Numerical Methods in Civil Engineering, 9-2 (2024) 40-54

Numerical Methods in Civil Engineering

Journal Homepage: https://nmce.kntu.ac.ir/

Machine learning models for predicting the bearing capacity of shallow foundations: A Comparative study and sensitivity analysis

Hamid Mohammadnezhad*, Seyedmohammad Eslami**

ARTICLE INFO

RESEARCH PAPER

Article history:
Received:
May 2024
Revised:
September 2024
Accepted:
December 2024

Keywords:
Machine learning
Shallow foundation
Prediction model
Bearing capacity
Regression models

Abstract:

Bearing capacity estimation of shallow foundations is an essential requirement in the design of structures and taking a calculation method into account is necessary. All the parameters and uncertainties cannot be factored in by the classic analytical-based methods. Moreover, performing in-site tests requires extensive time and many resources. With the development of new methods such as Machine Learning (ML) algorithms in recent decades, a resolution to these challenges has been identified. In this study, classic machine learning regression methods such as KNN, SVM, and decision-based models alongside the utilization of Artificial Neural Network (ANN) regression are examined and compelling results are demonstrated. The dataset in this study consists of 97 tests on model foundations and site loadings on granular soil. The results indicate that these ML regression methods will have reliable outcomes in the determination of bearing capacity compared to some other available approaches. But more importantly, the precision of the trained model is closely correlated to data splitting and the ratio of train and test series in the dataset. The importance of the splitting procedure was examined through trial and error with parameters of train test data ratio and the random state of sampling. It is indicated that a ratio of 80% for the training set would be an optimum value. Furthermore, the relative importance of the input features was examined through a sensitivity analysis which indicated that the internal friction angle of the soil and the depth of the foundation are the most important inputs while using ML regression methods.

1. Introduction

Construction has always been crucial in the development of civilizations. One essential component of a structure is the foundation which two factors of strength and serviceability of the structure are completely dependent on it. A shallow foundation, as the most common type of foundation, has a depth-to-width ratio less than or equal to four ($\frac{D}{B} \leq 4$) [1]. In the design of foundations, it is critical to perform an analysis on the soil's mechanical properties such as settlement and bearing capacity.

Determination of the bearing capacity of shallow foundations has always been an extensively studied matter in foundation engineering. The first attempts in this area were made by Prandtl by using slip line equations for the case of plane deformations, derived by Kotter, which resulted in an analytical form for strip footings [2]. Then Terzaghi derived the first comprehensive approach for calculating the bearing capacity of shallow foundations using the limit equilibrium method [3]. Afterward, many other researchers proposed several approaches; for instance, Meyerhof [4] Hansen [5], and Vesic [6] developed new equations by improving Terzaghi's method.

Not every problem necessarily has an analytic solution and cannot be written down in standard mathematics form [7]. In engineering and specifically geotechnical engineering we mostly face such problems which include myriad or unlimited variables and complex analysis. So, to solve these problems, an approximate answer with top precision would be sufficient and that is the reason humans have utilized numerical methods for addressing mathematical issues from ancient times [8]. Later in the 16th century, lots of

**Corresponding Author: Assistant Professor, Faculty of Civil, Water & Environmental Engineering, Shahid Beheshti University, Tehran, Iran.*
Email: h_mohammadnezhad@sbu.ac.ir
*** Bachelor's degree graduate, Faculty of Civil, Water & Environmental Engineering, Shahid Beheshti University, Tehran, Iran.*

https://doi.org/10.61186/NMCE.2405.1056

achievements in improving numerical methods were attained by scientists like Newton [9]. Furthermore, many methods which are used to analyze and design in engineering fields were invented based on numerical methods, like Finite Element Method (FEM) which has various applications in geotechnical engineering [10]. However, doing these numerical calculations would mostly take a lot of time and is complex to model. With the invention of electronic computers in the 20th century [11], numerical methods implementation entered a new era. For instance, Morgenstern and Price [12] programmed digital computers to analyze the stability of slip surfaces. Later in the second half of the 20th century, the intelligent behavior of computers was studied as Artificial intelligence (AI) [13] and then in the 60th decade, Rosenblatt used machines to solve cognitive human problems based on mankind's nervous system [14] and Machine Learning (ML) as a subset of AI was introduced. Since the 90th decade, myriad studies on the utilization of ML methods in foundation engineering have been reported. Yeh et al. [15] and Chan et al. [16] announced studies on driven piles using Artificial Neural Network (ANN). Lee, I. M. and Lee, J. H. [17] reported a study on the prediction of bearing capacity of deep foundations using ANN.

Furthermore, numerous studies in different foundation and soil conditions using ML methods have been declared in recent years. Shahin et al. [18] researched settlement prediction of shallow foundations on cohesionless soil using back-propagation artificial neural networks. Rezania and Javadi [19] suggested a genetic programming model for predicting the settlement of shallow foundations. Samui [20] utilized SVM to analyze the behavior of shallow foundations and Gajan [21] studied the application of ML methods to predict the performance of shallow foundations during earthquakes and dynamic soil situations. Altinok and Ülker [22] performed a study on the bearing capacity determination of closed-ended piles through non-linear machine learning methods. Jibanchand and Devi [23] applied ensemble learning methods for the estimation of settlement in shallow foundations.

In this study, the intended course of action includes the prediction of the bearing capacity of a further described dataset and the evaluating of the results. The dataset has been used for another study by Padmini et al. [24] to examine neuro-fuzzy models and comparison with some other approaches. As a result, a comprehensive study of the classic ML methods on the dataset would be beneficial as new approaches. Furthermore, the data splitting parameters affecting regression results consisting of train and test sets ratio and also sampling pattern are examined through trial and error to conclude with advantageous results. Moreover, the relative importance of the input features was examined through a sensitivity analysis.

2. Introduction to the applied models

Padmini et al. [24] conducted research on the performance of Adaptive Neuro-Fuzzy Inference System (ANFIS) in predicting the bearing capacity of shallow foundations alongside comparing its performance with Artificial Neural Networks (ANN) and Fuzzy Interface System (FIS) analysis results and revealed that ANFIS has significantly more precise predictions. Additionally the precision of the three mentioned methods was contrasted with analytical bearing capacity theories (like Terzaghi's correlation for bearing capacity) and the results indicated that ANN, FIS, and AFNIS exhibit remarkably improved accuracy. Later, Kalinli et al. [25] utilized ANN prediction with a different structure to the same dataset collected by Padmini et al. [24] and achieved enhanced results. They also developed an improved Meyerhof formula using an optimization technique called Ant Coloni Algorithm which led to increased precision than the analytical theories.

Because both special machine learning-based approaches and analytical theories were examined by previous researchers, a shortage of more commonly used machine learning approaches is apparent. Therefore, techniques known as Classic ML algorithms [26] were studied in the present literature. In addition, another experiment using ANN was performed with the purpose of a comparative study. All of the stated methods are classified as supervised learning models. Supervised learning does a mapping between a set of input variables and an output variable. To achieve that, the dataset is divided into two sets; training set and test set. Supervised ML methods learn using a train set and then evaluate the results by applying them to a test set [27].

The hyperparameters of the utilized algorithms and also dataset properties will be elaborated in sections 3 and 4.

Some utilized methods are introduced for a better understanding.

2.1. Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) is a supervised algorithm inspired by human brain functions. Neuron-like units do the process and provide the output [28]. ANN works as linked nodes in several layers of learning which generalize the relationship between a dataset of inputs and outputs [29]. Figure 1 shows the schematics of ANN. The first layer is the input layer. Hidden layers include nodes with activation functions, which are applied to the input of the nodes. Then through an optimization method like backpropagation, the network approximates values in each epoch to lower the error by modifying the weights of node links, and finally network learns that specific problem [28, 29]. Figure 2 illustrates the Neuron Model.

