Prediction of Pile Bearing Capacity Using XGBoost Algorithm: Modeling and Performance Evaluation

Maaz Amjad [1], Irshad Ahmad [1], Mahmood Ahmad [2,3,*], Piotr Wróblewski [4,5], Paweł Kamiński [6] and Uzair Amjad [1]

1   Department of Civil Engineering, University of Engineering and Technology, Peshawar 25120, Pakistan; maazamjad.civ@uetpeshawar.edu.pk (M.A.); drirshadahmad@uetpeshawar.edu.pk (I.A.); 18pwciv5168@uetpeshawar.edu.pk (U.A.)
2   Department of Civil Engineering, Faculty of Engineering, International Islamic University Malaysia, Jalan Gombak, Selangor 50728, Malaysia
3   Department of Civil Engineering, University of Engineering and Technology Peshawar (Bannu Campus), Bannu 28100, Pakistan
4   Faculty of Engineering, University of Technology and Economics H. Chodkowska in Warsaw, Jutrzenki 135, 02-231 Warsaw, Poland; piotr.wroblewski@wat.edu.pl
5   Faculty of Mechatronics, Armament and Aerospace of the Military University of Technology, Sylwestra Kaliskiego 2, 00-908 Warsaw, Poland
6   Faculty of Civil Engineering and Resource Management, AGH University of Science and Technology, 30-059 Krakow, Poland; pkamin@agh.edu.pl
*   Correspondence: ahmadm@iium.edu.my

Abstract: The major criteria that control pile foundation design is pile bearing capacity (P_u). The load bearing capacity of piles is affected by the various characteristics of soils and the involvement of multiple parameters related to both soil and foundation. In this study, a new model for predicting bearing capacity is developed using an extreme gradient boosting (XGBoost) algorithm. A total of 200 driven piles static load test-based case histories were used to construct and verify the model. The developed XGBoost model results were compared to a number of commonly used algorithms— Adaptive Boosting (AdaBoost), Random Forest (RF), Decision Tree (DT) and Support Vector Machine (SVM) using various performance measure metrics such as coefficient of determination, mean absolute error, root mean square error, mean absolute relative error, Nash–Sutcliffe model efficiency coefficient and relative strength ratio. Furthermore, sensitivity analysis was performed to determine the effect of input parameters on P_u. The results show that all of the developed models were capable of making accurate predictions however the XGBoost algorithm surpasses others, followed by AdaBoost, RF, DT, and SVM. The sensitivity analysis result shows that the SPT blow count along the pile shaft has the greatest effect on the P_u.

Keywords: pile bearing capacity; machine learning; extreme gradient boosting; adaptive boosting; random forest; decision tree; support vector machine

1. Introduction

A pile is a long, structural element used to allow structural loads to be transferred to the soils at a depth below the structure's base. Axial, lateral, and moment loads are examples of structural loads. The load transmission mechanism is based on pile toe and pile shaft resistances [1]. Deep foundations are another word for pile foundations that are often used in practice. Pile foundations are used to support structures that cannot be supported economically on shallow foundations. The most significant factor when designing a pile foundation is pile carrying capacity (P_u) [2]. Various ways to determine pile carrying capacity have been used during the years of research and development [3–13], including dynamic analysis, high strain dynamic test, pile load test, cone penetration test (CPT) and in situ tests. Some research, claims that the aforementioned connections

exaggerate the bearing capability [14]. However, the pile load test is considered as one of the best methods to determine the pile bearing capacity, although this strategy is costly for small-scale projects and time-consuming [10], it is critical to find a more practical approach. As a result, many studies using in situ test data to assess pile carrying capacity have been performed [9].

Lopes and Laprovitera [15], and Decort [16] proposed different formulas for determining pile carrying capacity for several soils, including clay and sand. Conventional approaches have used numerous main parameters to determine the mechanical properties of piles, including the diameter of pile, length of pile, type of soil, and SPT blow counts of each layer. Nevertheless, the selection of relevant parameters, along with the failure in covering other parameters, have led to the disagreement of results given by various approaches [17]. As a result, the development of an optimal model for selecting an appropriate set of parameters is critical.

A recently developed approach based on data mining techniques has been increasingly employed to resolve real-world problems for the past half-decade, particularly in the field of civil engineering [18–28]. Several practical problems have already been effectively performed using machine learning algorithms, paving the way for new prospects in the construction industry. Furthermore, a variety of machine learning algorithms, for example, random forest, artificial neural network (ANN), decision tree, adaptive neuro-fuzzy inference system (ANFIS), AdaBoost, SVM, XGBoost have been developed for addressing technical issues, such as pile mechanical behavior prediction.

Goh et al. [29,30] produced an ANN-based algorithm of piles driven in clays to predict the capacity of friction, using on-field data records to train the algorithm. Furthermore, Shahin et al. [31–34] employed the ANN-based model for forecasting pile load capacity using data that included in situ load testing and cone penetration test (CPT) results. Similarly, Nawari et al. [35] published an ANN approach that uses SPT data and shaft geometry to measure the settling of drilled shafts. Pham et al. [17] produced an ANN and RF to predict driven pile's capacity. Momeni et al. [36] created an ANN model modified with Genetic Algorithm (GA) which select appropriate biases and weights for predicting pile bearing capacity. Based on CPT data, Kordjazi et al. [37] employed an SVM model to forecast the pile ultimate load-bearing capability. Liu et al. [21] developed XGBoost, Backpropagation Neural Network (BPNN) and RF algorithm to estimate driven piles bearing Capacity. Liang et al. estimated stability of hard rock pillars applying XGBoost, gradient boosting decision tree (GBDT), and light gradient boosting machine (LightGBM) Algorithms [23]. Pham et al. [38] has also developed Deep Learning Neural Network to estimate the carrying capacity of piles.

In addition to machine learning (ML) techniques mentioned above, the GBDT method demonstrates excellent results in a variety of disciplines [39–41]. It uses the boosting strategy to incorporate many DTs into a strong classifier as one of the ensemble learning algorithms [42]. DTs belong to the ML approach which employs a tree-like framework to handle a wide range of input types while tracing each path to the prediction outcomes [43]. DTs, on the other hand, are easy to overfit and sensitive to dataset noise because errors of the DTs were offset by one another, the total prediction performance of GBDT improves with the integration of DTs. XGBoost [44] and LightGBM [45] have recently been proposed in the context of GBDT. They have also attracted a lot of attention as a result of their outstanding performances. These three techniques, in particular, operate well with tiny datasets. To some extent, overfitting, which occurs when results match existing data very closely but fail to correctly estimate future trends, can also be prevented [43].

The aim of the present study is to develop a robust model to estimate axial pile bearing capacity using the XGBoost algorithm based on reliable pile load test results. The scope of the present research includes the following:

- To develop a model that is able to learn the complex relationship among axial pile bearing capacity and its influencing factors with reasonable precision.

- To validate the proposed model by comparing the efficacy with prominent modeling techniques, such as AdaBoost, RF, DT, and SVM in terms of performance measure metrics.
- To conduct sensitivity analyses for the determination of the effect of each input parameter on P_u.

The framework of the paper is as follows: In Section 2, data collection and preparation are presented. Section 3 describes the machine learning approaches. The construction of the prediction model is presented in Section 4. Results and discussion are given in Section 5. Lastly, there are some closing remarks.

2. Data Collection and Preparation

2.1. Dataset

In this study, the dataset of 200 reinforced concrete piles at the test site in Ha Nam province–Vietnam (the complete database is available in Table A1) was used to train and test the model. As a first step, all known parameters affecting P_u were taken into account. Furthermore, it was discovered that the majority of traditional methods utilized three categories of parameters: geometry of pile, pile material quality, and soil attributes [3]. To achieve the measurements, hydraulic pile presses were used to drive pre-cast square-section piles with closed tips to the ground at a constant rate of penetration. The testing began at least seven days after the piles were driven, and the experimental setup is shown in Figure 1. The load increased gradually in each pile test, as can be observed. The load might be increased up to 200 percent of the pile load design depending on the design requirements. The time it takes to achieve 100 percent, 150 percent, and 200 percent of the load could take from around 6 to 12 h or 24 h, depending on the load [38]. These two principles were used to determine pile bearing capacity:

(i) the pile bearing capacity was taken as the failure load when the settlement of pile top at the current load level was five times or higher than the settlement of pile top at the previous load level;

(ii) when the load–settlement curve became linear at the last test load, condition (i) would not be used. In such a case, the test load at which progressive movement occurs or the total settlement exceeds 10 % of the pile diameter or width would be taken as the pile bearing capacity.
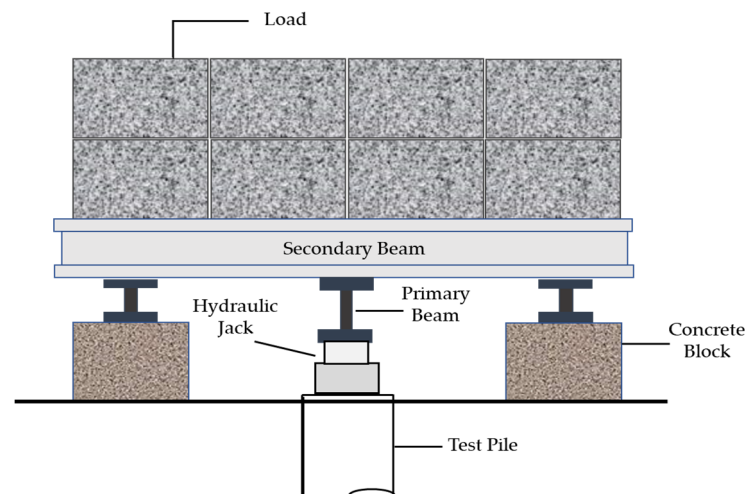
Figure 1. Schematic layout of pile load test.

As a result, previous studies (e.g., [38]) show that pile bearing capacity (P_u) is a function of (1) diameter of the pile (D); (2) depth of the first layer of soil embedded (X_1); (3) depth of the second layer of soil embedded (X_2); (4) depth of the third layer of soil embedded (X_3); (5) pile top elevation (X_p); (6) ground elevation (X_g); (7) extra pile top elevation (X_t); (8) pile tip elevation (X_m); (9) SPT blow count at pile shaft (N_S) and (10) SPT

blow count at pile tip (N_t) as shown in Figure 2. Therefore, in the current study, these input variables were used to develop the proposed models.
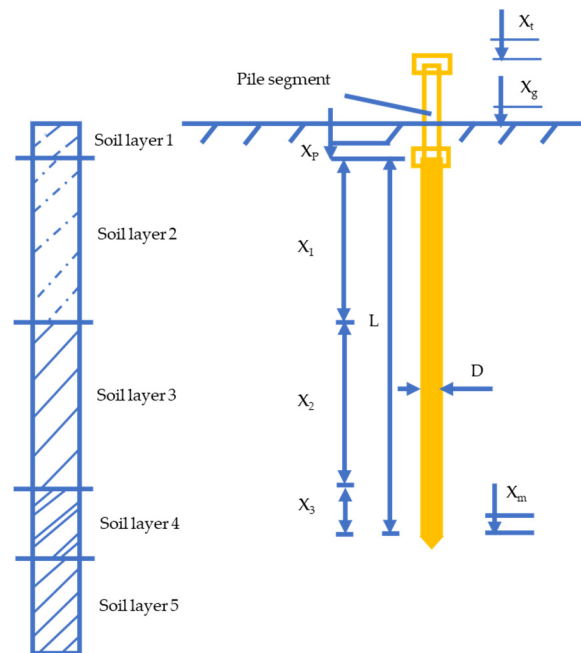
Figure 2. Diagram for stratigraphy and pile parameters [38].

