Top-Silt* and *Top-Clay*. The latter one also contains the 158 peat samples for which PSD data could be measured. They have high amounts of organic matter (more then 50% for 80 out of 158 samples) which is more determining for K_f than sand-lutum-silt percentages.

Saturated hydraulic conductivity K_f of non-cohesive samples was measured with an Eijkelkamp permeameter conform to European Standards (CEN ISO/TS 17892-11). K_f measurements were assessed on their quality and insufficient samples were filtered out. Details of the measurement process and data quality checks are provided in the *Supporting Information (SI)*. Further information on the *TopIntegraal* data set can be found in van Leer et al. (2023) and Buma et al. (2024).

Table 2 provides the descriptive statistics of the dataset for the key information we use. This contains the distribution of log_{10}-transformed measured conductivity K_f, statistics on sand, silt and clay content of the samples as well as d_{10} and d_{50}. The mean and median value of sand in Table 2 reflect the dominance of sandy samples in the dataset. Consequently, the median particle size d_{50} is relatively large and the mean $log_{10}\ K_f$ is high. Note the considerable spread of hydraulic conductivities (over 8 orders of magnitude) with values from $10^{-6.7}$ [m/d] to 10^2 [m/d].

Descriptive statistics for the subsets on sand (72.4% of all samples), clay (17%) and silt (10.6%) are provided in Tables S2, S3, and S4 in the *SI* as these subsets are also used for analysis with ML algorithms.

2.2 Empirical formulas for comparison

A common way to estimate hydraulic conductivities is by relating its value to PSD-derived quantities (such as d_{10}) through empirical relationships, see e.g. the summary of 15 methods in Devlin (2015).

Following Vukovic and Soro (1992), most methods can be written in the general form $K_f = \frac{\rho g}{\mu} \cdot N \cdot \phi(\theta) \cdot d_{\text{eff}}^2$ where $\phi(\theta)$ is a function of the porosity θ, d_{eff} is a form of effective grain diameter, and N is a method-specific constant. The factor $\frac{\rho g}{\mu}$ entails the water density ρ, the gravitational constant g, and the dynamic viscosity μ. The choices of ϕ, d_{eff} and N are method specific. Most empirical formulas come with an application limit, being only applicable to a certain range of effective grain diameters.

We make use of five empirical methods for comparison to the performance of the six ML algorithms: (i) *Barr* (Barr 2001), (ii) *Alyamani & Sen* (Alyamani and Şen 1993) [both as defined in Devlin (2015)], (iii) *Shepherd* (Shepherd 1989), (iv) *Kozeny* (Kozeny 1927), and (v) *Van Baaren* (Van Baaren 1979). The selection is based on their unlimited applicability. Specifics on equations and implementations are provided in the *SI*. Note that the empirical algorithms were typically developed for clean sand. Thus, poorer

Table 2 Descriptive statistics of entire dataset (4593 samples)

	$log_{10}K_f$ (m/d)	d_{10} (µm)	d_{50} (µm)	Clay (%)	Silt (%)	Sand (%)
Min	−6.66	0.4	2.7	0	0	0
Max	2.28	680	1059	85.2	82.3	100
Median	0.3	93	181	1.3	3.34	95.34
Mean	−0.63	102	202	7.37	15.9	76.8
Standard deviation	2	91	157	12.3	21.9	32.3

performance for clayey soils can be expected for methods without application limit.

2.3 ML model workflow

2.3.1 Algorithms and hyperparameters

We use six well established methods in machine and deep learning: three tree-based, two regression-based and one layered deep learning algorithm. The choice was made based on (i) established popularity (Sarker 2021), (ii) easiness to implement (using Python packages), (iii) and structural differences. The combination of the six selected techniques allows exploring trade-offs between complexity of the algorithms, training time and accuracy.

Used algorithms are shortly outlined with their specifics and the algorithm-specific hyperparameters which we tune during the training process. For detailed information on the principles of the algorithms, the reader is referred to literature (Hastie et al. 2009; Russell and Norvig 2020; Tahmasebi et al. 2020). All models are set up using the `scikit-learn` library in *Python* (Pedregosa et al. 2011). Selected hyperparameters are chosen from availability in `scikit-learn` and expected influence on the outcome. For tuning processes, the popular 10-times cross-validated grid search technique is used.

Decision tree (DT) is structured like a tree, in which branches and leaves arise from nodes. The algorithm partitions the dataset recursively based on the most significant features. The algorithm continues to split the data into groups till the target variable is sufficiently predicted to an error margin as defined by the user. Without predefining the end node, this algorithm is prone to overfitting. Consequently, the hyperparameters we tune are the maximum tree depth (*max-depth*) and the minimum number of samples in each split group (*min-samples-split*).

Random forest (RF) is an ensemble learning algorithm that operates on constructing multiple DTs. A multitude of DTs is drawn randomly by bootstrapping (Breiman 2001) where each DT selects a range of feature variables. The predicted outcome of the target variable is constructed by averaging over all trees and evaluating the process through aggregation. RF is less prone to overfitting than DT, but requires a larger training dataset and is subject to higher computation time. Tuned hyperparameters are the same as those of DT (*max-depth* and *min-samples-split*) as well as the number of estimators (*N-estimators*).

XGboost (XG) is a gradient boosting algorithm working on the ensemble technique like RF. It improves the performance of a predictive model iteratively by adding weak learners to the ensemble. Boosting is a technique that combines multiple weak learners to form a strong learner. In the case of XGBoost the weak learners are decision trees, and each tree is trained to correct the errors made by the previous tree. This process continues until the desired level of accuracy is achieved. To optimize the loss function, XGBoost uses a combination of gradient descent and second-order derivatives. This enables the algorithm to efficiently handle high-dimensional data and complex relationships between features. In addition to its efficient optimization techniques, XGBoost also uses regularization to prevent overfitting such as the Ridge expression (Chen and Guestrin 2016). This involves adding penalties to the loss function for complex models, ensuring that the algorithm produces a more generalizable model that performs well on unseen data. The downside of the algorithm is the huge amount of hyperparameters (35 in the `scikit-learn` implementation), entailing that model fitting is time and computer intensive. We limit hyperparameter tuning to the maximum tree depth *max-depth* and the learning rate while leaving the remaining hyperparameters on default values.

Linear regression (LR) is a classical mathematical method where the input variables are linearly fitted to the output variable by minimizing a cost function characterizing the difference between the linearly transformed input and the feature variables. Multiple ways exist to fit the optimization coefficients to the data. We use a cost function with a Ridge term to reduce over-fitting and to obtain the optimal cost function. Consequently, we tune the penalty term reg-α in the Ridge regression as hyperparameter. The higher its value, the higher the penalty given for increasing complexity in the algorithm.

