Contents lists available at SCCE

Journal of Soft Computing in Civil Engineering

Journal homepage: www.jsoftcivil.com

Use of ANN, C4.5 and Random Forest Algorithm in the Evaluation of Seismic Soil Liquefaction

Ravi Bhushan Bhardwaj[1*], S.R. Chaurasia[2]

1. M.Tech Student, Department of Civil Engineering, Madan Mohan Malaviya University of Technology, Gorakhpur, India
2. Professor, Department of Civil Engineering, Madan Mohan Malaviya University of Technology, Gorakhpur, India

Corresponding author: *2019013414@mmmut.ac.in*

https://doi.org/10.22115/SCCE.2022.314762.1380

ARTICLE INFO

Article history:
Received: 12 November 2021
Revised: 06 April 2022
Accepted: 08 April 2022

Keywords:
Resilient back propagation (Rprop);
C4.5 Algorithm;
Random forest algorithm;
Liquefaction.

ABSTRACT

Liquefaction is one of the disasters caused due to earthquake. In 1999, Chi-Chi, Taiwan, earthquake is an example of liquefaction prone disasters induced due to M_w 7.6 earthquake. This becomes major cause for prediction of the liquefaction in the soil with respect to geotechnical property. In this paper, we have use Artificial Neural Networks (ANN) model based on Resilient Back propagation (Rprop), Decision tree model (DT) and classifier are C 4.5 and Random Forest is done for comparing the performance and evaluation of liquefaction potential based on the obtained field CPT data (Juang et al. [1]) consisting 125 datasets over the simplified procedures that are being traditionally use for the classification of liquefaction of the soil by different researchers. It is observe that Resilient Back propagation Algorithm prediction is 100% whereas C 4.5 algorithm and Random forest Algorithm are 97.6% and 98.4% accurate for the evaluation of seismic soil liquefaction potential.

1. Introduction

Liquefaction is a phenomenon, the shear strength of soil becomes too low or zero that makes the soil is unable to support the structures (Kramer 1996). This causes failure of civil engineering

How to cite this article: Bhardwaj RB, Chaurasia SR. Use of ANN, C4.5 and random forest algorithm in the evaluation of seismic soil liquefaction. J Soft Comput Civ Eng 2022;6(2):94–113. https://doi.org/10.22115/scce.2022.314762.1380

structures during the earthquake and loss of lives and property [1–5]. Hence, this becomes important to monitor the liquefaction in the seismic zone area. Simplified procedures based on empirical and semi-empirical [6] are used to assess liquefaction potential hazards. Many of among them are based on extension of the 'simplified procedure' developed by Seed and Idris [4]. Methods use are based on the in-situ data. One of the liquefaction evaluations based on in-situ CPT (Cone Penetration Test) data methods used by Juang et al. and youd are widely used by different researchers in their research work. CPT test offers clear observation of subsurface profile of soil strata and penetration resistance than other in-situ test.

Artificial Neural Networks (ANN) models gain a lot of popularity [1–5] in the mid early of 2011 and successfully proved with promising results [5] but with the ANN machine learning methods 'Black Box' is a problem in the research [7–9] because 'Black Box' term in the ANN machine learning approaches means that we usually ignore how we are getting results by giving input and getting output. Simply we don't get idea what actually is happening in the process with inputs which is the limitation with all ANN studies. But also the results obtained by the ANN are quite good and to increase the accuracy for the model Resilient Back Propagation (Rprop) is used in this paper. To best of our knowledge, it is believe that Rprop machine learning used by the authors in this paper is first of its own kind in the assessment of the liquefaction analysis of soil. Also some other methods like support vector machine (SVM) [10,11]., patient rule induction method (PRIM) [12] Bayesian belief Network [13] have also help in the prediction of the soil liquefaction. Resilient back propagation (Rprop) algorithm is use due to its advantage of direct adaptation of weight based on local gradient information [8] to increase the accuracy. Also in this paper use of C 4.5 decision tree is the extended use of the seismic soil liquefaction potential [13] . Classifiers like; Random Forests gain popularity in the recent years [14] have proved promising results [1–5,13]. Random Forest Algorithm performed well if we compare to many other classifiers like Support Vector Machine(SVM) [1,10], C 4.5 decision tree [13].

After generating a correlation matrix between the input variables by the authors it is found that parameters like; Friction ratio R_f, peak horizontal acceleration (PGA) amax (g), vertical effective stress σ_{vo} (kPa), cone penetration resistance q_c, frictional resistance f_s are very important parameters in the use of liquefaction assessment of soil.

We have split the paper into different sections (1) Introduction, (2) Research Methodology, (3) Data Collection and Testing, (4) Conclusions. Section 2, deals with the methods and principle of algorithms used given by their inventers traditionally, Section 3 its data collection and testing, section 4 with the conclusions, scope of further study and limitations.

2. Research methodology

2.1. Working principle of algorithm

2.1.1. Artificial neural network (ANN)

In this paper, we have trained Artificial Neural Network (ANN) based on multi-layer perceptron (MLP) trained with Resilient back propagation (Rprop) algorithm [8] (Reidmiller) is used under the 'neuralnet' package provided by R language. The neuron's developed is having 5 inputs p =

(Rf, a_{max}, σ_{vo} , qc, fs) and weighted by elements (w_1, w_2, w_3, w_4, w_5) of the weight matrix W respectively with six hidden layers.

