Geomechanics and Engineering, *Vol. 37, No. 5 (2024) 475-498*
https://doi.org/10.12989/gae.2024.37.5.475

Improved prediction of soil liquefaction susceptibility using ensemble learning algorithms

Satyam Tiwari[1a], Sarat K. Das*[1], Madhumita Mohanty[1b] and Prakhar[2c]

[1]*Department of Civil Engineering, Indian Institute of Technology (ISM) Dhanbad, Jharkhand 826004, India*
[2]*Department of Civil Engineering, National Institute of Technology Jaipur, Rajasthan 302017, India*

(*Received April 6, 2023, Revised May 8, 2024, Accepted May 9, 2024*)

Abstract. The prediction of the susceptibility of soil to liquefaction using a limited set of parameters, particularly when dealing with highly unbalanced databases is a challenging problem. The current study focuses on different ensemble learning classification algorithms using highly unbalanced databases of results from in-situ tests; standard penetration test (SPT), shear wave velocity (V_s) test, and cone penetration test (CPT). The input parameters for these datasets consist of earthquake intensity parameters, strong ground motion parameters, and in-situ soil testing parameters. liquefaction index serving as the binary output parameter. After a rigorous comparison with existing literature, extreme gradient boosting (XGBoost), bagging, and random forest (RF) emerge as the most efficient models for liquefaction instance classification across different datasets. Notably, for SPT and V_s-based models, XGBoost exhibits superior performance, followed by Light gradient boosting machine (LightGBM) and Bagging, while for CPT-based models, Bagging ranks highest, followed by Gradient boosting and random forest, with CPT-based models demonstrating lower $G_{mean(error)}$, rendering them preferable for soil liquefaction susceptibility prediction. Key parameters influencing model performance include internal friction angle of soil (ϕ) and percentage of fines less than 75 μ (F_{75}) for SPT and V_s data and normalized average cone tip resistance (q_c) and peak horizontal ground acceleration (a_{max}) for CPT data. It was also observed that the addition of V_s measurement to SPT data increased the efficiency of the prediction in comparison to only SPT data. Furthermore, to enhance usability, a graphical user interface (GUI) for seamless classification operations based on provided input parameters was proposed.

Keywords: classification; ensemble learning; in-situ tests; liquefaction; machine learning

1. Introduction

Liquefaction is a phenomenon where the effective stress of soil significantly decreases as a response to a sudden increase in pore water pressure, causing the granular material of soil to transform into a liquid state from its solid state. Since the identification of dynamic liquefaction, during the Nigata earthquake in Japan in 1964 (Kramer 1996), various studies have been made to identify the liquefaction susceptibility of soil. Keeping in mind the difficulties in the collection of the so-called undisturbed sample, the in-situ tests; cone penetration test (CPT), standard penetration test (SPT), and shear wave velocity (V_s) test (SWVT) are generally used (Kulhawy and Mayne 1990). SPT is advantageous in terms of sample collection from the borehole (Eslami *et al.* 2020), the advantage of CPT is its ability to collect data continuously at a faster rate (Das 2014). It is also evident that V_s can be correlated more directly with the relative density than SPT and CPT tests.

Relative density strongly affects the cyclic behavior of saturated soil (Idriss and Boulanger 2006). Duman *et al.* (2014) used predictive equation of Seed and Idriss (1971) and Iwasaki *et al.* (1984) to determine the liquefaction potential in Erzincan city center. Reported works in the literature also focus on the predictive modeling of V_s with piezo-cone penetration test data (Abbaszadeh Shahri and Naderi 2016).

It has been observed that due to spatial variation of soil, mechanistic models of liquefaction impose high complexity. Therefore, simplified statistical analysis methods (Andrus and Stokoe II 2000, Juang *et al.* 2000, 2005, Stokoe *et al.* 1988, Tokimatsu and Uchida 1990) are mostly preferred. However, methods based on artificial intelligence (AI) like artificial neural networks (ANN) (Erzin and Ecemis 2015, Goh 1994, Hanna *et al.* 2007, Juang *et al.* 2000, Samui and Sitharam 2011), support vector machines (SVM) (Goh and Goh 2007, Oommen *et al.* 2010, Pal 2006, Samui and Sitharam 2011, Zhang *et al.* 2021a), relevance vector machines (RVM) (Samui 2007) and genetic programming (GP) (Gandomi and Alavi 2012, Muduli and Das 2013, 2015), kernel Fisher discriminant analysis with a least-squares support vector machine (KFDA-LSSVM) (Hoang and Bui 2018), multi-adoptive regression spline (MARS) (Zhang and Goh 2016), machine learning (ML)-based methods like random forest (RF) and extreme gradient boosting (XGBoost) (Wang *et al*. 2021, Zhang *et al.* 2021b) are found to more efficient in comparison to statistical methods.

Ozsagir *et al.* (2022) worked towards classifying liquefaction class instances using SPT data. For this

*Corresponding author, Professor
E-mail: saratdas@iitism.ac.in
[a]M.Tech.
[b]Ph.D.
[c]B.Tech.

ISSN: 2005-307X (Print), 2092-6219 (Online)

purpose, a series of ML algorithms have been utilized, and the decision tree (DT) algorithm was found to be most effective for this problem. Zhang *et al.* (2021a) employed the grey wolf optimization (GWO) to improve the accuracy of the SVM model. The study conducted by Zhang *et al.* (2021a) is important in the sense that it trains the GWO-SVM model with and without V_s parameter and concludes that combining the shear wave data along with SPT data significantly improves the prediction accuracy. The CPT test results are found to be very crucial in soil liquefaction analysis. Along with statistical and empirical relationships, CPT data is also employed in several ML models. Demir and Sahin (2022) have examined how well tree-based machine learning techniques, such as RF, rotation forest (RotFor), and canonical correlation forest (CCF), perform in predicting a soil's potential for liquefaction based on CPT data. Through rigorous experimentation and comparative analysis, the research seeks to elucidate the optimal ensemble algorithm tailored to the characteristics of each dataset, thereby advancing the discourse on predictive modeling in geotechnical engineering and hazard assessment

Several coupled AI techniques have also been suggested by the researchers. Atangana Njock *et al.* (2020) coupled the dimensionality reduction technique with an evolutionary neural network (ENN) and proposed that this algorithm performs very well in the prediction of soil liquefaction cases. Atangana Njock *et al.* (2020) also highlight the improvement in model performance by coupling an AI model with a dimensionality reduction technique which can reduce the model complexity Hybrid approaches combining numerical and probabilistic models have also gained importance in literature. Gupta *et al.* (2023) developed ML frameworks for the prediction of liquefaction-induced settlement, this framework utilizes the CPT data and considers the uncertainties of earthquakes as well as the soil special variabilities using a probabilistic model. Hyperparameterized models of ANN have also found their application in the field of liquefaction potential estimation. These architectures use random search, Bayesian optimization, and grid search algorithms to obtain superior results. Kurnaz *et al.* (2023) conducted a similar study and found out that hyperparameter optimization significantly improved the learning ability of ANN, and observed that it outperforms models like DT, and RF. The main drawback of AI and ML algorithms is their connection with physical parameters. Several researchers have used AI algorithms, such as ANN, that can approximate complex functions and their theoretical properties. Anitescu *et al.* (2019) used ANNs with an adaptive collocation strategy for solving second-order boundary value problems. Samaniego *et al.* (2020) explored deep neural networks (DNNs) to approximate the solutions of partial differential equations by concentrating more on the mechanical problems, hence increasing the physical information of the model. Others have used the deep collocation method (DCM) along with transfer learning techniques for effectively solving problems incorporating the practical governing equations (Guo *et al.* 2022a). These ensemble and deep learning methodologies present a robust framework for the development of interpretative AI models. However, there is always a conflict of thought on the acceptance of traditional statistical methods and heuristic AI and ML-based methods.

Among other challenges in the modeling of liquefaction, one is to deal with the unbalanced dataset. To indicate liquefaction susceptibility binary classification is used where liquefied and non-liquefied instances are represented by '1' and '0' respectively. An unbalanced dataset has the majority of its individuals of one type, this problem is encountered in several research areas. Such a type of dataset influences the classifier to tend more toward majority class instances. Models developed from such datasets are strong in the classification of majority-class instances, but weak for minority-class instances. Therefore, it is very important to choose the appropriate algorithm in the model development phase which has lesser biasing. Apart from that, while comparing different models it is of utmost importance to choose valid performance indicators that can efficiently show the performance of both major as well as minor class instances (Powers 2011). Some of the commonly used performance metrics for the imbalanced dataset are F_1 score, Precession, and recall. The F_1 score can be separately calculated for each class and then can be averaged which will account for the overall performance of the model. The area under the receiver operating characteristic curve (AUROC) is another measure to evaluate the performance of an unbalanced dataset. AUROC keeps into account the trade-off between the true and false positive rates (Davis and Goadrich 2006). Due to the problem of misleading and biased behavior of a classifier on the imbalanced dataset (Yuan and Liu 2012) and for better efficiency of computational algorithm another performance indicator is used in some literature, that is called geometric mean of the individual accuracies of each class instance (G_{mean}) (Kubat *et al.* 1997).

The present study focuses on evaluating the performances and identifying a better model of different ensemble classification algorithms in handling the unbalanced data of soil liquefaction. As AI and ML models alternate with statistical methods, various statistical aspects in data analysis are discussed for comparison and connections. The performance indicators are chosen based on their applicability to unbalanced dataset-based models. Different AI techniques have limitations in overfitting the data, limited learning of nonlinear relationships, and challenges with minority class instances classification (Das *et al.* 2020). Hence, ensemble learning algorithms are employed in this study due to their efficient performance in handling unbalanced datasets (Yuan *et al.* 2024). They also show good robustness in handling noise and variability of the dataset (Guo *et al.* 2022). Algorithms such as DT and RF offer intuitive interpretations for feature importance (Le *et al.* 2024), Moreover, multiple learning models, reduce overfitting, capture complex relationships, and handle unbalanced data efficiently (Zhang and Ma 2012, Yuan *et al*. 2024). The accessibility of streamlined and meticulously crafted ensemble methods within prevalent machine learning libraries such as scikit-learn, XGBoost, and Light Gradient Boosting Machine (LightGBM) can significantly impact decision-making, streamlining implementation and

experimentation endeavors for researchers and practitioners alike.

Most of the published literature focuses mainly on either SPT, CPT, or V_s-based classification. However, the uniqueness of this study is the collection and analysis of data derived from all three testing methodologies for the same region. Through rigorous experimentation and comparative analysis, the research seeks to elucidate the optimal ensemble algorithm tailored to the characteristics of each dataset and to identify the most appropriate in-situ test(s) to accurately predict the liquefaction susceptibility. The present study may help in advancing the discourse on predictive modeling in geotechnical engineering and hazard assessment. In the present study, LightGBM a novel ensemble machine learning algorithm, which is very efficient in handling large datasets has been used. To the best of the knowledge of the authors, this algorithm is yet to be applied in geotechnical engineering. The results from the study suggest that combining V_s measurements with the SPT dataset significantly improves the classification performance in comparison to the SPT and V_s datasets alone, this is a novel finding of this study.

The study also takes into account the relative importance of different input parameters in the liquefaction index identification. In total Fifteen algorithms namely: Simple Decision Tree Classifier (SDT), RF Classifier, Gradient Boosting Classifier (GB), Bagging Classifier, Adaptive Boosting Classifier (AdaBoost), Voting Classifier, Stacking Classifier, Naive Bayes Classifier (NB), K-Nearest Neighbor Classifier (K-NN), Logistic Regression Classifier (LR), SV Classifier, XGBoost Classifier, Extra Tree Classifier (ET), LightGBM, and Hist Gradient Boosting Classifier (HGB) are utilized for the development of classification models. Numerous combinations of the algorithm running parameters, as well as input parameters, have been tested to obtain the best results which have been presented in this study. The classification accuracy of the developed models has been compared with previously published models from the literature and algorithms are ranked based on their performance. To reduce the difficulties in the utilization of the developed AI-based models by professional engineers, user-friendly software has been presented. This software utilizes the best models developed using RF algorithms for SPT and CPT databases with and without V_s measurement parameters.

A stratified K-fold validation technique is employed in the current research as it is quite useful in unbalanced datasets. It divides the dataset in such a way that each fold maintains the target variable category proportion similar to the entire dataset. The current methodology also employs ensemble algorithms that train multiple classifiers on the balanced subsets of data and then use voting or averaging to combine the results. This approach provides a more diverse solution that is less affected by class imbalance. Along with that, the boosting algorithms provide more weightage to misclassified class instances hence focusing on learning more from hard-to-classify minority cases. The unbalanced dataset favors the classification of majority class instances (Das *et al.* 2020). Therefore, a suitable performance metric is crucial for both driving and assessing the classifier model's performance. Merely using the accuracy of liquefied (A_L) and non-liquefied cases (A_{NL}) as well as the AUROC (Oommen *et al.* 2010) may not be a sufficient measure to evaluate the true classification performance. Hence a single performance metric $G_{mean(error)}$ is used as given in Eq. (8). Because it takes into account the accuracy of both classes, it may be applied to unbalanced datasets efficiently. Apart from that, other performance metrics like the F_1 score and Matthews Correlation Coefficient (MCC) are used (Naser and Alavi 2021), which are very efficient for unbalanced dataset-related problems.

The overfitting of the models is monitored by evaluating the difference in model performance between the training and testing phases. Ensemble methodologies amalgamate diverse base learners, each trained through distinct methodologies or on varied data subsets, thereby capturing multifaceted aspects of the underlying data distribution, consequently mitigating overfitting risks. The gradient boosting algorithms are susceptible to the problems of gradient boosting. This has been taken care of by selecting a suitable base learner for the model by taking references from the literature. Ensemble learning algorithms combine multiple models, therefore, they are less prone to fall in the local minima in comparison to single models. The ensemble can explore a larger range of the parameter space due to the diversity among base learners, which lowers the chance of convergence to unsatisfactory solutions.

The results from the study indicate that ensemble learning models such as XGBoost, LightGBM, Bagging, and RF perform very well ($G_{mean(error)}$ < 4%) in the classification of liquefaction cases based on SPT and V_s database. On the other hand, Bagging, GB, RF, and XGBoost perform superior ($G_{mean(error)}$ < 2%) for classification tasks based on CPT data. The developed models also propose that for the classification of liquefaction cases soil friction angle (ϕ), and Percentage of fines (F_{75}) play a major role for SPT and V_s-based models, but for CPT-based models normalized average cone tip resistance (q_c) and peak horizontal ground acceleration (a_{max}) are more crucial parameters. The study also compares the developed models with previously published AI and ML-based models as well as the traditional empirical models.

2. Performance fitness and error metrics

2.1 Binary classification

Classification problems are one of the main AI applications, they can be of different types such as binary classification, Multi-class Classification, Multi-label Classification, Imbalanced Classification, Hierarchical Classification, etc. Liquefaction problems mainly lie in the domain of binary classification as the susceptibility towards liquefaction is denoted by either liquefied or non-liquefied cases (Kotsiantis *et al.* 2006). For the scope of this work liquefied cases are represented by '1' and non-liquefied cases are denoted by '0'. The basic terminology used in the binary classification is as follows:

TP: Total Nos. of correctly classified liquefaction instance

TN: Total Nos. of correctly classified non-liquefaction instance

FN: Total Nos. of misclassified liquefaction instance

FP: Total Nos. of misclassified non-liquefaction instance.

2.2 Performance metrics

The performance of the model can be calculated using different performance metrics. For the present study performance metrics are chosen based on their applicability in the unbalanced dataset. To evaluate the classification accuracy of different algorithms numerous performance metrics have been used in the literature. Some of the metrics used in this study are the rate of accuracy (*ACC*) or overall accuracy (*OA*), the precision rate (*PRE*), the recall rate (*REC*), the F_1 score, and *MCC* (Naser and Alavi 2021). The advantage of *MCC* is that it works very well where the classes are of different sizes (Naser and Alavi 2021).