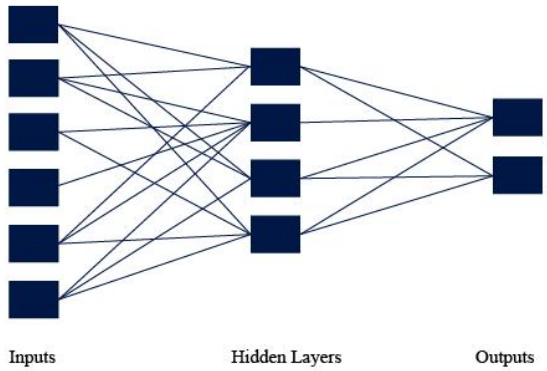


Fig. 1: Schematics of ANN

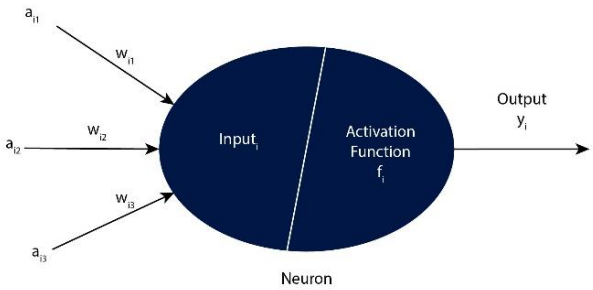


Fig. 2 – Schematic of Neuron Model

a_i is the feature vector and w_i is the weight vector. Equation 1 shows the $input_i$ function for a neuron [29].

$$Input_i = \sum_{i}^{n} w_i \, a_i \qquad (1)$$

Which w_i is the weight of the link and a_i is the function output. Various activation functions can be used for the output of the neuron such as the Sigmoid function, ReLU, and Selu.

2.2. Classic Machine Learning Algorithms

2.2.1 Support Vector Machines (SVM)

Support Vector Machines (SVM) is a supervised machine learning method for classification and regression. SVM is a system that utilizes a learning algorithm with the aim of optimization and executes a bias on data [30]. Simply, SVM finds a hyperplane (in high dimensional feature space) that classifies the training data with maximized margin. The simplest kind of SVM is linear SVM which employs a linear classifier. But the case is not always linear and that is the reason kernel functions are used. Kernel functions transform the input data into a feature space in higher dimensions to be separable with a plane. Figure 3 illustrates the idea of how SVM is implied. In Figure 3, red and blue dots are representative of two classes and the hyperplane in a new multi-dimensional feature space can easily separate them. There are different kernels like polynomial, RBF, and Sigmoid which utilize different mathematical functions.

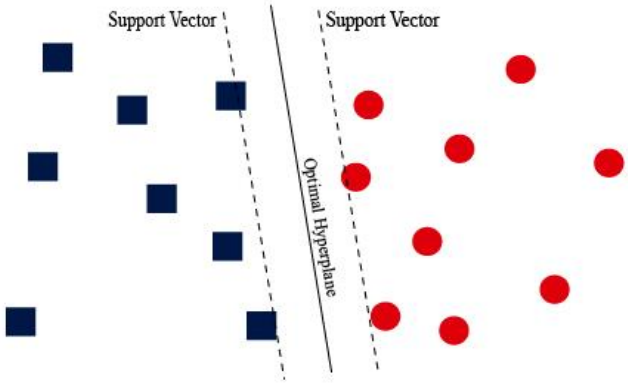


Fig. 3: SVM classifier

The main goal in utilizing SVM is to find a function that provides a deviation of error from the desired output. SVM is an optimization problem [31].

2.2.2. K-Nearest Neighbours method (KNN)

Fundamentally, K-Nearest Neighbours (KNN) assigns an unclassified vector a class based on a set of nearest classified points [32]. The value k for the number of nearest neighbors, which the method is biased by, is required and the success of the method is completely dependent on it. For the stated purpose, one approach might be trial and error of various k values. The drawback of KNN is that it requires storing the whole training set and it might get high-priced when the dataset is too large [33]. By finding the optimal k, machine is now capable of predicting unknown value new points. KNN is also proficient in doing regression based on the nearest neighbor's values. Figure 4 is a 2D simple demonstration of KNN.

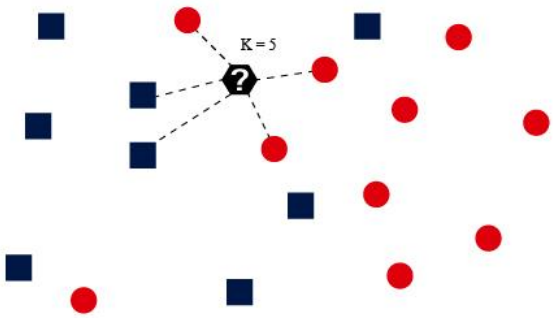


Fig. 4: Predicting an unknown point's class using KNN

In Figure 4 with a k equal to 5, the neighbors are shown and the point will be classified based on the weight of shapes as a red circle or a blue square and will be assigned in correlation with their values.

Suppose that r_i is the location vector of the sample and β is the property value. Firstly, the value of K should be set. Then the distance between the unknown object and existing samples is computed as d. for k = 1 to K, estimated value of β is calculated as Equation 2 and 3 [34].

$$\beta = \sum_{k=1}^{K} w_k \beta_k \quad (2)$$

$$w_k = \frac{1/d}{\sum_{k=1}^{K} \frac{1}{d}} \quad (3)$$

Which d is the stated distance and w_k is the weight considered for the vector.

Some other Classic ML algorithms were also studied in this paper including Decision Tree and Random Forest [35], ADABOOST [36], XGBOOST [37], CATBOOST [38], BAGGING [39], and LIGHTGBM [40]. For further details, see the references [27- 40].

The reason for choosing these methods would be the fact that SVM, Decision Tree Random Forest, and also Gradient Boosting algorithms are highly widespread worldwide and comparative study on them alongside some other Classic methods would offer valuable findings with a broader view.

3. Dataset introduction

A dataset, gathered by Padmini et al. [24] from several studies, has been implemented in this paper. The dataset is composed of load test data of real-sized shallow foundations on cohesionless sand soil with different densities. Real-sized refers to footings on the scale of existing buildings. The total number is 97 tests. 47 tests were performed on large-scale (dimensions between 0.5 m to 3.0 m) footings at the area of Degebo, Berlin in a submerged condition by multiple researchers and the angle of shearing resistance was measured through laboratory experiments [24]. The other 50 smaller-sized model footings (dimensions between 0.01 m and 0.15 m) are collected employing an experimental loading test designed by Gandhi [41]. Table 1 describe the dataset based on the literature.

Table. 1: The dataset properties considering the size of the footings

Count	Dimension range (m)			Label	Source
	Depth	Length	Width		
24	0 - 0.5	0.5 - 2.0	0.5	Large	Muhs and Weiß [42]
11	0 - 0.3	0.5 - 2.0	0.5	Large	Weiß [43]
5	0 - 0.3	1.2	0.6	Large	Muhs et al. [44]
2	0 - 0.2	3.0	1	Large	Muhs and Weiß [45]
5	0.7 - 0.9	1.0 - 3.0	1.0 - 3.0	Large	Briaud and Gibbens [46]
50	0.029 - 0.150	0.094 - 0.912	0.059 - 0.152	Small	Gandhi [41]

It is observed that there is a balance between the large and small-sized data in numbers which leads to more reliable answers. A thorough elaboration about the soil and foundation properties as well as the conditions during the experiments is provided in the corresponding references.

The input features are the width of footings (B), depth of footings (D), length-to-width ratio of footings (L/B), specific weight of soil mass (γ), friction angle of soil (φ) and the output feature is the bearing capacity of footings attained by load tests (q_u).

Table 2 demonstrates the statistical specifications for the following dataset parameters, including minimum, maximum, mean value, and standard deviation.

A part of the data is illustrated in Table 3. In addition, Figure 5 demonstrates the distribution of the dataset based on width, depth, and bearing capacity parameters in the form of histograms.