Support vector regression (SVR) is based on the Support Vector Machine principle, which trains by finding a hyperplane that separates the training data and assigns new data points to a class based on their position within the grid of a linear problem. To optimize prediction accuracy, the algorithm fits the error within a certain margin, either by a hard or soft margin that allows some misclassification. We tune the optimization parameter C. It compromises between correct classification of a sample against the maximization of the decision function's margin. For larger values of C, smaller margins will be preferred if this leads to better predictions. A lower C promotes a larger margin at the cost of the training efficiency.

SVR can be trained on a non-linear dataset by introducing kernel functions that transform the data into a higher dimensional space. We preliminary tested three kernels: linear, polynomial, and radial. We then focused on SVR with radial basis function $K(x, y) = e^{(-\gamma|x-y|^2)}$. We tune the hyperparameter γ which is the inverse of the radius that the algorithm utilizes to select samples as support vectors. Thus, low γ values use a wider range of support vectors and vice versa.

Artificial neural network (ANN) mimics the workings of the human brain and therefore belongs to the subsection of Deep Learning. ANN is structured into layers, including an input, output and a number of hidden layers. Each layer consists of a number of neurons, where number of neurons in the input and output layers refer to the number of input and output parameters. The number of hidden layers and the number of neurons per hidden layer are hyperparameters. Each neuron represents a parameterized and bounded function, which receives input data and subsequently produces output data to pass on (Tahmasebi et al. 2020). The algorithm is trained through back-propagation, where the derivative of the loss function (based on the mean square error between output values and target variables) is updated for all neurons in the network. It is then used to update the weight and bias according to the gradient descent technique.

We tested the number of hidden layers (up to three) and tuned the number of neurons (*hidden-layer-size*). We further tune the *Activation* which determines how the weighted sum of the input is transformed into output. We either use *relu* (Rectified Linear Unit) which transforms every negative input value into the value of zero, *logistic* which returns the logistic sigmoid functios $f(x) = \frac{1}{1+\exp(-x)}$, or *tanh* which returns the hyperbolic tangent of the input value, resulting in values between + 1 and − 1 (Pedregosa et al. 2011). We tune the *learning rate* which determines the size of the change that is made during each backpropagation. It is either constant or adaptive. The latter adjusts the learning rate based on the success of the model. The tested ranges for all hyperparameters and all algorithms are summarized in Table S6 in the *SI*.

2.3.2 Standard model application

Standard input features for the algorithms are the PSD values measured in 35 sieve sizes in cumulative form. The target variable for training and testing is the $\log_{10}$-transformed value of the saturated hydraulic conductivity K_f. 4593 samples of the *TopIntegraal* data set are included.

Each algorithm is applied with these features on: the entire data set *Top-All* and on three subsets based on the soil type: *Top-Sand*, *Top-Clay* and *Top-Silt*. Soil type classification is according to the NEN 5104 standard for Dutch soils (NEN2019 2019) as outlined in Sect. 2.1. Descriptive statistics of the subsets are provided in Tables S2, S3, and S4 in the *SI*. Each dataset is trained for the six algorithms.

The analysis is conducted with the Python package `scikit-learn` (Pedregosa et al. 2011). Python workflows are available at Github (Zech 2024). We used the common 80–20 ratio for split into training data and testing data. Preliminary tests showed that the initial split does not impact results. For ANN and SVR, all input variables are standardized. The full ML application workflow is explained in more detail in Sect. 2 of the *SI* including a graphical display of the workflow and the results of the hyperparameter tuning.

2.3.3 Additional strategies of data evaluation

We trained the six algorithms on other combinations of feature and target variables to test predictability of conductivity from PSD derived quantities and links to porosity. We started from the full data set (*Top-All*) and calculated d_{10}, d_{50} and d_{60} from the PSD. Furthermore, we filtered the data set to samples where porosity measurements were available resulting in a reduced data set *Top-Por* with 1768 samples. All samples of *Top-Por* belong to the sand soil class. Statistics for the *Top-Por* data set are available in Table S5 in the *SI*.

From the derived information and filtered data set, we came up with three additional application strategies, based on the feature and target variable combinations:

- Predict K_f (target) from d_{10}, d_{50}, d_{60} (features) for the *Top-All* dataset.
- Predict K_f (target) from d_{10}, d_{50}, d_{60} and porosity θ (features) applied to the reduced *Top-Por* dataset.
- Predict porosity θ (target) from PSD (features) using the *Top-Por* dataset.To allow performance comparison for the different feature/target combinations on the reduced *Top-Por* dataset, we repeated the training and testing of the standard target/feature combination (K_f from PSD) for the reduced *Top-Por* data set. Technical details and hyperparameter tuning for all additional ML applications are again provided in the *SI*.

2.3.4 Model evaluation

Our standard algorithm performance measure is the Nash-Sutcliffe model efficiency (NSE):

$$NSE(\mathrm{z},\mathrm{y}) = 1 - \frac{\sum_{i=1}^{n}(z_i - y_i)^2}{\sum_{i=1}^{n}(z_i - \bar{z})^2}, \tag{1}$$

where $\bar{z}$ is the average of z, which we consider as sample value vector while y is the model output vector.
The NSE relates the variability explained by the model to the total variability in the sample values. It thus measures the percentage of variability within the z values that can be explained by y. A value close to one indicates a useful model while a value close to or below zero indicates that the model is not well suited. Note that the NSE reflects the coefficient of determination (R^2) for statistical models, i.e. for the model performance during the training phase.

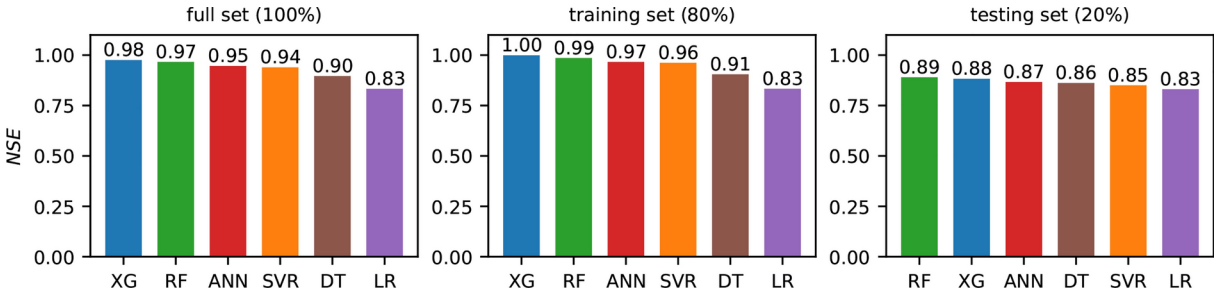

Fig. 1 Performance measure NSE of all six algorithms (in colored bars) for full set of samples (100%), the training data (80%), the testing data (20%) for the standard features (K_f from PSD) applied to the data set *Top-All*

