Geomechanics and Engineering, *Vol. 37, No. 6 (2024) 629-641*
https://doi.org/10.12989/gae.2024.37.6.629

Optimizing shallow foundation design: A machine learning approach for bearing capacity estimation over cavities

Kumar Shubham*, Subhadeep Metya and Abdhesh Kumar Sinha

Department of Civil Engineering, National Institute of Technology Jamshedpur, Jharkhand-831014, India

(*Received May 30, 2023, Revised June 11, 2024, Accepted June 12, 2024*)

Abstract. The presence of excavations or cavities beneath the foundations of a building can have a significant impact on their stability and cause extensive damage. Traditional methods for calculating the bearing capacity and subsidence of foundations over cavities can be complex and time-consuming, particularly when dealing with conditions that vary. In such situations, machine learning (ML) and deep learning (DL) techniques provide effective alternatives. This study concentrates on constructing a prediction model based on the performance of ML and DL algorithms that can be applied in real-world settings. The efficacy of eight algorithms, including Regression Analysis, k-Nearest Neighbor, Decision Tree, Random Forest, Multivariate Regression Spline, Artificial Neural Network, and Deep Neural Network, was evaluated. Using a Python-assisted automation technique integrated with the PLAXIS 2D platform, a dataset containing 272 cases with eight input parameters and one target variable was generated. In general, the DL model performed better than the ML models, and all models, except the regression models, attained outstanding results with an R^2 greater than 0.90. These models can also be used as surrogate models in reliability analysis to evaluate failure risks and probabilities.

Keywords: bearing capacity; machine learning models; PLAXIS 2D; soil-structure-cavity interaction

1. Introduction

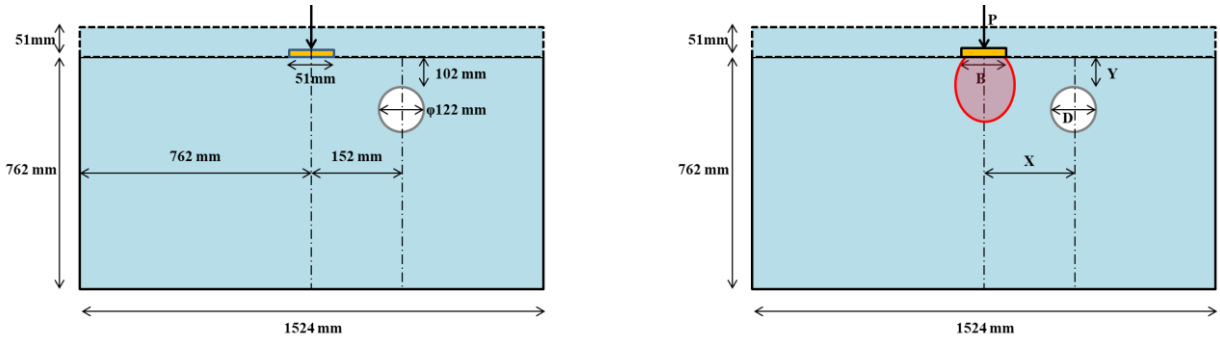
The increase in the city's population has led to the urban expansion being further extended into areas that had previously been dug. Concerns about the structural integrity of soft-ground tunnel foundations are increasing as a direct result of the expanding demand for public transportation subway systems in suburban and metropolitan locations (Broere 2016). These subways that are mined near the ground level usually change the level of the soil and may have a considerable influence on how shallow footings behave when positioned over cavities like these. Mining, canal digging, tunneling, sewer network construction, urban installation and subway excavation, and other activities that result in the underground excavation in soils can all lead to the formation of underground cavities (Goel 2015, Přibyl *et al.* 2021). Chemical reactions in regions containing water-soluble materials like dolomite and limestone can also lead to the formation of underground cavities. Underground cavities can also be formed as a result of a variety of other factors. Cavities create design and construction problems. When designing and building foundations, it is necessary to take into account any cavities, whether they are naturally occurring or were created by humans(Sabouni and Airan 2018, Keawsawasvong 2021). For the purpose of designing a safe footing system to be used over a cavity, a method of stability analysis for footings erected on a cavity is necessary.

In this direction, considerable research works have already been carried out which demonstrate the need for structure-cavity interaction analysis (Faherty *et al.* 2022, Kumar *et al.* 2022). Even when the cavities are present at a deeper level, experiments based on footing model testing and centrifuge models have demonstrated that there are considerable effects of the tunneling on the existing foundation (Lo and Ramsay 1991, Shahin *et al.* 2016, Sohaei *et al.* 2020, Fan *et al.* 2021). The presence of excavations near the buildings above the cavities also affects the stability (Li *et al.* 2019, Yu *et al.* 2021).

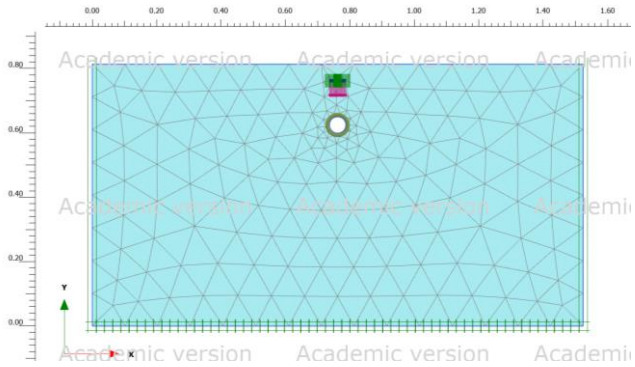
Analytical studies have also been used in the past which confirm the presence of critical zones within which the stability of foundations is highly affected by the cavities(Malhotra *et al.* 2020, Wu *et al.* 2020). Moreover, the behavior of lined cavities (such as Tunnels and Conduits) and unlined cavities (such as natural voids formed due to rock dissolution or mining activities) is completely different (Maria and Naggar 2015). Factors like cavity diameter, the width of the foundation, and vertical and horizontal positions are the main parameters that affect the bearing capacity in the presence of cavities (Habibagahi 1984, Wang and Hsieh 1987, Shubham *et al.* 2022). Stratification within the soil also contributes to the bearing capacity (Rajabi *et al.* 2022). In the case of deep foundations like piles and bridge piers, the construction of metro tunnels also influences the bearing capacity of piles (Faherty *et al.* 2022, Srivastava *et al.* 2024). However, it is very difficult to study the behavior of the foundation if the complexity increases when several parameters are considered. Special focus on cutting-edge geotechnical engineering methods used during the previous several years, with an eye towards developing methods that seem promising but still have room for development.

*Corresponding author, Research Scholar
E-mail: 2018rsce013@nitjsr.ac.in

ISSN: 2005-307X (Print), 2092-6219 (Online)



(a) Model Test (Badie and Wang 1984) (b) Schematic Diagram Used for Automation



(c) Developed Numerical Model in PLAXIS 2D

Fig. 1 Base Model considered in the present study

There is a widening scope of applications for machine learning (ML). It has created and will continue to have far-reaching effects, upending established research fields. Recent research trends show that researchers are working on finding the implementation of soft computing methods and determining the best-performing ML models in the context of geotechnical engineering (Miliziano and de Lillis 2019, Alzabeebee and Al-Taie 2022, Shubham *et al.* 2023a, Kumar *et al.* 2024). The ML models (supervised and unsupervised) like Regression Analysis, Decision Tree, Random Forest, k-Nearest Neighbors (kNN), Support Vector Machine (SVM), Multivariate Adaptive Regression Spline (MARS) along with Neural Network Techniques like Artificial Neural Network (ANN) and Deep Neural Network (DNN) and some genetic algorithms are showing some promising results in geotechnical field (Metya *et al.* 2017, Jabbar *et al.* 2018, Bong *et al.* 2020, Shubham *et al.* 2023b, Rabbani *et al.* 2024). There is an immense application of machine learning and deep learning techniques in the tunneling industry (Xue *et al.* 2020, Mirz 2021, Mirzaeiabdolyousefi *et al.* 2022). However, the recent trends used in the industry are mostly restricted only to image classification using Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN) and Generative Adversarial Networks (GAN) to predict the possible cases of failure (Zhang *et al.* 2022). Therefore, according to the authors, there exists a gap where the application of ML and DL can be made use of to study the behavior of foundations resting over cavities. Hence, this study mainly focuses on developing a suitable prediction model based on ML and DL algorithms which can be applied in real site conditions. This model will also serve the purpose of being used as a surrogate model in the reliability analysis for determining the risk and probability of failure (Fathima Sana *et al.* 2022, Chowdhury *et al.* 2023, Shubham *et al.* 2023a, Nazeeh *et al.* 2023, Kumar and Samui 2024).