Table. 2: Statistical characteristic of the dataset

Type	Parameter	Symbol	Min	Max	Mean	S.D
Input	Width (m)	B	0.0585	3.016	0.4024	0.5111
	Depth (m)	D	0	0.889	0.1654	0.2004
	Length to Width ratio	L/B	1	6	3.1	2.1397
	Specific Weight (kN/m3)	γ	9.85	17.1	14.1810	2.6229
	Internal Friction Angle (deg)	φ	32	44.8	38.5526	3.2774
output	Bearing Capacity (kPa)	qu	58.5	2847	439.6169	530.8789

Table. 3: A sample of the utilized dataset

No.	B (m)	D (m)	L/B	Gamma (kN/m3)	Phi (deg)	qu (kPa)
1	0.6	0.3	2	9.85	34.9	270
2	0.6	0	2	10.2	37.7	200
3	0.6	0.3	2	10.2	37.7	570
4	0.6	0	2	10.85	44.8	860
5	0.6	0.3	2	10.85	44.8	1760
6	0.5	0	1	10.2	37.7	154
7	0.5	0	1	10.2	37.7	165
8	0.5	0	2	10.2	37.7	203
9	0.5	0	2	10.2	37.7	195
10	0.5	0	3	10.2	37.7	214

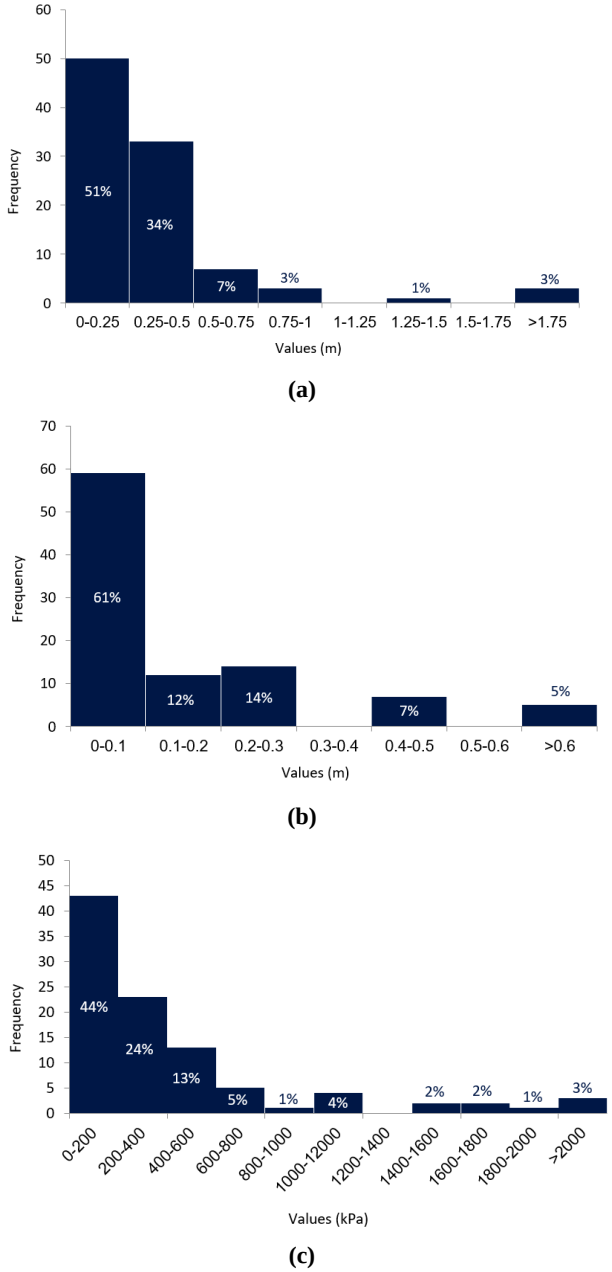


Fig. 5: Distribution of B, D, and qu. (a) Distribution of B ; (b) Distribution of D; (c) Distribution of qu

In terms of uncertainties and limitations associated with the dataset, the limited size of the dataset presents a challenge. Moreover, the friction angle of the soil was measured through sampling, and errors associated with in-site sampling are probable to be present in the data.

4. Accuracy metrics

After implementing ML methods, it is necessary to evaluate the accuracy and the performance of the model using accuracy measures including coefficient of determination (R^2), RMSE, and MAPE which will be further discussed, are used for the mentioned objective. In general, while reporting the error, the magnitude matters. So, all the following criteria have positive values.

4.1. Root Mean Square Error (RMSE)

This metric is calculated according to Equation 4 and is defined as the square root of the average of squared deviations of real values and predicted values [47].

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(z_{f_i} - z_{O_i}\right)^2}{N}} \quad (4)$$

Which z_f is the predicted value and z_O is the real value. The less RMSE is, the better the regression will be.

4.2. Mean Absolute Percentage Error (MAPE)

Another metric called MAPE can be used for the stated aim which is very similar to RMSE but uses absolute value instead of putting to power of two. The MAPE value should become minimum as well as RMSE and it is determined using Equation. 5 [48].

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right| \quad (5)$$

Where A_t is the real value and F_t is the predicted value.

4.3. Coefficient of Determination (R^2)

The coefficient of determination (R^2) is the most common metric and it simply shows the concordance of predicted values and the real ones and it is calculated using Equation 6 [49].

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (6)$$

In which RSS is the sum of squares of residuals and TSS is the total sum of squares. $\hat{y}_i$ stands for predicted value, y_i for real values and y_i for the mean value of the parameter under consideration. R^2 is a ratio and as it approaches 1.00, the precision of results increases.

4.4. Mean Absolute Error (MAE)

While modeling performance evaluation, it is suggested to determine Mean Absolute Error (MAE) as a more natural measure with less ambiguity than RMSE [50]. MAE is a straightforward metric and is defined as the average of absolute errors. Equation 7 represents the calculation of MAE [50].

$$MAE = n^{-1}\sum_{i=1}^{n}|e_i| \quad (7)$$

e_i is the difference between absolute and predicted values of data and n is the data quantity.

4.5. Variance Accounted For (VAF)

Variance Accounted For (VAF) simply indicates the accuracy of the model based on the variance of the predicted values and the real values. Equation 8 provides VAF calculation [51]:

$$VAF = \left[1 - \frac{var(y - y')}{var(y)}\right] \times 100 \quad (8)$$

Where y is the real value and y' is the predicted value. It is obvious that as the VAF approaches 100%, the model performs better.

4.6. A20 Score

A20 score is a recently proposed reliability criterion for Neural Networks and is expressed as the ratio of well-predicted samples (within a 20% deviation from the actual values). Equation 9 shows the calculation of a20 index. [52]:

$$a20_{index} = \frac{m_{20}}{M} \quad (9)$$

M is data quantity and m_{20} specifies the count of predictions within ±20% of the actual values. As this metric gets close to 1.0, the regression results get more reliable.

While examining ML models' performance, it is necessary to employ different accuracy criteria to find a more generalized answer. Moreover, some metrics have advantages in certain models compared to others (e.g., as stated before, sometimes MAE is less ambiguous than RMSE).

5. Methodology

5.1. Preprocessing and feature selection

Before performing any data analysis model, it is necessary to carry out preprocessing on the dataset. There are two main reasons for that which are the probability of defect in the data and the dataset compatibility for the modeling process [53]. Some issues with a dataset might be noise in the data, missing attributes, overfitting due to numerous features, symbolic data (like the shape of foundations), and a small amount of data. In preprocessing these concerns are addressed [53].

In the context of this study, the entire dataset is numerical and there is no need for converting symbolic data. Besides, there is no missing data and all the inputs and outputs are available from Table 2 . It is obvious that the variance of the variables is not very high in values.

However, since in geotechnical studies, a lack of data is apparent, the issue of too much data wouldn't occur and instead of dataset reduction, feature selection techniques, consisting of filter methods, wrapper methods, and embedded methods must be chosen based on their accuracy and efficiency in certain conditions and studies [54]. Since there are only five input features in the present case and they are the most basic characteristics of a foundation, feature set reduction wouldn't be necessary. Despite this, to obtain a more detailed view, a correlation analysis for all the input and output datasets was done based on the Pearson Correlation Coefficient. The procedure of determining the correlation coefficient is as Equation 10 [55]:

$$\boldsymbol{r_{xy}} = \frac{\sum_{i=1}^{n}(\boldsymbol{x_i} - \bar{\boldsymbol{x}})(\boldsymbol{y_i} - \bar{\boldsymbol{y}})}{\sqrt{\sum_{i=1}^{n}(\boldsymbol{x_i} - \bar{\boldsymbol{x}})^2}\sqrt{\sum_{i=1}^{n}(\boldsymbol{y_i} - \bar{\boldsymbol{y}})^2}} \quad (10)$$

Where n is the sample size, x_i and y_i are individual points (one in the predicted values set and another in the real values set) and $\bar{\boldsymbol{x}}$, $\bar{\boldsymbol{y}}$ are mean values of x and y sample sets respectively. As the correlation coefficient gets closer to 1 or -1, the x and y features are more related to each other. Figure 6 portrays the correlation matrix of input and output features.