Collected data were divided into training and testing sets, researchers have used a different percentage of the available data as the training set for different problems. For instance, Pham et al. [38] used 60%; Liang et al. [23] used 70%; while Ahmad et al. [28] used 80% of the data for training. The statistical consistency of training and testing datasets has a substantial impact on the results when using soft computing techniques which improves the performance of the model and helps in evaluating them better [22,46]. To choose the most consistent representation, statistical studies of input and output variables of the training and testing data were performed. It was accomplished through the use of a trial-and-error strategy. For training and testing datasets, the best statistically consistent combination was selected. The data division was performed in such a way that 140 (70%) samples were used for training, and 60 (30%) samples were used for testing the models considered in this study. The results of the statistical analysis of the finally selected combinations are shown in Table 1, which includes minimum, mean, maximum and standard deviation of the input and output variables.

Table 1. Statistical study of inputs and output data.

Dataset	Statistical Parameters	Input and Output Parameters										
		D (mm)	X_1 (m)	X_2 (m)	X_3 (m)	X_p (m)	X_g (m)	X_t (m)	X_m (m)	N_s	N_t	*Pu* (kN)
Training	Minimum	300	3.4	1.8	0	1.95	3.32	2	8.6	5.9	4.64	432
	Average	378.57	4.002	6.43	0.377	2.615	3.517	2.834	13.425	10.811	6.908	1064.739
	Maximum	400	4.75	8	1.18	3.4	3.7	4.45	15.58	13.63	7.69	1551
	Standard Deviation	41.179	0.455	2.039	0.467	0.552	0.069	0.609	2.207	2.550	0.914	363.681
Testing	Minimum	300	3.4	2.08	0	2.05	3.38	2.05	8.88	6.18	4.86	407.2
	Average	371.667	3.85	6.774	0.307	2.77	3.522	2.987	13.702	10.932	7.087	1023.266
	Maximum	400	4.75	8	1.18	3.4	3.72	4.05	15.58	13.53	7.73	1551
	Standard Deviation	45.442	0.472	1.594	0.438	0.585	0.078	0.549	1.719	2.123	0.636	362.003

2.2. Correlation Analysis

Correlation (ρ) was used to verify the intensity of correlation between different parameters (see Table 2). Given pair of random variables (m, n), the following equation for ρ is used:

$$\rho(m,\ n) = \frac{cov(m,\ n)}{\sigma_m \sigma_n} \tag{1}$$

where *cov* denotes covariance, σ_m denotes the standard deviation of m, and σ_n denotes the standard deviation of n. $|\rho| > 0.8$ represents a strong relationship among m and n, values between 0.3 and 0.8 represents medium relationship, while $|\rho| < 0.30$ represents weak relationship [47]. According to Song et al. [48], correlation is considered as "strong" if $|\rho| > 0.8$. Table 2 displays the correlations between input and output characteristics. The correlation coefficient has a maximum absolute value of 0.989, as shown in Table 2. There is a "strong to weak" relationship among various variable combinations so none of the input variables was removed.

Table 2. Correlation between parameters.

Parameters	D	X_1	X_2	X_3	X_p	X_g	X_t	X_m	N_s	N_t	P_u
D	1.000										
X_1	0.641	1.000									
X_2	0.462	0.329	1.000								
X_3	0.421	0.448	0.564	1.000							
X_p	−0.714	−0.935	−0.515	−0.672	1.000						
X_g	0.436	0.357	0.333	0.203	−0.377	1.000					
X_t	−0.481	−0.469	−0.331	−0.810	0.628	−0.135	1.000				
X_m	0.474	0.378	0.989	0.672	−0.571	0.334	−0.422	1.000			
N_s	0.577	0.572	0.947	0.719	−0.732	0.371	−0.533	0.969	1.000		
N_t	0.197	0.050	0.923	0.619	−0.289	0.198	−0.303	0.931	0.827	1.000	
P_u	0.735	0.706	0.785	0.474	−0.780	0.460	−0.336	0.785	0.846	0.558	1.000

3. Machine Learning Methods

3.1. Extreme Gradient Boosting Algorithm

Chen and Guestrin [44] suggested the XGBoost algorithm, which is based on the GBDT structure. It has attracted a lot of attention as a result of its outstanding results in Kaggle's ML competitions [49]. Unlike GBDT, the XGBoost goal function includes a regularization term to avoid overfitting. The main objective function is described as follows:

$$O = \sum_{i=1}^{n} L(y_i, F(x_i)) + \sum_{k=1}^{t} R(f_k) + C \tag{2}$$

where $R(f_k)$ represents the regularization term at iteration k, and C being a constant that can be removed selectively.

Regularization term $R(f_k)$ written as,

$$R(f_k) = \alpha H + \frac{1}{2}\eta \sum_{j=1}^{H} w_j^2 \tag{3}$$

where α is the complexity of leaves, H denotes the number of leaves, η signifies penalty variable, and ω_j represents output results in each leaf node. Leaves denote the expected categories based on classification criteria, whereas the leaf node denotes the tree node which cannot be divided.

Furthermore, unlike GBDT, XGBoost employs a second-order Taylor series of main functions rather than the first-order derivative. If the loss function is the mean square error (*MSE*), then the main function may be written as:

$$O = \sum_{i=1}^{n} \left[p_i \omega_{q(x_i)} + \frac{1}{2}\left(q_i \omega_{q(x_i)}^2\right) \right] + \alpha H + \frac{1}{2}\eta \sum_{j=1}^{H} \omega_j^2 \tag{4}$$

where $q(x_i)$ is a function that maps data points to leaves, g_i and h_i represents loss function's first and second derivatives, respectively.

The final loss value is calculated by adding all of the loss values together. Because samples in the DT corresponds to nodes of leaf, the ultimate loss value can be calculated by adding loss values of the leaf nodes. As a result, the main function can be written as:

$$O = \sum_{j=1}^{T} \left[P_j \omega_j + \frac{1}{2} \left(Q_j + \eta \right) \omega_j^2 \right] + \alpha H \tag{5}$$

where $P_j = \sum_{i \epsilon I_j} p_i$, $Q_j = \sum_{i \epsilon I_j} q_i$, and I_j are the total number of samples in leaf node j.

To summarize, the challenge of optimizing the main function is reduced to identifying the minimum of a quadratic function. Due to the addition of regularization phenomena, XGBoost has a stronger capability to avoid overfitting. The structure of XGBoost can be seen in Figure 3.
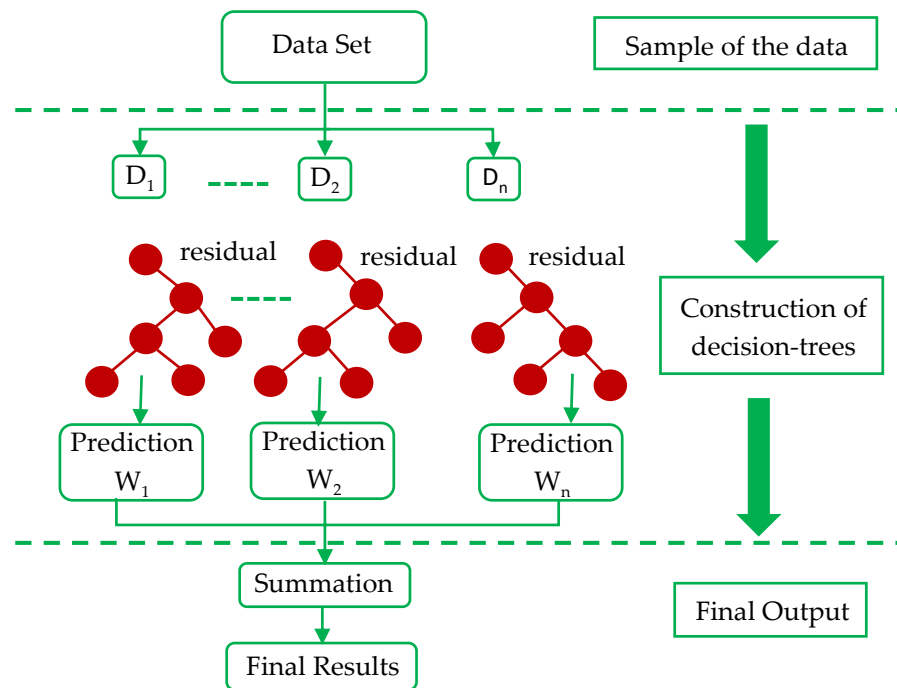
Figure 3. Structure of XGBoost Algorithm.

3.2. Random Forest (RF) Algorithm

Because of its simplicity and diversity, RF is the most applied ML method. Breiman in 2001, developed this supervised learning approach for classification and regression analysis [50]. RF is an integrated learning strategy that collects data from a single DT and improves prediction accuracy by using majority voting or mean findings, depending on the task.

Assume you have an input data set with the following values Q = q_1, q_2, q_3, . . . , q_n where n is the number of datasets. An RF model would be a set of T trees $T_1(Q)$, $T_2(Q)$, $T_3(Q)$ . . . , $T_n(Q)$. $\hat{R}_1, \hat{R}_2 \ldots \ldots \hat{R}_n$ is the predicted outcome of these decision-making trees. The eventual output of the RF model for the regression problem will be the average of all the above trees' prediction outcomes. The concept of splitting initial training sets into smaller sets, with only a few predictive elements picked at random in each split, has been used to construct tree-growing algorithms. Because the programmer fails to prune decision trees according to predetermined stopping criteria, they continue to grow indefinitely. Tree growth stops such as the Gini Diversity Index, *RMSE* and *MSE* are frequently utilized. Trees with appropriate predictions are picked in the final RF model, and trees with low predictive outcomes are excluded. The overfitting problem of the single DT model is eliminated by

randomly selecting predictor parameters and the final set of DTs [50,51]. Figure 4 illustrates the random forest's structure.
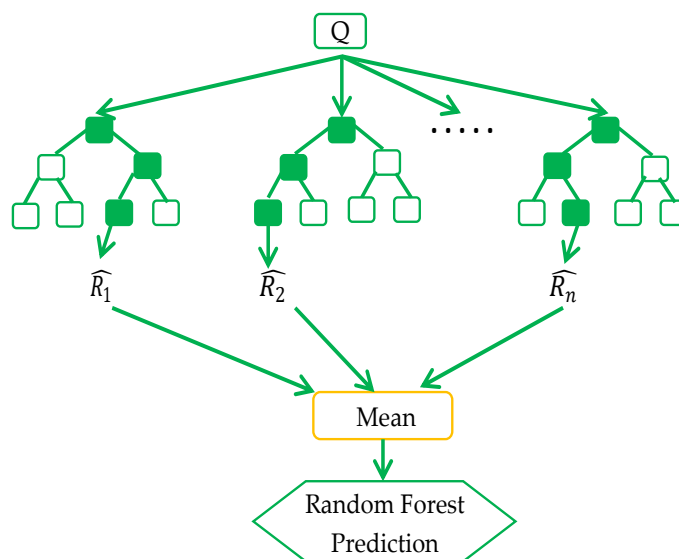

Figure 4. Random Forest structure.

3.3. AdaBoost Algorithm

AdaBoost or adaptive boosting is a sequential ensemble technique which is based on the principle of developing several weak learners using different training sub-sets drawn randomly from the original training dataset [52,53]. During each training, weights are assigned which are used when learning each hypothesis. The weights are used for computation of the error of the hypothesis on the dataset and are an indicator of the comparative importance of each instance. The weights are recalculated after every iteration, such that incorrectly classified instances by the last hypothesis receive higher weights. This enables the algorithm to focus on more difficult-to-learn instances. Assigning revised weights to the incorrectly classified instances is the most vital task of the algorithm. Unlike in classification, in regression, the instances are not correct or incorrect, rather they constitute a real-value error. By comparing the computed error to a predefined threshold prediction error, it can be labeled as an error or not an error and thus, the AdaBoost classifier can be used. Instances with larger errors on previous learners are more likely (i.e., higher probability) to be selected for training the subsequent base learner. Finally, weighted average or median is used to provide an ensemble prediction of the individual base learner predictions [54].

