

Article

Prediction of Acceleration Amplification Ratio of Rocking Foundations Using Machine Learning and Deep Learning Models

Sivapalan Gajan

College of Engineering, SUNY Polytechnic Institute, Utica, NY 13502, USA; gajans@sunypoly.edu

Abstract: Experimental results reveal that rocking shallow foundations reduce earthquake-induced force and flexural displacement demands transmitted to structures and can be used as an effective geotechnical seismic isolation mechanism. This paper presents data-driven predictive models for maximum acceleration transmitted to structures founded on rocking shallow foundations during earthquake loading. Results from base-shaking experiments on rocking foundations have been utilized for the development of artificial neural network regression (ANN), k-nearest neighbors regression, support vector regression, random forest regression, adaptive boosting regression, and gradient boosting regression models. Acceleration amplification ratio, defined as the maximum acceleration at the center of gravity of a structure divided by the peak ground acceleration of the earthquake, is considered as the prediction parameter. For five out of six models developed in this study, the overall mean absolute percentage error in predictions in repeated k-fold cross validation tests vary between 0.128 and 0.145, with the ANN model being the most accurate and most consistent. The cross validation mean absolute error in predictions of all six models vary between 0.08 and 0.1, indicating that the maximum acceleration of structures supported by rocking foundations can be predicted within an average error limit of 8% to 10% of the peak ground acceleration of the earthquake.

Keywords: geotechnical engineering; rocking foundations; earthquake engineering; soil-structure interaction; artificial neural network; machine learning



Citation: Gajan, S. Prediction of Acceleration Amplification Ratio of Rocking Foundations Using Machine Learning and Deep Learning Models. *Appl. Sci.* **2023**, *13*, 12791. <https://doi.org/10.3390/app132312791>

Academic Editors: Rosario Montuori and Wei Gao

Received: 27 September 2023

Revised: 1 November 2023

Accepted: 28 November 2023

Published: 29 November 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Dynamic soil-structure interaction in shallow foundations has generally been modeled using mechanics-based models such as simple spring-dashpot models, beam on nonlinear Winkler foundation models, plasticity-based macro-element models, and continuum-based models. A recent review article summarizes the computational methods generally used to model dynamic soil–foundation–structure interactions during earthquake loading, particularly in the context of geotechnical engineering [1]. As the development of large experimental databases becomes increasingly common, the application of machine learning techniques in geotechnical engineering has been improving and becoming more effective [2]. Machine learning models generalize observed experimental behavior, capture the salient features that may not be captured by mechanics-based models, and can be used with mechanics-based models as complementary measures in engineering applications or can be combined with engineering mechanics using the emerging framework of theory-guided machine learning [3].

Machine learning algorithms such as logistic regression, decision trees, decision tree-based ensemble models, and artificial neural networks have been used in a variety of geotechnical engineering applications that include mechanical properties of soils, strength of soils, soil slope stability, bearing capacity of foundations, and dynamic response of soils during earthquake loading [4–9]. Recently, in dynamic soil–foundation–structure interactions, machine learning algorithms have been used to develop data-driven models for rocking-induced seismic energy dissipation in soil, peak rotation of foundation, and factor of safety for tipping-over failure of rocking shallow foundations [10,11].

The earthquake-induced peak acceleration of structures is one of the key seismic design parameters of buildings and bridges, as the seismic performance of these structures depends heavily on the inertial forces experienced by the structural members and non-structural components induced by the acceleration of structures [12,13]. For instance, base shear force (and bending moment) of structures during earthquake loading, a commonly used seismic design parameter for structures, is directly proportional to the horizontal acceleration at the effective height of the structure [14]. There have been several studies related to floor acceleration demands on structures during seismic loading for structures supported by traditional, fixed-base foundations [15–17].

Rocking shallow foundations is a recent research phenomenon that has been investigated to some extent, particularly using centrifuge and shaking table experiments [18–23]. Research on rocking foundations reveals that they dissipate seismic energy in soil, reduce acceleration, force and flexural drift demands transferred to the structures, and effectively perform as a geotechnical seismic isolation mechanism [24–26]. Numerical modeling methods and empirical methods are available to quantify the moment-rotation response, rotational stiffness, damping ratio and settlement-rotation relationships of rocking foundations [27,28]. This paper presents the development of models to predict the rocking induced, reduced peak acceleration demands on structures using machine learning algorithms that are trained and tested on experimental results from a rocking foundation database that covers a wide range of soil properties, foundation geometry, and structural configurations. Whereas the previously published research on the application of machine learning algorithms to rocking foundations focused on rocking induced seismic energy dissipation, peak rotation, and tipping-over stability of rocking foundations, the current work focuses on the acceleration amplification ratio (AAR) of rocking foundations. The motivation for the current work stems from the importance of reduced acceleration demands transmitted to structures supported by rocking foundations (one of the major, potential beneficial effects of rocking foundations, if adopted in civil engineering practice).

The objective of this study is to develop data-driven models for the prediction of maximum acceleration transmitted to the effective height (center of gravity) of relatively rigid, single degree of freedom type structures founded on rocking shallow foundations during earthquake loading using multiple machine learning and deep learning algorithms. The machine learning algorithms utilized in this study include artificial neural network regression, k-nearest neighbors regression, support vector regression, random forest regression, adaptive boosting regression, and gradient boosting regression. The results of these machine learning model predictions are compared with those of a multivariate linear regression machine learning model (used as the baseline model) and a statistics-based simple linear regression model. A brief background to the problem considered is presented first, along with the experimental data used in this study and input features to machine learning models. It is followed by brief descriptions of the machine learning algorithms utilized and how they are applied to the problem considered. Finally, the results, discussion and conclusions of the study are presented.

2. Rocking Foundations for Seismic Loading

2.1. Rocking Mechanism and Acceleration Amplification Ratio

Figure 1 shows the schematic of a simplified, relatively rigid, shear wall-type rocking structure supported by a shallow foundation, the major forces acting on the structure, and the forces and displacements acting at the soil–footing interface. The key parameters that govern the behavior of a rocking system include the critical contact area ratio of the rocking foundation (A/A_c), slenderness ratio of the rocking structure (h/B), and rocking coefficient of the soil–foundation–structure system (C_r) [26]. C_r is essentially a non-dimensional, normalized form of the ultimate moment capacity of rocking foundations and can be expressed by [29]:

$$C_r = \frac{B}{2 \cdot h} \cdot \left[1 - \frac{A_c}{A} \right] \quad (1)$$

where B is the width of the foundation in the direction of shaking and h is the effective height of the structure. The A/A_c is essentially a factor of safety for rocking foundations and defined as the ratio of total base area of the foundation (A) to minimum foundation area required to be in contact with the soil (A_c) to support the applied vertical load [18].

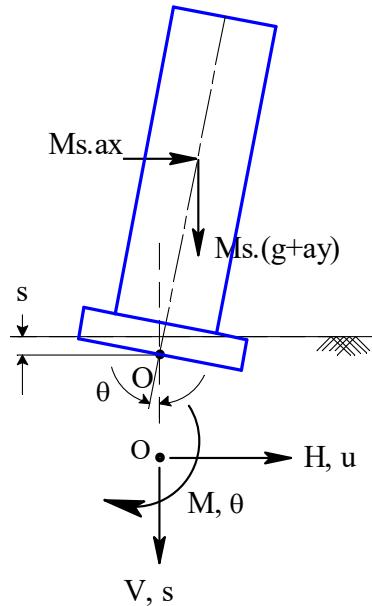


Figure 1. Simplified schematic of a rigid structure-foundation system rocking on soil and the major forces acting on it during earthquake loading.

The output parameter of machine learning models developed in this study is acceleration amplification ratio (AAR) of the rocking foundation and it is defined as:

$$AAR = \frac{a_{max,str}}{a_{max}} \quad (2)$$

where $a_{max,str}$ is the peak horizontal acceleration at the effective height of the structure and a_{max} is the peak horizontal ground acceleration of earthquake shaking. By comparing the maximum moment experienced by the soil–foundation system due to the inertial forces from the structure with the moment capacity of the rocking foundation, the following approximate relationship can be obtained for a theoretical upper bound for the AAR of a rocking foundation supporting a relatively rigid, SDOF-type structure [26].

$$AAR \leq \frac{C_r}{a_{max}} \quad (3)$$

Equation (3) implies that the foundation moment capacity limits the maximum seismic force demands transferred to the structure because of nonlinear soil–foundation–structure interaction. Though the relationship given in Equation (3) is approximate, it can be used to obtain simple, statistics-based empirical relationships for AAR.

2.2. Experimental Results

The experimental results utilized in this study are obtained from a total of nine series of centrifuge and shaking table experiments on rocking shallow foundations conducted at the University of California at Davis, the University of California at San Diego, and the National Technical University of Athens in Greece [30]. The details and major results of these individual test series are available in separate publications [18,29,31–36]. A summary of these experimental results and the effects of key rocking system capacity parameters (e.g., A/A_c and C_r) and earthquake demand parameters (e.g., a_{max}) on the performance

parameters of rocking foundations (e.g., AAR), derived from the data obtained from these experiments, are available in Gajan et al. (2021) [26].

Altogether, results obtained from 140 experiments on rocking foundations from the abovementioned series of experiments are utilized in this study. Figure 2 presents the variation of AAR with a_{max} (experimental results) for rocking structure-foundation systems with three different clusters of C_r , of which two clusters are for sandy soil foundations and one cluster is for clayey soil foundations. The AAR of rocking foundations are smaller than 1.0 for more than 82% of the experiments considered (for 116 out of 140 experiments), indicating that rocking foundations reduce the accelerations transferred to the structures they support (de-amplifying effect). This effect increases as C_r decreases, indicating that the foundations that have more tendency to rock (smaller C_r) de-amplify the acceleration more. The reduced acceleration demand on the structure during foundation rocking is a consequence of mobilization of bearing capacity and yielding of soil beneath the foundation during rocking. This de-amplifying effect is more pronounced for large amplitude shaking events (greater a_{max}) than for small shaking events, as soil yielding is even more significant during large amplitude shaking (a beneficial consequence of nonlinear soil-foundation interaction). Note that, for the experiments considered in this study, there is no noticeable difference between the AAR values of sandy and clayey soil foundations as long as the a_{max} and the range of C_r remain the same.

