Innovative Infrastructure Solutions (2021) 6:199
https://doi.org/10.1007/s41062-021-00568-z

TECHNICAL PAPER

Estimation of modified expansive soil CBR with multivariate adaptive regression splines, random forest and gradient boosting machine

Chijioke Christopher Ikeagwuani[1]

Received: 18 February 2021 / Accepted: 14 June 2021

Abstract

Construction of flexible pavement on expansive soil subgrade relies on the safe determination of California bearing ratio (CBR) value, a critical component in flexible pavement design. However, its determination, particularly in the laboratory, often consumes sufficient man-hours. This necessitated the urgency to explore alternative procedures, such as the development of reliable models to estimate the CBR of subgrade especially modified expansive soil subgrade. In the present study, three machines learning models, which are multivariate adaptive regression splines (MARS), random forest and gradient boosting machine models, were developed to predict the CBR of expansive soil subgrade blended with sawdust ash, ordinary Portland cement and quarry dust. The performance of the models was evaluated using several error indices, and the results obtained from the evaluation showed that the random forest model has superior predictive ability when compared with the MARS and gradient boosting machine models. Specifically, the R^2 values for the training and testing data for the random forest model, which were, respectively, 0.84829 and 0.75282, clearly indicated that the random forest model has good predictive ability and possesses greater generalization ability than the other developed models in this study.

Keywords California bearing ratio · Expansive soil · Gradient boosting machine · Multivariate adaptive regression splines · Random forest · Stabilizers

Introduction

Constructions of geo-infrastructures on expansive soils are usually limited by the obnoxious perennial swell-shrink behavior that often characterized the soil [1–5]. This obnoxious behavior is prevalent in expansive soils that contains smectite group of clay mineral. This clay mineral, which absorbs moisture easily during wet season, is widely known for the failure of most geo-infrastructures founded on the expansive soil. In order to strengthen the soil so that this limitation identified with it can be overcome and structures constructed safely on it, several mitigation approaches have been suggested by most geotechnical and highway researchers and engineers. The most widely used mitigation approaches are replacement strategy and stabilization approach. Stabilization approach involves blending of the soil with another soil that possesses good engineering qualities or mixing the expansive soil with either traditional or non-traditional stabilizers.

The traditional stabilizers, which are highly effective in reducing excessive volume change, are often calcium-based in nature. They have been applied successfully to minimize the swelling potential of expansive soil and also to strengthen it so that it can withstand enormous load without failure [6, 7]. The non-traditional stabilizers, on the other hand, are non-conventional materials that derive their strength basically through pozzolanic reaction that occurs between them and the expansive soil when blended together. The types of traditional stabilizers available in recent times are inexhaustible. They include but not limited to agricultural and industrial by-products. Some of the agricultural by-products used as traditional stabilizers include, nanosized palm bunch ash, sawdust ash, baggasse ash, coconut shell and coconut husk ash, rice husk ash, etc [2, 8–12]; while the industrial by-products include mine tailings, cement kiln dust, pulverized coal ash, rubber powder polymer, carpet waste fiber etc [13–19].

Interestingly, and as noted earlier, both the traditional and the non-traditional stabilizers have proved effective in

✉ Chijioke Christopher Ikeagwuani
chijioke.ikeagwuani@unn.edu.ng

[1] Civil Engineering Department, University of Nigeria, Nsukka, Enugu State, Nigeria

Published online: 17 July 2021

improving the strength properties of expansive soil used as subgrade. One of such strength properties, which is highly critical in the safe design of flexible pavement, is the California bearing ratio (CBR) [20, 21]. The CBR test is determined or performed either in the laboratory or in the field. The CBR test performed in the field is usually done on the ground surface or on the surface of an excavated test pit; while the CBR test performed in the laboratory is usually executed on compacted samples placed in a CBR machine [22]. The CBR performed in the laboratory, though not laborious, consumes much time and saps a lot of energy. Therefore, any approach that will lead to accurate determination or even the prediction of the CBR of expansive soil with less time and effort is usually a welcome development. The importance of accurate prediction of CBR of expansive soil especially an expansive soil treated with stabilizers cannot be overemphasized. Reliable prediction of CBR of modified or treated expansive soil ensures safe design of flexible pavement.

Several models have been developed to predict the CBR of either treated or untreated expansive soil used as subgrade, and these models are widely documented in the literature [23–25]. Most of these developed models, to the extent of the author's knowledge, do not consider the effect of stabilizers, thereby making their application in practice somewhat limited by the non-inclusion of stabilizer effect in their development. Moreover, most of the models were built with extremely few data points while some have been reported to be highly illogical [21, 26, 27]. For instance, some of the models have inverse relationship existing between the CBR and the maximum dry density (MDD) of soils while some were built using soil classification instead of values measured from the laboratory [21, 27–30]. Even the well-established model (Eq. 1) suggested by the National Cooperative Highway Research Programme (NCHRP) of the USA [31, 32] do not possess any additive parameter in their model.

$$\mathrm{CBR} = \frac{75}{1 + 0.728(\mathrm{wPI})} \tag{1}$$

where w represents the percentage of sample passing US sieve No. 200 and PI denotes plasticity index.

Furthermore, some of the developed predictive CBR models available in the literature have pretty low degree of correlation coefficient which prompted Taskiran [31] to suggest that CBR estimation is difficult to develop with classical statistical approaches. For this reason, Taskiran [31] recommended using machine learning techniques to build CBR models because of their robustness and their ability to handle complex computations with ease. To prove the efficaciousness of machine learning techniques in developing models to predict the CBR of soils, Taskiran [31], in a study on the development of CBR models, utilized two machine learning techniques namely, artificial neural network (ANN) and genetic expression programming to develop predictive CBR models. The results obtained from the study clearly revealed the superior capability of machine learning techniques over classical statistical approaches.

In a recent similar study, Taha et al. [21] employed ANN, a machine learning technique as mentioned earlier, to develop a model to predict the CBR of granular material and also reported that machine learning technique is an effective tool for the development of CBR models. In the same vein, Yilidirm and Gunaydin [25] utilized ANN to develop model to predict the CBR of fine-grained soil and concluded that ANN performed satisfactorily in the development of the CBR model based on the high correlation coefficient obtained from the study. Faris et al. [33], in a related study, developed predictive CBR models using radial basis network (RBN), a form of machine learning technique and linear polynomial regression. The results obtained from the study showed that machine learning technique is effective for developing CBR predictive model.

Innovation of the study

It is worth stating that most of the developed CBR models in the literature, whether with machine learning techniques or with classical statistical approaches, does not include additives effect as mentioned earlier. Therefore, in the light of the foregoing limitations in the preceding section, the present study adopted an innovative approach, which is the incorporation of the effect of stabilizers in the development of models for the predication of reliable CBR with machine learning techniques for modified expansive soil. As mentioned earlier, the effect of stabilizers, to the extent of the author's knowledge, is rarely integrated in the development of CBR models of modified expansive soil. In this study, the modified expansive soil was blended with ordinary Portland cement (OPC), sawdust ash (SDA) and quarry dust (QD). The models were developed using three machine learning techniques which are multivariate adaptive regression splines, random forest and gradient boosting machine. The machine learning techniques were adopted for the development of the models to predict the CBR of modified expansive soils because they have proven to be effective as prediction tools in most engineering disciplines [21, 34].

Multivariate adaptive regression splines

Multivariate adaptive regression splines (MARS) is a supervised learning technique that was invented and developed by Friedman [35]. It is a flexible, interpretable non-parametric regression algorithm that is included in a group of statistical approaches that can be employed

to fit a relationship between input variables and output variables. It utilizes divide and conquer strategy to make effective prediction [36, 37]. The divide and conquer strategy is executed in a manner that a training dataset is split into several linear piecewise linear segments with different piecewise linear regression functions that are added together to form the MARS model. The linear piecewise segment regression functions are known as basic functions (BFs). The BFs, which are essential part of MARS model, are added together to develop the function f of the MARS model as shown in Eq. (2).

$$y = f\left(X_1, \ldots, X_q\right) + \varepsilon \tag{2}$$

where y represents the output variable, $X_1, \ldots, X_q$ denotes the input variables in q space and ε represents the error distribution in the model.

It is of interest to state that the piecewise linear function in MARS model can be in any of the following forms: (1) constant function; (2); a hinge function;(3) a product of two or several hinge functions or even. Hinge function is usually expressed as shown in Eqs. (3) and (4):

$$\mathrm{BF}_m^+ = \max\left(0; x - t\right) = \begin{cases} x - t, & x \geq t \\ 0, & x < t \end{cases} \tag{3}$$

$$\mathrm{BF}_m^- = \max\left(0; t - x\right) = \begin{cases} t - x, & t \geq x \\ 0, & t < x \end{cases} \tag{4}$$

where t is the threshold value of the input variable x, termed as knot; BF_m^+ is the BFs that defines the right-side section of the knot while BF_m^- is the BFs that defines the left-side section of the knot. The global MARS model, which is expressed in Eq. (5), is the linear combination of the BFs together with their interactions.

$$\hat{y} = B_0 + \sum_{m=1}^{M} \psi_m B_m(x) \tag{5}$$

where $\hat{y}$ signifies the predicted output of the MARS model; B_0 represents the MARS model constant term; B_m signifies the m^{th} BF and ψ_m represents the coefficient of m^{th} BF.

