Neural Computing and Applications (2021) 33:1533–1546
https://doi.org/10.1007/s00521-020-05084-2

ORIGINAL ARTICLE

An ensemble method to improve prediction of earthquake-induced soil liquefaction: a multi-dataset study

Junfei Zhang[1] · Yuhang Wang[2]

Received: 4 September 2019 / Accepted: 4 June 2020 / Published online: 13 June 2020

Abstract
Evaluation of earthquake-induced liquefaction potential is crucial in the design phase of construction projects. Although several machine learning models achieve good prediction accuracy on their particular datasets, they may not perform well in other liquefaction datasets. To address this issue, we proposed a novel hybrid classifier ensemble to improve generalizability by combining the predictions of seven base classifiers using the weighted voting method. The applied base classifiers include back propagation neural network, support vector machine, decision tree, k-nearest neighbours, logistic regression, multiple linear regression and naïve Bayes. The hyperparameters and weights of the base classifiers were tuned using the genetic algorithm. To verify the robustness of the classifier ensemble, its performance was tested on three datasets collected from previous published researches. The results show that the proposed classifier ensemble outperforms the base classifiers in terms of a variety of performance metrics including accuracy, Kappa, precision, recall, F1 score, AUC and ROC on the three datasets. In addition, the importance of influencing variables was achieved by the classifier ensemble on the three datasets to facilitate the future data collecting work. This robust ensemble method can be extended to solve other classification problems in civil engineering.

Keywords Soil liquefaction · Earthquake · Machine learning · Classifier ensemble · Genetic algorithm · Prediction

1 Introduction

Soil liquefaction is the soil failure due to the loss of shear strength and an increase in pressure of pore water [1, 2]. When this hazard occurs, the sandy soil will exhibit a fluid-like behaviour. Soil liquefaction hazards may occur when the following conditions are satisfied: (1) the soil particles are cohesionless and loose; (2) the sizes of the soil particles range from fine sand to coarse silt; (3) sufficient shaking is provided by the earthquake; and (4) the ground is saturated. The earthquake-induced soil liquefaction may cause various types of structural damages and geo-disasters, such as lateral flow of ground, landslides, uplift of underground facilities, settlement of buildings and even heavy casualties [3, 4]. Therefore, the evaluation of the liquefaction potential induced by earthquake is necessary in the design phase of civil infrastructures. However, predicting soil liquefaction is a big challenge in the field of geotechnical engineering because of the highly nonlinear relationships between the soil liquefaction and a large number of influencing factors. By now, the prediction of soil liquefaction is far from being satisfactory and more and more researchers are focusing on developing approaches to predict soil liquefaction.

Two approaches are widely used to predict soil liquefaction [5, 6, 7]. One is experimental study based on laboratory and field data such as static penetration test (SPT) [8], shear wave velocity method [9] and seed method [10]. The other one is empirical analysis using discriminant formula with seismic field survey data. Discriminant analysis has been used in many fields as well as soil liquefaction prediction [10–15]. However, these methods separate non-liquefaction cases from liquefaction cases by using "limit states" which are developed based on

✉ Junfei Zhang
junfei.zhang@research.uwa.edu.au

[1] Department of Civil, Environmental and Mining Engineering, The University of Western Australia, Perth 6009, Australia

[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing 10093, China

engineering judgement, and therefore, many uncertainties are inevitably induced.

Recent studies have shown that the "limit state" can be established rationally using machine learning (ML) approaches. Hanna et al. established the relationship between potential of soil liquefaction and 12 influencing variables using artificial neural networks (ANNs) on a dataset containing 620 instances collected from earthquake cases in Turkey and Taiwan [16]. Another type of ANNs (fuzzy neural network) was developed by Chern et al. to assess the liquefaction potential of soils. The developed model was tested on a dataset containing actual liquefaction records from over 11 major earthquakes between 1964 and 1999 [17]. Genetic expression programming was also used to establish the complex correlations between the potential of soil liquefaction and seismic properties of soil by Kayadelen [18]. Xue and Yang utilized the advantages of support vector machine (SVM) that operate on the principle of structural minimization rather than minimization of errors to improve prediction of seismic liquefaction. The hyperparameters of SVM were tuned by particle warm optimization (PSO). Their results indicate that the PSO–SVM method surpassed grid search-based models and achieved similar prediction to adaptive network-based fuzzy inference system [19]. To handle the liquefaction dataset with incomplete data, Hu et al. used constructed generic Bayesian network models to replace missing values in the dataset [20]. Besides, tree-based models (e.g. C4.5 decision tree, multivariate adaptive regression splines and random forest) were introduced to predict liquefaction potential of soil due to the pruning process that can solve the problem of overfitting [21–23].

These advanced approaches can improve soil liquefaction prediction accuracy by formulating this problem as a binary classification problem. However, according to the "no free lunch" theorem of ML, there is no single algorithm that performs best on all dataset [24]. A base classifier may achieve high classification accuracy on a soil liquefaction dataset, while cannot perform well on another soil liquefaction dataset. To solve the "unstable" problem when predicting soil classification using base classifiers, this study has proposed a classifier ensemble by aggregating different base classifiers using the weighted voting method to improve soil liquefaction classification accuracy. Generally, an ensemble model is established by two steps: (1) training weak base classifiers and (2) aggregating the base classifiers [25, 26]. A number of aggregating methods have been proposed such as boosting [27], bagging [28], stacking [29] and weighted majority voting [30]. Among these aggregating methods, weighted majority voting is widely used due to their simple implementation procedure and high prediction accuracy. And weighted majority voting is more suitable in classification tasks when aggregating categorical results [31]. In weighted majority voting, weights are given for each classifier and this classifier with better prediction performance will get higher weights in voting. Recent studies have shown that weighted voting schemes can enhance the accuracy and robustness of the classifier ensemble compared with the majority voting [32–34]. When using weighted voting, the weights of different classifiers should be tuned to find the optimal weight combination. This weight tuning algorithm can be treated as an optimization problem, which can be addressed using metaheuristic algorithms such as the genetic algorithm (GA) [35–37].

In this study, we have proposed a novel classifier ensemble by aggregating different base classifiers using the weighted voting method, including backpropagation neural network (BP), support vector machine (SVM), random forest (RF), k-nearest neighbours (KNN), naïve Bayes (NB), logistic regression (LR) and multiple linear regression (MLR). The hyperparameters and weights of different base classifiers are tuned using GA. The construction of the classifier ensemble and its prediction performance will be further discussed in the following sections.

2 Dataset description

In this study, the proposed classifier ensemble is constructed and verified on three datasets collected from the previous literature.

Dataset 1 contains 620 soil liquefaction cases (256 liquefied cases and 364 non-liquefied cases), which are collected from two major earthquakes that occurred in Taiwan and Turkey using SPT test [16]. The liquefaction state of the soil layers is determined by the SPT test results. Twelve influencing variables are included in the dataset: depth of the soil layer (D); corrected standard penetration blow numbers ($(N_1)_{60}$); per cent finest content less than 75 μm (F75); depth of ground water table (d_w); total effective overburden stress (σ_v); effective overburden stress (σ_v'); threshold acceleration (a_t); cyclic stress ratio (CSR); shear wave velocity (V_s); internal friction angle of soil ($\emptyset'$); earthquake moment magnitude (M_v); and maximum horizontal acceleration at ground surface (a_{max}).

Dataset 2 consists of 226 historical cases based on CPT tests [38], including 133 non-liquefied cases and 93 liquefied cases. Six influencing parameters are applied in this dataset: total effective overburden stress (σ_v); effective overburden stress (σ_v'); the cone tip resistance (q_c); the sleeve friction ratio (R_f); the maximum horizontal at ground surface acceleration (a_{max}); and the earthquake moment magnitude (M_v).

In Dataset 3, 185 cases are included. The numbers of non-liquefied and liquefied instances are 78 and 107, respectively [39]. The seven influencing variables used in this dataset are depth of the soil layer (D); soil class (SC); total effective overburden stress (σ_v); effective overburden stress ($\sigma_v^{'}$); shear wave velocity (V_s); earthquake moment magnitude (M_v); and maximum horizontal acceleration at ground surface (a_{max}).