3.4. Support Vector Machine (SVM) Algorithm

Vapnik invented the SVM in 1995 [55], and it is a popular and successful learning algorithm for the classification of linear and nonlinear regression problems. The SVM algorithm delivers reliable prediction outcomes and is practicable for high-dimensional feature spaces, is robust and has good noise resistance [56,57]. In many disciplines, many effective SVM implementations with classification and regression issues have been documented [58–60]. The following is a summary of SVM's basic theory.

As illustrated in Figure 5, a training set $\{(u_k, v_k), k = 1, 2, \ldots \ldots, n\}$ is chosen for an SVM model, where $u_k = [u_{1k}, u_{2k}, \ldots \ldots, u_{nk}] \in R^{n_h}$ is the input data, $v_k \in R^{n_m}$ is the output data corresponding to u_k, and n is the number of training samples. The goal of the SVM is to identify an optimal hyperplane function $f(x)$ (defined by the weight vector w and the offset b), that passes through all data items with the insensitive loss coefficient ε (based on two supporting hyperplanes, w.u $- b = \varepsilon$ and w.u $- b = -\varepsilon$).
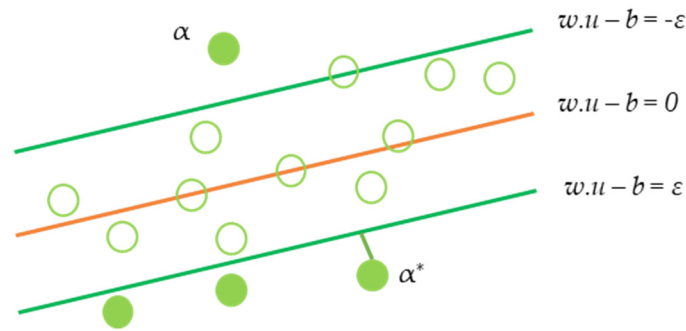
Figure 5. SVM for a regression problem.

The function *f(u)* in nonlinear regression is determined as follows:

$$f(u) = \sum \left({}_{i=1}^{n} (\alpha_i - \alpha_i^*) K(u_i, u_j) + b \right) \tag{6}$$

with

$$\sum_{i=1}^{n} (\alpha_i - \alpha_i^*) = 0, \ \ C \geq \alpha_i, \ \alpha_i^* \geq 0, \forall i \tag{7}$$

The penalty constant C is used to manage the penalty error, α_i, α_i^* are the Lagrange multipliers, and $K(u_i, u_j)$ is the kernel function defined as follows:

$$K(u_i, u_j) = < \Phi(u_i), \Phi(u_j) > \tag{8}$$

The mapping function F is a nonlinear mapping function. The most often used kernel functions are linear, polynomial, sigmoid, and Gaussian functions:

Linear kernel function:

$$K(u_i, u_j) = u_i \cdot u_j \tag{9}$$

Polynomial kernel function:

$$K(u_i, u_j) = (\gamma u_i \cdot u_j + c)^d \tag{10}$$

Sigmoid kernel function:

$$K(u_i, u_j) = tanh \, (\gamma u_i \cdot u_j + c)^d \tag{11}$$

Gaussian kernel function:

$$K(u_i, u_j) = exp \left(-\gamma (u_i - u_j)^2 \right)^d \tag{12}$$

3.5. Decision Tree (DT) Algorithm

A decision tree is a tool with a tree-like structure that predicts likely outcomes, resource costs, utility costs, and potential consequences. One of the benefits of the machine learning approach over traditional statistical approaches such as regression is that they can handle more than two-dimensional data. For data-driven prediction analysis of diverse geotechnical problems, many researchers have adopted the tree-based approach [20,61,62]. As a result, tree-based ML techniques, such as DT, were used to build models and identify the key predictors of pile–soil friction in this work. DT can be seen graphically, showing specific decision requirements as well as the complicated branching that occurs in a constructed decision. This is one of the most popular and commonly used supervised learning techniques for forecasting model accuracy.

DT is capable of performing all tasks including recognition, classification, and prediction. DT is a "tree"-shaped structure made up of a succession of questions, each of which is described by a set of parameters. Roots, branches, and leaves comprise a real tree. Similarly, the graph for DT is comprised of nodes, which are leaves, and branches, which represent

connections between nodes [63]. A variable is chosen as a root, also known as the initial node, during the DT process. By reference to the appointed features, the initial node is divided into many internal nodes. DT is a top-down tree, meaning the roots are at the very top. Roots, branches, and nodes are the end products of the branches [64]. Each node can be divided into two branches and each node has a relationship to a specific characteristic and branches that have been specified by a specific range of input. Figure 6 depicts a flowchart linked to the DT approach.
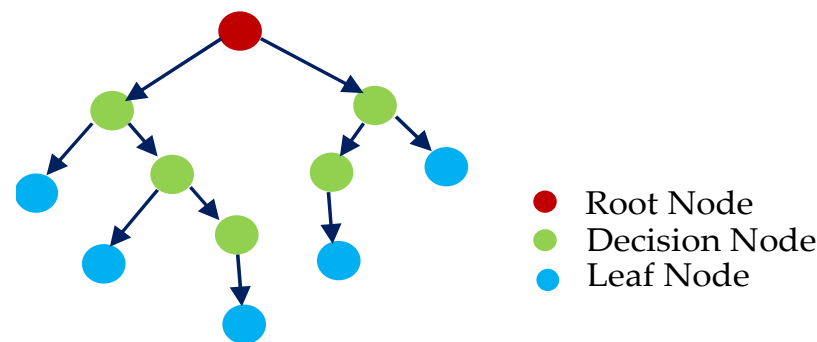
Figure 6. Decision tree structure.

4. Construction of Prediction Models

Orange software was used to create the proposed models for predicting pile bearing capacity. Orange is an open-source software package. Machine learning, preprocessing, and visualization methods are included in the default installation, which is divided into six widget sets i.e., Data, Visualize, Classify, Regression, Evaluate and Unsupervised. Orange is visual programming software for machine learning, visualization, data mining, data analysis.

The predictor variables were provided via an input set (x) defined by x = {D, X_1, X_2, X_3, X_p, X_g, X_t, X_m, N_S and N_t}, while the target variable (y) is P_u. The most important task in every modeling step is to pick the right number of training and testing datasets. As a result, 70% of the whole data was chosen to generate the models in this study, with the developed models being tested on the remaining data. On the other way, 140 and 60 sets were utilized for creating and testing the models, respectively. All models (XGBoost, AdaBoost, RF, DT, and SVM) were tweaked to optimize the P_u prediction using the trial-and-error process. Figure 7 shows how the prediction models were built.

4.1. Hyperparameter Optimization

ML algorithms have parameters that must be tuned. The optimization procedure seeks to find ideal settings for XGBoost, AdaBoost, RF, DT, and SVM to achieve accurate prediction. This study tunes various critical parameters in the XGBoost, AdaBoost, RF, DT and SVM, as well as clarifies the definitions of these hyperparameters. The tuning parameters for the models were chosen and then changed in the trials until the best metrics shown in Table 3 were achieved.
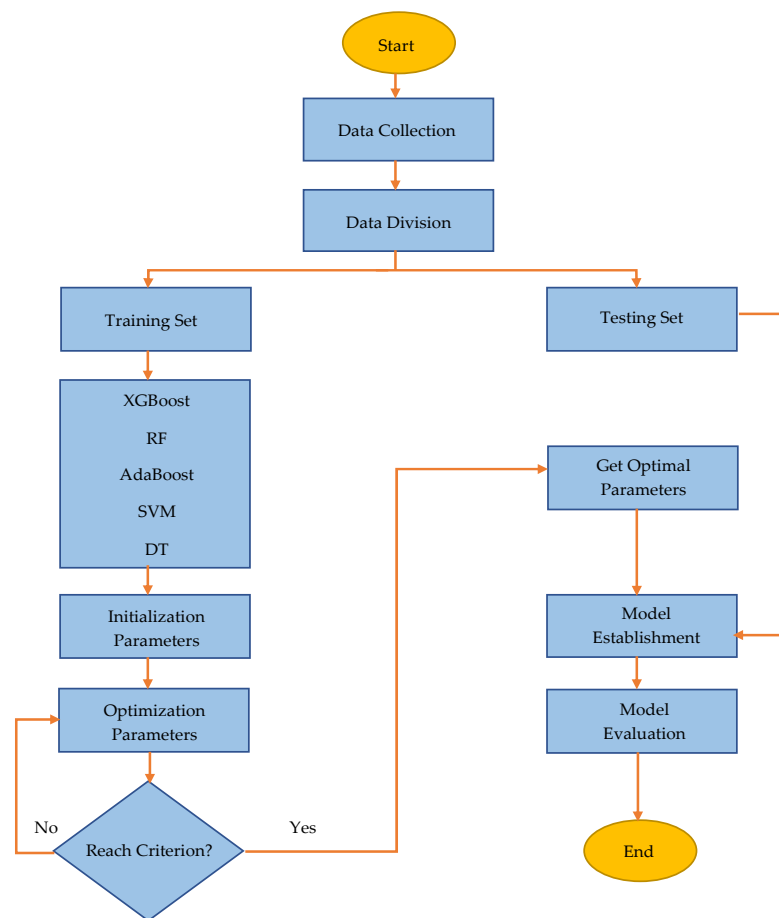
Figure 7. The flowchart for applying a data-driven technique to anticipate pile bearing capacity.

Table 3. Hyperparameters optimal results.

Algorithm	Hyperparameters	Meanings	Optimal Values
XGBoost	n estimators	Number of trees	133
	Learning rate	Shrinkage coefficient of tree	0.03
	Maximum depth	Maximum depth of a tree	4
RF	n estimators	Number of trees in forest	500
	Minimum split	Minimum samples of split for nodes	5
	Maximum depth	Maximum depth of a tree	5
	Minimum leaf	Minimum samples of nodes for leaf	8
AdaBoost	n estimators	Number of trees	500
	Learning rate	Shrinkage coefficient of tree	1
SVM	C_2	Regularization parameter	2.5
DT	Minimum split	Minimum samples of split for nodes	4
	Maximum depth	Maximum depth of a tree	100
	Minimum leaf	Minimum samples of nodes for leaf	7

4.2. Model Evaluation Indexes

The results of the proposed models are evaluated using *R²*, *MAE*, *RMSE*, *MARE*, *NSE* and *RSR*, as more commonly used criteria in the literature. The following equations are used to calculate these metrics:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{\sum_{i=1}^{n}(x_i - \hat{x})^2} \tag{13}$$

$$MAE = \frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i) \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - \hat{x}_i)^2} \tag{15}$$

$$MARE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{x_i - \hat{x}_i}{x_i} \times 100 \right| \tag{16}$$

$$NSE = 1 - \frac{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}{\sum_{i=1}^{n} (x_i - \overline{x})^2} \tag{17}$$

$$RSR = \frac{\sqrt{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}}{\sqrt{\sum_{i=1}^{n} (x_i - \overline{x})^2}} \tag{18}$$

where n denotes the number of points, x_i and $\hat{x}_i$ denotes the actual and expected outputs of i-th sample, respectively; $\overline{x}$ is data averaged actual output. R^2 is a number that ranges from 0 to 1, a higher R^2 value indicates a more efficient model. The model is considered effective when R^2 is more than 0.8 and close to 1 [22]. The mean squared difference between projected outputs and targets is the criterion *RMSE*, and the mean magnitude of errors is the criterion *MAE*, *RMSE* and *MAE* are similar in that the closer these criterion values of these errors are to 0, the better the model's performance. In circumstances where the *MAE* and *RMSE* are minimal, the model's accuracy is greater. Models yielded the lowest *MARE* value, indicating that the model has superior predictive power. The *RSR* ranges from 0 to a considerable positive number. Lower *RSR* indicates lower *RMSE*, indicating that the model is more productive. *RSR* and *NSE* categorization ranges as very good, good, satisfactory, and unsatisfactory with ranges $0 \leq RSR \leq 0.5$, $0.5 < RSR \leq 0.6$, $0.6 < RSR \leq 0.7$, $RSR > 0.7$ and $0.75 < NSE \leq 1$, $0.65 < NSE \leq 0.75$, $0.5 < NSE \leq 0.65$, and $NSE \leq 0.5$, respectively [65].

