



A comparative analysis of ensemble learning algorithms with hyperparameter optimization for soil liquefaction prediction

Alparslan Serhat Demir¹ · Talas Fikret Kurnaz² · Abdullah Hulusi Kökçam¹ · Caner Erden^{3,5} · Uğur Dağdeviren⁴

Received: 1 October 2023 / Accepted: 8 April 2024 / Published online: 2 May 2024
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

Accurate prediction of soil liquefaction potential is crucial for evaluating the stability of structures in earthquake regions. This study focuses on predicting soil liquefaction using a dataset that included historical liquefaction cases from the 1999 Turkey and Taiwan earthquakes. The dataset was divided into three subsets: Dataset A (fine-grained), Dataset B (coarse-grained), and Dataset C (all samples). Through the analysis of these subsets, the study aims to assess the performance of machine learning algorithms in predicting soil liquefaction potential. This study applied ensemble machine learning algorithms, including extreme gradient boosting, adaptive boosting, extra trees, bagging classifiers, light gradient boosting machine, and random forest, to accurately classify the liquefaction potential of fine-grained and coarse-grained soils. A comparison between the genetic algorithm approach for hyperparameter optimization and traditional methods such as grid search and random search revealed that genetic algorithms outperformed both in terms of average test and train accuracy. Specifically, the light gradient boosting machine yielded the best predictions of soil liquefaction potential among the algorithms tested. The study demonstrated that Dataset B achieved the highest learning performance with accuracy of 0.92 on both the test and training sets. Furthermore, Dataset A showed a training accuracy of 0.88 and a test accuracy of 0.84, while Dataset C exhibited a training accuracy of 0.87 and a test accuracy of 0.87. Future studies could build on these findings by evaluating the performance of genetic algorithms on a wider range of machine learning algorithms and datasets, thus advancing our understanding of soil liquefaction prediction and its implications for geotechnical engineering.

Keywords Soil liquefaction · Soil types · Hyperparameter optimization · Ensemble learning · Machine learning

Introduction

Liquefaction is one of the most important, interesting, complex, and controversial topics in geotechnical engineering. The great destruction caused by liquefaction in the Alaska earthquakes ($M_w = 9.2$) and Niigata ($M_s = 7.5$) that occurred in 1964 increased the interest of geotechnical engineers in this phenomenon. The term liquefaction describes a set of soil deformations that occur when saturated cohesionless soils are disturbed in undrained conditions in static, temporary, or cyclical ways (Kramer 1996). Although liquefaction was thought to occur only in sandy soils for many years, studies and observations have shown that liquefaction may occur in low-cohesive silts (Ishihara 1984, 1985) and gravelly soils (Evans and Seed 1987; Yegian et al. 1994). The emergence of liquefaction in the soil layers in earthquakes

can cause considerable damage to structures on the ground and underground structures. Therefore, determining the factors that cause liquefaction in soils, liquefaction hazards, and predicting possible harmful effects are considered among important research topics in geotechnical earthquake engineering.

Local ground conditions significantly affect the structural damage that can occur during earthquakes. Thus, it is important to predict the behavior of soils under cyclic loads and their static strength after earthquakes. It is possible to determine the stress–strain behavior of soils during and after earthquakes with various laboratory test systems. In particular, the liquefaction potentials and the post-liquefaction behavior of saturated sandy soils under cyclic loads can be investigated in the laboratory with dynamic test systems such as dynamic simple shear, dynamic triaxial, dynamic torsion test, and shaking table (Das 1993; Youd and Idriss 2001; Xue and Xiao 2016; Rahbarzare and Azadi 2019; Erzin and

Extended author information available on the last page of the article

Tuskan 2019; Alizadeh Mansouri and Dabiri 2021; Pham 2021; Zhou et al. 2022b). However, obtaining undisturbed samples from soils whose liquefaction risk is investigated, and the modeling of the field conditions in laboratory experiments may present some difficulties. Therefore, it is widely preferred to perform liquefaction risk analyses using methods based on field tests, such as the standard penetration test (SPT), cone penetration test (CPT), and measurement of shear wave velocity (V_s). The most significant advantage of field tests is that they allow evaluating the soil in its natural state. Experiments and identification can be made at the desired depth from the surface. However, since seismic activity is difficult to demonstrate in the field, soil parameters related to liquefaction strength cannot be obtained in experiments to determine the liquefaction potential. To reach the evaluation criteria, empirical relations have been developed by examining past earthquakes where liquefaction has occurred (Law and Wang 1994).

Methods based on SPT test results are widely used to determine the liquefaction potential among field tests (Ghani and Kumari 2022a; Ghani et al. 2022; Yilmaz et al. 2022). The most important of these is the “Simplified Method” published in 1971 (Seed and Idriss 1971) and has also been accepted as the standard in liquefaction analysis. This method is based on the principle of obtaining a factor of safety (FSL) by dividing the ratio of cyclic resistance to liquefaction (CRR) by the earthquake-induced cyclic stress (CSR). If the factor of safety is less than 1, liquefaction is expected. If it is greater than 1, there is no risk of liquefaction.

On the other hand, the fact that the SPT test contains many uncertainties in the application procedure necessitates correction in the obtained SPT-N values (Youd et al. 2001; Idriss and Boulanger 2010; Alizadeh Mansouri and Dabiri 2021). In addition to those that can be determined by laboratory and field tests, many parameters, such as earthquake magnitude and peak horizontal ground acceleration, are used in liquefaction analyses based on the SPT test. Although CSR and CRR parameters can be determined using empirical expressions developed based on past earthquakes (Youd et al. 2001; Cetin et al. 2004; Idriss and Boulanger 2008, 2010; Boulanger and Idriss 2014) and field observations, it is difficult to obtain a single appropriate empirical expression due to the need for many parameters and uncertainties (Cai et al. 2022; Zhou et al. 2022b). Although the deterministic methods mentioned above are still widely used in the evaluation of liquefaction, as they are simple and practical, they might contain significant uncertainties due to the variables used in these methods. In addition to the uncertainties related to the anisotropy and non-uniform properties resulting from the heterogeneous structure of the soils, the uncertainties in the seismic parameters caused by earthquakes are the major ones. Although some researchers propose

reliability methods for deterministic approaches used in liquefaction evaluations to overcome these uncertainties (Baecher and Christian 2005; Ghani and Kumari 2022b; Golmoghani Ebrahimi et al. 2023; Jha and Suzuki 2009; Johari et al. 2012; Juang et al. 2009; Kwak and Lee 1987), some researchers have also emphasized that it is inevitable to use reliable, simple, and predictable flexible calculation techniques to evaluate the liquefaction potential of any piece of soil, as in many fields of engineering science (Xue and Yang 2013, 2016; Xue and Xiao 2016; Juang et al. 2022).

Examining the literature in question, it is seen that different researchers have used different types of artificial intelligence (AI) and machine learning (ML) techniques, such as artificial neural network (ANN) (Hanna et al. 2007; Ramakrishnan et al. 2008; Mughieda et al. 2009; Xue and Liu 2017; Alizadeh Mansouri and Dabiri 2021; Kurnaz et al. 2023), adaptive neuro-fuzzy inference system (ANFIS) (Rahman and Wang 2002; Chern et al. 2008; Xue and Yang 2013), support vector machine (SVM) and relevance vector machine (RVM) (Samui and Sitharam 2011; Samui and Karthikeyan 2013; Rahbarzare and Azadi 2019) to overcome the uncertainties in the assessment of soil liquefaction risk. Additionally, optimization techniques have started to be used in liquefaction risk evaluations to improve the generalization ability of ML models and overcome limitations in recent years (Rahbarzare and Azadi 2019; Zhang et al. 2021; Zhao et al. 2021; Cai et al. 2022; Ghani and Kumari 2022a; Ghani et al. 2022; Umar et al. 2022; Zhou et al. 2022a), have been implemented in most complex geotechnical engineering problems with high success (Díaz et al. 2022; Li et al. 2022; Rehman et al. 2022; Wang et al. 2022).

Classification-based estimation studies also have a major role in the liquefaction phenomenon with AI and ML. Using historical case datasets, researchers have successfully used classifiers of different methods to predict the liquefaction potential. Ahmad et al. (2019) focused on case histories with Bayesian belief network (BBN) and C4.5 decision tree (DT) models to assess the liquefaction potential. They achieved satisfactory results, especially with the BBN model. Alobaidi et al. (2019) used ensemble models to predict the liquefaction potential and compared them with ML models. They have claimed that the ensemble models perform better in overcoming the uncertainties and predicting. Rahbarzare and Azadi (2019) studied CPT-based field cases and proposed a fuzzy support vector machine (FSVM) classifier supported by optimization algorithms for liquefaction prediction, and they achieved satisfactory results. Zhao et al. (2021) studied two case history databases. They proposed a hybrid ML model that includes particle swarm optimization and kernel extreme learning machine (PSO-KELM) to evaluate the liquefaction potential. They emphasized that the proposed model has achieved better results than the various ML models. Ahmad et al. (2021) used different BBN models

in CPT field cases to assess the liquefaction potential, and gave the results comparatively. Zhang et al. (2021) focused on the addition of the V_s parameter to the prediction models of the liquefaction potential based on the SPT field cases. They used a multi-layer fully connected network model optimized by the deep neural network (ML-FCN DNN). Thus, they stated that they achieved higher accuracy prediction success. Kumar et al. (2021) proposed a deep learning (DL) model for the prediction of classification-based liquefaction, and the results they achieved were compared with the emotional backpropagation neural network (EmBP), and highlighted the success of the proposed DL model. Zhang et al. (2021) proposed an SVM model optimized by gray wolf optimization (GWO) to present a prediction model of the liquefaction potential and suggested that GWO improved the prediction success. Zhang and Wang (2021) used three different datasets. They proposed an ensemble model that combined the predictions of the seven base classifiers in the prediction of liquefaction and achieved reliable results. Zhou et al. (2022a) proposed SVM models optimized by GWO and genetic algorithm (GA) using three historical datasets. The results indicated that GWO and GA have increased SVM performance; however, GWO was slightly more successful. Zhou et al. (2022b) used three historical datasets to propose random forest (RF) models optimized by GWO and GA. The results showed that the optimization of GWO and GA has improved the performance of the single RF model. Additionally, GWO-RF was more successful in two datasets and GA-RF was more successful in one. Demir and Sahin (2023) presented Adaptive Boosting (AdaBoost), Gradient Boosting Machine (GBM) and eXtreme Gradient Boosting (XGBoost) ensemble algorithms to evaluate soil liquefaction potential in SPT-based dataset. The results suggest that the proposed ensemble models obtained reliable results in liquefaction prediction. Kumar et al. (2022) used XGBoost, RF, GBM, support vector regression (SVR), and group method of data handling models in the prediction of soil liquefaction using a SPT-based database. Although all models were successful, the XGBoost model showed the best prediction performance for the liquefaction potential.