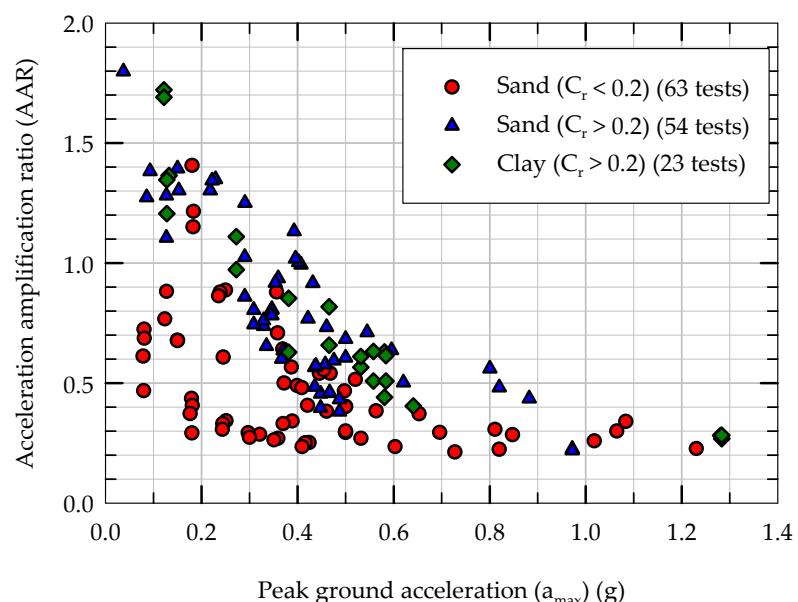


Figure 2. Results obtained from 140 centrifuge and shaking table experiments: Variation of AAR with a_{max} for rocking structure–foundation system with three clusters of rocking coefficient (C_r).

It should be noted that simplified procedures for estimating the peak acceleration demands on traditional fixed-base structures use a trapezoidal distribution, where peak acceleration at roof level could be about 3.0 to 4.0 times the peak ground acceleration of the earthquake [12]. For example, the American Society of Civil Engineers' document Minimum Design Loads for Buildings and Other Structures (ASCE 7-16) indicates that the floor acceleration amplification factor can be as high as 3.0 at roof level [37], while the National Earthquake Hazards Reduction Program's (NEHRP) Building Seismic Safety Council (BSSC) indicates that the floor acceleration amplification factor can be as high as 4.0 at roof level [38]. For the purpose of comparison, the abovementioned values correspond to an equivalent AAR of 3.0 to 4.0. This clearly shows that rocking foundations are much more efficient in de-amplifying the accelerations transferred to the structures during seismic loading (AAR < 1.5 for the vast majority of the experiments plotted in Figure 2).

Figure 3 plots the experimental results of AAR against C_r/a_{max} for all 140 rocking foundation experiments considered in this study (the same data plotted in Figure 2). Also,

included in Figure 3 are a 1:1 line and a best fit line obtained from statistics-based simple linear regression (SLR) using $\log(C_r/a_{max})$ and $\log(AAR)$ as the independent variable and the dependent variable, respectively. Though C_r/a_{max} is an approximate theoretical upper bound for AAR (Equation (3)), some of the experimental data show AAR values greater than C_r/a_{max} . This could possibly be because of the approximate nature of the upper bound relationship and the assumptions and simplifications involved in the derivations. The best fit SLR relationship yields a coefficient of determination (R^2) value of 0.75 (in log–log scale), indicating that there is room for improvement and better predictive relationships can be obtained by machine learning algorithms. In summary, the experimental results of AAR indicate that rocking foundations reduce the seismic force demands imposed on the structures by decreasing the acceleration transferred to structures, and that this beneficial de-amplifying effect increases as C_r decreases (for foundations that are more prone to rocking) and as a_{max} increases (for relatively larger magnitude earthquakes).

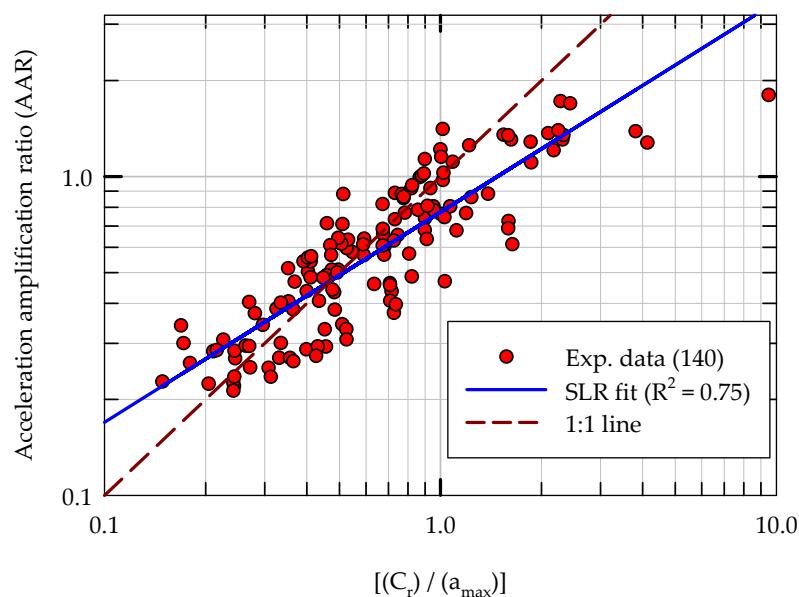


Figure 3. Experimental results for AAR as a function of C_r/a_{max} along with a 1:1 line and a statistics-based simple linear regression (SLR) best fit line in log–log space.

2.3. Input Features for Machine Learning Models

The input features for machine learning models have been selected based on their theoretical and experimentally observed close relationships with AAR, presented in Gajan et al. (2021) [26]. In addition, in order to predict other performance parameters of rocking foundations (namely, seismic energy dissipation, maximum rotation of rocking foundation and factor of safety for tipping over failure), the same set of input features have been found to be appropriate and successful [10,11]. The input features include three non-dimensional rocking system capacity parameters (A/A_c , h/B and C_r), and two earthquake loading demand parameters (a_{max} and Arias intensity of earthquake (I_a)). The a_{max} is the most commonly used ground motion intensity parameter in geotechnical earthquake engineering that characterizes the magnitude of shaking. Arias intensity of earthquake ground motion combines multiple key features of earthquake ground motion through numerical integration of acceleration time history in the time domain. These key features of ground motion include amplitude, duration, frequency content and number of cycles of earthquake loading. All the input feature parameters have been calculated for 140 individual experiments from the abovementioned series of experiments. Figure 4 presents the frequency plots, mean values, and standard deviations of all five input features. For ease of presentation, the frequency plots lump each input feature into five groups; the exact values of each input feature are used in training and testing of machine learning models.

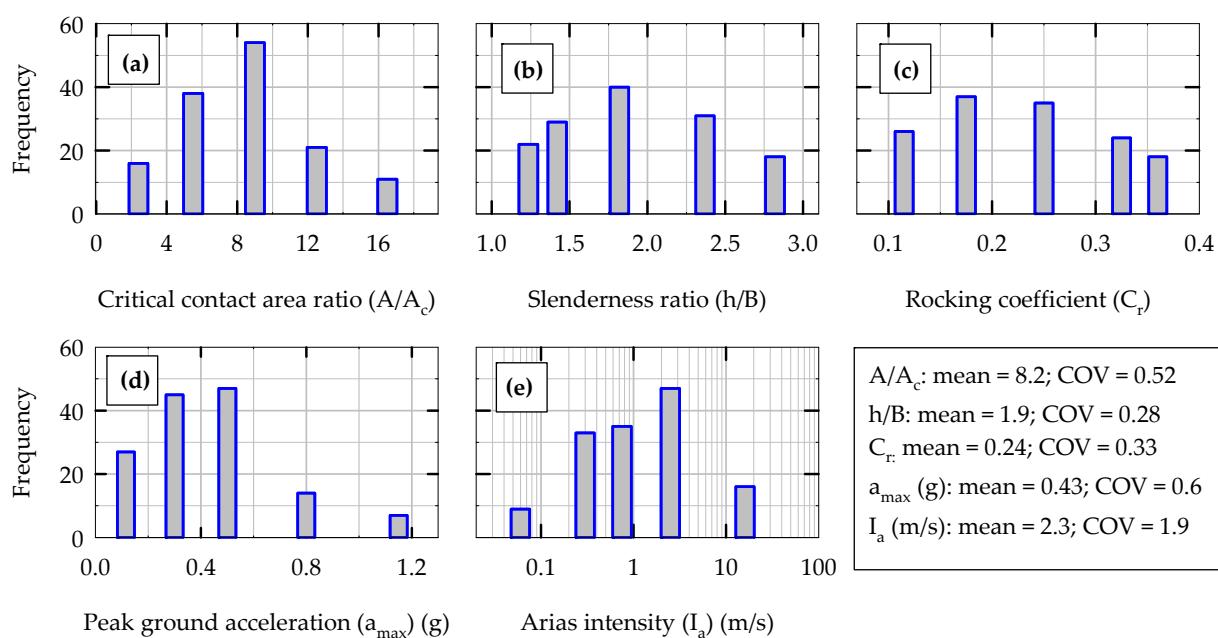


Figure 4. Frequency plots (frequency of occurrence in experiments) of input features for machine learning models developed in this study: (a) A/A_c , (b) h/B , (c) C_r , (d) a_{max} and (e) I_a .

As shown in Figure 4, the input features used in this study cover a wide range of rocking structure–foundation–soil system parameters (A/A_c , h/B and C_r) and earthquake demand parameters (a_{max} and I_a). As the variation of I_a is relatively high, it is transformed to a log scale (feature transformation). In addition, all the input feature values are normalized to vary between 0.0 and 1.0 (feature scaling). Figure 5 summarizes the research methodology in the form of a flow chart listing the experimental variables, input features, and the machine learning models developed to predict AAR.

