J. Inst. Eng. India Ser. A (December 2022) 103(4):1195–1208
https://doi.org/10.1007/s40030-022-00683-9

ORIGINAL CONTRIBUTION

Prediction of Probability of Liquefaction Using Soft Computing Techniques

Divesh Ranjan Kumar[1] · Pijush Samui[1] · Avijit Burman[1]

Received: 11 June 2022 / Accepted: 17 August 2022 / Published online: 7 September 2022

Abstract Prediction of liquefaction potential of any soil deposit is itself a very challenging task. The problem becomes even more demanding when it becomes necessary to incorporate the variability of all related parameters. Because the parameters that impact liquefaction potential are inherently unknown, the problem is probabilistic rather than deterministic. In the literature, probabilistic analysis of liquefaction potential has attracted a lot of attention, and it's been shown to be a useful technique for evaluating uncertainty inherent in the problem. Machine Learning (ML) techniques have found their applications in all fields of science and engineering while dealing with problems of stochastic nature. These techniques are capable of finding out the desired outputs very effectively. In this paper, five different ML models namely, extreme gradient boosting (XGBoost), random forest (RF), gradient boosting machines (GBM), support vector regression (SVR), and group method of data handling (GMDH) have been used for evaluation of probability of liquefaction based on standard penetration test data. In this study, analysis has been carried out with six input variable such as, depth of penetration, corrected standard penetration blow number, total vertical stress, fine content, maximum horizontal acceleration, total effective stress, and earthquake magnitude. To examine the capabilities of the suggested models in predicting the probability of liquefaction, several statistical parameters have been examined. To compare the accuracy of the proposed models, Taylor graph, REC curve, and error matrix have been developed. While all of the proposed models could efficiently predict the probability of liquefaction. XGBoost model has been found to give the best prediction among all five models. In summary, XGBoost model attained ($R^2 = 0.978$ for training and $R^2 = 0.799$ for testing), GBM model attained ($R^2 = .953$ for training and $R^2 = 0.780$ for testing), RF model attained ($R^2 = .930$ for training and $R^2 = 0.769$ for testing), SVR model attained ($R^2 = .702$. for training and $R^2 = 0.778$ for testing), GMDH model attained ($R^2 = 0.650$ for training and $R^2 = 0.701$ for testing). The proposed models can also be utilized as a valid model for forecasting the probability of liquefaction efficiently for complicated real-world earthquake engineering problems.

Keywords Liquefaction · SPT · XGBoost · GBM · SVR · GMDH

Introduction

An earthquake is a natural event that can be extremely devastating leading to Landslides, debris flows, soil liquefaction, etc. Soil liquefaction is one of the most hazardous phenomenon among them. Terzaghi and Peck (1948) discovered the soil liquefaction phenomenon in the early stages of soil mechanics to explain the loss of strength in saturated loose sand deposit [1]. There have been several reports of soil liquefaction induced due to earthquake in various locations of the world. Because of the ground water label proximity, the likelihood of liquefaction of soil induced by an earthquake is higher in some coastal locations [2]. The coastal soil usually has very little cohesive strength. The earth gets loosened and wet in the event of seismic shaking. If an earthquake happens under these conditions, the dramatically increase the pore water pressure in the layer of soil, causing severe deterioration of soil shear strength and decrease in bearing

✉ Divesh Ranjan Kumar
ranjandivesh453@gmail.com

[1] National Institute of Technology Patna, Patna, India

capacity of soil, which is the most common cause of soil liquefaction [3, 4].

Soil liquefaction can cause serious damage to the integrity of any building structure as the stability of the foundations is drastically affected. The surrounding region will sink as a result of the foundations collapsing, causing traffic gridlock and casualties [5, 6]. As a result, measuring the potential for soil liquefaction is of considerable importance and interest [7, 8]. If the risk of soil liquefaction can be more precisely assessed, countermeasures can be put in place ahead of time to reduce the adverse effects of soil liquefaction. Many factors, however, influence the existence and progression of soil liquefaction [9, 10]. Because the majority of these characteristics do not have a direct correlation with soil liquefaction, predicting the liquefaction potential of soil is difficult. Many attempts have been made by many authors around the world to estimate the liquefaction potential of soil, with the goal of proposing novel, relevant, and front-line research approaches in order to solve this issue [11–16]. Seed and Idriss "simplified technique" have a strong reputation for assessing the possibility for soil liquefaction [11]. Many scholars have built on the "simplified procedure" by expanding and developing it based on field test findings. Some common field test methods, namely standard penetration test (SPT), cone penetration test (CPT), and shear wave velocity (V_s) test (SWVT), are used to assessing the liquefaction potential [17–20]. Traditional prediction methods include semiempirical methods, such as the "simplified procedure," and pure empirical methods, which largely trust on "limit states." The limit states method can be used to differentiate between the region of liquefaction or non-liquefaction [21]. Empirical and semiempirical approaches, on the other hand, are insufficient to forecast soil liquefaction potential. In actuality, a variety of features, such as the earthquake's parameters, and qualities of soil, can influence the liquefaction of soil induced by earthquakes. However, because soil parameters are highly unknown, selecting a suitable and adequate experimental equation for evaluating liquefaction potential of soil is problematic [22]. The existing techniques appear to be ineffective in this regard, and another strategy is necessary to achieve a greater estimation capacity level for predicting liquefaction potential of soil.

Artificial intelligence (AI) methods have advanced quickly in recent decades. Machine learning (ML) technologies, in particular, have substantially aided the advancement of several engineering research topics [23–27]. As a result, various researchers have employed and created AI and ML algorithms in the field of soil liquefaction prediction [28–30]. Furthermore, as the amount of data generated by in situ tests has increased in recent decades, the use of AI and machine learning approaches in engineering practice has grown significantly. Approaches based on AI and machine learning may often reach greater accuracy in anticipating the possibility for soil liquefaction than traditional prediction methods [31, 32]. Artificial neural networks (ANNs) are a powerful algorithm that has been widely employed in geotechnical engineering among the various ML algorithms [33, 34]. Juang et al. (2000) construct an ANN model and developed a function of the liquefaction limit state using the SPT database of 243 data sample [35]. Xue and Yang [36] developed the adaptive neuro-fuzzy inference system (ANFIS), a particular neural network model that has great potential to predict the soil liquefaction potential. Other machine learning and AI strategies, such as support vector machine (SVM), relevance vector machine (RVM), and stochastic gradient boosting (SGB), have been effectively utilized to estimate liquefaction potential of soil [21, 29, 37]. These new AI prediction models are not only more accurate than traditional prediction methods, but they are also better alternatives when a large database with a large number of data samples. Furthermore, these techniques do not require the collection of correlation data between each input parameters and the output parameters, and they can successfully handle the complicated collaboration between each distinctive parameters [38]. However, all methods, AI and machine learning-based models have their own set of limitations and no perfect method for forecasting liquefaction potential. The most generally used ML method in this field, the ANN model does not consider the significance of each parameter while addressing a goal problem, it is unable to determine the relationship between output parameters and characteristic input parameters. Black box property, slow convergence, overfitting tendency, and poor generalization performance are some of the flaws of ANN models. Because the occurrence of soil liquefaction is difficult to assess and is frequently influenced by a variety of geological conditions, existing machine learning methods based on AI technique have limited application. As a result, additional prediction models are needed to provide more alternatives for future study.