2. Development of dataset and data summary

The experimental base model of Badie and Wang (Badie and Wang 1984) was redeveloped using PLAXIS 2D. The validation of the same FEM-based model has already been reported in the previous study done by the authors (Kumar *et al.* 2022). The diagrammatical representation and the developed PLAXIS model have been shown in Fig. 1. To prepare the dataset, Python assisted automation technique has been used. Automation technique is useful in those cases where the nature of action is repetitive (like the same action for input, reusing the data from a different source, multiple calculations, and optimization of parameters), the requirement of the results in a standard structure (like in reports or as spreadsheets) and the analysis that requires more of the same (like parametric study, stochastic analysis and back analysis).

The backend of the PLAXIS 2D (version. 2022) model is based on Python 3.8 scripts. Diameter to Width of foundation (D/B), Horizontal distance of cavity to the diameter of cavity ratio (X/D), vertical depth of cavity to

Table 1 Description of the Dataset used in the present study

Parameters	Notation	Count	Mean	Standard Deviation	Minimum	25%	50%	75%	Maximum
Diameter of Cavity to Width of Footing	D/B	272	2.29	0.42	0.0	2.39	2.39	2.39	3.0
Horizontal Distance to Diameter of Cavity	X/D	272	0.96	0.98	-3.0	1.25	1.25	1.25	3.0
Vertical Distance to Diameter of Cavity	Y/D	272	1.94	0.34	0.0	2.0	2.00	2.00	3.0
Dry Unit Weight of Soil	γ	272	14.18	1.34	13.5	13.78	13.78	13.78	21.0
Cohesion	c	272	161.71	35.60	12.0	158.7	158.7	158.7	360.0
Angle of Internal Friction	ϕ	272	8.79	3.76	0.0	8.0	8.0	8.0	30.0
Poisson's Ratio of Soil	μ	272	0.38	0.04	0.15	0.39	0.39	0.39	0.40
Modulus of Elasticity of Soil	E	272	23077.76	12749.59	3000.0	19872.0	19872.0	19872.0	93000.0
Bearing Capacity	BC	272	73.59	24.32	4.77	70.67	70.85	89.94	160.79

the diameter of cavity (Y/D), unit weight of soil (γ), cohesion (c), angle of internal friction (ϕ), Poisson's ratio of soil (μ) and Modulus of Elasticity of soil (E) were the different parameters considered for the present study. For the purpose of automated parametric variation, a remote scripting server is initially configured from the expert option in PLAXIS 2D. Different Integrated Development Environments (IDE) with the installation of SciTE along with some other IDE such as PyCharm, Spyder, etc., are also used. The entire dataset is described in Table 1. Then, based on the footing model test conducted by Badie and Wang (Badie and Wang 1984), the incremental loads were applied to the basic model (as shown in Fig. 1(b)) with soil layers, soil datasets and structural datasets as presented in Table 1. Mesh generation was done using the function "mesh_generation()" and integrated with the global object "g_i.gotomesh()".The different construction phases were defined to retrieve the output results in the form of M_{stage}. For each parametric variation, other parameters were kept constant. Therefore, based on the Python script programming; a total of 272 data were generated with Bearing Capacity as an output. Fig. 1(c) presents the numerical model (showing geometry configuration, mesh, and boundary conditions) developed in PLAXIS 2D for analysis.

3. Development of prediction models

The eight most commonly used ML-based regressor algorithms are used for a dataset containing 272 cases of bearing capacity for a shallow foundation resting over a cavity. A comparative study among these developed models has been carried out. A total of eight variables (D/B, X/D, Y/D, γ, c, ϕ, μ, and E) were selected as an input variable. Fig. 2 illustrates exploratory data analysis in the form of the heat map as a representation of correlation matrix, pair plots, histograms, and box plots. The correlation matrix provides the value of the correlation coefficient between the input variables. In this study, 70% of the data are used in training (191 samples) and the remaining 30% of the dataset (81 samples) are assigned for the testing dataset. Moreover, these samples are chosen randomly to maintain the generalization capability of the model.

3.1 Multivariate Linear (MLR) and Non-Linear Regression (MNLR) analysis

Regression is a statistical method for making predictions about dependent variables from known or observable ones. Foresight into the connection between variables is its primary use. By fitting the two sets of numbers into a straight line, this method establishes a connection between them. Linear Equations (refer to Eq. (1)) are used to depict this line, which is referred to as the regression line.

$$y_i = \beta_o + \beta_1 x_i + \varepsilon_i = h(x_i) + \varepsilon_i \Longrightarrow \varepsilon_i = y_i - h(x_i) \quad (1)$$

where, y_i are labels to the data, β_o is the intercept, β_1 is the slope, x_i is the training data and $h(x_i)$ is the value of the predicted response of the i^{th} observation. ε_i represents the error between the predicted value and the true value.

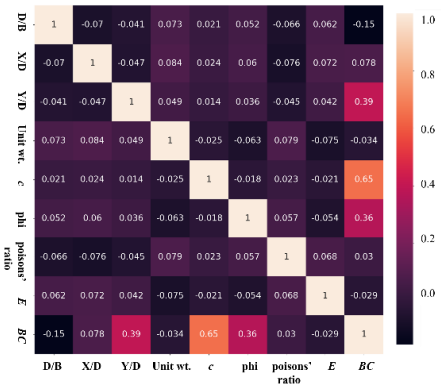
The linear regression line is plotted for the case where the value of ε_i is minimum, i.e., the best-fit line. Therefore, it becomes very important to gradually update the values of β_o and β_1 to obtain the optimum value that minimizes the error viz. defined in terms of Root Mean Square Error (RMSE) as shown in Eq. (2).

$$RMSE = \frac{\sum_{i=1}^{n}(\bar{y} - y_i)^2}{n} \quad (2)$$

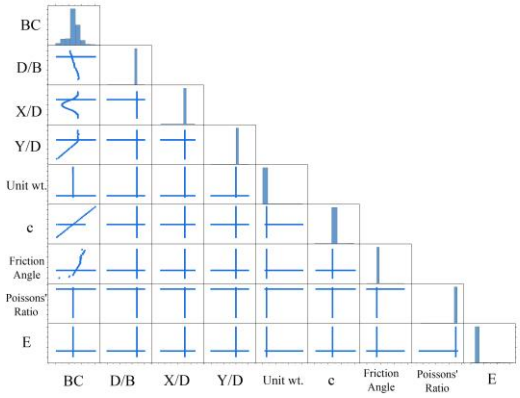
where, $\bar{y}$ is the predicted value y_i is the labelled data (actual data). In a regression model, the coefficient of determination (R^2) quantifies the percentage of variation found in the target variable that might be accounted for by the independent variables. The value of R-squared, thus, describes how well the data match the regression model. The mathematical formula to compute R^2 is given in Eq. (3).

$$R^2 = \frac{SS_{regression}}{SS_{total}} \quad (3)$$

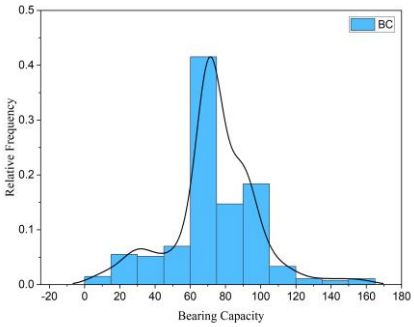
where, $SS_{regression}$ is the explained sum of squares and SS_{total} represents the total sum of squares.