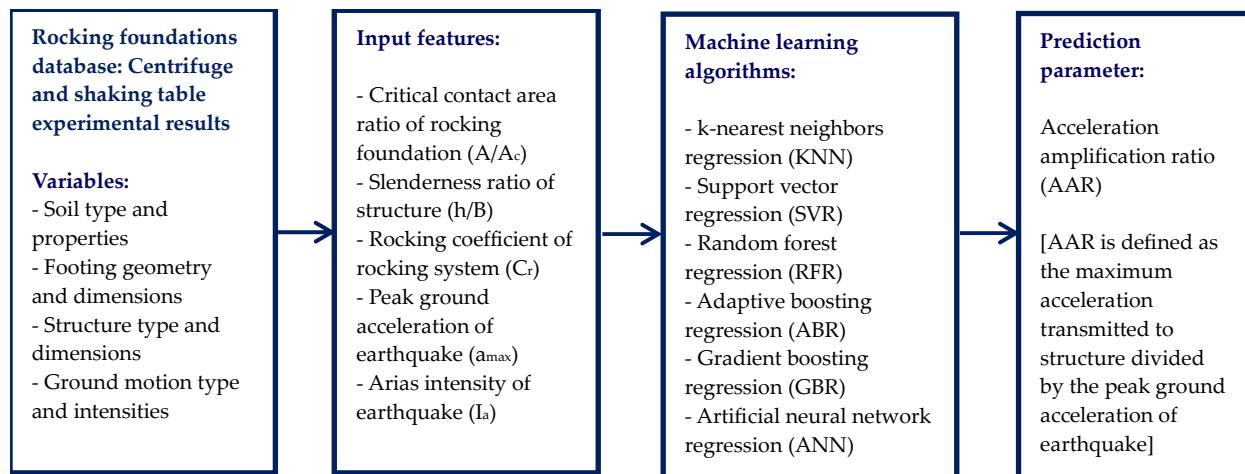


Figure 5. Flow chart showing the research methodology, experimental variables, input features, machine learning algorithms and prediction parameter.

3. Machine Learning Algorithms

3.1. Distance-Weighted K-Nearest Neighbors Regression (KNN)

The KNN algorithm considers data instances as multi-dimensional vectors, with the number of dimensions being equal to the number of input features. The algorithm calculates the distances between data points in this multi-dimensional space and assumes that the data points share similar properties with their close neighbors (and hence similar output values). The Euclidean distance measure is used to calculate the distance between any two

data points in 5-D space in this study. During the training phase, the KNN algorithm simply stores the entire training dataset as vectors. During the testing phase, the KNN algorithm goes through the entire training dataset and finds k number of training data points that are closer to the test data point (k nearest neighbors, where k is a hyperparameter of KNN model). The distance weighted KNN model used in this study predicts a weighted average output based on the outputs of the nearest neighbors of the test data and the inverse of the distances between the test data and its nearest neighbors.

3.2. Support Vector Regression (SVR)

Unlike commonly used regression machine learning algorithms (e.g., linear regression) that minimize the error between the predicted values and actual observations, the SVR algorithm fits a hyperplane to represent the training data within a threshold value. This threshold value (called the margin, ϵ) is a hyperparameter of the model. The SVR algorithm uses a kernel function to transform the data instances into multi-dimensional input feature space; the radial basis function (RBF) kernel is used in this study. As highly nonlinear data with multiple input features cannot be completely represented by a hyperplane and a margin, a tolerance is used for the margin. Another hyperparameter (called the penalty parameter, C) of the SVR algorithm controls the magnitude of this tolerance across all dimensions in input feature space. When making a prediction on test data, the SVR model simply uses the hyperplane to make the prediction.

3.3. Decision Tree Regression (DTR)

The DTR algorithm builds an inverted tree-type data structure by going through the training dataset and assigning data instances to branches of the tree using information gain as a measure of reduction in uncertainty in data. While building the tree, the DTR algorithm chooses the best input feature (k) and a threshold value (t_k) for that input feature to decide on the optimum split by minimizing a cost function. The cost function ($J(k, t_k)$) that the DTR algorithm minimizes is given by [39]:

$$J(k, t_k) = \frac{m_l}{m} \cdot E_l + \frac{m_r}{m} \cdot E_r \quad (4)$$

where E and m represent the mean absolute error and the number of data instances, respectively, and the subscripts l and r represent the left and right subsets of that node, respectively ($m = m_l + m_r$). The maximum depth of the tree is the major hyperparameter of the DTR model. When making a prediction on test data, the DTR model finds the appropriate leaf node and makes the prediction using the average value of the prediction parameter (AAR) in that leaf node.

3.4. Random Forest Regression (RFR)

The RFR is a bagging (bootstrap aggregation) ensemble machine learning algorithm that builds multiple DTR models of different depths using random subsets of training dataset (random sampling with replacement). To train individual (and independent) DTR models, the RFR model uses a random number of input features each time (i.e., the maximum number of features to be considered when building a DTR model is a hyperparameter of the RFR model). The idea is that by intentionally introducing randomness in the construction of the RFR model, the accuracy of predictions and the variance in prediction error will be reduced. The number of base DTR models in an RFR model is another hyperparameter of the model. When making a prediction on test data, the RFR model simply outputs the average of predictions of each individual DTR model in the ensemble.

3.5. Adaptive Boosting Regression (ABR)

The ABR algorithm uses a boosting technique, where multiple individual base DTR models are trained sequentially on the entire training dataset. Each successive DTR model attempts to focus more on the “difficult” data instances (i.e., the data instances for which

the prediction error of the preceding DTR model is high). Two sets of weights are used by the ABR algorithm: predictor weight for each individual DTR model and instance weight for each training data instance. During the training phase, these weights are adjusted in such a way that, when combined, the final prediction error will be minimum. When making a prediction on test data, the ABR model combines the predictions of all DTR models in the ensemble and weighs them using predictor weights.

3.6. Gradient Boosting Regression (GBR)

The GBR algorithm is similar to the ABR algorithm in that it builds multiple base DTR models in sequence on the entire training dataset with the successive DTR model attempting to correct the error made by its predecessor. The difference between ABR and GBR is that the GBR algorithm trains the successive base DTR models on the residual errors made by its predecessor. When making a prediction on test data, the GBR model simply adds the predictions made by all base DTR models in the ensemble. The optimum value for the learning rate, a hyperparameter of DTR-based boosting ensemble models, is found to be 0.1 for both ABR and GBR models using a trial and error procedure.

3.7. Artificial Neural Network Regression (ANN)

Figure 6 schematically illustrates the architecture of the multi-layer perceptron, deep learning ANN regression models considered in this study. The number of input neurons is equal to the number of input features (five), and the number of hidden layers and the number of neurons in each hidden layer are varied systematically using hyperparameter tuning, grid search and random search techniques. The commonly used feed-forward, back-propagation algorithm is used to propagate the input features and correct the errors during training of ANN models using the stochastic gradient descent (SGD) algorithm.

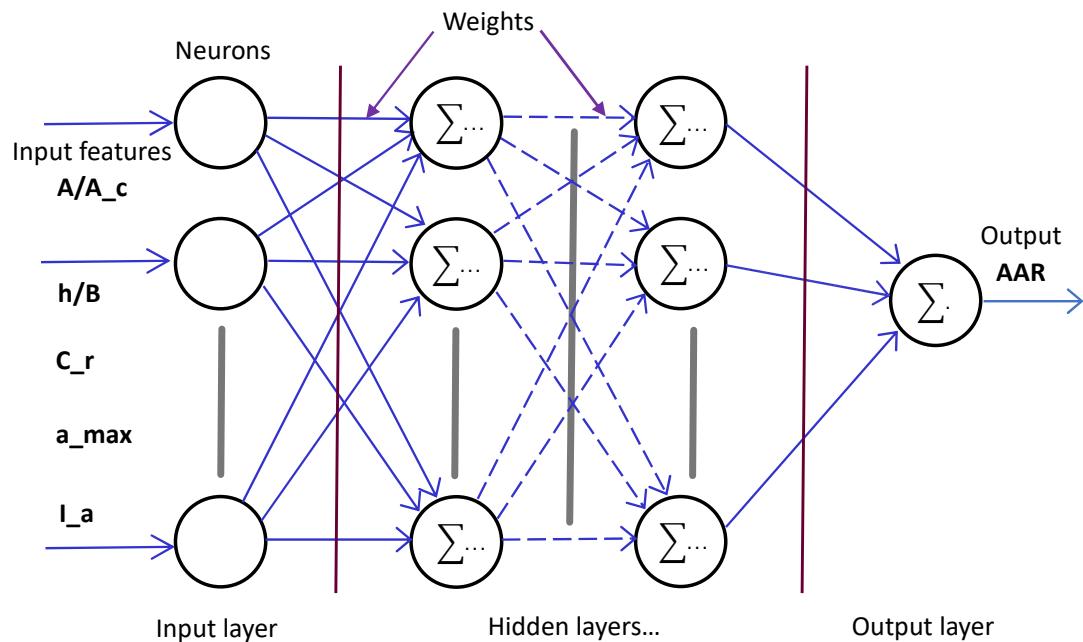


Figure 6. Schematic of the architecture of the multi-layer perceptron artificial neural network (ANN) regression model developed in this study.

In general, the relationship between the inputs and outputs of a neuron in the ANN model can be expressed by [39]:

$$y_i = g\left(\sum_{j=1}^k (W_{j,i}X_j) + b_i\right) \quad (5)$$

where y_i is the output of the i th neuron in any hidden layer, j goes from 1 to the number of neurons (k) in the previous layer, X are the outputs of neurons from the previous layer, W are the connection weights between neurons in the current layer and previous layer, b is the bias value, and $g()$ is an activation function. The rectified linear unit (ReLU) function is the activation function used in this study. For each training instance, the backpropagation algorithm first makes a prediction using the above relationship and measures the error using the mean squared error loss function. It then goes through each layer in reverse to measure the error contribution from each connection and adjusts the connection weights to reduce the error using the SGD algorithm. During testing, the ANN model simply propagates the input features through the network and calculates the prediction using the optimum connection weights determined in the training phase.