ACC can be described as the total fraction of correctly classified samples. *PRE* is described as the proportion of correctly classified positive samples to that of total samples that are classified as positive, which also includes the misclassified positive sample. The value of *REC* can be calculated by dividing the total number of the correctly classified positive samples by the total number of available positive samples in the present dataset. The values of the performance metrics *PRE* and *REC* are restricted mutually so that they do not become high simultaneously (Zhou *et al.* 2021). Hence, to balance the harmonic mean of these two is taken which is called the F_1 score. To measure the quality of binary classification analysis *MCC* has been used, *MCC* is favorable for both binary as well as multi-class classification. The statistical formulations of the above metrics are defined in Eqs. (1)-(5).

$$ACC\ or\ OA = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$PRE = \frac{TP}{TP + FP} \tag{2}$$

$$REC\ or\ A_L = \frac{TP}{TP + FN} \tag{3}$$

$$F_1 = 2\frac{PRE \times REC}{PRE + REC} = \frac{2TP}{2TP + FP + FN} \tag{4}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5}$$

In these formulations, the notations of *TP, TN, FP,* and *FN* remain the same as discussed previously. Several researchers also suggested new performance indicators for classification problems. Weiss and Provost (2003) used the following formulation to determine the overall accuracy of the proposed model for both classes. Zhou *et al.* (2021) named it as accuracy rate. The individual accuracy of both classes can also be defined using Eqs. (6) and (7)

$$A_L\ or\ REC = \frac{TP}{TP + FN} \tag{6}$$

$$A_{NL} = \frac{TN}{TN + FP} \tag{7}$$

Here A_L and A_{NL} measure the accuracy of a class instance of liquefied and non-liquefied soils. In the quantitative comparison presented by Oommen *et al.* (2010), A_L and A_{NL} are termed recall liquefied and recall non-liquefied respectively. A good classifier model can be ensured not only by lower values of A_L and A_{NL} but also by the close values of both metrics.

Due to the problem of misleading and biased behavior of a classifier on the imbalanced dataset (Yuan and Liu 2012) and for better efficiency of computational algorithm another performance indicator is used in some literature, that is called G_{mean} (Kubat *et al.* 1997) and can be calculated using Eq. (8).

$$G_{mean} = \sqrt{A_L \times A_{NL}} \tag{8}$$

The performance accuracy of the classifier model can also be defined in terms of $G_{mean(error)}$ which can be evaluated using Eq. (9).

$$G_{mean\ (error)} = 1 - G_{mean} \tag{9}$$

The value of the binary classification performance metric varies between 0 and 1. Where 0 indicates a cent percent accurate model and 1 indicates a model with zero percent classification accuracy.

The performance of different AI-based classification models can also be evaluated in terms of receiver operating characteristics (*ROC*), which represents the relationship between *TP* and *FP* within a threshold (set as 0.5 for this study). The area under the *ROC* is called as *ROC* score and is used for comparing the performances of different classification models. The *AUROC* usually ranges from 0 to 1. A score closer to 1 represents a good model, and a score closer to 0 represents a bad or worse model (Dadhich *et al.* 2023).

2.3 Ranking of the models

The current study adopts a simple methodology to rank the models based on multiple performance indicators. The ranking of the models is first performed based on individual performance metrics then these individual ranks are summed up to obtain the model that has the lowest score i.e. the best ranking (Mishra *et al.* 2017). The individual ranking is assigned based on the significance of the respective performance indicator. The indicators *ACC* or *OA, PRE, REC, MCC,* F_1 represent the best model if it attains a value closer to 1 for these metrics. Therefore, the model having a value closer to 1 for these metrics will attain a rank of one based on these indicators. For A_L, A_{NL} the best model acquires a value close to 100% and the worst model attains an accuracy of 0%. To that reference, a model having A_L, A_{NL} values closest to 100% will get the first rank and other models will get the subsequent ranking in

descending order. Similarly, the best ranking will be provided to the model having $G_{mean(error)}$ value closest to 0%.

3. Adopted methodology

The current study adopts ensemble ML models for the development of the prediction model. Ensemble algorithms are very popular machine learning techniques that offer several advantages over other algorithms. These algorithms offer increased accuracy, better generalization, robustness, flexibility, and improved interpretability. These algorithms incorporate models that are trained on different data subsets, and they utilize the prediction power of several small models, this helps in better generalization of the model. These algorithms are less sensitive to changes in data and model parameters. Another advantage of using these algorithms is that they can generate an importance score of features, which can help in the identification of the most important features in the model-building process (Zhang and Ma 2012). This section provides a general introduction to the algorithms utilized in this study.

3.1 Ensemble algorithms

3.1.1 Simple Decision Tree (SDT) classifier

A Simple decision tree (SDT) classifier is a simple and powerful ML algorithm for the operations of classification and regression, it uses a tree-based model for decision-making based on the input parameters. In the process of making a decision tree, the data is recursively split based on the feature value. This split is made such that it can maximize the class separation, or in the regression problem it can minimize the error. The prediction for any new data can be made with the help of the resulting tree, it can be done by following the tree branches based on the input features value (Hastie *et al.* 2009). These DTs are simple to interpret and can handle numerical as well as categorical data. The robustness of DTs in handling outliers, missing values, and high-dimensional data makes it a favorable algorithm for classification problems. The problem with the SDT is that sometimes it's prone to overfitting and gives less reliable results with unbalanced datasets.

3.1.2 Random Forest (RF) classifier

Random forest (RF) was introduced by Breiman (2001), it is a very efficient method for solving unsupervised learning, classification, and regression problems. The RF algorithms divide the actual dataset into *n* numbers of bootstrap samples. For every bootstrap sample, a classification tree is constructed by randomly sampling some number of predictors and choosing the best split among these variables (it can be said as a generalized case of bagging) (Liaw and Wiener 2002) With the combination of bagging and random subsurface concept RF gives final results by taking majority votes in classification and average in regression. RF process mainly consists of two steps training and classification. In the training process classification model is developed using training samples and decision tree theory (Eisavi and Homayouni 2016). A simple flowchart of RF is shown in Fig. 1.

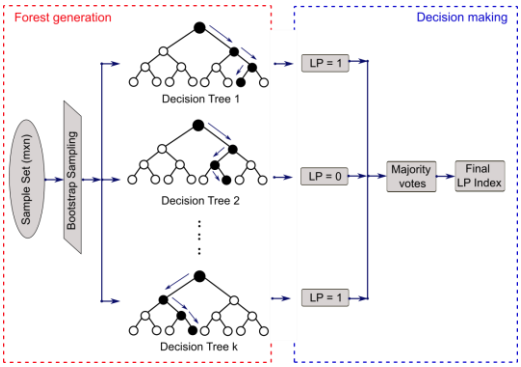


Fig. 1 Flow chart for the RF algorithm

3.1.3 Gradient Boosting (GB) classifier

Gradient boosting (GB) is an ensemble machine learning algorithm that is used extensively for the tasks of classification and regression. In GB a group of weak learners (called decision trees) collectively form a strong model. In this algorithm, a sequence of classifier decision trees is built, and every tree is trained in such a way that it tries to correct the errors of the tree formed before that. The main thing to note here is that the trees are not trained on actual data but on residual errors of the previous tree. To generate the final prediction this algorithm uses the aggregation of all the predictions made in the sequence (Friedman 2001). Similar to SDT, GB can also handle high-dimensional data and is robust to outliers. Other than that, the GB classifier can automatically capture the non-linear relationships among the input and target features.

3.1.4 Bagging classifier

Bagging or bootstrap aggregating is another ensemble learning algorithm for classification and prediction problems. Bagging usually involves the development of multiple typical decision trees with the help of training data subsamples, and then making use of the voting and averaging mechanism to combine their predictions. Bootstrapping is a mechanism of drawing a random data subset from training data, without replacement. Each bagging ensemble model is trained exactly on the same mechanism (Breiman 1996). The bagging technique reduces the overfitting in the model and increases the prediction's stability. This algorithm is easy to implement and is very handy for high-dimensional data with outliers and noise.

3.1.5 Adaptive Boosting (AdaBoost) classifier

AdaBoost or adaptive boosting belongs to the domain of classification algorithms in machine learning techniques. AdaBoost works on the same principle as GB, first, it forms a few weak classifiers and then uses their classification power to generate a strong classifier. These weak learners are trained using the training data, these learners are focused mainly on the samples that were not classified correctly by the parent learners. While in the training phase, AdaBoost assigns higher weights to misclassified samples so that they can be given more weightage by the next stage of weak learners. The accuracy of the final classifier is significantly improved by this adaptive weighing. A

weighted majority vote is used to combine all the predictions from the weak learners. AdaBoost is efficient in capturing nonlinear input and target variable relationships, this algorithm takes care of the overfitting and works better than many other ensemble learning algorithms (Freund and Schapire 1997).

3.1.6 Voting classifier

In the family of ensemble learning, the voting classifier is one of the algorithms used for the task of classification, this algorithm also makes a combined decision based on multiple independent models called classifiers. This can be used with both hard as well as soft voting. In hard voting, the final prediction is based on the mode of the individual classifier's prediction, this implies that the final prediction will be based on the class with the most votes. On the other hand, in soft voting, each classifier produces a probability-based distribution over classes, and the resultant prediction will be based on the highest average probability among all the classifiers (Kuncheva 2004). Voting is very efficient in handling numerical as well as categorical data. Its robust framework can also deal well with the noise and outliers in the dataset.

3.1.7 Stacking classifier

Stacking and ensemble machine learning algorithms are suitable for tasks of classification. This works by training multiple base classifier models using training data, and then utilizing the obtained predictions as inputs to the meta-classifier to make final predictions. Stacking classifiers are applicable for both categorical as well as numerical data, and are very helpful in capturing complex relationships between the inputs and the target variables. Stacking outperforms bagging and boosting in terms of avoiding overfitting in the model (Wolpert 1992).

3.1.8 Naive Bayes (NB) classifier

Naïve bayes (NB) classifier is mostly used for classification problems and is based on the concept of Bayes theorem. Bayes theorem provides the probability of a target class based on the given evidence or features. NB classifier assumes that the individual features are independent of each other. So, the overall probability of a particular set of features can be calculated by obtaining the products of the individual probabilities of those features. During the prediction operation, the NB classifier chooses the target plot that has the highest probability. NB classifier is simple, robust, and fast to use. It is advantageous in the case of high-dimensional data of numerical or categorical type (Domingos 2012).

3.1.9 K-Nearest Neighbor (K-NN) classifier

K-nearest neighbor (K-NN) classifier is a non-parametric machine learning algorithm. Being a non-parametric algorithm, it makes no assumptions about the data distribution. The K-NN algorithm works in feature space by finding out K training examples that may be located closer to the query point. Then K-NN chooses the most common K nearest neighbor and assigns the query point to that class. Euclidean distance between two data points is calculated in the feature space. K-NN classifier is a simple and flexible algorithm that can identify complex decision boundaries. This algorithm can be applied to binary as well as multi-class classification problems (Cover and Hart 1967).

3.1.10 Logistic Regression (LR) classifier

A logistic regression (LR) classifier is a linear models-based ML algorithm that predicts the target class based on the input features. The LR classifier uses the sigmoid function to model the correlation between the input parameter and the probability of the positive class. The sigmoid function scales the input values on a scale of 0 and 1. During the training operation, the LR classifier estimates the weights and biases using the training data. During the prediction operation, based on the observed features the probability of a positive class is calculated, and then based on the threshold value a classification between positive and negative classes is made. The main advantage of the LR classifier is its simplicity, easy interpretation, and its efficiency in handling numerical as well as categorical data. LR classifier is also an efficient algorithm for handling high-dimensional data and making use of regularization to prevent overfitting of the model (Hastie *et al.* 2009).

3.1.11 Support Vector (SV) Classifier

Support vector (SV) classifier is another ML algorithm frequently used for binary and multi-class classification. This algorithm works as a discriminative model that forms a hyperplane that can separate the binary classes efficiently in the feature space. This hyperplane usually consists of a set of weights and biases; these terms are defined by an optimization algorithm based on the training data. SV classifier uses quadratic programming (QP) for optimization. QP finds out the optimal weights and biases by solving a constrained optimization problem. In some of the cases, data may not be separated by a linear function, in those cases, the SV classifier transforms the data into a higher-dimensional space to make it linearly separable with the help of a kernel function (Cortes and Vapnik 1995). SV classifier is very efficient in capturing complex solution boundaries, and it can also be regularized for the prevention of overfitting.

3.1.12 Extreme Gradient Boosting (XGBoost) classifier

The XGBoost algorithm was primarily proposed by Chen and Guestrin (2016). This algorithm works within the framework of Gradient Boosting. It improves performance and efficiency by constantly adding new decision trees in multiple iterations to fit a value (Jiao *et al.* 2021). XGBoost holds a well-known reputation in competition because of the good efficiency and flexibility it offers. XGBoost avoids overfitting by adding a term for regularization that can help in smoothing final weights (Chen and Guestrin 2016). Apart from that XGBoost also performs sampling in both rows and columns to solve the issue of over-fitting.

3.1.13 Extra Tree (ET) classifier

An extra tree (ET) classifier is a prominent ensemble

ML algorithm suitable for classification tasks. ET classifier combines several DTs to improve prediction accuracy and robustness. Each tree of the ET classifier is built during the training phase using a group of random input variables. For each input feature, the ET classifier selects the best split using a random threshold. While performing prediction, The ET classifier uses the concept of majority vote or the method of weighted average to combine all the probabilities. In most of the terms ET classifier can be considered similar to the RF classifier, but it also has some key differences. The ET classifier uses a lot more DTs with a lesser amount of pruning and feature selection which usually results in higher variance but lower bias in comparison to the RF classifier. Due to the simple structure of ET with fewer hyper-parameters, it works faster in comparison to the RF classifier (Geurts *et al.* 2006).

3.1.14 Hist Gradient Boosting (HGB) classifier

Hist gradient boosting (HGB) Classifier is an extended version of the Gradient Boosting Classifier. In HGB each DT is formed based on the input feature's histogram instead of individual data points. HGB iteratively adds DTs to the ensemble, and each tree tries to rectify the error made by the previous tree. HGB uses the histogram-based split for the decision tree along with a technique named gradient-based one-side sampling (GOSS) for the reduction of the number of training samples to speed up the model training and for the reduction of overfitting (Chen and Guestrin 2016). HGB is well known for its ability to handle sparse data with high dimensionality.

3.1.15 Light Gradient Boosting Machine (LightGBM) classifier

LightGBM (Light Gradient Boosting Machine) is an ML algorithm that is widely used in different engineering fields due to its high efficiency and accuracy (Machado *et al.* 2019). This algorithm is known for its scalability and efficacy in performing classification and regression tasks. This algorithm is based on the framework of DT. LightGBM adopts an innovative leaf-wise strategy for tree growth, that maximizes the accuracy and minimizes the model complexity. LightGBM is commonly employed because of its fast speed, low memory consumption, and ability to process massive data by distributed support (Sui *et al.* 2023). More details of the LightGBM can be found in Machado *et al.* (2019).

3.2 Model hyperparameters

The performance of the ML models is strongly dependent on the hyperparameters used for their development. To provide a more comprehensive explanation of the employed ensemble methods the details regarding the hyperparameter are presented in Table 1 of the manuscript. These parameter ranges are chosen based on the insight from published literature as well as the machine leaning community. An exhaustive grid search technique has been used to generate the best combination of the hyperparameters (Suryadi *et al.* 2024). The best combination of the hyperparameters is decided based on a performance metric (Laghmati *et al.* 2024). Because the exhaustive search is computationally expensive for large grids, therefore a specific size of the grid was maintained by a careful literature survey to find the range for tuning hyperparameters.

4. Database and preprocessing

AI-based methods rely mostly on the quality of the dataset utilized in the model-building process. The current study works on the hypothesis that the geotechnical field investigation data can be used efficiently for the prediction of the liquefaction event. A comprehensive database of SPT and CPT has been used along with the measurements of V_s from different locations and events around the globe. The SPT dataset contains 620 case studies from two major earthquakes in Turkey and Taiwan in 1999, out of which 330 came from the Kocaeli earthquake and the remaining 290 were collected from the Taiwan earthquake. The information was collected by performing 46 soil boring with multiple SPT observations following the 1999 earthquake in Kocaeli, Turkey, and around 98 soil borings with SPT observations in the region of the 1999 earthquake which occurred in Chi-Chi, Taiwan. The dataset was interpreted using 38 SPT borings in the region of the Turkey earthquake and 25 borings of SPT from the Taiwan earthquake region. This observation was also accompanied by spectral analysis of surface wave test (SASW), and seismic cone penetration tests for the measurement of shear wave velocity (Hanna *et al.* 2007). This dataset consists of 620 data points with the proportion of liquefied and non-liquefied cases as 256 and 364 respectively. Another dataset that has been used in this study adopted from Moss *et al.* 2006, consists of 226 CPT test cases with a ratio of liquefied to non-liquefied instances as 133:93. This CPT data was documented from numerous earthquake sites from 1944 to 1995, Some of which are Argentina Earthquake 1977, Elmore ranch Earthquake 1987, Fukui Earthquake 1948, etc.