This study focused on a thorough investigation of the potential of ML algorithms, particularly classifiers, in accurately predicting soil liquefaction. In this regard, the dataset containing 620 SPT-based historical cases presented by Hanna et al. (2007) was used. It is seen that a significant part of the soil types in the dataset presented by Hanna et al. (2007) are classified as fine-grained soils. However, the dataset includes only the parameters used in coarse-grained soil liquefaction analysis methods based on SPT as input parameters. Nevertheless, the input parameters in the database do not include parameters that reflect the physical properties of soils that are known to be effective in the liquefiability of fine-grained soils. Since the development of

the pore water pressure of clay–silt mixtures during cyclic loads is quite different from that of sand soils, the liquefaction criteria developed for sands will not function efficiently for fine-grained soils. In terms of stress–strain behavior, the behavior of sands depends significantly on confining stress and relative density, while the behavior of clays depends on plasticity, consolidation stress, and stress history. Therefore, Boulanger and Idriss (2006) separated fine-grained soils into categories such as "sand-like" and "clay-like" according to their behavior. Over time, many criteria have been developed to determine the susceptibility to liquefaction of fine-grained soils based on the physical properties of the soil. Early on, the criteria proposed to separate liquefiable and non-liquefiable fine-grained soils were based on the percentage of clay (C%), liquid limit (LL), and the ratio of natural water content to liquid limit (w_n/LL) (Wang et al. 2022; Seed and Idriss 1971; Andrews and Martin 2000). In the following years, the average grain size (D_{50}) (Bol et al. 2010), and the soil plasticity index (PI) (Bray and Sancio 2006; Ghani and Kumari 2021; Seed et al. 2003) were used for liquefaction sensitivity.

The dataset presented by Hanna et al. (2007) has been widely used in many studies that propose prediction models on liquefaction evaluation (Gandomi et al. 2013; Hoang and Bui 2018; Kayadelen 2011; Kurnaz and Kaya 2019; Zhang and Wang 2021; Zhang et al. 2021; Zhao et al. 2021; Zhou et al. 2022a, 2022b). In the mentioned studies, the performance results of the models were obtained by performing analyzes with parameters affecting the liquefaction behavior of coarse-grained soils, without considering the presence of fine-grained soil types in the dataset. Considering that this situation may affect the performance of the proposed forecasting models, in addition to the entire dataset, the dataset was divided into fine and coarse-grained soils, and three different datasets were analyzed in the current study. A variety of powerful ML ensemble algorithms were used in the analysis, including XGBoost, AdaBoost, Extra Trees Algorithm (EXT), Bagging Classifier (Bagging), Light Gradient Boosting (LightGBM), and Random Forest (RF).

The main contribution of this paper is highlighted as follows:

- (a) Unlike previous studies that used the dataset presented by Hanna et al. (2007), prediction models were developed for different variations considering the difference in soil type within the dataset, and it is emphasized that grouping the dataset according to soil type positively affects the prediction model performance results.
- (b) The ML models' performance was enhanced through the utilization of optimization techniques, including random search (RS), grid search (GS), and GA.
- (c) By integrating ensemble methods and fine-tuning the hyperparameters, this study aims to provide valuable

insights into the practical application of these robust algorithms and their unparalleled effectiveness in precisely classifying the liquefaction potential across various soil types.

Dataset description

In this study, the dataset presented by Hanna et al. (2007), which contains a total of 620 SPT-based case records of two major earthquakes that occurred in 1999 in Türkiye ($M_w = 7.4$) and Taiwan ($M_w = 7.6$), was used in three different variations. Since the whole dataset contains different soil types, as mentioned in the Introduction part of this paper, the dataset was divided into two categories: fine-grained and coarse-grained soils. Three different datasets were analyzed in this study. The first dataset (A) consists of fine-grained soils only, the second dataset (B) consists of coarse-grained soils only, and the third dataset (C) consists of all samples. Among the case records in the entire dataset, 406 data are classified as fine-grained soil (dataset A) and 214 data are classified as coarse-grained soil (dataset B). All three datasets consist of 12 input parameters

and a single output parameter. The input parameters are the depth of the analyzed layer (z), the corrected SPT blow numbers ($N_{1,60}$), the percentage of fine content less than $75 \mu\text{m}$ (FC (%)), the depth of the groundwater table (d_w), total and effective overburden stresses (σ_v, σ_v'), threshold acceleration (a_t), cyclic stress ratio (CSR), shear wave velocity (v_s), internal friction angle of the soil (ϕ), earthquake magnitude (M_w), maximum ground acceleration (a_{max}). The output parameter of the binary classification problem is the liquefaction potential. The descriptive statistics for the dataset used in this study are presented in Table 1.

Additionally, Table 2 presents the correlation coefficients between the distinctive characteristics analyzed. The values range from -1 to 1, with higher absolute values indicating stronger correlations. As expected, σ_v and σ_v' exhibit a high correlation, with a coefficient of 0.98. $N_{1,60}$ also shows a positive correlation with ϕ , with a coefficient of 0.84. Furthermore, CSR exhibits a positive correlation with a_{max} , with a coefficient of 0.90. The presented correlations can help identify important relationships between distinctive features and can be utilized in predicting the values of certain variables based on others.

Table 1 Descriptive statistics for variables related to soil liquefaction

	z	$N_{1,60}$	F	d_w	σ	σ'	a_t	CSR	v_s	ϕ_0	M_w	a_{max}
Count	620	620	620	620	620	620	620	620	620	620	620	620
Mean	7.66	14.48	62.99	1.45	144.60	82.48	0.07	0.37	166.98	31.96	7.49	0.38
Std	4.90	11.39	34.28	1.20	98.20	52.84	0.07	0.15	67.09	4.85	0.10	0.15
Min	0.80	1.00	1.00	0.35	12.10	7.50	0.00	0.12	37	23.46	7.40	0.18
25%	3.80	7.00	29.00	0.71	67.90	41.60	0.04	0.25	130	28.40	7.40	0.38
50%	6.70	11.00	74.50	1.10	121.60	68.15	0.06	0.39	155	31.41	7.40	0.40
75%	10.20	18.00	96.00	1.78	202.45	113.93	0.08	0.45	200	34.70	7.60	0.40
Max	19.80	75.00	100.00	10.00	408.90	233.70	0.85	0.77	500	52.08	7.60	0.67

Table 2 Correlation matrix of input and output parameters

	z	$N_{1,60}$	F	d_w	σ	σ'	a_t	CSR	v_s	ϕ_0	a_{max}	M_w	Liq
z	1.00	0.39	-0.25	0.07	1.00	0.97	-0.24	-0.11	0.58	0.54	0.51	-0.08	-0.26
$N_{1,60}$	0.39	1.00	-0.55	0.10	0.41	0.42	0.01	0.16	0.40	0.84	0.14	0.19	-0.25
F	-0.25	-0.55	1.00	-0.15	-0.26	-0.29	-0.10	-0.07	-0.31	-0.52	-0.27	-0.16	-0.09
d_w	0.07	0.10	-0.15	1.00	0.06	0.21	0.11	-0.26	0.14	0.11	0.17	0.00	-0.13
σ	1.00	0.41	-0.26	0.06	1.00	0.98	-0.23	-0.08	0.60	0.55	0.52	-0.04	-0.26
σ'	0.97	0.42	-0.29	0.21	0.98	1.00	-0.20	-0.09	0.61	0.57	0.55	0.00	-0.25
a_t	-0.24	0.01	-0.10	0.11	-0.23	-0.20	1.00	-0.07	0.43	-0.03	-0.01	0.09	-0.08
CSR	-0.11	0.16	-0.07	-0.26	-0.08	-0.09	-0.07	1.00	-0.03	0.09	-0.20	0.90	0.27
v_s	0.58	0.40	-0.31	0.14	0.60	0.61	0.43	-0.03	1.00	0.48	0.35	0.05	-0.18
ϕ_0	0.54	0.84	-0.52	0.11	0.55	0.57	-0.03	0.09	0.48	1.00	0.27	0.13	-0.33
M_w	0.51	0.14	-0.27	0.17	0.52	0.55	-0.01	-0.20	0.35	0.27	1.00	-0.11	-0.12
a_{max}	-0.08	0.19	-0.16	0.00	-0.04	0.00	0.09	0.90	0.05	0.13	-0.11	1.00	0.20
Liq	-0.26	-0.25	-0.09	-0.13	-0.26	-0.25	-0.08	0.27	-0.18	-0.33	-0.12	0.20	1.00

Background of ensemble learning methods

The basic working mechanism of ensemble learning methods (ELM) is based on multiple learning models rather than the individual learning model in ML. ELM combines predictive values obtained with multiple learners to produce a single predictive value. In this way, ELM reduces variance and bias in estimation results compared to individual ML, which are mostly divided into boosting, bagging, and stacking (Mienye and Sun 2022). Boosting methods use data from the wrong test results of the learning model as input in the next stage until a successful prediction is obtained. The stages continue until a certain stopping condition is satisfied. However, methods that prefer the bagging approach obtain predictions with the learning model by taking different subsets from the training set. These estimation results are aggregated as a result of a single estimation by taking max voting or average (Dietterich 2000). The final predictions are obtained through a meta-model using different learning models' predictions in stacking. Under the heading of boosting, bagging, and stacking, many different methods have been developed so far, and their applications in various fields, such as landslide assessment, predicting water quality, have been compared and discussed (Dou et al. 2020; Almadani and Kheimi 2023). In this part of the study, the methods of XGBoost, AdaBoost, EXT, Bagging, LightGBM, and RF, which achieved satisfactory results in the application phase for the estimation of soil liquefaction classification, are explained.

Extreme gradient boosting

XGBoost is an improvised version of the gradient boosting algorithm, and the working procedure for both is the same. Its implementation of parallel processing at the node level makes it more powerful and faster than the gradient boosting algorithm (Walia 2021). XGBoost is used for supervised learning problems that try to predict the y_i variable from a x_i multi-feature training set. The objective function consists of two parts, the training loss $L(\theta)$ and the regulation term $\Omega(\theta)$ (Eq. 1):

$$obj(\theta) = L(\theta) + \Omega(\theta). \quad (1)$$

$L(\theta)$ represents how predictive the model is based on the training data. Several types of loss functions can be used. The regulation term $\Omega(\theta)$ controls the complexity of the model. It helps to make predictions with high accuracy by simplifying the model. In this way, it ensures that the model avoids overfitting. The trade-off between the training loss $L(\theta)$ Moreover, the regulation term $\Omega(\theta)$ is also called the bias-variance trade-off in ML (Chen and Guestrin 2016). In

the study, the hyperparameters of XGBoost, including the step size shrinkage (η), minimum loss reduction (γ), maximum depth of a tree (\max_depth), minimum sum of instance weight (\min_child_weight), maximum delta step (\max_delta_step), subsample ratio ($subsample$), and the number of runs XGBoost will try to learn ($n_estimators$), were identified as the parameters to be optimized. Some recent studies have emphasized that the various parameters of the XGBoost algorithm give superior results when optimized and adjusted for different problems (Abdu-Aljabar and Awad 2022; Janizadeh et al. 2022).