Many interpretations can be derived from Figure 6**.** It is indicated that bearing capacity (qu) is significantly correlated to the depth (D) of the footing in the prepared data and feature selection presence of D is crucial. Since all the features are strongly related to the output (based on Figure 5, the numbers are 0.25 and more), all the input features are selected.

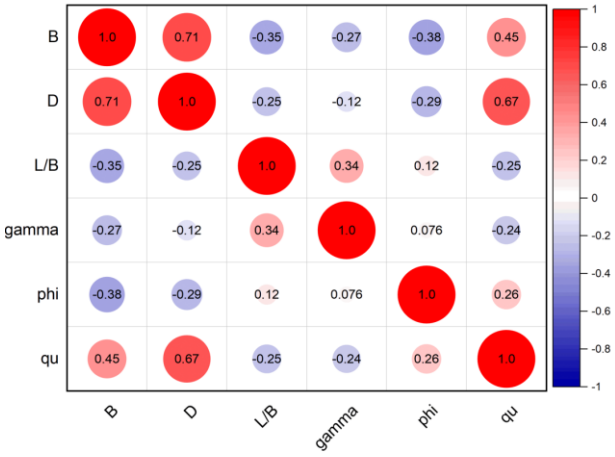


Fig. 6: Correlation matrix of features

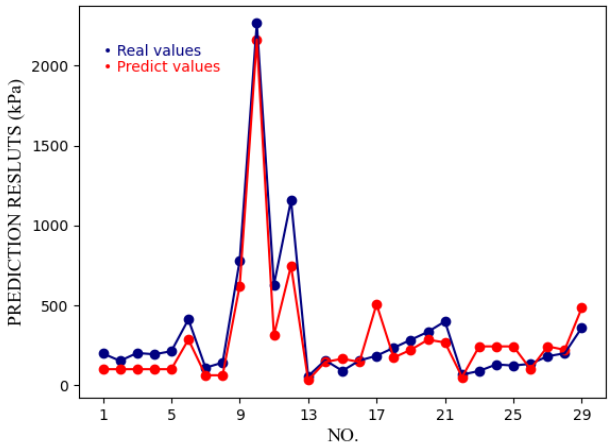
Along with that, for better compatibility of different features and output, the use of distance-based models, consistency, and the presence of different scale values, all the feature space was normalized on a certain scale of [-1, 1] before modeling.

5.2. The effect of sampling parameters optimization

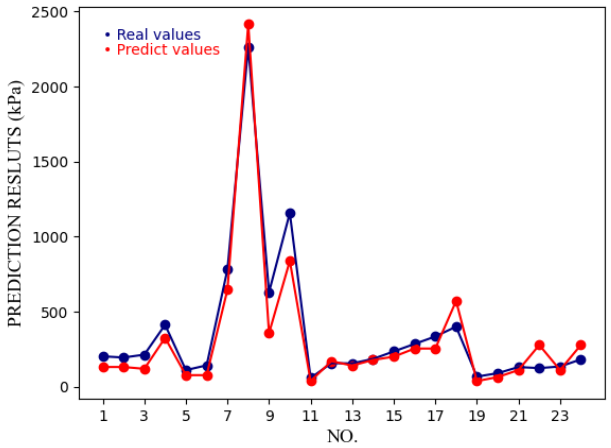
As previously mentioned, in the implementation of machine learning methods, it is necessary to split the data to train and test sets. There are myriad possible ways of splitting the dataset. To make the results comparable, a feasible approach must be taken into account for all methods. The random state of picking data, which has a built-in definition in Python libraries, and the train set size ratio are chosen as splitting parameters.

It is necessary to find the optimum values for the splitting parameters using trial and error. To achieve this, the results of the modeling must be compared (hyperparameters and modeling elaboration will be discussed further in the literature). The results were indicated for many different values for the proportion of the data used as training sets for all the models. To demonstrate the results, a scatter of bearing capacity values on the vertical axis for the selected test set was designed. Red dots and lines show the predicted value of the specific method and blue dots and lines plot real values. The amount of overlap shows the accuracy of the regression.

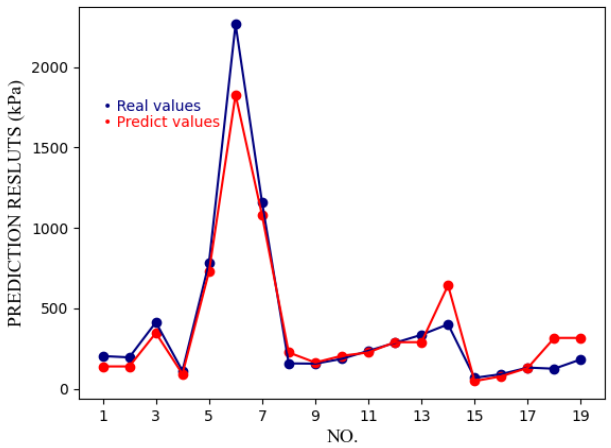
Notice that the samples are not sorted to show the randomness of sampling. Figure 7 illustrates the results for the decision tree algorithm for 0.7, 0.75, and 0.8 train ratios as some examples. Accuracy metrics are reported under the plots (here we use R2, RMSE, and MAPE). In addition, the random state of choosing the train and test data has a significant effect on the outcome. Several sampling patterns were examined using built-in Python libraries definitions.



(a)



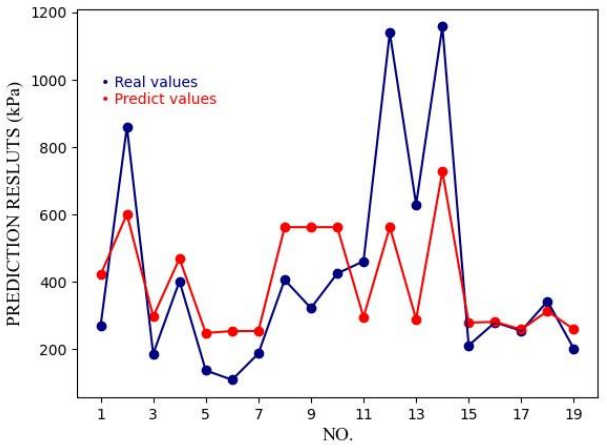
(b)



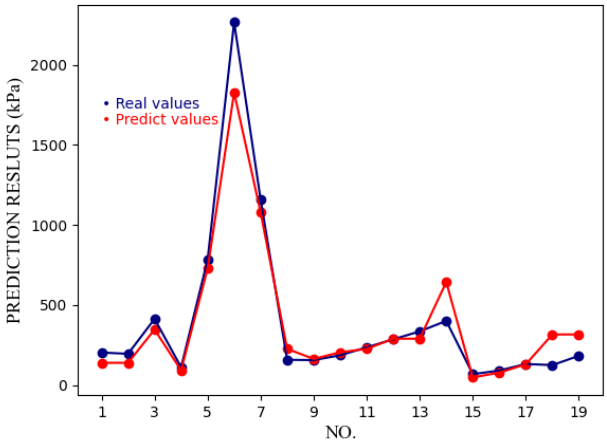
(c)

Fig. 7: Decision Tree results with different Train Ratios: (a) Train Ratio = 0.7, R^2 = 0.89, RMSE = 143, MAPE = 2.4; (b) Train Ratio = 0.75, R^2 = 0.93, RMSE = 123, MAPE = 10.8 ;and (c) Train Ratio = 0.8, R^2 = 0.93, RMSE = 130, MAPE = 3.7

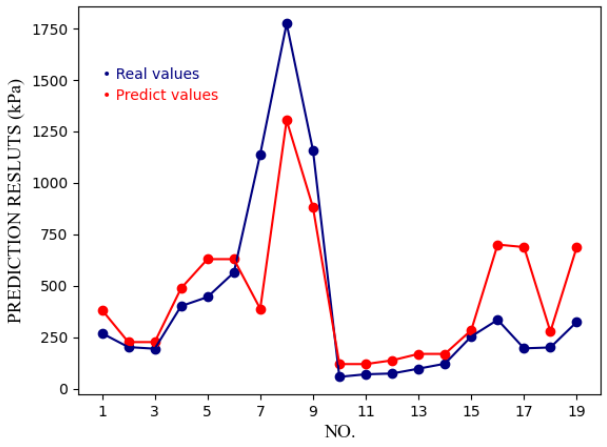
Figure 8 portrays the output of the decision tree algorithm using three different patterns (train ratio is equal to 0.8).