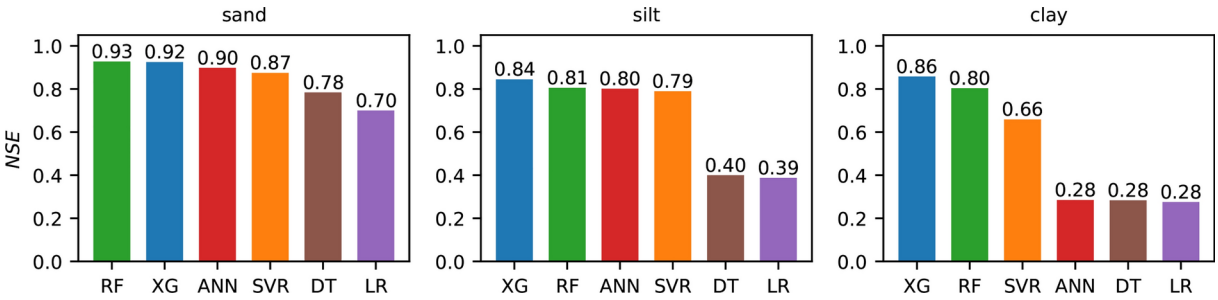

Fig. 2 Performance measure NSE of all six algorithms (in colored bars) for the data subsets on soil types *Top-Sand* (3326 samples), *Top-Silt* (485 samples) and *Top-Clay* (782 samples) for the standard features applied to all samples

3 Results and interpretation

3.1 Performance for standard features

Figure 1 shows a comparison of the performance, in terms of the measure NSE, of all six algorithms for the *Top-All* dataset. We show results for the training data (80%), the testing data (20%) and all samples (100%). Figure 2 shows the NSE of all six algorithms applied to the three soil type subsets with the standard features. The performance refers to the entire data sets using training and test data combined. Accompanying figures on algorithm performance comparison for the soil type subsets on the training and test data set are provided in the *SI*.

All algorithms show good performance for the entire dataset with high NSE values above 0.83. ANN, RF and XG perform best. Splitting the dataset into soil categories decreases performance. This is particularly the case for DT and LR for silt and clay data where the number of samples is significantly lower. Surprising is also the comparably poor performance of ANN for the clay data set. We explain part of the performance reduction with the smaller size of the training data. However, this is not impacting all algorithms equally as RF and XG still show good performance on all soil class sub sets with NSE values above 0.8.

Figure 3 shows scatter plots of predicted against measured log-hydraulic conductivity for the *Top-All* dataset with complementary figures for the soil type subset provided in the *SI*. Estimates for DT show clear discrete tree classes. The best performance of XG and RF in terms of the NSE (Fig. 1) is clearly supported by narrow distribution of the scatter around the 1:1 line. Also confidence intervals, represented by the $5th$ and $95th$ percentiles, of predictions are very narrow.

All algorithms tend to overestimate low conductivities and slightly underestimate high values. This lack in predicting the extreme values is related to the fact that algorithms are based on interpolation. When there are not enough extreme values in the training data set, these values are not predicted for the test data set. XG, SVR and ANN show the least variation around low conductivities. The density of (measured) samples is lower in the range of $K_f \in [10^{-1}, 10^{-3}]m/d$. This coincides with less accurate predictions visible also in broader error bands ($5th$ and $95th$ percentiles). All algorithms have the least deviation for sandy samples, which we relate to the abundance of samples within this soil category. At the same time, it supports our hypothesis that conductivity of sandy samples can be very well predicted purely on PSD information.

3.1.1 Comparison to empirical formulas

When estimating conductivity with a selection of empirical formulas (Sect. 2.2), we see that they perform less well than ML algorithms. The NSE of the best performing empirical formula, the *Barr* method, is 0.83 which is the same as the

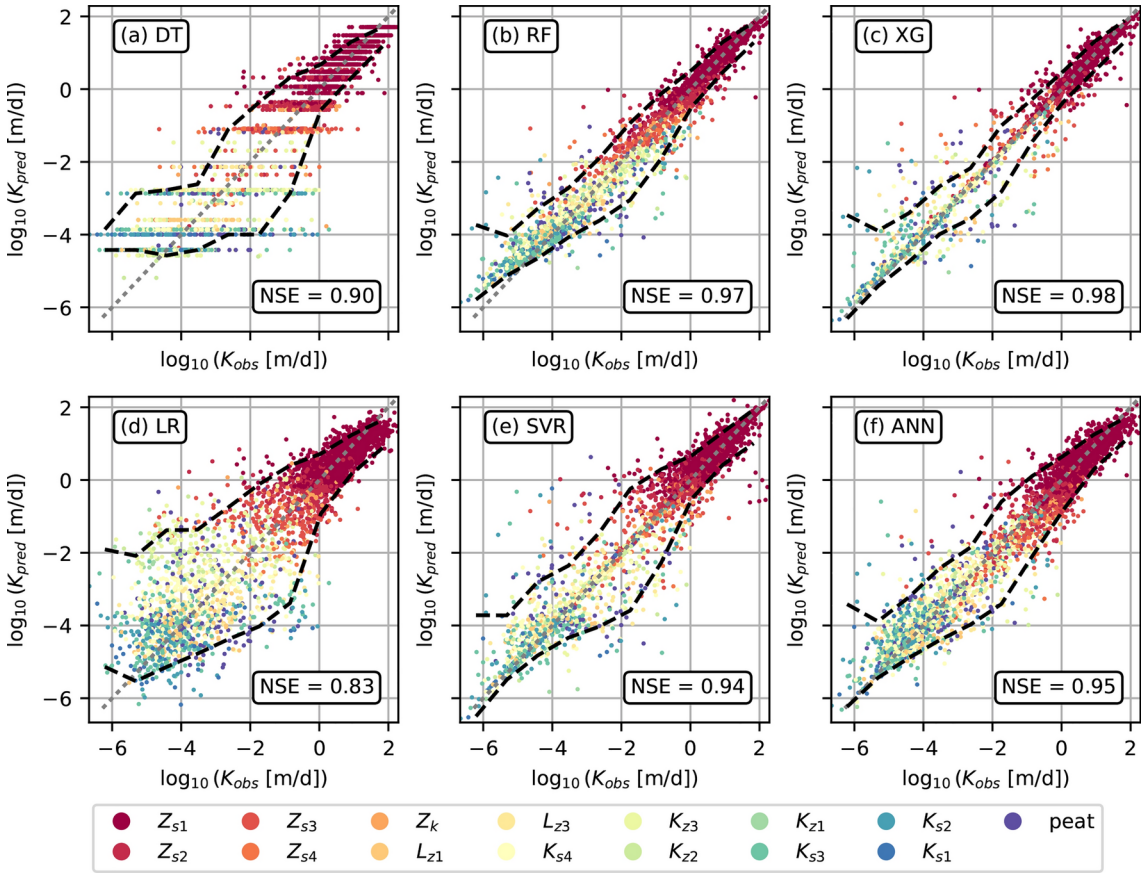

Fig. 3 Algorithm predicted log-conductivity (y-axis) versus measured data (x-axis) for all six algorithms applied to the full *Top-All* dataset. The gray line represents the 1:1 line while black lines represent the $5th$ and $95th$ percentiles of predictions. Colors indicate lithoclasses (Table 1)

NSE of LR being the worst performing ML algorithm with a NSE of 0.83 for the full data set.