Statistics of the input parameters for the three datasets are tabulated in Table 1. The correlation matrixes of the variables used in the three datasets are shown in Fig. 1. It can be seen that most correlation values are less than 0.5, indicating comparatively weak relationships between most influencing variables [40]. It can be observed that in Dataset 1, for the input parameters, the relationship between $\emptyset'$ and $(N_1)_{60}$ as well as a_{max} and SCR is very high. In Dataset 2, the input parameters $\sigma_v^{'}$ and σ_v have high correlation; in Dataset 3, besides the high relationship between $\sigma_v^{'}$ and σ_v, high relationship is also observed between σ_v and D as well as $\sigma_v^{'}$ and D. Highly correlated features are more linearly dependent and therefore have similar influence on the outputs. Future work may focus on

Table 1 Statistical description of the three datasets

Notation	Unit	Min	Max	Mean	Median	SD
Dataset 1						
D	m	0.8	19.8	7.66	6.70	4.89
σ_v	kPa	12.10	408.90	144.60	121.60	98.12
$\sigma_v^{'}$	kPa	7.50	233.70	82.48	68.15	52.80
F75	%	1.00	100.00	62.99	74.50	34.25
$(N_1)_{60}$	–	1.00	75.00	14.48	11.00	11.38
CSR	–	0.12	0.85	0.37	0.39	0.15
d_w	m	0.35	10.00	1.45	1.10	1.20
a_t	g	0.00	0.85	0.07	0.06	0.07
$\emptyset'$	°	23.46	52.08	31.96	31.41	4.84
V_s	m/s	37.00	500.00	166.98	155.00	67.03
M_v	–	7.40	7.60	7.49	7.40	0.10
a_{max}	g	0.18	0.67	0.38	0.40	0.15
Dataset 2						
q_c	MPa	0.90	25.00	5.82	4.90	4.08
R_f	%	0.10	5.20	1.22	0.90	1.05
a_{max}	g	0.08	0.80	0.29	0.25	0.14
M_v	–	6.00	7.60	6.95	7.10	0.44
σ_v	kPa	26.60	274.00	106.89	90.30	55.24
$\sigma_v^{'}$	kPa	22.50	215.20	74.65	62.80	34.32
Dataset 3						
D	m	2.00	14.80	5.30	4.50	2.34
SC	–	1.00	4.00	2.37	2.00	0.89
V_s	m/s	79.00	274.00	141.61	133.00	30.90
M_v	–	5.90	8.30	6.82	7.10	0.53
a_{max}	g	0.02	0.51	0.21	0.18	0.12
σ_v	kPa	38.40	251.60	94.77	85.40	40.37
$\sigma_v^{'}$	kPa	27.80	140.80	64.60	55.80	29.78

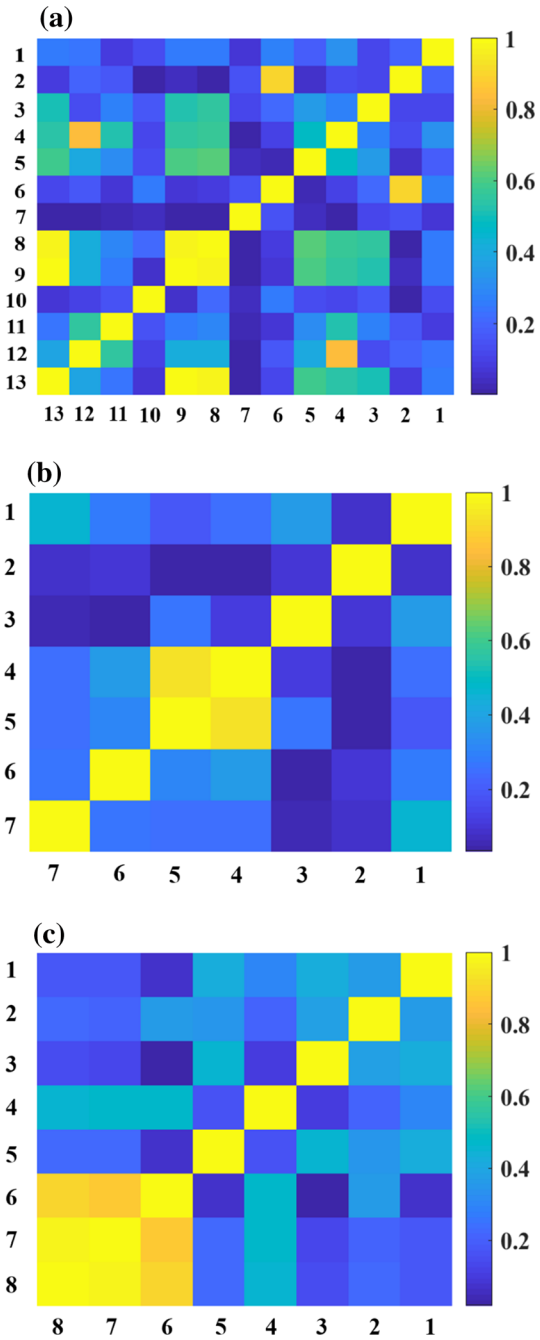

Fig. 1 Correlation matrix for Dataset 1 (**a**), Dataset 2 (**b**) and Dataset 3 (**c**). In Dataset 1, the numbers from 1 to 13 represent liquefaction, a_{max}, M_v, $\emptyset'$, V_s, CSR, a_t, $\sigma_v^{'}$, σ_v, d_w, F75, $(N_1)_{60}$ and D, respectively. In Dataset 2, the numbers from 1 to 7 represent liquefaction, M_v, a_{max}, σ_v, $\sigma_v^{'}$, R_f and q_c, respectively. In Dataset 3, the numbers from 1 to 8 represent liquefaction, M_v, a_{max}, V_s, SC, $\sigma_v^{'}$, σ_v, D, respectively

reduce several high correlated features to reduce computational time without decreasing prediction accuracy.

3 Methodology

3.1 Base classifiers

The classifier ensemble is constructed by combining the predictions of seven base classifiers. The strengths and weaknesses of the base classifiers are summarized in Table 2. Detailed description of each classifier can be found in the previous literature [41, 42].

3.1.1 Back propagation neural network (BP)

BP is a gradient-decent-based algorithm containing one input layer, one output layer and one or more hidden layers [43, 44]. Each layer is composed of one or more neuron nodes. The information propagates from the neuron of the previous layer to those of the next layer. The training procedure contains two processes: the positive information propagation and the reverse error propagation. The training process will stop, and the weights and thresholds will be determined when the mean square error (MSE) decreases to the goal setting range. The internal nodes collect the activation of previous nodes. The collection function can be written as

$$\text{net}_i = \sum_j \omega_{ij} a_j + b_i \tag{1}$$

where ω_{ij} is the weight value between current mode i and previous node j; a_j is the activation of node j; and b_i represents the bias value for node i. The output activation of the node can be determined by the result of the collection function net_i. The sigmoid function is used in this study to determine the output state:

$$a_i = \frac{1}{1 + e^{-\text{net}_i}} \tag{2}$$

3.1.2 Support vector machine (SVM)

Assuming a dataset for a binary classification problem as $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$, where $y_i \in \{-1, +1\}$. The samples of different classes are separated by a hyperplane $\boldsymbol{w}^T \boldsymbol{z} + b = 0$ using the following formula [45–48]:

$$\begin{cases} \boldsymbol{w}^T \boldsymbol{x}_i + b \geq +1, & y_i = +1; \\ \boldsymbol{w}^T \boldsymbol{x}_i + b \leq -1, & y_i = -1. \end{cases} \tag{3}$$

The support vectors are defined as those close to the hyperplane. The margin is defined as the distance between two support vectors of two classes:

$$\gamma = \frac{2}{\|\boldsymbol{w}\|} \tag{4}$$

The hyperplane maximizing γ is the optimal hyperplane. This optimization problem can be reorganized as follows:

$$\begin{aligned} &\max_{\boldsymbol{w},b} \frac{2}{\|\boldsymbol{w}\|} \\ &\text{s.t. } y_i\left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) \geq 1, \quad j = 1, 2, \ldots, m \end{aligned} \tag{5}$$

3.1.3 Decision tree (DT)

DT solves classification problems using a tree-like structure. In the structure, the test on an attribute is represented