Perceptron was invented by Frank Rosenblatt. Perceptron is the building block of the neural network. The used perceptron architecture consists of input values, weights, bias and an activation function. A single perceptron with n-inputs, n-weights, bias and with activation function shown in the Fig 1.

In the process, all the weights are multiplied with inputs they are taken and sum up to create weighted sum and then applied to the activation function producing perceptron output. The activation function is use for surety that the output must lies between the recorded values (0,1) in the perceptron architecture. Weight is referred as the strength of a node. An inputs bias function shifts the curve up or down. There are different types of activation function. In our neural network model, we have used Logistic function for the hidden layers. This function is used to output a number from 0 to 1. Logistical functions have the formula:

$$logsig(x) = \frac{1}{1+\exp(-x)} \tag{1}$$

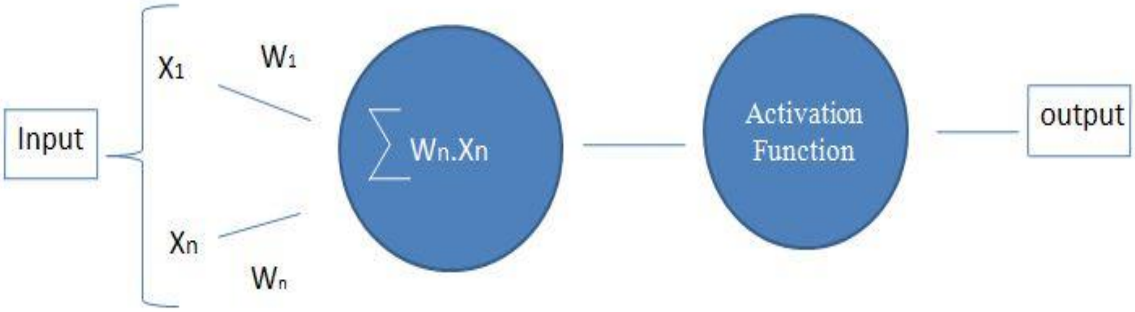


Fig. 1. A Single Perceptron.

2.1.2. Back propagation learning

Backpropagation algorithm is use for supervised learning with multi-layered feed-forward networks [8]. The algorithm forms a chain which is recall to calculate the effect of each weight in the network with respect of an arbitrary function (E):

$$\frac{\partial E}{\partial w_{ef}} = \frac{\partial E}{\partial s_e}\frac{\partial s_e}{\partial net_e}\frac{\partial net_e}{\partial w_{ef}} \tag{2}$$

Where, w_{ef} is the weight for the neuron f to neuron e, s_e the output, net_e the weighted sum of the input neuron's e. The partial derivative of each weight is called, minimizing error function obtained by performing simple gradient descent:

$$w_{ef}(t+1) = w_{ef}(t) - \epsilon_1 \frac{\partial E}{\partial w_{ef}}(t) \tag{3}$$

The rate of learning $\in_1$, which scales the derivative, has an important effect on the time needed until convergence is gained. If $\in_1$,is very small a number of many steps will require to obtain solution and a large learning rate will possibly lead to oscillation, preventing the error to fall below a definite value.

For solving the above mention problem, a momentum term is use-

$$\Delta w_{ef}(t) = -\in_1 \frac{\partial E}{\partial w_{ef}}(t) + \mu \Delta w_{ef}(t-1) \tag{4}$$

Where, μ is momentum parameter. It renders the learning process makes more stable and to stimulate convergence in shallow regions of the error function. But practice have shown that optimum value of the momentum parameter is equally problem dependent as the learning rate $\in_1$and hence no general progress is achieved. Various modifications have been proposed to the back propagation algorithm. RROP is one of the modifications to this algorithm to solve some adaptation problems.

Resilient backpropagation change the size of the weight-update Δw_{ef} directly, i.e. without taking the size of the partial derivative.

2.1.3. Resilient backpropagation (RPROP) algorithm

RPROP stands for Resilient backpropagation. This algorithm is use for supervise learning in the feed-forward ANN networks. This algorithm is of first-order optimization algorithm. This algorithm was invented by Martin Reidmiller and Heinrich Braun in 1992 [8]. It adopts a direct adaptation of the weight step depend on the local gradient information and its adaptation not fade by gradient behavior. For every particular weight single update value Δ_{ef}, this recognize the size of the weight update. The adaptive update value updates during the session of learning process based on its local sight on the error function (E), to the following learning-rule:

$$\begin{cases} ,\Delta_{ef}{}^{(t)} = \quad \eta^{+} * \Delta_{ef}{}^{(t-1)}, if\ \frac{\partial E}{\partial w_{ef}}^{(t-1)} * \frac{\partial E}{\partial w_{ef}}^{(t)} > 0 \\ \eta^{-} * \Delta_{ef}{}^{(t-1)}, if\ \frac{\partial E}{\partial w_{ef}}^{(t-1)} * \frac{\partial E}{\partial w_{ef}} < 0 \\ \Delta_{ef}{}^{(t-1)}, \text{else} \\ 0 < \eta^{-} < 1 < \eta^{+} \end{cases} \tag{5}$$

On every iteration, partial derivative of the weight changes its sign. The difference in sign indicates that the last update was very large and the algorithm has propped over a local minimum, the update-value Δ_{ef} reduced by the factor η^{-} . In addition, the sign if remains constant and does not change, the updated value get little bit increased in order to rate up the convergence in the surface of error.