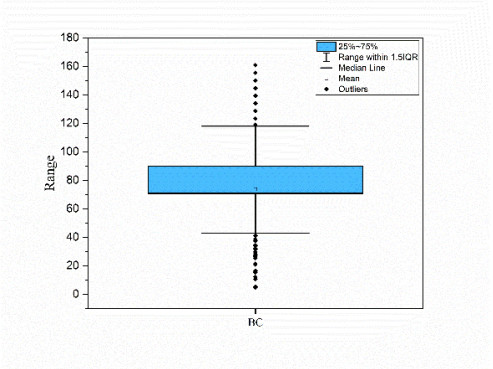
(a) Heat Map showing correlation of parameters



(b) Pair Scatter Plot of Dataset



(c) Histogram Plot showing range of output



(d) Box Plot Histogram showing output range

Fig. 2 Exploratory Data Analysis of Dataset

The relation between the input and the output variables doesn't need to be always linear. In that case, non-linear regression models are used. In this study, there are eight input parameters and one output. So, a multiple variables-based linear regression model was chosen instead of a simple linear regression model. The initial step includes importing the libraries (scikit-learn module used in this case) followed by importing the dataset (through Google Colab), which provides the features to be used for prediction. Then after, the dataset is split into training and testing datasets. The instance of the model was then developed that was fit on the training dataset to develop an MLR equation as shown in Eq. (4). The prediction was run on the test dataset, and the model was evaluated based on the R^2 value of 0.727. However, the values of Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Square Error (RMSE) are 8.0, 169.0 and 13.0 respectively

$$BC = 74.19 - 3.85\left(\frac{D}{B}\right) + 0.61\left(\frac{X}{D}\right) + 8.90\left(\frac{Y}{D}\right) - 0.28\gamma + 15.95c + 7.85\phi + 0.24\mu - 0.24 \quad (4)$$

Before generating predictions using a linear regression model, it is critical to examine the residuals to confirm that the model meets the linear regression assumptions of normal distribution, constant variance, and error term independence. When verifying for homoscedasticity, however, patterns in the error terms were noticed, suggesting a lack of linearity between some input variables (Y/D, X/D, and ϕ) and the target variable. Fig. 3(a) depicts the distribution of the error terms, which looked rather typical. Autocorrelation analysis revealed that errors were independent. Despite the lack of linear correlation in the data, a multiple linear regression (MLR) model was created, yielding an R^2 value of roughly 0.72.

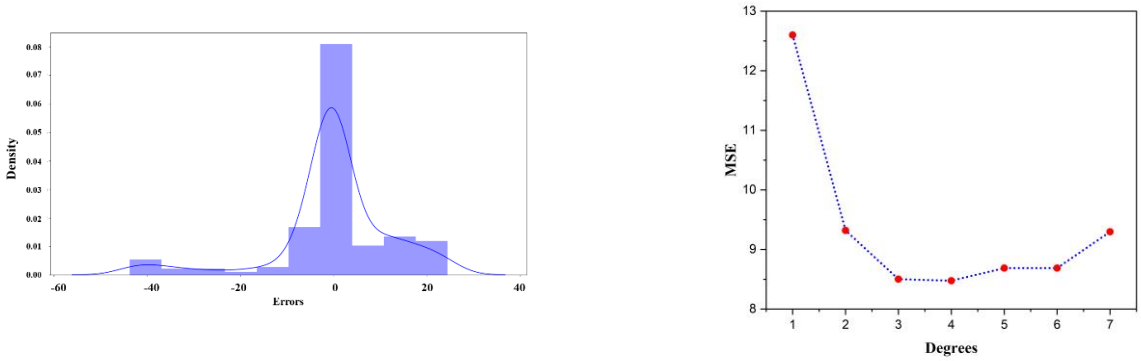
Table 2 Variation of R^2 and MSE with degree of polynomial in MNLR Analysis

Degree	R^2	MSE
1	0.7306	12.599
2	0.8526	9.320
3	0.8773	8.502
4	0.8775	8.496
5	0.8719	8.688
6	0.8719	8.687
7	0.8533	9.297

However, several MLR assumptions were not met, prompting a switch to a multivariate linear regression (MNLR) model. Nonetheless, multi-collinearity made fitting the model to the dataset difficult. A study of fits up to the seventh degree found that, as shown in Figure 3(b), a fourth-degree polynomial was best suited for this model, as summarized in Table 2.

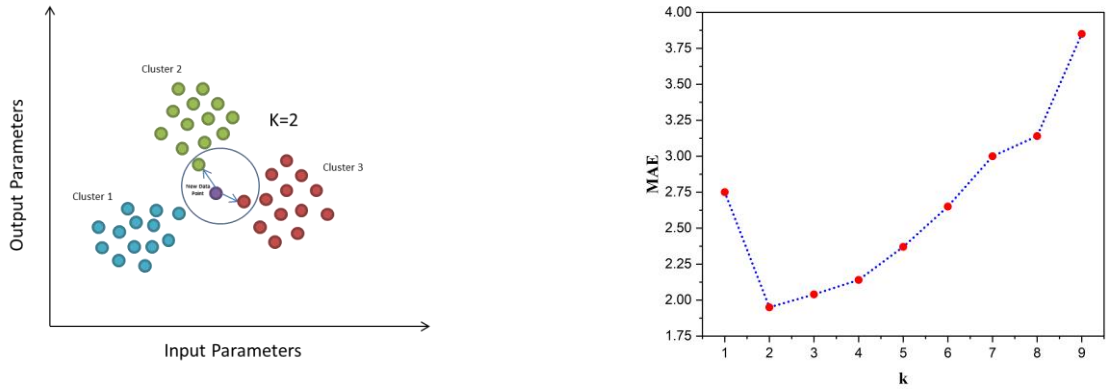
3.2 k-Nearest Neighbors (kNN)

The k-Nearest Neighbor (kNN) algorithm can be used for regression problems by predicting the value of a new



(a) Error Term distribution in MNLR Analysis (b) MSE vs Degree of model in MNLR Analysis

Fig. 3 Process of MNLR model development



(a) Working of kNN regressor (b) Curve showing the optimal k-value

Fig. 4 kNN model analysis

data point based on the values of its k-nearest neighbors (Kioumarsi *et al.* 2023). The choice of distance function is a crucial factor in KNN regression as it directly affects the prediction accuracy. The Minkowski distance function (as shown in Eq. (5)) is a widely used distance metric for continuous variables in KNN regression. It is a generalized distance metric that includes both the Manhattan distance (p=1) and the Euclidean distance (p=2) as special cases. Given a training set of data points, each consisting of a set of features (independent variables) and a corresponding target value (dependent variable), the k-NN algorithm determines the k nearest neighbors of a new data point in the feature space. The target value of the new data point is then predicted by taking the average of the target values of its k nearest neighbors. This is known as the k-NN regression estimate.

$$(\sum_{i=1}^{N}|(X_i - Y_i)|^p)^{\frac{1}{p}} \quad (5)$$

where, X_i and Y_i are two data points in N dimension space.

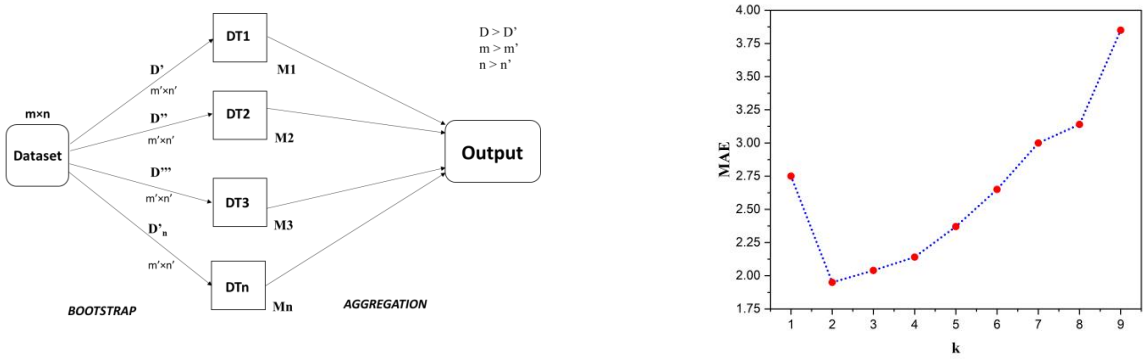
For the development of kNN model, the initial stage requires normalization of the dataset. The model is then set to be trained. Thereafter, an input data point is taken to find the distance from all the records present in the dataset.

Based on this, evaluation is done to find the optimum value of 'k' (number of nearest neighbors) as shown in Fig. 4. Through the analysis, the value of k was found to be 2. Thus, the model is set at the value of k as 2, and training is done again. The R^2 value obtained during the analysis was 0.9847. Moreover, the values of MAE, MSE, and RMSE were 2.0, 16.0 and 4.0, respectively.

3.3 Decision Tree (DT)

Decision trees effectively manage categorical data thanks to their high level of interpretability and their user-friendly algorithm. Decision trees are a representation of the way humans make decisions. Decision trees, in contrast to other techniques like logistic regression and support vector machines (SVMs), do not assist in determining whether or not there is a linear connection between the independent variable and the variable of interest (Dadhich *et al.* 2021). On the other hand, they may be used to analyze data that is very non-linear.