The global MARS model is often improved in the two stages that are involved in the development of MARS model. The two stages are the forward pass stage and the backward pass stage. Specifically, in the forward pass stage, the constant term is, first and foremost, added to the model. Then BFs, using an iterative procedure, are included in the MARS model to improve the model. In the iteration procedure, the best pairs of BFs are selected for the MARS model improvement. The pairs of BFs consist of truncated spline function for the left-sided segment as well as the right-sided segment. In addition, the MARS model algorithm finds the best possible knots location for every predictor [38]. The forward pass process is continued until the maximum user-specified BFs number is added to the model. The forward pass stage usually results in a significantly improved model. The limitation of the model is that it is complex and over-fitted and has low generalization ability. In order to overcome these limitations, the MARS model is made to undergo the second stage known as backward pass stage where the model developed in the forward pass stage is pruned so that the generalization ability of the model is enhanced. The pruning is executed by utilizing a "lack of fit" criterion known as generalized cross-validation (GCV). The role of the GCV is to evaluate the importance of each BF contribution to the model developed in the forward stage and then it expunge accordingly, BFs which have less contribution to the model. The GCV is expressed as:

$$\mathrm{GCV} = \frac{1}{k} \frac{\sum_{i=1}^{k} \left(y_i - \hat{y}\right)^2}{\left(1 - \frac{M + d\ x(M-1)/2}{k}\right)^2} \tag{6}$$

where k represents the number of observation; M denotes the number of BFs in the model; d signifies the penalty parameter while y and $\hat{y}$ denote the actual and predicted responses, respectively. It is worth noting that GCV penalizes both the number BF and knots present in the MARS model.

Random forest

Random forest is a non-parametric classification or regression technique that belongs to a family of supervised machine learning methodologies that incorporate several decision trees to produce desired response. It is an ensemble technique that was pioneered by Amit and Geman [39] and was subsequently expanded broadly by Breiman [40]. It utilizes a combination of random subspace approach developed by Ho [41, 42] and bootstrap aggregation technique proposed by Breiman [43–47] to generate highly reliable response during its prediction. The random subspace method, which is the first technique used in random forest, ensures that prediction error is drastically reduced as studies have shown that the output from a single decision tree is not much better than the output from a combination of several decision trees when random subspace is utilized [39, 42]. On the other hand, bootstrap aggregation technique, which is the second technique used in random forest approach, guarantees that each decision tree in the random forest technique is distinctive. This distinctive decision trees causes a significant reduction in the prediction variance and also minimizes overfitting of any model developed with random forest.

It is pertinent to point out that since the invention of random forest technique, which has gained wide acceptance among researchers and practitioners in industries, several

variants of the method earlier introduced by Breiman [40] have been developed. Some of the variants include random survival forests [48], multivariate random forest [49], enriched random forest [50], quantile regression forests [51], etc. Some of these variants were developed to tackle different problem contexts or were developed with an entirely different re-sampling fitting method with the sole intent of increasing the generalization ability of the random forest technique [52]. Most of the developed variants have been employed at different times by numerous researchers in civil engineering, and other disciplines to solve several problems and the results obtained, in terms prediction of responses, have been impressive [53–60].

Specifically, in civil engineering setting, Gong et al. [61] applied random forest to evaluate the effect of asphalt mixtures characteristics on the performance of pavement and reported that random forest proved highly effective as a prediction algorithm. Similarly, Zhang et al. [62] utilized random forest to model uniaxial compressive strength of self-compacting concrete and reported that random forest had an impressive performance in its ability to predict responses. It was also reported that the model developed with random forest showed high generalization ability. There are other areas in civil engineering discipline where random forest, which has a well-defined algorithm, was applied and it performed better than most supervised learning algorithm it was compared it [63].

Random forest algorithm

Assume that the training dataset $\boldsymbol{Q} = \{(\boldsymbol{x}_i, y_i), \ldots, (\boldsymbol{x}_n, y_n)\}$. h an n number of samples and d. -dimensional features, where $x_i \in \mathbb{R}^n$. and $y_i \in \mathbb{R}$. The random forest algorithm is described briefly as follows:

1. Produce bootstrap samples ($\boldsymbol{E}_i, \ \ldots, \ \boldsymbol{E}_K (i = 1, \ \ldots, \ K)$.) from $\boldsymbol{Q}$. The training dataset, $\boldsymbol{Q}$ is sampled with replacement. The sample sizes for the bootstrap samples are the same with that of the training dataset. This is done in line with the suggestion made by Brieman [43]. Recently, due ease of computation, numerous researchers frequently fewer sample size for the bootstrap samples.
2. Grow decision tree, $\boldsymbol{T}_m (i = 1, \ \ldots, \ M)$, from each of the bootstrap samples $\boldsymbol{E}_m$ by employing the following modification:

i. From each node, select the best split from a randomly selected subset of m_{try} predictors from the d-predictors.

ii Ensure the decision trees are fully grown without pruning as is the case with classification and regression tree (CART) where decision trees are pruned. The decision trees are grown in such a way that it would be impossible to further split any node.

3 Note that M denotes the number of trees that are present in the forest while m_{try} represents the number of randomly chosen,d input variables or predictors. Both m_{try} and M are defined by the user when tuning the parameters in random forest algorithm. Repeat steps 1–2 until enough $\boldsymbol{T}_m$ has been grown.

4. Predict the response for an entirely new dataset using the expression:

$$\hat{y}_m^*(x) = \frac{1}{M} \sum_{m=1}^{M} y_m(x) \tag{7}$$

where $\hat{y}_m^*(x)$ represents the prediction of the random forest prediction which is obtained from the summation of the *mth* individual trees; $y_m(x)$ denotes the prediction of the mth individual tree for the input vector x.

Estimation of error rate with random forest technique

Basically, there are two ways, error rate, which is used to evaluate model performance, can be executed with a model built with random forest algorithm. The first approach is the conventional method that involves splitting a dataset into training and testing datasets and then comparing the error of both the training and the testing datasets. The second approach is the so-called out-of-bag (OOB) error that is used in this study. OOB error is a kind of cross-validation method that is executed implicitly in the random forest algorithm.

Notably, during the development of a model using random forest algorithm, not all the training dataset are utilized for the development of the model. This is because the bootstrap sampling procedure in the random forest algorithm involves replacement. This allows some samples to be extracted more than once for the growing of the decision trees and some samples not to be extracted or "left out" for growing the decision trees. The samples that are left out or not extracted are referred to as the OOB samples. These are the samples that are used for the verification of the developed model [52]. Predictions executed with the OOB samples are described as OOB predictions, and any error obtained from the predictions is called OOB prediction error. The OOB prediction error for an *ith* input vector is usually aggregated, and the mean estimated using the expression shown in Eq. (7).

The expression for the OOB error is given as:

$$\hat{y}_i^{\mathrm{OOB}} = \frac{1}{k} \sum_{m=1}^{k} \left(y_{i,m}^{\mathrm{OOB}} \right) \tag{8}$$

where $y_{i,m}^{\mathrm{OOB}}$ represents the ith OOB prediction of the m th tree; k denotes the total number of trees used for the prediction of the OOB sample.

Gradient boosting regression

Gradient boosting machine (GBM), just like random forest, is an ensemble technique that belongs to a family of methodologies that uses several classification or regression trees in its algorithm to generate a reliable and desired response. The classification or regression trees, which are aptly described as base learners, are added sequentially in GBM to improve the performance of the algorithm. GBM was originally developed for only classification tasks but it was extended to regression tasks by Friedman [64, 65]. GBM algorithm, during each iteration, takes into account previously ensemble tree error, and while it performs prediction in the next tree, endeavors to recover the error. Thus, there is a constant decrease in the error in subsequent tree ensemble.