Adaptive boosting

AdaBoost's main difference from other ELM is that it adaptively adjusts the errors of weak hypotheses returned by weak learners (WeakLearn) (Freund and Schapire 1997). WeakLearn predictions are combined using a weighted majority vote (or sum) to produce the final prediction. In each successful iteration, the weights (w_1, w_2, \dots, w_N) assigned to the training samples are modified, and the learning algorithm is applied again. The weights of the incorrectly predicted training samples in the previous step are increased in the boosted model. Thus, WeakLearn is forced to guess correctly the wrongly guessed examples (Pedregosa et al. 2011). According to research by Rajasekar et al. (2022) and Amini et al. (2023), AdaBoost has shown promising performance in predicting problems compared to other algorithms, particularly when hyperparameterized.

Extra-trees algorithm

EXT is named after Extremely Randomized Trees. It is an ELM based on DTs. It acts randomly like RF while making certain decisions. It is similar to RF in creating multiple trees and splitting nodes using random subsets of features. However, it does not bootstrap the observations. It also randomly splits the nodes instead of the best ones (Geurts et al. 2006). This method applies a meta-estimator that fits a set of random DTs (a.k.a. EXT) to various sub-samples of the dataset. It uses averaging to improve prediction accuracy and overfitting (Pedregosa et al. 2011). Few studies emphasize that if the EXT algorithm is hyperparameterized in different application areas, it can greatly affect the results. One of these studies, Cuocolo et al. (2020), emphasized that the hyperparametric EXT classifier can achieve good results in predicting the surgical consistency of pituitary adenoma. In this study, \max_depth , the number of trees in the forest ($n_estimators$), the minimum number of samples required to split ($\min_samples_split$), the minimum number of samples required to be at a leaf node ($\min_samples_leaf$), whether

bootstrap samples are used when building trees (bootstrap), and the function to measure the quality of a split (criterion) parameters are tuned.

Bagging classifier

Bagging methods work well with powerful and complex models, such as advanced DTs, compared to boosting methods. Bagging methods use the randomness of an ensemble to reduce the variance of the weak estimator. It is a class of ensemble algorithms that obtain individual predictions from random subsets of the original training set via a black-box estimator and aggregate them to form the final prediction. It performs this aggregation using voting or averaging and generates the final prediction. This algorithm also includes several studies in the literature known as Pasting, Random Subspaces, and Random Patches (Pedregosa et al. 2011). Few studies in the literature use the bagging classifier for hyperparameterization. Sari and Maki (2023) emphasized that they obtained superior results using the face mask detection dataset over the Kaggle face mask detection on Kaggle by optimizing the bagging classifier using a hybrid bat algorithm. In this study, the Bagging method's hyperparameters were optimized, which involved tuning the parameters such as the number of samples to draw (max_samples), the number of base estimators in the ensemble (n_estimators), whether samples are drawn with replacement (bootstrap), and whether features are drawn with replacement (bootstrap_features).

Light gradient boosting

Many boosting tools use pre-sort-based algorithms for DT learning, while LightGBM uses histogram-based algorithms. The main advantages of using histogram-based algorithms are that it reduces the amount of memory usage in calculation processes and shortens the training time (Ke et al. 2017). This situation increases the usefulness of the LightGBM technique, especially in problems where the number of features is high, and the data volume is high. The LightGBM method uses gradient-based one-sided sampling (GOSS) and exclusive feature bundle (EFB) approaches to overcome the performance degradation of other methods in problems with many features and large data sizes. With GOSS, most data samples with small gradients are not used, focusing on a specific part of the dataset. With EFB, mutually exclusive features are bundled to reduce the number of features (Ke et al. 2017). These two techniques significantly reduce the solution time and processor requirements. Numerous studies emphasize that satisfactory results are obtained when the LightGBM method is hyperparametrized for different problems (Dhar 2022; Pei et al. 2022). In this study, the hyperparameters of the LightGBM method, including learning_rate,

the complexity of the tree model (num_leaves), max_depth, minimum number of data points needed in a child node (min_child_samples), subsample, and column sample by tree (colsample_bytree), were optimized.

Random forest

In RF, each tree in the ensemble is generated from a bootstrap sample taken with replacement from the training set. Additionally, when each node is constructed during the tree, the best split is determined from all the input features or a random subset of the features. Similarly, to other classifiers, the RF classifier utilizes two types of dataset that store the attributes and target values of the training samples. It uses randomness in generating classifiers, and the ensemble is estimated by combining the average predictions of the individual classifiers. The primary purpose of incorporating randomness is to mitigate the variance within the forest estimator. However, this effort to reduce variance might slightly increase bias (Pedregosa et al. 2011). It has been emphasized that using the RF method and population-based algorithms such as the GA and Harris Hawks optimization algorithm can obtain satisfactory results in various estimation problems if its parameters are tuned (Daviran et al. 2023; Ge et al. 2023). In this study, the hyperparameters of the RF algorithm, including max_depth, min_samples_split, min_samples_leaf, bootstrap, and criterion, were optimized.

Performance evaluation criteria for ensemble learning methods

Performance evaluation is a crucial step in developing and deploying machine learning models and algorithms. It allows us to assess the effectiveness and reliability of our models, determine their strengths and weaknesses, and make informed decisions about their performance in real world scenarios. Performance evaluation quantitatively measures how well a machine learning model or algorithm performs on a given task or dataset. Various performance metrics are available to evaluate a model's performance, and the choice depends on the nature of the task and the type of model. Common metrics include accuracy, representing the ratio of correct predictions to the total; precision, measuring the proportion of true positives among positive predictions; recall, indicating the ratio of true positives to all actual positives; and F1 score, a harmonic mean of precision and recall for balanced assessment. It is crucial to select the appropriate metric based on the task at hand and interpret the results contextually for an effective model assessment (Gong et al. 2023).

In our study, we evaluate the performance of the classification models mentioned using commonly accepted metrics, namely accuracy, which is a very basic and intuitive

performance measure. During binary classification, some samples may be misclassified into the opposite class. The accuracy score indicates how well the model classifies both positive and negative samples. Accuracy is calculated as the proportion of true predictions, both true positives (TP) and true negatives (TN), from the total number of samples, which are given below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2)$$

where TP are correctly predicted positive samples, TN are correctly predicted negative samples, FP are negative samples incorrectly predicted as positive, and FN are positive samples predicted as negative. High accuracy indicates that the model is effective in classifying samples into the right classes overall. However, precision and recall provide more detailed information on how well it identifies positive classes. We use these standard evaluations to evaluate the different models developed for our binary classification task and identify the most suitable approach.

Hyperparameter optimization and traditional methods

The performance of ML algorithms in terms of variance can vary depending on the parameters used. Although some parameters are determined internally by the algorithm during the training process, others are set by the user (Agrawal 2021). This multiplicity of parameters poses an optimization problem known as hyperparameter optimization, which is separate from the core problem being solved. Hyperparameter optimization is a challenging task that requires users to explore different parameter values through trial and error. To address this, various methods have been developed. Two commonly used methods in the field are GS and RS, which are considered traditional methods (Lim 2022). GS involves systematically evaluating all combinations of parameter values, which can be time-consuming and resource-intensive. On the other hand, RS reduces solution time and resource consumption by randomly sampling parameter values. However, this randomness can sometimes lead RS to deviate from the optimal solution compared to GS.

Genetic algorithm

GA is a widely used AI technique to deliver satisfactory results in a short time frame. It was initially developed by Holland in 1975 (Holland 1992) and is based on the principles of Darwin's theory of evolution. The algorithm follows a solution generation and evaluation mechanism to improve results by favoring well-performing individuals. GA incorporates crossover and mutation operators to create new

individuals. The process continues iteratively until a specific stopping condition is met. Through the use of these operators, poor-performing individuals in the initial population and among the newly generated individuals can be identified and, if necessary, removed or improved upon, contributing to the refinement of the population. GA has demonstrated success in numerous domains, including energy, health, industry, and earth science, and has been applied to a wide range of problem types, such as estimation, optimization, and classification (Jalal et al. 2021; De and Kundu 2022; Sharma et al. 2022; Kummer et al. 2023).

Given the complexity and multidimensionality of the optimization problem at hand, GAs were selected as the primary optimization method due to their qualities, such as versatility, effectiveness, and efficiency. GAs have inherent robustness to noise and outliers in the data, ensuring stable performance even under challenging conditions. Moreover, their parallelizability allows for significant acceleration of the optimization process, particularly for large-scale problems or computationally expensive fitness functions. Additionally, GAs can seamlessly handle a mix of categorical, continuous, and integer parameters, making them well suited for optimizing machine learning (ML) models with diverse parameter types. The combination of robustness, parallelizability, and the ability to handle various parameter types makes GAs an ideal choice for optimizing ML models in the context of this study.

Results

In this study, an Intel(R) Core(TM) i7-5600U CPU @ 2.60 GHz and a suite of Python libraries, encompassing numpy for efficient mathematical computations, pandas for seamless data manipulation, matplotlib and seaborn for informative data visualization, sklearn for comprehensive ML tasks, xgboost and lightgbm for boosting algorithms, and sklearn_genetic and hyperopt for hyperparameter optimization, were employed to augment soil liquefaction prediction in geotechnical engineering. The overall study workflow is depicted in Fig. 1.