Extreme gradient boosting (XGBoost), a powerful ML algorithm based on a gradient boosting system [39, 40] was proposed by Chen and Guestrin [41]. XGBoost, in particular, is a powerful datamining method that has been widely utilized and demonstrated to be useful in a variety of regression and classification applications [42, 43]. Random forest (RF) method proposed by Breiman [44], is an ML algorithm with a well-developed system and high flexibility that has been widely used in the field of civil and geotechnical engineering. RF model was used by various researchers to access the liquefaction potential of soil [38, 45]. RF is a very well-developed and commonly used integrated model. First, the RF model's strong performance and high accuracy, when solving variety of technical challenges. Second, whether working with huge amount of data samples or multidimensional input parameters, RF provides excellent processing

capabilities. More crucially, RF algorithms can determine the significance of each feature in the input sample, making them an ideal combination of AI and machine learning techniques [46]. Gradient boosting machine (GBM) is a ML technique for regression and classification problems proposed by [47]. GBM generates an ensemble of weak prediction models as prediction models [48]. Other research using GBM in the field of civil engineering and other technical problem are presented in is by [49]. Vapnik introduced Support vector machine (SVM) [50] is a strong ML algorithm based on theory of statistical learning. SVM has been expanded to address regression problems called support vector regression (SVR) after Vapnik introduced an intensive loss function [51]. Group method of data handling (GMDH) method is proposed by [52] to regulate the dead-end problem of equation multidimensional character and linear dependency found in normal regression equation. The GMDH technique has been effectively implemented in engineering and other fields [53].

The aim of this paper is to explore the feasibility of the proposed models such as, extreme gradient boosting (XGBoost), random forest (RF), gradient boosting machines (GBM), support vector regression (SVR), and group method of data handling (GMDH) based machine learning methods for predicting the soil liquefaction potential using the SPT dataset. In this study, the probability of soil liquefaction is estimated with the help of corrected standard penetration blow number ($N_{1,60}$), fine content (FC), depth (Z), vertical effective stress ($\sigma_v^{'}$), maximum horizontal ground acceleration ($a_{\max}$) magnitude moment (M_{w}), and total vertical total stress (σ_v). The deterministic method proposed by Idriss and Boulanger (2006) for evaluating liquefaction potential is used [13] in this work. A comparative study has also been carried out between the all developed models.

Historical Data Collection

The data sets used in this study contain two different databases, namely "A" and "B" respectively. Database A contains a total of 620 case records in the collection, in which 290 datasets collected from Taiwan earthquake and 330 datasets collected from Kocaeli earthquake. The suggested models were developed using the field test findings of two earthquakes that occurred in Chi-Chi, Taiwan (1999) and Kocaeli, Turkey (1999) [54].

In database "B" also contain SPT test based data. These data include a total of 214 cases documented by Cetin et al. [18] that provides the source of the liquefaction/non-liquefaction case histories. These datasets came from earthquakes in various places of the world. In these cases, the soil types ranged from pure gravels and sands to silt mixes. There are seven parameters in database A and B, which contain the depths ranging from 1 to 20 m. The fine content in percent ranged from 0 to 90, and the adjusted SPT blow count ranged from 2 to 50 (In kPa), the vertical effective and total stress ranged from 2 to 188 and 12 to 356, respectively. The value of maximum horizontal ground acceleration varied between 0.08 and 0.7. The magnitude of the earthquakes varied from 5 to 8. As an illustration, sample data of two locations are shown in Table 1.

Table 1 Sample dataset

Location	Z(m)	$N_{1,60}$)	FC (%)	σ_v (kPa)	$\sigma_v^{'}$ (kPa)	a_{max}(g)	M_w
Location 1	1.0	6	90	16.3	14	0.4	7.4
	1.8	8	94	30.9	20.6	0.4	7.4
	2.6	7	100	45.6	27.3	0.4	7.4
	3.4	5	87	60.3	34	0.4	7.4
	4.2	5	74	75.8	41.5	0.4	7.4
	5.0	3	92	90.1	47.8	0.4	7.4
	6.0	3	97	108.2	55.9	0.4	7.4
	7.0	19	70	127.9	65.6	0.4	7.4
	8.0	26	58	147.5	75.2	0.4	7.4
Location 2	1.0	3	74	18	15.8	0.4	7.4
	1.8	5	86	32.8	22.6	0.4	7.4
	3.4	2	85	62.4	36.2	0.4	7.4
	4.2	10	93	77.4	43.2	0.4	7.4
	6.0	4	99	109.5	57.3	0.4	7.4
	7.0	11	85	128.3	66.1	0.4	7.4
	8.5	39	8	156.2	79	0.4	7.4
	10.0	25	6	186.6	94.4	0.4	7.4

Methodology

Deterministic approach

In the current study, the soil liquefaction due to seismic loading is commonly expressed in terms of cyclic stress ratio (CSR). Boulanger and Idriss (2014) proposed a stress-based technique to determine the cyclic stress ratio (CSR) and using Eq. 1. The value of CSR is frequently "normalized" to a standard reference with moment magnitude $M_w = 7.5$ to simulate the field condition with different earthquake magnitude [55].

$$(\mathrm{CSR})_{M=7.5,\sigma_v'=1} = 0.65\left(\frac{\sigma_v}{\sigma_v'}\right)\left(\frac{a_{\max}}{g}\right)\frac{r_d}{\mathrm{MSF}} \tag{1}$$

here σ_v, and σ_v' represent the total vertical and effective vertical stress respectively. $a_{\max}$ represents the maximum ground acceleration and r_d represent the stress reduction factor. The factor of 0.65 is taken to transform the peak cyclic shear stress ratio to the most significant cycle during the entire loading period. Parameter r_d can be calculated using Eq. (2) as follows given by [55]:

$$r_d = \exp\left[\alpha(z) + \beta(z).M_w\right] \tag{2}$$

The parameters $\alpha(z)$ and $\beta(z)$ in the above Eq. 2 are calculated using Eqs. (3) and (4).

$$\alpha(z) = -1.012 - 1.126\sin\left(\frac{z}{11.73} + 5.133\right) \tag{3}$$

$$\beta(z) = 0.106 + 0.118\sin\left(\frac{z}{11,28} + 5.142\right) \tag{4}$$

here z denotes the depth below the ground surface. The value of r_d depends upon z. When the depth is more than 10 m, there is an increase in the uncertainty during estimation of r_d.

The influence of the improved MSF relationship on the SPT-based technique that accounts for the effects of soil type and density was investigated by Boulanger and Idriss [55]. The variable MSF can be calculated using following relationship.

$$\mathrm{MSF} = 1 + (\mathrm{MSF}_{\max} - 1) \times \left(8.64 \times e^{\left(-\frac{M_w}{4}\right)} - 1.325\right) \tag{5}$$

$$\mathrm{MSF}_{\max} = 1.09 + \left(\frac{(N_1)_{60cs}}{31.5}\right)^2 \le 2.2 \tag{6}$$

Based primarily on this concept, the following is recommended by [55] for determination of Cyclic resistance ratio (CRR) using SPT data is calculated using Eq. (7) in terms of $(N_{1,60,cs})$, where $N_{1,60cs}$ represents the clean sand equivalent SPT penetration resistance value which is calculated using Eq. (10) and (11) respectively.

$$\mathrm{CRR}_{M=7.5,\sigma_v'=1} = \exp\left(\frac{(N_1)_{60cs}}{14.1} + \left(\frac{(N_1)_{60cs}}{126}\right)^2 - \left(\frac{(N_1)_{60cs}}{23.6}\right)^3 + \left(\frac{(N_1)_{60cs}}{25.4}\right)^4 - 2.8\right) \tag{7}$$

$$(N_1)_{60cs} = (N_1)_{60} + \Delta(N_1)_{60} \tag{8}$$

$$\Delta(N_1)_{60} = \exp\left(1.63 + \frac{9.7}{\mathrm{FC} + 0.01} - \left(\frac{15.7}{\mathrm{FC} + 0.01}\right)^2\right) \tag{9}$$

here $(N_{1,60,cs})$ denotes the clean-sand equivalence of the overburden stress, for corrected SPT blow count calculation refer and FC represent the fine content in percentage. The final CRR for any other value of M and σ_v' is calculated using

$$\mathrm{CRR}_{M,\sigma_v'} = \mathrm{CRR}_{M=7.5,\sigma_v'=1} \times \mathrm{MSF} \times K_\sigma \tag{10}$$

Because the case history database is dominated by level or nearly level ground conditions, the effect of persistent static shear stresses, which can be described through a K_σ factor, is often minor for nearly level ground conditions and is not included herein.

