Wiley
Advances in Civil Engineering
Volume 2025, Article ID 3042895, 11 pages
https://doi.org/10.1155/adce/3042895

Research Article

Optimisation of Ensemble Learning Algorithms for Geotechnical Applications: A Mathematical Approach to Relative Density Prediction

Mahdy Khari[1], **Ali Dehghanbanadaki**[2,3] **Danial Jahed Armaghani**[4] and **Manoj Khandelwal**[5]

[1]*Department of Civil Engineering, Islamic Azad University, East Tehran Branch, Tehran, Iran*
[2]*Department of Civil Engineering, Islamic Azad University, Dam.C., Damavand, Iran*
[3]*Research Center of Concrete and Soil, Islamic Azad University, Dam.C., Damavand, Iran*
[4]*School of Civil and Environmental Engineering, University of Technology Sydney, Ultimo, New South Wales 2007, Australia*
[5]*Institute of Innovation, Science and Sustainability, Federation University Australia, Ballarat, Victoria 3350, Australia*

Correspondence should be addressed to Manoj Khandelwal; m.khandelwal@federation.edu.au

Received 11 January 2025; Revised 23 April 2025; Accepted 20 June 2025

Academic Editor: Xiaolong Sun

The challenge of predicting relative dry density (D_r) in granular materials is addressed through advanced mathematical modelling and machine learning (ML) techniques. A novel approach to optimise ensemble learning algorithms is presented, with a focus placed on the mathematical foundations of these methods. An experimental dataset obtained from a mobile pluviator was utilised to develop and analyse various ML models. The mathematical analysis was centred on the optimisation and comparative performance of ensemble methods, with particular emphasis given to gradient boosting regression (GBR), AdaBoost regression, and extreme gradient boosting (XGBoost). The mathematical formulation of the GBR model was rigorously examined and optimised using advanced tuning functions, achieving exceptional performance metrics (mean squared error [MSE] = 11.91, mean absolute error [MAE] = 1.93, $R^2 = 0.997$). Through sensitivity analysis, it was revealed that the distance between the shutter plate and the top sieve is the most significant factor affecting D_r prediction. A computational platform was developed within the Google Colab environment, demonstrating the practical application of the mathematical models. This research contributes to applied mathematics by showcasing advanced algorithmic approaches to solving complex geotechnical engineering problems while providing a rigorous mathematical foundation for future developments.

Keywords: gradient boosting regressor; machine learning; mobile pluviator; relative density; sandy soil

1. Introduction

Small-scale laboratory testing conducted either under normal gravity or in a centrifugal acceleration field is a crucial research strategy in geotechnical engineering. These tests are typically designed to replicate large-scale geotechnical issues using significantly smaller models in a controlled laboratory setting [1]. One of the main controlled parameters in small-scale laboratory testing on sandy soils is the relative density (D_r) [2]. This parameter has a positive correlation with sandy soil strength, as reported in previous studies [3–5]. It is crucial to obtain a uniform sand sample with a constant D_r to achieve high accuracy and consistency when investigating soil characteristics. Unlike clay specimens, which can be uniformly trimmed from a large consolidated block, sandy soils lack natural cohesion, making it difficult to create uniform samples. Consequently, each sandy specimen must be individually prepared for testing, complicating the process. Therefore, reconstituting sandy samples in the laboratory is desirable. The preparation methods for sandy soil samples must consider several criteria,

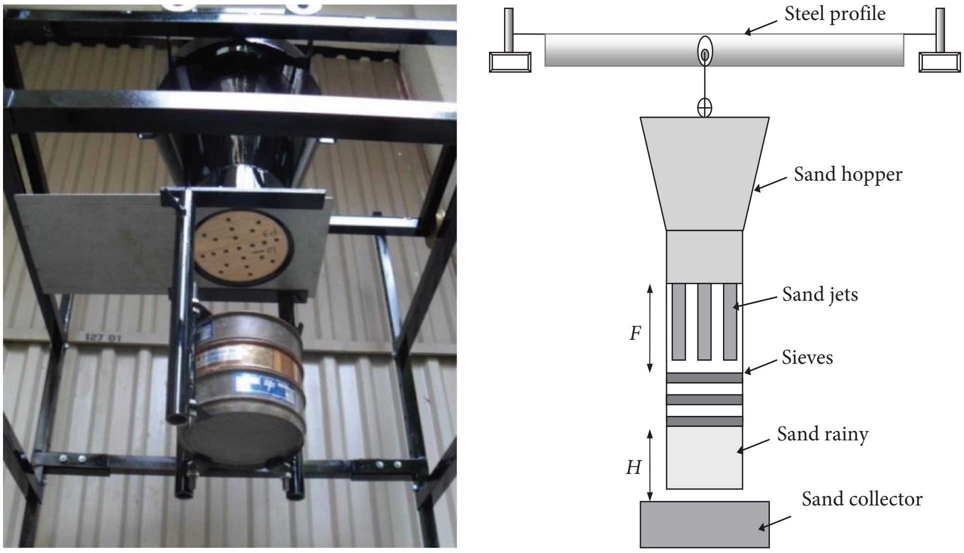


Figure 1: Mobile pluviator (Khari et al. [7]).

including the preparation of both dense and loose samples, ensuring a uniform void ratio, and achieving a well-mixed sample without particle segregation [6].

Reconstituted sands using the air pluviation method involve creating sand samples by allowing sand to fall through the air into a container, achieving a uniform distribution and desired density for laboratory testing. This method utilises a controlled setup where sand is released at a regulated rate from a specific height through a hopper or funnel, ensuring consistent deposition and reducing particle segregation. By adjusting the height and flow rate, different target densities can be achieved, with higher drop heights leading to denser packing. The air pluviation method offers advantages, such as uniformity, precise density control, and reproducibility, making it ideal for geotechnical testing of soil properties like shear strength, compressibility, and permeability, as well as for research purposes to study the behaviour of sands under various conditions and validate theoretical models [7].

Despite research on pluviation methods for preparing sandy soil samples [8–10] and the acknowledged challenges of this process in terms of time, cost, and complexity, a crucial gap remains. To date, no research has applied machine learning (ML) techniques to predict the D_r of sandy soils prepared using a mobile pluviator. Although ML has demonstrated significant success across various engineering problems [11–14] and geotechnical engineering tasks, including predicting pile bearing capacity [15–19], analysing slope stability [20–24], classifying soils [25, 26], evaluating ground improvement [27–32], assessing soil liquefaction [33–35], determining sand content [36], and estimating other geotechnical parameters [37], its application to predicting D_r based on mobile pluviator parameters has not yet been explored.

