Statistical analysis and prediction of spatial resilient modulus of coarse-grained soils for pavement subbase and base layers using MLR, ANN and Ensemble techniques

Chijioke Christopher Ikeagwuani, Donald Chimobi Nwonu

Abstract
Prediction methods for the estimation of the resilient modulus ($M_R$) of unbound granular materials continue to be fraught with challenges associated with the in situ dependency of the existing methods, which makes them unreliable when considering the spatial variability of soil properties. Moreover, the artificial intelligence methods, which prove to be superior for $M_R$ prediction, indicate that more recent and hybrid machine learning methods are more apt. Consequently, a comparative analysis was executed in the present study using advanced statistical approaches and a simple routine method for the prediction of the $M_R$ of coarse-grained soils used as unbound granular materials in highway pavements. This study considered also the spatial variability of the routine soil properties based on the long-term pavement performance database. The result of the descriptive analysis conducted using principal component analysis revealed that the variation in $M_R$ is due to three principal factors, which include the effect of moisture on the fines content, effect of coarse particles on the compaction characteristics and the effect of the soil stress state. Subsequently, prediction of the $M_R$ using multiple linear regression was unreliable based on residual analysis and multi-collinearity problems. Thus, predictive modeling of $M_R$ was then executed using artificial neural network (ANN) and ensemble techniques (random forest and gradient boosting machine), which resulted in very high $R^2$ values above 0.9, considered apt for MEPD. The ANN model was found to provide a superior prediction performance than the ensemble techniques.

Keywords Artificial neural network · Coarse-grained soils · Gradient boosting · Principal component analysis · Random forest · Resilient modulus

Introduction

In pavement engineering, the subbase and base course layers of highway pavements are very crucial components that ensure the safe transfer of traffic loads to the subgrade. Their major constituents are typically in form of unbound granular materials, which could be soils or other borrow materials considered to be of adequate strength to ensure safe load bearing throughout the design life of the pavement. The soils used as unbound granular materials are typically the coarse-grained soils. The suitability of these coarse-grained soils for load transmission to the subgrade is realistically measured by an elastic modulus, known as the resilient modulus ($M_R$). The expression for the $M_R$ of soil materials is shown in Eq. 1

$$M_R = \frac{\sigma_d}{\varepsilon_r}$$

where the respective $\sigma_d$ and $\varepsilon_r$ denote the average deviator stress and resilient strain in the last few cycles of the repeated load triaxial test (RLTT).

Interestingly, a lot of research has been done to determine the factors that significantly influence the $M_R$ of coarse-grained soils. Detailed information is reported in the literature. The upshots of studies on the influential factors affecting the $M_R$ show that aggregate gradation/morphology, density, stress state and moisture content are all crucial for granular materials [1, 2]. These aforementioned factors...
are from routine laboratory soil properties and the idealized stress state the soil is subjected to in situ. These factors have been shown to have significant influence on the $M_R$ of unbound granular materials. More specifically, soil gradation is known to affect the stiffness response with higher $M_R$ occurring in much coarser aggregates (>5 mm), while in terms of the morphology, higher stiffness is prevalent in aggregates with irregular shapes [3–6]. Also, in terms of the density, it has been shown that the variation of the compaction energy that changes the compaction density has a considerable effect on the $M_R$ of unbound granular materials [7, 8]. The moisture sensitivity of unbound granular materials has been suggested to impact the stiffness behavior, which is mainly attributed to the influence of moisture on fines and the workability [9, 10].

Furthermore, the already highlighted factors which affect the $M_R$ of soils have been found to be very useful in the predictive modeling of the $M_R$. The place of the factors affecting the $M_R$ of soils in pavement design cannot be overemphasized because more often than not, the $M_R$ used for mechanistic-empirical design are based on predictions since the laboratory method of determining $M_R$ through RLTT is cumbersome. The methods presented in the literatures for the prediction of the $M_R$ of soils include the stress-dependent models, models based on correlations from alternative in situ testing (like those of dynamic cone penetration, light weight deflectometer, geo-gauge, etc.), correlations with routine soil properties, predictions based on artificial intelligence particularly the artificial neural network (ANN) and predictions based on numerical modeling [11–17].

It is imperative to note that stress-dependent models require the calibration of the regression constants from the routine properties of the soil samples at a particular site, and therefore the spatial variability of soil properties cannot be accounted for. This is a major limitation to wide adoption. In the case of methods based on in situ testing, some significant advancements have been made in recent years to overcome some of the limitations including the assumption of constants (such as Poisson’s ratio and shape factor) in the light weight deflectometer test [18] and the estimation of $M_R$ only at a single stress state in the dynamic cone penetration methods [19]. However, these improvements were only achieved for particular soil types and site conditions, thereby limiting the applicability for wider use. In the attempt to assuage the effect of spatial variability, Liu et al. [20] used data from different sites based on cone penetration test for model calibration and employed an advanced multivariate technique. Unfortunately, the $M_R$ predictions were found to be biased when a different dataset was applied. This pinpoints the limitations of in situ methods for $M_R$ estimation when considering the spatial variability in soil properties. Likewise in numerical models, there are still fundamental challenges associated with the simulation and prediction of $M_R$ using the discrete element method. Some of these include irreproducibility of particle morphology (shapes and size), anisotropy and mineralogy as well as inability to capture moisture regime in granular materials [15, 17, 21–23]. All these limitations subsist under varying in situ conditions, and thus, the problem of spatially varying soil properties remains a major concern.

However, the adoption of AI models, particularly the ANN, provides a promising approach to assuage the problem of soil spatial variability. Typically, the prediction of $M_R$ based on AI models is a back-calculation technique as explicitly described by Ikeagwuani et al. [24], which makes the calibration of the mechanistic empirical pavement design (MEPD) model constants independent of site specific data. Thus, the MEPD model constants can easily be recalibrated for any new site data, given the routine soil properties. Consequently, the accuracy of the prediction is dependent on the robustness of the AI method adopted for the initial computation of the $M_R$ prior to implementation in the MEPD. The most recent studies on $M_R$ prediction based on various AI methods strongly highlight the need for hybrid and more thorough AI approaches to improve the prediction accuracy in comparison with routine AI methods, like the ANN. This is pertinent because the spatial variability of soil properties has been shown to have a great influence on the predictions of the $M_R$ of coarse-grained soils [25]. Heidarabadizadeh et al. [26] applied two AI methods, which include the hybrid support vector machine and colliding bodies optimization for the prediction of the $M_R$ of unbound subbase materials. The outcome of the study highlighted a superior performance of the hybrid method, surpassing the ANN method and ultimately culminated in excellent $R^2$ values above 0.97. Another study by Ghorbani et al. [27] utilized a hybrid technique based on genetic algorithm (GA) and ANN for the estimation of pavement subgrade $M_R$. The obtained results clearly indicated that the hybrid method resulted in excellent prediction with $R^2$ of 0.97 and grossly outperformed other methods which include the GA, genetic programming and nonlinear regression. In a related study, predictions of $M_R$ of fine-grained subgrade soils were executed using various machine learning methods and the upshots of the study clearly show that hybrid adaptive neuro-fuzzy inference system prevailed over gradient boosting regression and ANN with excellent $R^2$ above 0.98 for both the training and testing set [24]. Similar outcomes have also been widely reported [28–31]. Instructively, these AI models seem to be highly reliable, but their performance when considering the spatial variability of soil properties for predicting the $M_R$ of unbound granular materials has not been properly addressed.

In view of the foregoing discussion, the novelty of the present study is in the comparative evaluation of advanced statistical approaches and simple routine methods for the prediction of the $M_R$ of coarse-grained soils used as unbound...
granular materials in highway pavements. Moreover, the analysis captured the performance when considering the spatial variability in the soil properties. The first approach involves the descriptive analysis of the $M_R$ of the coarse-grained soils using principal component analysis (PCA). In addition, the predictive modeling of the $M_R$ was executed using multi-linear regression (MLR), ANN and ensemble techniques (random forest and gradient boosting) with the routine soil properties of long-term pavement performance (LTPP) data.

**Innovation of the study**

The present work is relevant to infrastructure systems because the resilient modulus provides the strength parameter for pavement design as per the Mechanistic Empirical Pavement Design Guide. The innovative aspect of the study is the identification of the influential factors for the resilient modulus for coarse-grained soils using multivariate statistics, which has not been previously reported. Moreover, the consideration of the resilient modulus based on these influential factors for predictive modeling using machine learning and evaluating the performance of ensemble techniques. The study result shows that traditional ANN method is still superior to the newer ensemble techniques.

**Principal component analysis**

When presented with a large array of data, it is often very difficult, if not impossible, to make meaningful deductions from the data by mere inspection or simple evaluation. Fortunately, very robust data analysis methods have been developed in recent years to handle large datasets. One of such methods, which is very potent for dimension reduction is a multivariate statistical technique known as principal component analysis (PCA), developed by Pearson and Hotelling [32, 33]. The method uses the principles of eigenvector analysis and coordinate transformation to reduce a set of $m$ by $n$ multidimensional data into a smaller subset known as principal components (PCs). The PCs are as a matter of fact, a set of linear combinations of the original $n$ correlated variables, which have been transformed into uncorrelated variables. Interestingly, these new set of variables (PCs) possess the ability to explicate the majority of the variance within the original dataset in the first few PCs. This is because the PCs are generated from the orthogonal directions that have the maximum variance in the original dataset. The reduction in dimension assists greatly in visualization and interpretation of the original dataset [34–36]. The beneficial and alluring feature of the PCA is the reduction in dimension of a set of multidimensional data. This feature has encouraged its widespread utilization across several disciplines [37–39].