In addition, the Taylor diagram was used to compare the model's performance visually [66]. Taylor diagram shows how similar patterns are and how closely a model pattern relates to reference. The standard deviation (σ), R^2, and the *RMSE* are three equivalent model performance statistics that can be shown on a two-dimensional plot using the law of cosines. Taylor diagram is the best method for comparing the performance of various models in particular.

5. Result and Discussion

5.1. Comparison of Models

This section evaluates the model's efficacy. Figures 8 and 9 depict the training and testing dataset's prediction performance in regression form, respectively, while Tables 4 and 5 provide a summary of the relevant data.

Table 4. Summary of Training model.

	Training Set					
Model	R^2	**MAE (kN)**	**RMSE (kN)**	**MARE (%)**	**NSE**	**RSR**
XGBoost	0.971	47.518	66.844	4.355	0.966	0.184
AdaBoost	0.957	56.671	82.495	5.252	0.948	0.228
RF	0.952	58.366	79.240	5.739	0.952	0.219
DT	0.932	68.912	94.304	6.911	0.932	0.260
SVM	0.887	88.801	123.375	8.507	0.884	0.340
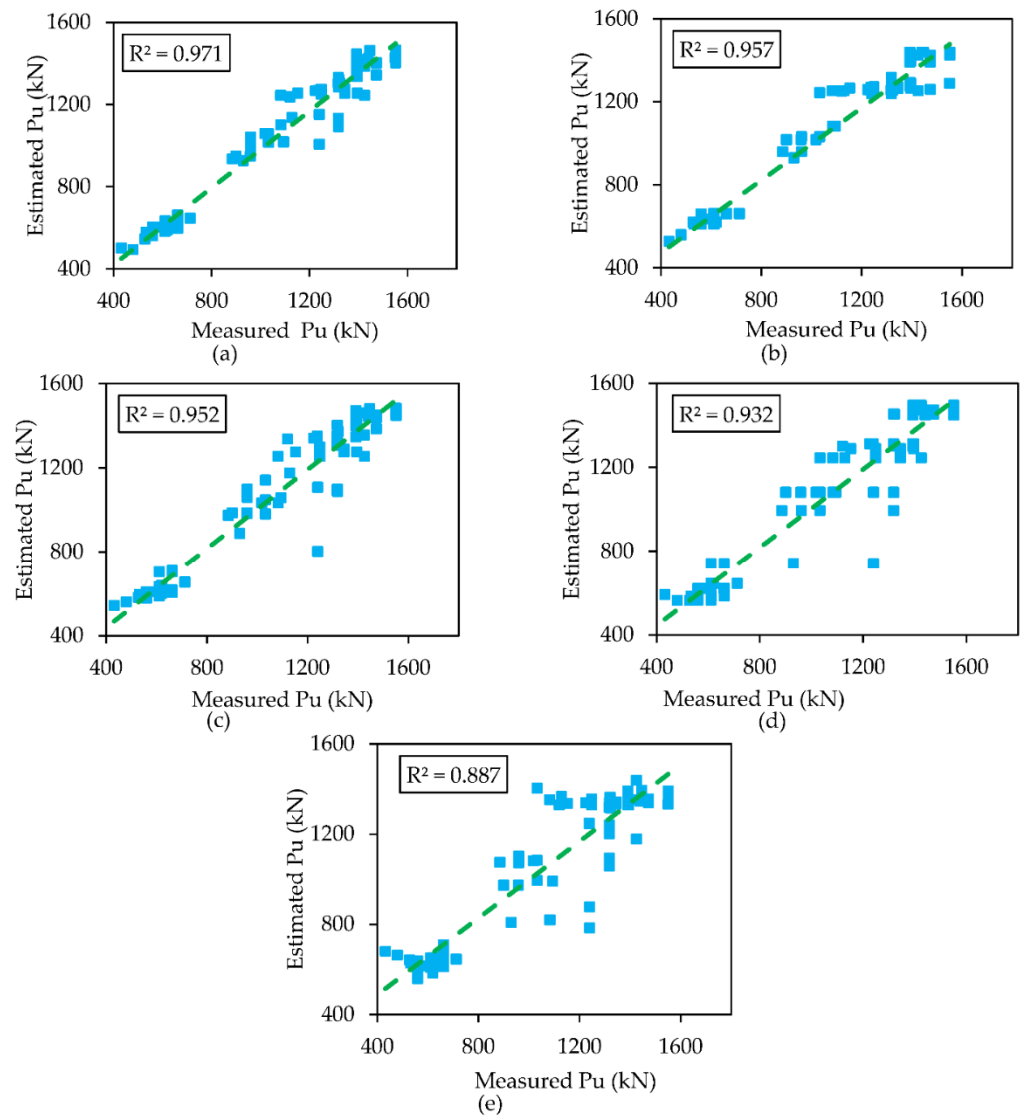
Figure 8. Measured Pu versus Estimated Pu for training models using (**a**) XGBoost, (**b**) AdaBoost, (**c**) RF, (**d**) DT, and (**e**) SVM.

Table 5. Summary of Testing model.

			Testing Set			
Model	R^2	*MAE* (kN)	*RMSE* (kN)	*MARE* (%)	*NSE*	*RSR*
XGBoost	0.955	59.929	80.653	6.600	0.950	0.225
AdaBoost	0.950	70.383	90.665	8.252	0.936	0.253
RF	0.945	69.030	86.348	8.014	0.942	0.241
DT	0.925	74.450	99.822	8.775	0.923	0.278
SVM	0.878	98.320	128.027	10.991	0.873	0.357

In terms of training, the XGBoost model produced the best prediction results (i.e., R^2 = 0.971, *MAE* = 47.518 and *RMSE* = 66.844) compared to AdaBoost (i.e., R^2 = 0.957, *MAE* = 56.671 and *RMSE* = 82.495), RF (i.e., R^2 = 0.952, *MAE* = 58.366 and *RMSE* = 79.240), DT (i.e., R^2 = 0.932, *MAE* = 68.912 and *RMSE* = 94.304) and SVM (i.e., R^2 = 0.887, *MAE* = 88.801 and *RMSE* = 123.375). This is also confirmed by the results of *MARE, NSE* and *RSR* in

Table 4. In training part, XGBoost produced lesser *MARE, NSE* and *RSR* values compared to AdaBoost, RF, DT and SVM.
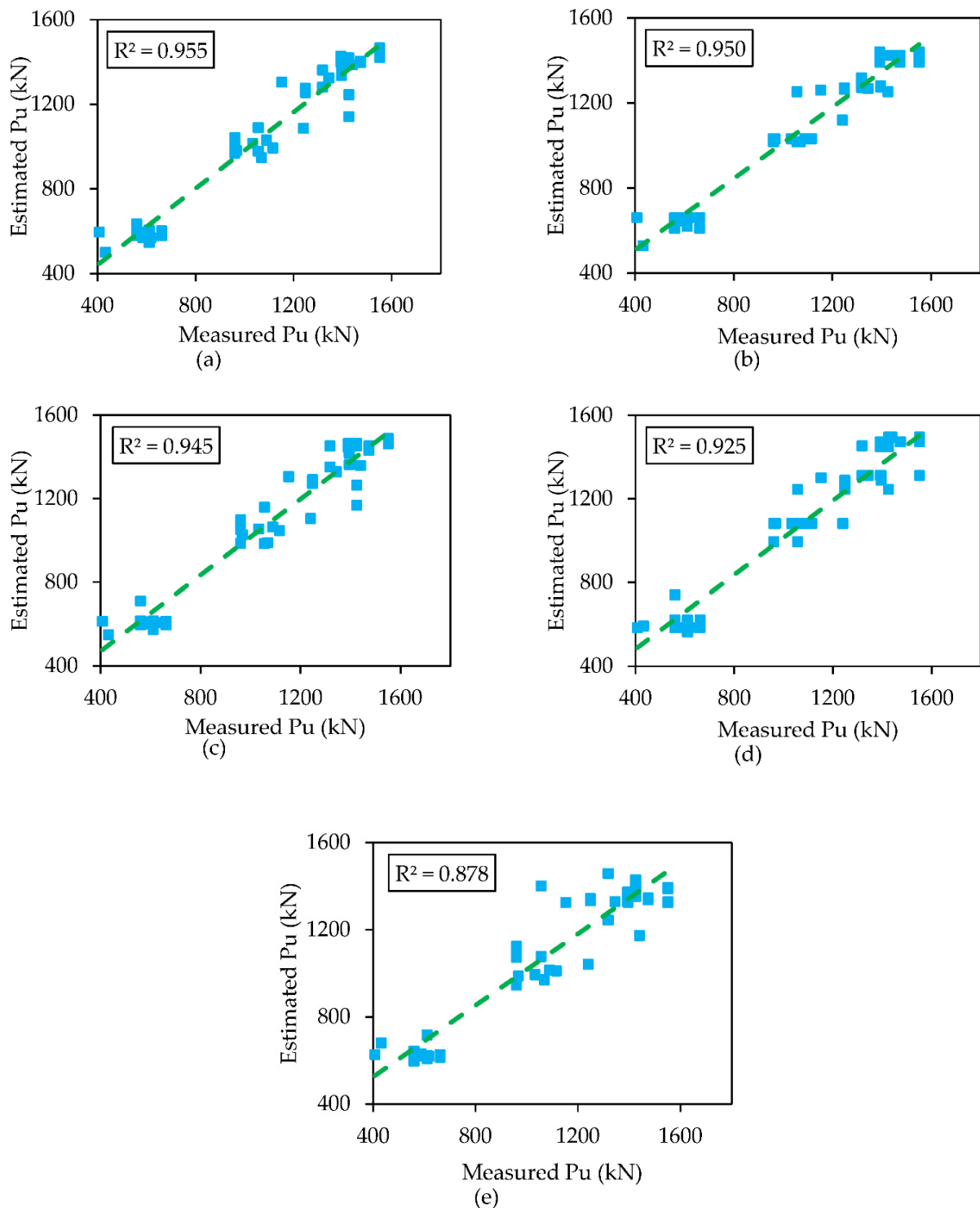
Figure 9. Measured Pu versus predicted Pu for testing models using (**a**) XGBoost, (**b**) AdaBoost, (**c**) RF, (**d**) DT, and (**e**) SVM.

In the testing part, the XGBoost model had the best prediction results with respect to R^2, *MAE, RMSE, MARE, NSE* and *RSR* (i.e., $R^2 = 0.955$, *MAE* = 59.929, *RMSE* = 80.653, *MARE* = 6.6, *NSE* = 0.950, and *RSR* = 0.225) compared to AdaBoost (i.e., $R^2 = 0.950$, *MAE* = 70.383, *RMSE* = 90.665, *MARE* = 8.252, *NSE* = 0.936, and *RSR* = 0.253), RF (i.e.,

$R^2 = 0.945$, $MAE = 69.030$, $RMSE = 86.348$, $MARE = 8.014$, $NSE = 0.942$, and $RSR = 0.241$), DT (i.e., $R^2 = 0.0.925$, $MAE = 74.450$, $RMSE = 99.822$, $MARE = 8.775$, $NSE = 0.923$, and $RSR = 0.278$) and SVM (i.e., $R^2 = 0.878$, $MAE = 98.320$, $RMSE = 128.027$, $MARE = 10.991$, $NSE = 0.873$, and $RSR = 0.357$) as shown in Table 5.

Comparing the above performance measures the proposed XGBoost model performed better than the AdaBoost, RF, DT and SVM. From these statistical analysis and prediction capabilities, we can state that the XGBoost model has good accuracy prediction for pile bearing capacity.