Now update value of the each weight is taken and weight updated follow simple rule. If the derivative is positive this refers to the increase in error, the weight decreases by the update value whereas, if derivative is found negative the update value is added.

$$\begin{cases} \Delta w_{ef}{}^{(t)} = \quad -\Delta_{ef}{}^{(t)},\ if\ \frac{\partial E}{\partial w_{ef}}^{(t)} > 0 \\ +\Delta_{ef}{}^{(t)}, if\ \frac{\partial E}{\partial w_{ef}}^{(t)} < 0 \end{cases} \tag{6}$$

0, else

Further more:

$$w_{ef}{}^{(t+1)} = w_{ef}{}^{(t)} + \Delta w_{ef}{}^{(t)} \tag{7}$$

$w_{ef}{}^{(t)}$ is the weighting between e and f neurons in two successive layers on the iteration t, $w_{ef}{}^{(t+1)}$is the new weight.

2.2. Decision trees (DTs) and its working principle

Nature is one of the best teacher and we have lot of things to learn from nature. In this machine learning approach, we use both classifications and regression problems in our real life and liquefaction analysis is one of the best examples to understand this. Decision trees [9,13] are non-parametric supervised learning technique that uses tree like model. The model is able to predict the value of the aimed variable by learning simple decision rules work out from the data features. This model follows rules that are generally in form of if-then-else statements. As long we go at depths the tree becomes more complex the rules and best is the model.

2.2.1. Entropy

Let us consider the probability distribution, P equals to (p_1, p_2 ….p_n) and a sample S, the information taken over by this distribution, also called the entropy of P calculated as follows:

$$Entropie(P) = -\sum_{i=1}^{n} p_i \times \log p_i \tag{8}$$

2.2.2. The gain information

For all instances, the functions we use enable us to measure the degree of mixing classes and any position of the tree in construction. It stays to define a function to choose the test that must mark the current node. It states the gain for a test T and a position p as:

$$Gain(p,T) = Entropie(p) - \sum_{j=1}^{n}(p_j \times Entropie(p_j)) \tag{9}$$

Where, values (p_j) is the all possible set for attribute T. We can use this measure to rank attributes, construct the model, at each node is located the attribute with the highest information gain.

2.2.3. C 4.5 algorithm

C4.5 is the successor of the ID3 algorithm [9,14]. Not using ID3 algorithm in this paper is that it has a several disadvantages like if we use a small sample of data for testing there will be problem of data over-classification. Hence using C4.5 algorithm is a good choice over ID3 algorithm. The algorithm was given by Ross Quinlan (1993) to overcome the limitation of ID3 algorithm.

The algorithm uses "Gain Ratio" which is a modification of information gain. Gain Ratio given as follows:

$$GainRatio(p,T) = \frac{Gain(p,T)}{SplitInfo(p,T)} \tag{10}$$

Split Info is:

$$SplitInfo(p,test) = -\sum_{j=1}^{n} p'\left(\frac{j}{p}\right) \times \log\left(p'\left(\frac{j}{p}\right)\right) \tag{11}$$

the proportion of elements present at the position- p is p'(j/p), taking the value of j-th test.

C4.5 chooses one attribute of the data at each node of the tree that most effectively separate its set of samples into subsets full within one class or the other. Its eligibility based on the normalized information gain that is outcome from choosing an attribute for separating the data. The attribute with the high-normalized information gain chosen to make the decision.

2.2.4. Random forest algorithm

Random Forest Algorithm is very popular in machine learning [15] as it performed well if we compare to many other classifiers, including discriminant analysis, support vector machines (SVM) and some traditional neural networks (NN), and is powerful against overfitting (Breiman, 2001). Random Forest is a supervised machine learning technique and concept of ensemble learning. Ensemble learning combines variety of classifiers to solve a very complex problem and to upgrade the efficiency of the model. Classifier contains number trees on several subsets of datasets and takes the average to upgrade the predictive accuracy of that dataset.

Keeping the above straight forward simply, "instead of depending on single tree, algorithm gains the prediction from each tree and based on the higher number of votes prediction, it predicts the final result". As mentioned in above paragraph it is robust against overfitting , the number of trees i.e. "greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting". If there is any missing values of data it can handle very efficiently.

Random forest algorithm works in the two phase creation. In the first phase by aggregating N decision trees and in second stage predicting for every decision trees created in first phase. The working of the algorithm given below in following steps:

In the first step, the algorithm selects random K data points from the training set. Stepping to the next the algorithm construct the decision trees model related with the selected data points known as subsets. After that, it chooses the N number for decision trees that we want to construct. The process keeps on going by selecting the random K data points and constructing the decision tree with selected subsets. In the final step based on the higher votes for new data points, it finds the prediction for every decision tree and assigns new data points to the each section.

3. Data collection and testing

In this paper we have used actual field data of CPT-based liquefaction case history in 1999, Chi-Chi Taiwan earthquake is used (Juang et al., [1]) shown in Table 1 and compared the predicted results with the actual ones. The database have total 125 instances, 41 are liquefied and other 84 are non-liquefied based on the field test values.