4. Results and Discussion

The performance of machine learning (ML) models developed in this study are evaluated mainly using mean absolute percentage error (MAPE) and mean absolute error (MAE) in predictions. MAPE is defined as:

$$\text{MAPE} = \frac{1}{n} \cdot \sum_{i=1}^n \left(\left| \frac{\tilde{y}_i - y_i}{y_i} \right| \right) \quad (6)$$

where y is the actual (experimental) value of AAR, \tilde{y} is the output value (AAR predicted by a particular model), and “ i ” goes from 1 up to the number of predictions (n). Note that MAE is a similar error measure that calculates the error in terms of the absolute difference between the predicted and experimental values of AAR (i.e., MAE does not normalize the difference between experimental and predicted values). A multivariate linear regression (MLR) ML model is also developed using the same dataset, the same input features, and supervised learning technique. It is used as the baseline model for comparison of performances of the nonlinear ML models developed in this study. All the ML models and deep learning ANN models in this study are developed in the Python programming platform using the implementations of the standard classes available in Scikit-Learn (<https://scikit-learn.org/stable/>, accessed on 1 June 2023) and TensorFlow and Keras (<https://keras.io/>, accessed on 1 June 2023) libraries of modules.

4.1. Initial Evaluation (Training and Testing) of Machine Learning Models

The experimental data and results obtained from the abovementioned series of experiments (140 tests) are split into two groups for initial training and testing of ML models using a 70–30% random split of data: training dataset (98 tests) and testing dataset (42 tests). After the initial training of ML models on the training dataset, the models are tested on previously unseen test dataset. Figure 7 presents the comparisons of ML model predictions with experimental results for AAR during the initial testing phase of the models for the KNN and SVR models along with the baseline MLR model. Note that the hyperparameters of the ML models are kept constant at their optimum values (described in Section 4.4): $k = 3$ in the KNN model, $C = 1.0$ and $\epsilon = 0.1$ in the SVR model. As seen in Figure 7, both the KNN and SVR models (MAPE = 0.17 and 0.16, respectively) outperform the baseline MLR model (MAPE = 0.21) in terms of accuracy of predictions.

Figure 8 presents the initial testing results of three DTR-based ensemble ML models (RFR, ABR and GBR) along with their MAPE values. As with the previous three models, the hyperparameters for all of these models are also kept at their optimum values: maximum depth of tree = 6 and number of trees in the ensemble = 100 for all three ensemble models. A single DTR model results in a MAPE of 0.17 during the initial testing phase (not shown in the figure). However, as can be seen from Figure 8, when 100 trees are combined together, all three DTR-based ensemble models (MAPE = 0.14 to 0.15) outperform other models presented in Figure 7 in terms of accuracy of predictions. In terms of consistency among different models, the MAPE of all five nonlinear, nonparametric ML models vary between 0.14 and 0.17 and the MAE values vary between 0.08 and 0.11. This shows excellent

consistency among the ML models developed for the problem considered in this study. For comparison, the MAPE and MAE resulting from the statistics-based simple liner regression model (SLR) presented in Figure 3 are 0.23 and 0.15, respectively. It should be noted that the SLR model uses the entire dataset for fitting a linear relationship and uses the same dataset for calculating the MAPE and MAE values. Despite that, it is interesting to note that the testing errors of the MLR model (MAPE = 0.21 and MAE = 0.12) are still slightly better than those of the statistics-based (non-ML) SLR model.

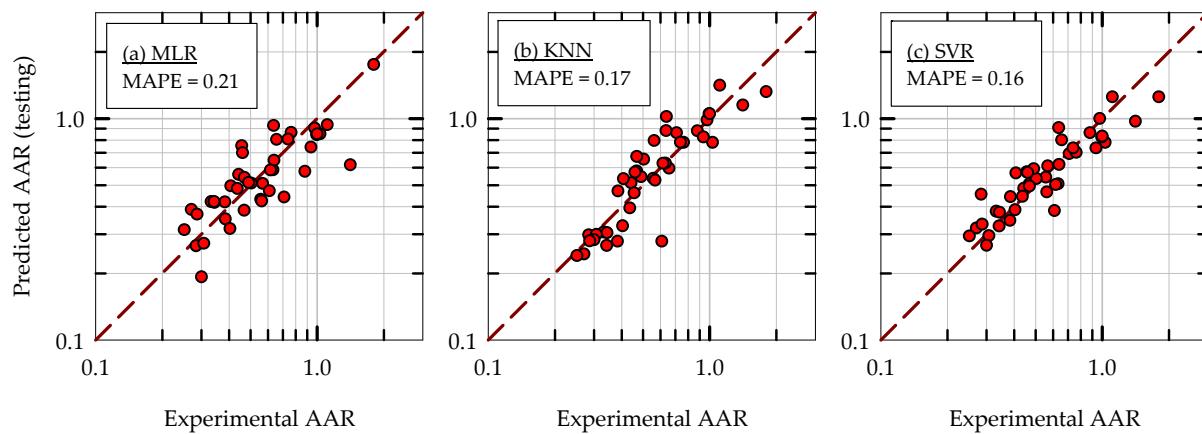


Figure 7. Comparisons of three ML model predictions with experimental results for the acceleration amplification ratio (AAR) during the initial testing phase of models: (a) MLR, (b) KNN and (c) SVR. Note: the dashed lines represent 1:1 lines.

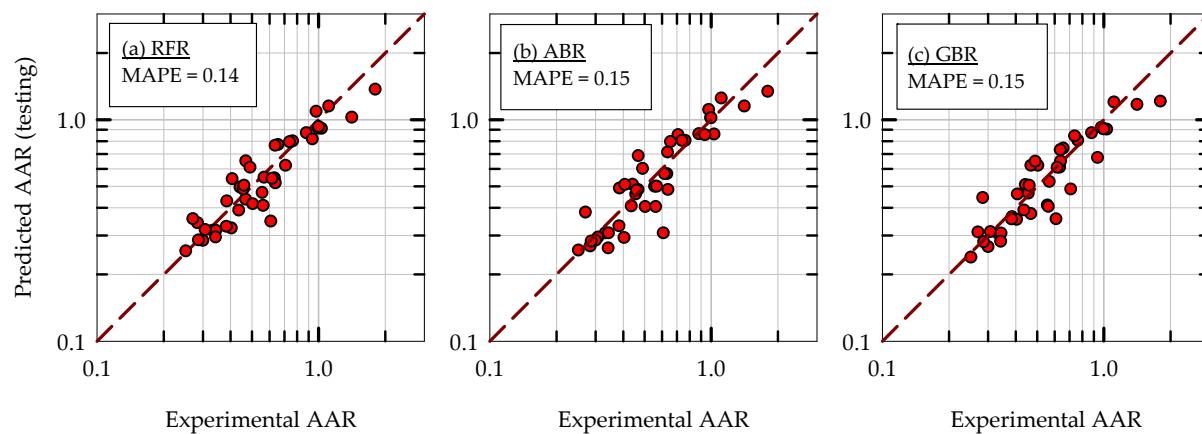


Figure 8. Comparisons of three DTR-based ML model predictions with experimental results for the acceleration amplification ratio (AAR) during the initial testing phase of the models: (a) RFR, (b) ABR and (c) GBR. Note: the dashed lines represent 1:1 lines.

4.2. Significance of Input Features

The significance of input features for the problem considered is quantified using the feature importance scores obtained from the RFR, ABR and GBR models. The feature importance scores are calculated based on how much the base decision-tree nodes that use an input feature reduce uncertainty in the data. The normalized feature importance scores of each input feature are presented in Figure 9 for three DTR-based ensemble ML models after the initial training phase. Figure 9 clearly shows that a_{max} has the highest normalized feature importance score (about 40% to 50%) in the predictions of AAR, followed by C_r (about 20%). This is consistent with the close relationship of AAR with a_{max} and C_r presented in Figure 3 and indicates that the AAR is more sensitive to a_{max} and C_r than the other parameters. The other three input features (A/A_c , h/B and I_a) have approximately 10% of feature importance scores each. These observations are consistent for all three DTR-based ensemble models and

confirm that none of the input features considered in this study are redundant. It should be noted that when the type of soil is included as an input feature to ML models, it results in feature importance scores of less than 5%, consistently for all three DTR-based models. In addition, it does not make any significant difference in ML model predictions when the type of soil is included as an input feature, and hence the type of soil is not included as an input feature in this study. However, the effect of soil type on rocking response of foundations is indirectly included in A/A_c and C_r through shear strength and the bearing capacity of the soil.

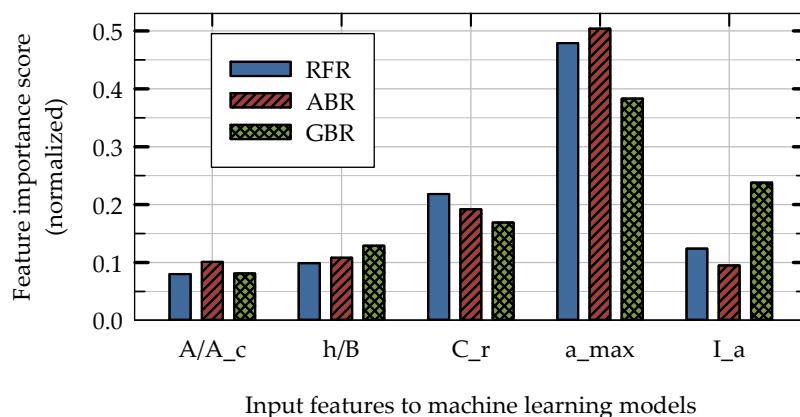


Figure 9. Results of feature importance scores based on the prediction of AAR obtained from three DTR-based ML models (RFR, ABR and GBR).

4.3. K-Fold Cross Validation Tests

In order to evaluate the performance of ML models on multiple, random pairs of training–testing datasets, the k-fold cross validation test is used. In a k-fold cross validation test, one fold of data is used for testing of ML models that are trained on $(k - 1)$ folds of data, and the process is repeated k times using every single fold as the test dataset once. In this study, five-fold cross validation tests with three repetitions (with different randomization of the data in each repetition) are carried out. This repeated cross validation yields 15 different sets of results for AAR and the corresponding MAPE and MAE values. Two types of repeated five-fold cross validations are carried out: (i) considering only the training dataset for hyperparameter turning of each ML model and (ii) considering the entire dataset for final evaluation and comparison of all the ML models developed in this study (in terms of accuracy of predictions and variance in prediction errors).