**Multi-linear regression**

Multi-linear regression is a simple method of regressing a set of predictors on an outcome variable, which is based on the principle of ordinary least squares (OLS). The OLS, which is a method of line fitting, originally proposed by Legendre and Gauss [40] is used for the minimization of ensuing errors, associated with the representation of a predictor and its outcome variable with a simple linear relationship. The OLS estimator thrives on conformance to certain basic assumptions, which include normality and independence in the distribution in the errors, homoscedasticity of the errors and the preclusion of errors in the measurement of the predictors [41]. Within the confines of these assumptions, the OLS purportedly produces the best estimators. The solution of a MLR problem requires the extension of the OLS procedure for an array of data and subsequently solving the resulting system of equations.

**Random forest**

Random forest (RF) is an ensemble machine learning technique that integrates numerous decision trees (classification and regression trees) to achieve desired and accurate output. It was originally proposed by Amit and Geman [42] but was later developed significantly by Breiman [43]. It is an effective and reliable nonparametric method suitable for prediction of both classification and regression-based problems. It combines bootstrap aggregation method and random subspace method to each classification and regression tree (CART) to achieve effective prediction of output [44, 45]. The bootstrap aggregation method and the random subspace method were, respectively, introduced by Breiman [46–50] and Ho [51, 52].

Bootstrap aggregation, which is popularly called bagging, ensures that there is uniqueness of each decision tree made in random forest. In addition, each decision tree undergo parallel training on various subset of the dataset used for training by using several subsets of the variables that are available. Thus, bootstrap reduces prediction variance to the barest minimum and also prevents overfitting of a model. On the hand, the random subspace method, which is the second technique used in random forest machine learning technique, is used to significantly reduce prediction error as it has been shown that the output from combined decision trees using random subspace technique is better than that of individual decision tree as regards prediction error [42, 52].

It is worth stating at this point that decision tree methodology, on its own, is a viable machine learning technique when used as a prediction model but it has a limitation which is that it suffers frequently from low bias and high variance when used alone as prediction model. Interestingly, in machine learning methods, there is a balance between bias
and variance, and it is usually referred to as bias-variance tradeoff. The bias-variance tradeoff is done so that effective and reliable model for prediction can be obtained particularly when the model is to be used for an entirely new dataset not used for training the model.

Remarkably, since the introduction of random forest methodology by Breiman [43], several researchers have developed different variants of the originally developed random forest. The developed variants, which include multivariate random forest [53], random survival forests [54], quantile regression forests [55], enriched random forest [56] and so on, deal either with several problem contexts or invented with different resampling fitting approaches that result in the generalization of random forest technique. These variants of random forest methodology have been widely utilized in different disciplines by most researchers, and the results obtained have been found to be impressive in terms of its predictive performance. Some of the disciplines in which random forest methodology have been utilized include in health setting where Yesilkanat [57], in a study, utilized random forest methodology to predict cases of epidemic like novel COVID-19 in the near future. The result obtained from the study proved that random forest methodology can be employed as a predictive tool because of the impressive performance gotten from the result. In a related setting, Khalilia et al. [58] predicted disease risks from an imbalanced dataset using random forest methodology and reported with concrete evidence that random forest methodology had a better predictive ability than support vector machine, which is another machine learning technique it was compared with in the study.

In another development, Yao et al. [59], in the field of atmospheric science, used random surface methodology to forecast hail across Shandong Peninsula and reported that random forest model gave excellent results in terms of its ability to forecast hail in the aforementioned studied area. Similarly, Pham et al. [60], in a study, evaluated the efficacy of utilizing random forest to forecast rainfall as well snowmelt which are driven by watersheds which concluded that random forest was superior, in terms of performance, to other machine learning techniques used in the study. In construction and building technology, Han et al. [61] utilized random forest to predict high-performance concrete compressive strength, and the result obtained showed that random forest was highly effective in the prediction of the strength of the aforementioned concrete. In the same vein, Chun et al. [62] employed random forest technique to evaluate internal damage present in a reinforced concrete and also reported that random forest was highly effective for the evaluation.

There are other numerous fields such as remote sensing, genetic analysis, financial forecasting and so on where random forest technique has also been successfully utilized [63–66]. As discussed in the foregoing literature on the application of random forest, it is widely reported that random forest has shown immense performance as prediction model. It uses a defined and well-built algorithm to achieve this impressive performance.

### Random forest algorithm

Consider an $n$ dataset for training which has $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ in $p$-dimensional space, where $x_i (i = 1, \ldots, n)$ and $y_i (i = 1, \ldots, n)$ represents the input vector and the corresponding output, respectively. To utilize the random forest on the training dataset for regression, as in the case in this study, the following outlined steps are executed:

1. The training dataset $D$ is sampled with replacement to generate a bootstrap sample $E_1, \ldots, E_M (i = 1, \ldots, M)$. Each of the bootstrap samples has the same sample size, $n$, as the training dataset. However, it has to be said that sometimes researchers, primarily due to computational efficiency, often use fewer sample size than $n$ sample size for the bootstrap samples. The use of $n$ sample size for the bootstrap sample size was originally suggested by Breiman [46].

2. Decision tree $T_{E_i}(i = 1, \ldots, M)$ are grown for each of the bootstrap samples $E_m$ using the following modification:

   (a) At every node, the best split is selected from a randomly chosen subset of $m_{try}$ (in lieu of the entire) predictors from the $p$ predictors.

   (b) Decision trees in random forest, unlike in CART where trees are pruned, are grown to their maximum size such that no further splitting of the any node is possible. Or put more succinctly, the decision trees in random forest are grown such that every node contain samples which are not more than a pre-specified parameter known as maximal terminal nodesize (MTN).

   Note that $M$, which is sometimes represented as $ntree$, signifies the number of trees present in the forest. $m_{try}$ denotes the number of randomly selected predictors from $p$ predictors. Both $M$ and $m_{try}$ are user-specified tuning parameters in random forest.

3. The above steps are repeated until a sufficiently large $M$ is grown.

4. After steps 1–3 have been executed, the output are then predicted by finding the mean of the aggregate of the prediction of the individual decision trees using the expression given in Eq. (9).

\[
\hat{y}_m(x) = \frac{1}{M} \sum_{1}^{m} y_m(x)
\]  

(9)

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 $m$th individual tree for the input vector $x$.

**Error rate evaluation of random forest**

Random forest has two different ways in which the error rate can be evaluated in order to ascertain the validity or performance of a model developed with random forest [67]. One of the ways is the conventional method of splitting the entire dataset into training dataset and testing dataset. The training dataset is used to develop the random forest model, while the testing dataset is used to evaluate the reliability of the model developed with the training dataset, which is done by comparing the error rates of both the training dataset and the testing dataset. The other way of ascertaining the validity or performance of a model developed using random forest is to use the so-called out-of-bag error.

Out-of-bag (OOB) error is a form of cross-validation performed in random forest. It is executed in parallel with the steps involved during the training process. Specifically, when using the training dataset to develop a model, some of the dataset are not used up during the bootstrap sampling stage. This is because the sampling is performed with replacement which gives room for some samples to be extracted more than once thereby leaving out some samples in the process. The extracted samples are used to grow the decision tree, which are subsequently used for the development of the model. On the other hand, the samples that are not extracted or are “left out” are known as the OOB samples. These OOB samples are later used to test the performance of the model developed with random forest. The predictions made with the OOB samples are referred to as OOB predictions, while the error obtained using the OOB samples are described as OOB prediction error. The OOB prediction for an $ith$ input vector is aggregated and the average value determined. Equation (10) shows the expression used for determining the OOB prediction.

$\hat{y}_{i,OOB}^{\text{OOB}} = \frac{1}{k} \sum_{m=1}^{k} (y_{i,m}^{\text{OOB}})$

where $y_{i,m}^{\text{OOB}}$ 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.

If a loss function $L(.)$ is used, then the OOB prediction error, $L_{y}^{\text{OOB}}$, can be obtained using the expression:

$L_{y}^{\text{OOB}} = \frac{1}{n} \sum_{i=1}^{n} L(y_{i}, \hat{y}_{i}^{\text{OOB}})$

**Gradient boosting**

Gradient boosting machine (GBM) was invented by Freund and Schapire [68–71]. It is a prediction model whose principle is based on the concept of integrating numerous regression trees to achieve a desired output. The regression trees, which are often described as base learners, have sizes that are fixed in GBM. Specifically, the regression trees are themselves prediction algorithm that are widely known to suffer from the limitation of high variance and low bias errors when the trees are fully grown. Similarly, they also suffer from low variance and high bias errors when the trees are shallow. In order to circumvent this problem so that the errors will be minimized, two groups of algorithms that apply a combination of several regression trees to generate their output have been developed [72]. The two groups are random forests, which uses the principle of bagging as explained earlier, and gradient boosting which uses the principle of boosting. GBM, which relies on boosting principle, uses a combination of models that has low variance and high bias error to effectively whittle the high bias while ensuring that the variance is kept low. In other word, GBM utilizes several shallow trees which are trained in the same input–output data to improve the output of the prediction. This improvement in the output by GBM has promoted its widespread in numerous disciplines [70, 73–75].