Instead of following only one type of input parameter for model testing, this study adopts a more comprehensive approach to consist of all soil and seismic parameters that are available for the field dataset. The authors believe that it is very helpful for the model robustness to include input features coming from different concepts such as SPT, CPT, and shear wave velocity measurements. The SPT database consists of 12 basic soil and seismic parameters including the V_s measurement. This includes the parameters M_w = moment magnitude of the earthquake; d = depth of soil layer in meter; d_w = depth of groundwater table in meter; σ_v = total vertical stress; σ'_v = effective vertical stress; a_{max} = peak horizontal ground acceleration (g); a_t = threshold acceleration (g); CSR = cyclic stress ratio; $(N_1)_{60}$ = corrected SPT value; F_{75} = Percentage of fines less than 75μ; V_s = shear wave velocity; and ϕ = internal friction angle of soil. This CPT database consists of the following input parameters: M_w = moment magnitude of the earthquake; d = depth of soil layer in meter; d_w = depth of groundwater table in meter; σ_v = total vertical stress; σ'_v =

Table 1 Details regarding the range of ML model hyperparameters along with the tuned hyperparameters

Model	Tuned Hyperparameters	Hyperparameters Grid
SDT	Criterion = 'gini'; Splitter = 'best'; Max depth = 5; Min Sample Split = 2; Min Sample Leaf = 1	param_grid = { 'criterion': ['gini', 'entropy'], 'max_depth': [None, 5, 10, 15, 20], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4]}
RF	N_estimators = 1000; Learning Rate = 0.1; Criterion = 'gini'; Splitter = 'best'; Max depth = 110; Min Sample Split = 2; Min Sample Leaf = 2; Max Features = 'Auto', Bootstrap = 'True'	param_dist = { 'n_estimators': [int(x) for x in np.linspace(start=200, stop=2000, num=10)], 'max_features': ['auto', 'sqrt'], 'max_depth': [int(x) for x in np.linspace(10, 110, num=11)] + [None], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'bootstrap': [True, False]}
GB	N_estimators = 100; Learning Rate = 0.1; Criterion = 'friedman_mse'; Max depth = 3; Min Sample Split = 5; Min Sample Leaf = 2; Max Features = 'None'; subsample = 0.9	param_dist = { 'n_estimators': [int(x) for x in np.linspace(start=100, stop=1000, num=10)], 'learning_rate': [0.001, 0.01, 0.1, 0.2, 0.3], 'max_depth': [3, 4, 5, 6, 7, 8], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'subsample': [0.5, 0.7, 0.9, 1.0],}
Bagging	N_estimators = 50; Learning Rate = 0.1; Max Features = 1; Bootstrap = 'True'; Max Samples =0.7; Bootstrap Features = 'True'	param_dist = { 'n_estimators': [10, 50, 100, 200], 'max_samples': [0.5, 0.7, 0.9, 1.0], 'max_features': [0.5, 0.7, 0.9, 1.0], 'bootstrap': [True, False], 'bootstrap_features': [True, False]}
AdaBoost	N_estimators = 100; Learning Rate = 0.5; Algorithm = 'SAMME'	param_dist = { 'n_estimators': [50, 100, 200, 300, 400], 'learning_rate': [0.001, 0.01, 0.1, 0.2, 0.3, 0.5, 1.0], 'algorithm': ['SAMME', 'SAMME.R']}
Voting	Estimators = [Random Forest]; Voting='Hard'	param_dists = { 'dt__max_depth': [None, 5, 10, 15], 'rf__n_estimators': [50, 100, 200],} random_search = RandomizedSearchCV(estimator=voting_classifier, param_distributions=param_dists, n_iter=100, cv=5, random_state=42) random_search.fit(X_train, y_train)
Stacking	Final_estimator = [Random Forest]; Stacking method= 'Auto'	param_dists = { 'dt__max_depth': [None, 5, 10, 15], 'rf__n_estimators': [50, 100, 200],} random_search = RandomizedSearchCV(estimator=stacking_classifier, param_distributions=param_dists, n_iter=100, cv=5, random_state=42) random_search.fit(X_train, y_train)
NB	Gaussian Naive Bayes	No tuning parameters
K-NN	N_neighbours = 9; Weights = 'Uniform'; leaf size = 30, p = 2; Metric = 'Euclidean'	param_grid = { 'n_neighbors': [3, 5, 7, 9, 11], 'weights': ['uniform', 'distance'], 'metric': ['euclidean', 'manhattan', 'chebyshev']}
LR	Penalty = 'l2'; C = 1; fir_intercept = 'True'; Solver = 'lbfgs'; Max Iterations = 100; Multi class = 'Auto', Class Weight = 'None'	param_grid = { 'C': [0.001, 0.01, 0.1, 1, 10, 100], 'penalty': ['l1', 'l2']}
SV	C = 10; Kernel = 'sigmoid'; Gamma = 'Scale'; Degree = '4'; Shrinking = 'False'; Class Weight = 'None'	param_grid = { 'C': [0.001, 0.01, 0.1, 1, 10, 100], 'kernel': ['linear', 'poly', 'rbf', 'sigmoid'], 'gamma': ['scale', 'auto'] }
XGBoost	Learning rate = 0.1; N_estimators = 200; Max Depth = 6; Min Child Weight = 5; Subsample = 0.8; Gamma = 0.1; L1 regulaization = 0.001; L2 regulaization = 0.1; Scale pos weight = 1; Colsample bytree = 0.8	param_grid = { 'learning_rate': [0.01, 0.1, 0.2, 0.3], 'max_depth': [3, 4, 5, 6, 7], 'min_child_weight': [1, 3, 5, 7], 'subsample': [0.6, 0.7, 0.8, 0.9, 1.0], 'colsample_bytree': [0.6, 0.7, 0.8, 0.9, 1.0], 'n_estimators': [100, 200, 300, 400, 500]}
ET	N_estimators = 100; Criterion = 'gini'; Max depth = 'None'; Min Sample Split = 5; Min Sample Leaf = 2; Max Features = 'Auto', Bootstrap = 'True'; Class Weight = 'None'	param_grid = { 'n_estimators': [50, 100, 200, 300, 400], 'max_depth': [None, 5, 10, 15, 20], 'min_samples_split': [2, 5, 10], 'min_samples_leaf': [1, 2, 4], 'max_features': ['auto', 'sqrt', 'log2']}
HGB	Learning rate = 0.01; Max Iteration = 100; Max Depth = 'None'; Max Leaf Nodes = 35; Min samples Leaf = 20; L2 Regularization = 0.1; Max bins = 256; Min samples Split = 5	param_grid = { 'learning_rate': [0.01, 0.1, 0.2], 'max_iter': [100, 200, 300], 'max_depth': [None, 3, 5, 7], 'max_bins': [10, 20, 30, 50,128,256]}
LightGBM	Boosting type = 'gbdt'; Learning rate =0.01; Max Iteration = 100; Max Depth = 5; Number of Leaf Nodes = 31; N_estimators = 100; Min Child Samples = 50; Subsample = 0.9	params = { 'boosting_type': ['gbdt', 'dart', 'goss'], 'num_leaves': [31, 50, 100], 'learning_rate': [0.01, 0.1, 0.2], 'n_estimators': [50, 100, 200], 'max_depth': [-1, 5, 10], }

Legend: see the List of Abbreviations and List of Symbols

effective vertical stress; a_{max} = peak horizontal ground acceleration (g); r_d = non-linear shear mass participation factor; CSR = cyclic stress ratio; R_f = friction ratio; q_c = normalized average cone tip resistance; and f_s = cone sleeve resistance.

The selection of the data samples relies upon their pertinence to the research question and problem in hand. For the consistency of the dataset, data points with uniform formats and structures with consistent measurement units have been selected. To ensure the high-quality of data,

Table 2 The statistical summary of SPT Dataset with V_s measurements

Variable	Mean	StDev	Variance	Minimum	Q1	Median	Q3	Maximum	Skewness	Kurtosis
M_w	7.49	0.10	0.01	7.40	7.40	7.40	7.60	7.60	0.13	-1.99
d (m)	7.66	4.90	23.98	0.80	3.80	6.70	10.20	19.80	0.71	-0.42
d_w (m)	1.45	1.20	1.44	0.35	0.71	1.10	1.78	10.00	3.52	19.75
σ_v (kPa)	144.60	98.20	9643.65	12.10	67.70	121.60	202.75	408.90	0.78	-0.31
σ'_v (kPa)	82.48	52.84	2792.18	7.50	41.60	68.15	113.97	233.70	0.82	-0.23
α_{max}/g	0.38	0.15	0.02	0.18	0.38	0.40	0.40	0.67	0.40	-0.14
α_t/g	0.07	0.07	0.01	0.00	0.04	0.06	0.08	0.85	5.19	43.57
CSR	0.37	0.15	0.02	0.12	0.24	0.39	0.45	0.77	0.34	-0.26
$(N_1)_{60}$	14.48	11.39	129.64	1.00	7.00	11.00	18.00	75.00	1.67	3.27
F_{75} (%)	62.99	34.28	1174.80	1.00	29.00	74.50	96.00	100.00	-0.46	-1.37
V_s (m/s)	166.98	67.09	4500.85	37.00	130.00	155.00	200.00	500.00	1.27	3.44
Φ (◦)	31.96	4.85	23.50	23.46	28.40	31.41	34.70	52.08	0.83	0.49

Legend: see the List of Abbreviations and List of Symbols

Table 3 The statistical summary of CPT Dataset

Variable	Mean	StDev	Variance	Minimum	Q1	Median	Q3	Maximum	Skewness	Kurtosis
M_w	6.95	0.44	0.19	6.00	6.60	7.10	7.10	7.60	-0.88	0.24
d (m)	5.66	2.89	8.36	1.40	3.50	4.80	6.93	14.10	1.06	0.50
d_w (m)	3.29	2.74	7.53	0.00	1.22	2.50	4.26	12.74	1.40	1.43
σ_v (kPa)	106.89	55.36	3064.65	26.60	67.25	90.30	128.82	274.00	1.08	0.56
σ'_v (kPa)	74.65	34.40	1183.05	22.50	51.80	62.80	97.20	215.20	1.06	0.99
α_{max}/g	0.29	0.14	0.02	0.08	0.19	0.25	0.37	0.80	1.06	0.55
r_d	0.95	0.04	0.00	0.80	0.95	0.96	0.97	0.99	-2.24	5.09
CSR	0.25	0.11	0.01	0.07	0.16	0.21	0.32	0.68	1.10	1.29
R_f	1.22	1.05	1.10	0.10	0.48	0.90	1.80	5.20	1.64	2.75
q_c (MPa)	5.82	4.09	16.75	0.90	2.98	4.90	7.50	25.00	1.62	3.41
f_s (kPa)	59.26	70.68	4995.44	1.00	24.50	39.95	66.40	562.60	3.95	19.37

Legend: see the List of Abbreviations and List of Symbols

unrelated and irrelevant samples were excluded and only reliable and accurate data was considered for the analysis. Noisy, incomplete, and erratic data points were filtered out by careful inspection. The outliers in the dataset were identified using the statical method incorporating interquartile range (IQR). However, these outliers were not removed as the ensemble algorithms are found to be performing very well even with a dataset containing outliers. The data is checked for the presence of any duplicates. Feature engineering was performed to handle any missing values and to tailor the input parameters. Some parameters were directly adopted from the dataset and some were derived from the primary parameters to provide more accuracy to the model. As the data was mainly collected from the Turkey and Taiwan earthquakes therefore a sampling bias is unavoidable. However, the data from these regions cover a wide spectrum of soil properties as well as seismic parameters (Hanna *et al.* 2007) as can be observed from Table 2-3. The variations in the techniques of measurement for different input parameters may contribute to the measurement bias, it mainly depends on the testing procedure employed while collecting the sample data.

Table 2 presents the summary of SPT and V_s data and Table 3 presents the CPT dataset. To better understand the distribution and characteristics of the datasets, some basic statistical measures have been studied. Looking at the central tendency and variation in the dataset it can be clearly stated that the parameters have a vast variation in their range. Therefore, if the original parameter values are used, then an input feature with a higher magnitude and range will more strongly affect the input-output relationship, in other terms will have more weight, and the inputs will have less significant magnitudes will lose their importance. To tackle this problem, the current study uses the concept of normalization to scale all the parameters in the range of 0 to 1. The formula for the normalization is provided in Eq. (10)

$$X_{Normalized} = \frac{X - X_{Minimum}}{X_{Maximum} - X_{Minimum}} \tag{10}$$

Here, $X_{Normalized}$ represents the normalized value of an individual data point, and $X_{Minimum}$ and $X_{Maximum}$ denote the minimum and maximum values of the particular feature values respectively. The dataset is highly imbalanced with liquefied cases and non-liquefied cases as 256:364, and

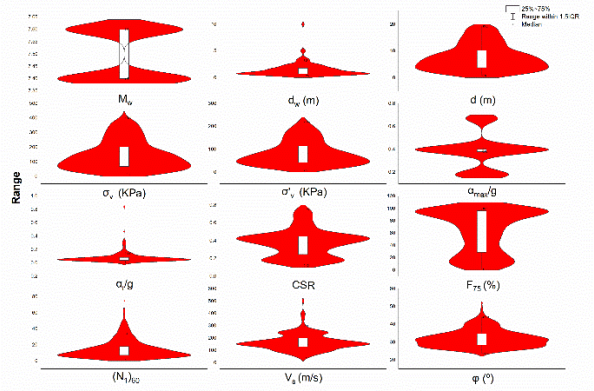


Fig. 2 Violin plot for the input parameters of the SPT dataset with V_s measurement

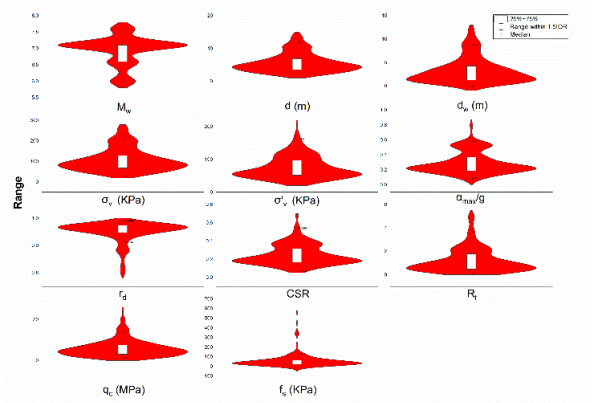
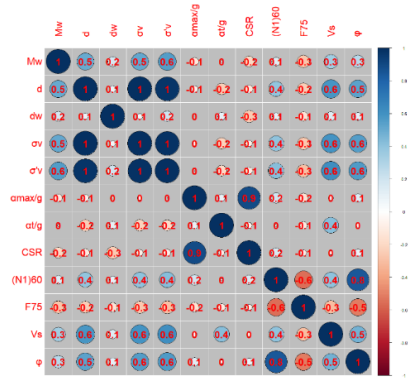
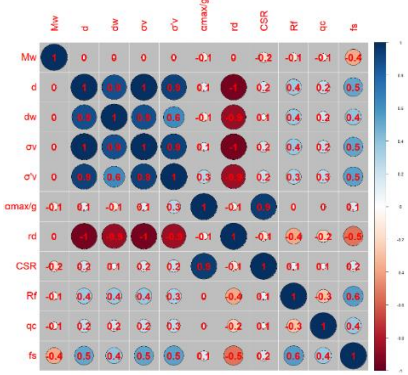


Fig. 3 Violin plot for the input parameters of the CPT dataset



(a)



(b)

Fig. 4 Correlation matrix showing Pearson's correlation coefficient for different pairs of input variables for (a) SPT dataset with V_s measurement, and (b) CPT dataset

133:93 for SPT (V_s) and CPT respectively. This study incorporates 5-fold cross-validation to evaluate the performance of the model. As a stratified 5-fold cross-validation technique is employed, the data is divided into 80% for training and 20% for testing.