Data preprocessing and model training strategy

The dataset utilized in this study was robustly scaled using the robust scaling technique. This scaling approach is highly suitable for datasets characterized by outliers or a non-normal distribution. By employing the Robust Scaler, the data were transformed by subtracting the median and dividing it by the interquartile range, effectively capturing the data's spread. Furthermore, to manage categorical data, the M_w and a_{max} features re-encoded using one-hot encoding, which allows for the representation of categorical information in a

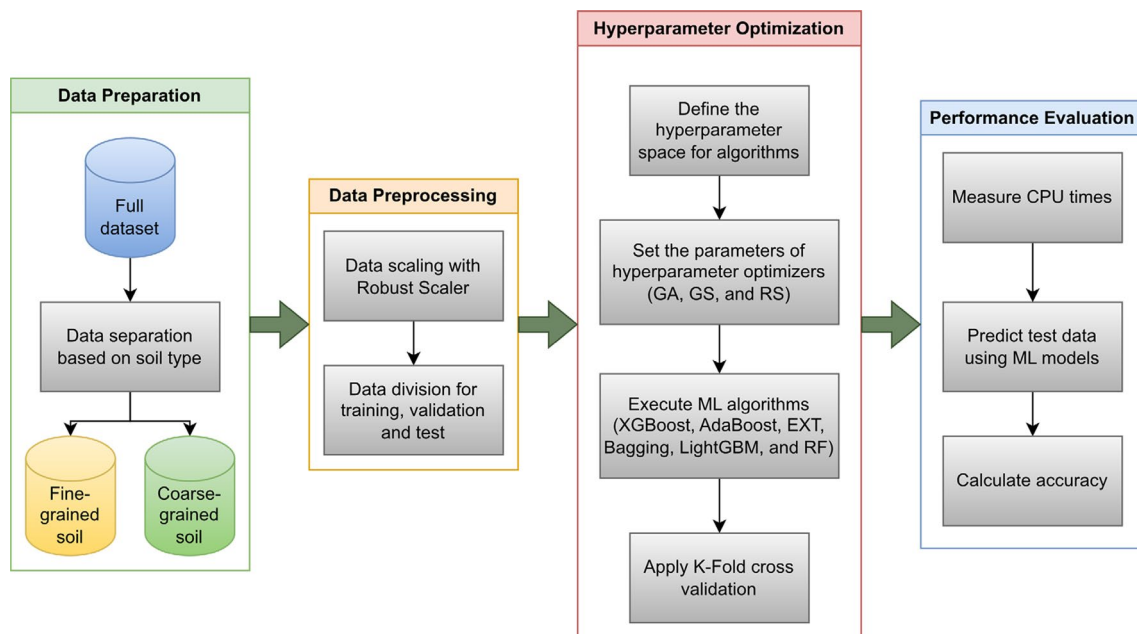


Fig. 1 General flow diagram of the study

format that machine learning algorithms can readily interpret. The algorithms can capture the inherent patterns and relationships associated with each category by converting the categorical variables into binary columns, enabling more comprehensive analysis and prediction.

To ensure the reliability and generalizability of the models, the dataset was divided into a training set, comprising 70% of the samples, and a test set, which constituted the remaining 30%. To avoid potential biases and objectively assess the models' performance, a tenfold cross-validation was conducted on the training set. The StratifiedKFold technique was used to select the validation sets, ensuring that the class distribution remained balanced throughout the cross-validation process. This rigorous evaluation methodology helps assess effectiveness and mitigate the risk of overfitting. In addition, the models' hyperparameters were fine-tuned to optimize their performance. To achieve high success in cross-validation results, searching in the hyperparameter space is a recommended and possible process. It is common for searching on a small subset of its parameters to have a large impact on the model's predictive and computational performance, while other parameters can be left at their default values (Pedregosa et al. 2011). In this study, a comparable strategy was employed, aligning with similar approaches found in the literature that have demonstrated successful outcomes (Abdu-Aljabar and Awad 2022; Amini et al. 2023; Cuocolo et al. 2020; Dhar 2022; Ge et al. 2023; Janizadeh et al. 2022; Sari and Maki 2023). These studies utilized parameters and parameters ranges similar to those outlined in Table 3. A range of values, as presented in

Table 3, were explored and adjusted to identify the optimal configuration for each algorithm. By carefully tuning these hyperparameters, the predictive capabilities were enhanced, improving accuracy and reliability in predicting soil liquefaction potential.

Hyperparameter optimization using GA

To investigate the classification of soil types, a grain size partitioning technique was used to analyze the soil samples. This process revealed two discernible categories of soil: fine-grained and coarse-grained soils. A comprehensive understanding of the different soil types was achieved by identifying and segregating the samples based on their grain size, facilitating a subsequent classification analysis. To assess the performance of the learning models, three distinct datasets were constructed, namely, as mentioned earlier, datasets A, B, and C, were constructed. Each dataset corresponded to a specific soil type, and the learning models were independently evaluated using these datasets. This approach allowed for meticulous examination of the effectiveness in accurately classifying the respective soil types. Therefore, a comprehensive and robust analysis was performed, allowing for a more nuanced understanding of the classification capabilities of the models.

In addition, this study used GAs as a hyperparameter optimization technique for ML algorithms. GA employs a population size parameter which determines the number of chromosomes within the population utilized in the algorithm. The population size plays a crucial role in the efficiency and

Table 3 Hyperparameters for different ML models

Algorithm	Hyperparameters	Range/values
XGBoost	eta	Continuous (0.01,0.999)
	gamma	Continuous (0.01,0.999)
	max_depth	Integer (2, 20)
	min_child_weight	Continuous (0.01,0.999)
	max_delta_step	Continuous (0.01,0.999)
	subsample	Continuous (0.01,1)
	n_estimators	Integer (5, 300)
AdaBoost	n_estimators	Integer (5, 300)
	learning_rate	Continuous (0.01,0.999)
EXT	max_depth	Integer (2, 20)
	n_estimators	Integer (5, 300)
	min_samples_split	Integer (2, X_train.shape[0] + 1)
	min_samples_leaf	Integer (2, X_train.shape[0] + 1)
	bootstrap	Categorical ([True, False])
Bagging	criterion	Categorical (["gini", "entropy"])
	max_samples	Integer (10, 50)
	n_estimators	Integer (10, 300)
	bootstrap	Categorical ([True, False])
LightGBM	bootstrap_features	Categorical([True, False])
	learning_rate	Continuous (0.01,0.999)
	num_leaves	Integer (2, 50)
RF	max_depth	Integer (2, 20)
	min_samples_split	Integer (2, X_train.shape[0] + 1)
	min_samples_leaf	Integer (2, X_train.shape[0] + 1)
	bootstrap	Categorical ([True, False])
	criterion	Categorical (["gini", "entropy"])

efficacy of the GA. A larger population size has the potential to enhance diversity within the population, increasing the chances of discovering the optimal global solution. However, this advantage comes at the expense of longer run times for each generation. Conversely, a smaller population size can lead to faster convergence but also increases the risk of premature convergence, where the algorithm may become trapped in a local optimum and fail to reach the global optimal solution. Choosing the most appropriate population size is thus crucial for the success of the GA and is often determined through experimentation and testing. This study evaluated the performance of ML algorithms using GA with different population sizes. The GA was implemented with different configurations. The population size varied between 10, 15, 20, and 30 individuals, while the number of generations was set to 100. A cross-over probability of 0.6 and a mutation probability of 0.2 were utilized. The tournament size for selection was set to 4, and the algorithm used for optimization was eaMuPlusLambda, that implements the $(\mu + \lambda)$ evolutionary algorithm, utilizing a provided population, toolbox of evolution operators, and various parameters to iteratively generate offspring, evaluate their fitness, and select the next generation for a specified number of generations, returning the final population and a logbook of evolution statistics. The GA was executed 20 times with a different random seed to mitigate the influence of randomness. The results of the experiment are presented in Fig. 2, which shows a comparison of the GA performance for varying population sizes. Based on the provided data, the highest average accuracy is achieved by Dataset B, while the lowest average accuracy is observed in Dataset C. Changes in population size do not appear to significantly affect the overall performance across datasets.

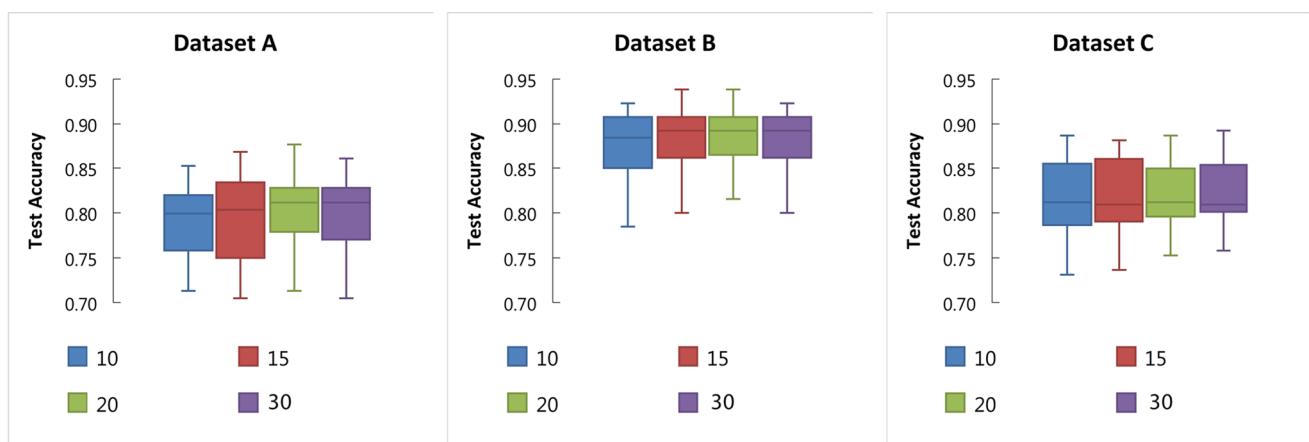


Fig. 2 Comparison of GA performance across different population sizes (Dataset A, B, and C, respectively)

After tuning the population size and other parameters of the GA, we applied ensemble learning algorithms, such as bagging and boosting, to further enhance the learning performance. However, to achieve optimal results with these algorithms, it is crucial to fine-tune their hyperparameters through experimental studies. To accomplish this, we leveraged the power of GA for hyperparameter optimization. The objective was to obtain the optimal set of hyperparameters for the ML algorithm, ensuring the best possible performance. This approach offers several benefits, including automating the hyperparameter tuning process and effectively handling non-linear and non-convex optimization problems with numerous hyperparameters. Using GA for hyperparameter optimization, our objective was to streamline and optimize the ensemble learning process, improving the overall performance of ML algorithms.

Tables 4 and 5 comprehensively evaluate the performance of ML accuracy and CPU times for different population sizes on three datasets. The values highlighted in bold indicate the highest performance metrics in each category: train accuracy, test accuracy, and CPU times, thereby aiding in the identification of the most effective ML algorithm per metric. Our experimental study showed that Dataset B achieved the highest performance, with a test accuracy of 0.9231 and a training accuracy of 0.9230. Dataset A exhibited a training accuracy of 0.8821 and a test accuracy of 0.8402, while Dataset C had a training accuracy of 0.8782 and a test accuracy of 0.8667.