For the Artificial Neural Network (ANN) testing and modeling we have used 'neuralnet' package provided in R language using R studio version 1.1.463. C 4.5 Algorithm and Random Forest Algorithm are designed and tested using the open source software "Waikato Environment for Knowledge Analysis (WEKA)", Java based programming language developed at the University of Waikato, New Zealand. This is open source and free to use software dedicated to the machine learning.

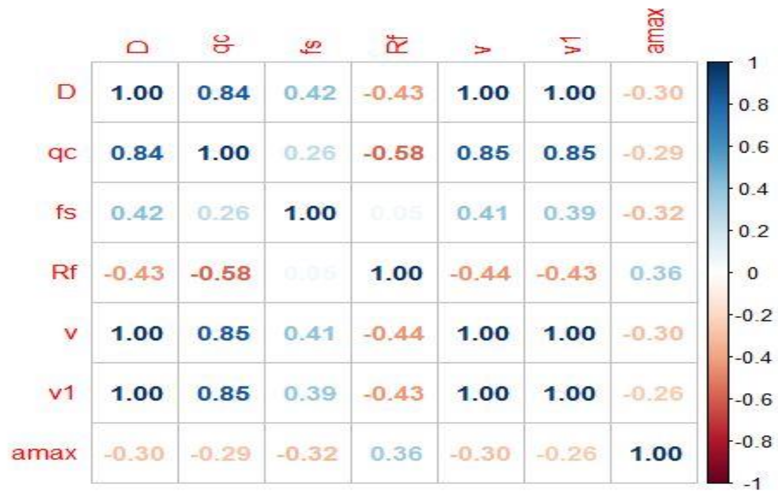
The correlation matrix has been generated between the input variables Depth D (m), Friction ratio R_f , peak horizontal acceleration (PGA) amax (g), total vertical stress σ'_{vo} (kPa), vertical effective stress σ_{vo} (kPa), cone penetration resistance q_c , frictional resistance f_s. In the correlation matrix it can be seen in the Fig 2, that the Depth D (m) is highly correlated to the variables total vertical stress σ'_{vo} (kPa), vertical effective stress σ_{vo} (kPa). Also, total vertical stress σ'_{vo} (kPa), vertical effective stress σ_{vo} (kPa) are highly correlated to each other which will affect the model. It is observe that (i) if we use higher number of parameters in the regression model it will increases the chances of error. (ii) On the other hand, it is also observe that higher number of correlation among the chosen input variables increases error in the model respectively. The first case is not problem for us but the second case will affect the model and the accuracy hence the important variables are pointed out and used in the model precisely.

Table 1
CPT Data for prediction of the Liquefaction Index (Juang et al. [1]).

Depth (m)	qc(MPa)	fs(kPa)	Rf(%)	σ'_0(kPa)	σ_0(kPa)	amax(g)	Liq
12.5	7.52	30.9	0.42	231.3	121.3	0.21	No
13.5	7.02	24.3	0.36	249.8	129.8	0.19	No
14.5	16.89	44	0.27	268.3	138.3	0.19	No
3.5	1.5	24.4	2.16	66.6	43	0.12	Yes
7.5	7.04	30	0.43	138.6	75	0.12	No
5	6.61	41.5	0.62	93.6	55	0.12	No
3.5	2.45	17.1	0.72	64.8	44.8	0.19	Yes
14.5	17.08	69.1	0.37	268.3	138.3	0.19	No
7.4	5.46	45.9	0.84	136.8	74.2	0.12	No
5	2.96	21.1	0.71	92.5	57.5	0.19	Yes
3.5	2.09	8.2	0.39	64.8	39.8	0.19	Yes
3.2	2.66	19.2	0.73	59.2	42.2	0.19	Yes
8	5.77	25	0.45	148	83	0.43	Yes
16.5	13.65	17.6	0.13	305.3	150.3	0.19	No
7.5	7.57	41.4	0.55	142.5	78.8	0.12	No
13.5	14.67	9.8	0.07	249.8	124.8	0.19	No
3.1	1.41	4.9	0.39	57.4	46.4	0.43	Yes
10.1	7.72	15.5	0.2	186.9	100.9	0.19	No
10.5	6.08	31.7	0.52	192.6	99	0.12	No
6.5	7.03	36.1	0.51	120.6	67	0.12	No