It poses a sequence of queries using an if-then-else structure of those nests inside itself. Users will need to pose a query at each node to further partition the data that is



(a) Schematic Diagram of RF Model (b) Optimal number of trees

Fig. 5 Random Forest Regression Model

being kept by the node. If the examination is successful, users will go to the left; if it is unsuccessful, users will proceed to the right. The node that is at the very top of a decision tree is referred to as the root node. In a decision tree, the arrows will never go toward this node; they will always point away from it. The node at the top of the tree that cannot be subdivided or categorized any further is known as the leaf node. In a decision tree, each of the arrows will always point towards this node. The term "internal node" refers to any node in the tree structure that is not a leaf node and has descendent nodes. A regression tree is a kind of additive model of the form shown in Eq. (6).

$$m(x) = \sum_{i=1}^{t} k_i \times I(x \in D_i) \quad (6)$$

where, k_i are constants, $I(.)$ is an indicator function returning 1 if its argument is true and 0 otherwise; and D_i are disjoint partitions of the training data D such that $\bigcup_{i=1}^{l} D_i = D$ and $\bigcap_{i=1}^{l} D_i = \phi$. In this current study, the DT model is applied to the regression model. The fit function accepts the arrays X and y as arguments, just as it did when it was used for classification; the only difference is that in this context, y is anticipated to contain floating point values rather than integer values. There is a straightforward solution to this kind of issue that involves constructing n separate models for each of the outputs and then using those models to make predictions for each of the n outputs. However, it is typically preferable to construct a single model that can predict all 'n' outputs concurrently, since it is probable that output values associated with the same input are also correlated. Since just a unique estimator is created, training time is reduced. Second, the resultant estimate may frequently have better generalization accuracy. The values of R2, MAE, MSE and RMSE are 0.9654, 2.0, 16.0 and 4.0, respectively.

3.4 Random Forest (RF)

Random Forest is an ensemble methodology that can handle both regression and classification problems. It does this via the use of many decision trees as well as a method known as Bootstrap and Aggregation, which is more frequently referred to as bagging (refer Fig. 5(a)). Instead, then depending on separate decision trees to conclude, the core concept that underpins this approach is to merge many decision trees to determine the final result (Kohestani *et al.* 2017). Multiple decision trees serve as the foundation for Random Forest's various learning methods. In order to generate sample datasets for each model, we first build sample datasets by randomly selecting rows and features from the dataset. This component is known as the Bootstrap. The approach for Random Forest regression method must be in the same manner as any other machine learning technique. The development of the Random Forest (RF) model starts with defining the features and the target.

The dataset is then split into training and testing datasets. Then, feature scaling is done to standardize the values. The optimal value of trees was found by plotting MAE versus number of tree curve (as shown in Fig. 5(b)). So, using the optimum value viz. 100; the model was trained again. The values of R^2, MAE, MSE and RSME are found to be 0.9809, 1.0, 10.0 and 3.16, respectively.

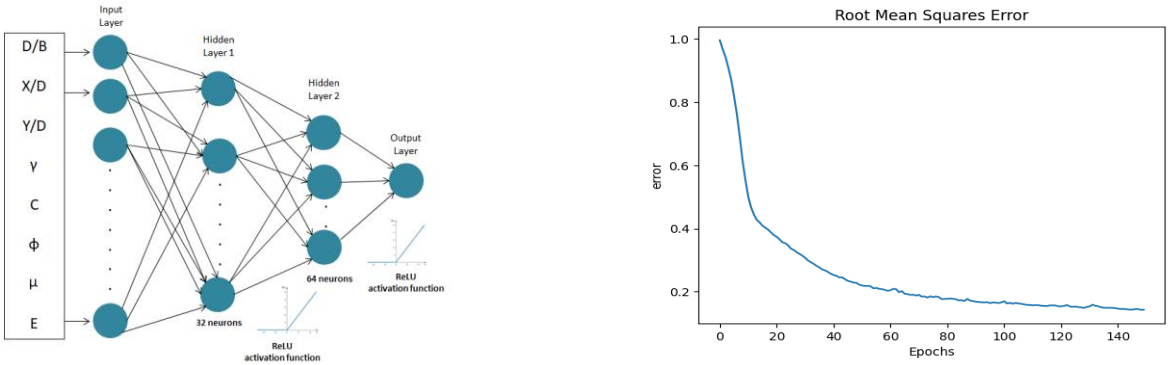
3.5 Multivariate Adaptive Regression Splines (MARS)

It is a technique for nonparametric regression, which means that it does not make any assumptions about the basic functional link that exists between the variables that are dependent and those that are independent (Metya *et al.* 2017). The MARS method uses an iterative process that moves in both forward and backward directions to pick a collection of 'basic functions' for the purpose of approximately modelling the response function. It is possible to summarize the MARS model as

$$Y = \sum_{k=1}^{n} \alpha_k H_k^f(x_i) \quad (7)$$

where $H_k^f(x_i)$ and α_k represent the basis function (represented in Eq. (7)) and the expansion function respectively. So, on solving Eq. (8), the first term obtained will be α_1 which acts as the intercept value of the function.

$$H_k^f(x_i) = \prod_{i=1}^{i_k} \left[z_{i,k} \left(x_{j(i,k)} - t_{i,k} \right) \right]_{tr}^{q} \quad (8)$$



(a) ANN Architecture (8-32-64-1) (b) Error vs Epochs Curve

Fig. 6 Development of ANN Model

where i_k represents the interaction order in the k^{th} basis function, $1 \leq j(i,k) \leq n$, $t_{i,k}$ is the knot location on each variable, q represents the order of splines, $x_{j(i,k)}$ is the j^{th} variable and $z_{i,k} = \pm 1$. In a manner that may be either forward or backward stepwise, MARS determines the position and quantity of the "basis functions" that are required for the spline. It starts by overfitting a spline function through each knot, and then it eliminates the knots that contribute the least to the overall fit of the model as assessed by the modified Generalized Cross-Validation (GCV) criteria. This is done to ensure that the model is as accurate as possible. This frequently results in the elimination of the variables that are the least significant in their entirety. For developing the MARS model, eight input variables and one output variable were stored in X and y datasets. The MARS model for this study has been developed in MATLAB. Training and testing datasets were further divided. Thereafter, base functions (BF) were introduced. BF with the lowest effect on the model was removed. It was done using generalized cross-validation (GCV) which is most often called parameter smoothening. In the first phase, a total of 14 BFs were added to the MARS model out of which 5 BFs were deleted to obtain the final prediction model (refer to Eq. (9)). The value of R^2 obtained through this analysis is 0.9546.

$$BC = 311.57 + 87.08 \times BF3 - 63.25 \times BF6 - 22.09 \times BF7 - 109.0 \times BF9 + 58.65 \times BF10 + 34.90 \times BF11 - 75.80 \times BF12 + 30.40 \times BF13 - 120.14 \times BF14 \quad (9)$$

3.6 Artificial Neural Network (ANN)

Motivated by the potential for massive parallelism enabled by distributed representations, ANN systems are inspired to capture this phenomenon. ANNs are typically constructed from a network of very basic units, each of which receives a set of real-valued inputs and generates a single real-valued output. A Multi-Layer Perceptron often known as a Multi-Layer Neural Network, is a kind of neural network that has one or more hidden layers (apart from one input and one output layer). A multi-layer perceptron, on the other hand, can learn non-linear functions in addition to linear ones, whereas a single-layer perceptron can only learn linear ones (Moayedi and Jahed Armaghani 2018).

The neurons in the input layer take the values of the input variable (and bias term) and provide an output function which is the summed input and bias. This function is called the activation function. The primary purpose of Bias is to provide each node with a value that can be trained to remain constant. Every activation function, also known as non-linearity, begins with a single integer and then applies a predetermined mathematical operation on that number. In a neural network, the activation function is applied to the weighted sum of the inputs to a neuron, which produces the output signal (as shown in Eq. (10)) of the neuron. Activation functions like sigmoid, tanh and ReLU are most often used (as shown in Eqs. (11)-(13) respectively). For developing the ANN model, input data is divided into training and testing dataset. The input dataset is then scaled to perform the standardization task. Two hidden layers were defined as having 32 and 64 neurons with ReLU (Rectified Linear Unit) and Normal kernel respectively as shown in Fig. 6(a) along with 'adam' optimizer. Adaptive Moment Estimation (adam) optimizes gradient descent. The approach works well for huge problems with many data or parameters. It's memory efficient. It effectively combines the "gradient descent with momentum" and "RMSP (Root Mean Square Propagation)" algorithms. Adam Optimizer improves on the following two approaches to optimize gradient descent. Then, RMSE variation with the number of epochs was found (as in Fig. 6(b)). The model was then evaluated. Then after, predictions and true outcomes were unscaled to obtain the R^2 value as 0.9857. Moreover, the analysis shows that the values of MAE, MSE and RMSE are 2.0, 15.0 and 3.87 respectively.