Notably, some of the disciplines in which GBM have been utilized successfully include economics where Zhou et al. [76] used GBM for the prediction of movements of short-horizon price. The predicted results were reported to be better than other results from other boosting algorithms that were used for comparison in the study. Also in economics discipline, Sun et al. [77] utilized GBM for the prediction of unidentified entities type of bitcoin and also reported that GBM was highly effective for prediction. In a related development, Herrema et al. [78], in transportation engineering, employed GBM for the prediction of the exit of a runway located at Vienna airport and equally noted that GBM was effective for prediction. In addition, they concluded that GBM performed better than other machine learning techniques, and it was compared with in the study. In another study, Barua et al. [79] used GBM for the investigation of factors that contribute to taxiway and runway pavement deterioration from an airport. In construction and building engineering, GBM was utilized by Nguyen et al. [80] to predict the strength of concrete, and it was shown to have impressive performance, while Thai et al. [81] evaluated the effect of employing GBM for the prediction of failure modes of concrete panels that are reinforced under impact loads and established that the GBM was effective for the failure modes prediction of the reinforced concrete panel. In all the aforementioned investigations, the results indicate that predictions using GBM outperformed other machine
learning methods. Remarkably, the formulation of GBM was limited to problems related to classification but it was subsequently extended to problems related to regression by Friedman [82, 83]. The GBM extended to regression problems is aptly described as gradient boosting regression, and it employs the steepest-descent approach for analysis.

**Gradient boosting regression**

Gradient boosting regression (GBR) is algorithm that searches for an additive model which reduces or minimizes a defined or specific loss function. The loss function, which is usually defined before it is applied to train a dataset, is chosen on the discretion of the user. The dataset usually comprised of input variables in $t$—dimensions and $v$ data point. It is frequently denoted as $(x_i, y_i)$, where, $x_i = (x_{i1}, x_{i2}, \ldots, x_{it})$ represents the input variable of the dataset, while $y$ symbolizes the target or output variable of the dataset. It is germane to point out that GBR seek to establish an approximate function which can conveniently map an input variable $X$ to the output variable $y$ in a manner where the prior defined function, $L(y, F(x))$, is reduced.

$$y = F^*(x)$$

$$F^*(x) = \arg\min_{F(x)} L(y, F(x))$$ \hspace{1cm} (12)

There are several loss functions available for the GBR algorithm. Typical among the loss functions are absolute error (Eq. 13), squared-error (Eq. 14) and the Huber loss function (Eq. 15). Interestingly, the most widely used of these aforementioned loss functions is the squared-error function because it is computationally friendly and can be differentiated easily [82, 84]. Furthermore, the square-error function is simply the residual $y - F$, i.e., it denotes that the GBR performs simply residual refitting [84, 85].

$$L(y, F)_{L_1} = |y - F|$$ \hspace{1cm} (13)

$$L(y, F)_{L_2} = \frac{1}{2} (y - F)^2$$ \hspace{1cm} (14)

$$L(y, F)_{\text{Huber,}\delta} = \begin{cases} \frac{1}{2} (y - F)^2 & |y - F| \leq \delta \\ \delta (|y - F| - \delta/2) & |y - F| > \delta \end{cases}$$ \hspace{1cm} (15)

where $\delta$ denotes the threshold in which the loss function changes from squared-error function to absolute error function.

Notably, once an adequate loss function is defined in the algorithm of GBR, the following described procedure is executed shortly after:

1. Firstly, the model is initialized by using a constant value which is obtained from Eq. (16):

$$F_0(x) = \arg\min_\rho \sum_{i=1}^v L(y_i, \rho) \hspace{1cm} (16)$$

2. With successive boosts say from $m = 1, 2, \ldots, M$, steps i to iii are carried out:

   i. The negative gradient is evaluated using:

   $$g_i = -\left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right] \bigg|_{F(x)=F_{m-1}(x)} \hspace{1cm} i = 1, 2, \ldots, v \hspace{1cm} (17)$$

   ii. The negative gradient evaluated in step i together with the input variables is used to fit a regression tress $h(x,a_m)$ which is subsequently utilized to determine the regression tree parameter, $a_m$ with the following expression:

   $$a_m = \arg\min_{a, \beta} \sum_{i=1}^v [g_i - \beta h(x_i, a)]^2 \hspace{1cm} (18)$$

   iii. The negative gradient is later replaced with the $h(x_i,a_m)$ obtained in the steepest-descent approach and used to calculate the best gradient descent step size, $\rho_m$. This is executed with the expression:

   $$\rho_m = \arg\min_\rho \sum_{i=1}^v L(y_i, F_{m-1}(x_i) + \rho h(x_i,a_m)) \hspace{1cm} (19)$$

3. The model is updated using this equation:

$$F_m(x) = F_{m-1}(x) + \varnothing \rho_m h(x,a_m) \hspace{1cm} (20)$$

where $\varnothing \in (0, 1]$ symbolizes the shrinkage parameter whose function is to control the contribution of each base learner $h(x,a_m)$ to $F_m(x)$. $\varnothing$ is usually introduced to prevent overfitting the model [82].

For brevity, the GBR algorithm can rightly be summarized as follows:

| GBR Algorithm |
|---------------|
| **1** | **Initialize:** |
| | $F_0(x) = \arg\min_\rho \sum_{i=1}^v L(y_i, \rho)$ |
| **2** | For $m = 1, 2, \ldots, M$, perform: |
| i | $g_i = -\left[ \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right] \bigg|_{F(x)=F_{m-1}(x)}$, $i = 1, 2, \ldots, v$ |
| ii | $a_m = \arg\min_{a, \beta} \sum_{i=1}^v [g_i - \beta h(x_i, a)]^2$ |
| iii | $\rho_m = \arg\min_\rho \sum_{i=1}^v L(y_i, F_{m-1}(x_i) + \rho h(x_i,a_m))$ |
Artificial neural network

Artificial neural network (ANN) is a computation system built to simulate the working principle of the human nervous system and the brain [86]. It is regarded as the foundation of artificial intelligence, and it is used to provide solutions to problems that would otherwise prove a difficult nut to crack when solved manually. ANNs are endowed with excellent self-learning ability, which assist them to churn out desired output, as more data are made available. ANNs are loosely described as neural networks and they rely on a family of connected units or nodes known as artificial neurons or simply neurons for effective computation. There are different types of ANN that have been developed and are used currently to solve either classification or regression based problems. Some of the developed ANNs include the convolutional neural network, recurrent neural network, and the widely utilized feedforward neural network. The feedforward neural network either consist of just one layer or more than one layer in its architecture. The feedforward neural network with one layer is usually called single-layer perceptron (SLP), while the one which consists of more than one layer is described aptly as multi-layers perceptron (MLP) [87, 88].

SLP has only one flow direction without a single cycle. The flow direction is basically from the input neuron to the output neuron. SLP can only be utilized for computational problems that can be separated linearly. It consists of one or several input neurons. The input neurons, whose only function is simply to transmit inputs to the output layer, are fully connected to the weighted synapses present in the SLP structure. The weighted synapses are also connected to the output neuron where the entire computation is performed. The computation performed in the output neuron is done by transforming the inputs into a combination that is linear in form through the utilization of input weights. Often times, a nonlinear activation function is used during the computation to obtain a weighted output, \( y \) in a SLP. The output is usually expressed as shown in Eq. (21).

\[
y = \varphi \left( \sum_{i=1}^{n} w_i x_i + b \right) = \varphi \left( w^T x + b \right) \tag{21}
\]

where \( w \) denotes the weights vector; \( x \) represents the inputs vector, \( \varphi \) signifies the nonlinear activation function, and \( b \) denotes the bias.

On the other hand, MLP, as explained earlier, comprised of more than one layer which are made up of the input, hidden and output layers. In the MLP architecture, the hidden layer could be present. The flow direction of MLP, like the SLP, is in one direction. In addition, the input layer in MLP architecture receives the incoming signals and transmits it to the hidden layer. From the hidden layer, the signals are transferred to the output layer. Frequently, the numbers of neurons present in the hidden layer are the same in number. Specifically, MLP is often employed to solve problems that cannot be separated linearly to obtain the output, \( O_k \) (Eq. 22). To solve such problems, nonlinear activation function is utilized in the MLP. Typical examples of such activation function are the hyperbolic tangent function (Tanh), sigmoid function, rectified linear unit (ReLU), softmax function, among others.

\[
O_k^i = g(a_i^k \langle i = 1 \ldots r^k \rangle k = 1 \ldots m) \tag{22}
\]

where \( a_i^k \) represents the weighted sum of the \( i \)th neuron in the \( k \)th layer; \( r^k \) symbolizes the total number of neurons present in the \( k \)th layer; \( m \) is the total number of layers present in the MLP; \( g \) denotes the hidden layer neurons activation function; and \( O_k \) signifies the output of the \( i \)th neuron in the \( k \)th layer.