14.5	8.01	20.9	0.26	268.3	138.3	0.19	No
18.5	10.05	46.1	0.45	346	172.3	0.19	No
12.5	9.19	33	0.4	231.3	121.3	0.19	No
12.5	8.3	12.7	0.15	231.3	121.3	0.19	No
6.5	7.12	50.7	0.71	120.6	67	0.12	No
2.5	3.26	9.5	0.29	48.6	35	0.12	Yes
2.5	2.54	23	0.97	46.3	36.3	0.19	Yes
6.5	2.69	28.8	1.09	120.3	65.3	0.19	Yes
2.5	3	7.4	0.25	46.3	31.3	0.19	Yes
8.5	7.47	34.8	0.47	156.6	83	0.12	No
4.05	2.61	23.5	0.95	74.9	49.4	0.19	Yes
12.5	5.47	63.3	1.17	228.6	115	0.12	No
3.1	2.54	11.9	0.57	57.4	41.4	0.19	Yes
12.5	7.38	42.9	0.57	228.6	115	0.12	No
14	13.65	21.8	0.16	259	134	0.19	No
2.5	0.23	0.9	0.42	50	36.3	0.12	Yes
6.5	7.94	45.1	0.57	124	70.3	0.12	No
17	7.68	60.8	0.81	314.5	159.5	0.19	No
3.5	2.49	10	0.41	68.5	44.8	0.12	Yes
11.8	8.15	37	0.46	218.3	115.3	0.19	No
18.5	9.48	86.1	0.79	336.6	163	0.12	No
2.5	0.92	18.9	2.54	48.6	35	0.12	Yes
9	6.67	14.2	0.21	166.5	91.5	0.19	No
10.35	11.32	114	0.73	191.5	108	0.43	No
9.5	6.76	64.9	0.96	174.6	91	0.12	No
15.5	8.74	41	0.46	286.8	146.8	0.19	No
11.6	7.72	62.6	0.81	218.3	113.6	0.12	No
8.5	5.38	26.1	0.48	156.6	83	0.12	No
8.5	6.73	49.2	0.73	156.6	83	0.12	No
10.5	7.46	35.8	0.48	189	99	0.19	No
10	11.96	162.2	1.35	185	105	0.43	No
4.5	6.01	27.2	0.46	83.3	58.3	0.43	Yes
10.5	8.25	70.6	0.86	194.3	104.3	0.19	No
3.5	2.65	9.3	0.36	66.6	43	0.12	Yes
3.5	11.56	170	1.51	68.5	49.8	0.43	No
12.5	8.27	0.2	0.24	231.3	121.3	0.19	No
4.5	1.73	25.8	1.59	83.3	53.3	0.21	Yes
5	2.22	23.4	1.06	92.5	57.5	0.19	Yes
5.5	1.89	6.7	0.37	105.5	61.8	0.12	Yes
4.5	0.64	9.9	1.91	84.6	51	0.12	Yes
3.7	2.7	32.4	1.24	68.5	46.5	0.19	Yes
11.5	7.62	27.9	0.36	207	107	0.19	No
11.5	6.83	24.5	0.35	212.8	112.8	0.21	No
3.5	3.86	24.3	0.78	64.8	49.8	0.43	Yes
3.5	2.62	11	0.41	64.8	44.8	0.19	Yes
14	12.77	22.8	0.18	259	134	0.19	No
9.5	7.43	57.7	0.77	179.5	95.8	0.12	No
14.5	10.61	19.2	0.18	268.3	133.3	0.19	No
2.6	1.18	11.4	0.79	48.1	37.1	0.19	Yes
7.5	6.23	1.7	0.27	138.8	78.8	0.19	No
6.5	7.4	30.3	0.4	120.6	67	0.12	No
3.5	0.2	3.7	1.96	68.5	44.8	0.12	Yes
10.5	6.49	55.2	0.86	192.6	99	0.12	No

9.5	6.62	37	0.57	174.6	91	0.12	No
9	12.89	138.8	1.08	170.2	96.5	0.43	No
5	2.54	13.8	0.54	92.5	57.5	0.19	Yes
12.5	6.8	37.2	0.55	231.3	121.3	0.19	No
13.5	6.85	59.1	0.87	246.6	123	0.12	No
13.5	16.3	130.1	0.8	249.8	134.8	0.43	No
7.9	6.05	43.3	0.71	145.8	78.2	0.12	No
7.5	8.03	2.6	0.32	138.8	78.8	0.19	No
11.5	7.41	55.5	0.76	212.8	112.8	0.19	No
6.5	1.54	5.8	0.41	124	70.3	0.12	Yes
12.5	7.76	53.9	0.7	228.6	115	0.12	No
13.5	8.3	43.3	0.53	249.8	129.8	0.19	No
4.1	0.9	9	0.59	75.9	54.9	0.43	Yes
14	12.43	28.2	0.23	259	134	0.19	No
3.5	1.28	8.8	1	63	43	0.12	Yes
6.5	6.68	41.2	0.62	124	70.3	0.12	No
7.5	5.91	28	0.47	138.6	75	0.12	No
6	6.64	36.9	0.55	111.6	63	0.12	No
2.5	0.94	22.4	2.54	46.3	41.3	0.43	Yes
3.5	1.47	24.6	1.94	64.8	49.8	0.43	Yes
12.5	10.08	22	0.23	231.3	121.3	0.19	No
2.5	1.62	15.5	1	46.3	36.3	0.19	Yes
4	1.87	23.6	1.3	74	49	0.43	Yes
12.5	7.58	44.6	0.6	228.6	115	0.12	No
13.5	8	26.8	0.36	249.8	129.8	0.19	No
11.5	8.32	27.1	0.34	216.5	112.8	0.19	No
3.5	0.18	0.6	0.37	68.5	44.8	0.12	Yes
19.5	11.26	35.5	0.32	364.5	180.8	0.19	No
12.5	7.68	58.7	0.77	228.6	115	0.12	No
6.1	7.24	41.4	0.57	116.6	66.9	0.12	No
11.5	7.99	43.3	0.54	210.6	107	0.12	No
13.5	6.54	49.8	0.76	246.6	123	0.12	No
5	5.93	54.4	0.92	96.2	57.5	0.12	No
4.5	2.78	20.7	0.74	96.2	48.3	0.19	Yes
8	6.61	26	0.4	148	83	0.19	No
7.5	5.59	21.8	0.4	138.6	75	0.12	No
8.5	6.12	30.6	0.51	161	87.3	0.12	No
13.5	7.41	58.9	0.79	246.6	123	0.12	No
13.9	11.58	29.5	0.28	257.2	133.2	0.19	No
9.5	7.18	45.5	0.64	179.5	95.8	0.12	No
4.5	2.01	5.1	0.25	87	53.3	0.12	Yes
13.5	6.32	61.5	0.98	246.6	123	0.12	No
7.5	5.21	28.8	0.55	142.5	78.8	0.12	No
4.5	1.82	22.8	1.25	83.3	53.3	0.19	Yes
8.5	6.21	24.8	0.4	161	87.3	0.12	No
15.5	14.74	26.2	0.2	286.8	141.8	0.19	No
7.5	3.05	32.5	1.07	138.8	73.8	0.19	Yes
11.1	6.7	46.9	0.72	205.4	109.4	0.19	No
12.5	8.83	57.7	0.66	235	121.3	0.12	No
13	5.16	62	1.21	237.6	119	0.12	No
14	12.15	0.3	0.25	259	134	0.19	No
4.5	0.64	27.5	4.2	84.6	51	0.12	Yes