This research directly addresses the current gap in sand D_r prediction by developing and applying ML models for accurate and efficient D_r estimation. This novel approach aims to improve the control and consistency of sand sample preparation for geotechnical testing, an area that has not been extensively explored using automated ML techniques. Specifically, this study is among the first to leverage automated model selection and explainable AI tools for D_r prediction in laboratory-prepared sandy soils. To achieve this, we collect data from mobile pluviator tests, incorporating key parameters, such as D50, shutter plate porosity, and sand fall height. The dataset is preprocessed and used to train a variety of ML models within a Google Colab environment using the PyCaret library, which systematically compares and ranks models based on their performance. Model accuracy is rigorously evaluated using metrics, including mean squared error (MSE), mean absolute error (MAE), and the coefficient of determination (R^2), ensuring the most robust model is selected. A comprehensive sensitivity analysis utilising SHAP values is conducted to identify the most influential factors affecting D_r, providing valuable interpretability to the ML predictions. As an additional contribution, the optimal model is further fine-tuned for maximum predictive accuracy, and a user-friendly web-based platform is developed within Google Colab. This platform enables geotechnical researchers to easily estimate D_r, representing a significant step towards the practical adoption of ML-driven tools in soil testing laboratories.

2. Data Collection for Estimating D_r

We encourage geotechnical researchers to examine the experimental dataset used in this study, which was also utilised for developing ML models in the study by Khari et al. [7]. Here, we briefly review the testing methods employed by Khari et al. [7]. The mobile pluviator used in their study is depicted in Figure 1. This device primarily consists of a soil bin (hopper), a diffuser system (three sieves), a sand collector, and a fixing device that holds these components together, allowing the entire system to be supported by a movable steel frame. The fixing device includes three steel pipes arranged in a circle with a 20 cm diameter. The sand hopper, which stores the sand, is conical with a base diameter of 20 cm. Controlling the sand deposition intensity was crucial for achieving the desired relative density, as it regulated the sand mass flow rate. To this end, various arrangements of holes, differing in diameter and number, were tested. These patterns were created on a wooden perforated plate with a diameter of 20 cm (known as the shutter plate). Four shutter plate patterns with different hole arrangements were used to control the soil discharge rate. These interchangeable circular wooden plates were installed at the bottom of the sand hopper, with porosities of 0.8%, 1.88%, 5.25%, and 5.5%. The drop distance (H), defined as the distance between the bottom sieve and the sand surface within the collector, is a crucial parameter for achieving terminal velocity and the desired relative density. Therefore, this distance should be equal to or greater than the height determined during sand pluviation. Additionally, the height F (distance between the shutter plate and the top sieve) and H complement each other in this setup.

To estimate the D_r, 84 tests from the study by Khari et al. [7] using three different sandy soils were selected. In these tests, the parameters D50 (mm), area (%) as porosities of the shutter plate, H (cm), and F (cm) were chosen as inputs for the ML models, with D_r (%) as the target variable. Figure 2a–d illustrates the statistical information of the parameters used.

3. Design of ML Models

The present study utilised Pycaret (https://pycaret.org/), a Python-based ML library developed by Ali [38]. Pycaret is an open-source, low-code tool built primarily on the scikit-learn library (https://scikit-learn.org/). It integrates various widely used ML libraries in Python (Table 1), facilitating a streamlined and user-friendly experience. Key features of Pycaret include "compare_models ()", which allows for the comparison of model performance, and "ensemble_model ()", which combines predictions from multiple models to enhance accuracy. Table 1 presents the details of the data divisions and the training process using the Pycaret library.

Our goal was to predict the D_r of sandy soils using ML models, treating D_r as a regression target. The original dataset comprised 84 observations, each with five features. Following preprocessing, the dataset maintained its original shape of 84 observations and five features. The data was then divided into a training set with 58 observations and a test set with 26 observations. To ensure robust model validation and minimise the risk of overfitting, we employed a 10-fold K-fold cross-validation approach during the training and evaluation of all ML models. The preprocessing steps and model training were conducted using Pycaret in the Google Colab Integrated Development Environment. The developed code is presented below in Table 1.

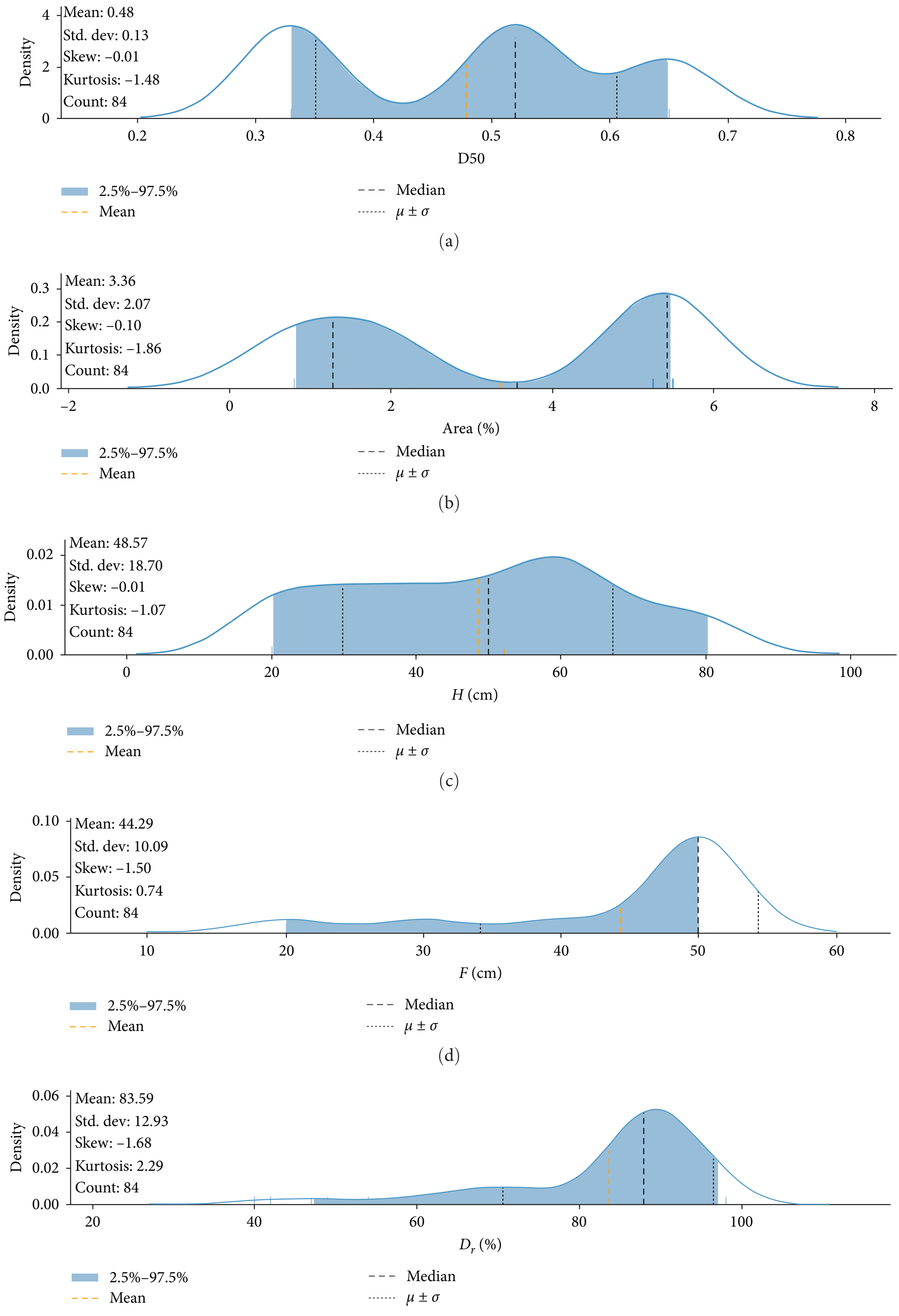
Developed code in PyCaret:

```
from pycaret.regression import *
import pandas as pd
data = pd.read_csv('data.csv')
exp = setup(data = data, target='Dr (%)', train_size = 0.7, fold = 10,
numeric_features = ['D50 (mm)', 'Area (%)', 'H (cm)', 'F (cm)'],
preprocess = True, fold_strategy = 'kfold', session_id = 123)            .
gbr = create_model('gbr')
tuned_gbr = tune_model(gbr, optimize='R2')
feature_importance = get_feature_importance(tuned_gbr)
final_model = finalize_model(tuned_gbr)
save_model(final_model, 'gbr_model_final')
```