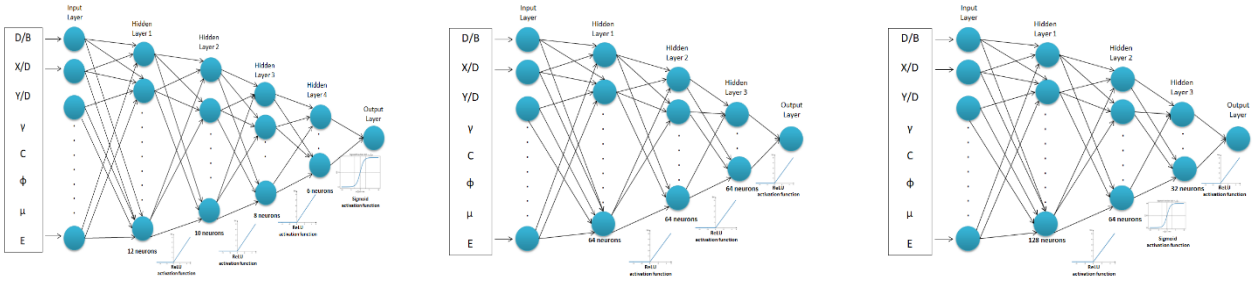
$$O = \sum_{i=1}^{i=k} w_{ij} I + b_j \quad (10)$$

where, O is the output, w_{ij} is the assigned weights, I is the input and b_j is the bias.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (11)$$

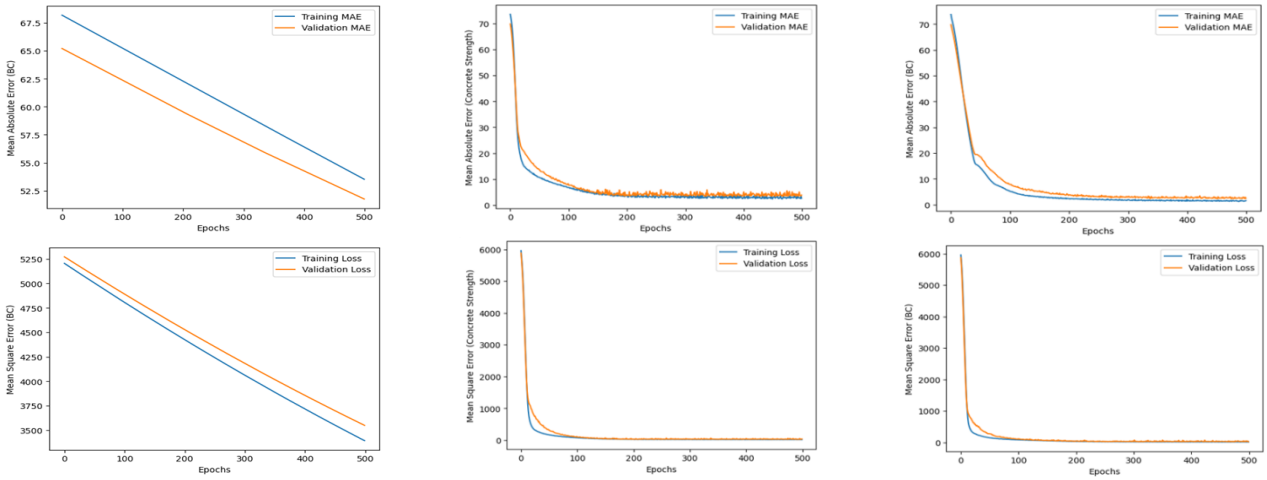
$$f(x) = \frac{1}{1 + e^{-x}} \quad (12)$$

$$f(x) = \max(0, x) \quad (13)$$



(a) Model 1 (8-12-10-8-6-1) (b) Model 2 (8-64-64-64-1) (c) Model 3 (8-128-64-32-1)

Fig. 7 Development of DNN Models.



(a) Model 1 (8-12-10-8-6-1) (b) Model 2 (8-64-64-64-1) (c) Model 3 (8-128-64-32-1)

Fig. 8. Loss (MSE and MAE) of BC vs. Epochs Curves in Different DNN Models.

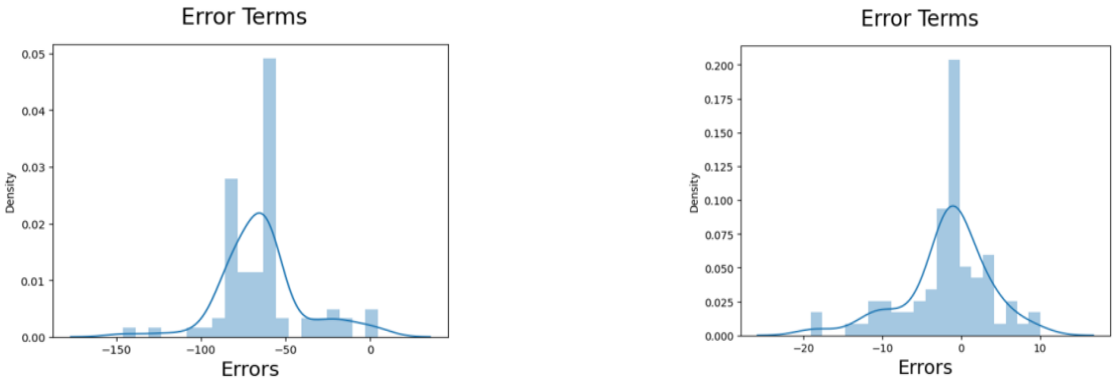
3.7 Deep Neural Network (DNN)

An artificial neural network (ANN) with several hidden layers between both the layer of inputs and outputs is called a deep neural network (DNN). DNNs, like shallow ANNs, can simulate complicated non-linear interactions. For deep learning, it's very uncommon for there to be hundreds or even a thousand hidden layers, most of which are non-linear. When compared to the neural network, the Deep Neural Network displays more originality and complexity. Algorithms based on deep neural networks are capable of speech recognition, prediction, analysis, and even creative thought. They exhibit behaviors typical of the human brain. When given enough data, Deep Neural Networks can find overall answers to issues. However, in order to mitigate the effects of the drawbacks associated with pre-existing multilayer neural networks, a DNN makes use of a variety of optimization techniques. The problem of overfitting, also known as overtraining, is one of the drawbacks of the current multilayer neural network. This means that the accuracy of predictions made using the data that was used for training is high, but the accuracy of predictions made using new data that was not used in training is very low (Bong *et al.* 2020). This is because only the training data are trained excessively.

Development of DNN model starts with loading the data and checking if there are any missing values in the dataset.

It was found that there were no missing values. The BC data is visualized using a histogram and box plot model. The input variables are then standardized using the centralized data. 50% of the data is split into training, 30% of the data into testing and the remaining 20% for validation. A linear regression model is used as a base model which gives R^2 value of 0.62. Base Model (Model 1) has 8 neurons in the input layer. There are 4 hidden layers containing 12, 10, 8 and 6 neurons respectively. ReLU and tanh were used as the activation functions (as shown in Fig. 7). The model was developed, trained with 100 epochs and then testing was done. After the training was done, loss was determined (MSE and MAE) corresponding to a number of epochs (as shown in Fig. 8). To check the efficiency, BC prediction on test data was plotted using a regression plot. Error terms were also checked which shows a large deviation from the desired values.