Table 2 Strength and weakness of used ML models

Model	Strength	Weakness
BP	Has no parameters to tune besides the numbers of input; does not require prior knowledge about the network; does not need to mention the features of the function to be learned	Sensitive to noisy data; needs to use the matrix-based approach for backpropagation instead of mini-batch
SVM	Performs well in complex domains; works well for nonlinear classification problems with outliers by using kernel methods	Cannot perform well for large datasets with noise
DT	Automatically selects features; does not need to normalize and scale features; insensitive to outliers	Requires to be remodelled for new data; prone to overfitting on high-variance data
KNN	Performs well on noisy data; nonparametric	Sensitive to outliers
NB	Behaves well in multi-class predictions; handles big data with noise	Assumes that all properties independently contribute to the probability, which is almost impossible in real life
LR	Robust to noise; output can be interpreted as probability	Hardly handles categorical features; can only solve problems with linear decision boundaries
MLR	Can determine the relative influence of one or more predictor variables to the criterion value; can effectively identify outliers or anomalies	Has lower accuracy on dataset with lower linearity; easy to get overfitting

by a node; the outcome of the test is represented by the branch; and the class label is represented by the leaf node. The algorithms ID3, C4.5 and CART are commonly used to construct DT [49, 50]. The main differences of these algorithms are the type of impurity and type of the tree. In this study, we use CART algorithm to construct a binary DT by calculating the Gini index as follows [51–53]:

$$\text{Gini}\ (D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{6}$$

where p_i denotes the possibility of a training sample belonging to class C_i in D. The value p_i is calculated by $|C_{i,D}|/|D|$; m is the total number of the classes. The Gini index on a feature A can be given by

$$\text{Gini}_A(D) = \frac{|D_1|}{|D|}\text{Gini}(D_1) + \frac{|D_2|}{|D|}\text{Gini}(D_2) \tag{7}$$

where D_1 and D_2 are two subset of D. An attribution will be chosen as the split if the minimum Gini index is given by the subset.

3.1.4 K-nearest neighbour (KNN)

The KNN algorithm is an instant-based algorithm. It determines the category of new instances using the nearest distance. In the training process, the attribute space is partitioned into several areas, each of which contains the instances with similar contents [54]. The unlabelled data are classified into a category that has the closet distance from the data point. The Euclidean function is the widely used distance function for KNN classifier, which can be written as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{8}$$

where $d(x, y)$ is the Euclidean distance between points x and y; and n denotes the number of dimensions.

3.1.5 Naïve Bayes (NB)

The NB classifier assumes the effect of a feature (x) on a given class (c) is not dependent of other predictors. This so-called class conditional independence can be explained using the following formula [55]:

$$P(c|F) = \frac{P(F|c)P(c)}{P(F)} \tag{9}$$

where c is the class variable; P (c) represents the marginal probability; and $P(F|c)$ and $P(c|F)$ represent the conditional probability and posterior probability, respectively. $P(F|c)$ can be calculated as:

$$P(F|c) = P(f_1, f_2, \ldots, f_n|c) = \prod_{i}^{n} P(f_i|c)$$

3.1.6 Logistic regression (LR)

Assume $\mathbf{x}$ is a d-dimensional attribute vector. A binary class variable can be denoted as $Y \in \{\theta_1, \theta_2\}$. The binary LR model is given by [56]:

$$\ln \frac{p_1(\mathbf{x})}{1 - p_1(\mathbf{x})} = \beta^T x + \beta_0 \tag{10}$$

where $p_1(\mathbf{x})$ represents the probability that $Y = \theta_1$; β_0 and β are parameters; and $p_1(\mathbf{x})$ can be solved:

$$p_1(\mathbf{x}) = \frac{1}{1 + \exp\left[-\left(\beta^T \mathbf{x} + \beta_0\right)\right]} \tag{11}$$

The parameters β_0 and β can be assessed by maximizing the conditional log-likelihood:

$$\ell(\beta, \beta_0) = \sum_{i=1}^{n} y_{i1} \ln p_1(\mathbf{x}_i) + (1 - y_{i1}) \ln[1 - p_1(\mathbf{x}_i)] \tag{12}$$

where $y_{i1} = 1$ if $y_i = \theta_1$ and $y_{i1} = 0$ otherwise.

3.1.7 Multiple linear regression (MLR)

MLR models the relationship between predictors and the response variables by fitting linear equations. A value of the dependent variable y is represented by each value of the independent variable x, as follows [57]:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{13}$$

where $\beta_i (i = 0, 1, \ldots, n)$ denotes the regression coefficients that are assessed by least squares.

3.2 Performance evaluation

3.2.1 Performance evaluation measures

In binary classification problems, the confusion matrix is widely used [58]. A confusion matrix consists of four indexes: true positive (TP), false positive (FP), true negative (TN) and false negative (FN), as shown in Table 3.

Table 3 Confusion matrix

	Actual class	
	Positive	Negative
Predicted class		
Positive	TP	FP
Negative	FN	TN

The following performance metrics can be deduced based on confusion matrix:

- Accuracy (ACC):

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{14}$$

- Cohen's kappa (K)

Cohen's kappa considers the possibility of the agreement that occurs by accident [59]. The Cohen's kappa is defined as follows:

$$K = \frac{P_0 - P_e}{1 - P_e} \tag{15}$$

where P_0 denotes the total agreement probability which equals the accuracy andP_e represents the probability expected by chance.

- Precision (P), recall (R) and F1 score (F1)

The precision and recall are applied together in the F1 score for assessing the prediction accuracy of a classification model. These measures are calculated as follows:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{16}$$

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{17}$$

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{18}$$

- ROC and AUC

In binary classification, receiver operating characteristic (ROC) and area under the curve (AUC) are widely used to evaluate the classification accuracy [60]. ROC is created by taking the true positive rate (TPR) as the vertical axis and false positive rate (FPR) as the horizontal axis at different thresholds. The "ideal" point is situated in the left corner with FPR = 0 and TPR = 1. Therefore, a larger AUC represents better classification accuracy.

3.2.2 K-fold CV

The bias related to the random sampling instances can be reduced by using k-fold cross-validation (CV). The previous study has shown that tenfold is preferable considering the acceptable bias, short validation testing time and variance [61]. Therefore, k is set as 10 in this study. The process of tenfold CV for model selection is shown in Fig. 2. The AUC metric is used to evaluate the performance of the classifier in this process.

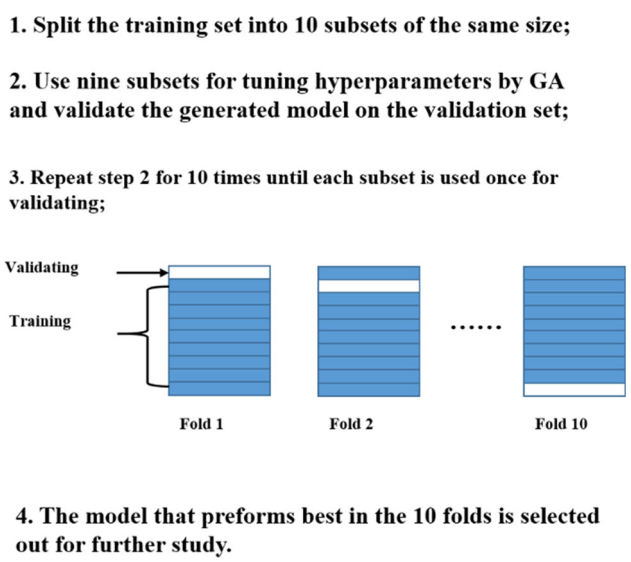

Fig. 2 Tenfold CV for model validation

3.3 Hyperparameter tuning for base classifiers using the genetic algorithm

Genetic algorithm (GA) is one of the widely used optimization algorithms [62]. The development of GA was inspired by natural selection. Compared with other optimization algorithms, GA is able to solve the optimization problems with unknown geometry of the searching space [63]. Therefore, this study tunes the hyperparameters of BP, SVM, DT and KNN using GA (Table 4 and Fig. 3).