In addition, GBM depends on the concept of boosting where several combination of models that possess high bias and low variance are employed to drastically minimize the high bias while maintaining the low variance. This implies that GBM combines several shallow trees to improve the performance of the prediction. The shallow trees are trained with the same dataset. It is worth stating that the improvement in the performance of the prediction has endeared many researchers from several disciplines including civil engineering to it [66–69]. Recently, Kaloop et al. [70] utilized GBM to predict the compressive strength of high-performance concrete and submitted that GBM is a powerful and reliable prediction algorithm. Other investigators, who have used GBM, particularly in civil engineering discipline, gave similar submission [71, 72]. Notably, GBM, as mentioned earlier, was originally developed for classification tasks alone but it was later extended to regression tasks using the algorithm described below:

Gradient boosting algorithm

Consider a dataset which consists of predictors with t features and p number of sample points which is represented as $(\mathbf{x}_i,\ y)_1^p$, where $\mathbf{x}_i = (x_1,\ x_2,\ \ldots, x_t)$ denotes the input variables while the output variables is represented as y. It is pertinent to point out at this juncture that GBM is a prediction algorithm that aims to look for an additive model that minimizes a defined loss function $L(y,\ F(\mathbf{x}))$. In addition, GBM is an algorithm that tries to establish an estimate function that maps an input vector (predictor) $\boldsymbol{X}$ to the predicted variable y such that the defined loss function is minimized.

$$\begin{aligned} y &= F^*(x) \\ F^*(x) &= \arg\min_{F(x)} L(y,\ F(x)) \end{aligned} \tag{9}$$

There are numerous well-established loss functions that can be employed in GBR algorithm. The three commonly used ones are the Huber loss function (Eq. 10), absolute error (Eq. 11) and squared error (Eq. 12). Of these aforementioned three loss functions, the squared error function seems to be the most appealing loss functions among users of GBR algorithm. This is because of ease computation and also the fact that it can easily be differentiated [64, 73]. Additionally, the squared error function is just the residual of $y - F$. This implies that the GBR algorithm executes just residual refitting [73–75].

$$L(y,\ F)_{Huber,\ \delta} = \begin{cases} \frac{1}{2}(y-F)^2 & |y-F| \le \delta \\ \delta(|y-F| - \delta/2) & |y-F| > \delta \end{cases} \tag{10}$$

$$L(y,\ F)_{L_1} = |y-F| \tag{11}$$

$$L(y,\ F)_{L_2} = \frac{1}{2}(y-F)^2 \tag{12}$$

where δ represents the threshold whereby the loss function transits from square error to absolute error loss function.

Remarkably, as soon as a loss function is defined during the application of GBR algorithm, the following outlined steps will be implemented:

1. The model, first and foremost, is initialized through the application of a constant value that is estimated from the expression shown in Equation (13).

$$F_0(\mathbf{x}) = \arg\min_{\rho} \sum_{i=1}^{p} L(y_i,\ \rho) \tag{13}$$

2. With incremental steps (boost) say from $m = 1,\ 2,\ \ldots,\ M$, steps 3 to 5 are performed:

3. Evaluate the negative gradient through the expression:

$$g_i = -\left[\frac{\partial L(y_i,\ F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})} \quad i = 1,\ 2,\ \ldots,\ p \tag{14}$$

4. Utilized the evaluated negative gradient with the predictors to fit a regression tree, $h(\mathbf{x}_i;\boldsymbol{a}_m)$, and then estimate the regression tree parameter, $\boldsymbol{a}_m$, with the expression shown in Eq. (15).

$$\boldsymbol{a}_m = \arg\min_{\boldsymbol{a},\ \beta} \sum_{i=1}^{p} \left[g_i,\ -\beta h(\mathbf{x}_i;\boldsymbol{a})\right]^2 \tag{15}$$

5. Replace the negative gradient with the $h(\mathbf{x}_i;\boldsymbol{a}_m)$ that was obtained with the steepest-descent strategy and utilized it to determine the best gradient descent step size, ρ_m. This is done using Eq. (15).

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{p} L\left(y_i,\ F_{m-1}\left(\mathbf{x}_i\right) + \rho h\left(\mathbf{x}_i;\boldsymbol{a}_m\right)\right) \tag{16}$$

6. Finally, update the model using this expression:

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \gamma \rho_m h\left(\mathbf{x};\boldsymbol{a}_m\right) \tag{17}$$

where M signifies the number of successive boosts; $\boldsymbol{a}_m$ represents the regression tree parameter and ρ_m signifies best gradient descent step size. $\gamma \in (0,\ 1]$ denotes the shrinkage parameter and it is incorporated to each base learner $h\left(\mathbf{x};\boldsymbol{a}_m\right)$ contribution to $F_m(\mathbf{x})$ [64]. The gradient boosting algorithm is written concisely as follows:

GBR Algorithm

1 **Initialize**: $F_0(\mathbf{x}) = \arg\min_{\rho} \sum_{i=1}^{p} L\left(y_i,\ \rho\right)$

2 For $m = 1,\ 2,\ \ldots,\ M$, perform:

3 $g_i = -\left[\frac{\partial L\left(y_i,\ F(\mathbf{x}_i)\right)}{\partial F(\mathbf{x}_i)}\right]_{F(\mathbf{x})=F_{m-1}(\mathbf{x})} \quad i = 1,\ 2,\ \ldots,\ p$

4 $\boldsymbol{a}_m = \arg\min_{a,\ \beta} \sum_{i=1}^{p} \left[g_i,\ -\beta h\left(\mathbf{x}_i;\boldsymbol{a}\right)\right]^2$

5 $\rho_m = \arg\min_{\rho} \sum_{i=1}^{p} L\left(y_i,\ F_{m-1}\left(\mathbf{x}_i\right) + \rho h\left(\mathbf{x}_i;a_m\right)\right)$

6 $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \gamma \rho_m h\left(\mathbf{x};a_m\right)$

End for

Cross-validation

Cross-validation is a re-sampling of data procedure which is frequently applied in machine learning technique to improve predictive model generalization ability as well as to prevent the model from being over-fitted [76–78]. There are several forms of cross-validation procedure that are usually utilized in machine learning technique. The most frequently used cross-validation procedures include Monte-Carlo, leave-one-out, bootstrap, k-fold [79, 80]. In the present study, the k-fold cross-validation procedure was employed as the cross-validation procedure. One of the commonly applied k values is 5, and it is known as "fivefold cross-validation." The present study also used the fivefold cross-validation for the model development.

Methodology

Dataset collection

The input–output dataset was extracted from the experiment conducted by Ikeagwuani [81]. The mix ratio is a mixture of two experimental designs which are Taguchi array and response surface methodology. The output variable is the CBR value while the input variables comprised the stabilizers and the modified expansive soil properties whose natural geotechnical properties are displayed in Table 1. The modified soil properties are made up of the Atterberg limits and compaction characteristics of the modified expansive soil. Specifically, the Atterberg limits consist of liquid limit (LL), plastic limit (PL) and plasticity index (PI), while the compaction characteristics are the optimum content (OMC) and maximum dry density (MDD). In addition, the stabilizers, which are also part of the input variables comprised of SDA, OPC and QD. In all, there were 8 input variables and 1 output variable. The total number of sample points as shown in Table 2 is 109.

Table 1 Geotechnical properties of natural expansive soil (Ikeagwuani, [81])

S/No	Property		Description
1	Specific gravity		2.66
2	Sand		4.16%
3	Fines	Silt fraction	95.84%
		Clay fraction	
4	Natural moisture content		8.65%
5	Liquid limit		67.1%
6	Plastic limit		23.64%
7	Plasticity index		43.46%
8	Optimum moisture content		23.4%
9	Maximum dry density (BSL)		1.546 g/cm^3
10	AASHTO classification		A-7–6 (33)
11	CBR	Unsoaked	9.41%
		Soaked	5.50%
12	Unconfined compressive strength		152.5kN/m^2
13	Differential free swell		71.6%

MARS model development

The ARESLab toolbox invented by Jekabson [82] for the development of MARS model and installed in MATLAB (2020a) software was used to build the MARS model in this study. The input–output dataset comprising the 8 input variables and 1 output variable as mentioned earlier was used to build the model. The dataset was randomly divided into training and testing datasets. Seventy percent of the input–output dataset was used as training dataset while the remaining thirty percent was used as testing dataset. The 70 percent training dataset was employed to build the model while the 30 percent testing dataset was utilized to evaluate the reliability of the model built with the training dataset. In order to establish a realistic relationship between the soil properties and additives, piecewise linear model was used.

Table 2 Atterberg limits, compaction characteristics and stabilizers of experimental data