(a)



(b)



(c)

Fig. 8: Decision Tree results with different Random States: (a) Random State 1, R^2 = 0.44, RMSE = 228, MAPE = 11.2; (b) Random State 2, R^2 = 0.93, MAPE = 8.9, RMSE = 132; and (c) Random State 3, R^2 = 0.6, RMSE = 280, MAPE = 48.3

5.3. Classic ML models development and hyperparameters

When utilizing ML methods, it is essential to choose optimized and compatible hyperparameters for a trustworthy performance and results of the models.

The protocol of ML regression bears a resemblance to all the models and consists of reading the dataset, pre-processing (Standard Scaling), fitting the regressor, regression hyperparameter optimization based on suitable metrics, and plotting the final results.

Model development initiated by data splitting is a critically important and highly effective part, which was done as mentioned in the previous section. Table 4 shows the optimization results of the sampling procedure via trial and error for the decision tree. Various algorithms may have different optimum parameter values and the stated validation process should be done for each. In this study, after validating 12 chosen algorithms, the training ratio was equal to 0.80 and the random state 2 was considered in learning.

Table. 4: Data splitting parameter optimization (Decision Tree)

Splitting parameter	Value	Result (R2)
Train set ratio	0.70	0.89
	0.75	0.93
	0.80	0.93
	0.85	0.90
Random state (Train ratio = 0.8)	1	0.44
	2	0.93
	3	0.6

XGBOOST simply follows the protocol. The number of trees in the model (n_estimators) was found out through trial and error (from 1 to 100) and R2, Mean Absolute Error (MAE), and Mean Squared Error (MSE) are used as the criteria for optimization.

In the Decision Tree, the selectable hyperparameters are the Node Splitting Criterion which indicates the quality of splitting in the tree and limitations for the shape of the tree. Gini impurity index as a robust splitting criterion is selected for the model.

While employing KNN it is necessary to find an optimum k value and it was done by trial and error (from 1 to the size of the sample) using R2 as the score (the k values change to maximize R2).

In SVM, the main hyperparameter is the selection of a compatible kernel. Table 5 shows the kernel selection via trial and error.

Table. 5: SVM kernel selection

Kernel	Result (R2)
Poly	0.70
RBF	0.91
Sigmoid	< 0.1

The maximum number of iterations for all the methods is set to 500 to prevent overfitting and excessive time allocation. All other hyperparameters were selected similarly. The hyperparameters are shown in Table 6.

Table. 6: Hyperparameters value for Classic ML methods

Model	Hyperparameter	Value
XGBOOST	Regressor	Linear
	N_estimators	60
DT	Splitting criterion	Gini
	Shape limitation	None
KNN	k	5
	Weights	Uniform
	Metric	Standard Euclidean distance
ADABOOST	N_estimators	50
	Learning rate	1.00
SVM	Kernel	RBF
	Degree	3
CATBOOST	Learning rate	1.00
	Maximum itterations	500
RF	N_estimators	100
	Shape limitation	None
LIGHTGBM	Boosting type	Gradient boosting
	Number of leaves	31
BAGGING	N_estimators	10
	Maximum samples	1

5.4. ANN model development

While training an ANN model using backpropagation, a separate series of data, termed validation data, must be designated for calibration of the learning process (justifying the weights of the nodes). Therefore, 10 percent of the train data was allocated to validation data. The Table 7 shows the specified structure of the networks.

Table. 7: Hyperparameters value for Classic ML methods

Layer	Number of Nodes	Activation Function
Input Layer	5	
Hidden Layer 1	100	SELU
Hidden Layer 2	100	SELU
Output Layer	1	RELU

Since there are 5 input features, the input layer of the networks consists of 5 nodes. Scaled Exponential Linear Unit (SELU) was selected as the activation function of the hidden layers because of its self-normalization property which helps in maintaining a consistent distribution of the activations across layers, resulting in speeding the process up. Equation 11 shows the SELU activation function [56].

$$SELU(x) = \gamma(\max(0\ x) + \min(0\ \alpha(e^x - 1))) \quad (11)$$

Which $\alpha \approx 1.67326$ and $\gamma \approx 1.05070$.

Rectified Linear Unit (ReLU), as shown in Equation 12, was selected as the activation function of the output layer due to its simplicity, non-negative predictions, and compatibility with regression tasks [57].

$$ReLU(x) = \max(0.x) \quad (12)$$

Adaptive Moment Estimation (Adam) is an algorithm for gradient-based optimization of objective functions. Adam is an adaptive method and changes the learning rate over time based on gradients before [58]. More details on the Adam optimizer are provided in the reference. Adam was chosen as the optimization algorithm owing to its self-tuning, bias-corrected estimations, and robust performance, with a learning rate of 0.1 and MAE as the loss function. Afterward, the model was trained during 500 epochs of backpropagation with MAE as the loss function.

6. Results and discussion

6.1. Dataset splitting

As outlined in section 5.2, it is evident that the final results are significantly connected to the splitting procedure. Subsequently, after extensive iterative refinement for all methods, the training ratio was set to 0.8 and a specific random state was selected for a dependable and balanced model performance (for example the incorporation of different foundation sizes in calibration and validation). Padmini et al. [24] also used 80% of the data for calibration and 20% for validation.

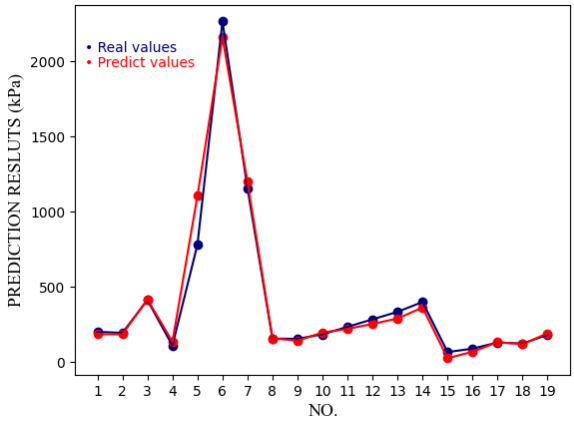
With a comparison between the states of b (Train set ratio = 0.75) and state c (0.8), as depicted in Figure 7, and also Table 4 (0.85), it is derived that the size increment of the train data would not necessarily generate better outcomes, and the risk of overfitting should be controlled. So, train and test data sizes remarkably impact the learning process and an optimum value should be considered.

Moreover, Figure 8 highlights that the random state has a considerably more significant effect on the outcome and various states must be evaluated to validate the learning process.

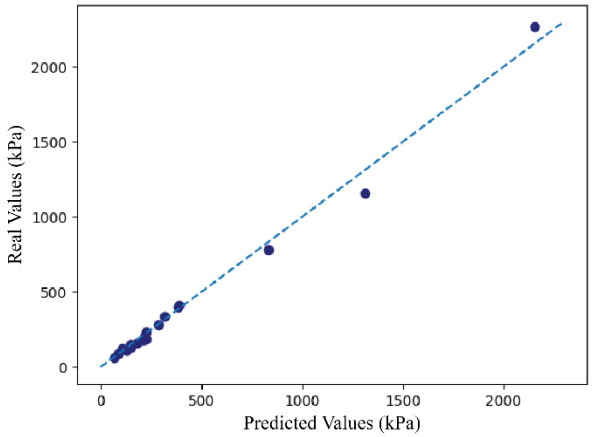
6.2. Results evaluation

After the machine has learned to predict values, it is time to proceed comparison with the real values of the test set and evaluate the results. Figure 7 and 8 presented one approach

to results visualization. In the following section, another result demonstrates a correlation of real values and predicted values (of bearing capacity) in the form of a scatter is regarded. The dotted line is the line of equality and as the points get close to the line, the predicted value gets closer to the real values. Figure 9 illustrates the results of the use of Artificial Neural Networks.