To apply GA, a chromosome is used to represent the hyperparameters. Each gene (bit) of the chromosome is represented by 1 or 0. The length of the bit strings depends on the required calculation precision and scope. The initial values of the hyperparameters are represented by a randomly generated initial population. The hyperparameters are then tuned on the training set (including 70% of the instances). The AUC metric is selected as the fitness value of GA. Larger AUC means better fitness value. After selection, crossover and mutation operation, the offspring of the population is generated.

In the selection procedure, the roulette wheel method is used to select the chromosomes with higher AUC values for yielding offspring. Selection influences significantly the convergence of GA. Roulette wheel selection is the mostly frequently used selection strategy due to its simplicity in implementation [64]. There are other selection strategies used in GA such as the stochastic universal selection and stochastic remainder [65]. These selection operators cause populations of large variability with different statistical properties. Another selection method called adaptive selection method ranks or orders individuals rather than uses fitness of individuals [66]. It is difficult to choose the

Table 4 Hyperparameters of the four classifiers

Classifier	Hyperparameter	Empirical scope	Initial value
BP	Number of hidden layers (layerNum)	[1, 4]	{1, 2, 3, 4}
	Number of neurons in each hidden layer (neuronNum)	[1, 20]	10
SVM	Coefficient of the penalty term (c)	[0.1,1000]	16
	Gamma value of Gaussian kernel (γ)	[0.001,100]	16
DT	The minimum number of samples required to split an internal node(min_samples_split)	[1, 10]	25
	The minimum number of samples required to be at a leaf node (min_samples_leaf)	[2, 10]	50
KNN	Number of neighbour samples (neighbor_num)	[1, 10]	30

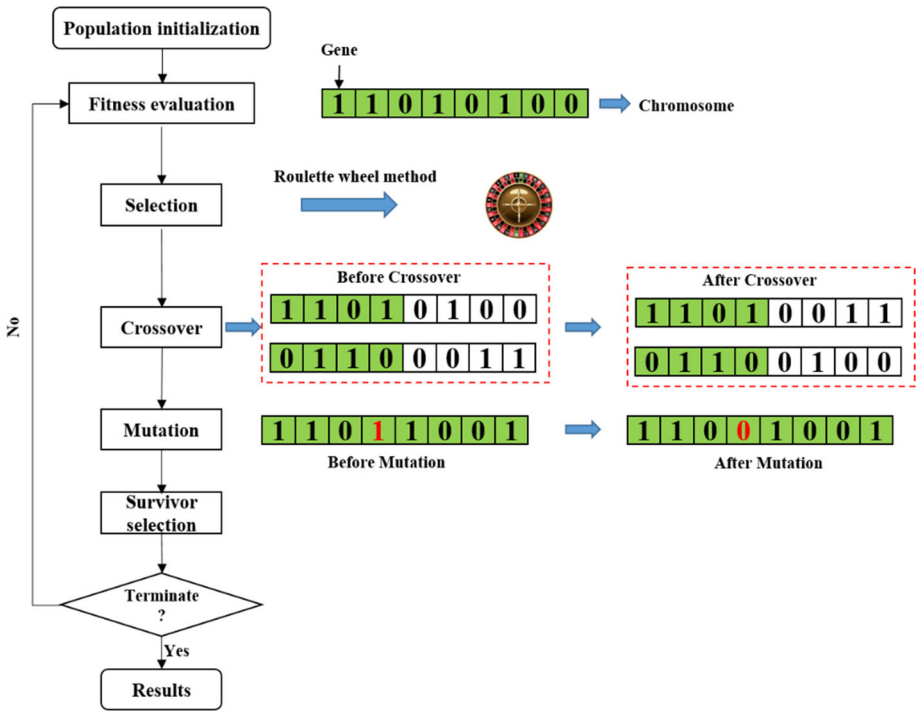

Fig. 3 Flow chart of the genetic algorithm

best selection operators. Although some researches indicate that scaling-related issues may limit the use of fitness-based methods [67], other works alleviate this difficulty [68]. Since the classifier ensemble proposed in this study is used to solve engineering problems, simplicity of implementation and straightforward interpretation are important to engineers, and hence, the roulette wheel method is selected in this study.

In the crossover procedure, the genes between two chromosomes are exchanged at a random point using single-point crossover principle. Researchers have developed many types of crossover strategies, including single-point crossover, multi-point crossover and uniform crossover [69]. In comparison with single-point crossover, multi-point crossover may reduce the quality of parents with good fitness and produce worse offspring, while uniform crossover will create genomes that will be very different from their parents if their parents are not similar. In addition, single-point crossover is easy to implement and therefore is used in this study. It should be noted that it is difficult to conclude that a crossover strategy is better than another one because most of the comparisons between the different types were conducted on a small group of test problems and more trial and error was needed [70]. The future work will focus on the development of new crossover operators to improve the performance of GA. GA has a mating probability (also known as crossover rate). Crossover rate represents the number of times a crossover occurs for chromosomes in one generation, i.e. the chance that two chromosomes exchange some of their parts. Every two individuals produce two new individuals through mating, replacing the original "old" individuals, while the

individuals who do not mate remain unchanged. If the crossover rate is 0.8, 80% of the "married couples" will have offspring. 100% crossover rate means that all offspring are made by crossover. If it is 0%, then the complete new generation of individuals is to be exactly copied from the older population, except those resulted from the mutation process. In this study, the crossover rate is selected to be 0.8 according to the recommendation of the previous literature [71, 72].

In the mutation procedure, the binary code of a gene is altered from 0 to 1 or vice versa at a small probability (0.1). Finally, part of the old population is replaced with the new offspring after the three operations. The evolutionary process iterates until the maximum iteration number (50) is reached (see Fig. 3). After tenfold, the highest AUC value is selected out, which corresponds to the best hyperparameters of a classifier. The hyperparameter tuning process of base classifiers using GA is shown in Fig. 4.

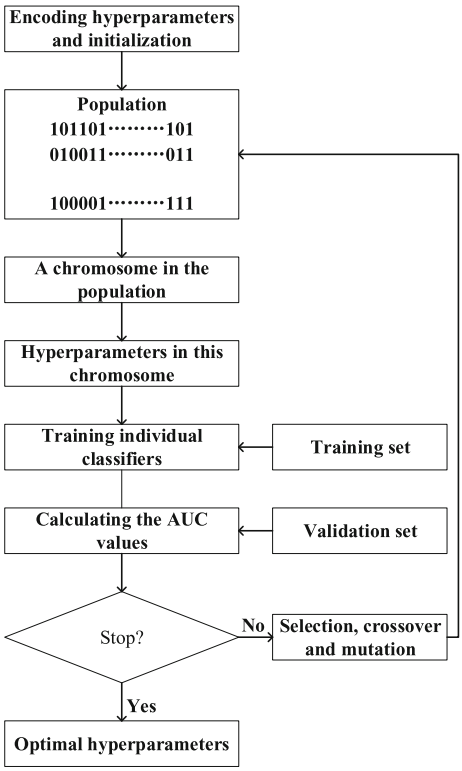

Fig. 4 Optimizing hyperparameters of individual classifiers using GA

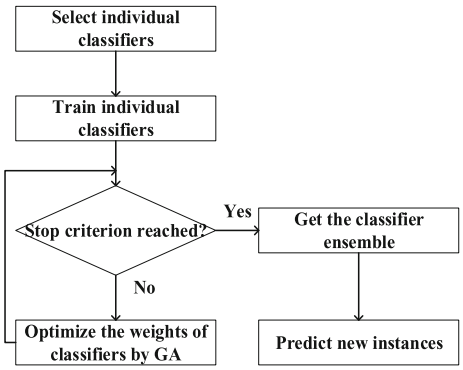

Fig. 5 Whole procedure of classifier ensemble construction

Table 5 Tuning hyperparameter of SVM using GA and tenfold CV

Fold	Hyperparameters		AUC	
	C	γ	Validation set	Test set
Dataset 1				
1	20.84	10.66	0.8909	0.8616
2	46.30	44.60	0.9058	0.8921
3	26.14	37.50	0.8833	0.8601
4	**23.29**	**2.45**	**0.9578**	**0.9173**
5	8.92	17.89	0.8714	0.8511
6	8.16	23.29	0.8974	0.8639
7	30.76	28.87	0.9097	0.8598
8	23.78	48.89	0.8972	0.8676
9	29.22	5.75	0.9054	0.8605
10	28.18	40.67	0.9481	0.9094
Dataset 2				
1	16.96	0.18	0.9588	0.9447
2	17.12	34.68	0.9473	0.9325
3	2.03	42.91	0.9739	0.9586
4	15.24	41.47	0.9312	0.9131
5	19.27	3.67	0.9267	0.9165
6	**12.50**	**53.00**	**0.9982**	**0.9851**
7	20.54	27.71	0.9398	0.9211
8	15.87	24.38	0.9771	0.9639
9	13.09	5.95	0.9626	0.9520
10	38.40	13.89	0.9804	0.9672
Dataset 3				
1	21.78	37.81	0.9640	0.9351
2	33.18	4.87	0.9209	0.9031
3	49.44	34.94	0.9383	0.9241
4	19.63	19.06	0.9734	0.9480
5	37.50	5.97	0.9774	0.9530
6	35.12	17.29	0.9363	0.9186
7	16.62	0.48	0.9728	0.9443
8	1.95	45.22	0.9338	0.9052
9	38.92	39.49	0.9674	0.9519
10	**1.33**	**1.76**	**0.9837**	**0.9659**