A visual comparison of predicted K_f values for each sample versus measured values is displayed in Fig. 4a for the *Barr* method. The *Barr* formula generally overestimates the K_f measurements particularly at the lower end. A similar display of results for the other tested empirical formulas is provided in the *SI*. All formulas tend to either systematically over- or underestimate K_f-values.

Figure 4b shows the scatter of hydraulic conductivities predicted by the *Barr* formula (being the best performing empirical formula) against the K_f-values estimated with RF (being one of the best performing ML algorithms with a NSE of 0.97 for the fit to the measurements). Notably, the NSE of 0.89 for the fit of *Barr* K_f-values to RF estimates is higher than the one for the fit to the measured data. This indicates an overlap in values not being well predicted by both methods. These are predominantly samples from the silt and clay soil classes with high measured K_f-values but low estimates of K_f (lower right corner in Fig. 4a). This might indicate a poorer quality of the samples which will be subject of future research.

3.1.2 Feature importance

We tested the feature importance, i.e. how much each feature - here each sieve size - contributes to the fitted ML algorithm's statistical performance. We made use of the permutation importance algorithm of `scikit-learn` (Pedregosa et al. 2011). This inspection technique involves random shuffling of the values of a single feature to observe the degradation of the model's score. Figure 5 shows the importance mean and standard deviation of each feature for the RF algorithm applied to the *Top-All* data set. The higher the importance mean, the more influence the feature has.

Notably, the small sieve fractions between 2 and 25 µm have the highest impact while the larger sieve sizes contribute very little to the performance of RF. Thus, the fraction of the clay and small silt particles determine the conductivity for the data set predominantly containing sandy samples. Figure 5 might suggest that it does not really matter how coarse a sand is for K_f. However, in combination with Fig. 3 we rather link this to the fact that K_f of (medium) coarse sand is spread over less orders of magnitude. The picture changes for the data set filtered to silt and clay samples (SI,

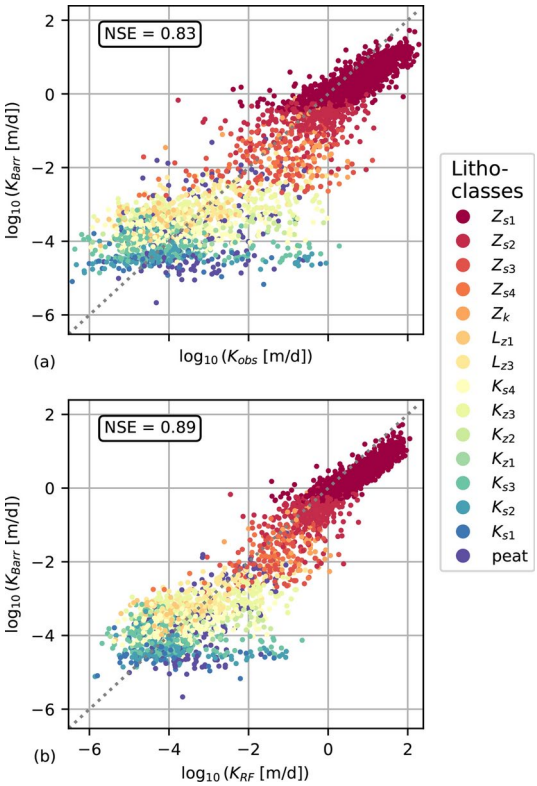

Fig. 4 Predicted log-conductivity of *Barr* formula (y-axis) versus measured data (top) and versus values predicted by RF (bottom) applied to the entire *Top-All* dataset. The gray line represents the 1:1 line. Colors indicate lithoclasses (Table 1)

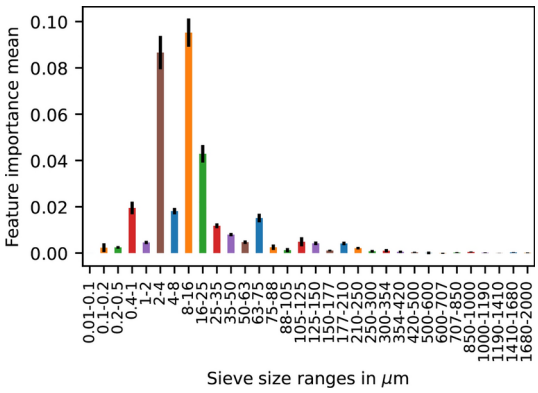

Fig. 5 Feature importance of each PSD sieve size, for the RF algorithm applied to the *Top-All* data set. The height of each bar indicates the mean importance of feature, black whiskers show standard deviation

Fig. S13), where there is no clear pattern about which sieve fractions dominate the RF algorithm accuracy.

The small sieve sizes (< 25 µm) show a high importance in all six tested algorithms (SI, Fig. S14). While DT and XG show similar patterns as RF (Fig. 5), LR, SVR and ANN take the higher sieve sizes stronger into account. The strong impact of small sieve sizes also supports the idea to test only effective grain diameters as features, where particularly d_{10} will be most significant for hydraulic conductivity prediction. The particle size range coincides relatively well with the arithmetic mean of d_{10} in Table 2, and even better with the geometric mean d_{10} of all samples, being 43 µm.

3.2 Performance for alternative feature-target combinations

3.2.1 PSD-derived variables on *Top-All*

Figure 6 shows the performance of all algorithms using only d_{10}, d_{50} and d_{60} as feature variables (instead of the entire PSD) for estimating hydraulic conductivity for the *Top-All* dataset. The tree based algorithms, DT and RF, as well as XG show high performance with NSE values above 0.9, almost as good as when using the entire PSD (Fig. 3), as is also visible in the similar scatter of predicted versus measured $log_{10}K_f$ values. The low NSE of 0.62 for LR is due to the poor prediction of low-conductivity values. Also, ANN and SVR have difficulties predicting very low K_f-values ($< 10^{-4}$m/d) resulting in lower NSEs then when using the entire PSD as feature variable. Note that the number of low K_f samples is much smaller than the number of sandy samples, thus the NSEs remain relatively high for all algorithms.