In the deterministic evaluation, factor of safety F_s, defined as $F_s = \mathrm{CRR}/\mathrm{CSR}$ is used to calculate the probability of liquefaction.

In the present work, the factor of safety (F_s) has been calculated using different random variable parameters of SPT based dataset. The various parameters involved in liquefaction analysis (i.e., corrected SPT value $(N_{1,60})$, maximum ground acceleration $(a_{\max})$, fine content (FC), total vertical stress (σ_v), effective vertical stress (σ_v'), and magnitude moment (M_w)) have been randomly generated with the help of coefficient of variation (COV). The coefficient of variation (COV) is expressed in Eq. (13) as the ratio of standard deviation of parameters to the mean of those parameters [7, 56–59].

$$\mathrm{COV} = \frac{\sigma_m}{\mu_m} \tag{11}$$

here σ_m and μ_m are denoted as mean of standard deviation and the mean value of ten randomly generated datasets of the input variables. Phule and Choudhury (2017) prescribed the range of COV values for various parameters used for generating the random datasets for all the six parameters, as shown in Table 2. Factor of safety is calculated for each random variable generated using the COV and finally reliability index β is calculated using Eq. (14).

$$\beta = \frac{\mu_F - 1}{\sigma_F} \tag{12}$$

Here σ_F represents the standard deviation of factor safety of ten randomly generated datasets of input variables and μ_F is the mean factor of safety. Then probability of failure (P_L) is estimated with the help of MATLAB using the relation $P_L = 1 - \varphi(\beta)$ where φ is denoted as standard normal cumulative distribution function.

Description of ML Techniques

In the present work, a few ML techniques namely extreme gradient boosting (XGBoost), Random Forest (RF), gradient boosting machines (GBM), support vector regression (SVR), and group method of data handling (GMDH) have been used to model the liquefaction potential probability of the study area. A brief description of all these ML techniques is provided in this section.

Extreme Gradient Boosting (XGBoost)

XGBoost is a machine learning algorithm that was recently invented by Chen and Guestrin (2016) and is now widely used in a variety of applications. Because it is well-organized, portable, versatile, and has the fastest and most-integrated decision tree algorithm, it will be appropriate for a extensive variety of applications. The algorithm combines the gradient boosting machine (GBM) and cause based decision tree (CBDT) into a single efficient approach. It increases the capacity of the tree boosting approach to process for almost all types of data fast and reliably. XGBoost is a powerful and adaptable tool that can handle a wide range of classifications and regressions, as well as user-defined objective functions for the desired output. XGBoost can also be used to process large datasets with numerous attributes and classifications. This method also provides realistic and capable solutions for new optimization problems, particularly when efficiency and accuracy trade-offs are taken into account.

The objective function of XGBoost is almost similar to that of other ML models, and it might be the combination of the regular term and the loss function. The accuracy of the model is controlled by the loss function, while the model's complexity is controlled by the regular term. At each iteration, XGBoost uses the residual to calibrate the prior prediction; this is a method of improving output of loss function. XGBoost incorporates regularization into the objective function to limit the danger of overfitting during the calibration phase.

$$L(\phi) = l(\phi) + \Omega(\phi) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i\right) + \sum_{K=1}^{K} \Omega\left(f_k\right) \tag{13}$$

Chen and Guestrin (2016) suggest the regularization term for the decision tree (DT):

$$\Omega(f) = yT + \frac{1}{2}\lambda \sum_{j-1}^{T} \omega_j^2 \tag{14}$$

where Φ is the parameter skilled from the provided data, the second term Ω represent the regularization term, which avoid overfitting because it can regulate the model complexity, l denotes the training loss function, y_i is the observed value, y represent the difficulty of individual leaf, T denotes the total leaves present in that DT, λ denotes the compromise parameter that is mostly used for grading the penalty, and ω_j signifies the score on the j-th leaf.

Tree ensemble model that contains functions as parameters shown in Eq. (10). It train the model in a different way, assuming that y is the estimate of the i^{th} instance at the t^{th} iteration, and a different function $f(t)$ is introduced to diminish the subsequent objective.

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t\left(x_i\right) + \Omega\left(f_t\right) \tag{15}$$

The solutions of (Eq. 11) is approximated by the expansion of Taylor's series [41, 62]. The objective function calculated using Eq. 11 and the mean square error (MSE) is expected as the loss function (LF).

$$L^{(t)} \approx \sum_{i=1}^{n} \left[g_i \omega_{q(x_i)} + \frac{1}{2}(h_i \omega_{q(x_i)}^2\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{16}$$

where g_i and h_i indicate the first and second derivatives of loss function respectively, and q indicate a function which associated to a data point of that corresponding leaf.

Equation (12) represent the combination of loss values of each data sample which corresponds to one leaf node.

$$L^{(t)} \approx \gamma T + \sum_{j=1}^{t} \left[\left(\sum_{i \in i_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum h_i + \lambda\right)\omega_j^2\right] \tag{17}$$

Accordingly, G_j and H_j are defined as

Table 2 Coefficient of variation for different parameters

S. No	Parameter	COV	Reference
1	Corrected 'N' value	0.1–0.40	[57, 58, 60]
2	Fine content (FC)	0.05–0.35	[58]
3	σ'_v	0.1–0.2	[59]
4	σ_v	5–20	[61]
5	a_{max}	10–20	[61]
6	M_w	5–10	[61]

$$G_j = \sum_{i \in i_j} g_i, H_j = \sum_{i \in i_j} h_i$$

where I_j represent total data samples in leaf node j.

Overall, finding the minimum of a quadratic function can be used to convert the optimization of the objective function. In other words, the objective function is used to measure the variation in model output when node in the DT separated. This split will be used if the model output is better than previous one; else, the separation will be terminated. Furthermore, regularization aids in the avoidance of over-fitting [41, 63].

Random Forest (RF)

In decision tree learning, the random forest (RF) model is very popular for performing regression and classification. It is incredibly efficient, and it outperforms other regression models in terms of regression accuracy. The concept of RF model is introduced by Breiman (2001). The model combines the outcomes of individually decision tree using a specific combination approach, resulting in a classifier having good simplification capability. There are two ways to seen the randomness of RF: (1) The characteristics used in training are chosen at random during the construction of each decision tree, and not all features are used in the training procedure: (2) To train the model the bootstrap samples are chosen at random from the training samples. Because each decision tree's development process is separate, parallel processing can be performed during the model construction phase to increase productivity. The following is the procedure for building a model:

- Feature selection: Each DT is built by choosing *m* features to avoid overfitting. at random from a entire of *M* features as input variables.
- Sample Selection: Each DT is built using bootstrap sampling, which selects *n* samples for training from a total of *N* samples, and those that aren't drawn considered as out-of-bag data (OOB).
- Grouping of decision tree: The model error is calculated using the out-of-bag data of each tree after n decision trees have been constructed and trained. The output of the model is then derived by averaging the values of all of the individual trees' outputs.

The following hyperparameters must be artificially determined before the RF model can be recognized: (1) Total number of DT, (2) The maximum quantity of features, (3) the least number of samples needed at the leaf node, (4) To split an inner node the minimum number of samples required (5) the maximum tree depth, and (6) the maximum number of leaf nodes. For final decision result of the RF model detail description refer [44, 64].