The best values of hyperparameters and the corresponding AUC values on the validation and test sets are typeset in bold

3.4 Constructing classifier ensemble using the weighted voting method

The classifier ensemble is constructed by aggregating seven base classifiers using the weighted voting method. In

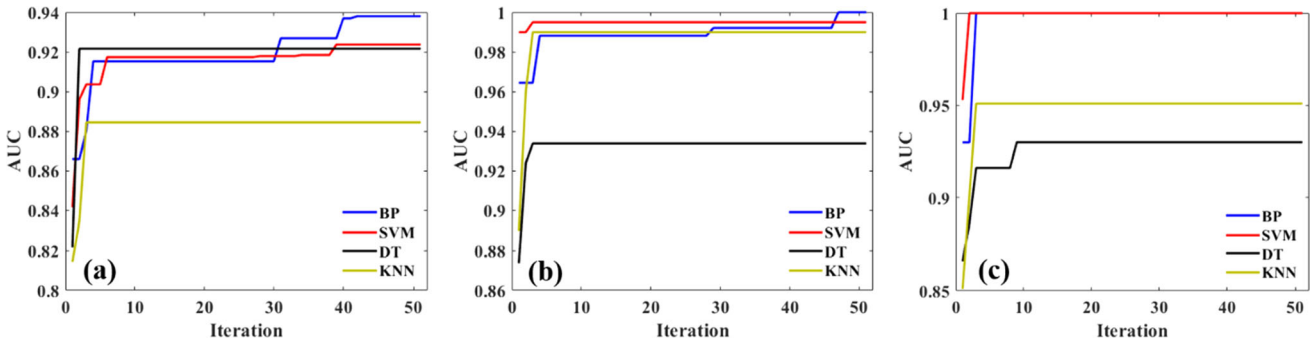

Fig. 6 Hyperparameter tuning using GA on Dataset 1 (**a**), Dataset 2 (**b**) and Dataset 3 (**c**)

Table 6 Obtained optimal hyperparameters for the four classifiers

Dataset	Hyperparameters						
	BP		SVM		DT		KNN
	Layer_num	Neuron_num	C	γ	Min_samples_split	Min_samples_leaf	Neighbor_num
1	1	9	23.29	2.45	8	3	4
2	1	12	12.5	53	4	3	3
3	1	12	1.33	1.76	7	3	5

this method, weights are given to each classier according to their confidence levels [73]. The weight values of the base classifiers are tuned using GA as follows:

$$\text{AUC} = \arg\max \sum_{i=1}^{n} (\text{AUC}_i \times \omega_i) \tag{19}$$

where ω_i denotes the weight value of a classifier and AUC_i is the AUC value of this classifier on the test set. The weight values are represented by the chromosome in GA, and the evolution process is the same as the hyperparameter tuning process mentioned in Sect. 3.3. The whole procedure of classifier ensemble construction is demonstrated in Fig. 5.

4 Results and discussion

4.1 Results of hyperparameter and weight tuning

As stated before, we have tuned the hyperparameters of four classifiers: BP, SVM, DT and KNN using GA and tenfold CV. The GA procedure is performed at each fold on the training set. The AUC value is selected as the performance metric to evaluate the tuning effects on the validation set. After convergence, the set of hyperparameters is selected out at this fold. Here, we only take the SVM hyperparameter tuning as an example. In each dataset, ten sets of hyperparameters (C and γ) are obtained, as shown in Table 5. The best values of hyperparameters and the corresponding AUC values on the validation and test sets are typeset in bold. For each dataset, we select the hyperparameters that correspond to the highest AUC value on the validation set as the optimal hyperparameters for further study (the hyperparameters at Fold 4 in Dataset 1, Fold 6 in Dataset 2 and Fold 10 in Dataset 3). We have also plotted the convergence versus iteration curve for the best fold, as shown in Fig. 6. It can be seen that all the AUC curves converge within 50 iterations (for several curves, within 10 iterations). This indicates that GA is very efficient in tuning hyperparameters. In addition, the significant increase in AUC shows that GA can find optimal hyperparameters for the base classifiers. The obtained hyperparameters of different base classifiers are summarized in Table 6.

After the base classifiers with optimal hyperparameters are obtained, the weights of the optimal base classifiers are also tuned using GA, as shown in Fig. 7. It can be observed that all the AUC values reach the maximum within 10 iterations, indicating GA is also efficient in weight tuning. The optimal weights for different base classifiers are summarized in Table 7.

4.2 Performance of the classifier ensemble and base classifiers

Table 8 summarizes the prediction performance of different base classifiers and the classifier ensemble in terms of ACC, K, P, R and F1 on the test sets of the three datasets.

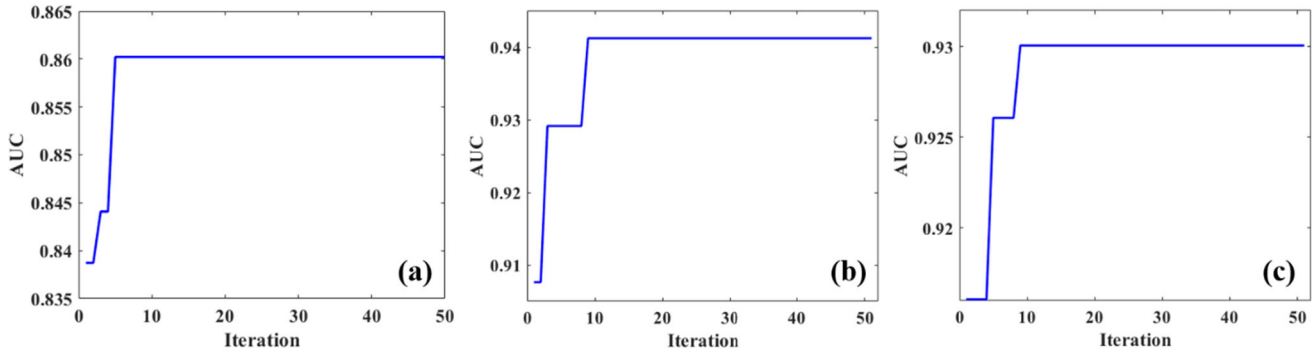

Fig. 7 Weight tuning using GA on Dataset 1 (**a**), Dataset 2 (**b**) and Dataset 3 (**c**)

Table 7 Optimal weights of base classifiers for the three datasets

Dataset	Hyperparameters						
	BP	SVM	DT	KNN	NB	LR	MLR
1	0.19231	0.21154	0.076923	0.13462	0.28846	0.019231	0.076923
2	0.047619	0.2381	0.047619	0.2381	0.047619	0.14286	0.2381
3	0.14063	0.14063	0.25000	0.046875	0.14063	0.21875	0.0625