The sensitivity results of the XGBoost model were assessed using Yang and Zang's [67] method for assessing the impact of input variables on P_u. This approach, which has been used in several investigations [22,28,68–70], is as follows:

$$r_{ij} = \frac{\sum_{k=1}^{n} (x_{im} \times x_{om})}{\sqrt{\sum_{k=1}^{n} x_{im}^2 \sum_{k=1}^{n} x_{om}^2}} \tag{19}$$

as n represents the number of values (i.e., 140); x_{im} and x_{om} denotes input and output variables, respectively. For each input parameter, the r_{ij} value ranges from zero to one, with the greatest r_{ij} values indicating the efficient output variable (i.e., Pu).

Figure 10 shows the r_{ij} scores for all input variables. Figure 10 demonstrates that SPT blow count at pile shaft (N_S) ($r_{ij} = 0.985$) has the greatest effect on the Pu.
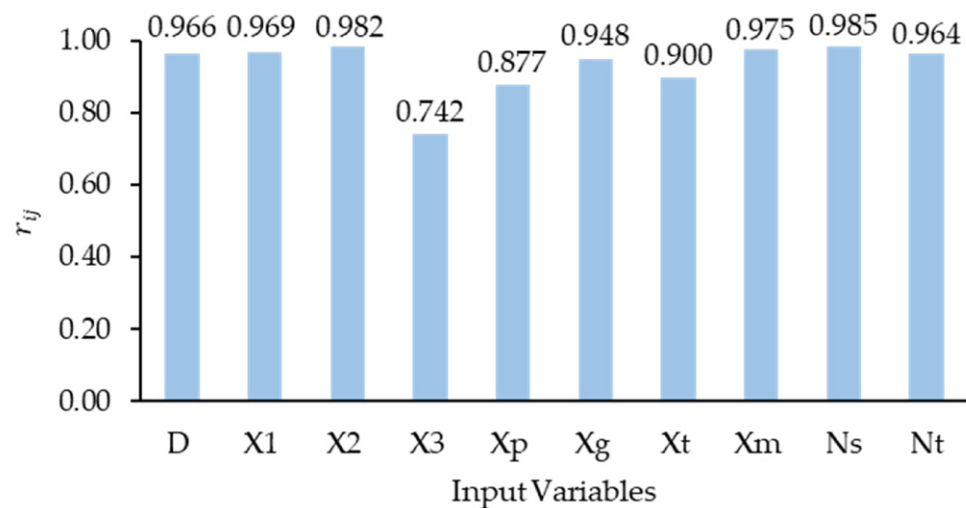
Figure 10. Sensitivity analysis of input variables.

With the use of the Taylor diagram (see Figure 11), we investigated the model's efficiency further. The better the performance, the closer each produced model's point is to the observed point location. The models demonstrated the best predictive capability, while the XGBoost method had a greater correlation and a lesser *RMSE*.

5.2. Comparison with Other Researchers

Table 6 shows some findings from a study on machine learning applications on pile bearing capacity. On the test data set, the expected efficiency of ML algorithms in foundation engineering having predictive outcomes of foundation load is mostly ranging R^2 from 0.71 to 0.918, according to the results of previous studies while in the present study it is 0.955. However, due to the use of different datasets, a comparison between these results is unwarranted. A project that uses different datasets is needed to give a generalized model to foundation engineering.
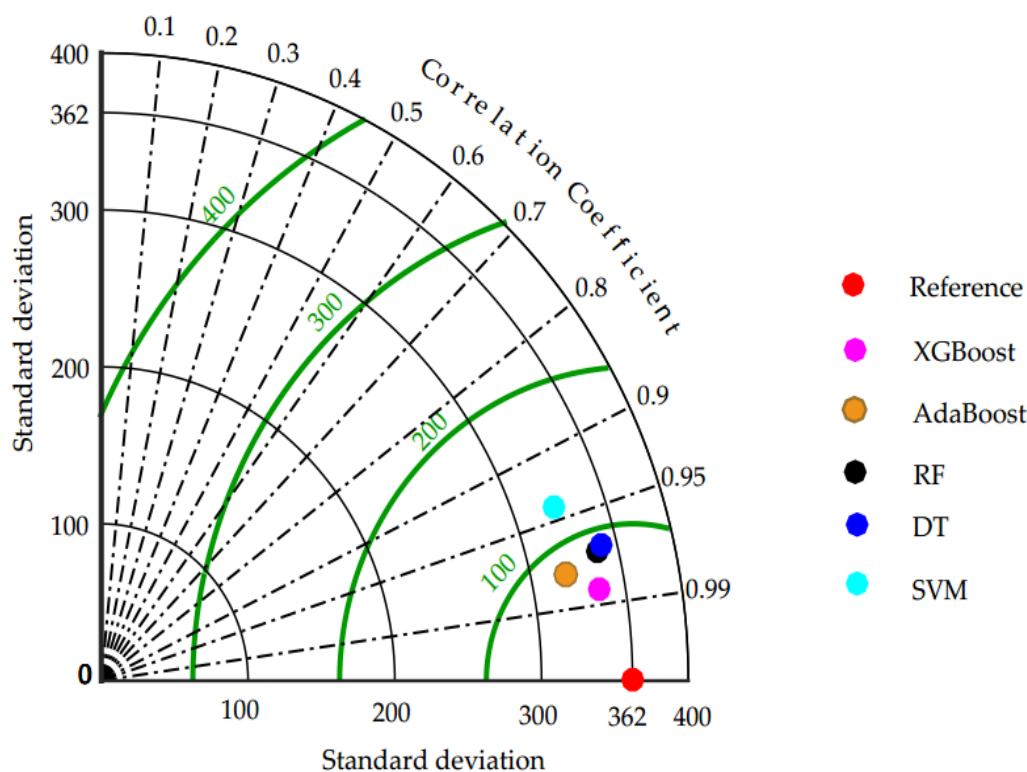
Figure 11. Taylor diagram of the models.

Table 6. Comparison with other studies.

Author	Model	Foundation Type	Number of Samples	R^2	RMSE
Momeni et al. [71]	ANFIS	Thin-walls	150	0.875	0.048
	ANN			0.71	0.529
Momeni et al. [72]	GPR	Piles	296	0.84	-
Kulkarni et al. [73]	GA-ANN			0.86	0.0093
Armaghani et al. [74]	ANN	Rock-socketed piles	132	0.808	0.135
	PSO-ANN			0.918	0.063
Pham et al. [38]	GA-DLNN	Piles	472	0.882	109.965
Present study	XGBoost	Piles	200	0.955	80.653

6. Conclusions

Pile bearing capacity values were estimated in this paper using five models. The prediction model was built with ten input parameters and one output parameter. The modeling results show that the XGBoost model has the best capability for accurate prediction of *Pu* when compared to other models such as AdaBoost, RF, DT and SVM. The following are some of the major findings of this study:

1. In testing phase, the XGBoost model (R^2 = 0.955, *MAE* = 59.929, *RMSE* = 80.653, *MARE* = 6.6, *NSE* = 0.950, and *RSR* = 0.225) has the highest performance capability as compared to other soft computing techniques considered in this study i.e., AdaBoost, RF, DT and SVM as well as the models used in the literature.
2. Sensitivity analysis results show that SPT blow count at pile shaft (N_S) was the most important parameter in predicting pile bearing capacity.
3. Taylor diagram also verified that all the models are good but the predictive power of the XGBoost algorithm had a higher correlation and lower *RMSE*.
4. Based on the results and analysis the XGBoost model can also be applied to solve a variety of geotechnical engineering problems.

Furthermore, the XGBoost technique has the advantage of being easily updated, it is obvious that the proposed model is open to further development, and that the collection of more data will result in significantly stronger prediction capability, avoiding the requirement for expertise and time to update an existing design aid or equation.

Abbreviation

Symbol	Explanation
P_u	Pile bearing capacity
ML	Machine learning
XGBoost	Extreme gradient boosting
AdaBoost	Adaptive boosting
RF	Random forest
DT	Decision tree
SVM	Support vector machine
ANN	Artificial neural network
ANFIS	adaptive neuro-fuzzy inference system
GA	Genetic algorithm
BPNN	Backpropagation neural network
GBDT	Gradient boosting decision tree
LightGBM	Light gradient boosting machine
DLNN	Deep Learning Neural Network
PSO-ANN	Particle swarm optimization—ANN
GPR	Gaussian process regression
R^2	Coefficient of determination
MAE	Mean absolute error
MSE	Mean square error
$RMSE$	Root mean square error
$MARE$	Mean absolute relative error
NSE	Nash–Sutcliffe model efficiency
RSR	Relative strength ratio
SPT	Standard penetration test
CPT	Cone penetration test
D	Diameter
X_1	Depth of first layer of soil embedded
X_2	Depth of second layer of soil embedded
X_3	Depth of third layer of soil embedded
X_p	Pile top elevation
X_g	Ground elevation
X_t	Extra pile top elevation
X_m	Pile tip elevation
N_S	SPT blow count at pile shaft
N_t	SPT blow count at pile tip

Appendix A

Table A1. Data Catalog.

S. No.	D	X_1	X_2	X_3	X_p	X_g	X_t	X_m	N_s	N_t	*Pu*
Unit	mm	m	m	m	m	m	m	m	-	-	kN
1	400	3.45	8	0.3	2.95	3.65	2.95	14.7	11.75	7.59	1017.9
2	400	4.25	8	1	2.15	3.56	2.16	15.4	13.25	7.67	1152
3	400	4.25	8	1.02	2.15	3.58	2.16	15.42	13.27	7.68	1344
4	400	4.25	8	0.1	2.15	3.58	3.08	14.5	12.35	7.14	1551
5	400	4.35	8	1.06	2.05	3.55	2.09	15.46	13.41	7.66	1321
6	300	3.4	5.25	0	3.4	3.49	3.44	12.05	8.65	6.75	559.8
7	400	4.25	8	1.02	2.15	3.58	2.16	15.42	13.27	7.68	1248
8	300	3.4	5.18	0	3.4	3.36	3.38	11.98	8.58	6.73	559.8
9	400	4.75	7.25	0	2.05	3.62	3.57	14.05	12	6.73	1425
10	300	3.4	5.25	0	3.4	3.47	3.42	12.05	8.65	6.75	559.8
11	300	3.4	5.2	0	3.4	3.42	3.42	12	8.6	6.73	660.6
12	400	3.45	5.24	0	3.35	3.44	3.4	12.04	8.69	6.72	1240
13	400	4.35	8	1.07	2.05	3.52	2.05	15.47	13.42	7.67	1425
14	400	4.1	2.17	0	2.7	3.7	2.73	8.97	6.27	4.92	661.6
15	400	3.55	5.39	0	3.25	3.44	3.25	12.19	8.94	6.72	1083
16	400	4.25	8	1	2.15	3.56	2.16	15.4	13.25	7.67	1152
17	400	3.4	7.3	0	3.4	3.61	3.51	14.1	10.7	7.28	1115.2
18	300	3.4	5.2	0	3.4	3.43	3.43	12	8.6	6.73	610.7
19	300	3.4	5.2	0	3.4	3.42	3.42	12	8.6	6.73	661.6
20	400	4.1	1.8	0	2.7	3.39	2.79	8.6	5.9	4.64	620
21	400	3.45	8	0.3	2.95	3.66	2.96	14.7	11.75	7.59	960
22	300	3.4	5.27	0	3.4	3.49	3.42	12.07	8.67	6.75	559.8
23	400	4.25	8	1	2.15	3.56	2.16	15.4	13.25	7.67	1248
24	400	4.65	7.4	0	2.15	3.59	3.39	14.2	12.05	6.80	1551
25	400	4.1	2	0	2.7	3.56	2.76	8.8	6.1	4.80	620
26	400	4.35	8	0.3	2.05	3.45	2.75	14.7	12.65	7.22	1473
27	400	4.35	8	1.03	2.05	3.48	2.05	15.43	13.38	7.65	1318
28	400	4.35	8	1.01	2.05	3.46	2.05	15.41	13.36	7.64	1473
29	400	4.1	1.72	0	2.7	3.27	2.75	8.52	5.82	4.57	423.9
30	400	3.4	7.28	0	3.4	3.48	3.4	14.08	10.68	7.27	1318
31	400	4.35	8	1.05	2.05	3.55	2.1	15.45	13.4	7.66	1221.5
32	300	3.4	5.2	0	3.4	3.43	3.43	12	8.6	6.73	559.8
33	400	4.25	8	0.96	2.15	3.53	2.17	15.36	13.21	7.65	1344
34	400	4.65	7.35	0	2.15	3.55	3.4	14.15	12	6.79	1392
35	400	3.85	7.5	0	2.95	3.68	3.38	14.3	11.35	7.13	1425
36	300	3.4	5.35	0	3.4	3.57	3.42	12.15	8.75	6.78	661.6
37	400	4.75	7.5	0	2.05	3.6	3.3	14.3	12.25	6.79	1425
38	400	4.35	8	0.95	2.05	3.41	2.06	15.35	13.3	7.60	1323.2