Fig. 2 represents the distribution of SPT data in the form of a violin plot and a box plot. Looking at the plots it can be said that most of the input parameters are free from the outliers except in the water table depth and threshold acceleration measurements. The shear wave velocity data showed few outliers. However, the CPT dataset violin plot in Fig. 3 denotes the presence of a few outliers in the measurements of friction ratio, normalized average cone tip resistance, and cone sleeve resistance. This outlier aspect has not been discussed in previous studies. In the present study, outliers are not removed from the dataset because most of the ensemble algorithms are proven to handle outliers efficiently (Breiman 2001, Friedman 2001).

Figs. 4(a) and 4(b) presents Pearson's correlation coefficient matrix between input variables. The SPT and CPT-based data showed that most of the input features are either not correlated with each other or have a very weak correlation. Some of the parameters such as the depth of the soil layer and the depth of the groundwater table are strongly correlated with each other. This strong correlation is also well-known in several works of literature. In the same way, *CSR* is also having a strong correlation with a_{max}, this correlation is because the calculation of *CSR* itself involves the term a_{max} in its formulation (Seed and Idriss 1971). The correlational analysis is crucial for parameter selection for statistical model development as it makes sure that only mutually independent parameters are selected for model development. This not only reduces the model complexity but also helps in the quick convergence of the algorithm. However, in AI-based algorithms, this aspect is taken care of by the algorithms through the appropriate selection of the inputs.

5. Results and discussion

The present study utilizes a series of ensemble machine-learning algorithms for a superior model development that can efficiently predict the liquefaction instance at a given site. For the development of a generalized model, several trials and errors are performed by varying the number of input parameters with different combinations by random sampling technique and only the best model parameters are presented in the results. The results are presented based on datasets of SPT with V_s measurements and CPT. The accurate prediction of non-liquefied cases is very essential for the success of the model as one of the focuses of the study is to accurately predict the minority class instances. In addition, different performance metrics for the model are evaluated and compared with the results published in other literature based on the overall ranking of the models (Das *et*

al. 2020; Hoang and Bui 2018; Muduli and Das 2013, 2015, Oommen *et al.* 2010).

5.1 Results and discussion based on the SPT database with V_s measurement

For the SPT-based model with shear wave velocity measurements, different combinations of parameters were tested to obtain the best model. From the discussion in the literature, it can be inferred that for the better generalization ability of a particular model, the values of $G_{mean(error)}$ for training and testing should be as less as possible (Das *et al.* 2020). Table 4 presents the stratified 5-fold cross-validation results of the developed models along with their average values. From Table 4 it can be observe that the results in terms of performance metrics do not show any abrupt variation in their values. For the best-performing model of XGBoost the $G_{mean(error)}$ values of each fold vary from, 1.92% to 4.12%, resulting in an average $G_{mean(error)}$ of 2.79%. similar observations can be made for other models presented in Table 4.

The comparative performance of all SPT and V_s-based models is summarized in Table 5. From Table 5 it can be observed that XGBoost, with an F_1 score of 0.97 and $G_{mean(error)}$ of 2.79% outperforms all other algorithms previously published in the literature. The consistent model performance in different folds indicates lesser overfitting in the model. Subsequent rankings are occupied by LightGBM, Bagging, and RF. Although based on the $G_{mean(error)}$, LightGBM performs equivalent to XGBoost, it lacks in performance based on other metrics. The greedy-based stacking ensemble learning (SEL) model developed by Sahin and Demir (2023) achieves second place based on its overall performance. These algorithms outperform previously published algorithms ANN/MARS+ multi-objective symbiotic organisms search algorithm (MOSOS) ($G_{mean(error)}$ of 6.89% and 8.47%), ANN/MARS+ non-dominated sorting genetic algorithm-II (NSGA-II) ($G_{mean(error)}$ of 8.87% and 9.49%) (Das *et al.* 2020), MGGP ($G_{mean(error)}$ of 12.05%) (Muduli and Das 2013), and others. The best XGBoost model not only performs well in classifying majority class instances i.e., non-liquefaction cases (A_{NL} = 98.19%) but also shows an accuracy of 96.25% in classifying minority class instances i.e., liquefied cases (A_L). The developed models are also compared with the empirical method suggested by Seed and Idriss (1971), which attains an overall $G_{mean(error)}$ of 33.22%. Other model comparisons are also presented in Table 5. This showed that ensemble algorithms can be very useful in tackling the problem of an unbalanced dataset. It may be mentioned here that these models are developed using all 12 input parameters namely: M_w, d, d_w, σ_v, σ'_v, a_{max}/g, a_l/g, CSR, $(N_1)_{60}$, F_{75} (%), V_s, and ϕ, whereas in previous studies V_s has not been considered as the input along with the SPT database. Hence, a combined SPT database with V_s measurement might be efficient in predicting the liquefaction susceptibility of soil.

Fig. 5 shows the area under the receiver operating curve (AUROC) for different models developed using the SPT database with V_s measurements. The algorithms having

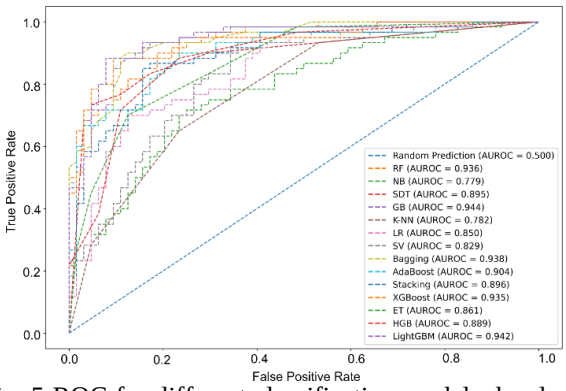


Fig. 5 ROC for different classification models developed using SPT dataset with V_s measurement

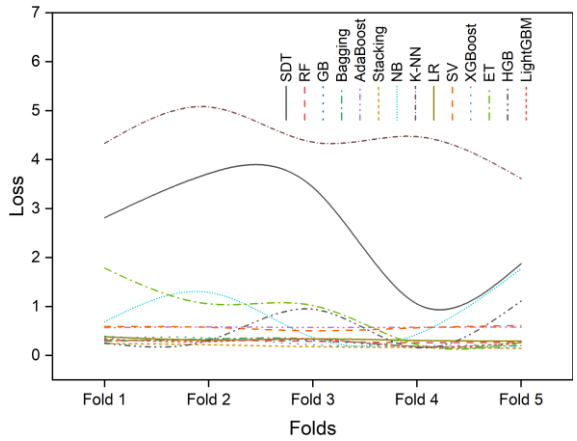


Fig. 6 Loss values per fold of the classification models developed using the SPT dataset with V_s measurement

AUROC as 1 can be said to be having 100% prediction accuracy while those having an AUROC value of 0 can be said to be having a prediction accuracy of 0%. Hence, it can be inferred from Fig. 5 that the XGBoost with an AUROC of 0.965 represents the best classification model followed by GB (0.944), LightGBM (0.942), and RF (0.936) respectively.

To monitor the performance of the models, loss values over folds are calculated and presented in Fig. 6. The loss values over folds indicate that the model performance is consistent over each fold, this represents a more generalized solution. The higher values of loss function for K-NN and SDT indicate poor performance. Fig. 7 shows the bar chart of input feature weight obtained from some of the algorithms used in the present study. These weights are assigned to different features (input parameters) during the development process of a model, which can be considered as the importance of the respective input parameters. Considering the best algorithms from Table 5, it can be observed from Fig. 7 that XGBoost considers the input parameters F_{75}, ϕ, *and* CSR as the most important parameters in descending order. While interpreting these inputs from the physical point of view, the fine fraction, strength of the soil, and seismic parameters are considered. GB and LighGBM also show a similar trend as XGBoost. RF considers F_{75} as the most important parameter followed

Table 4 Fold-wise classification performance of the developed ML models based on the SPT data with V_s measurements

Model	Overall							
	ACC or OA	PRE	REC	F_1	MCC	A_L (%)	A_{NL} (%)	$G_{mean(error)}$ (%)
SDT: Fold 1	0.87	0.86	0.82	0.84	0.73	82.03	90.66	13.76
SDT: Fold 2	0.88	0.87	0.84	0.85	0.75	83.59	91.21	12.68
SDT: Fold 3	0.86	0.81	0.86	0.83	0.71	85.94	85.44	14.31
SDT: Fold 4	0.88	0.86	0.84	0.85	0.75	83.98	90.66	12.74
SDT: Fold 5	0.88	0.80	0.94	0.86	0.77	94.39	84.33	10.78
SDT: Average	0.87	0.84	0.86	0.85	0.74	85.99	88.46	12.86
RF: Fold 1	0.96	0.95	0.95	0.95	0.92	95.31	96.70	3.99
RF: Fold 2	0.97	0.97	0.96	0.96	0.94	95.70	97.80	3.25
RF: Fold 3	0.97	0.97	0.96	0.96	0.94	95.70	97.80	3.25
RF: Fold 4	0.98	0.98	0.96	0.97	0.95	95.70	98.90	2.71
RF: Fold 5	0.96	0.98	0.93	0.96	0.93	92.97	98.90	4.11
RF: Average	0.97	0.97	0.95	0.96	0.93	95.08	98.02	3.46
GB: Fold 1	0.91	0.92	0.86	0.89	0.81	85.94	94.51	9.88
GB: Fold 2	0.92	0.92	0.88	0.90	0.84	88.28	94.78	8.53
GB: Fold 3	0.90	0.91	0.85	0.88	0.80	85.16	94.23	10.42
GB: Fold 4	0.91	0.93	0.86	0.89	0.82	85.94	95.33	9.49
GB: Fold 5	0.90	0.93	0.84	0.88	0.80	83.59	95.33	10.73
GB: Average	0.91	0.92	0.86	0.89	0.82	85.78	94.84	9.81
Bagging: Fold 1	0.97	0.97	0.95	0.96	0.93	94.92	98.08	3.51
Bagging: Fold 2	0.96	0.96	0.95	0.95	0.92	94.53	97.25	4.12
Bagging: Fold 3	0.97	0.97	0.96	0.97	0.95	96.48	98.08	2.72
Bagging: Fold 4	0.97	0.98	0.96	0.97	0.95	96.09	98.35	2.78
Bagging: Fold 5	0.97	0.98	0.95	0.96	0.94	94.53	98.63	3.44
Bagging: Average	0.97	0.97	0.95	0.96	0.94	95.31	98.08	3.32
AdaBoost: Fold 1	0.92	0.89	0.91	0.90	0.83	91.41	92.31	8.14
AdaBoost: Fold 2	0.91	0.89	0.89	0.89	0.81	89.45	92.03	9.27
AdaBoost: Fold 3	0.91	0.89	0.89	0.89	0.81	88.67	92.31	9.53
AdaBoost: Fold 4	0.91	0.89	0.89	0.89	0.82	89.06	92.58	9.19
AdaBoost: Fold 5	0.91	0.89	0.88	0.89	0.81	88.28	92.31	9.73
AdaBoost: Average	0.91	0.89	0.89	0.89	0.82	89.38	92.31	9.17
Voting: Fold 1	0.85	0.76	0.92	0.83	0.71	92.19	79.67	14.30
Voting: Fold 2	0.87	0.80	0.93	0.86	0.75	92.58	83.52	12.07
Voting: Fold 3	0.86	0.79	0.92	0.85	0.73	91.80	82.42	13.02
Voting: Fold 4	0.87	0.80	0.90	0.85	0.74	90.23	84.34	12.76
Voting: Fold 5	0.85	0.78	0.88	0.83	0.70	88.28	82.69	14.56
Voting: Average	0.86	0.79	0.91	0.84	0.73	91.02	82.53	13.33
Stacking: Fold 1	0.94	0.91	0.95	0.93	0.88	94.92	93.41	5.84
Stacking: Fold 2	0.95	0.94	0.93	0.94	0.90	93.36	96.15	5.25
Stacking: Fold 3	0.96	0.98	0.93	0.95	0.92	92.58	98.63	4.45
Stacking: Fold 4	0.96	0.98	0.92	0.95	0.92	91.80	98.90	4.72
Stacking: Fold 5	0.95	0.95	0.93	0.94	0.90	93.36	96.70	4.98
Stacking: Average	0.95	0.95	0.93	0.94	0.90	93.20	96.76	5.04
NB: Fold 1	0.68	0.58	0.84	0.68	0.41	83.59	57.14	30.89
NB: Fold 2	0.68	0.58	0.83	0.68	0.40	82.81	57.14	31.21
NB: Fold 3	0.69	0.58	0.82	0.68	0.41	82.03	59.07	30.39
NB: Fold 4	0.68	0.58	0.83	0.68	0.40	82.81	57.42	31.04
NB: Fold 5	0.68	0.58	0.83	0.68	0.40	83.20	57.14	31.05
NB: Average	0.68	0.58	0.83	0.68	0.41	82.89	57.58	30.91
K-NN: Fold 1	0.83	0.80	0.79	0.79	0.65	79.30	85.71	17.56
K-NN: Fold 2	0.82	0.80	0.75	0.78	0.63	75.39	86.54	19.23
K-NN: Fold 3	0.83	0.80	0.79	0.79	0.65	78.91	85.71	17.76
K-NN: Fold 4	0.84	0.82	0.79	0.80	0.67	78.52	87.91	16.92
K-NN: Fold 5	0.82	0.79	0.77	0.78	0.63	77.34	85.16	18.84
K-NN: Average	0.83	0.80	0.78	0.79	0.64	77.89	86.21	18.06
LR: Fold 1	0.74	0.68	0.68	0.68	0.45	67.97	77.47	27.43
LR: Fold 2	0.74	0.69	0.67	0.68	0.46	66.80	79.12	27.30
LR: Fold 3	0.74	0.70	0.66	0.68	0.47	65.63	80.49	27.32
LR: Fold 4	0.75	0.71	0.64	0.68	0.47	64.45	81.87	27.36
LR: Fold 5	0.73	0.69	0.64	0.66	0.45	63.67	80.22	28.53
LR: Average	0.74	0.70	0.66	0.68	0.46	65.70	79.84	27.57

Table 4 Continued-

SV: Fold 1	0.69	0.70	0.42	0.53	0.34	42.19	87.36	39.29
SV: Fold 2	0.67	0.68	0.37	0.48	0.30	37.11	87.91	42.88
SV: Fold 3	0.68	0.72	0.39	0.50	0.33	38.67	89.29	41.24
SV: Fold 4	0.67	0.74	0.32	0.44	0.31	31.64	92.31	45.96
SV: Fold 5	0.67	0.71	0.34	0.46	0.30	33.59	90.38	44.90
SV: Average	0.68	0.71	0.37	0.48	0.31	36.64	89.45	42.75
XGBoost: Fold 1	0.97	0.96	0.96	0.96	0.94	96.48	97.53	3.00
XGBoost: Fold 2	0.96	0.96	0.95	0.95	0.92	94.53	97.25	4.12
XGBoost: Fold 3	0.98	0.98	0.97	0.98	0.96	97.27	98.90	1.92
XGBoost: Fold 4	0.98	0.98	0.97	0.98	0.96	96.88	98.90	2.12
XGBoost: Fold 5	0.97	0.98	0.96	0.97	0.95	96.09	98.35	2.78
XGBoost: Average	0.97	0.97	0.96	0.97	0.95	96.25	98.19	2.79
ET: Fold 1	0.95	0.94	0.95	0.94	0.90	94.53	95.88	4.80
ET: Fold 2	0.95	0.93	0.95	0.94	0.90	95.31	95.05	4.82
ET: Fold 3	0.97	0.96	0.96	0.96	0.93	96.09	97.25	3.33
ET: Fold 4	0.97	0.97	0.96	0.96	0.94	95.70	98.08	3.12
ET: Fold 5	0.95	0.97	0.90	0.94	0.90	90.23	98.35	5.79
ET: Average	0.96	0.96	0.94	0.95	0.92	94.38	96.92	4.36
HGB: Fold 1	0.95	0.93	0.95	0.94	0.90	94.53	95.33	5.07
HGB: Fold 2	0.94	0.94	0.93	0.93	0.88	92.58	95.60	5.92
HGB: Fold 3	0.96	0.95	0.96	0.96	0.92	95.70	96.70	3.80
HGB: Fold 4	0.96	0.96	0.95	0.95	0.92	94.53	97.53	3.98
HGB: Fold 5	0.95	0.97	0.91	0.94	0.90	91.41	97.80	5.45
HGB: Average	0.95	0.95	0.94	0.94	0.91	93.75	96.59	4.84
LightGBM: Fold 1	0.97	0.96	0.97	0.97	0.94	97.27	96.98	2.88
LightGBM: Fold 2	0.96	0.96	0.95	0.95	0.92	95.31	96.98	3.86
LightGBM: Fold 3	0.98	0.98	0.96	0.97	0.95	96.48	98.63	2.45
LightGBM: Fold 4	0.98	0.97	0.97	0.97	0.95	97.27	98.08	2.33
LightGBM: Fold 5	0.98	0.98	0.96	0.97	0.95	96.48	98.63	2.45
LightGBM: Average	0.97	0.97	0.97	0.97	0.94	96.56	97.86	2.79

Legend: see the List of Abbreviations and List of Symbols

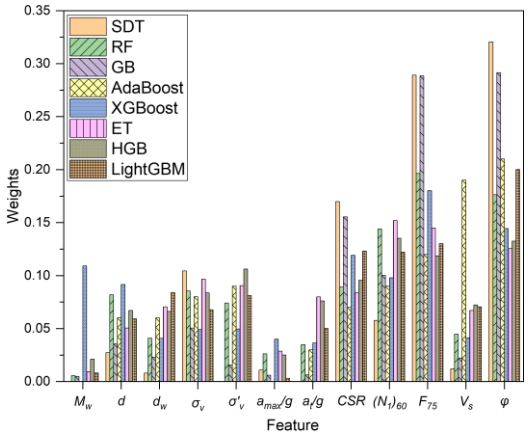


Fig. 7 Graph showing feature importance in terms of feature weights for different classification models developed for the SPT dataset with V_s measurements

by ϕ, $(N_1)_{60}$, and *CSR*, meaning emphasis on both ϕ and $(N_1)_{60}$.