Gradient Boosting Machines (GBM)

Boosting algorithms iteratively combine weak learners, i.e., learners who are somewhat better than random, into a strong learner. Gradient boosting is a regression approach that is similar to boosting [47]. The purpose of GBM algorithms is to estimate, $F(x)$ of the function $F^*(x)$ given a training dataset $= \{x_i, y_i\}_1^N$, which maps instance x to their target values y, by diminishing the predictable value of loss function denoted by,$L(y, F(x))$. Gradient boosting constructs an alternate estimate of $F^*(x)$ as a weighted sum of functions;

$$F_i(x) = F_{i-1}(x) + \rho_i h_i(x) \tag{18}$$

where ρ_i represent the ith function $h_i(x)$ weight, Iteratively, the approximation is built. First, it has been obtained a constant approximation of $F^*(x)$ as

$$F_o(x) = \underset{\alpha}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, \alpha) \tag{19}$$

Models after that should be proceed towards the minimum value.

$$\left(\rho_m, h_m(x)\right) = \underset{\rho,h}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, F_{i-1}\left(x_i\right) + \rho h(x_i)) \tag{20}$$

Rather than directly resolving the optimization issue, each h_m can be considered a greedy step in a gradient descent optimization for F^*. For a new dataset $D = \{x_i, r_{mi}\}_{i=1}^N$ "h_m" is trained for each model, where residuals r_{mi} are determined by following equation.

$$r_{mi} = \left[\frac{\partial L\left(y_i, F(x)\right)}{\partial F(x)}\right]_{F(x)=F_{i-1}(x)} \tag{21}$$

After then, the value of ρ_m is determined by solving a line search optimization problem.

If the model h_m completely fits the residuals for some loss functions, the residuals will become zero in the following iteration, and the process will end prematurely. Several hyperparameters are taken to govern the additive process of gradient boosting. Applying shrinkage to each gradient decent step to minimize each gradient decent step is a logical technique to regularize gradient boosting $F_i(x) = F_{i-1}(x) + v\rho_i h_i(x)$ with $v = (0.1]$, usually taken as 0.1. Furthermore, by reducing the complexity of the trained models, more regularization can be obtained. It can limit the depth of decision trees or the minimum number of instances required to split a node in the case of decision trees. Finally, hyper-parameters that randomize the

base learners, such as random subsampling without replacement, are incorporated in various versions of gradient boosting and can increase the generalization of the ensemble [47]. The gradient boosting approach is a sophisticated and efficient regression tool that can handle complex non-linear function dependencies [65].

Support Vector Regression (SVR)

The support vector machine (SVM) is a set of learning methods proposed by [66]for handling real issues with a short sample, non-linearity, and high dimensionality. It also has a high degree of generalization ability [67]. The SVM is a classification algorithm that can also be used to tackle regression problems. The sample point is meant to be as far away from the real-valued function as possible when constructing a real-valued function, with the divergence between the actual value of the sample and the output value of the function being as little as possible.

As an example, consider linear regression, for a given set of sample, $U = (x_i, y_i), i = 1.2, \ldots n$, in which x_i is represented as input vector, y_i is represented as target output, and n is the total number of samples in the sample set. Many non-linear problems in real life are solved by translating the sample point into a high-dimensional space ($x \to \phi(x)$) using a mapping function ϕ, the optimization problem can be defined as:

$$\min\phi(\lambda) = \frac{1}{2}(\lambda^2) \tag{22}$$

with the constraints,

$$\begin{cases} y_i - \lambda \cdot x_i - b \le \varepsilon \\ \lambda \cdot x_i + b - y_i \le \varepsilon \end{cases}$$

$i = 1.2, \ldots n,$

where λ and b are the normal vectors and offset of the regression function respectively.

Group Method of Data Handling (GMDH)

Group method of data handling (GMDH) is a type of neural network based on the principal of heuristic self-organizing that was proposed by Ivakhnenko (1971). Different types of Artificial Neural Networks are now being used to solve a variety of technical challenges. The GMDH-NN utilized in this paper is a powerful tool for tasks including prediction, data mining, optimization, and pattern recognition. The network structure is made up of numerous layers, each with several neurons and inputs that are chosen in a self-organized fashion.

To create general functional links between input variables and output variables, GMDH-NN often uses the Kolmogorov–Gabor polynomial [69] as a reference function.

$$\begin{aligned} Y &= f(x_1, x_2, \ldots .., x_n) \\ &= a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} x_i x_j \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{ijk} x_i x_j x_k + \ldots \end{aligned} \tag{23}$$

In the above equation $(x_1, x_2, \ldots .., x_n)$ are the input variables, $(a_1, a, \ldots .., a_n)$ represent the weight vector, n represent the total number of input variables, and Y represent the network output, in order to achieve a non-linear mapping between input variable and output variable through learning, GMDH-NN frequently uses a multi-layer iterative procedure to identify neurons in the model construction. The external standard is then used to select the best model. For more details of GMDH, the reader are encouraged to refer the works of Mo et al. (2018).

Data Analysis

This study proposes XGBoost, GBM, RF, SVR, and GMDH machine learning approaches to study the probabilistic nature of liquefaction phenomenon. The entire dataset used in this study is normalized between 0 to 1 using the min–max approach to enhance the performance of the proposed models.

$$A_{Nor} = \frac{A_{Act} - A_{min}}{A_{max} - A_{min}} \tag{24}$$

here A_{max} and A_{min} are the maximum and minimum value of the parameters respectively. A_{Act} and A_{Nor} are the actual and normalized value of the parameters respectively. After the process of normalization, the entire dataset is randomly divided into two parts i.e., training (70%) and testing (30%), training set used for construction of model and testing set used for validation of model. For all developed models input parameters have been taken as SPT blow count and Cyclic Stress Ratio and output parameters have been taken as probability of liquefaction. Input and output data statistical summary given in Table 3.

Different statistical parameters are used to evaluate the accuracy of the proposed model. Root mean square error (RMSE), coefficient of determination (R^2), Weighted mean absolute percentage error (WMAPE), Nash–Sutcliffe efficiency (NS), variance account factor (VAF), performance index (PI), RMSE-observations standard deviation ratio (RSR), mean absolute error (MAE), are calculated using available mathematical expression [70, 71]. It is observed that for a proposed model with higher accuracy, these parameters

Table 3 Statistical summary of input and output variable

Statistics	$N_{1,60}$	CSR	P_L
Max	50.4	0.77	1
Min	1	0.05	0.002
Median	11.000	0.360	0.920
Mean	13.320	0.346	0.732
1st Quartile	7	0.2	0.408
3rd Quartile	18	0.44	1
STDEV	9.051	0.153	0.312
Var	81.921	0.023	0.097
Skewness	1.192	0.395	-0.643
Kurtosis	1.103	−0.290	−1.210

Table 4 Parameters absolute value

Parameters	Absolute value	Parameters	Absolute value
R^2	1	RSR	0
WMAPE	0	NS	1
RMSE	0	MAE	0
VAF	100	PI	2

values should be equal or nearer to their ideal values (shown in Table 4). Among the eight statistical parameters, the four parameters namely R^2, VAF, NS, RSR and PI, were utilized to determine the trend value, while MAE, WMAPE, RMSE were utilized to determine the amount of error of the developed models.

Result and Discussion

The outcomes of the developed models are presented in this study for predicting the probability of liquefaction are evaluated in detail. As stated earlier, two soil parameters, namely corrected SPT blow count and CSR, were considered as input variable to predict the probability of liquefaction of soil. In the first stage, for development of model training dataset was used and subsequently, using testing dataset performance of developed model was assessed. Ten performance parameters were calculated to estimate the accuracy of the developed models from different aspects.

Actual versus Predicted Curve

The results of the several proposed models are plotted on a scatterplot shows the actual versus predicted values for the training and testing dataset in Fig. 1a–j. Based on the experimental and observed values, a scatter plot was plotted along the line $x = y$ to determine the model's prediction ability. The best model prediction is shown by data points on the $x = y$ line. For each of the models shown in Fig. 1a–j, the scatterplot plotted for the training dataset and testing datasets is shown individually.

The plotted points in Fig. 1 are near to the line $y = x$, showing that all models were able to successfully predict liquefaction probability P_L. The symbols T_r denotes training phase and T_s denotes the testing phase, respectively. The equation and R^2 value displayed in the scatterplot conclude that XGBoost model achieve highest R^2 value followed by GBM, RF, SVR, and GMDH. Therefore, XGBoost is found to be most reliable and efficient ML technique amongst all other techniques used in the current work.