4.4. Hyperparameter Tuning of Machine Learning Models

The purpose of hyperparameter tuning is two-fold: (i) to determine the optimum values of hyperparameters of ML models for the problem considered and (ii) to ensure that the ML models do not overfit or underfit the training data. The key hyperparameters of ML models are optimized by minimizing the testing MAPE obtained using repeated five-fold cross validation tests on the training dataset. Figure 10 presents the results of hyperparameter tuning of ML models in the form of average testing of MAPE versus the variation of corresponding major hyperparameters of the models. Note that each MAPE value in Figure 10 is the average of 15 different MAPE values resulting from repeated five-fold cross validation tests.

Results presented in Figure 10a show that the average testing MAPE of the KNN model first decreases as the number of nearest neighbors (k) increases, indicating an increase in accuracy. However, when k increases further ($k > 3$), the accuracy of the model decreases. This indicates that the critical value of k is 3, in order to avoid overfitting ($k < 3$) or underfitting ($k > 3$) the training data. Based on this observation, the optimum value for k in the KNN algorithm is chosen to be 3. Similar to the KNN model, the average testing MAPE of the SVR model decreases as the penalty parameter C increases (Figure 10b), indicating that relatively smaller values of C would underfit the training data. Though it is not very apparent from

Figure 10c, relatively larger values of C would overfit the training data. Based on the results obtained and to be consistent with the previously developed ML models related to this topic (performance prediction of rocking foundations), the optimum value for C is chosen to be 1.0.

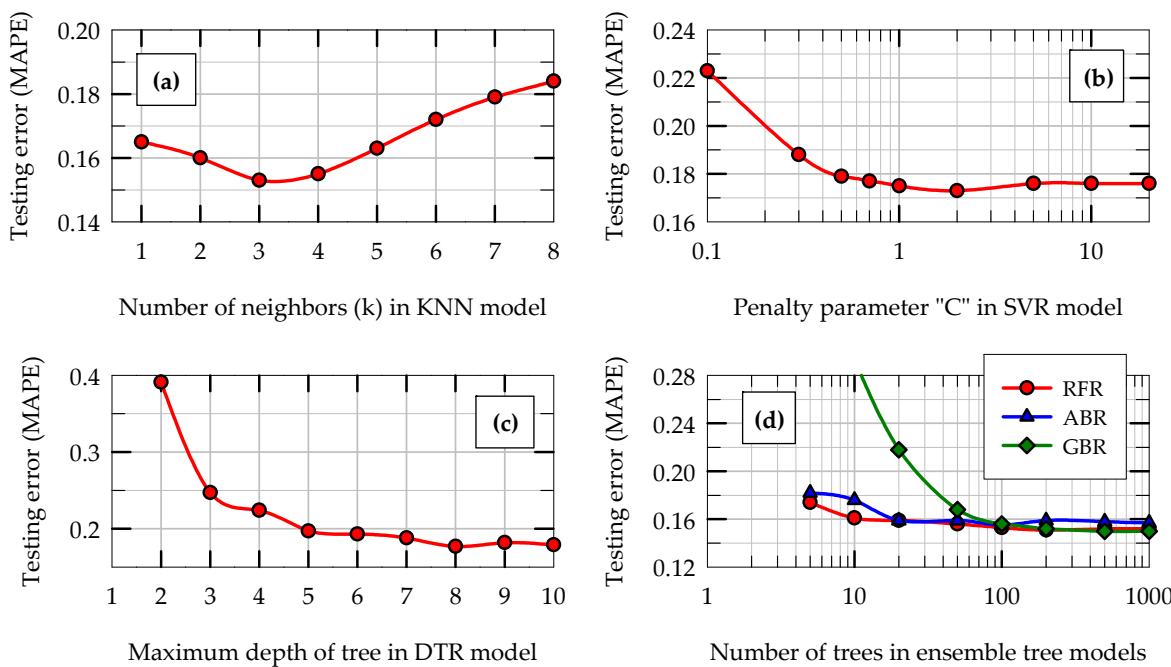


Figure 10. Variation of the average MAPE of machine learning models with their major hyperparameters from k-fold cross validation tests of models using training data for (a) KNN, (b) SVR, (c) DTR and (d) DTR-based ensemble models.

It is well known that deep DTR models tend to overfit the training data, while shallow DTR models tend to underfit the training data [39]. Based on the results shown in Figure 10c and to be consistent with the previously developed DTR-based models related to this topic, the optimum value for the maximum depth is set at 6 for base DTR models in the ensembles. Figure 10d shows that, for all three DTR-based ensemble models (RFR, ABR and GBR), the accuracy of the models increases as the number of trees increases (this is more apparent for the GBR model). The number of random features (maximum) to be considered is kept at 4 for the RFR model, and the learning rate is kept at 0.1 for both boosting models (ABR and GBR). When the number of trees in the ensembles increases beyond 100, the average testing MAPE of the models does not decrease any further. This indicates that the minimum number of trees required in DTR-based ensemble models is 100. This is remarkably consistent for all three DTR-based ensemble models. Table 1 summarizes the key hyperparameters chosen for five nonlinear ML models developed in this study.

Table 1. Optimum values chosen for major hyperparameters of machine learning models.

Machine Learning Model	Hyperparameter
k-nearest neighbors regression (KNN)	k = 3 weight = inverse distance
Support vector regression (SVR)	C = 1.0 epsilon = 0.1
Random forest regression (RFR)	mapping function = RBF ¹ max. depth = 6 max. features = 4 number of trees = 100
Boosting models (ABR and GBR)	max. depth = 6 learning rate = 0.1 number of trees = 100

¹ Radial basis function.

4.5. Initial Evaluation of ANN Models

Multiple, multi-layer perceptron, sequential ANN models, with different architectures (varying number of hidden layers and number of neurons in hidden layers) and hyperparameters, are developed and evaluated. The same training dataset and testing dataset are also used for the initial evaluation of ANN models, and the MAPE values of ANN models are calculated using the same procedure (same as described in Section 4.1). In addition to the testing error, the ANN models are also tested with the training data after the models are trained to compute the training error. The purpose of this exercise is to quantify how well the ANN models learn from the training data and their ability to generalize the patterns present in training data. The variation of predicted AAR with experimental results for AAR are presented in Figure 11a,b for the training phase and testing phase, respectively, for one particular ANN model.

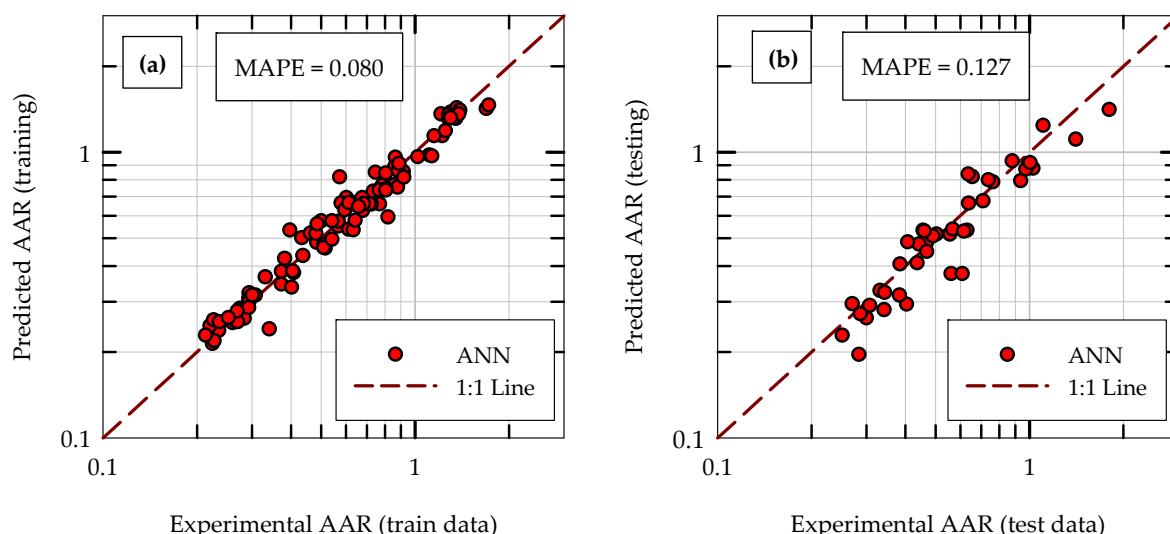


Figure 11. Comparisons of ANN model predictions with experimental results for AAR during initial evaluation of the model: (a) training phase and (b) testing phase.

The architecture of this particular ANN model consists of four hidden layers ($L = 4$) with forty neurons ($N = 40$) in each hidden layer. This is in addition to five neurons in the input layer (one for each input feature) and one neuron in the output layer (for output parameter, AAR). This particular set of hyperparameters turns out to be the optimum for the ANN model architecture for the problem considered (described in Section 4.6). Based on the comparison of predicted versus experimental AAR, with a MAPE of 0.08 and MAE of 0.053 during the training phase (Figure 11a), it is fair to say the ANN model extracts adequate information from data to build a reasonably good neural network structure during the training phase. The ANN model predictions during the initial testing phase are shown in Figure 11b and the resulting MAPE and MAE on test data are 0.127 and 0.082, respectively. This prediction accuracy places the ANN model superior to all other ML models developed in this study during the initial evaluation and testing phase.

4.6. Hyperparameter Tuning of the ANN Model

Similar to the other ML models, the key hyperparameters of the ANN model are optimized by minimizing the average MAPE values obtained from repeated five-fold cross validation tests (number of repeats = 3) carried out on the initial training dataset. The average values of testing MAPE of many different ANN models resulting from the cross validation tests are presented in Figure 12 (each data point represents the average of 15 different MAPE values). Multiple ANN models (with different architecture) are developed to find the optimum number of hidden layers (L) and number of neurons (N) in each

hidden layer (Figure 12a,b), while a fixed network architecture is used to tune number of epochs and the learning rate (LR) of the SGD algorithm (Figure 12c,d).