Experimental runs	LL	PL	PI	MDD	OMC	SDA (%)	QD (%)	OPC (%)	CBR (%)
1	52.1	32.6	19.5	1.422	26.5	0	0	2	27.57
2	52.1	32.6	19.5	1.422	26.5	0	0	2	23.78
3	40.6	24.3	16.3	1.528	24	0	10	5	26.22
4	40.6	24.3	16.3	1.528	24	0	10	5	30.58
5	40.6	25.3	15.3	1.528	24	0	10	5	24.35
6	38.9	27.3	11.6	1.53	18.9	0	20	8	33.03
7	38.9	27.3	11.6	1.53	18.9	0	20	8	34.03
8	38.9	27.3	11.6	1.53	18.9	0	20	8	32.3
9	44.1	26.8	17.3	1.38	29.5	4	0	2	21.61
10	44.1	26.8	17.3	1.38	29.5	4	0	2	23.7
11	44.1	26.8	17.3	1.38	29.5	4	0	2	19.7
12	33.1	22.4	10.7	1.495	23.2	4	10	5	33.3
13	33.1	22.4	10.7	1.495	23.2	4	10	5	28.6
14	33.1	22.4	10.7	1.495	23.2	4	10	5	34.48
15	33.1	22.4	10.7	1.495	23.2	4	10	5	32.58
16	39.1	30.9	8.2	1.581	22.1	4	20	8	46.23
17	39.1	30.9	8.2	1.581	22.1	4	20	8	45.43
18	39.1	30.9	8.2	1.581	22.1	4	20	8	45.95
19	39.1	30.9	8.2	1.581	22.1	4	20	8	43.42
20	42.1	29.6	12.5	1.43	20.4	8	0	5	30.17
21	42.1	29.6	12.5	1.43	20.4	8	0	5	27.41
22	42.1	29.6	12.5	1.43	20.4	8	0	5	36.3
23	42.1	29.6	12.5	1.43	20.4	8	0	5	36.32
24	33.9	24.6	9.3	1.526	22.4	8	10	8	26.76
25	33.9	24.6	9.3	1.526	22.4	8	10	8	19.69
26	37.8	26.6	11.2	1.553	23	8	20	2	36.55
27	37.8	26.6	11.2	1.553	23	8	20	2	34.35
28	37.8	26.6	11.2	1.553	23	8	20	2	38.78
29	37.8	26.6	11.2	1.553	23	8	20	2	33.93
30	36.7	27.1	9.6	1.434	26	12	0	8	35.1
31	36.7	27.1	9.6	1.434	26	12	0	8	31.95
32	36.7	27.1	9.6	1.434	26	12	0	8	31.19
33	38.9	30	8.9	1.412	26.5	12	10	2	36.23
34	38.9	30	8.9	1.412	26.5	12	10	2	34.89
35	38.9	30	8.9	1.412	26.5	12	10	2	33.54
36	38.9	30	8.9	1.412	26.5	12	10	2	34.85
37	31.9	25.3	6.6	1.506	22	12	20	5	49.22
38	31.9	25.3	6.6	1.506	22	12	20	5	44.39
39	31.9	25.3	6.6	1.506	22	12	20	5	48.2
40	31.9	25.3	6.6	1.506	22	12	20	5	43.5
41	31.9	25.3	6.6	1.418	24.5	16	0	5	43.61
42	37.9	28.7	9.2	1.418	24.5	16	0	5	38.23
43	37.9	28.7	9.2	1.418	24.5	16	0	5	36.37
44	37.9	28.7	9.2	1.418	24.5	16	0	5	38.31
45	36.2	27.8	8.4	1.461	24	16	10	8	38.23
46	36.2	27.8	8.4	1.461	24	16	10	8	36.37
47	36.2	27.8	8.4	1.461	24	16	10	8	38.31
48	35.3	26.6	8.7	1.527	23	16	20	2	38.22
49	35.3	26.6	8.7	1.527	23	16	20	2	35.81
50	35.3	26.6	8.7	1.527	23	16	20	2	39.59

Table 2 (continued)

Experimental runs	LL	PL	PI	MDD	OMC	SDA (%)	QD (%)	OPC (%)	CBR (%)
51	35.3	26.6	8.7	1.527	23	16	20	2	35.73
52	40.3	36.1	4.2	1.401	27.5	20	0	8	55.93
53	40.3	36.1	4.2	1.401	27.5	20	0	8	51.79
54	40.3	36.1	4.2	1.401	27.5	20	0	8	38.05
55	40.3	36.1	4.2	1.401	27.5	20	0	8	48.55
56	33.2	25.1	8.1	1.388	26.6	20	10	2	34.69
57	33.2	25.1	8.1	1.388	26.6	20	10	2	39.85
58	33.2	25.1	8.1	1.388	26.6	20	10	2	38.93
59	45.7	37.2	8.5	1.469	25.5	20	20	5	38.36
60	45.7	37.2	8.5	1.469	25.5	20	20	5	53.78
61	45.7	37.2	8.5	1.469	25.5	20	20	5	32.43
62	39.2	30	9.2	1.695	25.5	0	0	8	29.72
63	39.2	30	9.2	1.695	25.5	0	0	8	31.63
64	33.5	25.5	8	1.777	24	20	0	2	44.34
65	33.5	25.5	8	1.777	24	20	0	2	55.35
66	33.5	25.5	8	1.777	24	20	0	2	41.66
67	33.5	25.5	8	1.777	24	20	0	2	54.18
68	23.6	21.3	2.3	1.55	20	20	0	8	54.2
69	23.6	21.3	2.3	1.55	20	20	0	8	62.18
70	23.6	21.3	2.3	1.55	20	20	0	8	65.29
71	39.1	22.5	16.6	1.402	27.32	0	20	2	23.49
72	39.1	22.5	16.6	1.402	27.32	0	20	2	25.03
73	39.1	22.5	16.6	1.402	27.32	0	20	2	23.26
74	39.1	22.5	16.6	1.402	27.32	0	20	2	24.29
75	42.5	30.3	12.2	1.365	28.76	0	20	8	36.59
76	42.5	30.3	12.2	1.365	28.76	0	20	8	52.29
77	42.5	30.3	12.2	1.365	28.76	0	20	8	47.43
78	42.5	30.3	12.2	1.365	28.76	0	20	8	57.58
79	29.2	21.5	7.7	1.512	23.5	20	20	2	40.05
80	29.2	21.5	7.7	1.512	23.5	20	20	2	42.77
81	29.2	21.5	7.7	1.512	23.5	20	20	2	45
82	21.2	19.1	2.1	1.508	23.3	20	20	8	57.95
83	21.2	19.1	2.1	1.508	23.3	20	20	8	66.75
84	21.2	19.1	2.1	1.508	23.3	20	20	8	46.99
85	37.6	25.7	11.9	1.554	22	10	10	2	35.24
86	37.6	25.7	11.9	1.554	22	10	10	2	38.92
87	37.6	25.7	11.9	1.554	22	10	10	2	30.01
88	37.6	25.7	11.9	1.554	22	10	10	2	42.83
89	32.5	24.6	7.9	1.555	22.32	10	10	8	42.55
90	32.5	24.6	7.9	1.555	22.32	10	10	8	41.95
91	41.6	24.7	16.9	1.555	22.1	0	10	5	24.17
92	41.6	24.7	16.9	1.555	22.1	0	10	5	23.14
93	38.2	31.1	7.1	1.438	25.4	20	10	5	42.09
94	31.2	24.1	7.1	1.438	25.4	20	10	5	48.36
95	31.2	24.1	7.1	1.438	25.4	20	10	5	38.12
96	31.2	24.1	7.1	1.438	25.4	20	10	5	49.43
97	23.1	17.9	5.2	1.437	25.2	10	0	5	44.05
98	23.1	17.9	5.2	1.437	25.2	10	0	5	49.73
99	23.1	17.9	5.2	1.437	25.2	10	0	5	48.25
100	23.1	17.9	5.2	1.437	25.2	10	0	5	57.07

Table 2 (continued)

Experimental runs	LL	PL	PI	MDD	OMC	SDA (%)	QD (%)	OPC (%)	CBR (%)
101	29.9	24.5	5.4	1.555	22.32	10	20	5	37.64
102	29.9	24.5	5.4	1.555	22.32	10	20	5	42.58
103	29.9	24.5	5.4	1.555	22.32	10	20	5	42.27
104	29.9	24.5	5.4	1.555	22.32	10	20	5	49.63
105	33.2	30.3	2.9	1.501	24.7	10	10	5	55.31
106	33.2	30.3	2.9	1.501	24.7	10	10	5	62.68
107	33.2	30.3	2.9	1.501	24.7	10	10	5	62.08
108	33.2	30.3	2.9	1.521	23.6	10	10	5	57.87
109	33.4	30.4	3.0	1.521	23.6	10	10	5	60.23

The piecewise linear parameters used to build the model include the self-interactions (SF) of the basic functions whose values were taken as 1, while the maximum interaction between the basic functions and the maximum basic functions was chosen as 2 and 60, respectively.

Random forest model development

MATLAB (2020a) software was used to build the random forest model in this study. The hyperparameters that were employed to build the model include M and m_{try}. Interestingly, for comparison purpose, more than one value were chosen for both hyperparameters. Three values were chosen for the m_{try} (6; 8; 10) while two values were chosen for M (1000; 5000). It is usually a good practice and also it is highly recommended that the first value of m_{try} be taken as the result of the value obtained when the square root of the predictors $p^{1/2}$ is estimated [83].

Gradient boosting regression model development

The GBR model was built in MATLAB (2020a) software with the 8 input variables and the 1 output variable mentioned earlier. The error index used to build the model was least-squared error. For the maximum number of iteration used to build the model, 2 values, $M \in [1000; 2000]$, were chosen. The two values were chosen for comparison purpose. Furthermore, other parameter values were also chosen to build the model. They include the shrinkage parameter, γ, which was selected as 0.75 and the maximum size of the regression tree which was selected as 8. It is worth noting that studies performed by Hastie et al. [74] and Persson et al. [69] revealed that between 4 and 8 are highly suitable for boosting techniques. Finally, cost complexity pruning was employed to prune the trees to obtain the actual size of the trees.

Model performance metrics

The three developed models performance were validated using statistical error metric that included the coefficient of determination $\left(R^2\right)$ and mean absolute percentage error (MAPE). Other error metrics used are root mean square error (RMSE), mean squared error (MSE) and mean absolute error (MAE). The expressions for the error metrics are shown in Eqs. (18 – 22).