Performance Evaluation

The ability of the proposed models to solve geotechnical problems should be assessed. The performance of the created models was measured using a variety of performance criteria in this study. Table 5 represent eight statistical parameters calculated for training phase and testing phase for all developed models. Nash–Sutcliffe efficiency (NS), Coefficient of determination (R^2), root mean square error (RMSE), weighted mean absolute percentage error (WMAPE), variance account factor (VAF), performance index (PI), RMSE-observations standard deviation ratio (RSR), and mean absolute error (MAE), are used to evaluate the performance of developed models in this article. In both training phase and testing phase, the accuracy level of R^2 of models ranges from 65.01% to 97.18%. The errors parameters calculated for training and testing dataset are considered as best because, they are close to their ideal value as mentioned in Table 4.

Score Analysis

To compare the performance of the proposed models, a simple score analysis method is adopted. The score value is calculated for each predictive model for training and testing respectively. The range of score values is chosen on the basis of total number of predictive models i.e., 1 to "n" (n = total number of predictive models). In this article, the models having minimum values for error parameter are awarded a minimum rank 1, and maximum values for error parameters are awarded a maximum rank m, while the maximum value for accuracy parameter is awarded a maximum rank n, individually for training and testing phase. The overall performance (total rank) of all the above models is then calculated by adding the ranks of each dataset. Finally, the total score of both the training and test datasets are added together to get each model's final score. Based on the results of score

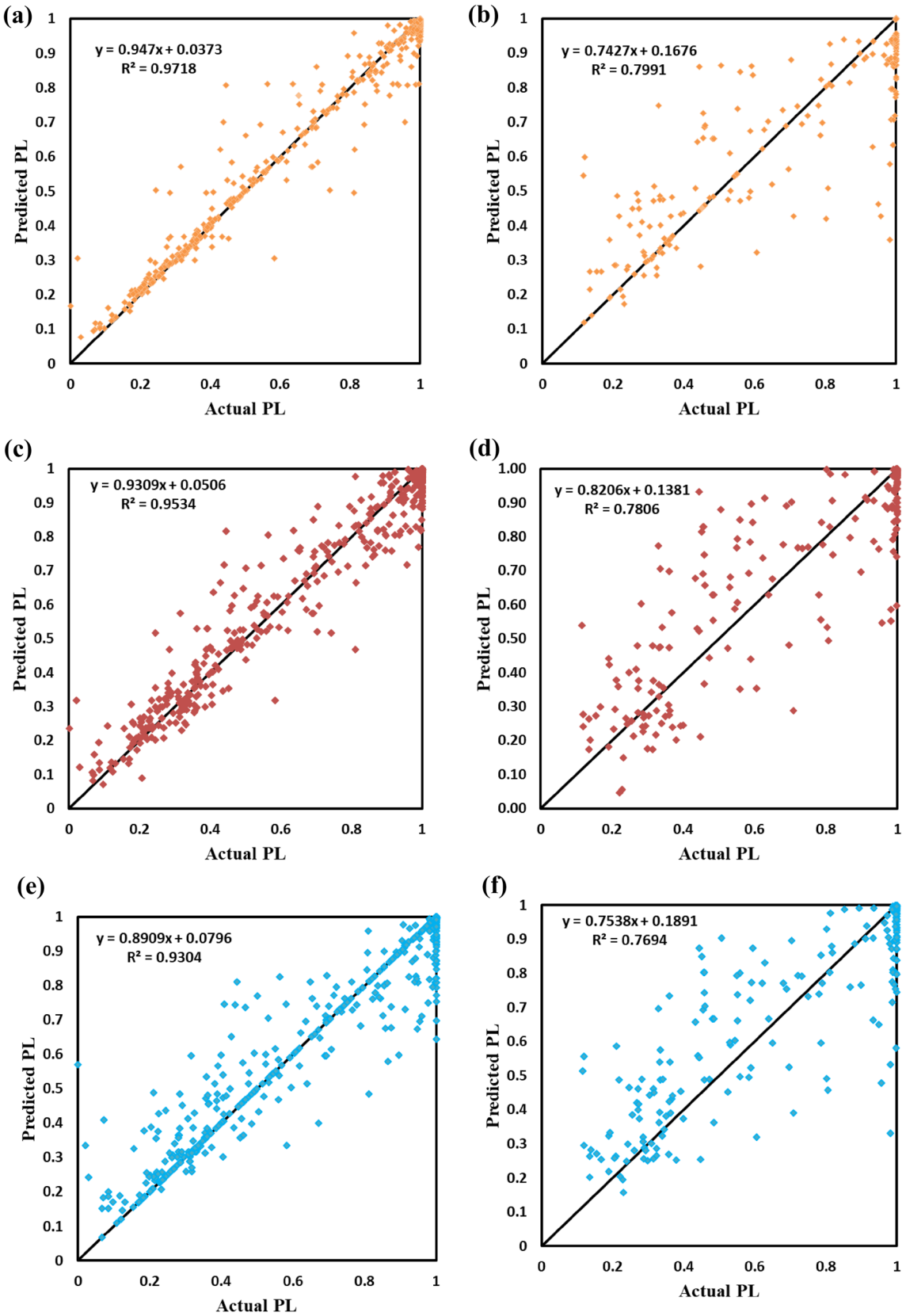

Fig. 1 Compression of Actual and predicted P_L for **a** XGBoost (T_r), **b** XGBoost (T_s), **c** GBM (T_r), **d** GBM (T_s), **e** RF (T_r), **f** RF (T_s), **g** SVR (T_r), **h** SVR (T_s), **i** GMDH (T_r), **j** GMDH (T_s)