Table 8 Performance of classifiers on the test sets of the three datasets

Test set	Metric	Classifier							
		BP	SVM	DT	KNN	LR	MLR	NB	Ensemble
1	ACC	0.828	0.833	0.817	0.785	0.758	0.753	0.656	**0.860**
	K	0.650	0.660	0.632	0.558	0.509	0.497	0.335	**0.715**
	P	0.840	0.835	0.857	0.781	0.781	0.774	0.803	**0.861**
	R	0.856	0.875	0.808	0.856	0.788	0.788	0.510	**0.894**
	F1	0.848	0.854	0.832	0.817	0.785	0.781	0.624	**0.877**
2	ACC	0.897	0.927	0.838	0.912	**0.941**	0.897	0.8529	**0.941**
	K	0.791	0.843	0.661	0.810	0.874	0.773	0.6728	**0.878**
	P	0.781	0.885	0.750	0.880	**0.920**	0.909	0.8571	0.862
	R	**1.000**	0.920	0.840	0.880	0.920	0.800	0.7200	**1.000**
	F1	0.877	0.902	0.793	0.880	0.920	0.851	0.7826	**0.926**
3	ACC	0.923	0.885	0.846	0.904	0.904	0.865	0.808	**0.942**
	K	0.845	0.768	0.693	0.806	0.807	0.714	0.616	**0.884**
	P	0.897	0.862	0.880	0.844	0.867	0.853	0.840	**0.929**
	R	0.963	0.926	0.815	**1.000**	0.963	0.935	0.778	0.963
	F1	0.929	0.893	0.846	0.915	0.912	0.892	0.809	**0.945**

The best values of performance metrics for different models are typeset in bold

The best values of performance metrics for different models are typeset in bold.It can be seen that the classifier ensemble achieves the highest ACC values on all the three datasets (0.860, 0.941 and 0.942 on Dataset 1, Dataset 2 and Dataset 3, respectively). For the base classifier, the highest accuracy values on Datasets 1, 2 and 3 are obtained by SVM (0.833), LR (0.941) and BP (0.923), respectively. This indicates that the base classifier is not stable when predicting earthquake-induced soil liquefaction. A classifier that performs well on a dataset does not mean it is still satisfactory on another dataset. However, the classifier ensemble is more stable as it achieves the highest accuracy values on all the three datasets. A K value higher than 0.4 is preferable [74]. It can be seen all the K values obtained by the classifiers are larger than 0.4 except that obtained by NB on Dataset 1 (0.335). The highest Kappa values on the three datasets are also obtained by the classifier ensemble. Although in Dataset 2, the highest P value is achieved by

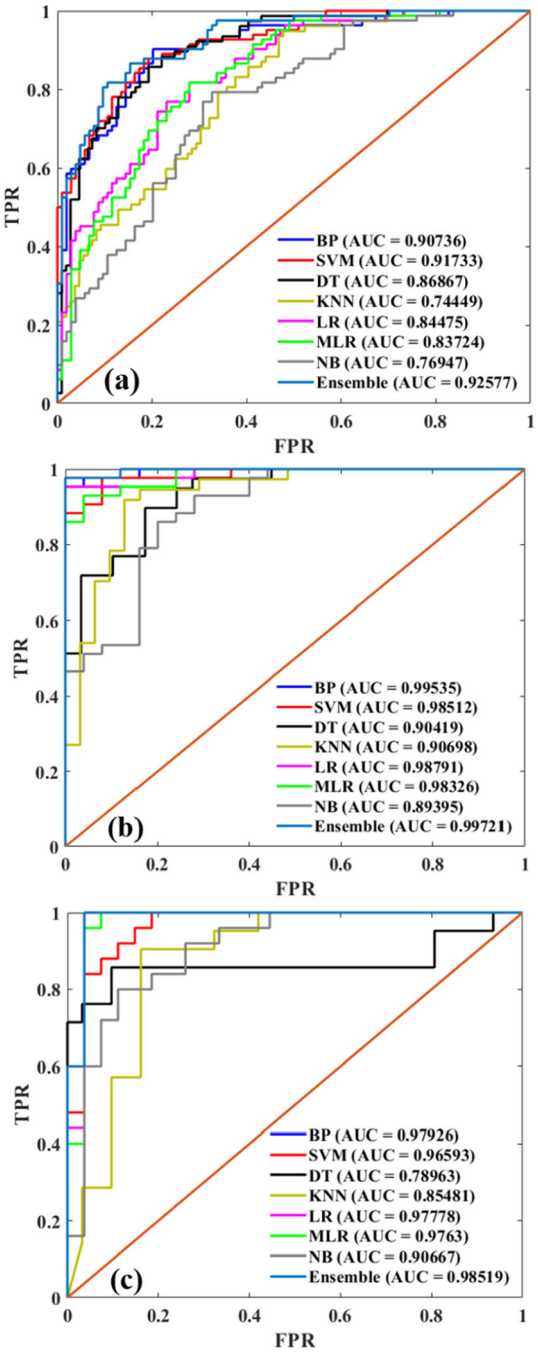

Fig. 8 ROC curve and AUC value obtained by base classifiers and the classifier ensemble on Dataset 1 (**a**); Dataset 2 (**b**) and Dataset 3 (**c**)

LR (0.92) and in Dataset 3, the highest R value is achieved by KNN (1.00), the highest F1 values derived by P and R are achieved by the classifier ensemble in all the three datasets (0.877, 0.926 and 0.945, respectively).

Figure 8 shows the ROC curve and AUC values obtained by the seven base classifiers and the classifier ensemble on the test sets of the three datasets. It can be seen that the classifier ensemble achieves the highest AUC values on the three datasets (0.92577, 0.99721 and 0.98519, respectively), while SVM on Dataset 1, BP on Datasets 2 and 3 also have good performance with pretty high AUC values of 0.91733, 0.99535 and 0.97926, respectively. From the ROC curve, it can be seen that the curve of the classifier ensemble dominates those of the other base classifiers for all the three datasets, indicating the classifier ensemble has the best performance in terms of ROC and AUC.

4.3 Variable importance

The variable importance is calculated using the random forest (RF) algorithm as follows [75]. The out-of-bag sample for a tree t is defined as OOB_t and the misclassification rate is denoted by errOOB_t. The influencing variables are permuted randomly in OOB_t to obtain a permuted sample $O\tilde{O}B_t$, and the error of predictor t $\mathrm{err}O\tilde{O}B_t$ is calculated. The variable importance (VI) of X can be written as

$$\mathrm{VI}(X)\frac{1}{\mathrm{ntree}}\sum_{t}\left(\mathrm{err}O\tilde{O}B_t-\mathrm{errOOB}_t\right) \tag{20}$$

where ntree is the number of tress in the forest.

The obtained results of variable importance are demonstrated in Fig. 9. From this figure, in Dataset 1, the $(N_1)_{60}$ (corrected standard penetration blow numbers) is the most sensitive influencing variable to soil liquefaction with an influencing score (IF) of 3.2930, slightly higher than $\emptyset'$ (internal friction angle of soil) with IF being 3.1403. The influencing scores of M_v (earthquake moment magnitude), V_s (shear wave velocity) and d_w (depth of ground water table) are relatively low. (IFs are 0.2075, 0.2600 and 0.3334, respectively.) The less sensitive influencing variables to the outputs can be deleted in model training. As for Dataset 2, q_c (the cone tip resistance) has the highest IF (3.5553), followed by a_{max} (the maximum horizontal ground surface acceleration) with an IF of 2.4446, while the $\sigma_v^{'}$ (effective overburden stress) achieves the lowest IF (0.4277). In Dataset 3, the highest and second highest influencing scores are obtained by a_{max} (2.2303) and V_s (1.3644), respectively, while the lowest and second lowest scores are obtained by $\sigma_v^{'}$ (0.5084) and σ_v (0.5165), respectively. From the variable importance calculated based on the three datasets, we can know that $(N_1)_{60}$, $\emptyset'$, q_c, a_{max} and V_s are very sensitive to the soil liquefaction prediction. These variables should be given priority when constructing soil liquefaction datasets in the future work.