	D	qc	fs	Rf	v	v1	amax
D	1.00	0.84	0.42	-0.43	1.00	1.00	-0.30
qc	0.84	1.00	0.26	-0.58	0.85	0.85	-0.29
fs	0.42	0.26	1.00	0.05	0.41	0.39	-0.32
Rf	-0.43	-0.58	0.05	1.00	-0.44	-0.43	0.36
v	1.00	0.85	0.41	-0.44	1.00	1.00	-0.30
v1	1.00	0.85	0.39	-0.43	1.00	1.00	-0.26
amax	-0.30	-0.29	-0.32	0.36	-0.30	-0.26	1.00

Fig. 2. Correlation between the input variables used in the prediction (note: v = $\sigma_{v0,}$ v1= σ'_{v0}).

3.1. Artificial neural network (ANN)

Training and testing performance (%) has been calculated by using the following formula:

Training performance (%) or Testing performance (%)

$$\left(\frac{No.of\ data\ accurately\ predicted\ by\ ANN}{Total\ data}\right)\times 100$$

The R “neuralet” package allows resilient back propagation algorithm and error function use sum of square error (sse).

(1) 125 out of 88 data is used for training NN.

(2) 125 out of 37 data is used for testing only.

The developed neural model shown in Fig 3, is introduced with the 37 data sets that are purely unknown and new data for the model, and 88 data sets are use for training purpose. Probability of liquefaction index, LI is taken “0” for non liquefied cases and LI “0.5 – 1” for the liquefied cases, if the value is between “0 - 0.5” it is considered as least criterion for susceptibility of soil.

In the test the variables like Depth D (m) and total vertical stress σ'_{vo} (kPa) because of their high correlation coefficients are excluded. And the input variables used are Friction ratio R_{f}, peak horizontal acceleration (PGA) amax (g), vertical effective stress σ_{vo} (kPa), cone penetration resistance q_c , frictional resistance f_s. The results observe has accuracy of 100 % as shown in the Table 2.

Table 2
Predictive Results obtained by ANN.

Input Variable	Error	Steps	Total Accuracy (%)
Rf, $a_{max,}$ σ_{v0} , qc, fs	0.00033	156	100

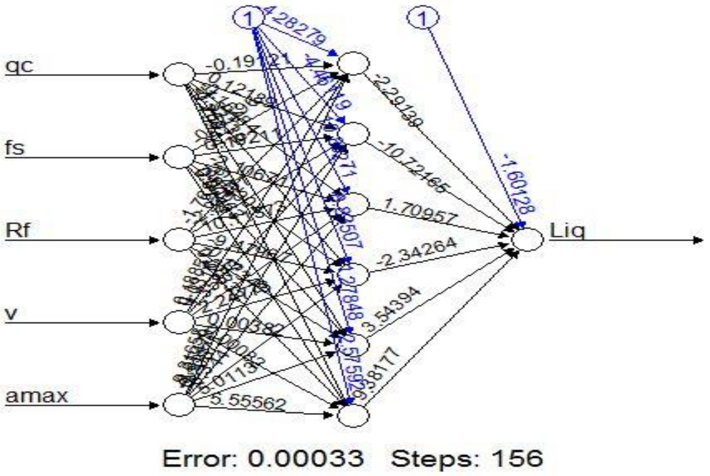


Fig. 3. Generated ANN model with the input variables (Rf, $a_{max,}$ σ_{v0} , qc, fs) (note: v = σ_{v0}).

3.2. C 4.5 algorithm

The size of tree generated is 7 and the number of leaves is 4 in the model. 10 fold cross validation as a test mode is used. The confusion matrix generated by the program correctly classified 122 instances and 3 instances incorrectly out of 125 instances thus 97.6% accuracy is found for the C4.5 DT. The preprocess statics of variable used in the model are shown in the Table 3. With the performance measure, AUC of ROC, MCC, precision, recall, and F-measure are use to select optimal model performance for liquefaction and non-liquefaction instances.

Table 3
Preprocess statics of variable used in the model.