For each model, overfitting (if present) had to be removed by changing the units, number of layers, activation functions, adding dropout layers or adding regularizer as per the need. So, the model was updated to increase the efficiency. The new model contains 8 neurons in the input layer and 3 hidden layers with 64 neurons in each layer with ReLU as the activation function. It was observed that the accuracy increased while the error value decreased for the new model. The error distribution (refer to Fig. 9) also shows that it follows a normal distribution pattern. The



(a) Model 2 (8-12-10-8-6-1) (b) Model 3 (8-128-64-32-1)

Fig. 9 Minimizing the error with upgradation of DNN Model

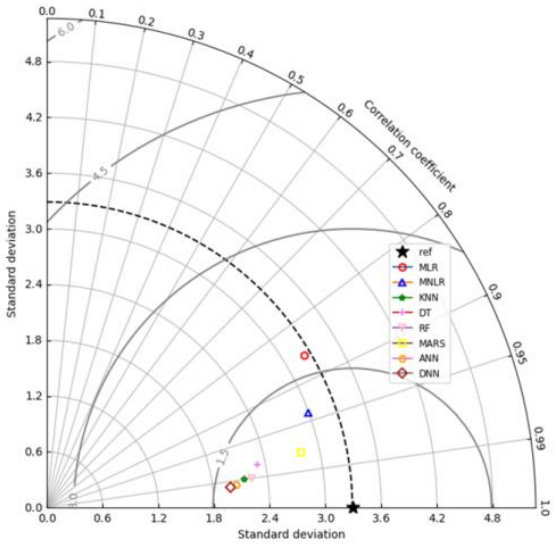


Fig. 10 Taylor Diagram showing the performance of different algorithms

analysis shows that the current DNN model has good efficiency in predicting the BC with R^2 value of 0.9871.

4. Performance comparison of algorithms

A Taylor diagram is a graphical representation used to compare and assess the similarity between different sets of data or models. It was developed by Karl E. Taylor in 2001 as a tool to evaluate how well a model simulates observed data. The diagram consists of a reference point at the origin, representing the ideal or reference dataset. Each dataset or model is represented by a point on the diagram, and its position is determined by three parameters: the correlation coefficient (r), the standard deviation ratio (ratio of the standard deviation of the dataset/model to the reference dataset/model), and the centered root-mean-square difference (CRMSD) between the dataset/model and the reference dataset/model.

The Taylor diagram allows for a quick visual comparison of multiple datasets/models. Points that are close to the reference point indicate a high level of agreement with the reference dataset/model, while points that are farther away indicate larger differences. Additionally, the distance between points on the diagram indicates the relative differences between datasets/models. The Taylor diagram provides valuable insights into the performance of models or the quality of datasets, enabling researchers to identify biases, errors, or areas for improvement.

The Taylor diagram was used to evaluate the effectiveness of eight different algorithms, including MLR, MNLR, KNN, DT, RF, MARS, and ANN, respectively. Fig. 10 shows the Taylor Diagram implemented to study the performance of the algorithms used in the present study. 3.29 was chosen as the value for the reference standard deviation. The investigation showed that there was a significant amount of variation in the algorithms' levels of performance in terms of the standard deviation and the coefficient of correlation. MLR, MNLR, KNN, DT, RF, MARS, and ANN each displayed standard deviations of 3.22, 2.99, 2.15, 2.31, 2.23, 2.8, 2.05, and 1.99, respectively. DNN demonstrated a standard deviation of 1.99. The respective coefficients of correlation for MLR, MNLR, KNN, DT, RF, MARS, and ANN, as well as DNN, were 0.86, 0.94, 0.99, 0.98, 0.990, 0.977, 0.993 and 0.994, respectively.

According to the Taylor diagram, it is possible to draw the conclusion that KNN and DNN performed exceptionally well because they exhibited the highest coefficients of correlation and

Table 3 Performance based Ranking of all the considered algorithms in the present study

Algorithm	R^2	RMSE	Ranking
MLR	0.7270	13.0	8
MNLR	0.8775	2.92	7
KNN	0.9847	4.0	3
DT	0.9654	4.0	5
RF	0.9809	3.16	4
MARS	0.9546	3.74	6
ANN	0.9857	3.87	2
DNN	0.987	3.46	1

the lowest standard deviation values. MNLR, RF, MARS, ANN, and DT all demonstrated satisfactory performance, with relatively lower standard deviations and strong correlations between variables. The MLR showed a level of performance that was in the middle of the pack, with a slightly higher standard deviation but a strong correlation coefficient. In conclusion, the Taylor diagram analysis offers extremely helpful insights into the operation of the eight different algorithms. In order to determine which algorithm is the most appropriate for the task at hand, additional thought should be given to the particular aims and prerequisites of the study.

The performances of the eight algorithms were checked based on R^2 and RMSE values, as shown in Fig. 11. In addition to that, the performance ranking of all the algorithms is mentioned in Table 3. It was found through this study that DNN and MLR have the highest and lowest R-squared values, respectively. So, DNN and ANN are found to be among the top in performance rankings. The comparative performance of different algorithms considered in this study based on R^2 value is shown in Fig. 10. During the study, it was observed that multivariate linear regression models are not good enough in predicting the bearing capacity. This model was thus improved by inculcating the non-linear behavior. It was seen that some input variables were showing a non-linear behavior, which was removed using a multivariate non-linear regression model. Moreover, this multivariate non-linear model had an R^2 value of 0.8775 which was not up to the mark for the prediction of bearing capacity as compared to the other models. So, there was a need for improvement in this model. MARS is one of the widely used models in the field of geotechnical engineering due to its optimized algorithm and can easily be used as a surrogate model in reliability analysis. The MARS model developed for this study gave an R^2 value of 0.9546 which can be considered as an excellent prediction model (Debbarma and Ransinchung 2022).

kNN model was then implemented with regressor based equations which provides an exceptional result with R^2 value of 0.9847. However, the drawback of this model is the presence of an overfitting issue, which is a tedious process to find in the model. So, tree models, i.e., Decision Tree and Random Forest models were developed. Among these, Random Forest shows a better efficiency with a low RMSE value. Many researchers are working on the integration of advanced machine learning models in the field of geotechnical engineering. Prediction models based on the neural networks are very much efficient with low errors. In this present study, an ANN model with 2 hidden layers was developed. The R^2 value obtained was the highest among all the previously developed models with the lower value of RMSE. Moving a step ahead, a Deep neural network with different combinations of hidden layers, activation functions, optimizers and number of neurons was tried and an optimized model of 3 hidden layers with 64 neurons in each layer was obtained to give the R^2 value of 0.9871 which is highest among all the types of algorithms considered for the current study.

5. Discussions

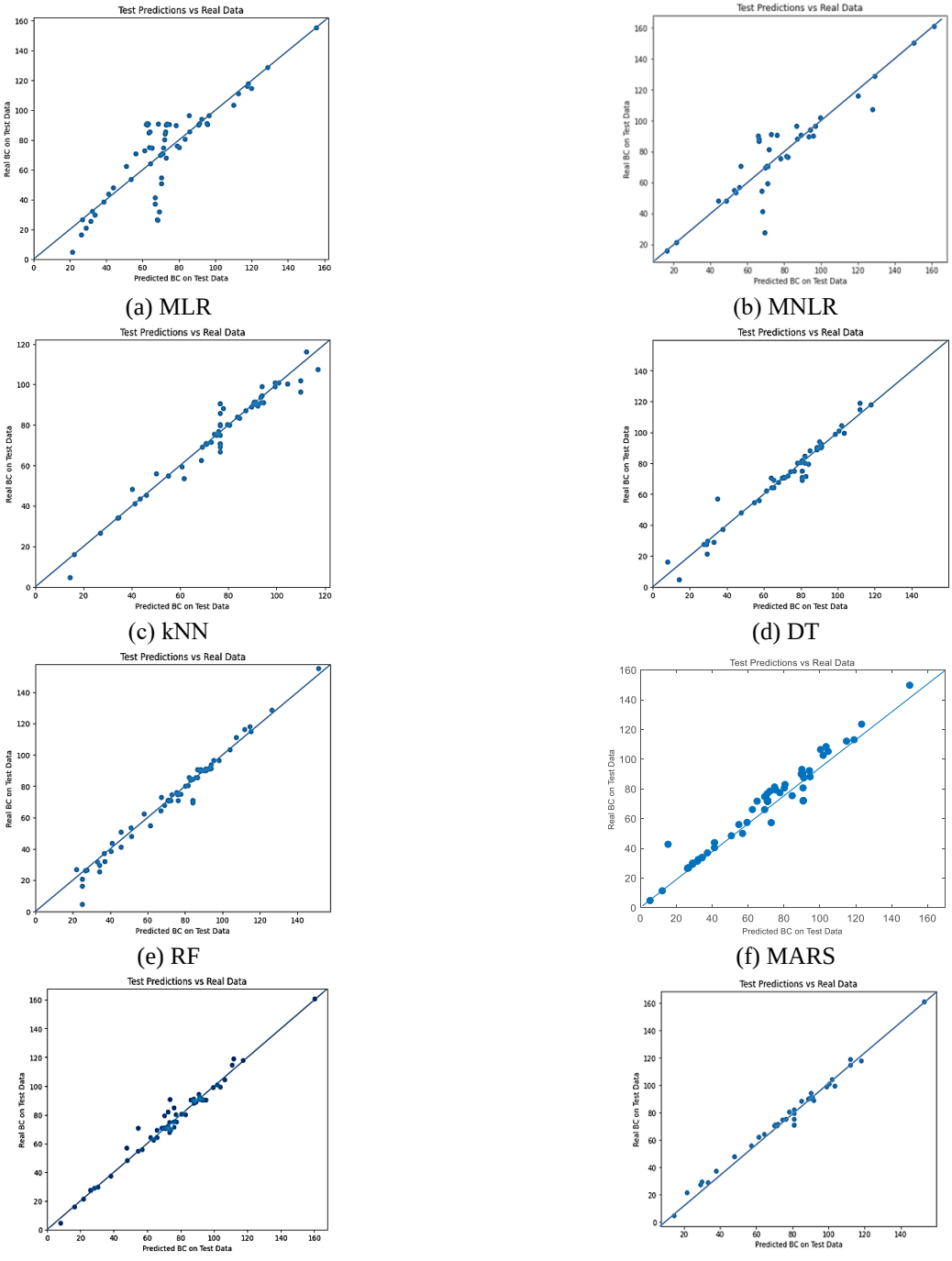
The primary aim of the current study is to propose the best suitable ML model based on a comparative study for the determination of the Bearing Capacity of a shallow foundation resting over a cavity. The current study effectively contributes to the analysis of the bearing capacity of shallow foundations in case of interacting with the underlying cavity. However, these models can also be implemented in other fields of geotechnical engineering in the following manner: *(i)* Deep Neural Networks are very impactful methods for the prediction and classification problems related to civil engineering *(ii)* the stability and effectiveness of these machine learning models can be checked using R^2 and RMSE values *(iii)* using ML approaches has resulted in the creation of some guidelines for the calculation of the bearing capacity of shallow foundations.*(iv)* the approach presented in this work has a high potential for further applicability in other geotechnical and structural engineering areas where regression difficulties are common.