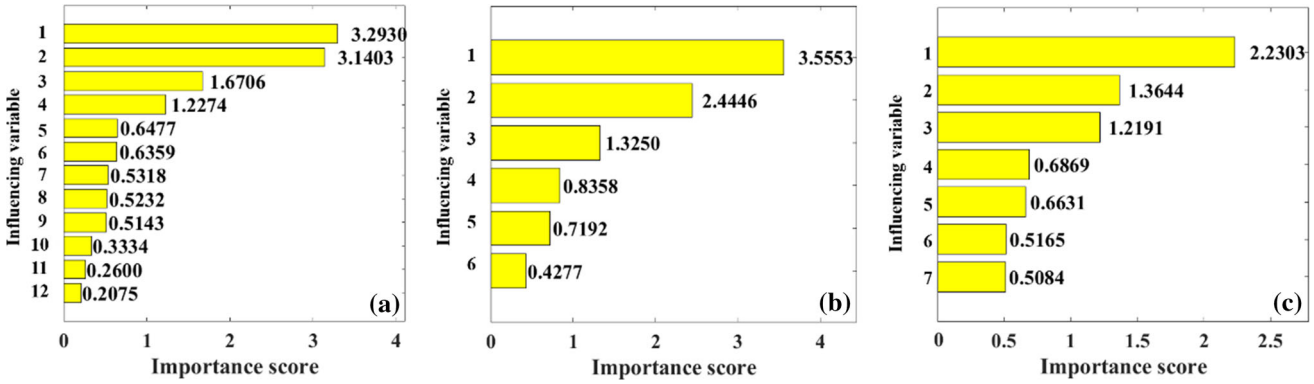

Fig. 9 Variable importance for Dataset 1 (**a**), Dataset 2 (**b**) and Dataset 3 (**c**). In Dataset 1, the numbers from 1 to 12 represent F75, $\emptyset'$, CSR, $(N_1)_{60}$, σ_v, D, σ_v', a_{max}, a_t, d_w, V_s and M_v, respectively. In Dataset 2, the numbers from 1 to 6 represent q_c, a_{max}, R_f, M_v, σ_v and σ_v', respectively. In Dataset 3, the numbers from 1 to 7 represent a_{max}, V_s, M_v, SC, D, σ_v and σ_v', respectively

Table 9 Performance comparison of the proposed model with models reported in the literature

References	Model	ACC on test set
Dataset 1		
Hoang and Bui [76]	KFDA-LSSVM	84.95
Gandomi et al. [77]	CHAID-SPT	86.7
Zhou et al. [78]	Gradient boosting	88.62
This study	Classifier ensemble	92.6
Dataset 2		
Hoang and Bui [76]	KFDA-LSSVM	93.19
Zhou et al. [78]	Gradient boosting	95.45
Muduli and Das [79]	Genetic programming	97
Goh and Goh [38]	SVM	98
This study	Classifier ensemble	99.7
Dataset 3		
Juang and Chen [39]	ANN	88
Muduli et al. [80]	ELM	88
Hoang and Bui [76]	KFDA-LSSVM	88.56
This study	Classifier ensemble	98.5

4.4 Performance compassion of the proposed model with existing models

Figure 8 shows the accuracy metric on the test sets for the proposed model in this study and the existing models in the literature. It can be seen that the proposed classifier ensemble outperforms the models in the literature, indicated by the highest accuracy metric which improves by 4.5%, 1.7% and 11.2%, respectively. Therefore, the comparison results confirm that the prediction capability of the proposed classifier ensemble is more accurate and reliable in comparison with other predictive models (Table 9).

5 Conclusion

This study proposes a hybrid classifier ensemble which improves the prediction accuracy of earthquake-induced soil liquefaction. This model is established by aggregating seven base classifiers using the weighted voting method. In the model construction phase, GA is applied to search for optimal hyperparameters of base classifiers and weights of the base classifiers for constructing the classifier ensemble. The fast convergence speed and large AUC increase indicate the high efficiency of GA in hyperparameter and weight tuning.

By comparing the performance of the classifier ensemble with those of different base classifiers, we confirm that the proposed classifier ensemble outperforms the base classifiers in terms of a variety of metrics including ACC, K, P, R, F1, AUC and ROC. In addition, the classifier ensemble has addressed the problem that base classifiers are not stable when predicting soil liquefaction using different datasets.

The importance of the influencing variable to the outputs on the three datasets indicates that several predictors are more sensitive such as $(N_1)_{60}$, $\emptyset'$, q_c, a_{max} and V_s. These variables should be given higher priority when enlarging the soil liquefaction datasets to improve the generalization capability of the proposed model in the future. This proposed classifier ensemble can be applied to solving other classification problems in civil engineering. Other robust base classifiers can be also considered to be incorporated in the classifier ensemble for improving its prediction performance.

Acknowledgements The first author is supported by the China Scholarship Council (Grant Number: 201706460008).

Compliance with ethical standards

Conflict of interest All the authors declare that there is no conflict of interests regarding the publication of this article.

References

1. Lashkari A, Karimi A, Fakharian K, Kaviani-Hamedani F (2017) Prediction of undrained behavior of isotropically and anisotropically consolidated Firoozkuh sand: instability and flow liquefaction. Int J Geomech 17:04017083
2. Dobry R, Abdoun T (2017) Recent findings on liquefaction triggering in clean and silty sands during earthquakes. J Geotech Geoenviron Eng 143:04017077
3. Hazout L, Zitouni ZE-A, Belkhatir M, Schanz T (2017) Evaluation of static liquefaction characteristics of saturated loose sand through the mean grain size and extreme grain sizes. Geotech Geol Eng 35:2079–2105
4. Shivaprakash B, Dinesh S (2017) Dynamic properties of sand–fines mixtures. Geotech Geol Eng 35:2327–2337
5. Johari A, Pour JR, Javadi A (2015) Reliability analysis of static liquefaction of loose sand using the random finite element method. Eng Comput 32:2100–2119
6. Huang S, Huang M, Lyu Y (2019) A novel approach for sand liquefaction prediction via local mean-based pseudo nearest neighbor algorithm and its engineering application. Adv Eng Inform 41:100918
7. Sun Y, Li G, Zhang J et al (2020) Experimental and numerical investigation on a novel support system for controlling roadway deformation in underground coal mines. Energy Sci Eng, 8(2):490–500
8. Yazdi J, Moss R (2016) Nonparametric liquefaction triggering and postliquefaction deformations. J Geotech Geoenviron Eng 143:04016105
9. Andrus RD, Stokoe KH II (2000) Liquefaction resistance of soils from shear-wave velocity. J Geotech Geoenviron Eng 126:1015–1025
10. Wu J, Seed RB (2004) Estimation of liquefaction-induced ground settlement (case studies). In: Proceedings of the 5th international conference on case histories in geotechnical engineering, vol 6. Springer
11. Tsaparli V, Kontoe S, Taborda DM, Potts DM (2017) An energy-based interpretation of sand liquefaction due to vertical ground motion. Comput Geotech 90:1–13
12. Gao X, Sun Q, Xu H, Gao J (2020) Sparse and collaborative representation based kernel pairwise linear regression for image set classification. Expert Syst Appl 140:112886
13. Gao J, Li L (2019) A robust geometric mean-based subspace discriminant analysis feature extraction approach for image set classification. Optik 199:163368
14. Gao J, Li L, Guo B (2020) A new extend face representation method for face recognition. Neural Process Lett 51:473–486
15. Gao X, Sun Q, Xu H, Wei D, Gao J (2019) Multi-model fusion metric learning for image set classification. Knowl Based Syst 164:253–264
16. Hanna AM, Ural D, Saygili G (2007) Neural network model for liquefaction potential in soil deposits using Turkey and Taiwan earthquake data. Soil Dyn Earthq Eng 27:521–540
17. Chern S-G, Lee C-Y, Wang C-C (2008) CPT-based liquefaction assessment by using fuzzy-neural network. J Mar Sci Technol 16:139–148
18. Kayadelen C (2011) Soil liquefaction modeling by genetic expression programming and neuro-fuzzy. Expert Syst Appl 38:4080–4087
19. Xue X, Yang X (2016) Seismic liquefaction potential assessed by support vector machines approaches. Bull Eng Geol Env 75:153–162
20. Hu J-L, Tang X-W, Qiu J-N (2016) Assessment of seismic liquefaction potential based on Bayesian network constructed from domain knowledge and history data. Soil Dyn Earthq Eng 89:49–60
21. Ardakani A, Kohestani V (2015) Evaluation of liquefaction potential based on CPT results using C4.5 decision tree. J AI Data Min 3:85–92
22. Kohestani V, Hassanlourad M, Ardakani A (2015) Evaluation of liquefaction potential based on CPT data using random forest. Nat Hazards 79:1079–1089
23. Zhang W, Goh AT, Zhang Y, Chen Y, Xiao Y (2015) Assessment of soil liquefaction based on capacity energy concept and multivariate adaptive regression splines. Eng Geol 188:29–37
24. Xu H, Caramanis C, Mannor S (2011) Sparse algorithms are not stable: a no-free-lunch theorem. IEEE Trans Pattern Anal Mach Intell 34:187–193
25. Zhou Z-H, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all. Artif Intell 137:239–263
26. Zhang J, Li D, Wang Y (2020) Predicting tunnel squeezing using a hybrid classifier ensemble with incomplete data. Bull Eng Geol Environ. https://doi.org/10.1007/s10064-020-01747-5
27. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55:119–139
28. Breiman L (1996) Bagging predictors. Mach Learn 24:123–140
29. Wolpert DH (1992) Stacked generalization. Neural Netw 5:241–259
30. Kuncheva LI, Rodríguez JJ (2014) A weighted voting framework for classifiers ensembles. Knowl Inf Syst 38:259–275
31. Werbin-Ofir H, Dery L, Shmueli E (2019) Beyond majority: label ranking ensembles based on voting rules. Expert Syst Appl 136:50–61
32. Ekbal A, Saha S (2011) A multiobjective simulated annealing approach for classifier ensemble: named entity recognition in Indian languages as case studies. Expert Syst Appl 38:14760–14772
33. Ekbal A, Saha S (2011) Weighted vote-based classifier ensemble for named entity recognition: a genetic algorithm-based approach. ACM Trans Asian Lang Inf Process 10:9
34. Kim H, Kim H, Moon H, Ahn H (2011) A weight-adjusted voting algorithm for ensembles of classifiers. J Korean Stat Soc 40:437–449
35. Yang XS (2010) Nature-inspired metaheuristic algorithms. Luniver Press
36. Zhang J, Huang Y, Wang Y et al (2020) Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms. Constr Build Mater 253:119208
37. Zhang J, Huang Y, Ma G et al (2020) Multi-objective beetle antennae search algorithm. arXiv:2002.10090
38. Goh AT, Goh S (2007) Support vector machines: their use in geotechnical engineering as illustrated using seismic liquefaction data. Comput Geotech 34:410–421
39. Juang CH, Chen CJ (2000) A rational method for development of limit state for liquefaction evaluation based on shear wave velocity measurements. Int J Numer Anal Methods Geomech 24:1–27
40. Koo TK, Li MY (2016) A guideline of selecting and reporting intraclass correlation coefficients for reliability research. J Chiropractic Med 15:155–163
41. Shalev-Shwartz S, Ben-David S (2014) Understanding machine learning: From theory to algorithms. Cambridge University Press, Cambridge