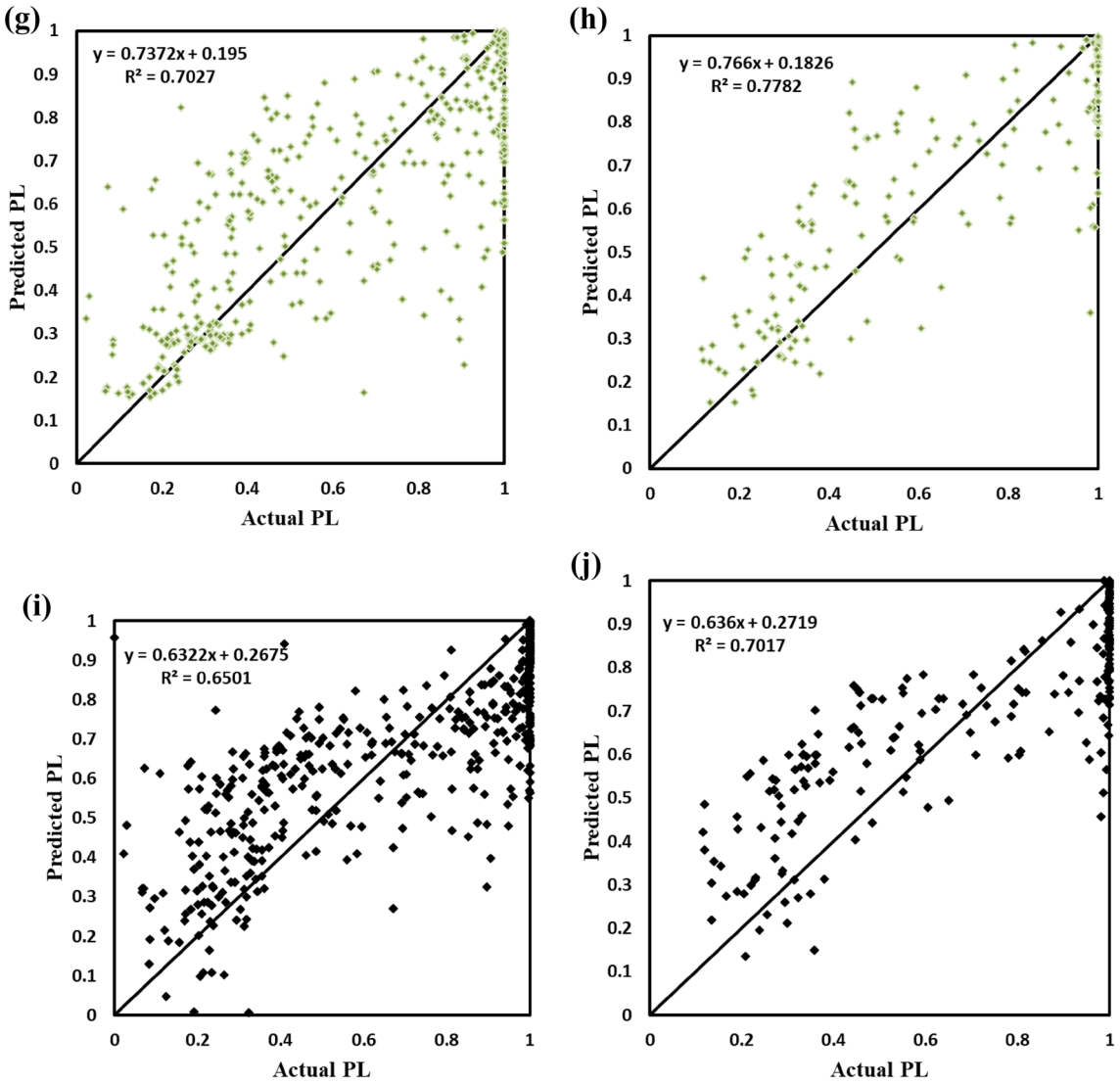

Fig. 1 (continued)

Table 5 Statistical Parameters

Statistical Parameters	XGBoost		GBM		RF		SVR		GMDH	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
R^2	0.9718	0.7991	0.9534	0.7806	0.9304	0.7694	0.7027	0.7782	0.6501	0.7017
WMAPE	0.0342	0.1314	0.0569	0.1334	0.0557	0.1326	0.1454	0.1354	0.1961	0.1893
NS	0.9713	0.7920	0.9531	0.7790	0.9290	0.7691	0.7021	0.7775	0.6510	0.6965
RMSE	0.0532	0.1416	0.0680	0.1459	0.0836	0.1492	0.1713	0.1464	0.1854	0.1710
VAF	97.1164	79.5115	95.2912	77.8522	92.8732	76.9045	70.0960	77.8043	64.9604	69.5513
PI	1.8897	1.4510	1.8382	1.4114	1.7753	1.3874	1.2313	1.4081	1.1131	1.2237
RSR	0.1695	0.4560	0.2166	0.4701	0.2664	0.4805	0.5458	0.4717	0.5908	0.5509
MAE	0.0250	0.0956	0.0417	0.0971	0.0408	0.0965	0.1065	0.0986	0.1436	0.1378

Table 6 Score analysis

Model	XGBoost		GBM		RF		SVR		GMDH	
Parameters	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS
R^2	5	5	4	4	3	2	2	3	1	1
WMAPE	5	5	3	3	4	4	2	2	1	1
NS	5	5	4	4	3	2	2	3	1	1
RMSE	5	5	4	4	3	2	2	3	1	1
VAF	5	5	4	4	3	2	2	3	1	1
PI	5	5	4	4	3	2	2	3	1	1
RSR	5	5	4	4	3	2	2	3	1	1
MAE	5	5	3	3	4	4	2	2	1	1
Sub total	40	40	30	30	26	20	16	22	8	8
Total score	80		60		46		38		16	
Rank	1		2		3		4		5	

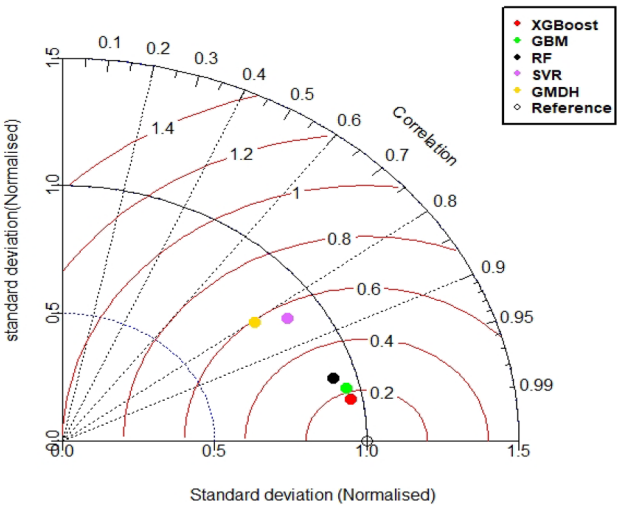

Fig. 2 Taylor graph for Training Dataset

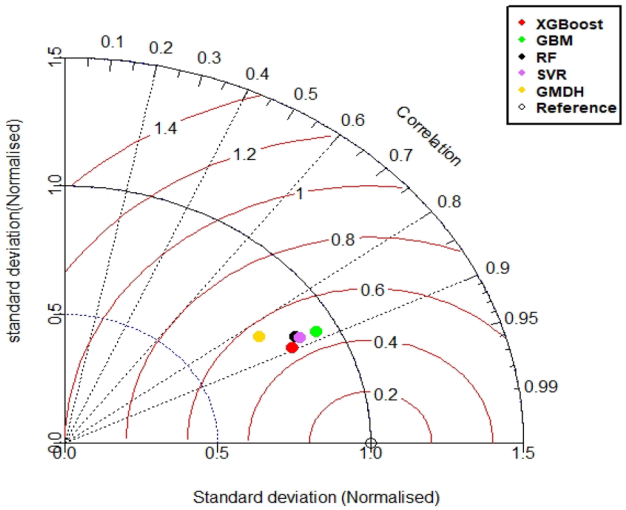

Fig. 3 Taylor graph for Testing dataset

analysis shown in Table 6, it is found that XGBoost has performed best as its rank is highest.

Taylor Graph

Taylor graph has been generated to observe the accuracy of the liquefaction potential predicting models during training and testing. Taylor (2001) established the concept of a Taylor graph, which is a two-dimensional graph designed to graphically depict the accuracy of multiple constructed models. In other words, this diagram presents a graphical and comparative analysis of many models in a single figure. The degree of correspondence between actual and expected behaviors is measured using R, RMSE, and the ratio of SD values, and is represented by a single point. The position of the point should be closer to the reference point in an ideal model. The relative advantages of the models proposed in this article are presented in Fig. 2, where the models' status have been represented by the respective points. It is observed from the given figure that XGBoost model achieves very close to the reference point therefore, XGBoost model is considered as the most robust model for prediction of liquefaction probability (Fig. 3).

Error Matrix

The performance of the models is evaluated using an error matrix, and it is a new graphical concept of showing the error value as heat map matrix. In this analysis, different statistical parameters considered to predict the performance of proposed model for training phase (TR) and testing phase (TS) separately [23]. Figure 4 shows the error matrix of five models as

a heat map matrix of error and trend parameters. Error matrix shows the obtained error (in %) as a function of the parameters by comparing them to their ideal values.

$$E_t = \left|1 - P_t\right| \times 100 \tag{25}$$

$$E_e = \frac{\left|P_e\right|}{i_a} \times 100 \tag{26}$$

where, E_t and E_e represent the error for the trend measuring and error parameters, i_a represent the ideal value of the parameters, P_t and P_e represent the observed values of the trend and error parameters respectively. Parameters which measure the trend (VAF, WI) and error (RMSE, WMAPE, MAE) are calculated using Eq. (28) and Eq. (29) respectively. Finally, all models are compared in terms of overall error, which ranges from 3 to 59%. XGBoost has attained the lowest error (3%) in training phase and GMDH model attain higher error (59%) shows in Fig. 4.

Regression Error Characteristic (REC) Curve

The REC curves achieve three objectives at a time. REC curves show absolute deviation of error tolerance on the *x*-axis of the plot and proportion of observations predicted within the specified tolerance on the *y*-axis. As a result, the error tolerance and regression function accuracy have an obvious trade-off. The idea of error tolerance is appealing because most regression data are intrinsically wrong. The generated curve estimates the error between the experimental and observed values' cumulative distribution function. The amount of error is expressed as an absolute deviation or a squared residual in the REC curve. Curve area provide a valid measure of a regression model's performance, sometimes known as the area over the curve (AOC), value of AOC value should be as low as feasible for proposed model, and the location of the curve should correspond with the y axis (Figs. 5, 6).