From the soil type perspective, Fig. 6 reveals that d_{10}, d_{50} and d_{60} are sufficient to predict K_f for sandy samples with any of the ML algorithms. Low K_f values for silty and clay samples are only well predicted by some of the algorithms.

3.2.2 Prediction for the *Top-Por* data set

We studied the impact of having information on porosity on the reduced data set *Top-Por* (1768 samples). Performance measures of the six algorithms for different tested feature-target combinations are compared in Fig. 7. Further visualizations of results are provided in the *SI*, including scatter plots and NSE values for performance on training and testing data.

We see that the algorithms perform very similar on the reduced data set for the standard feature-target combination. Comparing Fig. 1 (left) vs. Fig. 7 (center) shows only slightly lower NSE values for the *Top-Por* data set. The highest NSE reduction is for ANN, while LR remains the worst performing algorithm.

When comparing the algorithms' performances for estimating K_f from d_{10}, d_{50}, d_{60} and porosity θ, we see again that XG and RF outperform the other algorithms, while all perform fairly good with the lowest NSE of 0.66 for LR. Notably, all algorithms have relatively similar NSE for the testing data set with values between 0.57 and 0.71. While the NSE for the training data is only a little higher for ANN,

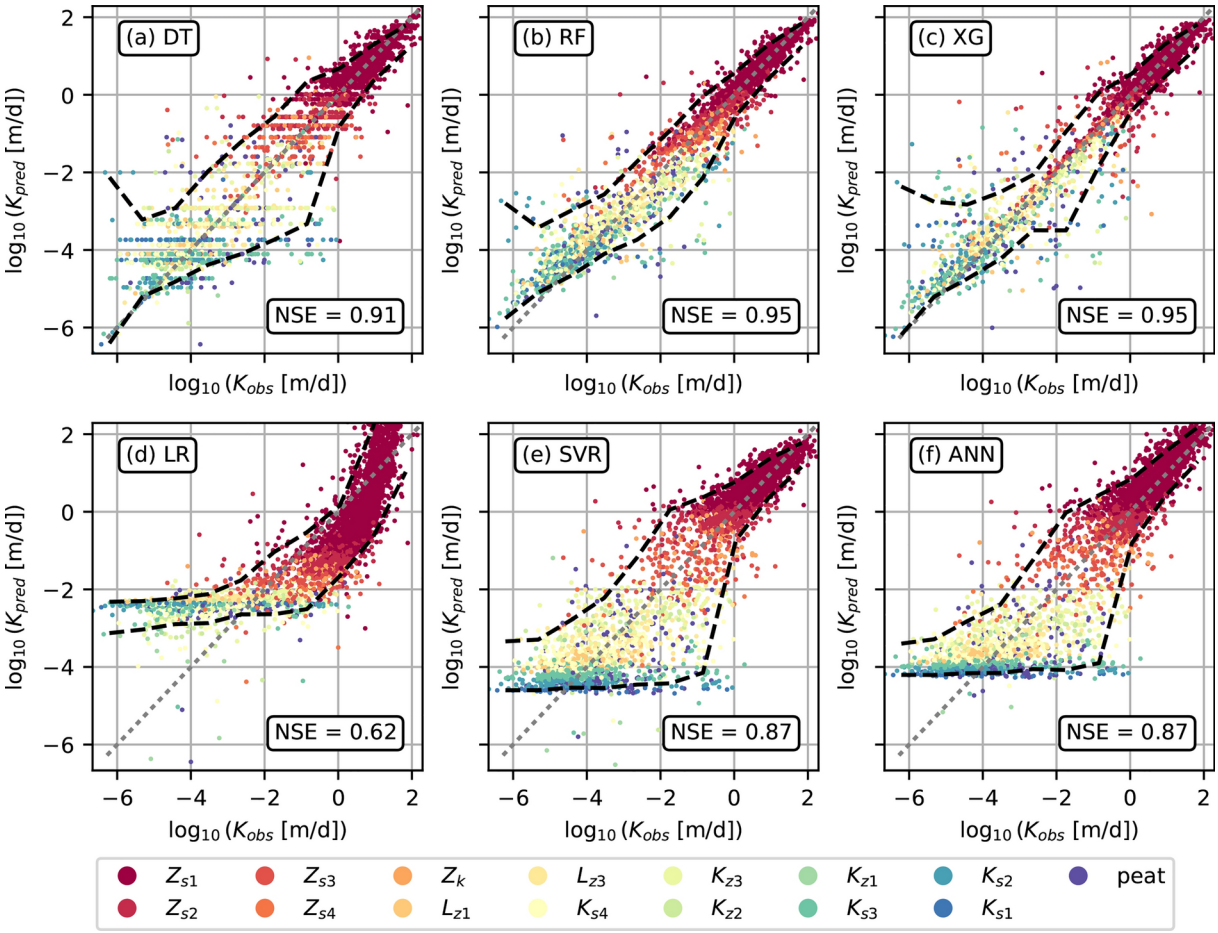

Fig. 6 Measured log-conductivity (x-axis) versus algorithm predicted K_f values (y-axis) based on d_{10}, d_{50} and d_{60} as feature variables for all six algorithms applied to the full *Top-All* dataset. The gray line represents the 1:1 line while black lines represent the 5^{th} and 95^{th} percentiles of predictions. Colors indicate lithoclasses (Table 1)

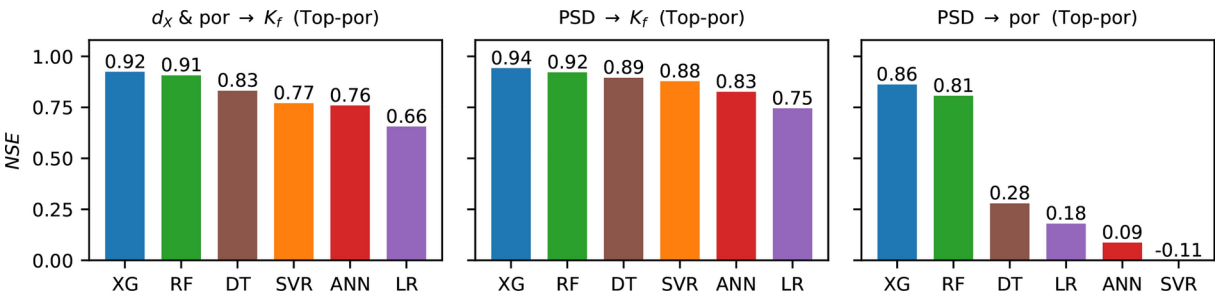

Fig. 7 Performance measure NSE of all six algorithms (colored bars) for the different target-feature variable combinations tested: estimating K_f from d_{10}, d_{50}, d_{60} and porosity (left), estimating K_f from the PSD (center, for comparison) and estimating porosity from the PSD (right) based on the data subset *Top-Por* for all samples (testing + training data)

SVR and LR (between 0.66 and 0.76), it is above 0.9 for XG, RF and DT (Fig. S.17). Thus, the better performance of the tree-based algorithms on the entire *Top-Por* data set (Fig. 7 (left) is due to their better training capability. NSE values for the combination $d_X \& \theta \rightarrow K_f$ are only slightly lower than those of the standard target-feature combination (PSD $\rightarrow K_f$) applied to *Top-Por*. We relate the reduced performance mostly to the smaller data set.