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i^a - y_i^p\right)^2}{\sum_{i=1}^{N}\left(y_i^a - \bar{y}^a\right)^2} \tag{18}$$

$$\text{MAPE} = \frac{\sum_{i=1}^{N}\left|\frac{y_i^a - y_i^p}{y_i^a}\right|}{N} \times 100 \tag{19}$$

$$\text{RMSE} = \left[\frac{\sum_{i=1}^{N}\left(y_i^p - y_i^a\right)^2}{N}\right]^{1/2} \tag{20}$$

$$\text{MSE} = \frac{\sum_{i=1}^{N}\left(y_i^p - y_i^a\right)^2}{N} \tag{21}$$

$$\text{MAE} = \frac{\sum_{i=1}^{N}\left|y_i^p - y_i^a\right|}{N} \tag{22}$$

where N symbolizes the testing or training dataset number, y_i^p represents the predicted values, y_i^a and $\bar{y}^a$ denotes the actual value and mean of the actual values, respectively.

Table 3 Equation of basis functions with their corresponding coefficients of the built MARS model

Basis functions	Equation	Coefficient
BF1	max(0,9.2—PI)	5.979
BF2	max(0,1.402—MDD) * max(0, LL—40.3)	169.984
BF3	max(0, MDD —1.402) * max(0,10—SDA)	−7.179
BF4	max(0, MDD —1.402) * max(0, PL—24.5)	10.409
BF5	max(0,7.1—PI)	−4.381
BF6	max(0, MDD —1.402) * max(0,5—OPC)	7.232
BF7	max(0,10—QD) * max(0, PI—9.2)	−0.390
BF8	max(0,22.32—OMC)	3.056
BF9	max(0, PI—9.2) * max(0, LL -38.9)	0.237

Results and discussion

MARS model performance

The BFs together with their coefficients for the developed MARS model are displayed in Table 3. As evident in Table 3, there are 9 BFs. The 9 BFs have no interaction terms, which indicate that the developed MARS model is simply an additive model, and that interaction terms have no role in the model. The interpretable mathematical equation for the developed MARS is expressed as:

$$\begin{aligned} CBR = {} & 32.040401 + 5.979359 * BF1 + 169.9843 * BF2 - 7.1787649 * BF3 + 10.409494 * BF4 \\ & - 4.381248 * BF5 + 7.2324356 * BF6 - 0.3900035 * BF7 + 3.0556179 * BF8 + 0.23687529 * BF9 \end{aligned} \quad (23)$$

Furthermore, the results obtained for the error indices of the MARS model as presented in Table 4 clearly indicate a fairly high R^2 value for both the training data (0.7985) and the testing data (0.73089). Other error indices presented in Table 4 such as MSE, RMSE, MAE and MAPE all have relatively small values, which indicated that the model has good predictive power, and can be utilized for the prediction of the CBR of expansive soil modified with the three stabilizers

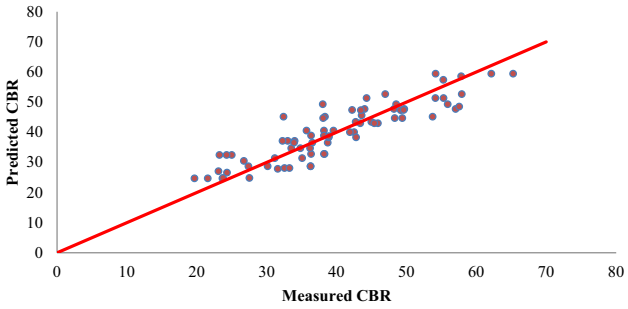

Fig. 1 Scatter plot for measured vs. predicted CBR values for MARS model training dataset

(SDA, QD and OPC) mentioned earlier. More so, the scatter plots of the MARS predicted vs. measured CBR shown in Figs. 1 and 2 for the training and testing data, respectively, reinforced the assertion that the model possesses good predictive power. From Figs. 1 and 2, it can be seen that most of the points lie or are pretty close to the line of equality. This shows that the model can be relied upon for the prediction of CBR of the modified expansive soil.

Random forest performance

The results of the model performance error indices are presented in Table 5. As can be seen from Table 5, three values of m_{try} (6; 8; 10) and two values of M (1000; 5000) were used for the random forest model development. As evident from Table 5, the number of M has no significant difference in the R^2 values obtained for both the training and testing data error. Similar results were also obtained for the other error indices displayed in Table 5 for both the training and testing data errors. In fact, the error indices appear identical for both values of M considered in the study. However, the best prediction, in the case of M equal to 1000, was observed when m_{try} was 10. Similarly, the best prediction, in the case

Table 4 MARS model error value

Training eror				
MSE	RMSE	MAE	MAPE	R^2
22.1401	4.70533	3.8075	10.2538	0.7985
Testing error				
MSE	RMSE	MAE	MAPE	RSQUARED
33.4118	5.7803	4.50106	12.5391	0.73089

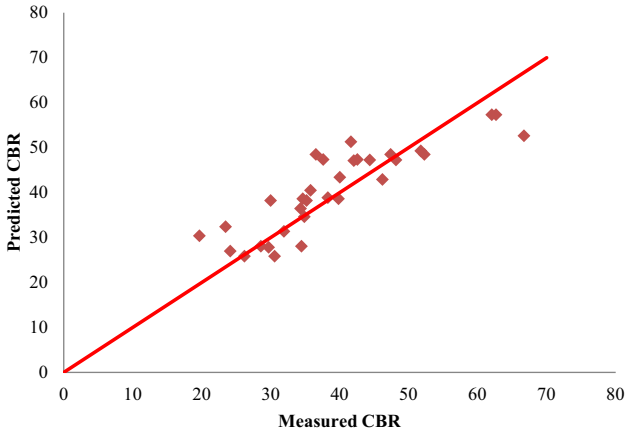

Fig. 2 Scatter plot for measured vs. predicted CBR values for MARS model testing dataset

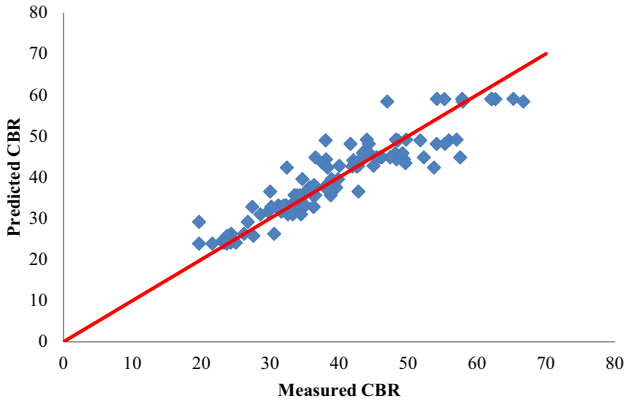

Fig. 3 Scatter plot for measured vs. predicted CBR values for random forest model training dataset

of *M* equal to 5000, was recorded when m_{try} was 8. The differences in the error metrics in both cases are marginal. This indicates that both cases are well suited for the development of the model. For both cases, the R^2 values for the training data are high (greater than 0.8) and that of the testing data is also fairly high (greater than 0.75). This is an indication that the model has good predictive power and can be used to predict the CBR of modified expansive soil.

In addition, the scatter plots of the measured vs. predicted CBR shown in Figs. 3 and 4, respectively, for the training and testing datasets for the case of when *M* is equal to 1000 and m_{try} is equal to 10 clearly helping to reinforce the assertion that the model has good predictive ability. In the scatter plots, most of the data points lie on the line of equality. The other points that do not lie on the line of equality are not too far from it. This also indicates that the model can be relied upon for the prediction of the CBR of modified expansive soil.

Gradient boosting regression performance

Table 6 shows the result of the error indices when only *M* is 1000. The error indices when *M* is 2000 was not shown because the model tends to be over-fitted at *M* equal to 2000, thus lacking in generalization ability. As can be seen from Table 6, both training and testing data have a moderately high R^2 values (greater than 0.7). This indicates that the gradient boosting machine model has good predictive ability and can be used for the prediction of CBR of modified expansive soil in the design of flexible pavement. Furthermore, Figs. 5 and 6 which show the scatter plots for the measured vs. predicted CBR, respectively, for the training and testing dataset also revealed that the model possesses

Table 5 Random forest model error value

Training error						Training error					
Number of regression trees, $M = 1000$						Number of regression trees, $M = 3000$					
m_{try}	MSE	RMSE	MAE	MAPE	R^2	m_{try}	MSE	RMSE	MAE	MAPE	R^2
6	17.58	4.193	3.06	7.784	0.846	6	17.60	4.196	3.062	7.759	0.846
8	17.43	4.175	3.02	7.636	0.847	8	17.45	4.177	3.023	7.655	0.847*
10	17.33	4.163	3.02	7.657	0.848*	10	17.44	4.177	3.024	7.655	0.847
OOB error						OOB error					
Number of regression trees,$M = 1000$						Number of regression trees, $M = 3000$					
m_{try}	MSE	RMSE	MAE	MAPE	R^2	m_{try}	MSE	RMSE	MAE	MAPE	R^2
6	28.49	5.338	3.970	10.189	0.751	6	28.53	5.341	3.974	10.161	0.751
8	28.36	5.326	3.914	10.005	0.752	8	28.32	5.321	3.931	10.057	0.752*
10	28.23	5.314	3.923	10.038	0.753*	10	28.49	5.337	3.946	10.097	0.751