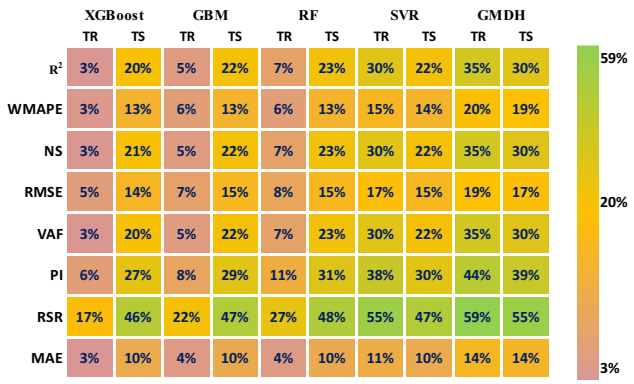

Fig. 4 Error matrix

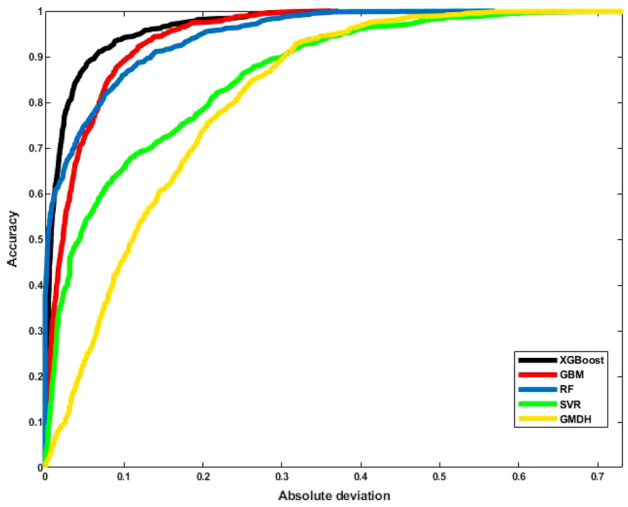

Fig. 5 REC curve for Training Dataset

Result and Discussion

The developed models are capable to estimate the probability of liquefaction in every possible way. Initially, the evaluation is carried out using eight performance indices., named as R^2, RMSE, WMAPE, NS, VAF, PI, RSR, and MAE. The best prediction performance was achieved by the XGBoost model with ($R^2 = 0.978$.and RMSE = 0.053) and ($R^2 = 0.799$ and RMSE = 0.141) in the training and testing phase, respectively. The GBM model attained ($R^2 = 0.953$. and RMSE = 0.068) and ($R^2 = 0.780$ and RMSE = 0.145), RF model attained ($R^2 = 0.930$.and RMSE = 0.083) and ($R^2 = 0.769$ and RMSE = 0.149), SVR model attained ($R^2 = 0.702$.and RMSE = 0.171) and ($R^2 = 0.778$ and RMSE = 0.146), GMDH model attained ($R^2 = 0.650$ and

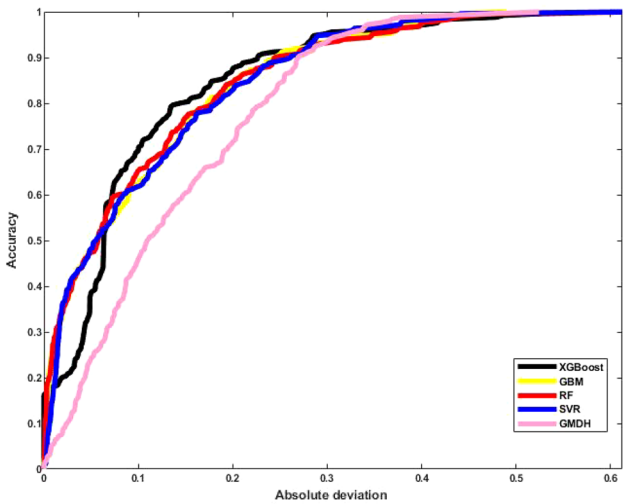

Fig. 6 REC curve for Training Dataset

RMSE = 0.185) and (R^2 = 0.701 and RMSE = 0.171) in the training and testing phase, respectively. The calculated statistical parameters value for the proposed models are presented in Table 5. It is concluded that XGBoost model outperforms all other ML techniques (i.e., GBM, RF, SVR and GMDH) while simulating the probabilistic nature of liquefaction. Additionally, visualization of obtained result using Taylor graph, REC curve, and error matrix have been presented. The Taylor's graph provides a clear comparative representation of the models in terms of coefficient of correlation, RMSE, and ratio of standard deviation. The REC curve and accuracy matrix, show the cumulative distribution of the squared error and the degree of inaccuracy respectively. Finally, the performance of the best prediction model was assessed using a novel method called as "Score Analysis". The XGBoost model attain the higher final score of 80 while the GMDH model attain lowest score (16) value. It may also be concluded that the XGBoost, GBM, RF, SVR, and GMDH algorithm is certainly helpful in the forecast of prediction of probability of liquefaction.

Conclusion

To predict the probability of liquefaction of soil, five models, namely XGBoost GBM, RF, and GMDH were advanced and validate in this paper. Experimental field test data was taken from the literature of several earthquakes study. First, the entire dataset was divided into two sets training and testing. Then. the training dataset is used to construct the model and the testing sets was used to validate the developed models. Statistical parameters calculation concludes that, the XGBoost model obtained the highest predictive efficiency in the testing phase with R^2 = 0.799, RMSE = 0.141, MAE = 0.0956, VAF = 79.511, PI = 1.451, RSR = 0.456, WMAPE = 0.1314, and NS = 0.792. The "Score Analysis" further demonstrated that, at all point, the XGBoost model outperforms the others models. The XGBoost algorithm is a promising tool in predicting the probability of liquefaction of soil based on the outcomes. This research aims to not only replace actual test operations for determining probability of liquefaction of soils, but also to offer details about previous AI-based investigations, including the results at all levels. In order to produce robust and efficient prediction models, five soft computing-based algorithms were developed. The developed models are notable for their strong generalization potentials, low computing costs, and little over-fitting problems. In comparison to the performance in the training phase, the models' performance in the testing phase was shown to be in good accord. During the testing phase, no significant fluctuations or undesired values were found, demonstrating the model's generalization capabilities and robustness. The XGBoost model followed by GBM, RF, SVR, and GMDH models can be introduced as a viable option to aid geotechnical specialists in predicting the probability of liquefaction of soils, based on its overall performance.

Funding The authors have not disclosed any funding.

Declarations

Conflict of interest The author has no conflict of interest to declare that are relevant to the content of this article.