Table 5 CPU times (seconds) for different classification algorithms on three datasets

Dataset	Algorithm	CPU times (s)			
		Population size			
		10	15	20	30
A	XGBoost	286	534	742	926
	AdaBoost	736	1555	1763	2502
	EXT	681	1023	1274	1984
	Bagging	1143	1590	2396	3346
	LightGBM	124	184	243	365
	RF	465	688	983	1373
B	XGBoost	174	288	382	434
	AdaBoost	729	1326	1251	2413
	EXT	636	1114	1265	1905
	Bagging	889	1173	1636	2253
	LightGBM	75	116	154	223
	RF	429	645	866	1285
C	XGBoost	444	537	1017	1352
	AdaBoost	1057	1543	1621	3323
	EXT	530	971	1199	1769
	Bagging	1152	1976	1952	3604
	LightGBM	104	174	210	329
	RF	492	764	996	1557

Tables 4, 5 and Fig. 3 compare the performance of multiple ML algorithms. LightGBM consistently emerges as the leading algorithm in accuracy across all three datasets. It

Table 4 Comparison of ML accuracy performance for different population sizes (Dataset A, B, C)

Dataset	Algorithm	Train accuracy				Test accuracy			
		Population size				Population size			
		10	15	20	30	10	15	20	30
A	XGBoost	0.8529	0.8556	0.8577	0.8586	0.8287	0.8402	0.8369	0.8369
	AdaBoost	0.8015	0.8057	0.8042	0.8094	0.7934	0.7959	0.7934	0.8033
	EXT	0.7523	0.7664	0.7581	0.7680	0.7336	0.7533	0.7369	0.7475
	Bagging	0.8129	0.8174	0.8181	0.8190	0.7967	0.8074	0.8123	0.8139
	LightGBM	0.8717	0.8744	0.8769	0.8821	0.8230	0.8279	0.8377	0.8254
	RF	0.8065	0.8012	0.8157	0.8010	0.7730	0.7451	0.7811	0.7541
B	XGBoost	0.9116	0.9177	0.9163	0.9184	0.8908	0.8938	0.8892	0.8877
	AdaBoost	0.9174	0.9176	0.9182	0.9202	0.8785	0.8862	0.8785	0.8738
	EXT	0.8393	0.8674	0.8821	0.8749	0.8123	0.8569	0.8692	0.8600
	Bagging	0.8996	0.9010	0.9010	0.8996	0.8723	0.8585	0.8446	0.8538
	LightGBM	0.9157	0.9177	0.9225	0.9230	0.9154	0.9200	0.9231	0.9200
	RF	0.8689	0.8639	0.8779	0.8813	0.8738	0.8646	0.9000	0.8923
C	XGBoost	0.8685	0.8706	0.8740	0.8731	0.8591	0.8640	0.8522	0.8581
	AdaBoost	0.8376	0.8388	0.8398	0.8424	0.8242	0.8231	0.8161	0.8253
	EXT	0.7811	0.7947	0.8188	0.8173	0.7731	0.7941	0.8048	0.8086
	Bagging	0.8355	0.8362	0.8372	0.8379	0.7887	0.8011	0.7909	0.7973
	LightGBM	0.8742	0.8752	0.8782	0.8747	0.8624	0.8667	0.8645	0.8602
	RF	0.8305	0.8337	0.8395	0.8438	0.7731	0.7753	0.7898	0.7866

Fig. 3 Comparison of ML algorithms test accuracy performances across different models

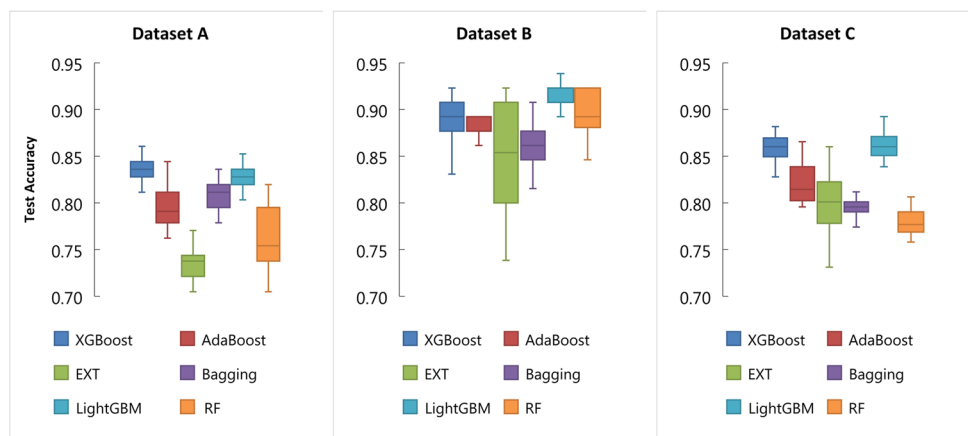
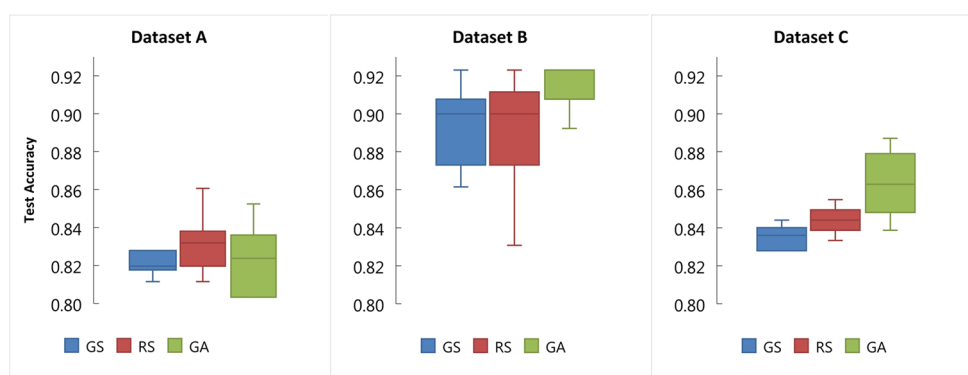


Fig. 4 Test accuracy performance comparisons of search algorithms (Dataset A, B, and C, respectively)



presents robust performance, particularly in predicting soil liquefaction potential, which is critically important in geotechnical engineering. AdaBoost and XGBoost also demonstrate strong accuracy results, making them viable alternatives. However, it is essential to consider computational efficiency, as some algorithms may have a longer CPU. As can be observed, training time generally increases with the population size of the dataset. Among the algorithms, Bagging consistently exhibits the highest training time, while LightGBM has the lowest training time across all datasets.

Comparison of genetic algorithm performances with other search algorithms

In the subsequent phase of our study, we compared the performance of the GA with other traditional search algorithms such as GS and RS. Details of the experimental setup and hyperparameters used for each algorithm are provided in the methodology section. The results showed that GA outperformed the other search algorithms mainly in terms of the mean and standard deviation of the results. It is important to note that the best results were obtained in dataset B. Figure 4 compares the performance of different search algorithms

Table 6 Performance comparison of search algorithms

Dataset	Algorithm	Avg. test	Avg. train	Avg. CPU (s)
A	GS	0.8230	0.8211	8.330
	RS	0.8320	0.8208	0.650
	GA	0.8230	0.8717	124.229
B	GS	0.8892	0.8898	8.076
	RS	0.8908	0.8805	0.627
	GA	0.9154	0.9157	74.854
C	GS	0.8349	0.8371	10.805
	RS	0.8446	0.8411	0.655
	GA	0.8624	0.8742	104.249

with respect to their ability to optimize hyperparameters for ML models. Regarding the standard deviation of the test precision, GS generally has the lowest standard deviation across the three datasets, indicating that its performance is more consistent compared to GA and RS. GA has a slightly higher standard deviation in Dataset A but performs better in Datasets B and C. RS has the highest standard deviation on Dataset B, suggesting that its performance is more variable on this particular dataset.

Figure 4 and Table 6 compare performance metrics for three search algorithms: GA, GS, and RS. The metrics evaluated include the average test accuracy, the average train accuracy, standard deviation of the test accuracy, the standard deviation of the train accuracy, and average CPU time. Based on the comparison, the GA algorithm achieves the highest average test and train accuracy, indicating its effectiveness in optimizing the model performance. Furthermore, the GA algorithm exhibits lower standard deviations, indicating more consistent performance across different trials. However, the GS and RS algorithms also show promising performance, although with slightly lower accuracy and varying standard deviations. Regarding computational efficiency, RS consistently achieves the lowest average CPU time across all datasets, suggesting its advantage in faster execution.

Discussion

This study evaluated the effectiveness of a GA approach for hyperparameter tuning of ML algorithms and compared it with traditional methods such as GS and RS. The study was divided into two parts. In the first part, the performance of six different ML algorithms was compared, with LightGBM showing the best performance. In the second part, the GA approach was evaluated, and it outperformed both GS and RS in terms of the average test and training accuracy. These findings suggest that GA is a promising method for hyperparameter tuning of ML algorithms. Overall, the study highlights the importance of selecting the appropriate population size for ML algorithms to achieve optimal performance and could be expanded further to evaluate the performance of GA on a larger set of ML algorithms and datasets.

While the GA may perform better in optimizing hyperparameters, it is important to consider that it can be computationally expensive and time-consuming, especially for large datasets. This aspect can be seen as a disadvantage when dealing with large datasets. Due to the nature of GA, which involves evaluating multiple individuals within a population over multiple generations, the computational cost increases as the dataset size grows. The time required to search the hyperparameter space and converge to the optimal solution can become a limiting factor, particularly when dealing with large datasets. In scenarios where computational efficiency is a priority, the time-consuming nature of GA may present a drawback. In such cases, alternative approaches that offer faster hyperparameter optimization methods, such as GS or RS, might be more suitable.

The choice of the hyperparameter optimization method should consider the trade-off between performance gains and computational time, considering the specific characteristics of the dataset and the available computational resources.

In addition to the findings related to hyperparameter optimization, the study also made significant contributions to soil mechanics and geotechnical engineering. Specifically, the study explored the use of ML algorithms to predict soil liquefaction potential based on particle size. Soil liquefaction is a phenomenon that occurs when saturated soil loses its strength and stiffness during an earthquake, leading to the inability of the soil to support structures such as buildings and bridges. Accurate prediction of soil liquefaction potential is critical to assess the safety and stability of structures in earthquake-prone regions.

As explained in the “Dataset Description” section of this study, the main dataset used in the study includes historical liquefaction cases on fine- and coarse-grained soils exposed to the 1999 Turkey and Taiwan earthquakes. When the data on the soil layers whose liquefaction status is given are examined, it is seen that all of them are data that affect the liquefaction sensitivity of coarse-grained soils and are used to determine their liquefaction risk (Table 1). It is known that liquefaction does not occur in all layers of the soil. Therefore, it is important to examine whether there are the necessary conditions for liquefaction in hazard assessments. The type of soil has a key place among these conditions. For years, it has been thought that soil liquefaction is a behavior peculiar only to water-saturated loose sands and that fine-grained soils cannot produce an excessive pore water pressure that causes liquefaction. However, it has been determined by research that low plasticity silty soils can liquefy easily, such as sands, and plasticity properties have been revealed to be important in affecting the liquefaction sensitivity of fine-grained soils (Ishihara 1984, 1985, 1996). The parameters and approaches used in the evaluation of liquefaction in fine- and coarse-grained soils are different from each other. When liquefaction in fine-grained soils, the decision is made mainly by examining the physical properties of the soil or based on laboratory test results. In the literature, there are many criteria based on the physical properties of the relevant soil to determine the liquefaction potential of fine-grained soils.