The expression for \( a_i^k \) in Eq. (22) is given as:

\[
a_i^k = b_i^k + \sum_{j=1}^{r^{k-1}} w_{ij}^k o_j^{k-1} (k = 1 \ldots m) \tag{23}
\]

Equation (23) can further be simplified. The simplification is performed by converting the bias, \( b_i^k \) into weights as \( w_{0i}^k \) and then making the output, \( o_j^{k-1} \) value unity. Thus, Eq. (23) becomes:

\[
a_i^k = \sum_{j=0}^{r^{k-1}} w_{ij}^k o_j^{k-1} \tag{24}
\]

where \( w_{ij}^k \) denotes the \( j \)th weight of the \( i \)th neuron in the \( k \)th layer; \( b_i^k \) is the bias of the \( i \)th neuron in the \( k \)th layer; \( r^{k-1} \) is the number of neuron present in the \( (k - 1) \)th layer; and \( o_j^{k-1} \) represents the output of the \( j \)th neuron in \( (k - 1) \)th layer.

Furthermore, the MLP utilizes a supervised learning technique known as backpropagation algorithm for the training of an input–output dataset, which involves the tuning of the parameter [89–93]. The tuning of the parameters, which are the weights and biases, are performed so that the error in the analysis can be minimized in the development of the model. Different types of training algorithms can be applied for the training of an input–output dataset in backpropagation neural network. Some of the commonly used training algorithms are Levenberg–Marquardt, Bayesian regularization, batch training with weight

---

**GBR Algorithm**

1. **Initialize:**
   \[ F_0(x) = \arg\min_{\rho} \sum_{i=1}^{L(y, \rho)} \]

3. \[ F_m(x) = F_{m-1}(x) + \tau h(x; a_m) \]

   **End for**
and bias learning rules, resilient propagation, etc. [94, 95]. Moreover, there are several error metrics that can be utilized in MLP for the evaluation of the reliability of a developed model in backpropagation algorithm. They include mean squared error (MSE), root-mean-squared error (RMSE), mean absolute error (MAE), etc. The error metrics depend on learning algorithm, which include Hebb with decay weight learning rule, gradient descent with momentum weight/bias, gradient descent strategy, etc., for the successful minimization of the error in a model.

**Cross-validation**

Cross-validation is described as a machine learning approach whose function is to seek the maximization of the capability of a machine learning model to effectively predict an output, with less error, when an entirely new dataset is attempted. It is used for the prevention of over-fitting of a model, which is often caused through fitting noise into the model. It is also used for the prevention of bias selection when training a set is often caused through fitting noise into the model. Cross-validation method comes in different types. The most widely used cross-validation methods are the hold-out, leave-one-out, k-fold, bootstrap and Monte Carlo cross-validation methods [99, 100].

In the present study, the k-fold cross-validation, which involves the random division of a dataset into equal parts known as k folds, was utilized because it has little or no selection bias. In the k-fold cross-validation procedure, the training dataset and the validation dataset are merged together. The merged dataset are divided randomly into k subsets. Then k-1 folds are used as training dataset, while the remaining fold is used as validation dataset. The procedure is repeated by selecting another dataset from the k folds as validation dataset and the remainder as training dataset until every fold is used once as validation dataset [98, 100–102]. The widely used k value is 10, which is termed as “tenfold cross-validation” [96]. The same value of k was used in the present study.

**Methodology**

**Data acquisition and descriptive analysis**

In the present study, the analysis was performed using long-term pavement performance (LTPP) data acquired from the InfoPave database [103]. In order to reasonably represent unbound granular materials, the required information was derived from the subbase and base layers denoted with layer numbers 2 and 3, respectively. Furthermore, the class of soils from which the data was obtained includes the American Association of State Highway and Transportation Officials (AASHTO) soil classes A1 to A3, which represent the coarse-grained soils used as unbound granular materials for road pavements. The spatial variability in the extracted soil properties was incorporated by considering different strategic highway research program numbers as reported in a previous study [25]. The spatial variability was ensured within each of the different US counties that include Wisconsin, Iowa, Delaware, Kansas, Oklahoma, Michigan, Louisiana, Utah, North Dakota and Washington, from which the data were acquired. The soil information acquired includes the clay content (C), silt content (Si), fines fraction (F), sand content (Sa) and gravel content (G). The compaction characteristics include the moisture content (Φ) and dry density (γ), while the stress state are the nominal maximum axial stress σa and confining pressure (σc). The data were manually sorted and pruned before usage to minimize the occurrence of data errors often associated with the LTPP data [25]. The basic descriptive statistics of the soil data is given in Table 1.

The descriptive analysis, which was done using PCA was executed in Minitab software. During the analysis, the correlation matrix was adopted because of the wide variation in the units of the soil properties involved. Subsequently, the eigenvector analysis was executed, and the factor loadings were obtained. The obtained factor loadings were then rotated based on the varimax rotation method, which is known to maximize the loading of the variables on the extracted principal components [104] and also helps to achieve reasonable communalities. This facilitates the interpretation of the resulting principal components.

| Table 1  | Descriptive statistics of the soil data |
|----------|----------------------------------------|
| Soil property | No of observation | Minimum | Maximum | Mean | Standard Deviation |
| LL (%) | 384 | 13.0 | 36 | 16.3 | 5.1 |
| PL (%) | 384 | 13.0 | 21 | 16.1 | 1.7 |
| G (%) | 384 | 0 | 45.6 | 10.5 | 12.7 |
| Sa (%) | 384 | 22.0 | 88.0 | 70.1 | 15.5 |
| Si (%) | 384 | 4.5 | 52.3 | 12.6 | 10.0 |
| C (%) | 384 | 2.5 | 23.1 | 6.8 | 4.8 |
| F (%) | 384 | 8.0 | 75 | 19.3 | 14.0 |
| γ (g/cm³) | 384 | 1644.4 | 2059.2 | 1865 | 100.0 |
| Φ (%) | 384 | 5.8 | 14.8 | 9.1 | 2.0 |
| σa (kPa) | 384 | 12.4 | 69.5 | 40.0 | 11.3 |
| σc (g/cm³) | 384 | 13.2 | 41.4 | 28.0 | 19.5 |
| Mr(MPa) | 384 | 37 | 143 | 68.6 | 20.1 |
Predictive modeling

Multiple linear regression

The statistical predictive modeling using the MLR method was done using Minitab software. The resulting variables from the PCA, which explained the majority of the variation in the data, were used for the regression analysis. The stepwise regression method was adopted to further facilitate the inclusion of variables with higher influence on the $M_R$ in the resulting model. The adequacy of the resulting regression model was evaluated based on the residual information and the collinearity diagnostics to ascertain if they are tenable for use in $M_R$ prediction.

Random forest model

The random forest model was developed in MATLAB (2020a) software. The hyperparameters used for the development of the model were $m_{try}$ and the number of regression tree, $M$. Three values were selected for $M \in \{500; 800; 1000\}$. The three values were selected simply for the purpose of comparison. In the case of $m_{try}$, four values were selected. The four values of $m_{try} \in \{2; 3; 4; 5\}$ were also selected for the purpose of comparison. It is often recommended that the first value to be selected should be the value obtained after determining the square root of the predictor.

GBR model

The development of the GBR model was executed in MATLAB (2020a) software using the input–output dataset which comprises 11 input and one output variables. Least-squared error index was initially defined as the error function to be used for the development of model during the stage of initialization in the GBR algorithm. Shortly after defining the error function, the maximum number of iteration, which is the same as the maximum number of trees, $M$, was selected. Two values of $M \in \{100; 200\}$ were selected simply for comparison sake. It has to be stated that there is no defined rule that stipulates the value of $M$ to be selected for the development of a model using GBR algorithm. In this study, however, the values of $M$ that were selected was based on the recommendations by Friedman [82]. Friedman [82], in a study on the use of GBR algorithm, submitted that $M$ values between 100 and 200 are suited for GBR model. It was reported that a larger of $M$, say when $M$ is greater than 200, could render a model complex notwithstanding that the model could be well-fitted. Similarly, a smaller value of $M$ might result in a poorly fitted model.

In addition, after the selection of $M$, the shrinkage parameter, $\varnothing$, was chosen. Four shrinkage parameter ($\varnothing \in \{0.25; 0.5; 0.75; 1.0\}$) were selected for comparison sake. Subsequently, the maximum size of the regression tree was selected. The values selected were 8. According to a study by Hastie et al. [85] and Persson et al. [75], values between 4 and 8 are well-suited for boosting techniques and which is why 8 was chosen. Thereafter, after the selection of the maximum size of the regression tree, cost complexity was used to prune the trees in order to determine the actual size of the regression tree in this study.