Table A1. *Cont.*

S. No.	D	X_1	X_2	X_3	X_p	X_g	X_t	X_m	N_s	N_t	*Pu*
39	400	4.25	8	0.9	2.15	3.57	2.27	15.3	13.15	7.61	1473
40	400	4.35	8	0.96	2.05	3.42	2.06	15.36	13.31	7.61	1244
41	400	4.35	8	1.05	2.05	3.5	4.35	15.45	13.4	7.66	1297.8
42	400	4.65	7.4	0	2.15	3.59	3.39	14.2	12.05	6.80	1551
43	400	4.65	7.2	0	2.15	3.58	3.58	14	11.85	6.75	1551
44	400	4.1	2	0	2.7	3.5	2.7	8.8	6.1	4.80	610.7
45	400	4.35	8	0.95	2.05	3.44	2.09	15.35	13.3	7.60	1152
46	400	4.05	8	0.66	2.35	3.46	2.4	15.06	12.71	7.56	1318
47	400	3.5	8	0.2	2.9	3.51	2.91	14.6	11.7	7.50	960
48	400	4.35	8	0.98	2.05	3.48	2.1	15.38	13.33	7.62	1224.8
49	400	4.65	7.5	0	2.15	3.59	3.29	14.3	12.15	6.82	1551
50	400	4.65	7.46	0	2.15	3.56	3.3	14.26	12.11	6.81	1551
51	400	4.25	8	0.2	2.15	3.55	2.95	14.6	12.45	7.20	1392
52	400	4.25	8	1.02	2.15	3.58	2.16	15.42	13.27	7.68	1344
53	400	3.4	7.24	0	3.4	3.44	3.4	14.04	10.64	7.26	967
54	400	4.25	8	0.99	2.15	3.54	2.15	15.39	13.24	7.66	1248
55	400	4.65	7.2	0	2.15	3.58	3.58	14	11.85	6.75	1392
56	400	4.1	2	0	2.7	3.54	2.74	8.8	6.1	4.80	712.5
57	400	4.65	6.3	0	2.15	3.55	4.45	13.1	10.95	6.53	1440
58	300	3.4	5.2	0	3.4	3.45	3.45	12	8.6	6.73	559.8
59	300	3.4	5.3	0	3.4	3.5	3.4	12.1	8.7	6.76	661.6
60	400	4.25	8	0.96	2.15	3.54	2.18	15.36	13.21	7.65	1395
61	400	4.25	8	1.02	2.15	3.58	2.16	15.42	13.27	7.68	1344
62	400	4.65	7.4	0	2.15	3.59	3.39	14.2	12.05	6.80	1551
63	400	3.4	7.35	0	3.4	3.56	3.41	14.15	10.75	7.29	1052.4
64	400	4.35	8	1.07	2.05	3.52	2.05	15.47	13.42	7.67	1082.3
65	400	4.75	7.6	0	2.05	3.44	3.04	14.4	12.35	6.81	1473
66	400	4.25	8	0.9	2.15	3.56	2.26	15.3	13.15	7.61	1395
67	300	3.4	5.35	0	3.4	3.57	3.42	12.15	8.75	6.78	661.6
68	400	3.5	8	0.18	2.9	3.5	2.92	14.58	11.68	7.49	1032.4
69	300	3.4	5.2	0	3.4	3.42	3.42	12	8.6	6.73	559.8
70	400	3.4	7.33	0	3.4	3.55	3.42	14.13	10.73	7.28	1094.25
71	400	4.25	8	1	2.15	3.55	2.15	15.4	13.25	7.67	1248
72	400	3.45	8	0.2	2.95	3.52	2.92	14.6	11.65	7.52	967
73	400	3.5	8	0.17	2.9	3.47	2.9	14.57	11.67	7.48	960
74	400	3.45	8	0.14	2.95	3.52	2.98	14.54	11.59	7.48	885
75	400	3.45	8	0.07	2.95	3.42	2.95	14.47	11.52	7.44	1240
76	400	5.4	6.3	0	2.15	3.52	1.06	13.1	14.7	5.50	1056
77	300	3.4	5.2	0	3.4	3.43	3.43	12	8.6	6.73	600.7
78	300	3.4	5.3	0	3.4	3.52	3.42	12.1	8.7	6.76	508.9
79	400	3.55	5.36	0	3.25	3.41	3.25	12.16	8.91	6.71	930

Table A1. *Cont.*

S. No.	D	X_1	X_2	X_3	X_p	X_g	X_t	X_m	N_s	N_t	*Pu*
80	400	4.35	8	1.18	2.05	3.66	2.08	15.58	13.53	7.73	1056
81	400	4.1	2	0	2.7	3.52	2.72	8.8	6.1	4.80	610.7
82	300	3.4	5.25	0	3.4	3.49	3.44	12.05	8.65	6.75	610.7
83	400	4.25	8	1	2.15	3.55	2.15	15.4	13.25	7.67	1344
84	300	3.4	5.2	0	3.4	3.38	3.38	12	8.6	6.73	610.7
85	400	4.25	8	0.9	2.15	3.59	2.29	15.3	13.15	7.61	1473
86	400	4.1	1.85	0	2.7	3.35	2.7	8.65	5.95	4.68	508.9
87	300	3.4	5.2	0	3.4	3.43	3.43	12	8.6	6.73	661.6
88	400	4.25	8	0.94	2.15	3.54	2.2	15.34	13.19	7.64	1395
89	400	4.25	8	0.9	2.15	3.59	2.29	15.3	13.15	7.61	1551
90	400	4.75	7.25	0	2.05	3.65	3.6	14.05	12	6.73	1425
91	400	4.25	8	1.02	2.15	3.58	2.16	15.42	13.27	7.68	1152
92	400	4.35	8	1.05	2.05	3.53	2.08	15.45	13.4	7.66	1473
93	400	3.45	8	0.14	2.95	3.52	2.98	14.54	11.59	7.48	885
94	400	4.1	1.9	0	2.7	3.43	2.73	8.7	6	4.72	620
95	400	4.35	8	0.97	2.05	3.42	2.05	15.37	13.32	7.61	1317
96	400	4.65	6.49	0	2.15	3.59	4.3	13.29	11.14	6.58	1551
97	400	3.4	7.31	0	3.4	3.56	3.45	14.11	10.71	7.28	1032.4
98	300	3.4	5.25	0	3.4	3.48	3.43	12.05	8.65	6.75	610.7
99	400	3.45	8	0.19	2.95	3.56	2.97	14.59	11.64	7.52	1318
100	400	3.45	6.29	0	3.35	3.44	3.35	13.09	9.74	7.02	1240
101	300	3.4	5.24	0	3.4	3.49	3.45	12.04	8.64	6.75	610.7
102	400	4.25	8	0.7	2.15	3.58	2.48	15.1	12.95	7.50	1392
103	300	3.4	5.25	0	3.4	3.47	3.42	12.05	8.65	6.75	585.4
104	400	4.25	8	1	2.15	3.56	2.16	15.4	13.25	7.67	1152
105	400	4.1	1.8	0	2.7	3.32	2.72	8.6	5.9	4.64	559.8
106	400	3.4	7.3	0	3.4	3.49	3.39	14.1	10.7	7.28	1068.8
107	400	4.35	8	1	2.05	3.45	2.05	15.4	13.35	7.63	1119.7
108	400	3.4	7.31	0	3.4	3.54	3.43	14.11	10.71	7.28	1032.8
109	400	3.45	8	0.1	2.95	3.54	3.04	14.5	11.55	7.46	1017.9
110	300	3.4	5.2	0	3.4	3.48	3.48	12	8.6	6.73	611.6
111	400	4.75	7.6	0	2.05	3.49	3.09	14.4	12.35	6.81	1473
112	400	4.35	8	1.04	2.05	3.52	2.08	15.44	13.39	7.65	1321
113	400	3.5	8	0.21	2.9	3.48	2.87	14.61	11.71	7.51	1032.4
114	400	4.65	7.2	0	2.15	3.55	3.55	14	11.85	6.75	1392
115	400	4.35	8	1.08	2.05	3.53	2.05	15.48	13.43	7.67	1248
116	300	3.4	5.25	0	3.4	3.46	3.41	12.05	8.65	6.75	661.6
117	300	3.4	5.2	0	3.4	3.41	3.41	12	8.6	6.73	610.7
118	400	4.35	8	1.1	2.05	3.55	2.05	15.5	13.45	7.69	1425
119	400	4.35	8	0.05	2.05	3.58	3.13	14.45	12.4	7.07	1344
120	400	4.1	2.08	0	2.7	3.63	2.75	8.88	6.18	4.86	432

Table A1. *Cont.*

S. No.	D	X_1	X_2	X_3	X_p	X_g	X_t	X_m	N_s	N_t	*Pu*
121	300	3.4	5.25	0	3.4	3.48	3.43	12.05	8.65	6.75	559.8
122	400	3.85	7.35	0	2.95	3.64	3.49	14.15	11.2	7.09	1425
123	300	3.4	5.25	0	3.4	3.48	3.43	12.05	8.65	6.75	508.9
124	400	4.65	7.5	0	2.15	3.59	3.29	14.3	12.15	6.82	1551
125	300	3.4	5.3	0	3.4	3.5	3.4	12.1	8.7	6.76	559.8
126	300	3.4	5.32	0	3.4	3.55	3.43	12.12	8.72	6.77	661.6
127	300	3.4	5.25	0	3.4	3.48	3.43	12.05	8.65	6.75	559.8
128	400	3.5	8	0.16	2.9	3.48	2.92	14.56	11.66	7.47	960
129	400	4.65	7.5	0	2.15	3.55	3.25	14.3	12.15	6.82	1551
130	400	4.75	7.5	0	2.05	3.45	3.15	14.3	12.25	6.79	1297.8
131	300	3.4	5.2	0	3.4	3.42	3.42	12	8.6	6.73	610.7
132	400	4.35	8	1.01	2.05	3.46	2.05	15.41	13.36	7.64	1550
133	300	3.4	5.2	0	3.4	3.41	3.41	12	8.6	6.73	610.7
134	400	3.4	7.3	0	3.4	3.54	3.44	14.1	10.7	7.28	967
135	400	4.25	8	1.03	2.15	3.58	2.15	15.43	13.28	7.69	1248
136	300	3.4	5.25	0	3.4	3.46	3.41	12.05	8.65	6.75	559.8
137	300	3.4	5.3	0	3.4	3.51	3.41	12.1	8.7	6.76	661.6
138	400	4.25	8	0.4	2.15	3.55	2.75	14.8	12.65	7.32	1392
139	400	4.35	8	0.95	2.05	3.41	2.06	15.35	13.3	7.60	1110.6
140	300	3.4	5.2	0	3.4	3.4	3.4	12	8.6	6.73	559.8
141	400	3.85	7.3	0	2.95	3.68	3.58	14.1	11.15	7.08	1440
142	400	4.1	2.08	0	2.7	3.58	2.7	8.88	6.18	4.86	480
143	400	4.45	8	1.18	1.95	3.58	2	15.58	13.63	7.69	1032.4
144	300	3.4	5.2	0	3.4	3.4	3.4	12	8.6	6.73	559.8
145	300	3.4	5.2	0	3.4	3.43	3.43	12	8.6	6.73	661.6
146	300	3.4	5.25	0	3.4	3.46	3.41	12.05	8.65	6.75	407.2
147	400	3.45	8	0.22	2.95	3.57	2.95	14.62	11.67	7.53	1318
148	400	4.25	8	1.01	2.15	3.57	2.16	15.41	13.26	7.68	1248
149	400	3.4	7.3	0	3.4	3.5	3.4	14.1	10.7	7.28	958
150	400	4.1	2.2	0	2.7	3.72	2.72	9	6.3	4.94	610.7
151	400	4.35	8	1.02	2.05	3.47	4.05	15.42	13.37	7.64	1318
152	400	4.25	8	0.9	2.15	3.53	2.23	15.3	13.15	7.61	1395
153	400	4.25	8	0.4	2.15	3.59	2.79	14.8	12.65	7.32	1551
154	300	3.4	5.24	0	3.4	3.48	3.44	12.04	8.64	6.75	559.8
155	400	4.25	8	0.4	2.15	3.55	2.75	14.8	12.65	7.32	1392
156	300	3.4	5.25	0	3.4	3.46	3.41	12.05	8.65	6.75	661.6
157	400	4.05	8	0.7	2.35	3.47	2.37	15.1	12.75	7.58	1318
158	300	3.4	5.23	0	3.4	3.44	3.41	12.03	8.63	6.74	585.35
159	400	4.35	8	0.7	2.05	3.49	2.39	15.1	13.05	7.46	1392
160	400	4.25	8	1	2.15	3.57	2.17	15.4	13.25	7.67	1248