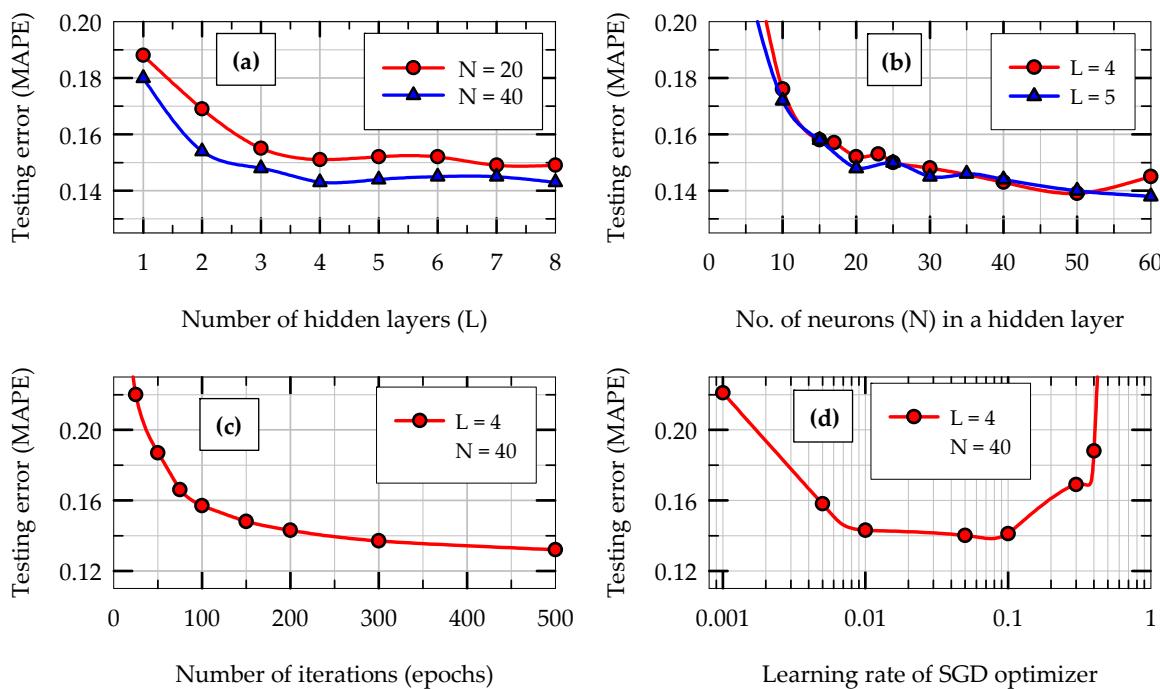


Figure 12. Results of hyperparameter tuning of the ANN model: Variation of average MAPE with (a) number of hidden layers, (b) number of neurons in each hidden layer, (c) number of iterations for each batch of training data, and (d) learning rate of SGD algorithm.

Results presented in Figure 12a show that as L increases, the error (average MAPE) in predictions decreases up to when $L = 4$. When L increases further, the ANN model seems to overfit the training data slightly (increase in testing error). A similar trend is observed for the number of neurons (N) used in each hidden layer (Figure 12b). Based on these observations, the combination of $L = 4$ and $N = 40$ is chosen as the optimum combination for the architecture of the ANN model for the problem considered. These observations are confirmed and verified independently by using grid search and random search algorithms going through multiple ANN model architectures with several possible combinations of L and N . As the number of iterations (epochs) increases, as expected, the average MAPE in predictions decreases (Figure 12c); however, once the number of iterations reaches around 200 to 300, no further significant improvement in MAPE is observed with the number of iterations. As for the learning rate (LR) of the SGD algorithm, the optimum learning rate is found to be between 0.01 and 0.1 (Figure 12d). The optimum values chosen for the number of iterations and the learning rate are 300 and 0.01, respectively. Table 2 summarizes the optimum values chosen for the key hyperparameters of the ANN model developed in this study.

Table 2. Optimum values chosen for major hyperparameters of the ANN model.

Hyperparameter of the ANN Model	Value
Number of hidden layers (L)	4
Number of neurons in each hidden layer (N)	40
Activation function	ReLU ¹
Optimizer	SGD ²
Learning rate	0.01
Batch size for training	2
Number of epochs	300

¹ Rectified linear unit function. ² Stochastic gradient descent.

4.7. Comparison of Overall Accuracy of Model Predictions and Variance in Prediction Error

Multiple k-fold cross validation tests ($k = 5$ and number of repeats = 3) are carried out considering the entire dataset to evaluate the overall performance (average testing MAPE and MAE, and the variance in testing MAPE and MAE) of all ML models. Note that for hyperparameter training, the k-fold cross validation tests are performed using the training dataset only, while this final k-fold cross validation test uses the entire dataset. Figure 13 presents the results of the MAPE of predictions of AAR obtained using six nonlinear machine learning and deep learning models (KNN, SVR, RFR, ABR, GBR, and ANN) along with the baseline MLR model. The hyperparameters of all models are kept constant as obtained from the hyperparameter tuning phase of each model. For each model, the testing MAPE results are plotted in the form of boxplots, showing the average MAPE, median MAPE, and the 10th, 25th, 75th and 90th percentile values of MAPE (obtained from 15 values of MAPE for each model).

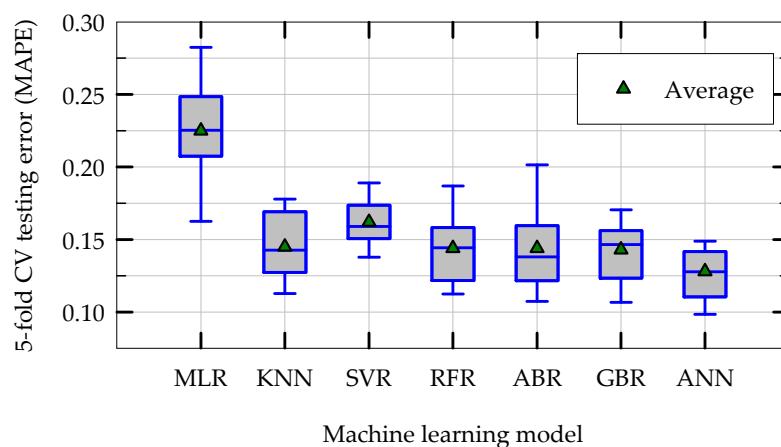


Figure 13. Boxplots of MAPE in the predictions of AAR of machine learning models during final five-fold cross validation tests of models.

The first observation from Figure 13 is that the average testing MAPE of all six nonlinear models are better (smaller) than that of the baseline MLR model. Among the six nonlinear models, except for the SVR model, the average testing MAPE of the models are smaller than 0.15, and the results for the average MAPE are remarkably consistent across different ML models. The ANN model has the smallest average MAPE value (0.128) and relatively lower variance in MAPE (0.05), indicating that the ANN model outperforms all other models developed in this study for the problem considered. Based on the overall average MAPE in predictions, the ANN model improves the prediction accuracy by 43% compared to the MLR model (MAPE of 0.128 versus 0.225). As the difference in overall model performance in terms of average accuracy among five nonlinear models is relatively small (average MAPE varies from 0.13 to 0.15), and if one prefers simpler ML models, KNN and all three DTR-based ensemble models are almost equally effective for the prediction of AAR. Figure 14 presents the results obtained from the same k-fold cross validation tests (same as the one presented in Figure 13), in the form of MAE in predictions of AAR. As can be observed from Figure 14, the results for testing MAE show a very similar trend that is observed for testing MAPE (Figure 13). Except for the SVR model, the overall average MAE of the other five nonlinear ML models varies between 0.083 and 0.092, once again indicating a remarkable consistency across different ML models. The overall average MAE in predictions of all six nonlinear ML models varies between 0.08 and 0.1, indicating that the maximum acceleration transmitted to structures supported by rocking foundations can be predicted within an average error limit of 8% to 10% of peak ground acceleration of the earthquake.

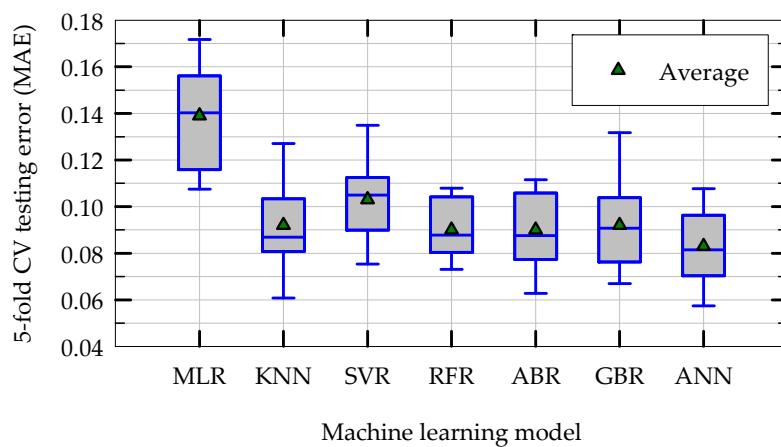


Figure 14. Boxplots of MAE in the predictions of AAR of machine learning models during final five-fold cross validation tests of models.

Table 3 summarizes the average MAPE and MAE of predictions of all seven ML models in repeated five-fold cross validation tests. Also included in Table 3 is the MAPE and MAE of the statistics-based simple linear regression (SLR) best fit model presented in Figure 3 (using the relationship between $\log(C_r/a_{max})$ and $\log(AAR)$). As can be seen from Table 3, from the statistics based SLR model to MLR machine learning model, the results do not seem to vary much. However, the six nonlinear ML models developed in this study for the prediction of AAR show a significant difference in accuracy. In order to compare the performance of the models using a different error criterion, which is not used in the training phase of models nor in the hyperparameter tuning, a third error measure is also considered. For this purpose, root mean squared error (RMSE), a commonly used error measure in machine learning, is selected. The last column of Table 3 presents the results obtained for average RMSE of predictions of all the models in repeated five-fold cross validation tests. As can be seen from Table 3, the trend in RMSE values is consistent with the trends observed in MAPE and MAE, and it leads to the same conclusion: among the six nonlinear ML models, the ANN model turns out to be the most accurate, the second most accurate model is the RFR, and it is followed by the ABR, GBR, and KNN models.

Table 3. Summary of average MAPE, MAE and RMSE (testing errors) of models in final five-fold cross validation tests.