4. Results and Discussions

4.1. Correlation Matrix. Figure 3 presents the correlation matrix illustrating the relationships between the input and target variables in the dataset. The matrix displays correlation coefficients, which indicate the degree to which changes in one variable are associated with changes in another, independent of the influence of other parameters. Therefore, interpretation of these coefficients should be done with caution. As shown, the parameter F (cm), the distance between the shutter plate and the top sieve, exhibits the strongest correlation with D_r, with a value of 0.78. This strong positive relationship is expected, given the physical significance of F in the experimental setup. In contrast, the parameter H, defined as the distance between the bottom sieve and the sand surface within the collector, has a negative correlation with D_r at −0.65, indicating an inverse relationship relative to F. The D50 parameter, representing the particle size at which 50% of the soil's mass is finer, shows a weaker positive correlation with D_r (0.15). While this value is relatively low, it suggests that a larger D50, which reflects coarser grains, may contribute to an increase in relative density and shear strength [39], thus justifying the observed positive correlation.

4.2. Identification of the Optimal ML Model. Pycaret simplifies the ML workflow by automating numerous steps, including



(a)

(b)

(c)

(d)

(e)

FIGURE 2: Details of the input and target parameters. (a) D50, (b) area (%), (c) H (cm), (d) F (cm), and (e) D_r.

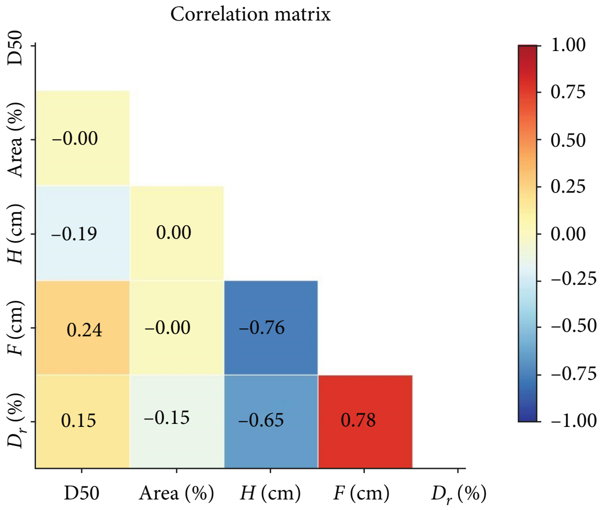


Figure 3: Correlation matrix of the parameters.

Table 1: Details of the ML training.

Parameters	Values
Target	D_r (%)
Target type	Regression
Original data shape	(84, 5)
Transformed data shape	(84, 5)
Transformed train set shape	(58, 5)
Transformed test set shape	(26, 5)
Numeric features	4
Preprocess	True
Fold generator	K-fold
Fold number	10

model training, comparison, ensembling, and tuning. It trains multiple models on the dataset, evaluates their performance, and aids in selecting the best model for the task. Upon initiating the training of ML models, Pycaret ranks all the default ML models based on performance indices, such as MSE, MAE, and R^2. Figure 4a–c illustrates these comparisons according to the mentioned performance indices.