Table A1. *Cont.*

S. No.	D	X_1	X_2	X_3	X_p	X_g	X_t	X_m	N_s	N_t	*Pu*
161	400	4.25	8	1	2.15	3.58	2.18	15.4	13.25	7.67	1395
162	400	4.25	8	1	2.15	3.56	2.16	15.4	13.25	7.67	1395
163	400	4.25	8	1.01	2.15	3.57	2.16	15.41	13.26	7.68	1248
164	400	4.25	8	0.1	2.15	3.53	3.03	14.5	12.35	7.14	1551
165	400	3.5	8	0.17	2.9	3.48	2.91	14.57	11.67	7.48	1056
166	400	4.25	8	1.02	2.15	3.58	2.16	15.42	13.27	7.68	1248
167	300	3.4	5.25	0	3.4	3.46	3.41	12.05	8.65	6.75	532.4
168	400	4.35	8	0.8	2.05	3.45	2.25	15.2	13.15	7.52	1392
169	300	3.4	5.2	0	3.4	3.45	3.45	12	8.6	6.73	610.7
170	400	4.25	8	0.98	2.15	3.54	2.16	15.38	13.23	7.66	1344
171	400	4.25	8	1	2.15	3.56	2.16	15.4	13.25	7.67	1344
172	400	3.45	8	0.25	2.95	3.6	2.95	14.65	11.7	7.55	960
173	400	4.65	7.24	0	2.15	3.54	3.5	14.04	11.89	6.76	1551
174	400	4.25	8	0.9	2.15	3.58	2.28	15.3	13.15	7.61	1395
175	400	3.4	7.3	0	3.4	3.5	3.4	14.1	10.7	7.28	900
176	400	3.4	7.4	0	3.4	3.61	3.41	14.2	10.8	7.30	1088.8
177	400	4.25	8	0.1	2.15	3.54	3.04	14.5	12.35	7.14	1551
178	400	3.4	7.23	0	3.4	3.43	3.4	14.03	10.63	7.26	960
179	300	3.4	5.3	0	3.4	3.52	3.42	12.1	8.7	6.76	610.7
180	400	4.1	2	0	2.7	3.55	2.75	8.8	6.1	4.80	610.7
181	400	4.25	8	1.03	2.15	3.58	2.15	15.43	13.28	7.69	1248
182	400	3.45	8	0.12	2.95	3.47	2.95	14.52	11.57	7.47	1318
183	400	4.25	8	1	2.15	3.58	2.18	15.4	13.25	7.67	1395
184	400	4.35	8	1.11	2.05	3.56	2.05	15.51	13.46	7.69	1128.6
185	400	4.45	7.21	0	2.35	3.41	2.4	14.01	11.66	6.83	1318
186	400	4.65	7.38	0	2.15	3.58	3.4	14.18	12.03	6.79	1551
187	400	4.25	8	1	2.15	3.56	2.16	15.4	13.25	7.67	1248
188	400	4.25	8	0.2	2.15	3.58	2.98	14.6	12.45	7.20	1551
189	400	4.65	7.6	0	2.15	3.58	3.18	14.4	12.25	6.84	1446
190	300	3.4	5.22	0	3.4	3.44	3.42	12.02	8.62	6.74	617
191	400	4.75	7.4	0	2.05	3.52	3.32	14.2	12.15	6.76	1425
192	400	4.65	7.4	0	2.15	3.59	3.39	14.2	12.05	6.80	1392
193	400	3.4	7.3	0	3.4	3.61	3.51	14.1	10.7	7.28	1115.2
194	300	3.4	5.25	0	3.4	3.49	3.44	12.05	8.65	6.75	559.8
195	300	3.4	5.25	0	3.4	3.46	3.41	12.05	8.65	6.75	559.8
196	400	4.25	8	1	2.15	3.58	2.18	15.4	13.25	7.67	1395
197	300	3.4	5.18	0	3.4	3.38	3.4	11.98	8.58	6.73	559.8
198	400	4.25	8	0.91	2.15	3.56	2.25	15.31	13.16	7.62	1473
199	400	4.05	8	0.7	2.35	3.48	2.38	15.1	12.75	7.58	1238
200	400	4.1	2.01	0	2.7	3.53	2.72	8.81	6.11	4.80	528

References

1. Momeni, E.; Nazir, R.; Armaghani, D.J.; Maizir, H. Application of artificial neural network for predicting shaft and tip resistances of concrete piles. *Earth Sci. Res. J.* **2015**, *19*, 85–93. [CrossRef]
2. Drusa, M.; Gago, F.; Vlček, J. Contribution to Estimating Bearing Capacity of Pile in Clayey Soils. *Civ. Environ. Eng.* **2016**, *12*, 128–136. [CrossRef]
3. Meyerhof, G.G. Bearing Capacity and Settlement of Pile Foundations. *J. Geotech. Eng. Div.* **1976**, *102*, 197–228. [CrossRef]
4. Shooshpasha, I.; Hasanzadeh, A.; Taghavi, A. Prediction of the axial bearing capacity of piles by SPT-based and numerical design methods. *Int. J. GEOMATE* **2013**, *4*, 560–564. [CrossRef]
5. Chai, X.J.; Deng, K.; He, C.F.; Xiong, Y.F. Laboratory model tests on consolidation performance of soil column with drained-timber rod. *Adv. Civ. Eng.* **2021**, *2021*, 6698894. [CrossRef]
6. ASTM. *American Society for Testing and Materials—ASTM D4945-08 Standard Test Method for High-Strain Dynamic Testing of Deep Foundations*; ASTM: West Conshohocken, PA, USA, 2008; Volume 1, p. 10.
7. Schmertmann, J. *Guidelines for Cone Penetration Test: Performance and Design*; (No. FHWA-TS-78-209); Federal Highway Administration: Washington, DC, USA, 1978.
8. Budi, G.S.; Kosasi, M.; Wijaya, D.H. Bearing capacity of pile foundations embedded in clays and sands layer predicted using PDA test and static load test. *Procedia Eng.* **2015**, *125*, 406–410. [CrossRef]
9. Kozłowski, W.; Niemczynski, D. Methods for Estimating the Load Bearing Capacity of Pile Foundation Using the Results of Penetration Tests—Case Study of Road Viaduct Foundation. *Procedia Eng.* **2016**, *161*, 1001–1006. [CrossRef]
10. Birid, K.C. Evaluation of Ultimate Pile Compression Capacity from Static Pile Load Test Results. In *International Congress and Exhibition "Sustainable Civil Infrastructures: Innovative Infrastructure Geotechnology"*; Springer: Cham, Switzerland, 2018; pp. 1–14. [CrossRef]
11. Ma, B.; Li, Z.; Cai, K.; Liu, M.; Zhao, M.; Chen, B.; Chen, Q.; Hu, Z. Pile-Soil Stress Ratio and Settlement of Composite Foundation Bidirectionally Reinforced by Piles and Geosynthetics under Embankment Load. *Adv. Civ. Eng.* **2021**, *2021*, 5575878. [CrossRef]
12. Tang, Y.; Huang, S.; Tao, J. *Geo-Congress 2020*; GSP 320 121; ASCE: Reston, VA, USA, 2020; pp. 121–131. [CrossRef]
13. Nurdin, S.; Sawada, K.; Moriguchi, S. Design Criterion of Reinforcement on Thick Soft Clay Foundations of Traditional Construction Method in Indonesia. *MATEC Web Conf.* **2019**, *258*, 03010. [CrossRef]
14. Momeni, E.; Maizir, H.; Gofar, N.; Nazir, R. Comparative study on prediction of axial bearing capacity of driven piles in granular materials. *J. Teknol.* **2013**, *61*, 15–20. [CrossRef]
15. Lopes, F.R.; Laprovitera, H. Prediction of the Bearing Capacity of Bored Piles from Dynamic Penetration Tests. In Proceedings of the 1st International Geoteclmical Seminar on Deep Foundations on Bored and Auger Piles, Ghent, Belgium, 7–10 June 1988; pp. 537–540.
16. Decourt, L. Prediction of load-settlement relationships for foundations on the basis of the SPT. In Proceedings of the Ciclo de Conferencias Internationale, Leonardo Zeevaert, UNAM, Mexico City, Mexico, 1995; pp. 85–104.
17. Pham, T.A.; Ly, H.-B.; Tran, V.Q.; Van Giap, L.; Vu, H.-L.T.; Duong, H.-A.T. Prediction of Pile Axial Bearing Capacity Using Artificial Neural Network and Random Forest. *Appl. Sci.* **2020**, *10*, 1871. [CrossRef]
18. Ahmad, M.; Ahmad, F.; Huang, J.; Iqbal, M.J.; Safdar, M.; Pirhadi, N. Probabilistic evaluation of CPT-based seismic soil liquefaction potential: Towards the integration of interpretive structural modeling and bayesian belief network. *Math. Biosci. Eng.* **2021**, *18*, 9233–9252. [CrossRef] [PubMed]
19. Ahmad, M.; Tang, X.-W.; Ahmad, F.; Jamal, A. Assessment of Soil Liquefaction Potential in Kamra, Pakistan. *Sustainability* **2018**, *10*, 4223. [CrossRef]
20. Ahmad, M.; Al-Shayea, N.A.; Tang, X.-W.; Jamal, A.; Al-Ahmadi, H.M.; Ahmad, F. Predicting the Pillar Stability of Underground Mines with Random Trees and C4.5 Decision Trees. *Appl. Sci.* **2020**, *10*, 6486. [CrossRef]
21. Liu, Q.; Cao, Y.; Wang, C. Prediction of Ultimate Axial Load-Carrying Capacity for Driven Piles Using Machine Learning Methods. In Proceedings of the 3rd Information Technology, Networking, Electronic and Automation Control Conference, Chengdu, China, 15–17 March 2019; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2019; pp. 334–340. [CrossRef]
22. Ahmad, M.; Ahmad, F.; Wróblewski, P.; Al-Mansob, R.A.; Olczak, P.; Kamiński, P.; Safdar, M.; Rai, P. Prediction of Ultimate Bearing Capacity of Shallow Foundations on Cohesionless Soils: A Gaussian Process Regression Approach. *Appl. Sci.* **2021**, *11*, 10317. [CrossRef]
23. Liang, W.; Luo, S.; Zhao, G.; Wu, H. Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms. *Mathematics* **2020**, *8*, 765. [CrossRef]
24. Pham, B.T.; Nguyen, M.D.; Van Dao, D.; Prakash, I.; Ly, H.-B.; Le, T.-T.; Ho, L.S.; Nguyen, K.T.; Ngo, T.Q.; Hoang, V.; et al. Development of artificial intelligence models for the prediction of Compression Coefficient of soil: An application of Monte Carlo sensitivity analysis. *Sci. Total Environ.* **2019**, *679*, 172–184. [CrossRef]
25. Ahmad, M.; Tang, X.-W.; Qiu, J.-N.; Ahmad, F. Evaluating Seismic Soil Liquefaction Potential Using Bayesian Belief Network and C4.5 Decision Tree Approaches. *Appl. Sci.* **2019**, *9*, 4226. [CrossRef]
26. Ahmad, M.; Kamiński, P.; Olczak, P.; Alam, M.; Iqbal, M.; Ahmad, F.; Sasui, S.; Khan, B. Development of Prediction Models for Shear Strength of Rockfill Material Using Machine Learning Techniques. *Appl. Sci.* **2021**, *11*, 6167. [CrossRef]
27. Ahmad, M.; Hu, J.-L.; Hadzima-Nyarko, M.; Ahmad, F.; Tang, X.-W.; Rahman, Z.; Nawaz, A.; Abrar, M. Rockburst Hazard Prediction in Underground Projects Using Two Intelligent Classification Techniques: A Comparative Study. *Symmetry* **2021**, *13*, 632. [CrossRef]
28. Ahmad, M.; Hu, J.-L.; Ahmad, F.; Tang, X.-W.; Amjad, M.; Iqbal, M.; Asim, M.; Farooq, A. Supervised Learning Methods for Modeling Concrete Compressive Strength Prediction at High Temperature. *Materials* **2021**, *14*, 1983. [CrossRef] [PubMed]