ANN model

The ANN model, just like the GBR and random forest model, was developed using the 11 input variables and one output variable in the study. The model was developed in MATLAB (2020a) software. The multi-layer feedforward neural network in the MATLAB software was employed. Levenberg–Marquardt training algorithm was utilized to train the dataset because it is computation friendly and consume less time during computation. In addition, when training the dataset, just one hidden layer was used and five values, chosen simple for comparison sake, were used for the hidden layer in the ANN model. The five values are 12, 13, 14, 15 and 16. Finally, the activation function employed for computation in the hidden is tan-sigmoid function, while in the case of the output layer, pure linear function was employed as the activation function.

Remarkably, to avoid the problem of overfitting during the development of the ANN model, early stopping was utilized. Early stopping is one of the techniques that is frequently used to avoid the development of overfitting problem when building ANN model. Interestingly, there are quite a few developed techniques that can also be utilized to avoid the development of overfitting problem. Some of these techniques are network reduction method, drop-out technique, expansion of training dataset technique, etc. [88, 89].

Results and discussion

Descriptive analysis

The result of the PC extraction of the original eleven variables considered in this study is displayed in the scree plot of Fig. 1. The scree plot provides a rough guide as to the number of components required to explicate the variance in the data. The scree plot is typically used alongside the Kaiser’s criterion, to provide an apt means of selecting the required number of components to explain the data variance
Based on the scree plot, the point of minimum slope in the curve corresponds to component 3. Moreover, the eigenvalues for PCs 1 to 3 are all greater than one and as such, satisfies the Kaiser’s criterion.

Furthermore, explication of the variance in the data proceeds through the loading plot of the rotated factor loadings as summarized in Table 2. The result of the rotated factor loadings for PCs 1 and 2 is represented by the plot in Fig. 2. It can be inferred from the plot that PC1 had high positive loading on the variables LL, Si, C, F and Φ. This component reveals an interaction between moisture content and the fines content of the coarse-grained soils, which had an influence on the liquid limit. As such, moisture induced changes significantly affect the MR of coarse-grained soils, based on the fines content. Therefore, PC 1 signifies the variation in MR due to the effect of moisture on the fines content.

More so, for PC 2, high loadings for G, Sa, γ and Φ resulted from the PCA. This shows that the coarse fraction present in the soils influences the moisture-density relationship of the soil, which in turn has a significant impact on the MR of the coarse-grained soils. Hence, PC 2 can be denoted as the variation in MR due to the effect of the coarse particles on the compaction behavior. The last PC, which is PC 3, had high negative loading on σc and σa simply delineates the effect of the soil stress state on the MR.

It is noteworthy that one of the variables, which is the PL, appeared not to have a significant effect on the MR variation for the coarse-grained soils based on the factor loadings and the communalities in Table 3, which shows that PL is poorly represented in the three selected PCs. Each of the PCs with individual percentage variances of 44% (PC 1), 22% (PC 2) and 9.8% (PC 3) explained a total variance of 77.8% in the data. The distribution of the data is roughly represented in the score plot of Fig. 3 for PC 1 and PC 2, which suggests that there is a pattern in the data. This preclusion of a random distribution is an indication of heterogeneity of variances in the MR data, which is a common observation for the LTPP data utilized in this study [25]. This lends credence to the adoption of a multivariate method for the descriptive analysis, which is independent of any prior data conformance in the original data.

### Table 2 Rotated factor loadings and communalities

| Soil property | PC 1     | PC 2     | PC 3     | Communality |
|---------------|----------|----------|----------|-------------|
| LL            | 0.965    | 0.137    | −0.012   | 0.950       |
| PL            | −0.307   | 0.517    | 0.038    | 0.364       |
| G             | −0.288   | −0.871   | 0.037    | 0.842       |
| Sa            | −0.650   | 0.688    | −0.029   | 0.897       |
| Si            | 0.954    | 0.015    | 0.014    | 0.911       |
| C             | 0.901    | 0.047    | −0.034   | 0.815       |
| F             | 0.993    | 0.027    | −0.002   | 0.987       |
| γ             | −0.288   | −0.885   | 0.010    | 0.866       |
| Φ             | 0.751    | 0.638    | −0.017   | 0.970       |
| σa            | −0.023   | 0.042    | −0.694   | 0.484       |
| σc            | 0.035    | −0.034   | −0.734   | 0.541       |
| Total variance (%) | 44.4 | 24.7 | 9.3 | 78.4 |

### Predictive modeling

#### MLR technique

The predictive model obtained from the MLR analysis is shown in Eq. 25, in which the MR is a function of LL, G, Sa, C, γ, σc and σa. Based on the ANOVA result in Table 3, all the factors had a statistically significant effect on the MR.
Fig. 2  Factor loading plot for PC1 and PC 2

Table 3  ANOVA and collinearity diagnostic

| Variance source | Adj SS  | Adj MS  | F-value | P-value | VIF  | $R^2$ | Adj $R^2$ | Pred $R^2$ |
|-----------------|---------|---------|---------|---------|------|-------|-----------|------------|
| Regression      | 99,274  | 14,182.1| 96.46   | 0.000   | –    | 0.642 | 0.636     | 0.626      |
| LL              | 618     | 617.8   | 4.20    | 0.041   | 27.52|       |           |            |
| G               | 9274    | 9274.3  | 63.08   | 0.000   | 15.06|       |           |            |
| Sa              | 6661    | 6660.6  | 45.30   | 0.000   | 13.19|       |           |            |
| C               | 2588    | 2588.2  | 17.60   | 0.000   | 19.23|       |           |            |
| $\gamma$       | 467     | 467.2   | 3.18    | 0.075   | 8.76 |       |           |            |
| $\sigma_c$     | 76,847  | 76,847.2| 522.69  | 0.000   | 1.01 |       |           |            |
| $\sigma_u$     | 1626    | 1625.8  | 11.06   | 0.001   | 1.00 |       |           |            |
| Lack-of-fit     | 55,211  | 165.3   | 99.89   | 0.000   | –    |       |           |            |

Fig. 3  Score plot for PC 1 and PC 2
with P-values < 0.05, excluding the γ. The coefficient of determination ($R^2$) value show that the variables accounted for about 64% variance in the $M_R$. However, the reliability of the predictive model based on the MLR is somewhat questionable. The collinearity diagnostic, which is the variance inflation factor (VIF) suggests multi-collinearity in the variables LL, G, Sa and C. This typically imparts spurious variations to the MLR, which renders the model unreliable. Moreover, analysis of the residuals is done based on the residual matrix plot in Fig. 4. The normal probability plot indicates non-normality in the data, as similarly shown by the histogram. Furthermore, the residual versus fits plot also indicate a pattern in the data which is an indication of heteroscedasticity in the data. It can also be seen from the residual values that residuals as high as 50 MPa were present in the fitted $M_R$ values. In recap, based on the residual analysis, the basic assumptions of the OLS were violated, in addition to the multi-collinearity in the predictors. These issues associated with the MLR model render it unreliable for the $M_R$ prediction and further analysis with the MLR is discontinued.

$$M_R = -31.4 + 1.295LL + 1.504G + 0.976Sa + 2.373C - 0.0327\gamma + 1.2603\sigma_c + 0.1057\sigma_a$$ (25)

**Ensemble techniques**

**Random forest**

The model performance indices including $R^2$, RMSE, MSE, MAE and mean absolute percentage error (MAPE) are summarized in Table 4 for the different tuning parameters considered (ntree and mtry). It can be clearly seen that the ntree considered were 500, 800 and 1000. The variation in the ntree appeared not to have a prominent effect on the performance indices for both the training and testing set. However, the different mtry, which include 2, 3, 4 and 5 affected the model performance indices. The best mtry value was 4 for ntree of 500, 800 and 1000 with $R^2$ value of 0.967 in all cases. Similarly, the $R^2$ values for the testing set in the respective ntree of 500, 800 and 1000 were 0.938, 0.936 and 0.936. The overall best random forest model can be considered to be the one with ntree of 500 and mtry of 4. More so, the random forest model can be considered to be reliable since the difference in the performance indices between the testing and training set is trivial. The $M_R$ prediction from the random forest model can be considered to be reliable with $R^2$ value greater than 0.9 [106, 107].
Table 4  Model performance indices for the random forest model

| Mtry | MSE   | RMSE  | MAE   | MAPE  | $R^2$  | Mtry | MSE   | RMSE  | MAE   | MAPE  | $R^2$  |
|------|-------|-------|-------|-------|--------|------|-------|-------|-------|-------|--------|
|      |       |       |       |       |        |      |       |       |       |       |        |
|      |       |       |       |       |        |      |       |       |       |       |        |
|      |       |       |       |       |        |      |       |       |       |       |        |
| 2    | 67.1  | 8.2   | 6.4   | 10.3  | 0.833  | 2    | 104.2 | 10.2  | 8.1   | 13.0  | 0.741  |
| 3    | 15.3  | 3.9   | 2.8   | 4.3   | 0.962  | 3    | 27.5  | 5.2   | 3.7   | 5.6   | 0.932  |
| 4    | 13.1  | 3.6   | 2.5   | 3.8   | 0.967  | 4    | 24.9  | 5.0   | 3.4   | 5.2   | 0.938  |
| 5    | 61.3  | 7.8   | 5.9   | 8.9   | 0.848  | 5    | 104.6 | 10.2  | 7.9   | 12.0  | 0.740  |