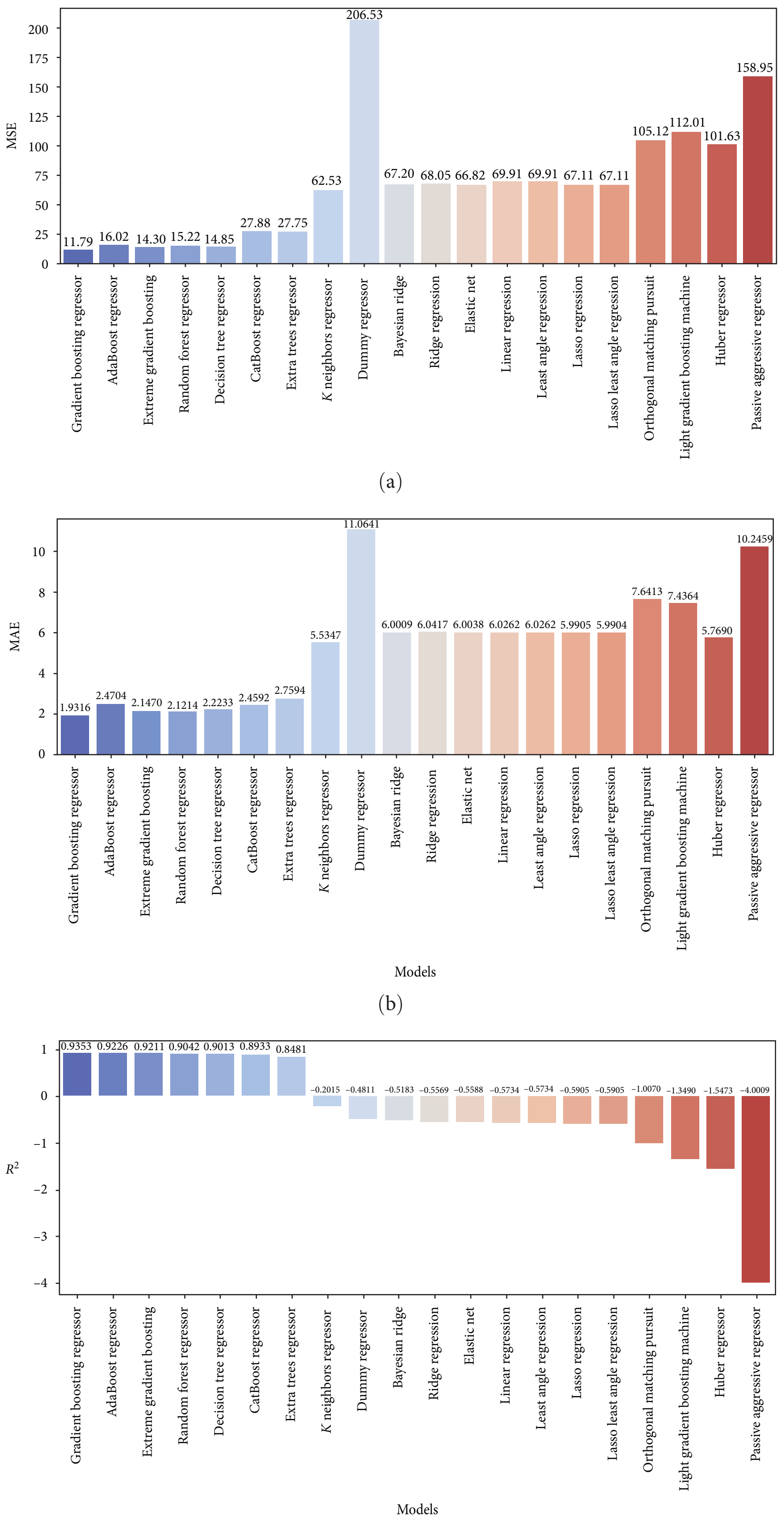
As shown in Figure 4a, the top three ML models based on MSE are gradient boosting regression (GBR), AdaBoost regressor, and extreme gradient boosting (XGBoost), with GBR having the lowest MSE of 11.91. Similarly, Figure 4b indicates that GBR also ranked first with an MAE of 1.93, while AdaBoost had an MAE of 2.47. In the final plot, as Figure 4c of performance indices, the R^2 values of the ML models are compared. The top five ML models demonstrated satisfactory R^2 values, with GBR showing an R^2 of 0.935. Conversely, several models, such as elastic net, linear regression, lasso regression, and Huber regression, exhibited negative R^2 values. This negative R^2 is a clear indication that these models are not suitable for the data, and alternative models or features should be considered. Based on the mentioned results, the GBR was found and selected to be the best ML model for this study. Therefore, in this section, the application and structure of this ML model are presented.

GBR is a powerful ML technique known for its predictive accuracy. It belongs to the family of boosting algorithms, which iteratively combine the predictions of multiple weak learners, typically decision trees, to create a strong learner. Unlike bagging methods like Random Forest, which train weak learners independently, GBR trains them sequentially, with each subsequent tree correcting the errors of its predecessors. This sequential approach allows GBR to focus on the more difficult-to-predict instances, gradually improving the overall model performance [38]. The core idea behind GBR is to minimise a loss function, which measures the difference between the predicted and actual values. In each iteration, a new decision tree is trained to predict the negative gradient of the loss function with respect to the previous predictions. This negative gradient represents the direction and magnitude of the steepest descent towards the minimum of the loss function. By adding the predictions of this new tree, weighted by a learning rate, the overall model moves closer to the optimal solution. This process is repeated for a specified number of iterations, or until the loss function converges [12].

Furthermore, its effectiveness with limited datasets aligns well with the practical constraints of laboratory testing. The ensemble nature of GBR, combining multiple weak learners, makes it robust against overfitting, a critical concern when developing predictive models with limited data. Finally, GBR's successful application in various geotechnical contexts, as demonstrated in existing literature [40–46], supports its suitability for predicting relative density from mobile pluviator data.

4.3. Tuning the GBR Model. Pycaret simplified the ML workflow by automating many steps, including model training, comparison, ensembling, and tuning. The tune_model function was used to optimise the hyperparameters of the GBR model. This function explores the hyperparameter space using a grid search or randomised search approach and evaluates the performance of the GBR model with different combinations of hyperparameters. The evaluation metric used for tuning was MSE. The detailed hyperparameters of the tuned gradient boosting regressor model are presented in Table 2. The value of alpha, set to 0.9, is used for quantile loss functions to specify the desired quantile. This setting fine-tunes the model to focus on a specific quantile, indicating an emphasis on robust performance across different quantiles of the target distribution. The parameter ccp_alpha, set to 0.0, represents the complexity parameter used for minimal cost-complexity pruning. This implies no pruning, allowing the model to grow without restrictions from cost-complexity considerations. Additionally, a learning rate of 0.1 was employed to control the contribution of each tree to the final model. This value balances the model's learning speed and accuracy, preventing overfitting while ensuring adequate learning. The parameter max_depth, set to 3, limits the maximum depth of individual trees in the ensemble. A depth of three strikes a balance between capturing complex patterns and preventing overfitting.

Furthermore, the parameter max_features is set to none, allowing each tree to consider all available features when



(a)

(b)

(c)

Figure 4: Performance indices of the ML models: (a) MSE, (b) MAE, and (c) R^2.

Table 2: Details of tuned_GBR model.

Parameters	Values
alpha	0.9
ccp_alpha	0.0
criterion	friedman_mse
Init	None
learning_rate	0.1
Loss	squared_error
max_depth	3
max_features	None
max_leaf_nodes	None
min_impurity_decrease	0.0
min_samples_leaf	1
min_samples_split	2

looking for the best split, ensuring comprehensive utilisation of the dataset's information. The parameter max_leaf_nodes, also set to None, permits unlimited growth in terms of leaf nodes per tree, enabling each tree to capture detailed patterns in the data. The parameter min_impurity_decrease, set to 0.0, indicates that a split will only be made if it results in a decrease in impurity greater than or equal to this value. This default setting ensures that splits are only made when they improve the model's accuracy. Lastly, the parameter min_samples_leaf, set to 1, specifies the minimum number of samples required to be at a leaf node, allowing the model to capture fine-grained patterns in the data [38]. The source code and the algorithm of the GBR model are presented after Table 2.