However, some influencing variables in the case of foundation cavity interaction have not been considered to avoid the complexity of the study. Another restriction of the current research is the modest size of the bearing capacity dataset. As a result, if additional datasets are available, the performance of the suggested models may be enhanced. Moreover, hyper-parameter tuning techniques like Swarm Particle Optimization (PSO) and Extreme Gradient Boosting (XgBoost) can be included to increase the performance of the models (Kardani et al. 2020).

6. Conclusions

As already mentioned, this research aimed to identify the most effective model for calculating the bearing capacity of a shallow foundation over a cavity by comparing eight ML methods. These algorithms included Regression (MLR and MNLR), kNN, DT, RF, MARS, ANN, and DNN. A dataset of 272 cases was employed for the ML algorithms. R-squared and Root Mean Squared Errors accounted for during the study were used as the performance metrics of the models. The novelty of the present research includes the development and application of ML models to approximate the relationship of the bearing capacity with the variation of soil strength parameters and some other parameters associated with cavity foundation interaction. Some of the key conclusions which can be drawn are as follows:

1. The choice of ML algorithms to be used is based on the nature of input variables. Some of the input variables, like X/D, Y/D and ϕ were not linearly related to the output variable. So, the base model of ML, which works on linear



(a) MLR (b) MNLR

(c) kNN (d) DT

(e) RF (f) MARS

(g) ANN (h) DNN

Fig. 11 Regression Plots of all the algorithms used in the present study.

regression, was modified to a non-linear regression model of degree 4. Then after, the remaining algorithms (kNN, DT, RF, MARS, ANN and DNN) were used to improve the efficiency of the model.

2. The optimum score of ML algorithms demarcates that DNN is the most effective algorithm in predicting the bearing capacity. However, other algorithms (kNN, DT, RF, MARS and ANN) have R^2 values greater than 0.90 which means they can also be considered under the category of excellent prediction model. Moreover, MNLR can also be considered a good model as the R^2 value is greater than 0.85.

3. The feature importance analysis reveals that cohesion (c) is the most influential parameter, followed by the vertical distance to the diameter of the cavity (Y/D), internal friction angle (ϕ), and the diameter of the cavity to the width of the footing (D/B). Cohesion, Y/D, and ϕ positively influence the bearing capacity of the foundation, meaning that increasing these values enhances bearing capacity. Conversely, D/B has a negative effect. These findings can guide the selection of suitable variables for reliability analysis using the described machine learning models.

The future scope of the present study lies in the application of these machine learning based models in association with reliability analysis algorithms, which can then be implemented within the probabilistic finite element analysis of the stability of shallow foundations resting over cavities such as tunnels.

Statements and declarations

Funding

No funding was received to conduct this study.

Conflicts of interest/Competing interests

No potential conflict of interest in the subject matter discussed in this manuscript.

Availability of data, material and code

Some or all data, models, or codes generated or used during the study are available from the corresponding author by request.

Authors' contributions

Consent to submit has been received explicitly from all co-authors. Authors whose names appear on the submission have contributed significantly to the work submitted.

References

Alzabeebee, S. and Al-Taie, A. (2022), "Development of new models to predict the compressibility parameters of alluvial soils", *Geomech. Eng.*, **30**(5), 437-448. https://doi.org/10.12989/gae.2022.30.5.437.

Badie, A. and Wang, M.C. (1984), "Stability of spread footing above void in clay", *J. Geotech. Eng.*, **110**, 1591-1605. https://doi.org/10.1061/(ASCE)0733-9410(1984)110:11(1591).

Bong, T., Kim, S.R. and Kim, B.I. (2020), "Prediction of ultimate bearing capacity of aggregate pier reinforced clay using multiple regression analysis and deep learning", *Appl. Sci.*, **10**(13), 4580. https://doi.org/10.3390/app10134580.

Broere, W. (2016), "Urban underground space: Solving the problems of today's cities", *Tunn. Undergr. Sp. Tech.*, 55, 245-248. https://doi.org/10.1016/j.tust.2015.11.012.

Chowdhury, R., Bhattacharya, G. and Metya, S. (2023), Geotechnical Slope Analysis, 2nd Ed., CRC Press, Balkema.

Dadhich, S., Sharma, J.K. and Madhira, M. (2021), "Prediction of ultimate bearing capacity of aggregate pier reinforced clay using machine learning", *Int. J. Geosynth. Ground Eng.*, **7**, 1-16. https://doi.org/10.1007/s40891-021-00282-x.

Debbarma, S. and Ransinchung, G.D. (2022), "Using artificial neural networks to predict the 28-day compressive strength of roller-compacted concrete pavements containing RAP aggregates", *Road Mater. Pavement Des.*, **23**, 149-167. https://doi.org/10.1080/14680629.2020.1822202.

Faherty, R., Acikgoz, S., Wong, E.K.L., Hewitt, P. and Viggiani, G.M.B. (2022), "Tunnel-soil-structure interaction mechanisms in a metallic arch bridge", *Tunn. Undergr. Sp. Tech.*, **123**, 104429. https://doi.org/10.1016/j.tust.2022.104429.

Fan, S., Song, Z., Xu, T., Wang, K. and Zhang, Y. (2021), "Tunnel deformation and stress response under the bilateral foundation pit construction: A case study", *Arch. Civil Mech. Eng.*, **21**. https://doi.org/10.1007/s43452-021-00259-7.

Fathima Sana, V.K., Nazeeh, K.M., Dilip, D.M. and Sivakumar Babu, G.L. (2022), "Reliability-based design optimization of shallow foundation on cohesionless soil based on surrogate-based numerical Mmodeling", *Int. J. Geomech.*, **22**, 1-8. https://doi.org/10.1061/(asce)gm.1943-5622.0002274.

Goel, R.K. (2015), Use of underground space for the development of cities in India, Water and Energy International 58RNI:41-45.

Habibagahi, K. (1984), "Bearing capacity of strip footing above void", *J. Geotech. Eng.*, **110**, 137. https://doi.org/10.1061/(ASCE)0733-9410(1984)110:1(137).

Jabbar, S.F., Hamed, R.I. and Alwan, A.H. (2018), "The potential of nonparametric model in foundation bearing capacity prediction", *Neural Comput. Appl.*, **30**, 3235-3241. https://doi.org/10.1007/s00521-017-2916-9.

Kardani, N., Zhou, A., Nazem, M. and Shen, S.L. (2020), "Estimation of bearing capacity of piles in cohesionless soil using optimised machine learning approaches", *Geotech. Geol. Eng.*, **38**, 2271-2291. https://doi.org/10.1007/s10706-019-01085-8.