Parameters	Minimum	Maximum	Mean	Standard Deviation
Cone Penetration Resistance, qc (MPa)	0.18	17.08	6.376	3.807
Frictional Resistance, fs (kPa)	0.2	170	34.991	28.776
Friction ratio, Rf (%)	0.07	4.2	0.702	0.557
Vertical effective stress, σ_{v0} (kPa)	46.3	364.5	162.49	80.128
Peak horizontal acceleration, amax (g)	0.12	0.43	0.185	0.091

The summary of stratified cross validation shown below in the Table 4. In the table, mean absolute error, root mean squared error, relative absolute error, and root relative squared error calculated by the program and based on the results correctly and incorrectly classified instances are shown.

Table 4
Summary of stratified cross validation for C 4.

Correctly Classified Instances	122	97.6 %
Incorrectly Classified Instances	3	2.4 %
Kappa statistic	0.9445	
Mean absolute error	0.0254	
Root mean squared error	0.153	
Relative absolute error	5.7355 %	
Root relative squared error %	32.5743	
Total Number of Instances	125	

The confusion matrix generated by the algorithm and it classified -

a b <-- classified as	
84 0	**a = No**
3 38	**b = Yes**

In the confusion matrix it is observed that the algorithm has classified 3 instances wrong. The detailed accuracy by class is made on the TP Rate, FP Rate, Precision, Recall, F-Measure MCC, ROC Area, PRC Area. For the performance measure, AUC (area under curve) of ROC, MCC (Matthews correlation coefficient), precision, recall, and F-measureare, OA overall accuracy use to select optimal model performance separately for liquefaction and non-liquefaction instances shown in Table 5.

Table 5
Predictive results by C4.5 DT model.

Model	OA	AUC	MCC	Recall	Precision	F-Measure	Liquefaction
C4.5 DT	0.976	0.993	0.946	1.000	0.966	0.982	No
		0.979	0.946	0.927	1.00	0.962	Yes

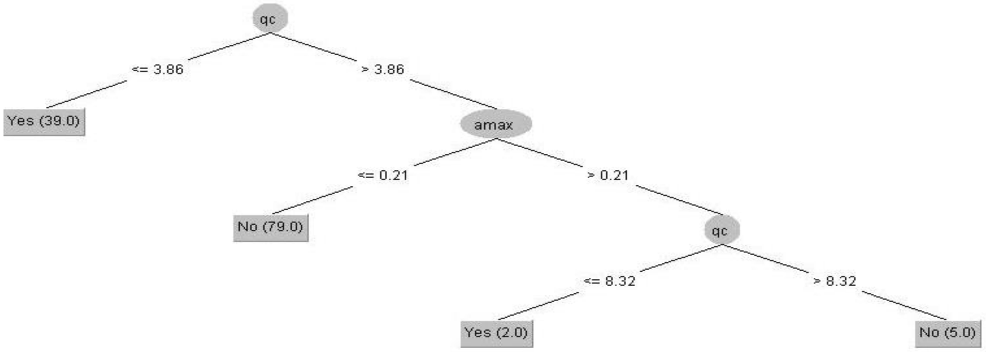


Fig. 4. Decision Tree Visualization.

3.3. Random forest algorithm

The confusion matrix generated by program correctly classified instances 123 and incorrectly classified instances 2 out of 125 instances thus the accuracy found 98.4% shown in Table 7. A 10 fold cross validation as a test mode is used. In the model, bagging with 100 iterations and base learner is used. In Table 6 Summary of stratified cross validation for Random Forest is shown.

Table 6
Summary of stratified cross validation for Random Forest.

Correctly Classified Instances	123	98.4 %
Incorrectly Classified Instances	2	1.6 %
Kappa statistic	0.9632	
Mean absolute error	0.0293	
Root mean squared error	0.1144	
Relative absolute error	6.6245 %	
Root relative squared error	24.3573 %	
Total Number of Instances	125	

Table 7
Predictive Results by Random Forest Algorithm.

Model	OA	AUC	MCC	Recall	Precision	F-Measure	Liquefaction
RF	0.984	0.999	0.964	1.000	0.977	0.988	No
		0.997	0.964	0.951	1.000	0.975	Yes

The confusion matrix generated by the algorithm and it classified -

a b <-- classified as	
84 0	**a = No**
2 39	**b = Yes**

3.4. Comparison between C 4.5 and random forest

On the basis of the performance measure, AUC of ROC, MCC, precision, recall, and F-measure is shown below for the both C 4.5 and Random Forest in the Table 8. From the table AUC, MCC and F-Measure of Random Forest are higher in comparison with C4.5 this shows that results obtained from the Random Forest model is an ideal one.

C4.5 DT model 97.6 % accurate and Random Forest model 98.4% accurately predicted the liquefaction and non-liquefaction cases.

Table 8
Comparison between C4.5 and Random Forest.