The results from Sonmezer *et al.* (2020) indicate that relative density plays an important role in liquefaction potential determination, which in turn depends on the fines content. But, (Hoang and Bui 2018) identified $(N_1)_{60}$, V_s, a_{max}/g, M_w, and d_w as the most influential input parameters. On the other hand, for the SPT dataset without V_s, (Kubat *et al.* 1997) obtained $(N_1)_{60}$, a_{max}/g, and M_w as the parameters having the most impact on the model. Whereas for only the shear wave velocity data, they observed V_s, *CSR*, a_{max}/g, and M_w. It may be mentioned here that *CSR* is a function of a_{max}/g. It may be mentioned here that previous studies have not considered V_s parameter along with the SPT dataset. The variations can also be explained based on the efficacy of the algorithm used and its closeness to physical phenomena.

5.2 Results and discussion based on the CPT database

Similar to the SPT-based model, the CPT model is also developed after incorporating different parameters from CPT test results. The 5-fold cross-validation is applied to the dataset and the results from each fold are presented in Table 6. These results indicate that for each fold division of the dataset, the ensemble models perform very well in terms of performance metrics. The bagging model shows an overall $G_{mean(error)}$ of 1.32% with performance variation from 0.38% overall $G_{mean(error)}$ to 2.91% $G_{mean(error)}$. The GB and RF models perform subsequently with consistent performance in each fold as shown in Table 6.

For the CPT dataset, the tree-based algorithms show good performance, the best model is obtained from the bagging classifier which shows an overall $G_{mean(error)}$ of 1.32%, apart from that GB, RF, and XGBoost also shows very efficient performance with an overall $G_{mean(error)}$ of 1.76%, 1.92%, and 1.94% respectively as shown in Table 7.

Table 5 The classification performance comparison of present models with previous studies for the SPT data with V_s measurements

References	*L:NL (L/NL)*	Model	Overall								
			ACC or OA	PRE	REC	F_1	MCC	A_L (%)	A_{NL} (%)	$G_{mean(error)}$ (%)	Ranking
Present Study	256:364 (0.70)	SDT	0.87	0.84	0.86	0.85	0.74	85.99	88.46	12.86	29
		RF	0.97	0.97	0.95	0.96	0.93	95.08	98.02	3.46	6
		GB	0.91	0.92	0.86	0.89	0.82	85.78	94.84	9.81	20
		Bagging	0.97	0.97	0.95	0.96	0.94	95.31	98.08	3.32	4
		AdaBoost	0.91	0.89	0.89	0.89	0.82	89.38	92.31	9.17	19
		Voting	0.86	0.79	0.91	0.84	0.73	91.02	82.53	13.33	28
		Stacking	0.95	0.95	0.93	0.94	0.90	93.20	96.76	5.04	10
		NB	0.68	0.58	0.83	0.68	0.41	82.89	57.58	30.91	39
		K-NN	0.83	0.80	0.78	0.79	0.64	77.89	86.21	18.06	36
		LR	0.74	0.70	0.66	0.68	0.46	65.70	79.84	27.57	41
		SV	0.68	0.71	0.37	0.48	0.31	36.64	89.45	42.75	40
		XGBoost	0.97	0.97	0.96	0.97	0.95	96.25	98.19	2.79	1
		ET	0.96	0.96	0.94	0.95	0.92	94.38	96.92	4.36	8
		HGB	0.95	0.95	0.94	0.94	0.91	93.75	96.59	4.84	9
		LightGBM	0.97	0.97	0.97	0.97	0.94	96.56	97.86	2.79	3
Dadhich *et al.* (2023)	256:364 (0.70)	ANN	0.94	0.93	0.93	0.93	0.87	92.66	94.75	6.30	12
Sahin and Demir (2023)*	256:364 (0.70)	Greedy based SEL: Backwords	0.97	0.96	0.99	0.97	0.95	98.70	96.10	2.61	2
Wu *et al.* (2023)	419:377 (1.11)	HBF	0.92	0.91	0.95	0.93	0.84	94.75	89.39	7.97	14
Demir and Sahin (2022)*	256:364 (0.70)	AdaBoost	0.95	0.96	0.95	0.95	0.91	94.81	96.10	4.55	7
		XGBoost	0.97	0.97	0.96	0.97	0.94	96.10	97.40	3.25	5
		GBM	0.90	0.93	0.86	0.89	0.79	85.71	93.51	10.47	21
Das *et al.* (2020)[2]	287:124 (2.31)	ANN + NSGA-II	0.91	0.97	0.90	0.93	0.79	89.55	92.74	8.87	17
		MARS + NSGA-II	0.90	0.96	0.90	0.93	0.78	89.90	91.13	9.49	18
		ANN + MOSOS	0.93	0.97	0.93	0.95	0.84	92.68	93.55	6.89	11
		MARS + MOSOS	0.91	0.97	0.90	0.93	0.80	89.55	93.55	8.47	15
Das *et al.* (2020)[1]	109:84 (1.30)	ANN + NSGA-II	0.92	0.96	0.90	0.93	0.85	89.91	95.24	7.46	13
		MARS + NSGA-II	0.89	0.92	0.88	0.90	0.78	88.07	90.48	10.73	21
		ANN + MOSOS	0.92	0.93	0.93	0.93	0.83	92.66	90.48	8.44	16
		MARS + MOSOS	0.89	0.91	0.90	0.90	0.78	89.99	88.10	10.96	23
Hoang and Bui (2018)	256:364 (0.70)	KFDA-LSSVM	0.85	0.82	0.81	0.82	0.69	81.00	88.00	15.60	30
Hoang and Bui (2018)	107:78 (1.37)	KFDA-LSSVM	0.88	0.92	0.86	0.89	0.75	86.00	90.00	12.02	25
Muduli and Das (2015)	88:98 (0.90)	MGGP	0.88	0.84	0.91	0.87	0.76	91.00	85.00	12.05	24
Gandomi *et al.* (2013)	256:364 (0.70)	CHAID	0.84	0.83	0.77	0.80	0.67	77.30	88.70	17.20	31
		E-CHAID	0.82	0.82	0.73	0.78	0.63	73.40	88.70	19.31	37
		CART	0.83	0.84	0.73	0.78	0.65	72.70	90.40	18.93	34
		CHAID-SPT	0.82	0.78	0.78	0.78	0.62	78.10	84.10	18.96	38
		E-CHAID-SPT	0.84	0.83	0.76	0.79	0.66	75.80	89.00	17.86	33
		CART-SPT	0.83	0.79	0.80	0.80	0.65	79.70	85.40	17.50	34
		LR	0.73	0.67	0.70	0.68	0.45	69.90	75.50	27.35	41
Muduli and Das (2013)	115:112 (1.03)	MGGP	0.88	0.88	0.87	0.88	0.75	87.00	88.00	12.50	26
Oommen *et al.* (2010)	109:87 (1.25)	SVM	0.85	0.84	0.90	0.87	0.70	89.90	79.30	15.57	27
Seed and Idriss (1971)	256:364 (0.70)	Empirical Method	0.67	0.56	0.95	0.70	0.45	94.92	46.98	33.22	32

*Based on testing dataset only; *[1]Dataset 1; [2]Dataset 2; Legend: see the List of Abbreviations and List of Symbol

The proposed models are better than the existing AI-based models of Ghanizadeh *et al.* (2023) (WNN-PSO with a $G_{mean(error)}$ of 5.94%), CCF, Rotfor, and RF models of Demir and Sahin (2022) with a model ranking of 26, 28, and 25 respectively. ANN/MARS+NSGA-II ($G_{mean(error)}$ of 5.94% and 7.81%), ANN/ MARS+MOSOS ($G_{mean(error)}$ of 4.45% and 7.81%) (Das *et al.* 2020), MGGP models (Model 1 and 2) ($G_{mean(error)}$ of 17.60% and 17.87%) (Muduli and

Table 6 Fold-wise classification performance of the developed ML models based on the CPT data

Model	Overall							
	ACC or OA	**PRE**	**REC**	**F_1**	**MCC**	**A_L (%)**	**A_{NL} (%)**	***$G_{mean(error)}$ (%)***
SDT: Fold 1	0.91	0.87	0.99	0.93	0.82	99.25	78.49	11.74
SDT: Fold 2	0.91	0.90	0.85	0.87	0.80	84.62	94.62	10.52
SDT: Fold 3	0.92	0.94	0.92	0.93	0.83	91.73	91.40	8.44
SDT: Fold 4	0.94	0.93	0.98	0.95	0.88	97.74	89.25	6.60
SDT: Fold 5	0.95	0.97	0.95	0.96	0.90	94.74	95.70	4.78
SDT: Average	0.93	0.92	0.94	0.93	0.85	93.61	89.89	8.42
RF: Fold 1	0.97	0.96	0.99	0.97	0.94	99.25	93.55	3.64
RF: Fold 2	0.97	0.99	0.96	0.98	0.95	96.24	98.92	2.43
RF: Fold 3	0.99	0.99	0.98	0.99	0.97	98.45	98.92	1.31
RF: Fold 4	0.99	0.99	0.98	0.99	0.97	98.50	98.92	1.29
RF: Fold 5	0.99	0.99	0.99	0.99	0.98	99.25	98.92	0.91
RF: Average	0.98	0.99	0.98	0.98	0.96	98.34	97.85	1.92
GB: Fold 1	0.97	0.95	1.00	0.97	0.94	100.00	92.47	3.84
GB: Fold 2	0.99	0.99	0.99	0.99	0.98	99.25	98.92	0.91
GB: Fold 3	0.98	0.99	0.98	0.98	0.96	97.67	98.92	1.70
GB: Fold 4	0.99	0.99	0.99	0.99	0.98	99.25	98.92	0.91
GB: Fold 5	0.99	0.99	0.99	0.99	0.97	99.25	97.85	1.45
GB: Average	0.98	0.98	0.99	0.99	0.97	99.08	97.42	1.76
Bagging: Fold 1	0.97	0.97	0.98	0.98	0.95	98.50	95.70	2.91
Bagging: Fold 2	0.99	1.00	0.98	0.99	0.97	97.74	100.00	1.13
Bagging: Fold 3	0.98	0.99	0.98	0.98	0.96	97.74	98.92	1.67
Bagging: Fold 4	1.00	1.00	0.99	1.00	0.99	99.25	100.00	0.38
Bagging: Fold 5	1.00	0.99	1.00	1.00	0.99	100.00	98.92	0.54
Bagging: Average	0.99	0.99	0.99	0.99	0.97	98.65	98.71	1.32
AdaBoost: Fold 1	0.97	0.96	0.98	0.97	0.94	98.50	94.62	3.46
AdaBoost: Fold 2	0.97	0.97	0.98	0.97	0.94	97.74	95.70	3.28
AdaBoost: Fold 3	0.98	0.98	0.98	0.98	0.95	98.50	96.77	2.37
AdaBoost: Fold 4	0.99	0.99	0.98	0.99	0.97	98.50	98.92	1.29
AdaBoost: Fold 5	0.97	0.98	0.98	0.98	0.95	97.74	96.77	2.74
AdaBoost: Average	0.98	0.98	0.98	0.98	0.95	98.20	96.56	2.63
Voting: Fold 1	0.91	0.87	0.99	0.93	0.82	99.25	79.57	11.13
Voting: Fold 2	0.92	0.91	0.97	0.94	0.85	96.99	86.02	8.66
Voting: Fold 3	0.92	0.90	0.96	0.93	0.83	96.24	84.95	9.58
Voting: Fold 4	0.93	0.90	0.98	0.94	0.86	98.50	84.95	8.53
Voting: Fold 5	0.91	0.88	0.97	0.92	0.81	96.99	81.72	10.97
Voting: Average	0.92	0.89	0.98	0.93	0.83	97.59	83.44	9.76
Stacking: Fold 1	0.94	0.93	0.97	0.95	0.88	96.99	90.32	6.40
Stacking: Fold 2	0.96	0.97	0.97	0.97	0.93	96.99	95.70	3.66
Stacking: Fold 3	0.95	0.95	0.95	0.95	0.89	95.49	93.55	5.49
Stacking: Fold 4	0.96	0.97	0.97	0.97	0.93	96.99	95.70	3.66
Stacking: Fold 5	0.96	0.96	0.97	0.96	0.91	96.99	93.55	4.75
Stacking: Average	0.95	0.96	0.97	0.96	0.91	96.69	93.76	4.78
NB: Fold 1	0.82	0.79	0.94	0.86	0.63	93.98	64.52	22.13
NB: Fold 2	0.80	0.78	0.92	0.84	0.58	91.73	62.37	24.36
NB: Fold 3	0.81	0.79	0.94	0.86	0.62	93.98	63.44	22.78
NB: Fold 4	0.80	0.77	0.94	0.85	0.59	93.98	60.22	24.77
NB: Fold 5	0.79	0.77	0.92	0.84	0.57	92.48	60.22	25.38
NB: Average	0.80	0.78	0.93	0.85	0.60	93.23	62.15	23.88
K-NN: Fold 1	0.77	0.79	0.83	0.81	0.51	82.71	67.74	25.15
K-NN: Fold 2	0.81	0.83	0.85	0.84	0.61	84.96	75.27	20.03
K-NN: Fold 3	0.77	0.79	0.81	0.80	0.51	81.20	69.89	24.66
K-NN: Fold 4	0.76	0.77	0.84	0.80	0.49	84.21	63.44	26.91
K-NN: Fold 5	0.77	0.79	0.83	0.81	0.52	83.46	67.74	24.81
K-NN: Average	0.77	0.79	0.83	0.81	0.53	83.31	68.82	24.28