Source code of the GBR model in scikit-learn:

$$\begin{gathered} class\ sklearn.ensemble.GradientBoostingRegressor \\ (*, loss = 'squared\_error', learningrate = 0.1, nestimators = 100, subsample \\ = 1.0, criterion = 'friedman\_mse'; \sim min\_samples\_split = 2, minsamplesleaf \\ = 1,\ min\_weight\_fraction\_leaf = 0.0, max\_depth = 3,\ min\_impurity\_decrease \\ = 0.0, init = None, random\_state = None,\ max\_features = None,\ alpha \\ = 0.9,\ verbose = 0,\ max\_leaf\_nodes = None,\ warm\_start \\ = False,\ validation\_fraction = 0.1,\ niternochange = None,\ tol \\ = 0.0001,\ ccpalpha = 0.0) \end{gathered}$$

4.4. Performance Indices of the GBR Model. Figure 5 depicts the recursive feature elimination with cross-validation (RFECV) process for the tuned_GBR model, which is used to identify the optimal number of features for predicting the D_r. In this figure, the x-axis shows the number of features included in each iteration of the RFECV process, while the y-axis displays the performance score of the GBR model, assessed through cross-validation. A higher score on the y-axis indicates better model performance. The solid line represents the change in performance score as more features are added, with an upward trend indicating improved model performance with additional features. In this study, all four features were selected as the optimal subset, achieving the highest score of 0.935.

Figure 6 illustrates the regression indices of the tuned_GBR model for both the training and test datasets. The training data, represented by blue points, shows an exceptionally high R^2 value of 0.997, indicating that the model captures the training data very accurately. In contrast, the test data, represented by green points, have a lower R^2 value of 0.876, reflecting a reduction in performance when the model is tested on new data. The residuals for both sets are generally centred around zero, suggesting that the model's predictions are unbiased on average. However, there are some larger residuals in the test set, particularly around the predicted value of 60, where several green points fall significantly below the zero line. The accompanying histogram on the right side of the plot displays the residual distribution, showing that while most errors are close to zero, there are some noticeable outliers, particularly on the negative side.

4.5. Sensitivity Analysis and Model Deployment. Figure 7 presents the feature importance as determined by SHAP (SHapley Additive exPlanations) values for the tuned GBR model. SHAP values provide an interpretable measure of each feature's contribution to the model's predictions, with the x-axis representing the average impact on the predicted D_r. The bar chart clearly indicates that "F (cm)" the distance between the shutter plate and the top sieve has the highest average SHAP value, making it the most influential feature for D_r prediction. This

```
Input:
  • Training dataset containing features and target values
  • Number of iterations ($M$)
  • Learning rate ($\eta$)
  • Loss function
  • Base learner type (typically decision trees)
Ourput:
  • Final ensemble model
Algorithm Steps:
1. Initialisation Phase:
  ○ Initialise the model with the mean value of the target variable
  ○ This serves as the starting point for predictions
2. Iterative Boosting Process: For each iteration $m$ from 1 to $M$:
Step A: Residual Calculation
  ○ Calculate the difference between actual and predicted values
  ○ These differences represent the current model's errors
  ○ Store these residuals for the next step
Step B: Base Learner Training
  ○ Train a new base learner (decision tree) using:
  ■ Input features as independent variables
  ■ Residuals from Step A as target variables
  ○ The base learner aims to predict the residuals
Step C: Optimisation
  ○ Determine the optimal weight for the new base learner
  ○ This weight controls how much the new learner contributes
  ○ Usually implemented through line search optimization
Step D: Model Update
  ○ Add the weighted base learner to the ensemble
  ○ Scale the contribution by the learning rate
  ○ Update the current model's predictions
3. Final Model Assembly:
  ○ Combine all base learners into the final ensemble
  ○ Each learner is weighted by their determined contribution
  ○ The learning rate scales all contributions
```

ALGORITHM 1: Gradient boosting regression (GBR).

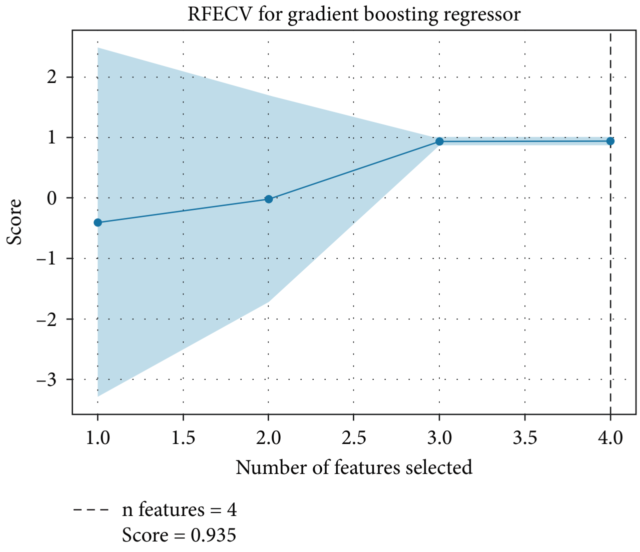


FIGURE 5: RFECV curve of the GBR model. The blue line represents the model's accuracy as the number of selected features varies, helping to identify the optimal number of features that yields the best performance.

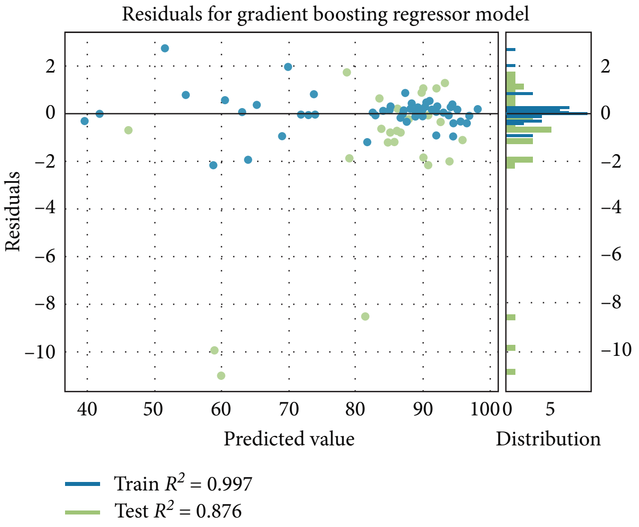
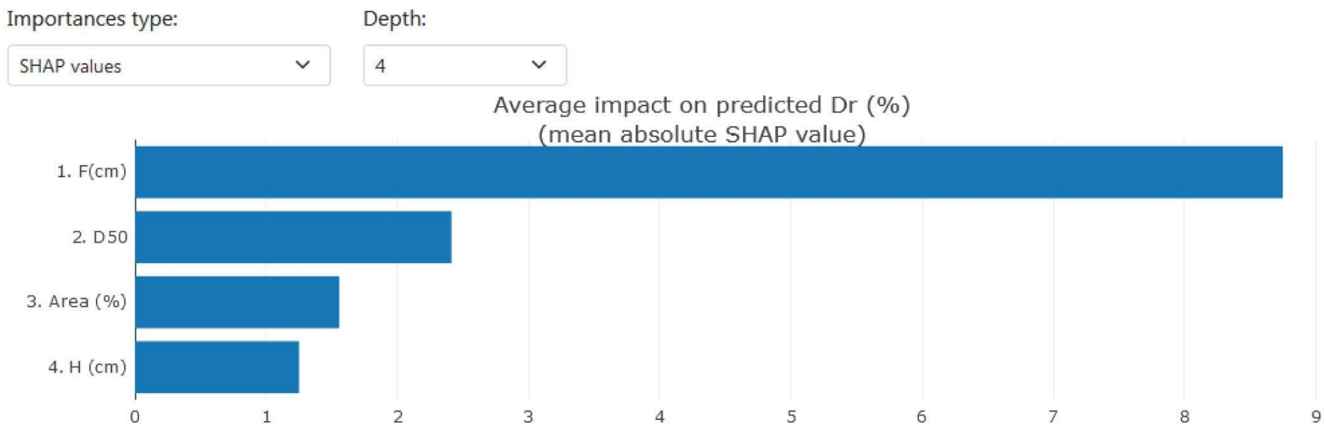