(a) Predicted versus real values for different observations



(b) Correlation between predicted and actual values

Fig. 9: ANN results: (a) Predicted versus real values for different observations and (b) Correlation between predicted and actual values

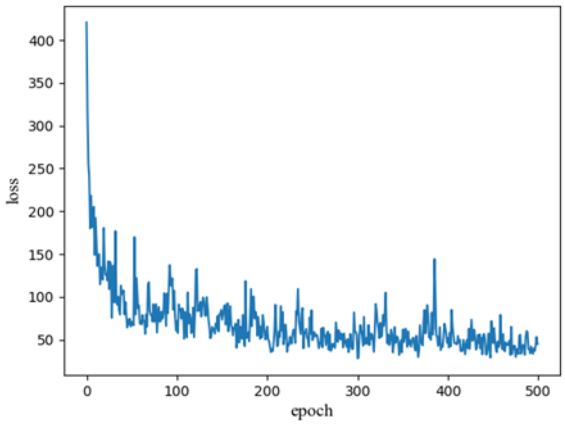
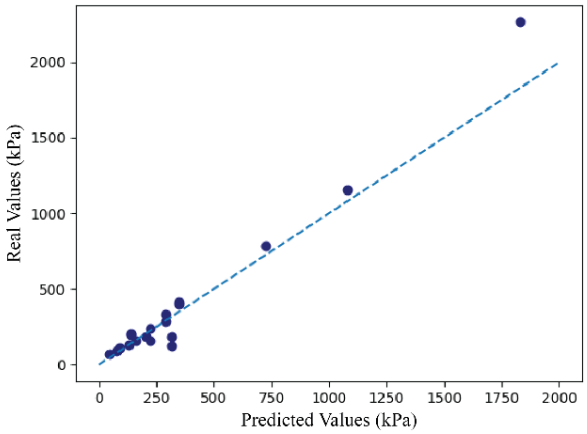
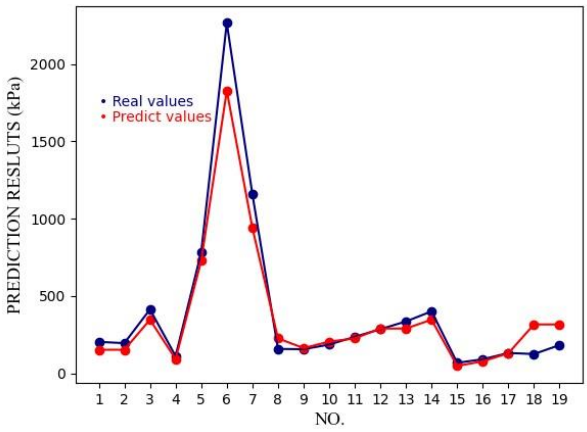


Fig. 10: ANN loss diagram after 500 epochs

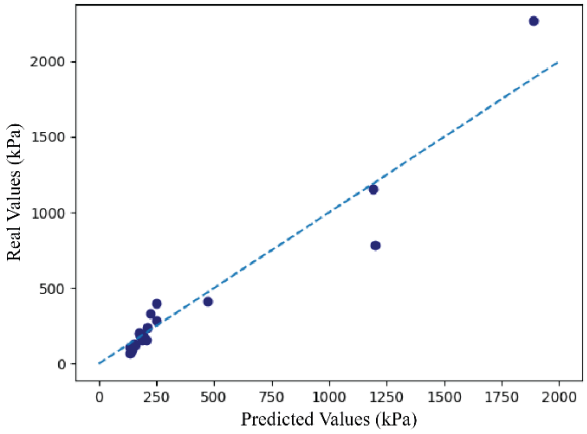
Excellent estimations were noted while using ANN. But despite the precision, the learning process was done in several learning epochs which have taken plenty of time. In clear terms, ANN will take more time than Classic ML algorithms but has higher accuracy. Figure 10 shows the reduction in loss function after 500 epochs. MAE was chosen as the loss function here. 10 Classic ML algorithms are also examined and the results of some are illustrated in Figure 11.



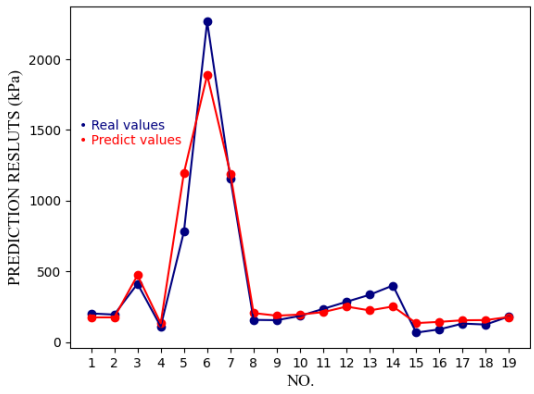
(a) Best-fit line plot for Decision Tree results



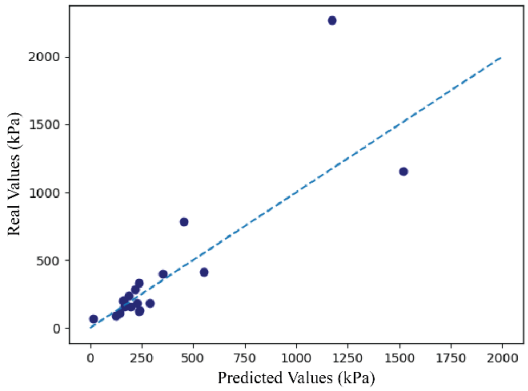
(b) Real vs. Predicted values for Decision Tree



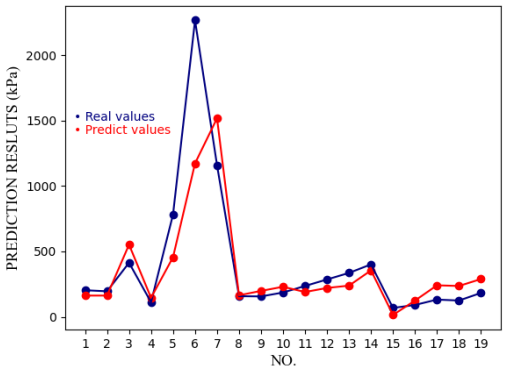
(c) Best-fit line plot for KNN results



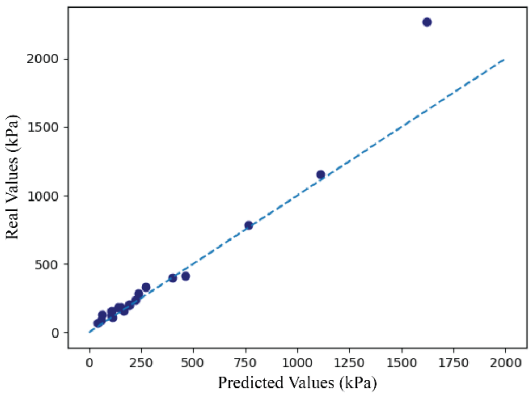
(d) Real vs. Predicted values for KNN



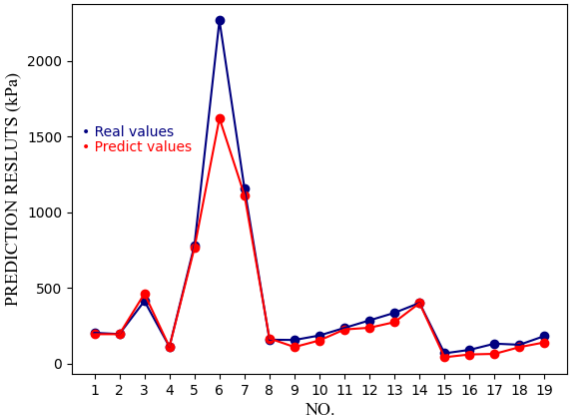
(e) Best-fit line plot for SVM Poly results



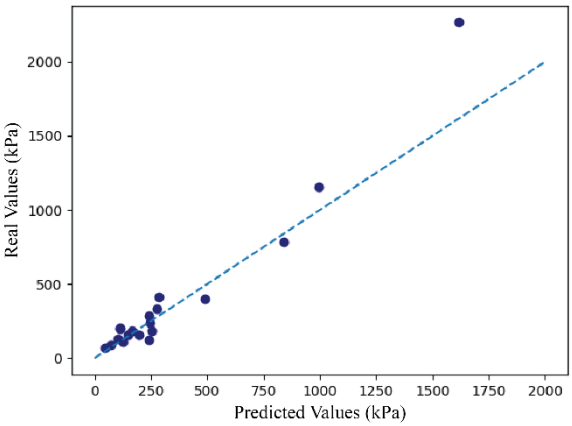
(f) Real vs. Predicted values for SVM Poly