| Mtry | MSE   | RMSE  | MAE   | MAPE  | $R^2$  | Mtry | MSE   | RMSE  | MAE   | MAPE  | $R^2$  |
|------|-------|-------|-------|-------|--------|------|-------|-------|-------|-------|--------|
|      |       |       |       |       |        |      |       |       |       |       |        |
|      |       |       |       |       |        |      |       |       |       |       |        |
|      |       |       |       |       |        |      |       |       |       |       |        |
| 2    | 72.8  | 8.5   | 6.8   | 11.0  | 0.819  | 2    | 110.8 | 10.5  | 8.6   | 13.7  | 0.725  |
| 3    | 14.2  | 3.8   | 2.7   | 4.1   | 0.965  | 3    | 25.4  | 5.0   | 3.5   | 5.4   | 0.937  |
| 4    | 13.1  | 3.6   | 2.5   | 3.8   | 0.967  | 4    | 25.6  | 5.1   | 3.5   | 5.3   | 0.936  |
| 5    | 56.2  | 7.5   | 5.6   | 8.5   | 0.860  | 5    | 95.1  | 9.8   | 7.6   | 11.5  | 0.764  |

| Mtry | MSE   | RMSE  | MAE   | MAPE  | $R^2$  | Mtry | MSE   | RMSE  | MAE   | MAPE  | $R^2$  |
|------|-------|-------|-------|-------|--------|------|-------|-------|-------|-------|--------|
|      |       |       |       |       |        |      |       |       |       |       |        |
|      |       |       |       |       |        |      |       |       |       |       |        |
|      |       |       |       |       |        |      |       |       |       |       |        |
| 2    | 70.7  | 8.4   | 6.6   | 10.7  | 0.824  | 2    | 107.9 | 10.4  | 8.4   | 13.4  | 0.732  |
| 3    | 15.6  | 3.9   | 2.8   | 4.3   | 0.961  | 3    | 27.7  | 5.3   | 3.7   | 5.6   | 0.931  |
| 4    | 13.3  | 3.6   | 2.5   | 3.8   | 0.967  | 4    | 25.6  | 5.1   | 3.4   | 5.2   | 0.936  |
| 5    | 72.0  | 8.5   | 6.8   | 11.0  | 0.821  | 5    | 109.7 | 10.5  | 8.5   | 13.7  | 0.727  |

Bold values represent best model in each sub group, while bold and italicized represent the overall best model
**Table 5** Model performance indices for the GBR model

| TRAINING ERROR | TRAINING ERROR |
|----------------|----------------|
| ITERATION NUMBER = 100 | ITERATION NUMBER = 200 |
| LR   | MSE | RMSE | MAE | MAPE | $R^2$ | LR   | MSE | RMSE | MAE | MAPE | $R^2$ |
| 0.25 | 12.6 | 3.5  | 2.2 | 3.2  | 0.965 | 0.25 | 9.5  | 3.1  | 2.0  | 3.0  | 0.974 |
| 0.5  | 15.1 | 3.9  | 2.4 | 3.6  | 0.958 | 0.5  | 14.7 | 3.8  | 2.5  | 3.7  | 0.959 |
| 0.75 | 13.1 | 3.6  | 2.5 | 3.8  | 0.963 | 0.75 | 11.1 | 3.3  | 2.3  | 3.6  | 0.969 |

| TRAINING ERROR | TRAINING ERROR |
|----------------|----------------|
| ITERATION NUMBER = 100 | ITERATION NUMBER = 200 |
| LR   | MSE | RMSE | MAE | MAPE | $R^2$ | LR   | MSE | RMSE | MAE | MAPE | $R^2$ |
| 0.25 | 23.5 | 4.8  | 3.5 | 5.5  | 0.954 | 0.25 | 17.8 | 4.2  | 3.0  | 4.9  | 0.965 |
| 0.5  | 22.2 | 4.7  | 3.2 | 5.2  | 0.956 | 0.5  | 22.1 | 4.7  | 3.2  | 4.8  | 0.957 |
| 0.75 | 27.7 | 5.3  | 3.7 | 6.1  | 0.946 | 0.75 | 16.7 | 4.1  | 3.1  | 5.3  | 0.967 |
| 1    | 33.0 | 5.7  | 4.3 | 6.9  | 0.935 | 1    | 31.5 | 5.6  | 4.0  | 6.4  | 0.938 |

*Bold values represent best model*
The model performance indices for the GBR is summarized in Table 5 for the various learning rate (LR) and iteration number considered. The result suggests that increase in the iteration number has a slight effect on the GBR model prediction performance. In a similar way, selection of various LR (0.25, 0.5, 0.75 and 1.0) had a small impact on the GBR model performance. In the GBR model with an iteration number of 100, the best model in the training set had a learning rate of 0.25, while the best model in the testing set was achieved with a LR of 0.5. This presents a little difficulty in the selection of the optimal parameters for the $M_R$ prediction using an iteration number of 100. However, for the prediction done with an iteration number of 200, the LR of 0.25 was optimal in both the training and testing set, which makes the model more stable. Moreover, the GBR model with iteration number of 200 and LR

Table 6  Model performance indices for the ANN model

| TRAINING ERROR | HL | MSE | RMSE | MAE | MAPE | $R^2$ |
|----------------|----|-----|------|-----|------|-------|
| 12             | 3.1| 1.8 | 1.3  | 2.1 | 0.991|
| 13             | 6.3| 2.5 | 1.6  | 2.5 | 0.982|
| 14             | 2.5| 1.6 | 0.9  | 1.5 | 0.993|
| 15             | 20.8| 4.6 | 3.4  | 5.2 | 0.942|
| 16             | 4.5| 2.1 | 1.5  | 2.4 | 0.987|

| TESTING ERROR  | HL | MSE | RMSE | MAE | MAPE | $R^2$ |
|----------------|----|-----|------|-----|------|-------|
| 12             | 16.2| 4.0 | 3.1  | 5.4 | 0.968|
| 13             | 16.6| 4.1 | 2.9  | 4.6 | 0.967|
| 14             | 6.8 | 2.6 | 2.1  | 3.3 | 0.987|
| 15             | 35.1| 5.9 | 4.7  | 7.5 | 0.931|
| 16             | 21.9| 4.7 | 3.3  | 5.1 | 0.957|

Bold values represent the best model

Fig. 5  Comparative analysis of the models. (a) Training set for MSE, RMSE, MAE and MAPE. (b) Testing set for MSE, RMSE, MAE and MAPE. (c) $R^2$ values
of 0.25 produced the best model based on all the model performance indices in Table 5 for both the training and testing set. The $R^2$ values of 0.974 and 0.965 obtained for the respective training and testing set are considered to be highly reliable for $M_R$ prediction since the values were above 0.90 recommended for the predictive modeling of $M_R$ [106, 107].

ANN technique

Prediction performance indices of the $M_R$ based on ANN modeling, which is one of the most common machine learning methods for $M_R$ prediction is shown in Table 6. Different numbers of hidden layers (HL) were considered for the modeling, which include 12, 13, 14, 15 and 16. It can be inferred from the results that the selection of the HL has a little influence on the ANN model performance, albeit not critical. Furthermore, a HL of 14 is optimal for the $M_R$ prediction, recording very high $R^2$ values of 0.993 and 0.987 in the respective training and testing set. Apparently, the ANN is a very potent technique for $M_R$ prediction, and there are no issues of overfitting due to stability in performance for both the training and the testing dataset. As such, the cross-validation technique, which is the stopping criterion adopted to prevent overfitting in the machine learning methods applied in the present study is very efficient for checkmating model overfitting. Also, based on the $R^2$ values which were well-above 0.9, the ANN model prediction is laudable.

Comparative analysis of the $M_R$ prediction models

Comparison of the three machine learning techniques, which were used for the $M_R$ prediction is displayed in Fig. 5. The best model can be clearly seen in Fig. 5a and b to be the ANN model, followed by the GBR and then the RF for the respective training and testing data. Furthermore, the ANN and GBR can be considered to be more stable since the $R^2$ in the training and testing set had very minimal difference in comparison with that of the RF (Fig. 5c). The outcome of this study is consistent with that of Khasawneh et al. [29] that reported excellent $M_R$ prediction from LTPP data for fine-grained soils using the ANN technique. The three models, however, are considered apt for $M_R$ prediction since the $R^2$ values were generally above the threshold of 0.9 recommended for level two design of the mechanistic empirical pavement design guide (MEPDG) [106, 107].

Conclusion

The main objective of the present study was to evaluate the predictive capability of the prediction models for estimation of $M_R$ of unbound granular materials, while considering the spatial variability of the soil properties. The upshots of the study revealed a very poor predictive capability of MLR method in comparison with the machine learning methods. Among the machine learning methods, the ANN was found to be superior followed by the gradient boosting regression, and then the random forest technique, which all resulted in reliable $R^2$ values. The following highlights some notable findings of this study:

1. Descriptive analysis based on PCA revealed that three components, which explained a total variance of 77.8%, were required to explain the variation in $M_R$ due to spatial variability of routine soil properties. These include $M_R$ variation due to the effect of moisture-induced changes on the fines content, $M_R$ variation due to the effect of coarse particles on the compaction characteristics and the effect of the soil stress state on the $M_R$.