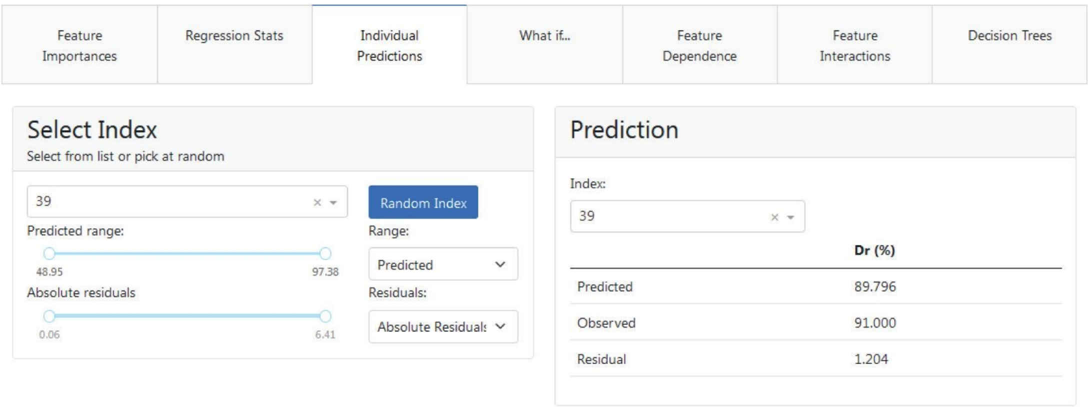
FIGURE 6: Regression indices of the tuned_GBR.



Importances type: SHAP values

Depth: 4

Figure 7: Sensitivity analysis of the tuned_GBR model.



Select Index

Select from list or pick at random

39

Predicted range: 48.95 – 97.38

Range: Predicted

Absolute residuals: 0.06 – 6.41

Residuals: Absolute Residuals

Prediction

Index: 39

	Dr (%)
Predicted	89.796
Observed	91.000
Residual	1.204

Figure 8: Predictions using the tuned_GBR model.

underscores the importance of "*F*" in laboratory settings, as changes in this distance directly affect the texture and packing structure of the deposited sand. Variations in "*F*" alter the energy and trajectory of falling sand grains, which influence their spatial arrangement and compaction, ultimately shaping the relative density and formation of the soil layer.

Following "*F*," the D50 parameter, the median particle size emerges as the next most significant feature. D50 represents the grain size at which 50% of the soil's mass is finer, serving as a key indicator of gradation and textural properties. An increase in D50 indicates coarser grains, which can lead to changes in packing density, permeability, and shear strength. Thus, the prominence of D50 in the model aligns with its influence on achievable relative density during sand deposition. The features area (%) and *H* (cm) also contribute to the model's predictions, though to a lesser extent, as reflected in their lower average SHAP values.

In the era of ML, merely developing predictive models is not sufficient without accompanying advancements in usability and practical application. As highlighted by the study, while ML models, such as the gradient boosting regressor, AdaBoost regressor, and XGBoost can significantly enhance the accuracy of soil density predictions, their effectiveness is limited if not coupled with user-friendly platforms. Without these practical tools, the full potential of ML models remains untapped in geotechnical engineering. The development of accessible interfaces, such as the web platform created in this study, is crucial for making these sophisticated models practical and beneficial for researchers and practitioners. For this matter, Figure 8 illustrates an online platform provided by the PyCaret library, designed for checking and evaluating individual predictions. This platform allows users to input data into the tuned GBR model and compare the model's output with the actual target values. For instance, as shown in Figure 8, test case number 39 was entered into the model, resulting in a predicted D_r value of 89.79%, while the observed value was 91%. This platform serves as a reliable and user-friendly tool for geotechnical researchers, facilitating accurate estimation of D_r and offering a straightforward interface for practical application.

5. Limitations and Future Work

This study has certain limitations, most notably the relatively small dataset of 84 tests, which may affect the generalizability of the developed ML models. Although the models demonstrated strong performance within the existing dataset, their effectiveness may vary when applied to other sand types or under real-world conditions. Additionally, our analysis primarily focused on ensemble learning algorithms, namely GBR, AdaBoost, and XGBoost. We acknowledge that a broader exploration of alternative ML approaches could further strengthen the robustness and applicability of our findings. Methods, such as support vector regression (SVR) and Gaussian Process Regression (GPR), which are well-suited for small datasets, may offer valuable insights for predicting relative density. Advanced frameworks like LightGBM and CatBoost, as well as neural models, such as small-scale multilayer perceptrons (MLPs) with appropriate regularisation, could also provide complementary performance metrics and interpretability. Systematic comparison with these techniques was beyond the scope of this study, but we recommend their inclusion in future research to provide a more comprehensive assessment of ML methods in geotechnical applications.

To address these limitations, future work should focus on expanding the dataset to include a broader range of sand types and experimental conditions, as well as incorporating additional influential features. Furthermore, evaluating the models with independent datasets and exploring more advanced ML techniques could further improve their performance, generalizability, and robustness.