Table 6 Continued-

LR: Fold 1	0.86	0.83	0.95	0.89	0.71	94.74	73.12	16.77
LR: Fold 2	0.87	0.89	0.89	0.89	0.74	88.72	84.95	13.19
LR: Fold 3	0.87	0.88	0.91	0.89	0.73	90.98	81.72	13.78
LR: Fold 4	0.87	0.88	0.90	0.89	0.72	90.23	81.72	14.13
LR: Fold 5	0.86	0.86	0.90	0.88	0.71	90.23	79.57	15.27
LR: Average	0.87	0.87	0.91	0.89	0.72	90.98	80.22	14.57
SV: Fold 1	0.71	0.69	0.94	0.79	0.41	93.98	38.71	39.68
SV: Fold 2	0.69	0.70	0.85	0.77	0.35	84.96	47.31	36.60
SV: Fold 3	0.70	0.69	0.89	0.78	0.37	88.72	44.09	37.46
SV: Fold 4	0.70	0.70	0.88	0.78	0.37	87.97	45.16	36.97
SV: Fold 5	0.70	0.70	0.87	0.78	0.37	87.22	46.24	36.50
SV: Average	0.70	0.69	0.89	0.78	0.38	88.57	44.30	37.36
XGBoost: Fold 1	0.98	0.96	1.00	0.98	0.95	100.00	94.62	2.73
XGBoost: Fold 2	0.97	0.98	0.97	0.98	0.95	96.99	97.85	2.58
XGBoost: Fold 3	0.98	0.99	0.98	0.98	0.96	97.74	98.92	1.67
XGBoost: Fold 4	0.99	0.99	0.98	0.99	0.97	98.50	98.92	1.29
XGBoost: Fold 5	0.99	0.99	0.99	0.99	0.97	99.25	97.85	1.45
XGBoost: Average	0.98	0.98	0.98	0.98	0.96	98.50	97.63	1.94
ET: Fold 1	0.98	0.96	1.00	0.98	0.95	100.00	94.62	2.73
ET: Fold 2	0.97	0.98	0.97	0.97	0.94	96.99	96.77	3.12
ET: Fold 3	0.98	0.97	1.00	0.99	0.96	100.00	95.70	2.17
ET: Fold 4	1.00	1.00	0.99	1.00	0.99	99.25	100.00	0.38
ET: Fold 5	0.96	0.96	0.98	0.97	0.93	98.50	93.55	4.01
ET: Average	0.98	0.97	0.99	0.98	0.95	98.95	96.13	2.47
HGB: Fold 1	0.97	0.96	0.99	0.98	0.95	99.25	94.62	3.09
HGB: Fold 2	0.97	0.98	0.97	0.97	0.94	96.99	96.77	3.12
HGB: Fold 3	0.97	0.98	0.98	0.98	0.95	97.74	96.77	2.74
HGB: Fold 4	0.98	0.98	0.98	0.98	0.95	98.50	96.77	2.37
HGB: Fold 5	0.96	0.97	0.96	0.97	0.92	96.24	95.70	4.03
HGB: Average	0.97	0.97	0.98	0.98	0.94	97.74	96.13	3.07
LightGBM: Fold 1	0.97	0.96	0.99	0.97	0.94	99.25	93.55	3.64
LightGBM: Fold 2	0.98	0.99	0.97	0.98	0.95	96.99	98.92	2.05
LightGBM: Fold 3	0.98	0.99	0.98	0.98	0.96	97.74	98.92	1.67
LightGBM: Fold 4	0.99	0.99	0.98	0.99	0.97	98.50	98.92	1.29
LightGBM: Fold 5	0.98	0.98	0.98	0.98	0.96	98.50	97.85	1.83
LightGBM: Average	0.98	0.98	0.98	0.98	0.96	98.20	97.63	2.09

Legend: see the List of Abbreviations and List of Symbols

Das 2013), LR+MARS models of Zhang and Goh (2016) ($G_{mean(error)}$ of 6.74%, 9.17%, and 8.04% for three models developed on three different datasets). The Bagging model not only shows good accuracy in the classification of majority class instances (A_{NL} = 98.71%) but also performs very well in the classification of minority class instances (A_L = 98.65%). The close values of these parameters indicate lesser overfitting in the model. The AUROCs for CPT-based ML models, as given in Fig. 8, also support the superiority of Bagging, GB, XGBoost, an RF models. These ML-based models are also compared with the conventional liquefaction potential assessment model suggested by Robertson and Wride (1998) as shown in Table 7. This empirical model attains an overall rank of 27 with a $G_{mean(error)}$ of 24.66%. Previously published literature has discussed the importance of different parameters in the classification operation of liquefaction instances (Bhowan *et al.* 2010; Das *et al.* 2020; Hoang and Bui 2018; Kayen *et al.* 2013; Muduli and Das 2013; Zhang and Goh 2016; Zhou *et al.* 2021). It's worth mentioning here that the developed models use the following 11 features as input parameters: M_w, d, d_w, σ_v, σ'_v, a_{max}/g, r_d, CSR, R_f, q_c, and f_s. It can be observed from the results that the model with better $G_{mean(error)}$ and MCC values also attained a better ranking in Table 7, which assures the suitability of these performance metrics for the biased dataset.

Fig. 8 provides the AUROC curve of different machine learning models trained on the CPT dataset. From Fig. 8 it can be ensured that algorithms such as Bagging, RF, XGBoost, stacking, etc. show better performance in terms of AUROC. The GB and AdaBoost perform best in terms of AUROC with a value of 0.989, followed by Bagging (0.988), but these models fail to attain a higher ranking based on other performance metrics. Comparing the model performances from Table 7 and the AUC represents the ambiguity in model ranking based on different performance metrics. This in turn raises the question regarding the choice of the correct performance indicator. Das *et al.* (2020) also discussed the problem of having different performance metrics. AUROC seldom can be misleading as these curves

Table 7 The classification performance comparison of present models with previous studies for the CPT data

References	*L:NL (L/NL)*	Model	Overall								
			ACC or OA	PRE	REC	F_1	MCC	A_L (%)	A_{NL} (%)	$G_{mean(error)}$ (%)	Ranking
Present Study	133:93 (1.43)	SDT	0.93	0.92	0.94	0.93	0.85	93.61	89.89	8.42	16
		RF	0.98	0.99	0.98	0.98	0.96	98.34	97.85	1.92	3
		GB	0.98	0.98	0.99	0.99	0.97	99.08	97.42	1.76	2
		Bagging	0.99	0.99	0.99	0.99	0.97	98.65	98.71	1.32	1
		AdaBoost	0.98	0.98	0.98	0.98	0.95	98.20	96.56	2.63	8
		Voting	0.92	0.89	0.98	0.93	0.83	97.59	83.44	9.76	15
		Stacking	0.95	0.96	0.97	0.96	0.91	96.69	93.76	4.78	11
		NB	0.80	0.78	0.93	0.85	0.60	93.23	62.15	23.88	29
		K-NN	0.77	0.79	0.83	0.81	0.53	83.31	68.82	24.28	30
		LR	0.87	0.87	0.91	0.89	0.72	90.98	80.22	14.57	24
		SV	0.70	0.69	0.89	0.78	0.38	88.57	44.30	37.36	31
		XGBoost	0.98	0.98	0.98	0.98	0.96	98.50	97.63	1.94	4
		ET	0.98	0.97	0.99	0.98	0.95	98.95	96.13	2.47	7
		HGB	0.97	0.97	0.98	0.98	0.94	97.74	96.13	3.07	9
		LightGBM	0.98	0.98	0.98	0.98	0.96	98.20	97.63	2.09	6
Ghanizadeh *et al.* (2023)	74:35 (2.11)	WNN-PSO	0.99	1.00	0.99	0.99	0.98	90.58	97.67	5.94	5
Demir and Sahin (2022)	133:93 (1.43)	CCF	0.84	0.81	0.95	0.88	0.67	94.74	68.50	19.44	26
		RotFor	0.81	0.76	0.97	0.85	0.61	96.99	56.46	26.00	28
		RF	0.83	0.79	0.97	0.87	0.66	96.99	63.45	21.55	25
Das *et al.* (2020)	138:43 (3.21)	ANN + NSGA-II	0.92	0.99	0.91	0.95	0.82	90.58	97.67	5.94	12
		MARS + NSGA-II	0.92	0.96	0.89	0.92	0.84	89.13	95.35	7.81	18
		ANN + MOSOS	0.95	0.98	0.94	0.96	0.91	93.48	97.67	4.45	10
		MARS + MOSOS	0.92	0.96	0.89	0.92	0.84	89.13	95.35	7.81	18
Hoang and Pham (2016)	133:93 (1.43)	KFDA-LSSVM	0.93	0.94	0.94	0.94	0.85	94.00	91.00	7.51	13
Zhang and Goh (2016)	133:93 (1.43)	LR+MARS	0.92	0.98	0.90	0.94	0.85	90.28	96.34	6.74	14
	104:66 (1.57)	LR+MARS	0.91	0.95	0.90	0.93	0.80	90.00	91.67	9.17	21
	250:216 (1.16)	LR+MARS	0.92	0.91	0.94	0.92	0.84	94.19	89.78	8.04	17
Muduli and Das (2013)	58:38 (1.53)	MGGP (model 1)	0.86	0.84	0.97	0.90	0.72	97.00	70.00	17.60	21
		MGGP (model 2)	0.85	0.83	0.95	0.89	0.70	95.00	71.00	17.87	23
Oommen *et al.* (2010)	139:43 (3.23)	SVM	0.89	0.89	0.98	0.93	0.68	97.80	60.40	23.14	20
Robertson and Wride (1998)	133:93 (1.43)	Empirical Method	0.81	0.77	0.98	0.86	0.63	97.74	58.06	24.66	27

Legend: see the List of Abbreviations and List of Symbols

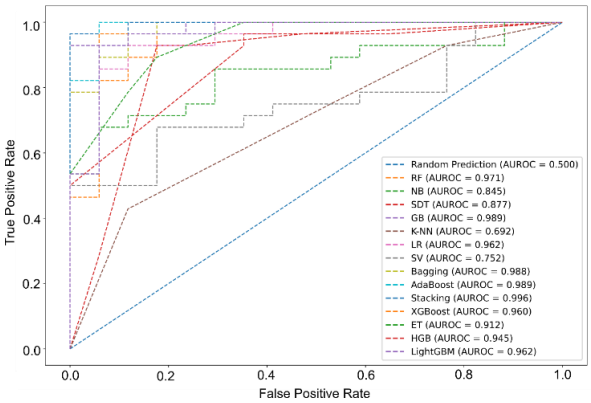


Fig. 8 ROC for different classification models developed using the CPT dataset

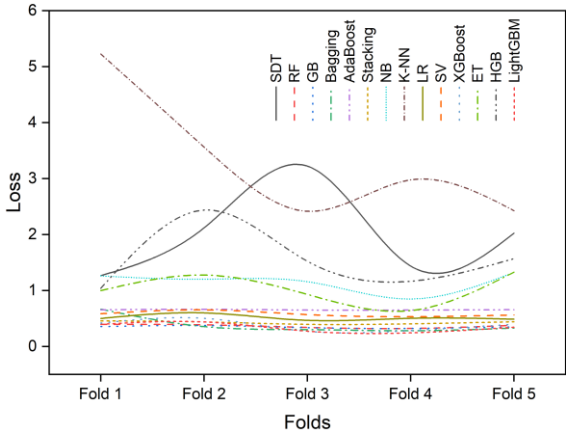


Fig. 9 Loss values per fold of the classification models developed using the CPT dataset

are based on an assumed threshold value and are specific to a particular project. They also require more computation power as they are calculated again and again during the model development process. These issues bring out the need to adopt a ranking system that can include the results obtained from the different performance metrics.

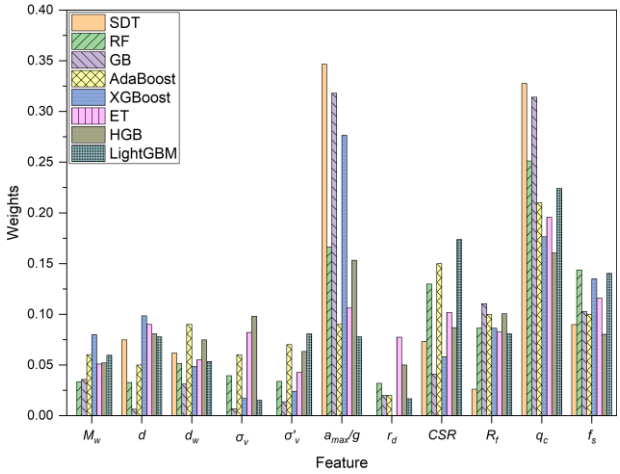


Fig. 10 Graph showing feature importance in terms of feature weights for different classification models developed for the CPT dataset

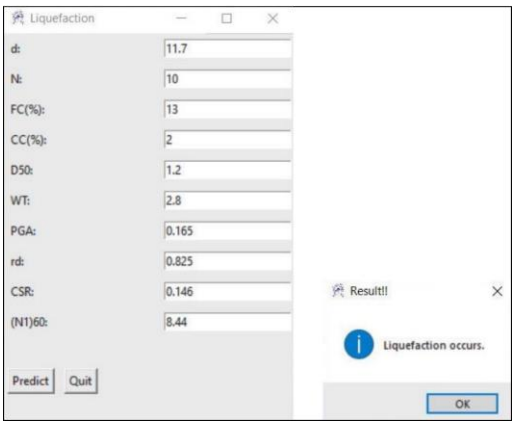


Fig. 11 Screenshot of the graphical user interfaces for liquefaction classification prediction software for SPT and V_s-based RF model

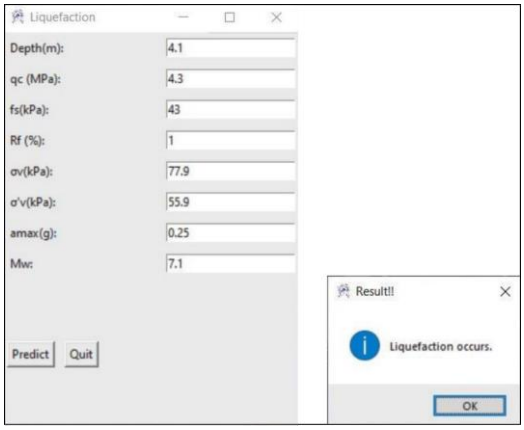


Fig. 12 Screenshot of the graphical user interfaces for liquefaction classification prediction software for CPT-based RF model

The loss values of each fold of the developed models are presented in Fig. 9 to check the consistency of the model convergence. These results indicate that mainly K-NN, SDT, HGB, and ET have abrupt variations in loss values in each fold, this indicates poor convergence of the model. These variations also reflected in the poor model performance.

In Fig. 10, a graph showing the comparative weights of different input features in different ensemble machine-learning algorithms has been shown. The higher weight of a parameter represents the higher importance of the parameter in the model-building process. From Fig. 10 it can be inferred that GB assigns maximum weightage to a_{max}/g, and q_c followed by R_f, and f_s. Based on the RF algorithm q_c can be considered the most important feature followed by a_{max}/g, f_s, and CSR. Similarly, XGBoost gives more weightage to a_{max}/g followed by q_c, f_s, and respectively. This trend indicates that the best-developed models are more sensitive to model parameters a_{max}/g, q_c, and f_s. Although, Muduli and Das (2013) identified CSR, q_c, σ_v as the most influential parameters for the first MGGP model and q_c, a_{max}/g, and σ_v for another MGGP model. Zhang and Goh (2016) reported a_{max}/g *and* q_c as the most important parameters based on sensitivity analysis. Das *et al*. (2020) identified q_c, a_{max}/g, and CSR as the most crucial parameters based on the multi-objective feature selection (MOFS) algorithm. Although there are few variations between the models while choosing the most crucial parameters, this can be explained by the variability in the dataset and algorithm parameters chosen by the researcher during the model development process. The information provided by the sensitivity analysis of parameters can sometimes be an important guideline while choosing the inputs for the analysis.