Key: * Signifies the best prediction

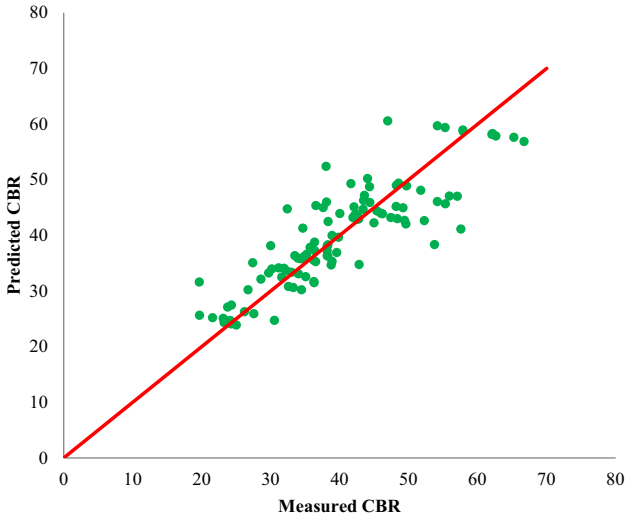

Fig. 4 Scatter plot for measured vs. predicted CBR values for random forest model testing dataset

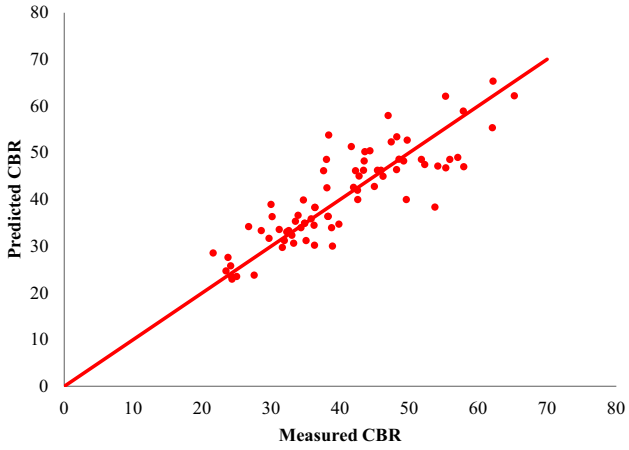

Fig. 5 Scatter plot for measured vs. predicted CBR values for gradient boosting model training dataset

fairly good predictive ability as most of the data points lie on the line of equality.

Comparison between MARS, random forest and GBM model performance

Figures 7 and 8 were used to show, respectively, the comparisons of the training and testing errors among the models developed in this study. From Figs. 7 and 8, the errors metrics used to evaluate the performance of the three models (MARS, random forest and GBM) revealed that the random forest model had the best predictive ability, followed by the MARS model and lastly the gradient boosting machine model. In addition, the error values from all the error metrics in Fig. 8 show that the MARS model has the best generalization ability due to its smaller error value in comparison with that for other models. Similar results are also obtained in

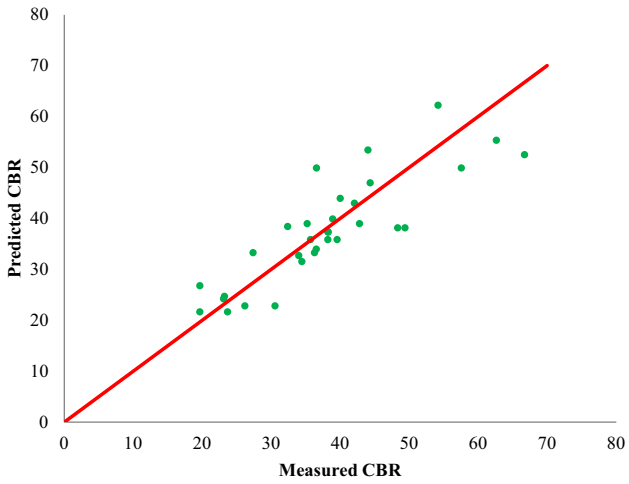

Fig. 6 Scatter plot for measured vs. predicted CBR values for gradient boosting model testing dataset

Table 6 Gradient boosting machine model error value

Training error					
Iteration number = 1000					
γ	MSE	RMSE	MAE	MAPE	R^2
0.75	22.1401	4.70533	3.8075	10.2538	0.7985
Testing error					
Iteration number = 1000					
γ	MSE	RMSE	MAE	MAPE	R^2
0.75	33.4118	5.7803	4.50106	12.5391	0.73089

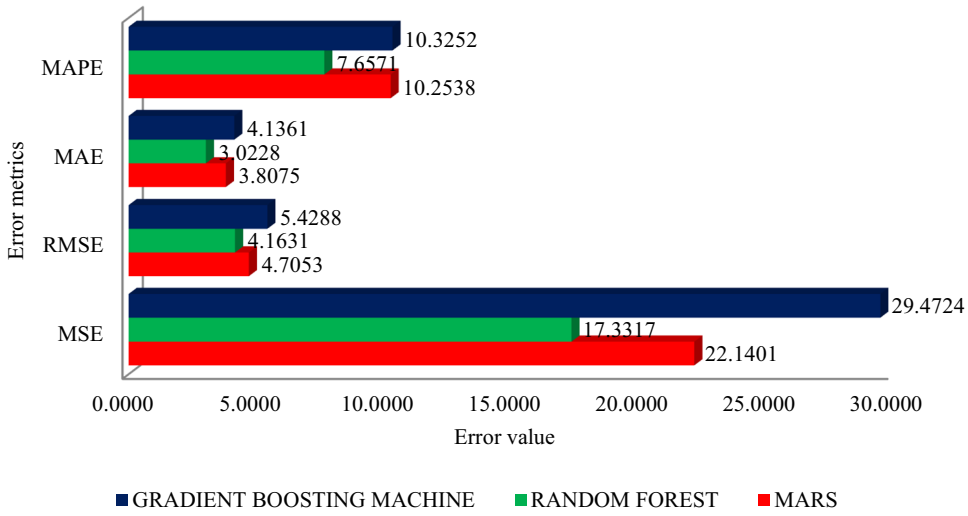

Fig. 7 Comparison of training errors between MARS, random forest and gradient boosting models

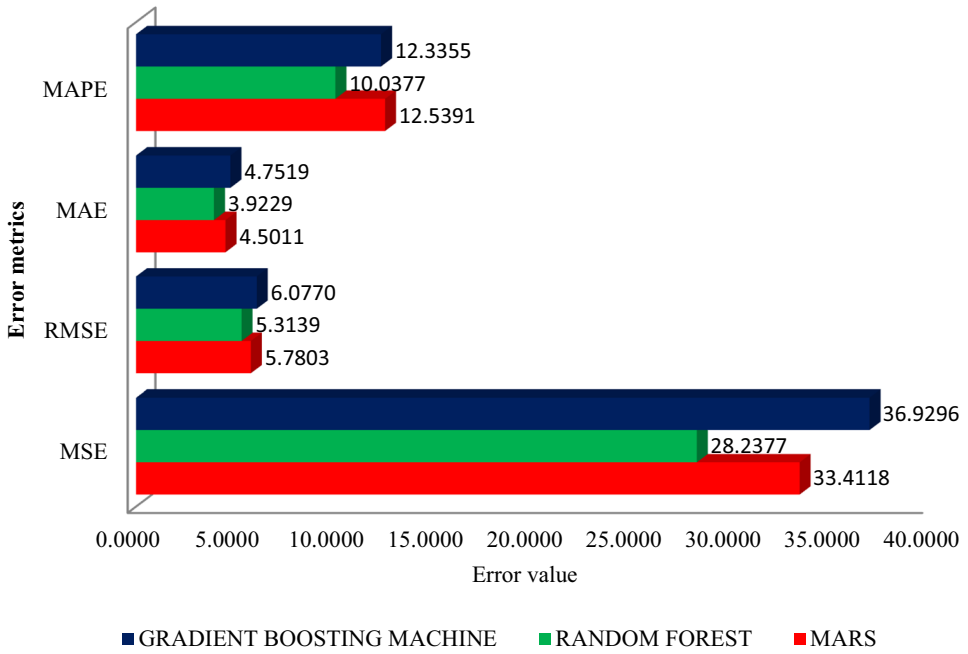

Fig. 8 Comparison of testing errors between MARS, random forest and gradient boosting models

Fig. 9 which is used to show the comparison of R^2 values for the training and testing dataset of the MARS, random forest and gradient boosting machine models. From Fig. 9, the R^2 values for the training and testing data of the random forest were observed to be higher than that obtained for the MARS and GBM models. This further indicates that the random forest has superior predictive ability than MARS and GBM models and will perform better in practice.