The benefit of having additional information on porosity is limited. Scores for the target-feature combination d_X & $\theta \rightarrow K_f$ are almost identical (marginally higher) than those of $d_X \rightarrow K_f$ for the *Top-Por* data set. Thus, in contrast to empirical formulas, ML algorithms do not benefit from information on porosity.

Algorithm testing for predicting porosity from PSD (as often used in practice) showed a relatively poor fit. Of all

algorithms, only XG and RF have high NSE values above 0.8 for the fit to the entire data. But this is the result of a high fit to the training data, while the score on the 20% testing data is much lower with 0.41 and 0.39. Thus, predictive capacity of ML algorithms for this feature-target combination is very limited.

4 Discussion

Our study is the first to show that PSD data is sufficient to estimate saturated hydraulic conductivity with high accuracy using ML. Considering the challenges associated with measuring hydraulic conductivity and the limitations in the use of empirical formulas, the performance of ML is very promising. While many studies report that ANN is highly suitable, we find that tree-based algorithms such as Random Forest and XGBoost outperform ANN for this type of application.

We could confirm literature results on the good performance of ML algorithms for estimating hydraulic conductivity from soil structural properties: Rogiers et al. (2012) for instance, obtained an R^2 of 0.93 on their entire dataset of 173 samples using an adapted ANN. Similarly applying ANN, other authors achieved high R^2 using a larger set of feature variables, including carbon content, pH, bulk density, or the plasticity index (Williams and Ojuri 2021; Albalasmeh et al. 2022; Yamaç et al. 2022). Noticeably, Trejo-Alonso et al. (2021) were able to predict K_f over a large range with an R^2-value of 0.97 on a dataset totaling 900 samples. They made use of ANN and seven types of measurement for feature variables: percentage of clay, sand and silt, bulk density, permanent wilting point, moisture content, and field capacity. The largest study in this context was conducted by Araya and Ghezzehei (2019) who made use of over 27,000 samples from 45 US datasets with predominantly sandy samples. Using a large set of feature variables, including bulk density, organic carbon content, clay, silt and sand fraction, coarse sand fraction, d_{10}, d_{50} and d_{60}, they obtained an overall R^2-value of 0.90.

We observed a strong impact of data set size and of the soil type, i.e. sand, silt, and lutum fraction of the samples. The entire data set *Top-All* showed the best algorithm performance, which we partly link to the highest amount of training data, but also to the broad range of soil types in the samples providing the best base for training. As described in Althnian et al. (2021), the representative variation of the original sample compared to the training sample is most important for good model performance. The need for large training datasets holds especially for non-tree algorithms like ANN and LR. We see that the predictive power of ANN significantly reduces when applied to small training data sets (less then a thousand samples) with high variation. Tree-based algorithms can deal better with smaller datasets, but still perform better with an abundance of data, including all soil types.

All algorithms performed significantly less well for the data subsets consisting of silty and clay samples. Both sets are much smaller (less then 800 samples). Specifically, DT and LR underperform, which we attribute to the relatively simple mechanism behind the algorithms. They cannot predict the more non-linear variations in the data, specifically with a lack of sufficient training data. Boosting and a multitude of decision tree's (as with XG and RF) improve the capability to deal with additional variation, similar to how SVR is superior to LR through its ability to employ non-linear assumptions. We link the performance reduction for the clay data set, specifically for ANN, also to the characteristic of clay samples. Higher clay content and burial depth lead to higher particle aggregation levels and/or elevated levels of compaction reducing conductance of water. Both these processes can not be predicted by PSD data. As van Leer et al. (2023) explicitly showed, PSD is not the dominant factor for predicting hydraulic conductivity in aquitards where most of the clay samples originate from.

Our results on the performance of empirical formulas with NSE/R^2-values ranging between 0.62 and 0.82 agree with previous studies. For instance, Rogiers et al. (2012) reported values between 0.62 and 0.75. Typically, Kozeny-Carman and Hazen are consistently identified as best performing relationships (Chandel and Shankar 2022). Both come with application restrictions, being only suitable for coarse sand. These limitations are easily overcome with ML algorithms.

Tests on feature importance showed that sieve sizes of 1–25 µm contribute most to an accurate prediction consistently throughout all tested algorithms. This sieve size coincides roughly with d_{10}. Thus, the abundance or lack of very fine particles determines the ability to conduct flow in the pore space rather then the median grain size (d_{50}). This finding is in line with those of Rehman et al. (2022), however in contrast to them we did not identify a high influence of d_{50}. The key influence of d_{10} on estimating hydraulic conductivity is also reflected in many empirical relationships using it as effective grain diameter, such as the equations of Hazen (1892) and Kozeny-Carman (Bear 1972).

Our results confirm that predictability of K_f from d_{10}, d_{50}, d_{60} using ML algorithms is high, particularly for sandy samples and tree-based algorithms. Model performance (NSE ≥ 0.87, excluding LR) is slightly higher than obtained by Rehman et al. (2022) who trained ANN on multiple PSD-derived variables (d_5, d_{10}, d_{30}, d_{50} d_{60}, d_{90}) and deposition rates, but only to 180 sandy soils samples. Notably, using

d_{10}, d_{50}, and d_{60} is not recommended for clay dominated samples.

Additional information on porosity does not contribute to a significant improvement in model performance compared to only using PSD-derived quantities. It rather limits the performance when reducing data set size by a lack of porosity measurements. Although porosity is one of the few input parameters of empirical formulas, it does not significantly contribute to the estimation of K_f.

None of the tested algorithms were able to accurately predict porosity from PSD (for our dataset). While some algorithms managed a good training performance, the testing performance was poor for all. We did not identify a strong correlation between PSD and total porosity although this is typically assumed and porosity values are estimated from PSD for many empirical formulas (Devlin 2015). However, effective porosity predictions are often conducted in the context of gas reservoir exploration at greater depths than for shallow samples as considered here. But also here, correlation between PSD and total porosity is highly unlikely, given the importance of compaction which changes porosity but not PSD, as discussed in Richard et al. (2001).