References

1. N. Najdanovic and R. Obradovic, *(Soil Mechanics in Engineering Practice).* (1981).
2. C. Guoxing, K. Mengyun, S. Khoshnevisan, C. Weiyun, L. Xiaojun, Bull. Eng. Geol. Environ. **78**, 945 (2019)
3. P. Samui, T.G. Sitharam, Nat. Hazards Earth Syst. Sci. **11**, 1 (2011)
4. C.S. El Mohtar, A. Bobet, V.P. Drnevich, C.T. Johnston, M.C. Santagata, Geotechnique **64**, 108 (2014)
5. T.L. Youd, I.M. Idriss, J. Geotech. Geoenvironmental Eng. **127**, 297 (2001)
6. A. Ter-Martirosyan and L. D. Anh, in *IOP Conf. Ser. Mater. Sci. Eng.* (IOP Publishing, 2020), p. 52025.
7. C. H. Juang and T. Jiang, in *Proc. Sess. Geo-Denver 2000 - Soil Dyn. Liq. 2000, GSP 107* (2000), pp. 148–162.
8. P. Samui, D. Kim, T.G. Sitharam, J. Appl. Geophys. **73**, 8 (2011)
9. A. Mahmood, X. Wei Tang, J. Nan Qiu, W. Jing Gu, A. Feezan, J. Cent. South Univ. **27**, 500 (2020)
10. M. Ahmad, X. Tang, F. Ahmad, M. Hadzima-Nyarko, A. Nawaz, and A. Farooq, in *Earthquakes—From Tectonics to Build.* (IntechOpen, 2021).
11. H.B. Seed, I.M. Idriss, ASCE J. Soil Mech. Found. Div. **97**, 1249 (1971)
12. H.B. Seed, I.M. Idriss, I. Arango, J. Geotech. Eng. **109**, 458 (1983)
13. I.M. Idriss, R.W. Boulanger, Soil Dyn. Earthq. Eng. **26**, 115 (2006)
14. C.H. Juang, J. Ching, Z. Luo, C.S. Ku, Eng. Geol. **133–134**, 85 (2012)
15. X. Xue, M. Xiao, Environ Earth Sci. **75**, 1 (2016)
16. L. Zhang, Soil Dyn. Earthq. Eng. **17**, 219 (1998)
17. A.T.C. Goh, Can. Geotech. J. **39**, 219 (2002)
18. K.O. Cetin, R.B. Seed, A. Der Kiureghian, K. Tokimatsu, L.F. Harder, R.E. Kayen, R.E.S. Moss, J. Geotech. Geoenviron. Eng. **130**, 1314 (2004)
19. G. Zhang, P.K. Robertson, R.W.I. Brachman, J. Geotech. Geoenviron. Eng. **130**, 861 (2004)
20. C. Hsein Juang, H. Yuan, D. H. Lee, and C. S. Ku, Soil Dyn. Earthq. Eng. **22**, 241 (2002).
21. M. Pal, Int. J. Numer. Anal. Methods Geomech. **30**, 983 (2006)
22. X. Xue, X. Yang, Bull. Eng. Geol. Environ. **75**, 153 (2016)
23. T. Pradeep, A. Bardhan, P. Samui, Innov. Infrastruct. Solut. **7**, 37 (2022)
24. D.J. Armaghani, H. Harandizadeh, E. Momeni, H. Maizir, J. Zhou, Artif. Intell. Rev. **55**, 2313 (2022)
25. M. Hasanipanah, H. Bakhshandeh Amnieh, Eng. Comput. **37**, 1879 (2021)

26. M. Hasanipanah, D. Meng, B. Keshtegar, N.T. Trung, D.K. Thai, Neural Comput. Appl. **33**, 4205 (2021)
27. M. Hasanipanah, H. Bakhshandeh Amnieh, Nat. Resour. Res. **29**, 669 (2020)
28. A.M. Hanna, D. Ural, G. Saygili, Eng. Comput. (Swansea, Wales) **24**, 5 (2007)
29. P. Samui, J. Karthikeyan, Int. J. Numer. Anal. Methods Geomech. **37**, 1154 (2013)
30. Y. Gang Zhang, J. Qiu, Y. Zhang, Y. Wei, Nat. Hazards **107**, 539 (2021)
31. C.Y. Lee, S.G. Chern, J. Mar. Sci. Technol. **21**, 318 (2013)
32. N.D. Hoang, D.T. Bui, Bull. Eng. Geol. Environ. **77**, 191 (2018)
33. P. Samui, Comput. Geotech. **35**, 419 (2008)
34. A. Abbaszadeh Shahri, Geotech. Geol. Eng. **34**, 807 (2016)
35. C.H. Juang, C.J. Chen, T. Jiang, R.D. Andrus, Can. Geotech. J. **37**, 1195 (2000)
36. X. Xue, X. Yang, Nat. Hazards **67**, 901 (2013)
37. A.T.C. Goh, S.H. Goh, Comput. Geotech. **34**, 410 (2007)
38. V.R. Kohestani, M. Hassanlourad, A. Ardakani, Nat. Hazards **79**, 1079 (2015)
39. J. Zhou, X. Li, H.S. Mitri, Nat. Hazards **79**, 291 (2015)
40. J. Zhou, X. Li, H.S. Mitri, J. Comput. Civ. Eng. **30**, 4016003 (2016)
41. T. Chen and C. Guestrin, in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* (2016), pp. 785–794.
42. L.T. Le, H. Nguyen, J. Zhou, J. Dou, H. Moayedi, Appl. Sci. **9**, 2714 (2019)
43. Z. Ding, H. Nguyen, X.N. Bui, J. Zhou, H. Moayedi, Nat. Resour. Res. **29**, 751 (2020)
44. L. Breiman, Mach. Learn. **45**, 5 (2001)
45. J.R. Harris, E.C. Grunsky, Comput. Geosci. **80**, 9 (2015)
46. R. Genuer, J.M. Poggi, C. Tuleau-Malot, Pattern Recognit. Lett. **31**, 2225 (2010)
47. J.H. Friedman, Comput. Stat. Data Anal. **38**, 367 (2002)
48. S. Touzani, J. Granderson, S. Fernandes, Energy Build. **158**, 1533 (2018)
49. P. Nie, M. Roccotelli, M.P. Fanti, Z. Ming, Z. Li, Energy Rep. **7**, 1246 (2021)
50. S. R. Sain and V. N. Vapnik, *The Nature of Statistical Learning Theory* (Springer science & business media, 1996).
51. A.J. Smola, B. Schölkopf, Stat. Comput. **14**, 199 (2004)
52. A. G. Ivakhnenko, G. I. Krotov, and V. N. Visotsky, in *Theor. Syst. Ecol.* (Academic Press New York, 1979), pp. 325–352.
53. M. Najafzadeh, G.A. Barani, M.R. Hessami Kermani, Ocean Eng. **59**, 100 (2013)
54. A.M. Hanna, D. Ural, G. Saygili, Soil Dyn. Earthq. Eng. **27**, 521 (2007)
55. R. W. Boulanger and I. M. Idriss, Cent. Geotech. Model. 1 (2014).
56. R.R. Phule, D. Choudhury, Nat. Hazards **85**, 139 (2017)
57. K.K. Phoon, F.H. Kulhawy, Can. Geotech. J. **36**, 612 (1999)
58. M. Gutierrez, J.M. Duncan, C. Woods, E. Eddy, *Virginia Polytech* (State Univ, Inst, 2003)
59. M. Naghizaderokni and A. Janalizade, COMPDYN 2015 - 5th ECCOMAS Themat. Conf. Comput. Methods Struct. Dyn. Earthq. Eng. **125**, 4214 (2015).
60. M. E. Harr, *Reliability-based design in civil engineering*. Vol. 20. Department of Civil Engineering, School of Engineering, North Carolina State University (1984)
61. C. H. Juang and T. Jiang, Proc. Sess. Geo-Denver 2000 - Soil Dyn. Liq. 2000, GSP 107 **295**, 148 (2000).
62. Y. Xia, C. Liu, Y.Y. Li, N. Liu, Expert Syst. Appl. **78**, 225 (2017)
63. J. Zhou, Y. Qiu, S. Zhu, D.J. Armaghani, M. Khandelwal, E.T. Mohamad, Undergr. Sp. **6**, 506 (2021)
64. M. Belgiu, L. Drăgu, ISPRS J. Photogramm. Remote Sens. **114**, 24 (2016)
65. A. Natekin, A. Knoll, Front. Neurorobot. **7**, 21 (2013)
66. C. Cortes, V. Vapnik, Mach. Learn. **20**, 273 (1995)
67. C.N. Ko, C.M. Lee, Energy **49**, 413 (2013)
68. A.G. Ivakhnenko, IEEE Trans. Syst. Man Cybern. **1**, 364 (1971)
69. L. Mo, L. Xie, X. Jiang, G. Teng, L. Xu, J. Xiao, Appl. Soft Comput. J. **62**, 478 (2018)
70. T. Chai, R.R. Draxler, Geosci. Model Dev. **7**, 1247 (2014)
71. T. Pradeep, A. GuhaRay, A. Bardhan, P. Samui, S. Kumar, and D. J. Armaghani, Arab. J. Sci. Eng. (2022).