42. Flach P (2012) Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press, Cambridge
43. Hecht-Nielsen R (1992) Theory of the backpropagation neural network, neural networks for perception. Elsevier, Amsterdam, pp 65–93
44. Sun Y, Zhang J, Li G et al (2019) Optimized neural network using beetle antennae search for predicting the unconfined compressive strength of jet grouting coalcretes. Int J Numer Anal Methods Geomech 43(4):801–813
45. Scholkopf B, Smola AJ (2001) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge
46. Zhang J, Huang Y, Ma G et al (2020) A metaheuristic-optimized multi-output model for predicting multiple properties of pervious concrete. Constr Build Mater 249:118803
47. Sun Y, Zhang J, Li G et al (2019) Determination of young's modulus of jet grouted coalcretes using an intelligent model. Eng Geol 252:43–53
48. Sun J, Zhang J, Gu Y et al (2019) Prediction of permeability and unconfined compressive strength of pervious concrete using evolved support vector regression. Constr Build Mater 207:440–449
49. Quinlan JR (2014) C4.5: programs for machine learning. Elsevier, Amsterdam
50. Breiman L (2017) Classification and regression trees. Routledge, Abingdon
51. Zhang J, Ma G, Huang Y et al (2019) Modelling uniaxial compressive strength of lightweight self-compacting concrete using random forest regression. Constr Build Mater 210:713–719
52. Zhang J, Li D, Wang Y (2020) Predicting uniaxial compressive strength of oil palm shell concrete using a hybrid artificial intelligence model. J Build Eng 30:101282
53. Zhang J, Li D, Wang Y (2020) Toward intelligent construction: prediction of mechanical properties of manufactured-sand concrete using tree-based models. J Clean Prod 258:120665
54. Cunningham P, Delany SJ (2007) k-Nearest neighbour classifiers. Mult Classif Syst 34:1–17
55. Berger JO (2013) Statistical decision theory and Bayesian analysis. Springer, Berlin
56. Menard S (2002) Applied logistic regression analysis. Sage, Thousand Oaks
57. Kutner MH, Nachtsheim CJ, Neter J, Li W (2005) Applied linear statistical models. McGraw-Hill, Boston
58. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manag 45:427–437
59. Kraemer HC (2014) Kappa coefficient, Wiley StatsRef: Statistics Reference Online, pp 1–4
60. Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27:861–874
61. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. IJCAI, Montreal, pp 1137–1145
62. Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Amsterdam
63. Schmitt LM (2001) Theory of genetic algorithms. Theoret Comput Sci 259:1–61
64. Goldberg DE (2006) Genetic algorithms. Pearson Education India, Bengaluru
65. Goldberg DE, Deb K (1991) A comparative analysis of selection schemes used in genetic algorithms, foundations of genetic algorithms. Elsevier, Amsterdam, pp 69–93
66. Baker JE (1985) Adaptive selection methods for genetic algorithms. In: Proceedings of an international conference on genetic algorithms and their applications. Hillsdale, New Jersey, pp 101–111
67. Hancock PJ (1994) An empirical comparison of selection methods in evolutionary algorithms. AISB workshop on evolutionary computing. Springer, pp 80–94
68. Gupta S (2009) Relative fitness scaling for improving efficiency of proportionate selection in genetic algorithms. In: Proceedings of the 11th annual conference companion on genetic and evolutionary computation conference: late breaking papers, 2009, pp 2741–2744
69. Vose MD (1999) The simple genetic algorithm: foundations and theory. MIT Press, Cambridge
70. Hassanat AB, Alkafaween EA (2017) On enhancing genetic algorithms using new crossovers. Int J Comput Appl Technol 55:202–212
71. Hassanat A, Almohammadi K, Alkafaween E, Abunawas E, Hammouri A, Prasath V (2019) Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. Information 10:390
72. Segura C, Coello CAC, Segredo E, Aguirre AH (2015) A novel diversity-based replacement strategy for evolutionary algorithms. IEEE Trans Cybern 46:3233–3246
73. Tóth N, Pataki B (2008) Classification confidence weighted majority voting using decision tree classifiers. Int J Intell Comput Cybern 1:169–192
74. Landis JR, Koch GG (1977) The measurement of observer agreement for categorical data. Biometrics 33:159–174
75. Genuer R, Poggi J-M, Tuleau-Malot C (2010) Variable selection using random forests. Pattern Recogn Lett 31:2225–2236
76. Hoang N-D, Bui DT (2018) Predicting earthquake-induced soil liquefaction based on a hybridization of kernel Fisher discriminant analysis and a least squares support vector machine: a multi-dataset study. Bull Eng Geol Environ 77:191–204
77. Gandomi AH, Fridline MM, Roke DA (2013) Decision tree approach for soil liquefaction assessment. Sci World J 2013:346285
78. Zhou J, Li E, Wang M, Chen X, Shi X, Jiang L (2019) Feasibility of stochastic gradient boosting approach for evaluating seismic liquefaction potential based on SPT and CPT case histories. J Perform Constr Facil 33:04019024
79. Muduli PK, Das SK (2014) CPT-based seismic liquefaction potential evaluation using multi-gene genetic programming approach. Indian Geotech J 44:86–93
80. Muduli PK, Das SK (2015) Evaluation of liquefaction potential of soil based on shear wave velocity using multi-gene genetic programming, handbook of genetic programming applications. Springer, Berlin, pp 309–343