6. Conclusions

This study presented a comprehensive investigation into predicting the relative dry density (D_r) of sand in laboratory settings through advanced ML techniques. The research addressed a critical challenge in geotechnical engineering by developing reliable and accurate prediction models for soil density estimation. Among the various ML algorithms tested, the gradient boosting regressor, AdaBoost regressor, and XGBoost models demonstrated exceptional predictive capabilities, with the GBR model emerging as the superior performer. The GBR model achieved remarkable accuracy metrics, including a MSE of 11.91, a MAE of 1.93, and a coefficient of determination (R^2) of 0.997, indicating its robust predictive power. Through detailed sensitivity analysis, the study identified that the distance between the shutter plate and the top sieve emerged as the most influential parameter affecting D_r predictions, providing valuable understanding for experimental setup optimisation. To bridge the gap between research and practical application, user-friendly web platforms were developed within the Google Colab interface, enabling geotechnical researchers to readily access and utilise these advanced prediction tools in their laboratory work. The platform's accessibility and ease of use represented a significant step forward in implementing ML techniques in geotechnical engineering practice. The study's findings not only demonstrated the potential of ML in enhancing the precision of soil density predictions but also provided practical tools and methodologies that contributed to advancing geotechnical engineering research and practice. This integration of advanced computational methods with traditional geotechnical engineering approaches marked a significant advancement in the field, offering new possibilities for more accurate and efficient soil characterisation methods.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

The authors declare that no funding was received for this research.

References

[1] Y. Yang, D. A. de Lange, H. Wang, and A. Askarinejad, "Multi-Scale Calibration of a ine-Style Sand Pluviator," *Geomechanics and Engineering* 37 (2024): 431–441.

[2] X. Y. Liu, A. Franza, and R. Jimenez, "Effects of Relative Density and Dilatancy on Stress and Deformation Arching of Sand over an Active Trapdoor," *Computers and Geotechnics* 173 (2024): 106485.

[3] A. Albatal, N. Stark, and B. Castellanos, "Estimating In Situ Relative Density and Friction Angle of Nearshore Sand From Portable Free-Fall Penetrometer Tests," *Canadian Geotechnical Journal* 57, no. 1 (2020): 17–31.

[4] R. A. Raja, V. A. Sakleshpur, M. Prezzi, and R. Salgado, "Effect of Relative Density and Particle Morphology on the Bearing Capacity and Collapse Mechanism of Strip Footings in Sand," *Journal of Geotechnical and Geoenvironmental Engineering* 149, no. 8 (2023): 04023052.

[5] D. Xu, Y. Gan, Y. Qin, W. Du, and H. Shen, "Effect of Gradation and Relative Density on Shear Strength of Coral Sand," *Geomechanics and Geoengineering* 19, no. 5 (2024): 754–771.

[6] R. Kuerbis and Y. P. Vaid, "Sand Sample Preparation-the Slurry Deposition Method," *Soils and Foundations* 28, no. 4 (1988): 107–118.

[7] M. Khari, K. A. Kassim, and A. Adnan, "Sand Samples' Preparation Using Mobile Pluviator," *Arabian Journal for Science and Engineering* 39, no. 10 (2014): 6825–6834.

[8] P. Dastpak, S. Abrishami, M. Rezazadeh Anbarani, and A. Dastpak, "Effect of Perforated Plates on the Relative Density of Uniformly Graded Reconstituted Sands Using Air Pluviation Method," *Transportation Infrastructure Geotechnology* 8, no. 4 (2021): 569–589.

[9] Z. Zhang and S. C. Chian, "Layering Effects of Sand Samples Prepared by Travelling Pluviators," *International Journal of Physical Modelling in Geotechnics* 22, no. 2 (2022): 88–98.

[10] O. Al-salih and I. Al-abboodi, "Assessment of a New Pluviation System Designed to Prepare Uniform Samples of Sand," *Journal of King Saud University - Engineering Sciences* (2023).

[11] B. Ghorbani, E. Yaghoubi, P. L. P. Wasantha, R. Van Staden, M. Guerrieri, and S. Fragomeni, "Machine Learning-Based Prediction of Resilient Modulus for Blends of Tire-Derived Aggregates and Demolition Wastes," *Road Materials and Pavement Design* 25, no. 4 (2024): 694–715.

[12] R. Wazirali, E. Yaghoubi, M. S. S. Abujazar, R. Ahmad, and A. H. Vakili, "State-of-the-Art Review on Energy and Load Forecasting in Microgrids Using Artificial Neural Networks, Machine Learning, and Deep Learning Techniques," *Electric Power Systems Research* 225 (2023): 109792.

[13] E. Yaghoubi, E. Yaghoubi, A. Khamees, D. Razmi, and T. Lu, "A Systematic Review and Meta-Analysis of Machine Learning, Deep Learning, and Ensemble Learning Approaches in Predicting EV

Charging Behavior," *Engineering Applications of Artificial Intelligence* 135 (2024): 108789.

[14] M. Khaleel, E. Yaghoubi, E. Yaghoubi, and M. Z. Jahromi, "The Role of Mechanical Energy Storage Systems Based on Artificial Intelligence Techniques in Future Sustainable Energy Systems," *International Journal of Electrical Engineering and Sustainability* 1, no. 4 (2023): 01–31, https://ijees.org/index.php/ijees/article/view/65.

[15] S. T. Amiri, A. Dehghanbanadaki, R. Nazir, and S. Motamedi, "Unit Composite Friction Coefficient of Model Pile Floated in Kaolin Clay Reinforced by Recycled Crushed Glass Under Uplift Loading," *Transportation Geotechnics* 22 (2020): 100313.

[16] H. Fattahi, H. Ghaedi, F. Malekmahmoodi, and D. J. Armaghani, "Optimizing Pile Bearing Capacity Prediction: Insights From Dynamic Testing and Smart Algorithms in Geotechnical Engineering," *Measurement* 230 (2024): 114563.

[17] M. Duan and X. Xiao, "Enhancing Soil Pile-Bearing Capacity Prediction in Geotechnical Engineering Using Optimized Decision Tree Fusion," *Multiscale and Multidisciplinary Modeling, Experiments and Design* 7, no. 3 (2024): 2861–2876.

[18] X. Yang, "Prediction of Pile-Bearing Capacity Using Least Square Support Vector Regression: Individual and Hybrid Models Development," *Multiscale and Multidisciplinary Modeling, Experiments and Design* 7, no. 3 (2024): 2701–2715.

[19] L. Cai, D. Zhu, and K. Xu, "The Implementation of a Machine-Learning-Based Model Utilizing Meta-Heuristic Algorithms for Predicting Pile Bearing Capacity," *Indian Geotechnical Journal* 55, no. 1 (2025): 210–225.

[20] T. F. Kurnaz, C. Erden, U. Dağdeviren, A. S. Demir, and A. H. Kökçam, "Comparison of Machine Learning Algorithms for Slope Stability Prediction Using an Automated Machine Learning Approach," *Natural Hazards* 120, no. 8 (2024): 6991–7014.