Conclusions

The design and construction of safe flexible pavement, more often than not, relies on the utilization of reliable estimation or determination of the CBR of subgrade such as modified expansive soil. Regrettably, the CBR test procedure used for the determination of CBR of subgrade is often challenging due to the enormous time involved

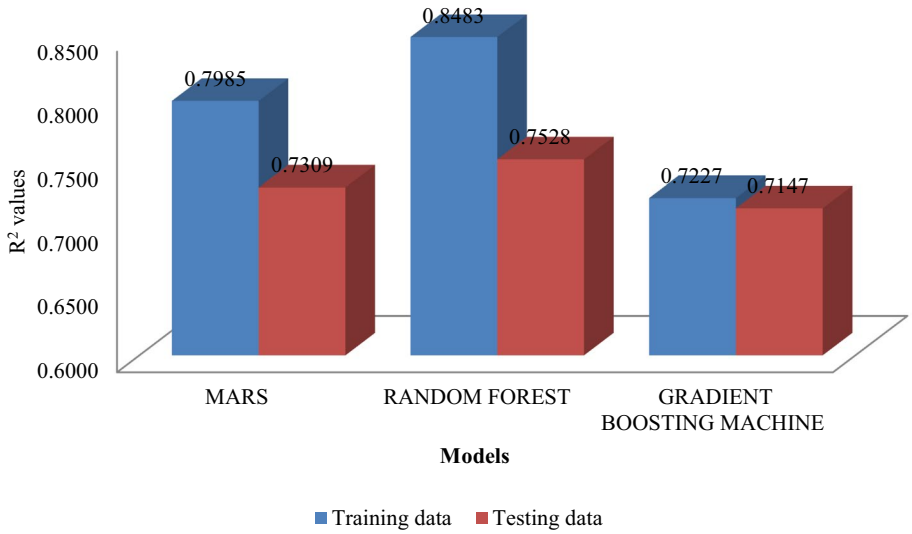

Fig. 9 Comparison of R^2 values for the training and testing dataset of the MARS, random forest and gradient boosting machine models

in performing the test. This, therefore, necessitated the need to explore alternative procedures such as the development of predictive models to estimate the CBR of modified expansive soil subgrade. In the light of this discussion, three models, in the present study, were developed using three machine learning techniques, namely, MARS, random forest and gradient boosting machine to predict the CBR of expansive soil subgrade blended with SDA, OPC and QD. The developed models were compared with each other and the following enumerated conclusions are drawn:

1. The random forest model that gave the best prediction was found when the hyperparameters M and m_{try} were equal to 1000 and 10, respectively.
2. The number of trees used in the development of the random forest models has no significant effect on the predicted output.
3. The gradient boosting machine model that resulted in the best prediction was recorded when the hyperparameters γ and M were equal to 0.75 and 1000, respectively.
4. The developed MARS model, which is interpretable, is an additive model with no interaction terms.
5. The random forest model gave better predictions than the MARS and gradient boosting models and it is highly recommended for the prediction of modified expansive soil.

This present study in which three machine learning techniques (MARS, random forest and gradient boosting machines) were used to develop predictive models to estimate the CBR of enhanced expansive soil subgrade has shown the capacity of machine learning to develop models capable of predicting geotechnical soil parameters. Furthermore, from the study, random forest technique, apparently due to the random trees in its algorithm, was found to have the highest predictive power among the three models considered in this study. Notably, the random trees, which are unique in the algorithm of random forest technique, contributed to the superior power of the random forest technique. Moreover, the number of trees has little effect on the predictive model developed with random forest algorithm. Finally, this study has shown that the effect of additives should be incorporated in the development of models for geotechnical soil properties.

Acknowledgements The author is highly indebted to the anonymous reviewer of this paper who assisted greatly to improve the quality of this paper.

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Wu J, Liu Q, Deng Y, Yu X, Feng Q, Yan C (2019) Expansive soil modified by waste steel slag and its application in subbase layer of highways. Soils Found 59:955–965
2. Ikeagwuani CC, Obeta IN, Agunwamba JC (2019) stabilisation of black cotton soil subgrade using sawdust ash and lime. Soils Found 59(1):162–175
3. Ikeagwuani CC, Nwonu DC (2019) Emerging trends in expansive soil stabilisation. J Rock Mech Geotech Eng 11:423–440

4. Tiwari N, Satyam N, Patva J (2020) Engineering characterisitcs and performance of polypropylene fibre and silican fume treated expansive soil subgrade. Int J Geosynth Ground Eng 6(18):1–11
5. Bhuvaneshwari S, Robinson RG, Gandhi SR (2020) Effect of functional group of the inorganic additives on index and microstructural properties of expansive soil. Int J Geosynth Ground Eng. https://doi.org/10.1007/s40891-020-00235-w
6. Jain AK, Jha AK (2020) Geotechnical behaviour and micro-analyses of expansive soil amended with marble dust,. Soils Found 60:737–751
7. Ikeagwuani CC, Nwonu DC, Onah HN (2020) Min-max fuzzy goal programming - Taguchi model for multiple additives optimization in expansive soil improvement. Int J Numer Anal Methods Geomech. https://doi.org/10.1002/nag.3163
8. Ikeagwuani CC (2019) Comparative assessment of the stabilization of lime-stabilized lateritic soil as subbase material using coconut shell ash and coconut husk ash. Geotech Geol Eng 37(4):3065–3076
9. Onyelowe KC (2019) Nanosized palm bunch ash (NPBA) stabilisation of lateritic soil for construction purposes. Int J Geotech Eng 13(1):83–91
10. Onyelowe KC, Duc BV (2018) Durability of nanostructured biomassess ash (NBA) stabilized expansive soils for pavement foundation. Int J Geotech Eng. https://doi.org/10.1080/19386362.2017.1422909
11. Onyelowe KC, Vsn DB, Ubachukwu O, Ezugwu C, Salahudeen B, Van MV, Ikeagwuani CC, Ahmadi T, Sosa F, Wu W, Duc TT, Eberemu A, Ducc TP, Barah O, Ikpa C, Orji F, Alaneme G, Amanamba E, Ugwuanyi H, Sai V, Kadurumba C, Subburaj S, Ugorji B (2019) Recycling and reuse of solid wastes: a hub for ecofriendly, ecoefficient and sustainable soil, concrete, wastewater and pavement reengineering. Int J Low-Carbon Technol 14(3):440–451
12. Ikeagwuani CC, Nwonu DC (2021) Integration of data envelopment analysis and AL-Rafaie and Al-Tahat model in Taguchi method for the optimization of additives in expansive soil treatment. Geomech Geoeng. https://doi.org/10.1080/17486025.2021.1912402
13. Soltani A, Deng A, Taheri A, Mirzababaei M (2018) Rubber powder-polymer combined stabilization of South Australian expansive soils. Geosynth Int 25(3):304–321
14. Estabragh AR, Rafatjo H, Javadi AA (2014) Treatment of an expansive soil by mechanical and chemical techniques. Geosynth Int 21(3):233–243
15. Etim RK, Eberemu OA, Osinubi KJ (2017) Stabilization of black cotton soil with lime and iron ore tailings admixture. Transport Geotechnics 10:85–95
16. Olgun M (2013) The effects and optimization of additives for expansive clays under freeze-thaw conditions. Cold Reg Sci Technol 93:36–46
17. Ikeagwuani CC, Nwonu DC (2020) Application of fuzzy logic and grey based Taguchi approach for additives optimization in expansive soil treatment. Road Mater Pavement Design. https://doi.org/10.1080/14680629.2020.1847726
18. Nwonu DC, Ikeagwuani CC (2021) Microdust effect on the physical condition and microstructure of tropical black clay. Int J Pavement Res Technol 14(1):73–84
19. Ikeagwuani CC, Agunwamba JC, Nwankwo CM, Eneh M (2020) Additives optimization for expansive soil subgrade modification based on Taguchi grey relational analysis,". Int J Pavement Res Technol. https://doi.org/10.1007/s42947-020-1119-4
20. Duque J, Fuentes W, Rey S, Molina E (2020) Effect of grain size distributionon Californai bearing ratio (CBR) and modified proctor parameters for granular materials. Arab J Sci Eng 45:8231–8238
21. Taha S, Gabr A, El-Badawy S (2019) Regression and neural network models for California bearing ratio prediction of typical granular materials in Egypt. Arab J Sci Eng 44:8691–8705
22. Bristish Standard Institute (1990) Methods of testing soils for civil engineering purposes, London: BS 1377, Part 4
23. Sreelekshmypillai G, Vinod P (2019) Prediction of CBR of fine grained soils at any rational compactive effort. Int J Geotech Eng 13(6):560–565
24. Black WPM (1962) A method of estimating the CBR of cohesive soils from plasticity data. Geotechnique 12:271–282
25. Yildirim B, Gunaydin O (2011) Estimation of California bearing ratio using soft computing systems. Experts Syst Appl 38:6381–6391
26. Bassey OB, Attach IC, Ambrose EE, Etim RK (2017) Correlation between CBR values and index properties of soils? A case study of Ibiono, Oron and Onna in Akwa Ibom State. Resour Environ 7(4):94–102
27. Singh D, Reddy KS, Yadu L (2011) Moisture and compaction baseed statisitcal model for estimating CBR of fine grained subgrade soils. Int J Earth Sci Eng 4(6):100–1034
28. Ramasubbarao GV, Sankar GS (2013) Predicting soaked CBR value of fine grained soils using index and compaction characterisitcs. Jordan J Civil Eng 7(3):354–360
29. Aderinola OS, Oguntoyinbo E, Quadri AI (2017) Correlation of California bearing ratio value of clays with soil index and compaction characterisitics. Int J Sci Resour Innov Technol 4(4):12–22
30. Aderinola OS (2017) Prediciting the Californai bearing ratio value of low compressible clays with its index compaction characterisitics. Int J Sci Eng Resour 8(5):1460–1472
31. Taskiran T (2010) Prediction of California bearing ratio (CBR) of fine grained soils by AI methods. Adv Eng Softw 41:886–892
32. NCHRP (2021) "National Cooperative Highway Research Program. Guide for mechanisitic and empiricla - design for new and rehabilitaed pavement structures, final document," *Appendix CC-1: Correlation of CBR values with soil index properties: West Univsersity Avenue Champaign, ILLinois: Ara, Inc.,*
33. I. G. Farias, W. Araujo and G. Ruiz, "Prediction of California bearing ratio from index properties of soils using parametric and non-parametric properties models," *Geotechnical and geological engineering,* vol. 36, no. https://doi.org/10.1007/s10706-018-0548-1, pp. 3485–3498, 2018.
34. Tenpe AR, Patel A (2020) Utilization of support vector models and gene expression programming for soil strength modeling. Arab J Sci Eng 45:4301–4319
35. Friedman JH (1991) Multivariate adaptive regression splines. Ann Stat 19(1):1–141
36. Goh ATC, Zhang W, Zhang Y, Xiao Y (2018) Determination of earthe pressure balance tunnel-related maximum surface settlement: a multivariate adaptive regression splines approach. Bull Eng Geol Env 77:489–500
37. Zhang W, Goh AT, Zhang Y (2016) Multivariate adaptive regression splines application for multivariate geotechnical problems with big data. Geotech Geol Eng 34(1):193–204
38. Acciani C, Fucilli V, Sardaro R (2011) Data mining in real estate appraisal a model tree and multivariate adaptive regression spline approach. AESTIMUM 58:27–45
39. Amit Y, Geman D (1997) Shape quantization and recognition with randomized trees. Neural Comput 9(7):1545–1588
40. Breiman L (2001) Random forests. Mach Learn 45:5–32
41. Ho T (1995) Random decision forest," in *Proceedings of the 3rd International conference on document analysis and recognition*, Montreal, QC, 14–16: 278–282
42. Ho T (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844
43. Breiman L (1996) Bagging predictors. Mach Learn 26(2):123–140