The physical properties of soils, particularly fine-grained soils, are important factors that can affect the liquefaction behavior of soils. The missing parameters in the dataset, such as soil grain size distribution, plasticity index, liquid limit and water content, can have a significant impact on the accuracy of the models. For example, Ghani et al. (2022) conducted a study in which physical properties such as w_n/LL and PI were used as an input parameter in addition to seismic properties to examine the liquefaction behavior of fine-grained soils under seismic conditions. They integrated artificial intelligence to enhance accuracy and reduce uncertainties associated with traditional deterministic approaches. A hybrid method combining optimization algorithms and

the adaptive neuro-based fuzzy inference system (ANFIS) was introduced to determine the safety factor against earthquake-induced liquefaction. The ANFIS firefly hybrid model demonstrated high predictive capabilities, with R^2 values of 0.976 and 0.982 in the training and testing phases, respectively. Moreover, it is important to consider the broader context of soil properties, particularly in fine-grained soils, which significantly influence the behavior of liquefaction. Ozsagir et al. (2022) emphasized the importance of incorporating comprehensive datasets, including parameters such as grain size distribution, plasticity index, and specific gravity, to enhance model accuracy. They achieved remarkable success in predicting soil liquefaction for fine-grained soils using machine learning models, with an accuracy of nearly 90%. Their findings highlighted the mean grain size (D_{50}) as a critical physical property that influences the performance of machine learning models.

Therefore, considering a range of physical properties in fine-grained soils is crucial to improving the predictive capabilities of models. On the other hand, in evaluating the liquefaction of sandy soil layers, the degree of firmness of the ground and the stress state are considered. The simplified method proposed by Seed and Idriss (1971) is the most widely used approach. The authors conducted the study by dividing the dataset into three groups according to the type of soil, with the foresight that the parameters used as input data in estimating the state of liquefaction could lead to erroneous interpretations, since they are only used for coarse soils. However, the cases of fine- and coarse-grained soils are combined in the dataset. Dataset A represented fine-grained soils, while Dataset B represented the coarse-grained soils, and Dataset C included all 620 soil cases. The ML algorithms applied in this study were evaluated on all three datasets. The best prediction performances were obtained with the LightGBM algorithm for the three datasets. Based on the test sets, these were calculated as 0.9231 for coarse-grained soils, 0.8402 for fine-grained soils, and 0.8667 for all soils.

As a result, the models used in the study could classify the soil liquefaction potential of fine-grained or coarse-grained soils with high accuracy. The best predictions were obtained for coarse-grained soils, while the lowest prediction performance was obtained for fine-grained soils. Taking into account the susceptibility to liquefaction of soils, the best prediction performance was obtained in coarse-grained soils, and the lowest prediction performance in fine-grained soils supported the authors' approach in this study to separate the main dataset according to the soil type. With the effect of including parameters used in the assessment of sandy soils in the

input layer of all models, the best estimation performance was achieved for the dataset representing coarse-grained soils. However, although none of the parameters given in this dataset are used in the calculations for the evaluation of fine-grained soils in the literature, they have provided a good prediction performance on fine-grained soils with ML methods. This evidence shows that the classification-based prediction models applied and proposed in this study can be useful in evaluating a complex phenomenon such as liquefaction.

Limitations of the study

Although this study has achieved significant learning outcomes, it is important to acknowledge certain limitations and challenges encountered during the research. Machine learning, in general, is limited by the availability and quality of data. Overfitting and underfitting are common challenges that can arise when training machine learning models. Overfitting happens when a model learns the specific details of the training data too well and starts to make predictions that are too closely tied to the training data, which can lead to poor performance on new, unseen data. On the other hand, this occurs when a model is too simple to capture the complexity of the data, resulting in poor predictions.

In the context of geotechnical engineering, the data used for training machine learning models often come from complex and multidimensional datasets. This can make it challenging to develop models that can accurately depict the underlying relationships in the data. Additionally, geotechnical problems often involve multiple interacting factors, which can make it difficult to identify the most important features for model training.

Genetic algorithm-optimized machine learning algorithms can be computationally costly, especially for large datasets. Especially the large number of populations and increasing the number of individuals that will undergo mutation and crossover can significantly increase the solution time and the amount of resource usage. This is true not only for GA, but also for other soft computing techniques with a wide choice of parameters. Although having many options may seem to increase uncertainty, it also creates an opportunity to improve the quality of the solution. A second hyperparameter optimization process can be applied to find the most suitable parameters. Or, initial parameter values can be created with the parameter values of good results obtained by examining similar studies in the literature. This can limit the practical applicability of these algorithms, particularly for real-time applications or for problems with a large number of input features.

In addition, in hyperparameter-based studies, choosing wide search ranges and choosing continuous variables instead of discrete and categorical variables has the chance to increase the solution quality, but may increase the calculation time and amount of resource usage.

Finally, machine learning models can suffer from generalization issues, meaning that they may not perform as well on new data that are different from the data used for training. This is a common challenge in machine learning, and it is important to carefully evaluate the performance of models on a variety of datasets to ensure that they are able to generalize well to new data.

In conclusion, while this study has demonstrated the potential of genetic algorithm-optimized machine learning for geotechnical engineering applications, it is important to be aware of the limitations and challenges associated with these methods. Future research should focus on addressing these limitations, such as developing more efficient algorithms, improving the generalization of models, and exploring new approaches to handling complex and multidimensional geotechnical data.

Conclusions

In this study, we investigate hyperparameter optimization techniques for ML algorithms, focusing on their application to prediction of soil liquefaction. By applying our approach, we were able to accurately predict soil liquefaction potential, highlighting the importance of utilizing artificial intelligence to improve the accuracy and reliability of these predictions. Our findings indicate that LightGBM emerged as the most effective among the ML algorithms evaluated. This observation underscores the importance of using sophisticated algorithms to improve understanding and prediction of soil liquefaction. Accurate prediction of soil liquefaction is crucial in preventing or mitigating the potential damage caused by this natural disaster. On the other hand, our study also draws attention to the effect of soil type on the performance results obtained in studies for the prediction of liquefaction.

In this study, it is emphasized that the input parameters to be selected based on the type of soil to estimate the liquefaction potential in coarse and fine-grained soils have an effect on increasing performance in prediction studies. Since there is no input parameter for the susceptibility to liquefaction of fine-grained soils in the dataset used in the study, the authors divided the dataset according to the type of soil and the best performance results were obtained with the dataset containing coarse-grained soils (Dataset B). This situation is explained by the fact that all input parameters in the dataset are parameters used to determine the liquefaction sensitivity of coarse-grained soils (especially sands). However, although there are no parameters to evaluate the liquefaction

potential of fine-grained soils in the dataset, high precision was achieved in the prediction of fine-grained soil liquefaction. This observation is another detail that reflects the success of the prediction performances of the methods used in this study.

Class redundancy in any dataset negatively affects the performance of prediction techniques. Poor performance in one class also reduces the success of the other class. By dividing the dataset into classes and applying the methods separately, the individual prediction performance is higher than when considering all classes as a whole. The main reason for this is that the parameters are determined specifically for each dataset class. In particular, machine learning algorithms based on learning have the opportunity to adjust the parameters with less complexity in the training processes by dividing the dataset into classes, allowing one to obtain high-accuracy results in a shorter time. As emphasized in this study, when machine learning algorithms are hyperparameterized using successful artificial intelligence techniques such as GA, the quality of the solution is positively affected.

Soil liquefaction is a critical geotechnical issue that poses significant risks, including physical damage and loss of life. Additionally, due to the heterogeneous characteristics of soils and the participation of many factors that affect the occurrence of liquefaction due to an earthquake, the determination of the liquefaction potential is considered one of the most complex problems in geotechnical engineering. This study contributes to the field by providing information on GA for hyperparameter optimization of ML algorithms, specifically in the prediction of soil liquefaction in geotechnical engineering. The successful application of the GA highlights its potential for broader applications in this domain.