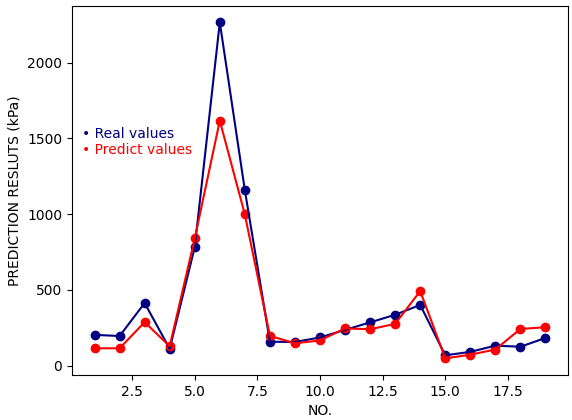
(g) Best-fit line plot for SVM RBF results



(h) Real vs. Predicted values for SVM RBF



(i) Best-fit line plot for CatBoost results



(j) Real vs. Predicted values for CatBoost model

Fig. 11: The result of used methods: (a, b) Decision Tree; (c, d) KNN; (e, f) SVM, poly kernel; (g, h) SVM, rfb kernel, and (i, j) CATBOOST

From the figures, it is evident that more errors have occurred in the greater data values. One reason can be data shortage in larger data values. However valid predictions are shown to be done using both ANN and classic ML algorithms. On the whole, Table 8 reveals the accuracy scores of the utilized methods.

To demonstrate the significance of our findings, a comparison of the previously studied models with the applied methods in this study has been provided in Table 9, Based on reported R2 and RMSE values. Table 9 also reveals a scoring system, developed to suggest a priority for using the methods. The least accurate model takes the score of 1 and the most accurate takes the score of 17, then the scores associated with R2 and RMSE are added together to assign a total score to the methods. Table 9 validates the efficacy of our research. Initially, most of the models in this study have shown a better reliability than the analytical bearing capacity theories. Additionally, the XGBOOST model in our study outperformed the FIS model by Padmini et al. despite its simplicity relative to Fuzzy Interface models. And more importantly, our ANN model exhibited better precision compared to the previous study.

Table. 8: Accuracy metrics values

Priority	Method	R2	RMSE	MAPE	MAE	VAF	A20 Score
1	ANN	0.988	56.44	38.72	0.28	98.9	0.95
2	XGBOOST	0.973	677.21	39.27	0.22	97.5	
3	DECISION TREE	0.933	132.00	8.90	0.44	93.4	
4	KNN	0.926	139.56	81.15	2.70	92.6	
5	ADABOOST	0.918	148.00	42.30	1.67	92.6	
6	SVM RBF	0.912	152.00	14.20	0.75	92.3	
7	CATBOOST	0.896	165.00	3.70	0.19	90.4	
8	RF	0.83	209.00	11.80	0.87	83.4	
9	BAGGING	0.75	255.00	26.30	0.88	80.8	
10	SVM POLY	0.70	282.00	6.80	0.36	70.2	
12	LIGHTGBM	0.54	346.00	67.40	3.55	56.1	
	Min	0.54	56.44	3.70	0.19	56.07	
	Max	0.99	677.21	81.15	3.55	98.93	

6.3. Importance Analysis

To explore the relative importance of the input features, a sensitivity analysis needs to be conducted. Figure 12 represents the relative importance of the input variables according to XGBOOST algorithm results. Here, we used the gain index for each feature. Gain determines the improvement in the model performance when a feature is used to split the data in tree nodes. Gain is calculated based on the Equation 13 [59].

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} + \frac{(G_L + G_R)^2}{H_L + H_R}\right] \quad (13)$$

Where G_L and G_R are the sum of the gradients for the left and right child nodes after the split and H_L and H_R are the sum of the second-order derivatives for the left and right child nodes after the split.

The reference [55] provides further elaboration on gain and its determination.

As the gain Index gets closer to 1 (100%), the results get more sensitive to the corresponding feature.

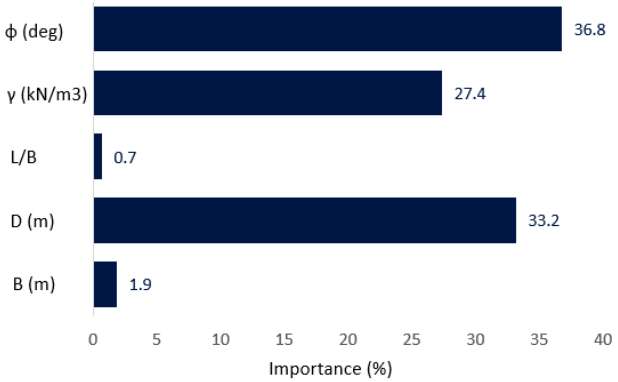


Fig. 12: Importance analysis results in XGBOOST

Figure 12 reveals that the results are first extremely responsive to the internal friction angle of the soil (φ) with 36.8% sensitivity and then to the depth of the foundation (D) as the second most important input with a slight difference in relative importance. The third most influential variable is proven to be the specific weight of the soil (γ). The other features consisting of length to width ratio of the footing (L/B) and the width of the foundation (B) are considerably less effective in the outcome in contrast to other inputs. So in the feature selection phase, it is assumed that with

removing B and L/B of the features the results won't be affected significantly in such conditions.

Table. 9: Suggested priorities

Priority	Method	R2	RMSE	Rating Score
1	ANFIS (Padmini et al.)	0.992 17	52.30 17	34
2	ANN (this paper)	0.988 16	56.44 16	32
3	ANN (Padmini et al.)	0.983 15	77.20 15	30
4	XGBOOST	0.973 14	84.07 14.00	28
5	FIS (Padmini et al.)	0.972 13	98.00 13	26
6	DECISION TREE	0.933 12	131.59 12	24
7	KNN	0.926 11	139.56 11	22
8	ADABOOST	0.918 10	150.55 10	20
9	SVM RBF	0.912 9	152.15 9	18
10	CATBOOST	0.896 8	165.01 8	16
11	Meyerhof (Padmini et al.)	0.874 7	207.30 7	14
12	RF	0.83 6	209.00 6	12
13	Vesic (Padmini et al.)	0.815 5	251.30 5	10
14	BAGGING	0.75 4	255.00 4	8
15	Hansen (Padmini et al.)	0.727 3	305.30 2	5
16	SVM POLY	0.70 2	282 3	5
17	LIGHTGBM	0.54 1	346 1	2

8. Conclusion

The most important factor in the design of foundations is bearing capacity determination and that is the reason myriad researchers have always been studying more precise ways to achieve the stated goal. Concisely stated data splitting parameters consisting of train and test data sizes and the random state of picking data for each have a significant impact on the learning process and the final results. It is concluded that a training sample with the size of 80 percent of the dataset is an optimal choice for better accuracy.

Moreover, growing the training sample would not essentially induce better results. Another final remark is that based on diagrams, it can be derived that results are less accurate in larger data values. The reason can be that there is less amount of data in those regions. It is also indicated that the internal friction angle of the soil and depth of footing are the most important inputs during the implementation of ML methods in similar cases.

Overall, in granular cohesionless soils, which were the study case in this paper, the use of Artificial Neural Networks and classic ML algorithms will result in reliable accurate predictions. XGBOOST, Decision Tree, ADABOOST, and KNN are proven to be more reliable than the analytical theories and while facing a design demand, utilizing them using an acceptable dataset along with other analytical methods will help us with more dependable outcomes. However, it would be advantageous to address some limitations of the study in the future. For instance, collecting sufficient data, resolving uncertainties with the soil and experiment, testing different models and attributes, different types of foundations, and deriving new correlations would be beneficial.

References

[1] Das, B. M., & Sivakugan, N. (2018). Principles of foundation engineering. Cengage learning.

[2] Dewaikar, D. M., & Mohapatra, B. G. (2003). Computation of bearing capacity factor Nγ-Prandtl's mechanism. Soils and foundations, 43(3), 1-10.

[3] Dewaikar, D. M., & Mohapatro, B. G. (2003). Computation of bearing capacity factor N γ—Terzaghi's mechanism. International Journal of Geomechanics, 3(1), 123-128.

[4] Meyerhof, G. G. (1963). Some recent research on the bearing capacity of foundations. Canadian geotechnical journal, 1(1), 16-26.

[5] Hansen, J. B. (1970). A revised and extended Equation for bearing capacity.

[6] Vesić, A. S. (1973). Analysis of ultimate loads of shallow foundations. Journal of the Soil Mechanics and Foundations Division, 99(1), 45-73.