44. Breiman L (1996) Heuristics of instability and stabilization in model selection. Ann Stat 24(6):2350–2383
45. Buhlmann P, Yu B (2002) Analyzing bagging. Ann Stat 30(4):927–961
46. Chen X, Ishwaran H (2012) Random forests for genomic data analysis. Genomics 99:323–329
47. Efron B, Tibshirani RJ (1993) An introduction to the bootstrap. Chapman & Hall, London
48. Ishwaran H, Kogalur U, Blackstone E, Lauer M (2008) Random survival forest. Annals Appl Stat 2(3):841–860
49. Segal M, Xiao Y (2011) Multivariate random forests. WIRESs Data Min Knowl Discov 1:80–87
50. Amaratunga D, Cabrera J, Lee Y-S (2008) Enriched random forests. Bioinformatics 24(18):2010–2014
51. Meinshausen N (2006) Quantile regression forests. J Mach Learn Res 7:983–999
52. Xu R (2013) "Improvement to random forest methodology," *PhD thesis, Iowa State University, Iowa*
53. Yesilkanat CM (2020) Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm. Chaos, Solut Fractals 140(110210):1–8
54. Yao H, Li X, Pang H, Sheng L, Wang W (2020) Application of random forest algorithm in hail forecasting over Shandong Peninsula. Atmos Res. https://doi.org/10.1016/j.atmosres.2020.105093
55. Chun P, Ujike I, Mishima K, Kusumoto M, Okazaki S (2020) Random forest-based evaluation technique for internal damage in reinforced concrete featuring multiple nondestructive testing results. Constr Build Mater 253(119238):1–11
56. Khalilia M, Chakraborty S, Popescu M (2011) Predicting disease risks from highly imbalanced data using random forest. BMC Med Inform Decis Mak 11(51):1–13
57. Yesilkanat CM (2020) Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm. Chaos, Solitions Fractals 140(110210):1–8
58. Goldstein B, Polley E, Briggs F (2011) Random forests for genetic association studies. Stat Appl Genet Mole Biol 10(1):1–34
59. Jiang P, Wu H, Wang W, Ma W, Sun X, Lu Z (2007) Mipred: classification of real and pseudo microrna precursors using random forest prediction model with combined features. Nucleic Acids Res 35(2):339–344
60. Ward M, Pajevic S, Dreyfuss J, Malley J (2006) Short-term prediction of mortality in patients with systemic lupus erythematosus: Classification of outcomes using random forests. Arthritis Rheum 55:74–80
61. Gong H, Sun Y, Hu W, Polaczyk P, Huang B (2019) Investigating impacts of asphalt mixture properties on pavement performance using LTTP data through random forests. Constr Build Mater 204:203–212
62. Zhang J, Ma G, Huang Y, Sun J, Asiani F, Nener B (2019) Modelling uniaxial compressive strength of lightweight self-compacting concrete using random forest regression. Constr Build Mater 210:713–719
63. Gong M, Bai Y, Qin J, Wang J, Yang P, Wang S (2020) Gradient boosting machine for predicting return temperature of district heating system: a case study for residential buildings in Tianjin. J Build Eng 27:1–9
64. Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(3):1189–1232
65. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
66. Freund Y, Freund Y, Shapire RE (1996) "Experiements with a new boosting algorithm," in *Machine learning: proceedings of the thirteenth international conference*, San Francisco: Morgan Kaufmann Publishers
67. Zhang M, Gong H, Jia X, Xiao R, Jiang X, Ma Y, Huang B (2020) Analysis of critical factors to asphalt overlay performance using gradient boosted models. Constr Build Mater 263(120083):1–9
68. De Clereq D, Wen Z, Fei F (2019) Determinants of efficiency in anaerobic bio-waste co-digestion facilities: a data envelopment analysis and gradient boosting appraoch. Appl Energy 253(113570):1–11
69. Persson C, Bacher P, Shiga T, Madsen H (2017) Multi-site solar power forecasting using gradient boosted regression. Sol Energy 150:423–430
70. Kaloop MR, Kumar D, Sammui P, Hu JW, Kim D (2020) Compressive strength prediction of high-performance concrete using gradient tree boosting machine. Constr Build Mater 264:1–11
71. Barua L, Zou B, Noruzoliaee M, Derrible S (2020) A gradient boosting appraoch to understanding airport runway and taxiway pavement deterioration,". Int J Pavment Eng. https://doi.org/10.1080/10298436.2020.1714616
72. Thai DK, Tu TM, Bui TQ, Bui TT (2019) Gradient tree boosting machine learning on predicting the failure modes of the RC panels under loads. Eng Comput. https://doi.org/10.1007/s00366-019-00842-w
73. Natekin A, Knoll A (2013) Gradient boosting machines, a tutorial. Front Neurorobot 7(21):1–21
74. Hastie T, Tibshirani R, Friedman J (2009) The elements of statisitical learning, New York. Springer, NY
75. Ikeagwuani CC, Nwonu DC, Nweke CC (2021) Resilient modulus descriptive analysis and estimation for fine-grained soils using multivariate and machine learning methods. Int J Pavement Eng. https://doi.org/10.1080/10298436.2021.1895993
76. Kor K, Altun G (2020) Is support vector regression method suitabe for predicting rate of penetration? J Petrol Sci Eng 194:1–18. https://doi.org/10.1016/j.petrol.2020.107542
77. Dietterich T (1995) Overfitting and undercomputing in machine learning. ACM Comput Surv 27(3):326–327
78. Flach P (2012) Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press, UK
79. Saud S, Jamil B, Upadhyay Y, Irshad K (2020) Performance improvement of empirical models for estimation of global solar radiation in india: a k-fold cross-validation approach. Sustain Energy Technol Assess. https://doi.org/10.1016/j.seta.2020.100768
80. Xiong Z, Cui Y, Liu Z, Zhao Y, Hu M, Hu J (2020) Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation. Comput Mater Sci 171:1–12. https://doi.org/10.1016/j.commatsci.2019.109203
81. Ikeagwuani CC (2019) Optimisation of additives for expansive soil reinforcement," Unpublished PhD thesis, 2019
82. Jekabsons G (2010) Areslab: Adaptive regression splines toolbox for matlab/octave
83. Liaw A, Wiener M (2002) Classification and regression by Random forest. R News 2:18–22