Keawsawasvong, S. (2021), "Limit analysis solutions for spherical cavities in sandy soils under overloading", *Innov. Infrastruct. Solutions*, **6**, 1-8. https://doi.org/10.1007/s41062-020-00398-5.

Kioumarsi, M., Dabiri, H., Kandiri, A. and Farhangi, V. (2023), "Compressive strength of concrete containing furnace blast slag; optimized machine learning-based models", *Clean Eng. Technol.*, **13**, 100604. https://doi.org/10.1016/j.clet.2023.100604.

Kohestani, V.R., Vosoughi, M., Hassanlourad, M. and Fallahnia, M. (2017), "Bearing capacity of shallow foundations on cohesionless soils: A random forest based approach", *Civil Eng. Infrastruct. J.*, **50**, 35-49. https://doi.org/10.7508/ceij.2017.01.003.

Kumar, D.R., Wipulanusat, W., Kumar, M., Keawsawasvong, S. and Samui, P. (2024), "Optimized neural network-based state-of-the-art soft computing models for the bearing capacity of strip footings subjected to inclined loading", *Intell. Syst. Appl.*, **21**, 200314. https://doi.org/10.1016/j.iswa.2023.200314.

Kumar, P., Metya, S., Shubham, K. and Prashad, D. (2022), "Behaviour of strip footing over cavity subjected to inclined and eccentric loads", *Arabian J. Geosci.*, 15. https://doi.org/10.1007/s12517-022-10739-6.

Kumar, P. and Samui, P. (2024), "Reliability-based load and resistance factor design of an energy pile with CPT data using machine learning techniques", *Arab. J. Sci. Eng.*, **49**, 4831-4860. https://doi.org/10.1007/s13369-023-08253-2.

Li, M.G., Xiao, X., Wang, J.H. and Chen, J.J. (2019), "Numerical study on responses of an existing metro line to staged deep excavations", *Tunn. Undergr. Sp. Tech.*, **85**, 268-281. https://doi.org/10.1016/j.tust.2018.12.005.

Lo, K.Y. and Ramsay, J.A. (1991), "The effect of construction on existing subway tunnels-a case study from Toronto", Tunnelling and Underground Space Technology incorporating Trenchless **6**, 287-297. https://doi.org/10.1016/0886-7798(91)90140-Y.

Malhotra, M., Sahu, V., Srivastava, A. and Misra, A.K. (2020), "Experimental and numerical investigation of the effect of pre-existing utility tunnel on the bearing capacity of shallow footing in sandy soils", *J. Eng. Design Tech.*, **18**, 513-529. https://doi.org/10.1108/JEDT-04-2019-0102.

Maria, A. and Naggar, E. (2015), Effects of tunnelling on the bearing capacity of shallow foundations. In: GeoQuébec 2015.

Metya, S., Mukhopadhyay, T., Adhikari, S. and Bhattacharya, G. (2017), "System reliability analysis of soil slopes with general slip surfaces using multivariate adaptive regression splines", *Comput. Geotech.*, **87**, 212–228. https://doi.org/10.1016/j.compgeo.2017.02.017.

Miliziano, S. and de Lillis, A. (2019), "Predicted and observed settlements induced by the mechanized tunnel excavation of metro line C near S. Giovanni station in Rome", *Tunn. Undergr. Sp. Tech.*,

86, 236-246. https://doi.org/10.1016/j.tust.2019.01.022.
Mirzaeiabdolyousefi, M., Mahmoodzadeh, A., Ibrahim, H.H., Rashidi, S., Kamal Majeed, M. and Mohammed, A.H. (2022), "Prediction of squeezing phenomenon in tunneling projects: Application of Gaussian process regression", *Geomech. Eng*., **30**(1), 11-26. https://doi.org/10.12989/gae.2022.30.1.011.
Moayedi, H. and Jahed Armaghani, D. (2018), "Optimizing an ANN model with ICA for estimating bearing capacity of driven pile in cohesionless soil", *Eng. Comput*., **34**, 347-356. https://doi.org/10.1007/s00366-017-0545-7.
Nazeeh, K.M., Dilip, D.M. and Sivakumar Babu, G.L. (2023), "Quantile-based design and optimization of shallow foundation on cohesionless soil using adaptive Kriging surrogates", *Int. J. Geomech*., 23. https://doi.org/10.1061/ijgnai.gmeng-8226.
Přibyl, O., Přibyl, P. and Svítek, M. (2021), Interdisciplinary urban tunnel control within smart cities, Applied Sciences (Switzerland), 11. https://doi.org/10.3390/app112210950.
Rabbani, A., Muslih, J.A., Saxena, M., Patil, S.K., Mulay, B.N., Tiwari, M., Usha, A., Kumari, S. and Samui, P. (2024), "Utilization of tree-based ensemble models for predicting the shear strength of soil", *Transport. Infrastruct. Geotech*., https://doi.org/10.1007/s40515-024-00379-6.
Rajabi, A.M., Saadati, M., Mahmoudi, M. and Fijani, E. (2022), "Effect of the circular cavity on the undrained bearing capacity of shallow strip footing", *Arabian J. Geosci*., **15**, 1-10. https://doi.org/10.1007/s12517-022-10503-w.
Sabouni, R. and Airan Eng, M. (2018), "Evaluation of foundation on soil with cavities: A case study from the UAE", *Int. J. Struct. Civil Eng. Res*., 358-363. https://doi.org/10.18178/ijscer.7.4.358-363.
Shahin, H.M., Nakai, T., Ishii, K., Iwata, T. and Kuroi, S. (2016), "Investigation of influence of tunneling on existing building and tunnel: Model tests and numerical simulations", *Acta Geotech*., **11**, 679-692. https://doi.org/10.1007/s11440-015-0428-2.
Shubham, K., Metya, S. and Bhattacharya, G. (2022), "Reliability analysis of settlement of a foundation resting over a circular void", (Eds., Satyanarayana Reddy, C.N.V., Krishna, A.M. and Satyam, N.), Dynamics of Soil and Modelling of Geotechnical Problems. Lecture Notes in Civil Engineering, 186th edn. Springer, Singapore.
Shubham, K., Metya, S. and Sinha, A.K. (2023a), "Surrogate model-based prediction of settlement in foundation over cavity for reliability analysis", *Transport. Infrastruct. Geotech*.,. https://doi.org/10.1007/s40515-023-00329-8.
Shubham, K., Rout, M.K.D. and Sinha, A.K. (2023b), "Efficient compressive strength prediction of concrete incorporating industrial wastes using deep neural network", *Asian J. Civil Eng*., https://doi.org/10.1007/s42107-023-00726-x.
Sohaei, H., Namazi, E., Hajihassani, M. and Marto, A. (2020), "A review on tunnel–pile interaction applied by physical modeling", *Geotech. Geol. Eng*., **38**, 3341-3362. https://doi.org/10.1007/s10706-020-01240-6.
Srivastava, A., Kothari, S. and Jawaid, S. (2024), "Numerical simulation-based performance assessment of pile group placed over buried utility tunnel", *Iranian J. Sci. Tech. T. Civil Eng*., https://doi.org/10.1007/s40996-023-01321-5.
Wang, M.C. and Hsieh, C.W. (1987), Collapse Load of Strip Footing Above Circular Void., **113**, 511-515.
Wu, G., Zhao, M., Zhao, H. and Xiao, Y. (2020), "Effect of eccentric load on the undrained bearing capacity of strip footings above voids", *Int. J. Geomech*., **20**, 04020078. https://doi.org/10.1061/(asce)gm.1943-5622.0001710.
Xue, Y., Li, X., Li, G., Qiu, D., Gong, H. and Kong, F. (2020), "An analytical model for assessing soft rock tunnel collapse risk and its engineering application", *Geomech. Eng*., **23**(5), 441-454. https://doi.org/10.12989/gae.2020.23.5.441.
Yu, Z.T., Wang, H.Y., Wang, W., Ling, D.S., Zhang, X.D., Wang, C. and Qu, Y.H. (2021), "Experimental and numerical investigation on the effects of foundation pit excavation on adjacent tunnels in soft soil", *Math. Probl. Eng*., https://doi.org/10.1155/2021/5587857.
Zhang, W., Gu, X., Tang, L., Yin, Y., Liu, D. and Zhang, Y. (2022), "Application of machine learning, deep learning and optimization algorithms in geoengineering and geoscience: Comprehensive review and future challenge", *Gondwana Res*., **109**, 1-17. https://doi.org/10.1016/j.gr.2022.03.015.

CC