After closely looking at the comparative analysis of results provided in Table 5 and Table 7, it can be concluded that the proposed ensemble algorithms perform better than the other AI-based models available in the literature (Das *et al.* 2020; Ghanizadeh *et al.* 2023; Hoang and Bui 2018; Muduli and Das 2013, 2015; Oommen *et al.* 2010) in terms of the utilized performance metrics. However, the overall efficiency of the model is found to be very much dependent on the type of the dataset and consideration of input parameters. Apart from developing the model a GUI is also developed that utilizes the classification model of RF to make predictions about the liquefiable or non-liquefiable nature of a particular case. This GUI-enabled software is trained based on the two datasets utilized in this study. The software requires all the input parameters incorporated in the model development process. The screenshot of this GUI is presented in Figs. 11 and 12 for the SPT dataset and CPT dataset respectively. This GUI is developed as executable software that can be run on any local system having a Python version greater than 3.9.0. The user will need to supply the numerical value of the required parameters; based on the type of in-situ test and the software will automatically display the susceptibility of that soil toward liquefaction.

5.3 Validation of models

The developed ML models were validated on an unforeseen dataset of SPT with V_s measurement and CPT dataset. The results are presented in Tables 8 and 9

Table 8 Validation results for the developed models based on unforeseen SPT and V_s dataset

Model	Overall							
	ACC or OA	PRE	REC	F_1	MCC	A_L (%)	A_{NL} (%)	$G_{mean(error)}$ (%)
SDT	0.81	0.83	0.84	0.83	0.61	84.15	76.61	19.71
RF	0.94	0.93	0.97	0.95	0.88	96.95	90.32	6.42
GB	0.83	0.85	0.85	0.85	0.65	85.37	79.84	17.44
Bagging	0.92	0.94	0.91	0.93	0.84	91.46	92.74	7.90
AdaBoost	0.83	0.86	0.84	0.85	0.65	84.15	81.45	17.21
Voting	0.80	0.83	0.80	0.82	0.58	80.49	78.23	20.65
Stacking	0.89	0.90	0.91	0.90	0.77	90.85	86.29	11.46
NB	0.77	0.80	0.80	0.80	0.53	79.88	73.39	23.44
K-NN	0.78	0.81	0.79	0.80	0.55	79.27	75.81	22.48
LR	0.78	0.81	0.82	0.81	0.56	81.71	74.19	22.14
SV	0.77	0.81	0.79	0.80	0.53	78.66	75.00	23.19
XGBoost	0.97	0.96	0.98	0.97	0.93	97.56	95.16	3.65
ET	0.93	0.92	0.96	0.94	0.86	95.73	89.52	7.43
HGB	0.91	0.92	0.92	0.92	0.81	92.07	88.71	9.62
LightGBM	0.95	0.96	0.95	0.95	0.89	95.12	94.35	5.26

Legend: see the List of Abbreviations and List of Symbols

Table 9 Validation results for the developed models based on unforeseen CPT dataset

Model	Overall							
	ACC or OA	PRE	REC	F_1	MCC	A_L (%)	A_{NL} (%)	$G_{mean(error)}$ (%)
SDT	0.84	0.62	0.93	0.74	0.66	92.86	80.49	13.55
RF	0.95	0.87	0.93	0.90	0.86	92.86	95.12	6.02
GB	0.95	0.87	0.93	0.90	0.86	92.86	95.12	6.02
Bagging	0.98	0.93	1.00	0.97	0.95	100.00	97.56	1.23
AdaBoost	0.95	0.92	0.86	0.89	0.85	85.71	97.56	8.55
Voting	0.82	0.61	0.79	0.69	0.57	78.57	82.93	19.28
Stacking	0.85	0.67	0.86	0.75	0.66	85.71	85.37	14.46
NB	0.82	0.60	0.86	0.71	0.60	85.71	80.49	16.94
K-NN	0.82	0.61	0.79	0.69	0.57	78.57	82.93	19.28
LR	0.78	0.55	0.86	0.67	0.55	85.71	75.61	19.50
SV	0.78	0.55	0.79	0.65	0.51	78.57	78.05	21.69
XGBoost	0.93	0.81	0.93	0.87	0.82	92.86	92.68	7.23
ET	0.89	0.75	0.86	0.80	0.73	85.71	90.24	12.05
HGB	0.89	0.72	0.93	0.81	0.75	92.86	87.80	9.70
LightGBM	0.91	0.76	0.93	0.84	0.78	92.86	90.24	8.46

Legend: see the List of Abbreviations and List of Symbols

respectively. The developed models are tested on an SPT dataset of the records from the Chi-Chi earthquake 1999 (1999 Jiji earthquake) (Hwang and Yang 2001). This dataset contains data from 164 liquefaction records and 124 non-liquefaction cases totaling 288 sets of records. The data was collected from several boreholes from different locations in the city. Due to the limited availability of the input parameters required for the developed models, few parameters of the Chi-Chi earthquake dataset were derived from the primarily available parameters. Similarly, the developed CPT-based ensemble models are tested on an unforeseen dataset of 55 case histories of CPT from the Chi-Chi earthquake 1999 (1999 Jiji earthquake) (Ku *et al.* 2004). The test results from hole nos. LW-A1, LW-A2, LW-A3, LW-A5, LW-A7, LW-A9, LW-A10, LW-C1, LW-C2, LW-D1, and LW-D3 are used for the validation. This data contains 14 liquefied cases and 41 non-liquefied reported cases from the site.

The validation results from Table 8 and Table 9 indicate that the developed SPT and V_s-based models as well as CPT-based models perform satisfactorily for the unforeseen datasets. For SPT and V_s-based models, the algorithm XGBoost performs the best with a $G_{mean(error)}$ of 3.65%. The next best performing algorithms are LightGBM and RF ($G_{mean(error)}$ of 5.26% and 6.42% respectively) as shown in Table 8. This sequence also follows the ranking trend obtained from the model development. Similarly, for the CPT-based model, the validation models are presented in

Table 9. The validation results show that the ranked 1 Bagging model from the model development stage also provides the best classification results on an unforeseen CPT dataset. It is followed by RF and GB with a $G_{mean(error)}$ of 6.02%. This validation performance is consistent with the performance of the model-building phase.

6. Conclusions

This study is conducted to assess the comparative performance of different ensemble ML algorithms in the task of classifying the liquefiable and non-liquefiable class instances based on a biased dataset. After performing a detailed analysis of the datasets following contributions can be made based on this study:

- The ensemble algorithms based on boosting (XGBoost ($G_{mean(error)}$ = 2.79%) and LightGBM ($G_{mean(error)}$ = 2.79%)) are preferable to classify liquefaction instances for available SPT database with V_s measurement based on an unbalanced dataset. Instead of using only SPT data, it is preferable to include V_s measurements also as it significantly improves the prediction performance
- For the classification of liquefaction instances based on CPT data, Bagging ($G_{mean(error)}$ = 1.32%), GB ($G_{mean(error)}$ = 1.76%), RF ($G_{mean(error)}$ = 1.92%), and XGBoost ($G_{mean(error)}$ = 1.94%) models should be preferred as they show better performances and efficacy for the biased database.
- While dealing with a biased dataset it is preferred to use ACC, PRE, REC, MCC, F_1 scores, A_L, $A_{NL,}$ and $G_{mean(error)}$ as performance metrics. These metrics offer better insight into the model efficacy for a dataset with majority and minority class instances.
- SPT identifies ϕ, F_{75}, *CSR, and* $(N_1)_{60}$ as the most influential parameters, while CPT considers a_{max}/g, q_c, f_s, and CSR as the most important inputs. Although the order of features' importance varies significantly from literature to literature, the results obtained from the current study show reasonable agreement with physical phenomena and previous works. The choice of input is an important consideration while developing an AI or ML-based model, and the proposed algorithms can surely provide some guidelines for feature selection.
- To better assist the professional engineers in the quick estimation of the liquefaction susceptibility without going through the rigorous methods of machine learning a user-friendly Python-based software with a GUI has been developed which utilizes the best RF model developed in this study.
- The performances of the models are found to depend upon the fine-tuning of the algorithm hyper parameters, which is achieved by an exhaustive grid search technique. This is mostly influenced by the dataset utilized in the process of development. Hence, a more sophisticated, standardized, and acceptable database should be used to develop such models by different researchers to find the best model for future projects.
- The field has benefited greatly from the insights gained from this dataset, but it is important to recognize that there may be restrictions on how broadly these findings may be applied to other regional contexts or datasets. To achieve a more thorough grasp of the phenomenon being studied, future research attempts should validate and extend these findings across a variety of datasets from different regions.

Acknowledgments

Authors acknowledge the Ministry of Education, Government of India, for the **Prime Minister Research Fellowship and Grant** (PMRF ID: 1601650) for providing necessary funding for this research.

Funding

The necessary fundings for this research have been provided by Prime Minister Research Fellowship and Grant (PMRF ID: 1601650).

References

Abbaszadeh Shahri, A. and Naderi, S. (2016), "Modified correlations to predict the shear wave velocity using piezocone penetration test data and geotechnical parameters: a case study in the southwest of sweden", *Innov. Infrastruct. Solutions*, **1**(1), 1-9. https://doi.org/10.1007/s41062-016-0014-y.

Andrus, R.D. and Stokoe II, K.H. (2000), "Liquefaction resistance of soils from shear-wave velocity", *J. Geotech. Geoenviron. Eng.*, **126**(11), 1015-1025. https://doi.org/10.1061/(ASCE)1090-0241(2000)126:11(1015).

Anitescu, C., Atroshchenko, E., Alajlan, N. and Rabczuk, T. (2019), "Artificial neural network methods for the solution of second order boundary value problems", *Comput. Mater. Continua*, **59**(1), 345-359. https://doi.org/10.32604/cmc.2019.06641.

Atangana Njock, P.G., Shen, S.L., Zhou, A. and Lyu, H.M. (2020), "Evaluation of soil liquefaction using AI technology incorporating a coupled ENN / t-SNE model", *Soil Dyn. Earthq. Eng.*, **130**, 105988. https://doi.org/10.1016/j.soildyn.2019.105988.

Bhowan, U., Zhang, M. and Johnston, M. (2010), "Genetic programming for classification with unbalanced data", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6021 LNCS1–13. https://doi.org/10.1007/978-3-642-12148-7_1.

Breiman, L. (1996), "Bagging predictors", *Mach. Learn.*, **24**(2), 123-140. https://doi.org/10.1007/BF00058655.

Breiman, L. (2001), "Random forests", *Machine Learning*, **45**(1), 5-32. https://doi.org/10.1023/A:1010933404324.

Chen, T. and Guestrin, C. (2016), "XGBoost: A scalable tree boosting system", *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Association for Computing Machinery*, 785-794. https://doi.org/10.1145/2939672.2939785.

Cortes, C. and Vapnik, V. (1995), "Support-vector networks", *Machine Learning 1995 20:3*, **20**(3), 273–97. https://doi.org/10.1007/BF00994018.

Cover, T.M. and Hart, P.E. (1967), "Nearest neighbor pattern

classification", *IEEE Transactions on Information Theory*, **13**(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964.

Dadhich, S., Kumar, J. and Madhav, S. (2023), "Assessment of earthquake-induced liquefaction susceptibility using ensemble learning", *Multiscale Multidiscip. Model. Exper. Design*. https://doi.org/10.1007/s41939-023-00146-z.

Das, B.M. (2014), *Principles of Geotechncial Engineering,* 8th ed. Cengage Learning, India. ISBN: 9788131526132.

Das, S.K., Mohanty, R., Mohanty, M. and Mahamaya, M. (2020), "Multi-Objective feature selection (MOFS) algorithms for prediction of liquefaction susceptibility of soil based on in situ test methods", *Nat. Hazards*, **103**(2), 2371-2793. https://doi.org/10.1007/s11069-020-04089-3.

Davis, J. and Goadrich, M. (2006), "The relationship between precision-recall and ROC curves", *Proceedings of the ACM International Conference Proceeding Series*, **148**(2), 33-40. https://dl.acm.org/doi/10.1145/1143844.1143874.

Demir, S. and Sahin, E.K. (2022), "Comparison of tree-based machine learning algorithms for predicting liquefaction potential using canonical correlation forest, rotation forest, and random forest based on CPT data", *Soil Dyn. Earthq. Eng.*, **154**, 107130. https://doi.org/10.1016/j.soildyn.2021.107130.

Domingos, P. (2012), "A few useful things to know about machine learning", *Communications of the ACM*, **55**(10), 79-88. http://dx.doi.org/10.1145/2347736.2347755.

Duman, E.S., Ikizler, S.B., Angın, Z. and Demir, G. (2014), "Assessment of liquefaction potential of the erzincan, eastern Turkey", *Geomech. Eng.*, **7**(6), 589-612. https://doi.org/10.12989/gae.2014.7.6.589.

Eisavi, V. and Homayouni, S. (2016), "Performance evaluation of random forest and support vector regressions in natural hazard change detection", *J. Appl. Remote Sens.*, **10**(4), 046030. https://doi.org/10.1117/1.JRS.10.046030.

Erzin, Y. and Ecemis, N. (2015), "The use of neural networks for CPT-based liquefaction screening", *Bull. Eng. Geol. Environ.*, **74**(1), 103-116. https://doi.org/10.1007/s10064-014-0606-8.

Eslami, A., Moshfeghi, S., MolaAbasi, H. and Eslami, M.M. (2020), "Soil behavior classification (SBC) using CPT and CPTu records", *Piezocone and Cone Penetration Test (CPTu and CPT) Applications in Foundation Engineering*, 111-144. http://dx.doi.org/10.1016/B978-0-08-102766-0.00005-5.

Freund, Y. and Schapire, R.E. (1997), "A decision-theoretic generalization of on-line learning and an application to boosting", *J. Comput. Syst. Sci.*, **55**(1), 119-139. https://doi.org/10.1006/jcss.1997.1504.

Friedman, J.H. (2001), "Greedy function approximation: a gradient boosting machine.", *Annals of statistics,* **29**(5), 1189-1232. http://dx.doi.org/10.1214/aos/1013203451.

Gandomi, A.H. and Alavi, A.H. (2012), "Krill herd: A new bio-inspired optimization algorithm", *Communications in Nonlinear Science and Numerical Simulation*, **17**(12), 4831-4845. https://doi.org/10.1016/j.cnsns.2012.05.010.

Gandomi, A.H., Fridline, M.M. and Roke, D.A. (2013), "Decision tree approach for soil liquefaction assessment", *The Scientific World J.*, https://doi.org/10.1155/2013/346285.

Geurts, P., Ernst, D. and Wehenkel, L. (2006), "Extremely randomized trees", *Machine Learning*, **63**(1), 3-42. https://doi.org/10.1007/s10994-006-6226-1.

Ghanizadeh, A.R., Aziminejad, A., Asteris, P.G. and Armaghani, D.J. (2023), "Soft computing to predict earthquake-induced soil liquefaction via CPT results", *Infrastructures*, **8**(8) 125. https://doi.org/10.3390/infrastructures8080125.

Goh, A.T.C. (1994), "Seismic liquefaction potential assessed by neural networks", *J. Geotech. Eng.*, **120**(9), 1467-1480. https://doi.org/10.1061/(ASCE)0733-9410(1994)120:9(1467).

Goh, A.T.C. and Goh, S.H. (2007), "Support vector machines: their use in geotechnical engineering as illustrated using seismic liquefaction data", *Comput. Geotech.*, **34**(5), 410-421. https://doi.org/10.1016/j.compgeo.2007.06.001.

Guo, H., Zhuang, X., Chen, P., Alajlan, N. and Rabczuk, T. (2022a), "Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media", *Eng. Comput.*, **38**(6), 5173-5198. https://doi.org/10.1007/s00366-021-01586-2.

Guo, H., Rabczuk, T., Zhu, Y., Cui, H., Su, C. and Zhuang, X. (2022b), "Soil liquefaction assessment by using hierarchical gaussian process model with integrated feature and instance based domain adaption for multiple data sources", *AI Civ. Eng.*, **1**(1), 1-32. https://link.springer.com/article/10.1007/s43503-022-00004-w.

Gupta, T., Ramana, G.V. and Elgamal, A. (2023), "A hybrid numerical–probabilistic approach for machine learning-based prediction of liquefaction-induced settlement using CPT data", *Arabian J. Geosci.*, **16**(6), 1-16. https://doi.org/10.1007/s12517-023-11500-3.

Hanna, A.M., Ural, D. and Saygili, G. (2007), "Neural network model for liquefaction potential in soil deposits using Turkey and Taiwan earthquake data", *Soil Dyn. Earthq. Eng.*, **27**(6), 521-540. https://doi.org/10.1016/j.soildyn.2006.11.001.