Model	Ave. MAPE	Ave. MAE	Ave. RMSE
Simple linear regression (SLR) *	0.228	0.148	0.232
Multivariate linear regression (MLR)	0.225	0.139	0.185
Support vector regression (SVR)	0.162	0.103	0.145
k-nearest neighbors regression (KNN)	0.145	0.092	0.137
Random forest regression (RFR)	0.144	0.090	0.124
Adaptive boosting regression (ABR)	0.144	0.090	0.125
Gradient boosting regression (GBR)	0.143	0.092	0.133
Artificial neural network regression (ANN)	0.128	0.083	0.113

* Statistics-based (non-ML) model.

4.8. Parametric Sensitivity Analysis of Models

In order to study the sensitivity of ML model predictions to variations in input feature values, a parametric sensitivity analysis is carried out. For this exercise, the input feature values are systematically varied and are fed into the ML models. As a baseline case, all input feature values are kept at their mean values and the predicted AAR corresponding to this scenario is the most likely value (MLV) of prediction for a particular model. In addition, each input feature is varied to include two other values: mean minus standard deviation and mean plus standard deviation. The predictions of the ML models are obtained by this

method using these two extreme values for a certain input feature, while all other input feature values are kept at their mean value. As there are five input features, this method results in eleven combinations of input features. The results of this parametric sensitivity analysis are presented in Figure 15 for four models in the form of “tornado diagrams”. In the tornado diagrams presented in Figure 15, the x-axis represents the predicted AAR values by that particular model when the input feature values are varied (mean \pm standard deviation). Note that in a tornado diagram, the absolute difference between the prediction values corresponding to the two extreme values of an input feature is called the “swing”, and the input feature that has greatest swing is plotted at the top of the plot (the input features are plotted on the y-axis in descending order of their swing values). Also included in these figures are the most likely value (MLV) of predicted AAR (vertical dashed lines), when all the input features are kept at their mean values. Table 4 presents summary results of predicted AAR in parametric sensitive analysis (MLV, minimum and maximum) for all seven ML models.

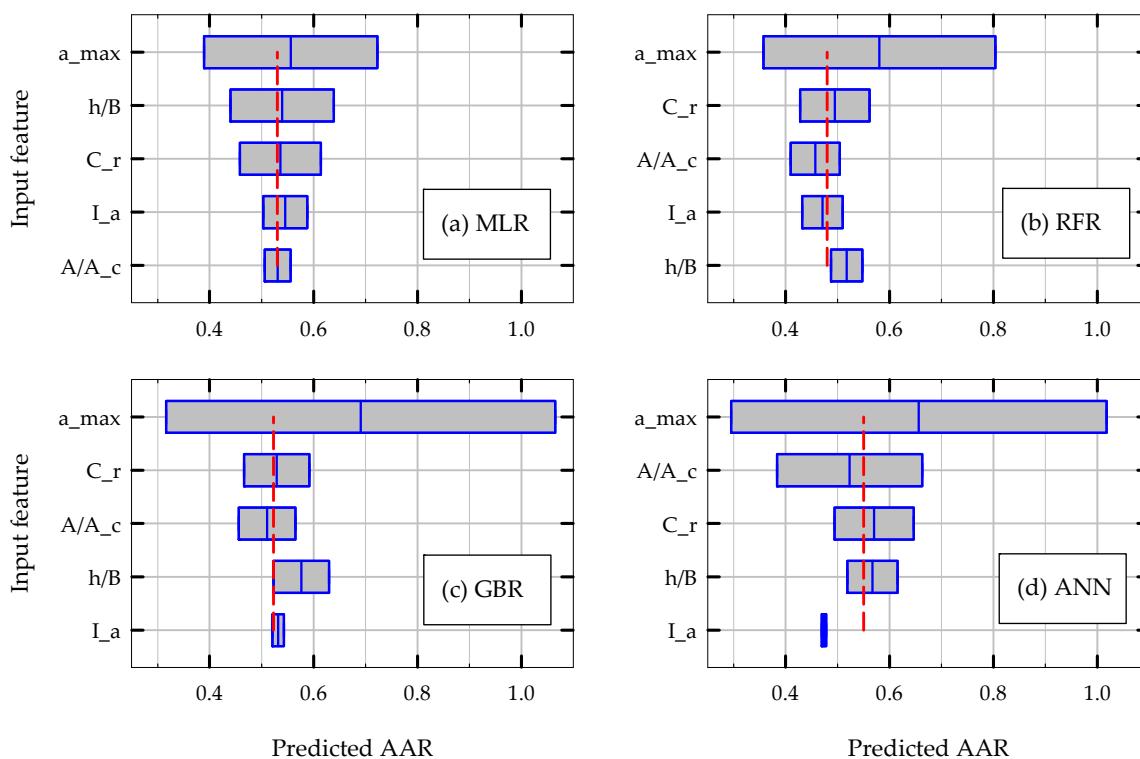


Figure 15. Results of parametric sensitivity analysis of ML models in the form of tornado diagrams when the input feature values are varied one at a time: (a) MLR, (b) RFR, (c) GBR and (d) ANN. Note: the dashed vertical lines correspond to the predicted AAR (most likely value) when all input features are at their mean values.

As the results presented in Figure 15 and Table 4 indicate, the predicted AAR is more sensitive to peak ground acceleration (a_{max}) than any other input feature for all the models (i.e., a_{max} produces the maximum swing in predicted AAR). Only the MLR model shows an almost symmetric response around the MLV in tornado diagrams, mostly because it is a linear ML model. The unsymmetric nature of the tornado diagrams of all nonlinear ML models indicates that the relationship between AAR and input features are highly nonlinear. For all six nonlinear ML models, about 45% to 75% of the variance in the prediction of AAR results from the variation in a_{max} (variance in this context is defined as half-swing divided by the most likely value of predicted AAR). Next to a_{max} , C_r and A/A_c have more effect on model predictions in general when compared to h/B and I_a . This trend is consistent with the experimental results plotted in Figure 3, where a_{max} and C_r are identified as the key variables that dictate AAR. This is also consistent with the results presented in Figure 9,

where two DTR-based ensemble models (RFR and ABR) identify a_{max} and C_r as the features with the highest and the second highest, respectively, feature important scores to predict AAR. It should also be noted that none of the ML model predictions are extremely high or extremely low when the input feature values are varied. This indicates that the ML models developed in this study do not tend to extrapolate the data beyond a reasonable range of AAR values.

Table 4. Summary results of predicted AAR in parametric sensitivity analysis of ML models when the input feature values are varied one at a time.

Model	MLV *	Minimum	Maximum
Multivariate linear regression (MLR)	0.530	0.389	0.723
Support vector regression (SVR)	0.586	0.332	1.046
k-nearest neighbors regression (KNN)	0.435	0.349	0.634
Random forest regression (RFR)	0.480	0.357	0.804
Adaptive boosting regression (ABR)	0.541	0.341	1.150
Gradient boosting regression (GBR)	0.523	0.316	1.065
Artificial neural network regression (ANN)	0.469	0.295	1.018

* Most likely value.

The interpretability of ML models is often thought to be challenging, as they are agnostic to the underlying scientific principles driving the physical mechanisms of the problem considered. However, apart from the ANN model, all other ML models developed in this study are based on simple, straight forward logic, and they are relatively easy to interpret (i.e., why the model predicts a certain value for AAR given the input feature values). The new data science paradigm of theory-guided machine learning combines the beneficial features of both mechanics-based models and ML models while minimizing or eliminating their adverse effects [3]. This concept forms the basis for future research on this topic.

5. Conclusions

Multiple machine learning (ML) models are developed to predict the maximum acceleration transferred to the center of gravity of structures founded on rocking shallow foundations during earthquake loading. Based on this study, the following major conclusions are drawn.

- Given the five input features representing the key properties of the rocking foundation and earthquake loading (A/A_c , h/B , C_r , a_{max} and I_a), the ML models presented in this paper can be used to predict the maximum acceleration transmitted to structures supported by rocking foundations with reasonable accuracy.
- Based on k-fold cross validation tests, the overall average MAPE in predictions of the KNN, RFR, ABR, GBR, and ANN models are all smaller than 0.145, with ANN being the most accurate and most consistent (MAPE = 0.128). For comparison, the MAPE of the MLR model and statistics based SLR model are around 0.23. This corresponds to an improvement in prediction accuracy of about 43%. Next to the ANN model, the second most accurate model is RFR, and it is followed by ABR, GBR, and KNN. This finding is also supported by another error measure criterion, namely, root mean squared error (RMSE) of model predictions.
- The overall average MAE in predictions of all six nonlinear ML models vary between 0.08 and 0.1, indicating that the maximum acceleration transferred to structures supported by rocking foundations can be predicted within an average error limit of 8% to 10% of the peak ground acceleration of the earthquake.
- Hyperparameter tuning is carried out to obtain the optimum values for hyperparameters and to ensure that the ML models presented in this paper do not overfit or underfit the training data. In terms of the architecture of the ANN model, a relatively simple network (only four hidden layers with 40 neurons in each layer) is found to be

the optimum and most efficient for the problem considered in terms of accuracy of predictions without overfitting the training data.

- Feature importance analysis using the RFR, ABR and GBR models reveals that the chosen five input features capture the maximum acceleration of structures (through AAR) supported by rocking foundations satisfactorily. Parametric sensitivity analysis of all ML models reveals that AAR is more sensitive to peak ground acceleration of the earthquake motion than to other input features.
- The ML models presented in this paper can be used with numerical simulation results as complementary measures in modeling of rocking foundations or can be combined with mechanics-based models using the emerging framework of theory-guided machine learning. This forms the basis for future research on this topic.

Funding: This research was funded by the US National Science Foundation (NSF) through award number CMMI-2138631.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study and more details about the machine learning models developed in this study are available on request from the corresponding author. The data are not publicly available due to ethical.

Conflicts of Interest: The author declares no conflict of interest.