While we consider our results concerning the use of ML with PSD data solely and the performance of structurally different algorithms as general outcomes, there are limitations for application to other data sets. Our results are also subject to uncertainty. The algorithms were tailored for application to Dutch aquifer samples which show a relative abundance of sandy samples. We use cross-validation during hyperparameter tuning to reduce uncertainty by limiting overfitting to the training dataset. However, aleatoric uncertainty related to presence of residual noise in the dataset (Kendall and Gal 2017) cannot be ruled out despite extreme care taken to avoid measurement noise (Buma et al. 2024). The heterogeneity of other parameters within the subsurface, with varying structural properties, mineral content, and organic matter across different regions, may contribute to prediction uncertainty. Since the used dataset is large (around 4600 points) with a rich sample distribution, we consider epistemic uncertainty (model performance variability on unseen test data) to be small. We consider this also for the clay and silt subsets which may be small in the context of ML training and testing but still are substantial (hundreds of samples). However, the dataset might lack residual anomalies present in other soils. Thus, application of the trained algorithms to datasets of non-Dutch soils require careful consideration.

In this respect, very coarse sands and gravels deserve special attention since they have so far been underrepresented in the *TopIntegraal* dataset, as evidenced by Table 2. For example, only 5% of the sand samples (N=3326) have sand medians larger than 560 µm. This is caused by physical constraints to the sample volumes that can be handled in the lab where the dataset was established. The applied ML algorithms are not geared towards extrapolation of results beyond the PSD-ranges of the training data. Therefore, applicability to very coarse sand and gravel samples could neither be confirmed nor rejected.

Despite this, the results of this study offer sufficient potential benefits for water resource management, geothermal applications (closed loop, seasonal storage) and environmental engineering. These applications tend to compete for limited subsurface space, notably in fine and medium coarse sand aquifers. While K_f is key in determining suitability and potential, the availability of undisturbed core data as a source of K_f measurement data is limited. Sediment samples from drillings, on the other hand, are generally more widely available, e.g. for thousands of wells in the Dutch national drilling repository. In addition, PSDs are measured routinely and cheap. The good results obtained using the ML algorithms indicate that these PSDs can be used for predicting K_f, thereby significantly contributing to the parametrization of hydrological models and geothermal potential calculations. Because compaction effects are ignored in the sand subset of the *TopIntegraal* database, applicability is still limited to the depth range specified by the sample depths. Conversion of the results to a depth domain where the importance of compaction is larger, using methods that are commonly used in reservoir modeling, will therefore be an important direction for future research.

5 Summary and conclusion

We studied the application of six machine learning (ML) algorithms to a soil sample dataset containing textural and structural information of over 4,500 samples from the shallow Dutch subsurface. We trained all algorithms with measured particle size distributions targeted at predicting hydraulic conductivity measurements from laboratory investigations. We compared algorithm performances for different data-sub sets; we compared ML-estimates of hydraulic conductivity with five empirical formulas; we assessed the performance of the algorithm with a reduced set of feature variables based on PSD-derived variables; and we evaluated the potential of ML for porosity prediction from PSD. From our results we draw the following main conclusions as:

- The particle size distribution solely is well suited for estimating hydraulic conductivity from shallow subsurface soil samples (including sand, clay and silt types).
- Tree-based algorithms such as Random Forest and XGBoost are best suited for prediction.

- PSD-trained algorithms outperform the empirical formulas.
- Prediction improves with increasing dataset size. Tree algorithms are least sensitive to dataset size reduction.
- Sieve sizes in the range of d_{10} are most influential on model outcomes.
- PSD-derived quantities (d_{10}, d_{50}, d_{60}) are well suited for hydraulic conductivity prediction for sandy samples. Models trained on these parameters are less appropriate for predictions of hydraulic conductivity of silty and clay-rich samples. Again tree-based algorithms and XGBoost perform best.
- PSD is not a suitable predictor for porosity.Our study demonstrates as one of the first, the feasibility and benefits of utilizing ML algorithms, particularly tree-based algorithms for predicting hydraulic conductivity from PSD data solely. The use of ML can support time-intensive fieldwork and replace inaccurate empirical formulas for estimating hydraulic conductivity. Relying solely on PSD-data reduces the dependence on other soil properties.

As within all ML applications, the use of the trained algorithms with other data(sets) has to be carefully evaluated case-specifically. However, due to the variety within the *Topintegraal* dataset used, we expect that the trained algorithms will perform accurately for PSD data from different areas. Algorithm application to other datasets from the Dutch subsurface is work in progress. We see large potential for application to deep aquifers, which are explored for geothermal potential. There, in-situ measurements of hydraulic conductivity are infeasible and PSDs from a few samples are the only source for information.

Other areas of application within the context of the *Topintegraal* dataset are filling gaps in the data set (i.e. samples missing a K_f value), quality checking of permeameter test results, and parametrization of hydrological models.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00477-024-02861-6.

Acknowledgements The python scripts for data analysis and reproducing the Figures are provided in an open-source repository on Github (Zech 2024). The *TopIntegraal*-data set is available at Doorlatendheden.

Author contributions V.d.R. setup the data analysis with support of A.Z. V.d.R. and A.Z wrote the main manuscript text, prepared all figures and the associated python scripts. J.B. and H.M. provided the data and supported analysis of results. All authors reviewed the manuscript and contributed to the revision process.

Funding The authors have not disclosed any funding.

Data availability The data (TopIntegraal data set) under use is published at: https://www.grondwatertools.nl/doorlatendheden. The python scripts for data analysis and reproducing the manuscript figures are provided in an open-source repository on Github (https://github.com/AlrauneZ/PSD_Analysis_MachineLearning) and published alongside at Zenodo (doi: https://doi.org/10.5281/zenodo.12543937).

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Albalasmeh A, Mohawesh O, Gharaibeh M et al (2022) Artificial neural network optimization to predict saturated hydraulic conductivity in arid and semi-arid regions. Catena 217(106):459. https://doi.org/10.1016/j.catena.2022.106459

Althnian A, AlSaeed D, Al-Baity H et al (2021) Impact of dataset size on classification performance: An empirical evaluation in the medical domain. Appl Sci 11:1–18. https://doi.org/10.3390/app11020796

Alyamani MS, Şen Z (1993) Determination of hydraulic conductivity from complete grain-size distribution curves. Groundwater 31(4):551–555. https://doi.org/10.1111/j.1745-6584.1993.tb00587.x

Araya SN, Ghezzehei TA (2019) Using machine learning for prediction of saturated hydraulic conductivity and its sensitivity to soil structural perturbations. Water Resour Res 55(7):5715–5737. https://doi.org/10.1029/2018WR024357

Baars J (2004) Bepaling van korrelgrootte van grondmonsters m.b.v. laserdiffractie apparatuur (malvern mastersizer 2000) (in Dutch). Tech. Rep. Werkvoorschrift GL-WV-004, Geïntegreerd Laboratorium TNO-NITG en Universiteit Utrecht, Faculteit Geowetenschappen