[21] K. Gao and L. Xu, "Novel Strategies Based on a Gradient Boosting Regression Tree Predictor for Dynamic Multi-Objective Optimization," *Expert Systems with Applications* 237 (2024): 121532.

[22] M. Lin, L. Zeng, S. Teng, G. Chen, and B. Hu, "Prediction of Stability of a Slope With Weak Layers Using Convolutional Neural Networks," *Natural Hazards* 120, no. 13 (2024): 12081–12105.

[23] T. Pei and T. Qiu, "Machine Learning With Monotonic Constraint for Geotechnical Engineering Applications: An Example of Slope Stability Prediction," *Acta Geotechnica* 19, no. 6 (2024): 3863–3882.

[24] R. Mustafa and M. T. Ahmad, "Internal Stability of Mechanically Stabilized Earth Wall Using Machine Learning Techniques," *Transportation Infrastructure Geotechnology* 11, no. 5 (2024): 3204–3234.

[25] A. T. Chala and R. Ray, "Assessing the Performance of Machine Learning Algorithms for Soil Classification Using Cone Penetration Test Data," *Applied Sciences* 13, no. 9 (2023): 5758.

[26] H. O. Ulloa, A. Ramirez, N. H. Jafari, S. Kameshwar, and I. Harrouch, "Machine Learning-Based Organic Soil Classification Using Cone Penetrometer Tests," *Journal of Geotechnical and Geoenvironmental Engineering* 150, no. 9 (2024): 05024008.

[27] A. Dehghanbanadaki, "Intelligent Modelling and Design of Soft Soil Improved With Floating Column-Like Elements as a Road Subgrade," *Transportation Geotechnics* 26 (2021): 100428.

[28] Q. Zhao and Y. Shi, "Prediction of Unconfined Compressive Strength of Stabilized Sand Using Machine Learning Methods," *Indian Geotechnical Journal* 55, no. 1 (2025): 315–332.

[29] I. Thapa and S. Ghani, "Enhancing Unconfined Compressive Strength Prediction in Nano-Silica Stabilized Soil: A Comparative Analysis of Ensemble and Deep Learning Models," *Modeling Earth Systems and Environment* 10, no. 4 (2024): 5079–5102.

[30] S. Sert, E. Arslan, P. Ocakbaşı, et al., "Stabilization of Expansive Clays With Basalt Fibers and Prediction of Strength by Machine Learning," *Arabian Journal for Science and Engineering* 49, no. 10 (2024): 13651–13670.

[31] I. T. Bahmed, J. Khatti, and K. S. Grover, "Hybrid Soft Computing Models for Predicting Unconfined Compressive Strength of Lime Stabilized Soil Using Strength Property of Virgin Cohesive Soil," *Bulletin of Engineering Geology and the Environment* 83, no. 1 (2024): 46.

[32] A. Dehghanbanadaki, A. S. A. Rashid, K. Ahmad, N. Z. M. Yunus, and K. N. M. Said, "A Computational Estimation Model for the Subgrade Reaction Modulus of Soil Improved With DCM Columns," *Geomechanics and Engineering* 28, no. 4 (2022): 385–396.

[33] S. Alioua, A. Arab, M. A. Benbouras, and A. Leghouchi, "Modeling Static Liquefaction Susceptibility of Saturated Clayey Sand Using Advanced Machine-Learning Techniques," *Transportation Infrastructure Geotechnology* 11, no. 5 (2024): 2903–2931.

[34] D. R. Kumar, P. Samui, A. Burman, and S. Kumar, "Seismically Induced Liquefaction Potential Assessment by Different Artificial Intelligence Procedures," *Transportation Infrastructure Geotechnology* 11, no. 3 (2024): 1272–1293.

[35] X. Feng, J. He, and B. Lu, "Accurate and Generalizable Soil Liquefaction Prediction Model Based on the CatBoost Algorithm," *Acta Geophysica* 72, no. 5 (2024): 3417–3426.

[36] L. Qu, H. Lu, Z. Tian, J. M. Schoorl, B. Huang, and Y. Liang, "Spatial Prediction of Soil Sand Content at Various Sampling Density Based on Geostatistical and Machine Learning Algorithms in Plain Areas," *CATENA* 234 (2024): 107572.

[37] E. Yaghoubi, E. Yaghoubi, A. Khamees, and A. H. Vakili, "A Systematic Review and Meta-Analysis of Artificial Neural Network, Machine Learning, Deep Learning, and Ensemble Learning Approaches in Field of Geotechnical Engineering," *Neural Computing and Applications* 36, no. 21 (2024): 12655–12699.

[38] M. Ali, "Pycaret: An Open Source, Low-Code Machine Learning Library in Python," (2020): 2PyCaret Version 2020 (2020).

[39] B. M. Das and N. Sivakugan, *Fundamentals of Geotechnical Engineering* (Cengage Learning, 2016).

[40] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A Comparative Analysis of Gradient Boosting Algorithms," *Artificial Intelligence Review* 54, no. 3 (2021): 1937–1967.

[41] C. Xu, K. Zhou, X. Xiong, F. Gao, and Y. Lu, "Prediction of Mining Induced Subsidence by Sparrow Search Algorithm With Extreme Gradient Boosting and TOPSIS Method," *Acta Geotechnica* 18, no. 9 (2023): 4993–5009.

[42] N. Bozorgzadeh and Y. Feng, "Evaluation Structures for Machine Learning Models in Geotechnical Engineering," *Georisk* 18, no. 1 (2024): 52–59.

[43] S. Demir and E. K. Sahin, "Predicting Occurrence of Liquefaction-Induced Lateral Spreading Using Gradient Boosting Algorithms Integrated With Particle Swarm Optimization: PSO-XGBoost, PSO-LightGBM, and PSO-CatBoost," *Acta Geotechnica* 18, no. 6 (2023): 3403–3419.

[44] E. Díaz and G. Spagnoli, "Gradient Boosting Trees With Bayesian Optimization to Predict Activity From Other Geotechnical Parameters," *Marine Georesources and Geotechnology* 42, no. 8 (2024): 1075–1085.

[45] S. Lin, Z. Liang, S. Zhao, M. Dong, H. Guo, and H. Zheng, "A Comprehensive Evaluation of Ensemble Machine Learning in Geotechnical Stability Analysis and Explainability," *International Journal of Mechanics and Materials in Design* 20, no. 2 (2024): 331–352.

[46] E. Ghorbani and S. Yagiz, "Estimating the Penetration Rate of Tunnel Boring Machines via Gradient Boosting Algorithms," *Engineering Applications of Artificial Intelligence* 136 (2024): 108985.