Author contribution Alparslan Serhat Demir: methodology, writing—original draft, supervision. Talas Fikret Kurnaz: data collection, conceptualization, resources, writing—original draft, writing—review and editing, supervision. Abdullah Hulusi Kökçam: methodology, writing—original draft, formal analysis, visualization, validation. Caner Erden: methodology, writing—review and editing, software, data curation, visualization, validation. Uğur Dağdeviren: data collection, conceptualization, resources, writing—original draft, writing—review and editing, supervision. All authors read and approved the final manuscript.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability Data is available upon request.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Abdu-Aljabar RDA, Awad OA (2022) Improving lung cancer relapse prediction using the developed Optuna_XGB classification model. *Int J Intellig Eng Syst*
- Agrawal T (2021) Hyperparameter optimization in machine learning: make your machine learning and deep learning models more efficient. Apress Berkeley CA. <https://doi.org/10.1007/978-1-4842-6579-6>
- Ahmad M, Tang X-W, Qiu J-N, Ahmad F, Gu W-J (2021) Application of machine learning algorithms for the evaluation of seismic soil liquefaction potential. *Front Struct Civ Eng* 15:490–505. <https://doi.org/10.1007/s11709-020-0669-5>
- Ahmad M, Tang X-W, Qiu J-N, Ahmad F (2019) Evaluating seismic soil liquefaction potential using bayesian belief network and C45 decision tree approaches. *Appl Sci* 9:4226. <https://doi.org/10.3390/app9204226>
- Alizadeh Mansouri M, Dabiri R (2021) Predicting the liquefaction potential of soil layers in Tabriz city via artificial neural network analysis. *SN Appl Sci* 3:719. <https://doi.org/10.1007/s42452-021-04704-3>
- Almadani M, Kheimi M (2023) Stacking artificial intelligence models for predicting water quality parameters in rivers. *J Ecol Eng* 24:152–164
- Alobaidi MH, Meguid MA, Chebana F (2019) Predicting seismic-induced liquefaction through ensemble learning frameworks. *Sci Rep* 9:11786. <https://doi.org/10.1038/s41598-019-48044-0>
- Amini A, Dolatshahi M, Kerachian R (2023) Effects of automatic hyperparameter tuning on the performance of multi-variate deep learning-based rainfall nowcasting. *Water Resour Res* e2022WR032789
- Andrews DC, Martin GR (2000) Criteria for liquefaction of silty soils. In: *Proc., 12th World Conf. on Earthquake Engineering* pp. 1–8
- Baecher GB, Christian JT (2005) Reliability and statistics in geotechnical engineering. John Wiley & Sons
- Bol E, Önalp A, Arel E, Sert S, Özocak A (2010) Liquefaction of silts: the Adapazari criteria. *Bull Earthq Eng* 8:859–873
- Boulanger RW, Idriss IM (2014) CPT and SPT based liquefaction triggering procedures (No. UCD/CGM-14/01). Center for Geotechnical Modeling, University of California at Davis
- Boulanger RW, Idriss IM (2006) Liquefaction susceptibility criteria for silts and clays. *J Geotech Geoenviron Eng* 132:1413–1426. [https://doi.org/10.1061/\(ASCE\)1090-0241\(2006\)132:11\(1413\)](https://doi.org/10.1061/(ASCE)1090-0241(2006)132:11(1413))
- Bray JD, Sancio RB (2006) Assessment of the liquefaction susceptibility of fine-grained soils. *J Geotech Geoenviron Eng* 132:1165–1177. [https://doi.org/10.1061/\(ASCE\)1090-0241\(2006\)132:9\(1165\)](https://doi.org/10.1061/(ASCE)1090-0241(2006)132:9(1165))
- Cai M, Hocine O, Mohammed AS, Chen X, Amar MN, Hasanipanah M (2022) Integrating the LSSVM and RBFNN models with three optimization algorithms to predict the soil liquefaction potential. *Eng Comput* 38:3611–3623. <https://doi.org/10.1007/s00366-021-01392-w>
- Cetin KO, Seed RB, Der Kiureghian A, Tokimatsu K, Harder LF, Kayen RE, Moss RES (2004) Standard penetration test-based probabilistic and deterministic assessment of seismic soil liquefaction potential. *J Geotechn Geoenviron Eng* 130:1314–1340. [https://doi.org/10.1061/\(ASCE\)1090-0241\(2004\)130:12\(1314\)](https://doi.org/10.1061/(ASCE)1090-0241(2004)130:12(1314))
- 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, KDD '16*. ACM, New York, NY, USA, pp. 785–794. <https://doi.org/10.1145/2939672.2939785>
- Chern S.-G, Lee C.-Y, Wang C.-C (2008) CPT-based liquefaction assessment by using fuzzy-neural network. *J Marine Sci Technol* <https://doi.org/10.51400/2709-6998.2024>
- Cuocolo R, Ugga L, Solari D, Corvino S, D'Amico A, Russo D, Capabianca P, Cavallo LM, Elefante A (2020) Prediction of pituitary adenoma surgical consistency: radiomic data mining and machine learning on T2-weighted MRI. *Neuroradiology* 62:1649–1656. <https://doi.org/10.1007/s00234-020-02502-z>
- Das BM (1993) Principles of soil dynamics. PWS-KENT Publishing Company, Boston, USA
- Daviran M, Shamekhi M, Ghezelbash R, Maghsoudi A (2023) Landslide susceptibility prediction using artificial neural networks, SVMs and random forest: hyperparameters tuning by genetic optimization algorithm. *Int J Environ Sci Technol* 20:259–276. <https://doi.org/10.1007/s13762-022-04491-3>
- De M, Kundu A (2022) A hybrid optimization for threat detection in personal health crisis management using genetic algorithm. *Int J Inf Technol* 14:2603–2618. <https://doi.org/10.1007/s41870-022-00927-8>
- Demir S, Sahin EK (2023) An investigation of feature selection methods for soil liquefaction prediction based on tree-based ensemble algorithms using AdaBoost, gradient boosting, and XGBoost. *Neural Comput Appl* 35:3173–3190. <https://doi.org/10.1007/s00521-022-07856-4>
- Dhar J (2022) An adaptive intelligent diagnostic system to predict early stage of parkinson's disease using two-stage dimension reduction with genetically optimized lightgbm algorithm. *Neural Comput Appl* 34:4567–4593. <https://doi.org/10.1007/s00521-021-06612-4>
- Díaz JP, Sáez E, Monsalve M, Candia G, Aron F, González G (2022) Machine learning techniques for estimating seismic site amplification in the Santiago basin, Chile. *Eng Geol* 306:106764. <https://doi.org/10.1016/j.enggeo.2022.106764>
- Dietterich TG (2000) Ensemble methods in machine learning. In: *Proceedings of the First International Workshop on Multiple Classifier Systems, MCS '00*. Springer-Verlag, Berlin, Heidelberg, pp. 1–15
- Dou J, Yunus AP, Bui DT, Merghadi A, Sahana M, Zhu Z, Chen C-W, Han Z, Pham BT (2020) Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan. *Landslides* 17:641–658. <https://doi.org/10.1007/s10346-019-01286-5>
- Erzin Y, Tuskan Y (2019) The use of neural networks for predicting the factor of safety of soil against liquefaction. *Scientia Iranica* 26:2615–2623
- Evans MD, Seed HB (1987) Undrained cyclic triaxial testing of gravels: the effect of membrane compliance. University of California, College of Engineering
- Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55:119–139. <https://doi.org/10.1006/jcss.1997.1504>
- Gandomi AH, Fridline MM, Roke DA (2013) Decision tree approach for soil liquefaction assessment. *Scient World J*
- Ge D-M, Zhao L-C, Esmaeili-Falak M (2023) Estimation of rapid chloride permeability of SCC using hyperparameters optimized random forest models. *J Sustain Cement-Based Mater* 12:542–560. <https://doi.org/10.1080/21650373.2022.2093291>
- Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. *Mach Learn* 63:3–42. <https://doi.org/10.1007/s10994-006-6226-1>
- Ghani S, Kumari S (2022a) Liquefaction behavior of Indo-Gangetic region using novel metaheuristic optimization algorithms coupled with artificial neural network. *Nat Hazards* 111:2995–3029. <https://doi.org/10.1007/s11069-021-05165-y>
- Ghani S, Kumari S (2022b) Reliability analysis for liquefaction risk assessment for the City of Patna, India using hybrid computational modeling. *J Geol Soc India* 98:1395–1406




- Ghani S, Kumari S, Ahmad S (2022) Prediction of the seismic effect on liquefaction behavior of fine-grained soils using artificial intelligence-based hybridized modeling. *Arab J Sci Eng* 47:5411–5441. <https://doi.org/10.1007/s13369-022-06697-6>
- Ghani S, Kumari S (2021) Effect of plasticity index on liquefaction behavior of silty clay, In: Sitharam TG, Dinesh SV, Jakka R (Eds.), *soil dynamics, lecture notes in civil engineering*. Springer Singapore, Singapore, pp. 289–298. https://doi.org/10.1007/978-981-33-4001-5_26
- Golmoghani Ebrahimi S, Noorzad A, Kupaei HJ (2023) Reliability analysis of soil liquefaction using improved hypercube sampling (IHS) method. *Int J Civ Eng*. <https://doi.org/10.1007/s40999-023-00863-z>
- Gong Y, Liu G, Xue Y, Li R, Meng L (2023) A survey on dataset quality in machine learning. *Inf Softw Technol* 162:107268. <https://doi.org/10.1016/j.infsof.2023.107268>
- Hanna AM, Ural D, Saygili G (2007) Neural network model for liquefaction potential in soil deposits using Turkey and Taiwan earthquake data. *Soil Dyn Earthq Eng* 27:521–540. <https://doi.org/10.1016/j.soildyn.2006.11.001>
- Hoang N-D, Bui DT (2018) Predicting earthquake-induced soil liquefaction based on a hybridization of kernel Fisher discriminant analysis and a least squares support vector machine: a multi-dataset study. *Bull Eng Geol Environ* 77:191–204. <https://doi.org/10.1007/s10064-016-0924-0>
- Holland JH (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press
- Idriss IM, Boulanger RW (2008) *Soil liquefaction during earthquakes (Monograph No. MNO-12)*. Earthquake Engineering Research Institute, Oakland, CA
- Idriss IM, Boulanger RW (2010) SPT-based liquefaction triggering procedures (No. UCD/CGM-10/02). Center for Geotechnical Modeling, University of California at Davis.
- Ishihara K (1984) Post-earthquake failure of a tailings dam due to liquefaction of the pond deposits. *Proceeding of International Conference on Case Histories in Geotechnical Engineering*. University of Missouri, St. Louis, pp 1129–1143
- Ishihara K (1996) *Soil Behaviour in Earthquake Geotechnics*, 1st edn. Clarendon Press, Oxford
- Ishihara K (1985) Stability of natural deposits during earthquakes, in: *Proceedings of the 11th International Conference on Soil Mechanics and Foundation Engineering*, A. Balkema. Rotterdam, The Netherlands, pp. 321–376
- Jalal FE, Xu Y, Iqbal M, Jamhiri B, Javed MF (2021) Predicting the compaction characteristics of expansive soils using two genetic programming-based algorithms. *Transport Geotechnol* 30:100608. <https://doi.org/10.1016/j.trgeo.2021.100608>
- Janizadeh S, Vafakhah M, Kapelan Z, Mobarghaee Dinan N (2022) Hybrid XGboost model with various Bayesian hyperparameter optimization algorithms for flood hazard susceptibility modeling. *Geocarto Int* 37:8273–8292. <https://doi.org/10.1080/10106049.2021.1996641>
- Jha SK, Suzuki K (2009) Reliability analysis of soil liquefaction based on standard penetration test. *Comput Geotech* 36:589–596
- Johari A, Javadi AA, Makiabadi MH, Khodaparast AR (2012) Reliability assessment of liquefaction potential using the jointly distributed random variables method. *Soil Dyn Earthq Eng* 38:81–87
- Juang C, Fang S, Tang W, Khor E, Kung GT-C, Zhang J (2009) Evaluating model uncertainty of an SPT-based simplified method for reliability analysis for probability of liquefaction. *Soils Found* 49:135–152
- Juang CH, Gong W, Wasowski J (2022) Trending topics of significance in engineering geology. *Eng Geol* 296:106460. <https://doi.org/10.1016/j.enggeo.2021.106460>
- Kayadelen C (2011) Soil liquefaction modeling by genetic expression programming and neuro-fuzzy. *Expert Syst Appl* 38:4080–4087
- Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y (2017) Lightgbm: a highly efficient gradient boosting decision tree. *Adv Neural Inf Process Syst* 30:3146–3154
- Kramer SL (1996) *Geotechnical earthquake engineering*. Prentice-Hall, Inc
- Kumar D, Samui P, Kim D, Singh A (2021) A novel methodology to classify soil liquefaction using deep learning. *Geotech Geol Eng* 39:1049–1058. <https://doi.org/10.1007/s10706-020-01544-7>
- Kumar DR, Samui P, Burman A (2022) Prediction of probability of liquefaction using soft computing techniques. *J Inst Eng India Ser A* 103:1195–1208. <https://doi.org/10.1007/s40030-022-00683-9>
- Kummer AF, de Araújo OCB, Buriol LS, Resende MGC (2023) A biased random-key genetic algorithm for the home health care problem. *Int Trans Oper Res*. <https://doi.org/10.1111/itor.13221>
- Kurnaz TF, Kaya Y (2019) A novel ensemble model based on GMDH-type neural network for the prediction of CPT-based soil liquefaction. *Environ Earth Sci*. <https://doi.org/10.1007/s12665-019-8344-7>
- Kurnaz TF, Erden C, Kökçam AH, Dağdeviren U, Demir AS (2023) A hyper parameterized artificial neural network approach for prediction of the factor of safety against liquefaction. *Eng Geol* 319:107109. <https://doi.org/10.1016/j.enggeo.2023.107109>
- Kwak BM, Lee TW (1987) Sensitivity analysis for reliability-based optimization using an AFOSM method. *Comput Struct* 27:399–406
- Law KT, Wang J (1994) *Siting in earthquake zones*. A.A. Balkema/ Rotterdam/Brookfield
- Li Y, Rahardjo H, Satyanaga A, Rangarajan S, Lee DT-T (2022) Soil database development with the application of machine learning methods in soil properties prediction. *Eng Geol* 306:106769
- Lim Y (2022) *State-of-the-Art Machine Learning Hyperparameter Optimization with Optuna [WWW Document]*. Medium. URL <https://towardsdatascience.com/state-of-the-art-machine-learning-hyperparameter-optimization-with-optuna-a315d8564de1> (accessed 5.25.23).
- Mienye ID, Sun Y (2022) A survey of ensemble learning: concepts, algorithms, applications, and prospects. *IEEE Access* 10:99129–99149. <https://doi.org/10.1109/ACCESS.2022.3207287>
- Mughieda O, Bani-Hani K, Safieh B (2009) Liquefaction assessment by artificial neural networks based on CPT. *Int J Geotech Eng* 3:289–302. <https://doi.org/10.3328/IJGE.2009.03.02.289-302>
- Ozsagir M, Erden C, Bol E, Sert S, Özocak A (2022) Machine learning approaches for prediction of fine-grained soils liquefaction. *Comput Geotech* 152:105014. <https://doi.org/10.1016/j.compgeo.2022.105014>
- Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
- Pei X, Mei F, Gu J, Chen Z (2022) Research on real-time state identification model of electricity-heat system considering unbalanced data, In: *2022 IEEE 5th International Conference on Electronics Technology (ICET)*. Presented at the 2022 IEEE 5th International Conference on Electronics Technology (ICET), pp. 501–505. <https://doi.org/10.1109/ICET55676.2022.9824069>
- Pham TA (2021) Application of feedforward neural network and SPT results in the estimation of seismic soil liquefaction triggering. *Comput Intellig Neurosci*. <https://doi.org/10.1155/2021/1058825>