Barr DW (2001) Coefficient of permeability determined by measurable parameters. Groundwater 39(3):356–361. https://doi.org/10.1111/j.1745-6584.2001.tb02318.x

Bear J (1972) Dynamics Fluids in Porous Media. Elsevier, New York

Breiman L (2001) Random Forests. Mach Learn 45(1):5–32. https://doi.org/10.1023/A:1010933404324

Buma J, de Heer E, Bus S, et al (2024) Topintegraal, het boor- en meetprogramma van de ondiepe ondergrond van nederland, deelrapport 1. meetgegevens en kentallen verzadigde doorlatendheid, versie 1.0. Tech. Rep. TNO 2023 R10561, TNO Geologische Dienst Nederland

Cabalar AF, Akbulut N (2016) Evaluation of actual and estimated hydraulic conductivity of sands with different gradation and shape. SpringerPlus 5:820. https://doi.org/10.1186/s40064-016-2472-2

Chandel A, Shankar V (2022) Evaluation of empirical relationships to estimate the hydraulic conductivity of borehole soil samples. ISH J Hydr Eng 28:368–377. https://doi.org/10.1080/09715010.2021.1902872

Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, New York, NY, USA, KDD '16, pp 785–794. https://doi.org/10.1145/2939672.2939785

Devlin JF (2015) HydrogeoSieveXL: an Excel-based tool to estimate hydraulic conductivity from grain-size analysis. Hydrogeol J 23(4):837–844. https://doi.org/10.1007/s10040-015-1255-0

Granata F, Di Nunno F, Modoni G (2022) Hybrid machine learning models for soil saturated conductivity prediction. Water 14(11):1729. https://doi.org/10.3390/w14111729

Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York

Hazen A (1892) Physical properties of sands and gravels with reference to their use infiltration. Tech. Rep, Twenty Fourth Annual Report, Massachusetts state board of health

Jorda H, Bechtold M, Jarvis N et al (2015) Using boosted regression trees to explore key factors controlling saturated and near-saturated hydraulic conductivity. Eur J Soil Sci 66(4):744–756. https://doi.org/10.1111/ejss.12249

Kendall A, Gal Y (2017) What uncertainties do we need in bayesian deep learning for computer vision? https://doi.org/10.48550/arXiv.1703.04977

Kotlar AM, Iversen BV, de Lier QJV (2019) Evaluation of parametric and nonparametric machine-learning techniques for prediction of saturated and near-saturated hydraulic conductivity. Vadose Zone J. https://doi.org/10.2136/vzj2018.07.0141

Kozeny J (1927) Über kapillare Leitung des Wassers im Boden (Aufstieg, Versickerung u. Akademie der Wissenschaften, Anwendung auf die Bewässerg)

van Leer MD, Zaadnoordijk WJ, Zech A et al (2023) Dominant factors determining the hydraulic conductivity of sedimentary aquitards: a random forest approach. J Hydrol 627(130):468. https://doi.org/10.1016/j.jhydrol.2023.130468

Mady AY, Shein EV (2018) Support vector machine and nonlinear regression methods for estimating saturated hydraulic conductivity. Moscow Univ Soil Sci Bull 73(3):129–133. https://doi.org/10.3103/S0147687418030079

NEN2019 (2019) Geotechnical investigation, testing and classification of soil - part 2: Principles for classification. Tech. rep

Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Rehman Z, Khalid U, Ijaz N et al (2022) Machine learning-based intelligent modeling of hydraulic conductivity of sandy soils considering a wide range of grain sizes. Eng Geol 311(106):899. https://doi.org/10.1016/j.enggeo.2022.106899

Richard G, Cousin I, Sillon JF et al (2001) Effect of compaction on the porosity of a silty soil: influence on unsaturated hydraulic properties. Euro J Soil Sci 52:49–58. https://doi.org/10.1046/j.1365-2389.2001.00357.x

Rogiers B, Mallants D, Batelaan O et al (2012) Estimation of hydraulic conductivity and its uncertainty from grain-size data using GLUE and artificial neural networks. Math Geosci 44:739–763. https://doi.org/10.1007/s11004-012-9409-2

Russell S, Norvig P (2020) Artificial Intelligence: A Modern Approach, 4th edn. Pearson, Hoboken

Sarker IH (2021) Machine learning: algorithms, real-world applications and research directions. SN Comput Sci. https://doi.org/10.1007/s42979-021-00592-x

Schaap MG, Leij FJ, van Genuchten MT (2001) Rosetta: a computer program for estimating soil hydraulic parameters with hierarchical pedotransfer functions. J Hydrol 251(3):163–176. https://doi.org/10.1016/S0022-1694(01)00466-8

Shepherd RG (1989) Correlations of permeability and grain size. Groundwater 27(5):633–638. https://doi.org/10.1111/j.1745-6584.1989.tb00476.x

Sihag P, Esmaeilbeiki F, Singh B et al (2019) Modeling unsaturated hydraulic conductivity by hybrid soft computing techniques. Soft Comput 23(23):12897–12910

Tahmasebi P, Kamrava S, Bai T et al (2020) Machine learning in geo- and environmental sciences: from small to large scale. Adv Water Resour 142(103):619. https://doi.org/10.1016/j.advwatres.2020.103619

Trejo-Alonso J, Fuentes C, Chávez C et al (2021) Saturated hydraulic conductivity estimation using artificial neural networks. Water. https://doi.org/10.3390/w13050705

Van Baaren J (1979) Quick-look permeability estimates using sidewall samples and porosity logs. In: Transactions of the 6th Annual European Logging Symposium, Society of Professional Well Log Analysts, pp 1–20

Veldkamp JG, Geel CR, Peters E (2021) Characterization of aquifer properties of the brussels sand member from cuttings particle size distribution and permeability characterization of aquifer properties of the brussels sand member from cuttings. Tech. rep

Vienken T, Dietrich P (2011) Field evaluation of methods for determining hydraulic conductivity from grain size data. J Hydrol 400(1–2):58–71. https://doi.org/10.1016/j.jhydrol.2011.01.022

Vukovic M, Soro A (1992) Determination of Hydraulic Conductivity of Porous Media from Grain-size Composition. Water Resources Publications

Williams CG, Ojuri OO (2021) Predictive modelling of soils' hydraulic conductivity using artificial neural network and multiple linear regression. SN Appl Sci 3:152. https://doi.org/10.1007/s42452-020-03974-7

Yamaç SS, Negiş H, Şeker C et al (2022) Saturated hydraulic conductivity estimation using artificial intelligence techniques: a case study for calcareous alluvial soils in a semi-arid region. Water. https://doi.org/10.3390/w14233875

Zech A (2024) AlrauneZ/PSD_analysis_machinelearning. https://doi.org/10.5281/zenodo.12543937, https://github.com/AlrauneZ/PSD_Analysis_MachineLearning