2. Predictive modeling of the $M_R$ using MLR produced unreliable results based on violation of the OLS assumptions in the model and the problem of multi-collinearity, in addition to low $R^2$ values of about 0.64. Hence, the MLR method is not recommended for the $M_R$ prediction.

3. Variation in the tuning parameter, ntree for the RF model, was not influential on the predictive performance of the RF model; however, the mtry had a significant influence. The best RF model was achieved with ntree of 500 and mtry of 4 and produced $R^2$ of 0.967 and 0.938 in the respective training and testing set.

4. The effects of iteration number and LR in the $M_R$ prediction performance of the GBR model were minimal. Optimal prediction for the GBR model, which achieved $R^2$ values of 0.974 and 0.964 in the respective training and testing dataset, was achieved at an iteration number of 200 and LR of 0.25.

5. Changes in the HL had a trifling effect on the prediction performance of the ANN model. The ANN model was found to produce very high $R^2$ values of 0.993 and 0.987 in the respective training and testing set with 14 HL.

6. All the three machine learning methods were very effective for the $M_R$ prediction based on $R^2$ values above 0.9. However, the ANN was the best model, followed by the GBR and then the RF. Moreover, the ANN and GBR can be considered to be more reliable than the RF based on comparison of the error indices in the training and testing set.
Acknowledgements  The authors wish to appreciate their undergraduate students, Oti Moses Chikadibia and Kalu Stephen Eke for their immeasurable and kind assistance in sorting the dataset used for this study. More importantly, the authors also commend the long-term pavement performance program that generously provided the data for the analysis.

Declarations

Conflict of interest  The corresponding author states that there is no conflict of interest.