Hastie, T., Tibshirani, R. and Friedman, J. (2009), "The elements of statistical learning: data mining, inference, and prediction", *Springer*, **2**, New York. https://doi.org/10.1007/978-0-387-84858-7

Hoang, N.D. and Bui, D.T. (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**(1), 191-204. https://doi.org/10.1007/s10064-016-0924-0.

Hwang, J.H. and Yang, C.W. (2001), "Verification of critical cyclic strength curve by Taiwan Chi-Chi earthquake data", *Soil Dyn. Earthq. Eng.*, **21**(3), 237-257. https://doi.org/10.1016/S0267-7261(01)00002-1.

Idriss, I.M. and Boulanger, R.W. (2006), "Semi-empirical procedures for evaluating liquefaction potential during earthquakes", *Soil Dyn. Earthq. Eng.*, **26**(2-4), 115-130. https://doi.org/10.1016/j.soildyn.2004.11.023.

Iwasaki, T., Arakawa, T. and Tokida, K.I. (1984), "Simplified procedures for assessing soil liquefaction during earthquakes", *Soil Dyn. Earthq. Eng.*, **3**(1), 49-58. https://doi.org/10.1016/0261-7277(84)90027-5.

Jiao, W., Hao, X. and Qin, C. (2021), "The image classification method with CNN-XGBoost model based on adaptive particle swarm optimization", *Information*, **12**(4), 156. https://doi.org/10.3390/info12040156.

Juang, C.H., Chen, C.J., Tang, W.H. and Rosowsky, D.V. (2000), "CPT-based liquefaction analysis, part 1: determination of limit state function", *Geotechnique*, **50**(5), 583-592. https://doi.org/10.1680/geot.2000.50.5.583.

Juang, C.H., Yuan, H., Li, D.K., Yang, S.H. and Christopher, R.A. (2005), "Estimating severity of liquefaction-induced damage near foundation", *Soil Dyn. Earthq. Eng.*, **25**(5), 403-411. https://doi.org/10.1016/j.soildyn.2004.11.001.

Suryadi, M.K., Herteno, R., Saputro, S.W., Faisal, M.R. and Nugroho, R.A. (2024), "Comparative study of various hyperparameter tuning on random forest classification with SMOTE and feature selection using genetic algorithm in software defect prediction", *J. Electron. Electromedical Eng. Medical Inform.*, **6**(2), 137-147. https://doi.org/10.35882/jeeemi.v6i2.375.

Kayen, R., Moss, R.E.S., Thompson, E.M., Seed, R.B., Cetin, K.O., Kiureghian, A. Der, Tanaka, Y. and Tokimatsu, K. (2013), "Shear-Wave velocity–based probabilistic and deterministic assessment of seismic soil liquefaction potential", *J. Geotech. Geoenviron. Eng.*, **139**(3), 407-419.

https://doi.org/10.1061/(ASCE)GT.1943-5606.0000743.
Kotsiantis, S.B., Zaharakis, I.D. and Pintelas, P.E. (2006), "Machine learning: A review of classification and combining techniques", *Artif. Intell. Review*, **26**(3), 159-190. https://doi.org/10.1007/s10462-007-9052-3.
Kramer, S.L. (1996), *Geotechnical Earthquake Engineering,* Prentice Hall, New Jersey, USA.
Ku, C.S., Lee, D.H. and Wu, J.H. (2004), "Evaluation of soil liquefaction in the Chi-Chi, Taiwan earthquake using CPT", *Soil Dyn. Earthq. Eng.*, **24**(9-10), 659-673. https://doi.org/10.1016/j.soildyn.2004.06.009.
Kubat, M., Kubat, M. and Matwin, S. (1997), "Addressing the curse of imbalanced training sets: One-sided selection", *Proceedings of the 14th International Conference on Machine Learning*. https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.43.4487.
Kulhawy, F.H. and Mayne, P.W. (1990), "Manual on estimating soil properties for foundation design", *Ostigov*, 299. http://www.osti.gov/energycitations/product.biblio.jsp?osti_id=6653074.
Kuncheva, L.I. (2004), "Combining pattern classifiers: methods and algorithms", *John wiley & sons, Inc. Publication*, Hoboken. https://doi.org/10.1002/0471660264.
Kurnaz, T.F., Erden, C., Kökçam, A.H., Dağdeviren, U. and Demir, A.S. (2023), "A hyper parameterized artificial neural network approach for prediction of the factor of safety against liquefaction", *Eng. Geol.*, **319**, 107-109. https://doi.org/10.1016/j.enggeo.2023.107109.
Laghmati, S., Hamida, S., Hicham, K., Cherradi, B. and Tmiri, A. (2024), "An improved breast cancer disease prediction system using ML and PCA", *Multimedia Tools Appl.*, **83**(11), 33785-821. https://link.springer.com/article/10.1007/s11042-023-16874-w.
Le, T.T.H., Shin, Y., Kim, M. and Kim, H. (2024), "Towards unbalanced multiclass intrusion detection with hybrid sampling methods and ensemble classification", *Appl. Soft Comput.*, **157**, 111-517. https://doi.org/10.1016/j.asoc.2024.111517.
Liaw, A. and Wiener M. (2002), "Classification and regression by random forest", *Open J. Stat.*, **4**(7).
Machado, M.R., Karray, S. and De Sousa, I.T. (2019), "LightGBM: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry", *Proceedings of the 14th International Conference on Computer Science and Education, ICCSE 2019*, (Nips). https://doi.org/10.1109/ICCSE.2019.8845529
Mishra, P.N., Suman, S. and Das, S.K. (2017), "Experimental investigation and prediction models for thermal conductivity of biomodified buffer materials for hazardous waste disposal", *J. Hazardous, Toxic, and Radioactive Waste*, **21**(2), 1-13. https://doi.org/10.1061/(ASCE)HZ.2153-5515.0000327
Moss, R.E., Seed, R.B., Kayen, R.E., Stewart, J.P., Der Kiureghian, A. and Cetin, K.O. (2006), "CPT-based probabilistic and deterministic assessment of in situ seismic soil liquefaction potential", *J. Geotech. Geoenviron. Eng.*, **132**(8), 1032-1051. https://doi.org/10.1061/(ASCE)1090-0241(2006)132:8(1032).
Muduli, P.K. and Das, S.K. (2013), "SPT-based probabilistic method for evaluation of liquefaction potential of soil using multi-gene genetic programming", *Int. J. Geotech. Earthq. Eng.*, **4**(1), 42-60. https://doi.org/10.4018/jgee.2013010103.
Muduli, P.K. and Das, S.K. (2015), "Model uncertainty of SPT-based method for evaluation of seismic soil liquefaction potential using multi-gene genetic programming", *Soils Found.*, **55**(2), 258-275. https://doi.org/10.1016/j.sandf.2015.02.003.
Naser, M.Z. and Alavi, A.H. (2021), "Error metrics and performance fitness indicators for artificial intelligence and machine learning in engineering and sciences", *Architecture, Struct. Constr.*, 1-19. https://doi.org/10.1007/s44150-021-00015-8.
Oommen, T., Baise, L.G. and Vogel, R. (2010a), "Validation and application of empirical liquefaction models", *J. Geotech. Geoenviron. Eng.*, **136**(12), 1618-1633. https://doi.org/10.1007/s44150-021-00015-8/10.1061/ASCEGT.1943-5606.0000395.
Ozsagir, M., Erden, C., Bol, E., Sert, S. and 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.
Pal, M. (2006), "Support vector machines-based modelling of seismic liquefaction potential", *Int. J. Numer. Anal. Method. Geomech.*, **30**(10), 983-996. https://doi.org/10.1002/nag.509.
Powers, D.M.W. (2011), "Evaluation: from precision, recall and F-measure to ROC, Informedness, Markedness and Correlation", *J. Mach. Learn. Tech.*, **2**(1), 37-63. https://doi.org/10.48550/arXiv.2010.16061.
Robertson, P.K. and Wride, C. (1998), "Evaluating cyclic liquefaction potential using the cone penetration test", *Can. Geotech. J.*, **35**(3), 442-459. https://doi.org/10.1139/t98-017.
Sahin, E.K. and Demir, S. (2023), "Greedy-AutoML: A novel greedy-based stacking ensemble learning framework for assessing soil liquefaction potential", *Eng. Appl. Artif. Intell.*, **119**, 105732. https://doi.org/10.1016/j.engappai.2022.105732.
Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V.M., Guo, H., Hamdia, K., Zhuang, X. and Rabczuk, T. (2020), "An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications", *Comput. Method. Appl. Mech. Eng.*, **36**(2), 112790. https://doi.org/10.1016/j.cma.2019.112790.
Samui, P. (2007), "Seismic liquefaction potential assessment by using relevance vector machine", *Earthq. Eng. Eng. Vib.*, **6**(4), 331-336. https://doi.org/10.1007/s11803-007-0766-7.
Samui, P. and Sitharam, T.G. (2011), "Machine learning modelling for predicting soil liquefaction susceptibility", *Nat. Hazards Earth Syst. Sci.*, **11**(1), 1-9. https://doi.org/10.5194/nhess-11-1-2011.
Seed, H.B. and Idriss, I.M. (1971), "Simplified procedure for evaluating soil liquefaction potential", *J. Soil Mech. Found. Division*, **97**(9), 1249-1273. https://doi.org/10.1061/JSFEAQ.0001662.
Sonmezer, Y.B., Akyuz, A., Kayabali, K., Sonmezer, Y.B., Akyuz, A. and Kayabali, K. (2020), "Investigation of the effect of grain size on liquefaction potential of sands", *Geomech. Eng.*, **20**(3), 243-254. https://doi.org/10.12989/gae.2020.20.3.243.
Stokoe, K., Roesset, J., Bierschwale, J.G. and Aouad, M. (1988), "Liquefaction potential of sands from shear wave velocity", *Proceedings of the 9th World Conference on Earthquake Engineering*, Tokyo-Kyoto, Japan.
Sui, Q., Chen, Q., Wang, D. and Tao, Z. (2023), "Application of machine learning to the V_s-based soil liquefaction potential assessment", *J. Mountain Sci.*, **20**(8), 2197-2213. https://doi.org/10.1007/s11629-022-7809-4.
Tokimatsu, K. and Uchida, A. (1990), "Correlation between liquefaction resistance and shear wave velocity", *Soils Found.*, **30**(2), 33-42. https://doi.org/10.3208/sandf1972.30.2_33.
Wang, X., Wang, L., Wang, S., Chen, J. and Wu, C. (2021), "An XGBoost-enhanced fast constructive algorithm for food delivery route planning problem", *Comput. Ind. Eng.*, **152**(4), 107029. https://doi.org/10.1016/j.cie.2020.107029.
Weiss, G.M. and Provost, F. (2003), "Learning when training data are costly: the effect of class distribution on tree induction", *J. Artif. Intell. Res.*, **19**, 315-354. https://doi.org/10.48550/arXiv.1106.4557.
Wolpert, D.H. (1992), "Stacked generalization", *Neural Networks*,

5(2), 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1.

Wu, M.H., Wang, J.P., Sung, C.Y. (2023), "Performance of HBF method for soil liquefaction assessment", *J. GeoEng.*, **18**(4), 195-202. https://doi.org/10.6310/jog.202312_18(4).3.

Yuan, B. and Liu, W. (2012), "A measure oriented training scheme for imbalanced classification problems", *In New Frontiers in Applied Data Mining: PAKDD 2011 International Workshops*, Shenzhen, China. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-28320-8_25.

Yuan, X., Sun, C. and Chen, S. (2024), "A clustering-based adaptive undersampling ensemble method for highly unbalanced data classification", *Appl. Soft Comput.*, **159**, 111-659.

Zhang, C. and Ma, Y. (2012), *Ensemble Machine Learning: Methods and Applications*, https://doi.org/10.1007/978-1-4419-9326-7.

Zhang, W. and Goh, A.T.C. (2016), "Evaluating seismic liquefaction potential using multivariate adaptive regression splines and logistic regression", *Geomech. Eng.*, **10**(3), 269-284. https://doi.org/10.12989/gae.2016.10.3.269.

Zhang, Y., Qiu, J., Zhang, Y. and Xie, Y. (2021a), "The adoption of a support vector machine optimized by GWO to the prediction of soil liquefaction", *Environ. Earth Sci.*, **80**(9), 1-9. https://doi.org/10.1007/s12665-021-09648-w.

Zhang, W., Wu, C., Zhong, H., Li, Y. and Wang, L. (2021b), "Prediction of undrained shear strength using extreme gradient boosting and random forest based on bayesian optimization", *Geosci. Front.*, **12**(1), 469-477. https://doi.org/10.1016/j.gsf.2020.03.007.

Zhou, J., Huang, S., Wang, M. and Qiu, Y. (2021), "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**(5), 1-19. https://doi.org/10.1007/s00366-021-01418-3

IC

List of abbreviations

ACC: Accuracy Rate; AdaBoost: Adaptive Boosting; AI: Artificial Intelligence; ANN: artificial neural network; AUROC: Area under the Receiver Operating Characteristic Curve; CART: Classification and Regression Trees; CCF: Canonical Correlation Forest; CHAID: Chi-squared automatic interaction detection; CPT: Cone Penetration Test; *CSR*: Cyclic Stress Ratio; DCM: Deep Collocation Method; DNN: Deep Neural Networks; DT: Decision Tree; ENN: Evolutionary Neural Network; ET: Extra Tree; E-CHAID: Exhaustive-Chi-squared automatic interaction detection; FN: Total Number of misclassified liquefaction instances; FP: Total Number of misclassified non-liquefaction instances; GB: Gradient Boosting; GBM: Gradient Boosting Machine; GOSS: Gradient-based one-side sampling; GP: Genetic Programming; GUI: Graphical User Interface; GWO: Grey Wolf Optimization; HBF: Hyperbolic Function; HGB: Hist Gradient Boosting; IQR: Interquartile Range; KFDA: Kernel Fisher discriminant analysis; K-NN: K-Nearest Neighbor; L: Liquefied; LightGBM: Light gradient boosting machine; LP: Liquefaction potential; LR: Logistic Regression; LSSVM: Least squares support vector machine; MARS: Multivariate Adaptive Regression Splines; MCC: Matthews Correlation Coefficient; MGGP: Multi-gene Genetic Programming; ML: Machine Learning; MOFS: Multi-Objective Feature Selection; MOSOS: Multi-objective Symbiotic Organisms Search Algorithm; NB: Naive Bayes; NL: Non-liquefied; NSGA-II: Non-dominated Sorting Genetic Algorithm; OA: Overall Accuracy; PRE: Precision rate; PSO: Particle Swarm Optimization; Q1: Quartile 1; Q3: Quartile 3; QP: Quadratic Programming; REC: Recall rate; RF: Random Forest; ROC: Receiver Operating Characteristic Curve; RotFor: Rotation Forest; RVM: Relevance Vector Machines; SASW: Spectral Analysis of Surface Wave; SDT: Simple Decision Tree; SEL: Stacking Ensemble Learning; SPT: Standard Penetration Test; StDev: Standard Deviation; SV: Support Vector; SVM: Support vector machine; SWVT: Shear Wave Velocity Test; TN: Total Number of correctly classified non-liquefaction instances; TP: Total Number of correctly classified liquefaction instances; WNN: Wavelet Neural Network; XGBoost: Extreme Gradient Boosting.

List of symbols

A_L: Liquefaction class instance accuracy; a_{max}: Peak horizontal ground acceleration; A_{NL}: Non-liquefaction class instance accuracy; a_t: Threshold horizontal ground acceleration; d: depth of the soil layer; d_w: depth of the water table; F_1: F_1 Score; F_{75}: Percentage of fines less than 75μ; f_s: Cone sleeve resistance; g: Gravitational acceleration; $G_{mean(error)}$: G_{mean} in terms of error rate; G_{mean}: the geometric mean of the individual accuracies of each class instance; M_w: Moment Magnitude; $(N_1)_{60}$: Corrected SPT value; q_c: Normalized average cone tip resistance; r_d: Non-linear shear mass participation factor; R_f: Friction ratio; V_s: shear wave velocity; σ'_v: Effective vertical stress; σ_v: Total vertical stress; ϕ: Initial friction angle of soil.