[7] Hahn, B. H., & Valentine, D. T. (2017). Introduction to Numerical Methods. Essent. MATLAB Eng. Sci, 295-323.

[8] Fowler, D., & Robson, E. (1998). Square root approximations in Old Babylonian mathematics: YBC 7289 in context. Historia mathematica, 25(4), 366-378.

[9] Goldstine, H. H. (2012). A History of Numerical Analysis from the 16th through the 19th Century (Vol. 2). Springer Science & Business Media.

[10] Bathe, K. J. (2007). Finite element method. Wiley encyclopedia of computer science and engineering, 1-12.

[11] Rosen, S. (1969). Electronic computers: A historical survey. ACM Computing Surveys (CSUR), 1(1), 7-36.

[12] Morgenstern, N. U., & Price, V. E. (1965). The analysis of the stability of general slip surfaces. Geotechnique, 15(1), 79-93.

[13] Nilsson, N. J. (1998). Artificial intelligence: a new synthesis. Morgan Kaufmann.

[14] Fradkov, A. L. (2020). Early history of machine learning. IFAC-PapersOnLine, 53(2), 1385-1390.

[15] Yeh, Y. C., Kuo, Y. H., & Hsu, D. S. (1993). Building KBES for diagnosing PC pile with artificial neural network. Journal of Computing in Civil Engineering, 7(1), 71-93.

[16] Chan, W. T., Chow, Y. K., & Liu, L. F. (1995). Neural network: an alternative to pile driving Equations. Computers and geotechnics, 17(2), 135-156.

[17] Lee, I. M., & Lee, J. H. (1996). Prediction of pile bearing capacity using artificial neural networks. Computers and geotechnics, 18(3), 189-200.

[18] Shahin, M. A., Jaksa, M. B., & Maier, H. R. (2000). Predicting the settlement of shallow foundations on cohesionless soils using back-propagation neural networks. Australia: Department of Civil and Environmental Engineering, University of Adelaide.

[19] Rezania, M., & Javadi, A. A. (2007). A new genetic programming model for predicting settlement of shallow foundations. Canadian Geotechnical Journal, 44(12), 1462-1473.

[20] Samui, P. (2008). Support vector machine applied to settlement of shallow foundations on cohesionless soils. Computers and Geotechnics, 35(3), 419-427.

[21] Gajan, S. (2021). Application of machine learning algorithms to performance prediction of rocking shallow foundations during earthquake loading. Soil Dynamics and Earthquake Engineering, 151, 106965.

[22] Altınok, E., & Ülker, M. B. (2023, September). Ultimate bearing capacity of closed-ended piles using nonlinear machine learning methods. In AIP Conference Proceedings (Vol. 2849, No. 1). AIP Publishing.

[23] Jibanchand, N., & Devi, K. R. (2023). Application of ensemble learning in predicting shallow foundation settlement in cohesionless soil. International Journal of Geotechnical Engineering, 1-12.

[24] Padmini, D., Ilamparuthi, K., & Sudheer, K. P. (2008). Ultimate bearing capacity prediction of shallow foundations on cohesionless soils using neurofuzzy models. Computers and Geotechnics, 35(1), 33-46.

[25] Kalinli, A., Acar, M. C., & Gündüz, Z. (2011). New approaches to determine the ultimate bearing capacity of shallow foundations based on artificial neural networks and ant colony optimization. Engineering Geology, 117(1-2), 29-38.

[26] Colliot, O. (Ed.). (2023). Machine Learning for Brain Disorders.

[27] Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In Machine learning techniques for multimedia: case studies on organization and retrieval (pp. 21-49). Berlin, Heidelberg: Springer Berlin Heidelberg.

[28] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[29] Elsafi, S. H. (2014). Artificial neural networks (ANNs) for flood forecasting at Dongola Station in the River Nile, Sudan. Alexandria Engineering Journal, 53(3), 655-662.

[30] Jakkula, V. (2006). Tutorial on support vector machine (svm). School of EECS, Washington State University, 37(2.5), 3.

[31] Samui, P. (2011). Prediction of pile bearing capacity using support vector machine. International Journal of Geotechnical Engineering, 5(1), 95-102.

[32] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27.

[33] Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. In On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings (pp. 986-996). Springer Berlin Heidelberg.

[34] Lee, K. C., & Lee, C. H. (2020) Prediction of Complicated Mathematical Problems by Machine Learning of KNN Regression.

[35] Faouzi, J., & Colliot, O. (2023). Classic machine learning algorithms. Machine Learning for Brain Disorders.

[36] Ying, C., Qi-Guang, M., Jia-Chen, L., & Lin, G. (2013). Advance and prospects of AdaBoost algorithm. Acta Automatica Sinica, 39(6), 745-758.

[37] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[38] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Advances in neural information processing systems, 31.

[39] González, S., García, S., Del Ser, J., Rokach, L., & Herrera, F. (2020). A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. Information Fusion, 64, 205-237.

[40] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W.,& Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

[41] Gandhi, G. N. (2001). Study of Bearing Capacity Factors Developed from Laboratory Experiments on Shallow Footings Founded on Cohesionless Soil.

[42] Muhs, H., & Weiß, K. (1971). Untersuchung von grenztragfaehigkeit und setzungsverhalten flachgegruendeter einzelfundamente in ungleichfoermingen nichtbindigen boeden.

[43] Weiß, K. (1970). Der Einfluß der Fundamentform auf die Grenztragfähigkeit flachgegründeter Fundamente, Untersuchungen ausgef von Klaus Weiß: mit 14 Zahlentaf. Ernst.

[44] Muhs, H., Elmiger, R., & Weiß, K. (1969). Sohlreibung und Grenztragfähigkeit unter lotrecht und schräg belasteten Einzelfundamenten; mit 128 Bildern und 13 Zahlentafeln. Ernst.

[45] Muhs, H., & Weiss, K. (1974). Inclined load tests on shallow strip footings.

[46] Briaud, J. L., & Gibbens, R. (1999). Behavior of five large spread footings in sand. Journal of geotechnical and geoenvironmental engineering, 125(9), 787-796.

[47] Barnston, A. G. (1992). Correspondence among the correlation, RMSE, and Heidke forecast verification measures; refinement of the Heidke score. Weather and Forecasting, 7(4), 699-709.

[48] Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. International journal of forecasting, 22(4), 679-688.

[49] Renaud, O., & Victoria-Feser, M. P. (2010). A robust coefficient of determination for regression. Journal of Statistical Planning and Inference, 140(7), 1852-1862.

[50] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research, 30(1), 79-82.

[51] Jahed Armaghani, D., Hajihassani, M., Monjezi, M., Mohamad, E. T., Marto, A., & Moghaddam, M. R. (2015). Application of two intelligent systems in predicting environmental impacts of quarry blasting. Arabian Journal of Geosciences, 8, 9647-9665.

[52] Armaghani, D. J., Hatzigeorgiou, G. D., Karamani, C., Skentou, A., Zoumpoulaki, I., & Asteris, P. G. (2019). Soft computing-based techniques for concrete beams shear strength. Procedia Structural Integrity, 17, 924-933.

[53] Famili, A., Shen, W. M., Weber, R., & Simoudis, E. (1997). Data preprocessing and intelligent data analysis. Intelligent data analysis, 1(1), 3-23.

[54] Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1200-1205). Ieee.

[55] Zhou, H., Deng, Z., Xia, Y., & Fu, M. (2016). A new sampling method in particle filter based on Pearson correlation coefficient. Neurocomputing, 216, 208-215.

[56] Rasamoelina, A. D., Adjailia, F., & Sinčák, P. (2020, January). A review of activation function for artificial neural network. In 2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI) (pp. 281-286). IEEE.

[57] Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function.

[58] Jiang, Y., & Han, F. (2017). A hybrid algorithm of adaptive particle swarm optimization based on adaptive moment estimation method. In Intelligent Computing Theories and Application: 13th International Conference, ICIC 2017, Liverpool, UK, August 7-10, 2017, Proceedings, Part I 13 (pp. 658-667). Springer International Publishing.

[59] Chen, M., Liu, Q., Chen, S., Liu, Y., Zhang, C. H., & Liu, R. (2019). XGBoost-based algorithm interpretation and application on post-fault transient stability status prediction of power system. IEEE Access, 7, 13149-13158.