- Rahbarzare A, Azadi M (2019) Improving prediction of soil liquefaction using hybrid optimization algorithms and a fuzzy support vector machine. *Bull Eng Geol Env* 78:4977–4987. <https://doi.org/10.1007/s10064-018-01445-3>
- Rahman MS, Wang J (2002) Fuzzy neural network models for liquefaction prediction. *Soil Dyn Earthq Eng* 22:685–694. [https://doi.org/10.1016/S0267-7261\(02\)00059-3](https://doi.org/10.1016/S0267-7261(02)00059-3)
- Rajasekar V, Krishnamoorthi S, Saračević M, Pepic D, Zajmovic M, Zogic H (2022) Ensemble machine learning methods to predict the balancing of ayurvedic constituents in the human body: ensemble machine learning methods to predict. *Comput Sci*. <https://doi.org/10.7494/csci.2022.23.1.4315>
- Ramakrishnan D, Singh TN, Purwar N, Barde KS, Gulati A, Gupta S (2008) Artificial neural network and liquefaction susceptibility assessment: a case study using the 2001 Bhuj earthquake data, Gujarat, India. *Comput Geosci* 12:491–501. <https://doi.org/10.1007/s10596-008-9088-8>
- Rehman ZU, Khalid U, Ijaz N, Mujtaba H, Haider A, Farooq K, Ijaz Z (2022) Machine learning-based intelligent modeling of hydraulic conductivity of sandy soils considering a wide range of grain sizes. *Eng Geol*. <https://doi.org/10.1016/j.enggeo.2022.106899>
- Samui P, Karthikeyan J (2013) Determination of liquefaction susceptibility of soil: a least square support vector machine approach. *Int J Numer Anal Meth Geomech* 37:1154–1161. <https://doi.org/10.1002/nag.2081>
- Samui P, Sitharam TG (2011) Machine learning modelling for predicting soil liquefaction susceptibility. *Nat Hazard* 11:1–9. <https://doi.org/10.5194/nhess-11-1-2011>
- Sari SA, Maki WFA (2023) Masked Face Images Based Gender Classification using Hybrid Bat Algorithm Optimized Bagging. In: 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC). Presented at the 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), pp. 091–096. <https://doi.org/10.1109/ICAIIIC57133.2023.10067008>
- Seed RB, Cetin KO, Moss RE, Kammerer AM, Wu J, Pestana JM, Riemer MF, Sancio RB, Bray JD, Kayen RE (2003) Recent advances in soil liquefaction engineering: a unified and consistent framework. In: Proceedings of the 26th Annual ASCE Los Angeles Geotechnical Spring Seminar: Long Beach, CA
- Seed HB, Idriss IM (1971) Simplified procedure for evaluating soil liquefaction potential. *J Soil Mechan Found Div* 97:1249–1273. <https://doi.org/10.1061/JSFEAQ.0001662>
- Sharma R, Kodamana H, Ramteke M (2022) Multi-objective dynamic optimization of hybrid renewable energy systems. *Chem Eng Process Intens* 170:108663. <https://doi.org/10.1016/j.cep.2021.108663>
- Umar SK, Kumari S, Samui P, Kumar D (2022) A liquefaction study using ENN, CA, and biogeography optimized-based ANFIS technique. *Int J Appl Metaheuristic Comput (IJAMC)* 13:1–23. <https://doi.org/10.4018/IJAMC.290535>
- Walia MS (2021) Best boosting algorithm in machine learning In: 2021. *Analytics Vidhya*. URL <https://www.analyticsvidhya.com/blog/2021/04/best-boosting-algorithm-in-machine-learning-in-2021/> (Accessed 5.25.23)
- Wang Y, Tang H, Huang J, Wen T, Ma J, Zhang J (2022) A comparative study of different machine learning methods for reservoir landslide displacement prediction. *Eng Geol* 298:106544. <https://doi.org/10.1016/j.enggeo.2022.106544>
- Xue X, Liu E (2017) Seismic liquefaction potential assessed by neural networks. *Environ Earth Sci* 76:192. <https://doi.org/10.1007/s12665-017-6523-y>
- Xue X, Xiao M (2016) Application of genetic algorithm-based support vector machines for prediction of soil liquefaction. *Environ Earth Sci* 75:874. <https://doi.org/10.1007/s12665-016-5673-7>
- Xue X, Yang X (2013) Application of the adaptive neuro-fuzzy inference system for prediction of soil liquefaction. *Nat Hazards* 67:901–917. <https://doi.org/10.1007/s11069-013-0615-0>
- Xue X, Yang X (2016) Seismic liquefaction potential assessed by support vector machines approaches. *Bull Eng Geol Environ* 75:153–162. <https://doi.org/10.1007/s10064-015-0741-x>
- Yegian MK, Ghahraman VG, Harutiunyan RN (1994) Liquefaction and embankment failure case histories, 1988 Armenia earthquake. *J Geotechn Eng* 120:581–596
- Yılmaz F, Öztürkoğlu Ş, Kamiloğlu HA (2022) A hybrid approach for computational determination of liquefaction potential of Erzurum City Center based on SPT data using response surface methodology. *Arab J Geosci* 15:95. <https://doi.org/10.1007/s12517-021-09312-4>
- Youd TL, Idriss IM, Andrus RD, Arango I, Castro G, Christian JT, Dobry R, Finn WDL, Harder LF, Hynes ME, Ishihara K, Koester JP, Liao SSC, Marcuson WF, Martin GR, Mitchell JK, Moriwaki Y, Power MS, Robertson PK, Seed RB, Stokoe KH (2001) Liquefaction resistance of soils: summary report from the 1996 NCEER and 1998 NCEER/NSF workshops on evaluation of liquefaction resistance of soils. *J Geotechn Geoenviron Eng* 127:817–833. [https://doi.org/10.1061/\(ASCE\)1090-0241\(2001\)127:10\(817\)](https://doi.org/10.1061/(ASCE)1090-0241(2001)127:10(817))
- Youd TL, Idriss IM (2001) Liquefaction resistance of soils: summary report from the 1996 NCEER and 1998 NCEER/NSF workshops on evaluation of liquefaction resistance of soils. *J Geotechn Geoenviron Eng* 127:297–313
- Zhang J, Wang Y (2021) An ensemble method to improve prediction of earthquake-induced soil liquefaction: a multi-dataset study. *Neural Comput Appl* 33:1533–1546. <https://doi.org/10.1007/s00521-020-05084-2>
- Zhang Y, Qiu J, Zhang Y, Xie Y (2021) The adoption of a support vector machine optimized by GWO to the prediction of soil liquefaction. *Environ Earth Sci* 80:360. <https://doi.org/10.1007/s12665-021-09648-w>
- Zhao Z, Duan W, Cai G (2021) A novel PSO-KELM based soil liquefaction potential evaluation system using CPT and Vs measurements. *Soil Dyn Earthq Eng* 150:106930. <https://doi.org/10.1016/j.soildyn.2021.106930>
- Zhou J, Huang S, Wang M, Qiu Y (2022a) Performance evaluation of hybrid GA–SVM and GWO–SVM models to predict earthquake-induced liquefaction potential of soil: a multi-dataset investigation. *Eng Comput* 38:4197–4215. <https://doi.org/10.1007/s00366-021-01418-3>
- Zhou J, Huang S, Zhou T, Armaghani DJ, Qiu Y (2022b) Employing a genetic algorithm and grey wolf optimizer for optimizing RF models to evaluate soil liquefaction potential. *Artif Intell Rev* 55:5673–5705. <https://doi.org/10.1007/s10462-022-10140-5>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Alparslan Serhat Demir¹  · Talas Fikret Kurnaz²  · Abdullah Hulusi Kökçam¹  · Caner Erden^{3,5}  ·
Uğur Dağdeviren⁴ 

✉ Caner Erden
cerden@subu.edu.tr

Alparslan Serhat Demir
alparslanserhat@sakarya.edu.tr

Talas Fikret Kurnaz
fkurnaz@mersin.edu.tr

Abdullah Hulusi Kökçam
akokcam@sakarya.edu.tr

Uğur Dağdeviren
ugur.dagdeviren@dpu.edu.tr

² Technical Sciences Vocational School, Transportation Services, Mersin University, Mersin, Türkiye

³ AI Research and Application Center, Sakarya University of Applied Sciences, Sakarya, Türkiye

⁴ Department of Civil Engineering, Faculty of Engineering, Kutahya Dumlupinar University, Kutahya, Türkiye

⁵ Department of Computer Engineering, Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, Türkiye

¹ Department of Industrial Engineering, Faculty of Engineering, Sakarya University, Sakarya, Türkiye