Nomenclature

AAR	Acceleration amplification ratio
ABR	Adaptive boosting regression model
a_{max}	Peak ground acceleration of earthquake
ANN	Artificial neural network regression model
A/A_c	Critical contact area ratio of rocking foundation
C_r	Rocking coefficient of rocking system
GBR	Gradient boosting regression model
h/B	Slenderness ratio of rocking system
I_a	Arias intensity of earthquake
KNN	k-nearest neighbors regression model
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MLR	Multivariate linear regression model
R^2	Coefficient of determination
RFR	Random forest regression model
RMSE	Root mean squared error
SLR	Simple linear regression (non-ML) model
SVR	Support vector regression model

References

1. Bapir, B.; Abrahamczyk, L.; Wichtmann, T.; Prada-Sarmiento, L.F. Soil-structure interaction: A state-of-the-art review of modeling techniques and studies on seismic response of building structures. *Front. Built Environ.* **2023**, *9*, 1120351. [[CrossRef](#)]
2. Ebied, A.M. 35 years of AI in geotechnical engineering: State of the art. *Geotech. Geol. Eng.* **2021**, *39*, 637–690. [[CrossRef](#)]
3. Karpatne, A.; Atluri, G.; Faghmous, J.H.; Steinbach, M.; Banerjee, A.; Ganguly, A.; Shekhar, S.; Samatova, N.; Kumar, V. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2318–2331. [[CrossRef](#)]
4. Mozumder, R.A.; Laskar, A.I. Prediction of unconfined compressive strength of geopolymer-stabilized clayey soils using artificial neural network. *Comput. Geotech.* **2015**, *69*, 291–300. [[CrossRef](#)]
5. Pham, B.T.; Bui, D.T.; Prakash, I. Landslide susceptibility assessment using bagging ensemble based alternating decision trees, logistic regression and J48 decision trees methods: A comparative study. *Geotech. Geol. Eng.* **2017**, *35*, 2597–2611. [[CrossRef](#)]
6. Jeremiah, J.J.; Abbey, S.J.; Booth, C.A.; Kashyap, A. Results of application of artificial neural networks in predicting geo-mechanical properties of stabilized clays—A review. *Geotechnics* **2021**, *1*, 144–171. [[CrossRef](#)]

7. Amjad, M.; Ahmad, I.; Ahmad, M.; Wroblewski, P.; Kaminski, P.; Amjad, U. Prediction of pile bearing capacity using XGBoost algorithm: Modeling and performance evaluation. *Appl. Sci.* **2022**, *12*, 2126. [\[CrossRef\]](#)
8. Rateria, G.; Maurer, B.W. Evaluation and updating of Ishihara's (1985) model for liquefaction surface expression with insights from machine and deep learning. *Soils Found.* **2022**, *62*, 101131. [\[CrossRef\]](#)
9. Raja, M.N.A.; Abdoun, T.; El-Sekelly, W. Smart prediction of liquefaction-induced lateral spreading. *J. Rock Mech. Geotech. Eng.* **2023**, *in press*.
10. Gajan, S. Modeling of seismic energy dissipation of rocking foundations using nonparametric machine learning algorithms. *Geotechnics* **2021**, *2*, 534–557. [\[CrossRef\]](#)
11. Gajan, S. Data-driven modeling of peak rotation and tipping-over stability of rocking shallow foundations using machine learning algorithms. *Geotechnics* **2022**, *2*, 781–801. [\[CrossRef\]](#)
12. Huang, B.; Lu, W. Evaluation of the floor acceleration amplification demand of instrumented buildings. *Adv. Civ. Eng.* **2021**, *2021*, 7612101. [\[CrossRef\]](#)
13. Wang, T.; Shang, Q.; Li, J. Seismic force demands on acceleration-sensitive nonstructural components: A state-of the-art review. *Earthq. Eng. Eng. Vib.* **2021**, *20*, 39–62. [\[CrossRef\]](#)
14. Rutenberg, A. Seismic shear forces on RC walls: Review and bibliography. *Bull. Earthq. Eng.* **2013**, *11*, 1727–1751. [\[CrossRef\]](#)
15. Clvi, P.M.; Sullivan, T.J. Estimating floor spectra in multiple degree of freedom systems. *Earthq. Struct.* **2014**, *7*, 17–38. [\[CrossRef\]](#)
16. Vukobratovic, V.; Fajfar, P. Code-oriented floor acceleration spectra for building structures. *Bull. Earthq. Eng.* **2017**, *15*, 3013–3026. [\[CrossRef\]](#)
17. Perrone, D.; Brunesi, E.; Filiatrault, A.; Nascimbene, R. Probabilistic estimation of floor response spectra in masonry infilled reinforced concrete building portfolio. *Eng. Struct.* **2020**, *202*, 109842. [\[CrossRef\]](#)
18. Gajan, S.; Kutter, B.L. Capacity, settlement, and energy dissipation of shallow footings subjected to rocking. *J. Geot. Geoenviron. Eng.* **2008**, *134*, 1129–1141. [\[CrossRef\]](#)
19. Paolucci, R.; Shirato, M.; Yilmaz, M.T. Seismic behavior of shallow foundations: Shaking table experiments versus numerical modeling. *Earthq. Eng. Struct. Dyn.* **2008**, *37*, 577–595. [\[CrossRef\]](#)
20. Gelagoti, F.; Kourkoulis, R.; Anastopoulos, I.; Gazetas, G. Rocking isolation of low-rise frame structures founded on isolated footings. *Earthq. Eng. Struct. Dyn.* **2012**, *41*, 1177–1197. [\[CrossRef\]](#)
21. Pelekis, I.; Madabhushi, G.; DeJong, M. Seismic performance of buildings with structural and foundation rocking in centrifuge testing. *Earthq. Eng. Struct. Dyn.* **2018**, *47*, 2390–2409. [\[CrossRef\]](#)
22. Khosravi, M.; Boulanger, R.W.; Wilson, D.W.; Olgun, C.G.; Shao, L.; Tamura, S. Stress transfer from rocking shallow foundations on soil-cement reinforced clay. *Soils Found.* **2019**, *59*, 966–981. [\[CrossRef\]](#)
23. Irani, A.E.; Bonab, M.H.; Sarand, F.B.; Katebi, H. Overall improvement of seismic resilience by rocking foundation and trade-off implications. *Int. J. Geosynth. Ground Eng.* **2023**, *9*, 40. [\[CrossRef\]](#)
24. Anastopoulos, I.; Gazetas, G.; Loli, M.; Apostolou, M.; Gerolymos, N. Soil failure can be used for seismic protection of structures. *Bull. Earthq. Eng.* **2010**, *8*, 309–326. [\[CrossRef\]](#)
25. Pecker, A.; Paolucci, R.; Chatzigos, C.; Correia, A.A.; Figini, R. The role of non-linear dynamic soil-foundation interaction on the seismic response of structures. *Bull. Earthq. Eng.* **2014**, *12*, 1157–1176. [\[CrossRef\]](#)
26. Gajan, S.; Soundararajan, S.; Yang, M.; Akchurin, D. Effects of rocking coefficient and critical contact area ratio on the performance of rocking foundations from centrifuge and shake table experimental results. *Soil Dyn. Earthq. Eng.* **2021**, *141*, 106502. [\[CrossRef\]](#)
27. Gajan, S.; Raychowdhury, P.; Hutchinson, T.C.; Kutter, B.L.; Stewart, J.P. Application and validation of practical tools for nonlinear soil-foundation interaction analysis. *Earthq. Spectra* **2010**, *26*, 119–129. [\[CrossRef\]](#)
28. Hamidpour, S.; Shakib, H.; Paolucci, R.; Correia, A.A.; Soltani, M. Empirical models for the nonlinear rocking response of shallow foundations. *Bull. Earthq. Eng.* **2022**, *20*, 8099–8122. [\[CrossRef\]](#)
29. Deng, L.; Kutter, B.L.; Kunnath, S.K. Centrifuge modeling of bridge systems designed for rocking foundations. *J. Geot. Geoenviron. Eng.* **2012**, *138*, 335–344. [\[CrossRef\]](#)
30. Gavras, A.G.; Kutter, B.L.; Hakhamaneshi, M.; Gajan, S.; Tsatsis, A.; Sharma, K.; Kouno, T.; Deng, L.; Anastopoulos, I.; Gazetas, G. Database of rocking shallow foundation performance: Dynamic shaking. *Earthq. Spectra* **2020**, *36*, 960–982. [\[CrossRef\]](#)
31. Deng, L.; Kutter, B.L. Characterization of rocking shallow foundations using centrifuge model tests. *Earthq. Eng. Struct. Dyn.* **2012**, *41*, 1043–1060. [\[CrossRef\]](#)
32. Hakhamaneshi, M.; Kutter, B.L.; Deng, L.; Hutchinson, T.C.; Liu, W. New findings from centrifuge modeling of rocking shallow foundations in clayey ground. In Proceedings of the Geo-Congress 2012, Oakland, CA, USA, 25–29 March 2012.
33. Drosos, V.; Georgarakos, T.; Loli, M.; Anastopoulos, I.; Zarzouras, O.; Gazetas, G. Soil-foundation-structure interaction with mobilization of bearing capacity: Experimental study on sand. *J. Geot. Geoenviron. Eng.* **2012**, *138*, 1369–1386. [\[CrossRef\]](#)
34. Anastopoulos, I.; Loli, M.; Georgarakos, T.; Drosos, V. Shaking table testing of rocking—Isolated bridge pier on sand. *J. Earthq. Eng.* **2013**, *17*, 1–32. [\[CrossRef\]](#)
35. Antonellis, G.; Gavras, A.G.; Panagiotou, M.; Kutter, B.L.; Guerrini, G.; Sander, A.; Fox, P.J. Shake table test of large-scale bridge columns supported on rocking shallow foundations. *J. Geot. Geoenviron. Eng.* **2015**, *141*, 04015009. [\[CrossRef\]](#)
36. Tsatsis, A.; Anastopoulos, I. Performance of rocking systems on shallow improved sand: Shaking table testing. *Front. Built Environ.* **2015**, *1*, 00009. [\[CrossRef\]](#)

37. American Society of Civil Engineers (ASCE). *Minimum Design loads for Buildings and Other Structures*; SEI/ASCE 7-16; American Society of Civil Engineers (ASCE): Reston, VA, USA, 2017.
38. Building Seismic Safety Council (BSSC). *Recommended Provisions for the Development of Seismic Regulations for New Buildings and Structures*; National Earthquake Hazard Reduction Program (NEHRP): Washington, DC, USA, 2015.
39. Geron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools and Techniques to Build Intelligent Systems*, 2nd ed.; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2019.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.