References

1. Lekarp F, Isacsson U, Dawson A (2000) State of the art. I: Resilient response of unbound aggregates. J Transp Eng 126:66–75
2. Ng K, Mebrahtom D, Ksaibati K, Wulff SS (2019) Characterisation of crushed base for mechanistic-empirical pavement design guide. Road Mater Pavement Des 22:230
3. Pan T, Tutumluer E, Carpenter SH (2005) Effect of coarse aggregate morphology on the resilient modulus of hot-mix asphalt. Transp Res Rec. https://doi.org/10.1177/0361198105192900101
4. Bilodeau JP, Plamondon CO, Dore G (2016) Estimation of resilient modulus of unbound granular materials used as pavement base: combined effect of grain-size distribution and aggregate source frictional properties. Mater Struct 49:10
5. Bilodeau JP, Dore G, Pierre P (2008) Gradation influence on frost successceptibility of base granular materials. Int J Pavement Eng 9(6):397–411
6. Esfahani MA, Goli A (2018) Effects of aggregate gradation on resilient modulus and CBR in unbound granular materials. Int J Transp Eng 5(4):367–381
7. Cetin A, Kaya Z, Cetin B, Aydilek AH (2014) Influence of laboratory compaction method on mechanical and hydraulic characteristics of unbound granular base materials. Road Mater Pavement Des 15(1):220–235
8. Zhang J, Peng J, Liu W, Lu W (2019) Predicting resilient modulus of fine-grained subgrade soils considering relative compaction and matric suction. Road Mater Pavement Des 703:202
9. Sormunen LA, Kolisoja P (2018) Mechanical properties of recovered municipal solid waste incineration bottom ash: the influence of ageing and changes in moisture content. Road Mater Pavement Des 19(2):252–270
10. Bassani M, Riviera PP, Tefa L, Chiappinelli G (2019) Effects of quantity and plasticity of fine particles on the workability and resilient behaviour of aggregate-soil mixtures for granular pavement layers. Road Mater Pavement Des 22:444
11. Elbagalati O, Elseifi MA, Gaspard K, Zhang Z (2017) Development of an artificial neural network model to predict subgrade resilient modulus from continuous deflection testing. Can J Civ Eng 1:1–33
12. Puppala AJ (2008) Estimating stiffness of subgrade and unbound materials for pavement design. Transportation Research Board, Washington, DC
13. Solanki P, Zaman M, Ebrahimi A (2009) "Regression and artificial neural network modeling of resilient modulus of subgrade soils for pavement design. In: Intelligent and soft computing in infrastructure systems engineering. Springer, Berlin, pp 269–304
14. Tutumluer E (2013) Practices for unbound aggregate pavement layers. Transportation Research Board, Wasington, DC
15. Uthus L, Hopkins MA, Horvii I (2008) Discrete element modeling of the resilient behaviour of unbound granular aggregates. Int J Pavement Eng 9(6):387–395
16. Yan KZ, Xu HB, Shen GH (2014) Novel approach to resilient modulus using routine subgrade soil properties. Int J Geomech 14:1–9
17. Zeghal M (2004) Discrete-Element method investigation of the resilient behaviour of granular material. J Transp Eng 130:503–509
18. Mousavi SH, Gabr MA, Borden RH (2016) Subgrade resilient modulus prediction from light weight deflectometer. Can Geotech J 54:3
19. Mousavi SH, Gabr MA, Borden RH (2018) Resilient modulus prediction of soft low-plasticity Piedmont residual soil using dynamic cone penetrometer. J Rock Mech Geotech Eng 10(2):323–332
20. Liu S, Zou H, Cai G, Bheemasetty TV, Puppala AJ, Lin J (2016) Multivariate correlation among resilient modulus and cone penetration test parameters of cohesive subgrade soils. Eng Geol 209:128–142
21. Ng TT (2004) Triaxial test simulation with discrete element method and hydrostatic boundaries. J Eng Mech 130(10):1188–1194
22. Hill K, Yohannes B, Khazanovich L (2012) Unified mechanistic approach for modeling tests of unbound pavement materials. J Transp Eng 138:1091–1098
23. Zhang X (2013) Discrete element modeling of the resilient behaviour of subgrade sands. Proc Soc Behav Sci 96:490–495
24. 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
25. Ikeagwuani CC, Nwonu DC (2020) Model performance assessment in resilient modulus modelling: a multimodel approach. Road Mater Pavement Des 22:2310
26. Heidarabadizadeh N, Ghani-zadeh AR, Behnood A (2021) Prediction of the resilient modulus of non-cohesive subgrade soils and unbound subbase materials using a hybrid support vector machine method and colliding bodies optimization algorithm. Constr Build Mater 275:122140
27. Ghorbani B, Arulrajah A, Narasilio G, Horpibulsuk S, Bo MW (2020) Development of genetic-based models for predicting the resilient modulus of cohesive pavement subgrade soils. Soils Found 60:398–412
28. Kaloop MR, Kumar D, Samui P, Gabr AR, Hu JW, Jin X, Roy B (2019) Particle swarm optimization algorithm-extreme learning machine (PSO-ELM) model for predicting resilient modulus of stabilized aggregate bases. Appl Sci 9:3221
29. Khasawneh MA, Al-jamel NF (2019) Modeling resilient modulus of fine-grained materials using different statistical techniques. Transp Geotech 21:100263
30. Hanande h S, Ardah A, Abu-Farsakh M (2020) Using artificial neural network and genetics algorithm to estimate the resilient modulus for stabilized subgrade and propose new empirical formula. Transp Geotech 24:100358
31. Oskooei PR, Mohammadinia A, Arulrajah A, Horpibulsuk S (2020) cApplication of artificial neural network models for predicting the resilient modulus of recycled aggregates. Int J Pavement Eng 23:1121
32. Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2:559–572
33. Hotelling H (1933) Analysis of complex of statistical variables into principal components. J Educ Psychol 24(6):417–441
34. Jolliffe IT (1986) Principal component analysis. Springer, New York
35. Gu F, Hall P, Miles NJ (2016) Performance evaluation for composites based on recycled polypropylene using principal component analysis and cluster analysis. J Clean Prod 115:343–353
36. Korucu H, Simsek B, Guvenc AB, Kucuk VA (2019) Homogeneous graphene oxide production with the variance reduction techniques: Taguchi method with principal component analysis. Vib Spectrosc 104(102977):1–11
37. Sen D, Erazo K, Zhang W, Nagarajaiah S, Sun L (2019) On the effectiveness of principal component analysis. J Sound Vib 457:280–298
38. Wang Y, Li H (2019) Complex chemical process evaluation using a novel analytic hierarchy process model integrating deep residual network with principal component analysis. Chemom Intell Lab Syst 191:118–128
39. Banguero E, Correcher A, Navarro A, Garcia E, Aristizabal A (2020) Diagnosis of a battery energy storage system based on principal component analysis. Renew Energy 146:2438–2449
40. Hardle WK, Simar L (2015) Applied multivariate statistical analysis. Springer, Berlin
41. Rawlings JO, Pantula SG, Dickey DA (1998) Applied regression analysis: a research tool. Springer, New York
42. Amit Y, Geman D (1997) Shape quantization and recognition with randomized trees. Neural Comput 9(7):1545–1588
43. Breiman L (2001) Random forests. Mach Learn 45:5–32
44. Mangalathu S, Sun H, Nweke CC, Zhengxiang Y, Burton HV (2020) Classifying earthquake damage to buildings using machine learning. Earthq Spectra 36(1):183–208
45. Simsekler MCE, Qazi A, Alalami MA, Ellahham S, Ozonoff A (2020) Evaluation of patient safety culture using a random forest algorithm. Reliab Eng Syst Saf. https://doi.org/10.1016/j.ress.2020.107186
46. Breiman L (1996) Bagging predictors. Mach Learn 26(2):123–140
47. Breiman L (1996) Heuristics of instability and stabilization in model selection. Ann Stat 24(6):2350–2383
48. Buhlmann P, Yu B (2002) Analyzing bagging. Ann Stat 30(4):927–961
49. Chen X, Ishwaran H (2012) Random forests for genomic data analysis. Genomics 99:323–329
50. Efron B, Tibshirani RJ (1993) An introduction to the bootstrap. Chapman & Hall, London
51. Ho T (1995) Random decision forest. In: Proceedings of the third international conference on document analysis and recognition, Montreal, QC, 14–16, pp. 278–282
52. Ho T (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844
53. Segal M, Xiao Y (2011) Multivariate random forests. WIRES Data Mining Kowl Discov 1(80–87)
54. Ishwaran H, Kogalur U, Blackstone E, Lauer M (2008) Random survival forest. Ann Appl Stat 2(3):841–860
55. Meinshausen N (2006) Quantile regression forests. J Mach Learn Res 7:983–999
56. Amaratunga D, Cabrera J, Lee Y-S (2008) Enriched random forests. Bioinformatics 24(18):2010–2014
57. Yesilkagit CM (2020) Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm. Chaos Solit Fract 140(110210):1–8
58. 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
59. 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
60. Pham LT, Luo L, Finley AO (2020) Evaluation of random forest for short-term daily streamflow forecast in rainfall and snowmelt driven watersheds. Hydrol Earth Syst Sci. https://doi.org/10.5194/hess-2020-305
61. Han Q, Gui C, Xu J, Lacidogna G (2019) A generalized method to predict the compressive strength of high-performance concrete by improved random forest algorithm. Constr Build Mater 226:734–742
62. Chan 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
63. 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
64. Siroky D (2009) Navigating random forests and related advances in algorithmic modeling. Statistics surveys 3:147–163
65. Goldstein B, Polley E, Briggs F (2011) Random forests for genetic association studies. Stat Appl Genet Mol Biol 10(1):1–34
66. 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
67. Hasan MA, Nasser M, Ahmad S, Molla KI (2016) Feature selection for intrusion detection using random forest. J Inf Secur 7:129–140
68. Schapire RE (1990) The strength of weak learnability. Mach Learn 5:197–227
69. Freund Y (1995) Boosting a weak learning algorithm by majority. Inf Comput 121:256–285
70. Y Freund, Y Freund, RE Shapire (1996) Experiments with a new boosting algorithm. In: Machine learning: proceedings of the thirteenth international conference. Morgan Kaufmann Publishers, San Francisco
71. Freund Y, Schapire RE (1997) A decision theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
72. Sapountzoglou N, Lago I, Raison B (2020) Fault diagnosis in low voltage smart distribution grids using gradient boosting trees. Electric Power Syst Res 182(106254):1–12
73. 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
74. De Clercq D, Wen Z, Fei F (2019) Determinants of efficiency in anaerobic bio-waste co-digestion facilities: a data envelopment analysis and gradient boosting approach. Appl Energy 253(113570):1–11
75. Persson C, Bacher P, Shiga T, Madsen H (2017) Multi-site solar power forecasting using gradient boosted models. Sol Energy 253(113570):1–11
76. Zhou N, Cheng W, Qin Y, Yin Z (2015) Evolution of high-frequency systematic trading: a performance-driven gradient boosting model. Quant Finance 15(8):1387–1403
77. Yin HHS, Langenheldt K, Harlev M, Mukkamala RR, Vatrapu R (2019) Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the bitcoin blockchain. J Manag Inf Syst 36(1):37–73
78. Herrema F, Curran R, Hartjes S, Ellejmi M, Bancroft S, Schultz M (2019) A machine learning model to predict runway exit at Vienna airport. Transp Res Part E 131:329–342
79. Barua L, Zou B, Noruzoliaee M, Derrible S (2020) A gradient boosting approach to understanding airport runway and taxiway pavement deterioration. Int J Pavement Eng 22:1673
80. Nguyen H, Vu T, Vo TP, Thai HT (2021) Efficient machine learning models for prediction of concrete strengths. Constr Build Mater 266:120950
81. Thai DK, Tu TM, Bui TQ, Bui TT (2019) Gradient tree boosting machine on predicting the failure modes of the RC panels under loads. Eng Comput. https://doi.org/10.1007/s00366-019-00842-w
82. Friedman JH (2001) Greedy Function Approximation: a gradient boosting machine. Ann Stat 29(3):1189–1232
83. Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
84. Natekin A, Knoll A (2013) Gradient boosting machines, a tutorial. Front Neurorobot 7(21):1–21
85. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning. Springer, New York
86. McCulloch WS, Pitts W (1943) A logical calculus of ideas immanent in nervous activity. Bull Math Biophys 5:115–133
87. Gunaydin O (2009) Estimation of soil compaction parameters by using statistical analyses and artificial neural networks. Environ Geol 57:203–215
88. Kuo YL, Jaksa MB, Lyamin AV, Kaggwa WS (2009) ANN-based model for predicting the bearing capacity of strip footing on multi-layered cohesive soil. Comput Geotech 36(3):503–516
89. Hasanpour R, Rostami J, Schmitt J, Ozcelik Y, Sohrabian B (2020) Prediction of TBM jamming risk in squeezing grounds using Bayesian and artificial neural networks. J Rock Mech Geotech Eng 12(1):21–31
90. Jayawardana P, Thambiratnam DP, Perera N, Chan T, Subashi De Silva G (2019) Use of artificial neural network to evaluate the vibration mitigation performance of geofoam-filled trenches. Soils Found 59(4):874–887
91. Rashidian V, Hassanlourad M (2014) Application of an artificial neural network for modeling mechanical behaviour of carbonate soils. Int J Geomech ASCE 14(1):142–150
92. Shahin MA, Maier HR, Jaksa MB (2002) Predicting settlement of shallow foundations using neural networks. J Geotech Geoenviron Eng ASCE 128(9):785–793
93. Young-Su K, Byung-Tak K (2006) Use of artificial neural networks in the prediction of liquefaction resistance of sands. J Geotech Geoenviron Eng ASCE 132(11):1502–1504
94. Iscan B (2020) ANN modeling for justification of thermodynamic analysis of experimental applications on combustion parameters of a diesel engine using diesel and safflower biodiesel fuels. Fuel 279(118391):1–14
95. Ochmanski M, Modoni G, Bzowska J (2015) Prediction of the diameter of jet grouting columns with artificial neural networks. Soils Found 55(2):425–436
96. Kor K, Altun G (2020) Is support vector regression method suitable for predicting rate of penetration? J Petrol Sci Eng 194:1–18. https://doi.org/10.1016/j.petrol.2020.107542
97. Dietterich T (1995) Overfitting and undercomputing in machine learning. ACM Comput Surv 27(3):326–327
98. Flach P (2012) Machine learning: the art and science of algorithms that make sense of data. Cambridge University Press, Cambridge
99. Saud S, Jamil B, Upadhayay 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 Assessm 40:1–15. https://doi.org/10.1016/j.seta.2020.100768
100. 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
101. Adagbasu EG, Adelabu SA, Okello TW (2019) Application of deep learning with stratified k-fold for vegetation species discrimination in a protected mountainous region using Sentinel-2 image. Geocarto Int. https://doi.org/10.1080/10106049.2019.1704070
102. Ling H, Qian C, Kang W, Liang C, Chen H (2019) Combination of support vector machine and k-fold cross validation to predict compressive strength of concrete in marine environment. Constr Build Mater 206:355–363
103. LTPP (2018) Long-term pavement performance IMS package, Long-term Pavement Performance, 2018. https://infopave.fhwa.dot.gov/DownloadTracker/Bucket/23228
104. Gomez-Armesto A, Martinez-Cortizas A, Ferro-Vazquez C, Mendez-Lopez M, Arias-Estevez M, Novoa-Munoz JC (2020) Modelling Hg mobility in podzols: role of soil components and environmental implications. Environ Pollut 260:114040
105. Nnaji CC, Agunwamba JC (2013) The environmental impact of crude oil formation water: a multivariate approach. J Water Chem Technol 35(5):222–232
106. ArA, Inc. (2004) ERES Consultants Division, Guide for mechanistic-empirical design of new and rehabilitated pavement structures. Transportation Research Board of the National Academies, Washington, DC
107. Solanki P, Zaman MM, Dean J (2010) Resilient modulus of clay subgrades stabilized with lime, class C fly ash, and cement kiln dust for pavement design. J Transp Res Board 2186:101–110