Model	**OA**	**AUC**	**MCC**	**Liquefaction**			**No Liquefaction**		
				Recall	**Precision**	**F-Measure**	**Recall**	**Precision**	**F-Measure**
C4.5	0.976	0.989	0.946	0.927	1.000	0.962	1.000	0.966	0.982
RF	**0.984**	**0.999**	**0.964**	0.951	1.000	**0.975**	1.000	0.977	**0.988**

4. Conclusions

Resilient Backpropagation algorithm is easier to implement and robust against the input parameters and found very effective than the traditional back propagation algorithm used by other researchers [1–5] in predicting the liquefied and non-liquefied cases in the neural networks. The model classified 37 cases correctly and the accuracy is found 100 % which is accurate in comparison with other two algorithms and has shown good results in comparison with other implementations done by researchers in AI technique [1–5]. C4.5 and Random Forest Algorithm are 97.6% and 98.4% accurate in prediction of liquefied and non-liquefied cases. The C4.5 took only 0.05 seconds if we compare to the random forest, which took 0.16 seconds to build the model. C4.5 incorrectly classified 3 cases whereas as Random forest classified 2 cases incorrectly. This is found that Friction ratio R_f, peak horizontal acceleration (PGA) amax (g), vertical effective stress σ_{vo} (kPa), cone penetration resistance q_c , frictional resistance f_s are very important variable and helps in getting good results for all three algorithms used in the paper.

The Resilient Back propagation algorithm (Rprop) is very effective and must be implementing in monitoring the liquefaction susceptibility of saturated soils. However, neural network has its own limitation like 'Black Box' in which we mainly get only output results via input variables without knowing what actually is happening in the model whereas in C 4.5 and Random Forest Algorithm we can see what actually is happening with the models inputs and output variables by parameters discussed in tables above. We can use further more field data like; Standard Penetration Test (SPT) data can be implement with the model to increase the accuracy and working of the model for future predictions of liquefaction in the particular area. Finally, this paper encourages the use of the Resilient Back propagation algorithm (Rprop) in prediction of the Liquefaction cases.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Juang CH, Yuan H, Lee D-H, Lin P-S. Simplified Cone Penetration Test-based Method for Evaluating Liquefaction Resistance of Soils. J Geotech Geoenvironmental Eng 2003;129:66–80. https://doi.org/10.1061/(ASCE)1090-0241(2003)129:1(66).

[2] Young-Su K, Byung-Tak K. Use of Artificial Neural Networks in the Prediction of Liquefaction Resistance of Sands. J Geotech Geoenvironmental Eng 2006;132:1502–4. https://doi.org/10.1061/(ASCE)1090-0241(2006)132:11(1502).

[3] Pal M. Support vector machines-based modelling of seismic liquefaction potential. Int J Numer Anal Methods Geomech 2006;30:983–96. https://doi.org/10.1002/nag.509.

[4] Seed HB, Idriss IM. Simplified Procedure for Evaluating Soil Liquefaction Potential. J Soil Mech Found Div 1971;97:1249–73. https://doi.org/10.1061/JSFEAQ.0001662.

[5] Samui P. Seismic liquefaction potential assessment by using Relevance Vector Machine. Earthq Eng Eng Vib 2007;6:331–6. https://doi.org/10.1007/s11803-007-0766-7.

[6] Youd TL, Idriss IM. Liquefaction Resistance of Soils: Summary Report from the 1996 NCEER and 1998 NCEER/NSF Workshops on Evaluation of Liquefaction Resistance of Soils. J Geotech Geoenvironmental Eng 2001;127:297–313. https://doi.org/10.1061/(ASCE)1090-0241(2001)127:4(297).

[7] Ahmad M, Tang X-W, Qiu J-N, Ahmad F. Evaluating Seismic Soil Liquefaction Potential Using Bayesian Belief Network and C4.5 Decision Tree Approaches. Appl Sci 2019;9:4226. https://doi.org/10.3390/app9204226.

[8] Riedmiller M, Braun H. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. IEEE Int. Conf. Neural Networks, IEEE; n.d., p. 586–91. https://doi.org/10.1109/ICNN.1993.298623.

[9] Tabari MMR, Azadani MN, Kamgar R. Development of operation multi-objective model of dam

reservoir under conditions of temperature variation and loading using NSGA-II and DANN models: a case study of Karaj/Amir Kabir dam. Soft Comput 2020. https://doi.org/10.1007/s00500-020-04686-1.

[10] Samui P, Sitharam TG. Machine learning modelling for predicting soil liquefaction susceptibility. Nat Hazards Earth Syst Sci 2011;11:1–9. https://doi.org/10.5194/nhess-11-1-2011.

[11] Chen S-M, Shie J-D. Fuzzy classification systems based on fuzzy information gain measures. Expert Syst Appl 2009;36:4517–22. https://doi.org/10.1016/j.eswa.2008.05.020.

[12] Heidarzadeh H, Kamgar R. Necessity of applying the concept of the steady state on the numerical analyses of excavation issues: laboratory, field and numerical investigations. Geomech Geoengin 2022;17:413–25. https://doi.org/10.1080/17486025.2020.1755466.

[13] Kaveh A, Hamze-Ziabari SM, Bakhshpoori T. Patient rule-induction method for liquefaction potential assessment based on CPT data. Bull Eng Geol Environ 2018;77:849–65. https://doi.org/10.1007/s10064-016-0990-3.

[14] Heidarzadeh H, Kamgar R. Evaluation of the Importance of Gradually Releasing Stress Around Excavation Regions in Soil Media and the Effect of Liners Installation Time on Tunneling. Geotech Geol Eng 2020;38:2213–25. https://doi.org/10.1007/s10706-019-01158-8.

[15] Zhou J, Li E, Wang M, Chen X, Shi X, Jiang L. Feasibility of Stochastic Gradient Boosting Approach for Evaluating Seismic Liquefaction Potential Based on SPT and CPT Case Histories. J Perform Constr Facil 2019;33:04019024. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001292.