29. Goh, A.T.C.; Kulhawy, F.H.; Chua, C.G. Bayesian Neural Network Analysis of Undrained Side Resistance of Drilled Shafts. *J. Geotech. Geoenviron. Eng.* **2005**, *131*, 84–93. [CrossRef]
30. Goh, A.T.C. Back-propagation neural networks for modeling complex systems. *Artif. Intell. Eng.* **1995**, *9*, 143–151. [CrossRef]
31. Shahin, M.A.; Jaksa, M.B. Neural network prediction of pullout capacity of marquee ground anchors. *Comput. Geotech.* **2005**, *32*, 153–163. [CrossRef]
32. Shahin, M.A. Intelligent computing for modeling axial capacity of pile foundations. *Can. Geotech. J.* **2010**, *47*, 230–243. [CrossRef]
33. Shahin, M.A. Load–settlement modeling of axially loaded steel driven piles using CPT-based recurrent neural networks. *Soils Found.* **2014**, *54*, 515–522. [CrossRef]
34. Shahin, M.A. State-of-the-art review of some artificial intelligence applications in pile foundations. *Geosci. Front.* **2016**, *7*, 33–44. [CrossRef]
35. Nawari, N.O.; Liang, R.; Nusairat, J. Artificial intelligence techniques for the design and analysis of deep foundations. *Electron. J. Geotech. Eng.* **1999**, *4*, 1–21.
36. Momeni, E.; Nazir, R.; Armaghani, D.J.; Maizir, H. Prediction of pile bearing capacity using a hybrid genetic algorithm-based ANN. *Measurement* **2014**, *57*, 122–131. [CrossRef]
37. Kordjazi, A.; Nejad, F.P.; Jaksa, M. Prediction of ultimate axial load-carrying capacity of piles using a support vector machine based on CPT data. *Comput. Geotech.* **2014**, *55*, 91–102. [CrossRef]
38. Pham, T.A.; Tran, V.Q.; Vu, H.L.T.; Ly, H.B. Design deep neural network architecture using a genetic algorithm for estimation of pile bearing capacity. *PLoS ONE* **2020**, *15*, e0243030. [CrossRef] [PubMed]
39. Tama, B.A.; Rhee, K.H. An in-depth experimental study of anomaly detection using gradient boosted machine. *Neural Comput. Appl.* **2019**, *31*, 955–965. [CrossRef]
40. Sun, R.; Wang, G.; Zhang, W.; Hsu, L.T.; Ochieng, W.Y. A gradient boosting decision tree based GPS signal reception classification algorithm. *Appl. Soft Comput.* **2020**, *86*, 105942. [CrossRef]
41. Lombardo, L.; Cama, M.; Conoscenti, C.; Märker, M.; Rotigliano, E. Binary logistic regression versus stochastic gradient boosted decision trees in assessing landslide susceptibility for multiple-occurring landslide events: Application to the 2009 storm event in Messina (Sicily, southern Italy). *Nat. Hazards* **2015**, *79*, 1621–1648. [CrossRef]
42. Sachdeva, S.; Bhatia, T.; Verma, A.K. GIS-based evolutionary optimized Gradient Boosted Decision Trees for forest fire susceptibility mapping. *Nat. Hazards* **2018**, *92*, 1399–1418. [CrossRef]
43. Kotsiantis, S.B. Decision trees: A recent overview. *Artif. Intell. Rev.* **2013**, *39*, 261–283. [CrossRef]
44. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
45. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
46. Javadi, A.A.; Rezania, M.; Nezhad, M.M. Evaluation of liquefaction induced lateral displacements using genetic programming. *Comput. Geotech.* **2006**, *33*, 222–233. [CrossRef]
47. van Vuren, T. Modeling of transport demand—Analyzing, calculating, and forecasting transport demand. *Transp. Rev.* **2020**, *40*, 115–117. [CrossRef]
48. Song, Y.; Gong, J.; Gao, S.; Wang, D.; Cui, T.; Li, Y.; Wei, B. Susceptibility assessment of earthquake-induced landslides using Bayesian network: A case study in Beichuan, China. *Comput. Geosci.* **2012**, *42*, 189–199. [CrossRef]
49. Kaggle. Your Machine Learning and Data Science Community. Available online: https://www.kaggle.com/ (accessed on 2 December 2021).
50. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
51. Ahmad, M.W.; Mourshed, M.; Rezgui, Y. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy Build.* **2017**, *147*, 77–89. [CrossRef]
52. Freund, Y.; Schapire, R.E. Experiments with a New Boosting Algorithm. In Proceedings of the 13th International Conference on International Conference on Machine Learning, Bari, Italy, 3–6 July 1996; pp. 148–156.
53. Schapire, R.E. Explaining AdaBoost. In *Empirical Inference*; Schölkopf, B., Luo, Z., Vovk, V., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 37–52. [CrossRef]
54. Seo, D.K.; Kim, Y.H.; Eo, Y.D.; Park, W.Y.; Park, H.C. Generation of Radiometric, Phenological Normalized Image Based on Random Forest Regression for Change Detection. *Remote Sens.* **2017**, *9*, 1163. [CrossRef]
55. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
56. Hsu, C.W.; Chang, C.C.; Lin, C.J. *A Practical Guide to Support Vector Classification*; National Taiwan University: Taipei, Taiwan, 2003; Available online: http://www.csie.ntu.edu.tw/~cjlin (accessed on 1 July 2021).
57. Fowler, B. A sociological analysis of the satanic verses affair. *Theory Cult. Soc.* **2000**, *17*, 39–61. [CrossRef]
58. Barakat, N.; Bradley, A.P. Rule extraction from support vector machines: A review. *Neurocomputing* **2010**, *74*, 178–190. [CrossRef]
59. Martens, D.; Huysmans, J.; Setiono, R.; Vanthienen, J.; Baesens, B. Rule extraction from support vector machines: An overview of issues and application in credit scoring. *Rule Extr. Support Vector Mach.* **2008**, *80*, 33–63. [CrossRef]
60. Uslan, V.; Seker, H. Support Vector-Based Takagi-Sugeno Fuzzy System for the Prediction of Binding Affinity of Peptides. In Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Osaka, Japan, 3–7 July 2013; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2013; pp. 4062–4065. [CrossRef]

61. Gandomi, A.H.; Fridline, M.M.; Roke, D.A. Decision Tree Approach for Soil Liquefaction Assessment. *Sci. World J.* **2013**, *2013*, 346285. [CrossRef]

62. Amirkiyaei, V.; Ghasemi, E. Stability assessment of slopes subjected to circular-type failure using tree-based models. *Int. J. Geotech. Eng.* **2020**, 1862538. [CrossRef]

63. Tiryaki, B. Predicting intact rock strength for mechanical excavation using multivariate statistics, artificial neural networks, and regression trees. *Eng. Geol.* **2008**, *99*, 51–60. [CrossRef]

64. Hasanipanah, M.; Faradonbeh, R.S.; Amnieh, H.B.; Armaghani, D.J.; Monjezi, M. Forecasting blast-induced ground vibration developing a CART model. *Eng. Comput.* **2017**, *33*, 307–316. [CrossRef]

65. Khosravi, K.; Mao, L.; Kisi, O.; Yaseen, Z.M.; Shahid, S. Quantifying hourly suspended sediment load using data mining models: Case study of a glacierized Andean catchment in Chile. *J. Hydrol.* **2018**, *567*, 165–179. [CrossRef]

66. Taylor, K.E. Summarizing multiple aspects of model performance in a single diagram. *J. Geophys. Res. Atmos.* **2001**, *106*, 7183–7192. [CrossRef]

67. Yang, Y.; Zhang, Q. A hierarchical analysis for rock engineering using artificial neural networks. *Rock Mech. Rock Eng.* **1997**, *30*, 207–222. [CrossRef]

68. Faradonbeh, R.S.; Armaghani, D.J.; Majid, M.Z.; Tahir, M.M.; Murlidhar, B.R.; Monjezi, M.; Wong, H.M. Prediction of ground vibration due to quarry blasting based on gene expression programming: A new model for peak particle velocity prediction. *Int. J. Environ. Sci. Technol.* **2016**, *13*, 1453–1464. [CrossRef]

69. Chen, W.; Hasanipanah, M.; Rad, H.N.; Armaghani, D.J.; Tahir, M.M. A new design of evolutionary hybrid optimization of SVR model in predicting the blast-induced ground vibration. *Eng. Comput.* **2021**, *37*, 1455–1471. [CrossRef]

70. Rad, H.N.; Bakhshayeshi, I.; Jusoh, W.A.W.; Tahir, M.M.; Foong, L.K. Prediction of Flyrock in Mine Blasting: A New Computational Intelligence Approach. *Nat. Resour. Res.* **2020**, *29*, 609–623. [CrossRef]

71. Momeni, E.; Armaghani, D.J.; Fatemi, S.A.; Nazir, R. Prediction of bearing capacity of thin-walled foundation: A simulation approach. *Eng. Comput.* **2018**, *34*, 319–327. [CrossRef]

72. Momeni, E.; Dowlatshahi, M.B.; Omidinasab, F.; Maizir, H.; Armaghani, D.J. Gaussian Process Regression Technique to Estimate the Pile Bearing Capacity. *Arab. J. Sci. Eng.* **2020**, *45*, 8255–8267. [CrossRef]

73. Kulkarni, R.U.; Dewaikar, D.M. Prediction of Interpreted Failure Loads of Rock-Socketed Piles in Mumbai Region using Hybrid Artificial Neural Networks with Genetic Algorithm. *Int. J. Eng. Res.* **2017**, *6*, 365–372. [CrossRef]

74. Armaghani, D.J.; Shoib, R.S.N.S.B.R.; Faizi, K.; Rashid, A.S.A. Developing a hybrid PSO–ANN model for estimating the ultimate bearing capacity of rock-socketed piles. *Neural Comput. Appl.* **2017**, *28*, 391–405. [CrossRef]