Modeling of Rocking Induced Permanent Settlement of Shallow Foundations Using Machine Learning Algorithms

Sivapalan Gajan, Ph.D., M.ASCE[1]

[1]Associate Professor, College of Engineering, SUNY Polytechnic Institute, Utica, NY.
Email: gajans@sunypoly.edu

ABSTRACT

The objective of this study is to develop data-driven predictive models for permanent settlement of rocking shallow foundations during seismic loading using multiple machine learning algorithms and supervised learning technique. Data from a rocking foundation database consisting of dynamic base shaking experiments conducted on centrifuges and shaking tables have been used for the development of k-nearest neighbors regression, support vector regression, and random forest regression models. Based on repeated k-fold cross validation tests of models and mean absolute percentage errors in their predictions, it is found that all three models perform better than a baseline multivariate linear regression model in terms of accuracy and variance in predictions. The average mean absolute errors in predictions of all three models are around 0.005 to 0.006, indicating that the rocking induced permanent settlement can be predicted within an average error limit of 0.5% to 0.6% of the width of the footing.

INTRODUCTION

Recent research findings reveal that properly designed shallow foundations, with controlled rocking, possess many desirable characteristics such as beneficial seismic energy dissipation and reduced force and ductility demands transmitted to the structures (e.g., Gavras et al. 2020 and Gajan et al. 2021). However, the possibility of excessive settlement and rotation of foundation and the challenges associated with accurate prediction of foundation deformations are among primary concerns that hinder the use of foundation rocking as a designed mechanism for improving the seismic performance of structures. The accurate estimation of rocking induced permanent settlement of shallow foundations poses significant challenges due to complex, coupled material and geometrical nonlinearities associated with the soil-foundation system behavior during rocking (yielding of soil and uplift of footing). As globally available experimental databases become increasingly common, machine learning algorithms in predictive modeling have become efficient in many fields (e.g., Geron 2019 and Deitel and Deitel 2020). Models based on machine learning algorithms have the ability to learn directly from experimental data and generalize experimental behavior, capture the effects and propagation of uncertainties, and hence can be used in combination with mechanics-based models as complementary measures in practical applications.

The objective of this study is to develop data-driven predictive models for permanent settlement of rocking shallow foundations during seismic loading using multiple machine learning algorithms and supervised learning technique. Data from a rocking foundations database, consisting of 140 dynamic base shaking experiments conducted on centrifuges and shaking tables in the US and Greece, are used for training and testing of machine learning models. Primarily, three nonlinear and nonparametric machine learning algorithms are

considered in this study: distance-weighted k-nearest neighbors regression (KNN), support vector regression (SVR), and random forest regression (RFR). The input features to machine learning algorithms include critical contact area ratio of foundation (related to the factor of safety for bearing capacity during rocking), slenderness ratio and rocking coefficient of rocking system, peak ground acceleration and Arias intensity of earthquake motion, and a categorical binary feature that separates sandy soil foundations from clayey soil foundations.

DATA MINING AND DATA PREPARATION

Database. The results obtained from five series of centrifuge experiments and four series of shake table experiments (altogether 140 individual experiments) have been utilized in this study. The centrifuge experiments were conducted in University of California at Davis (Gajan and Kutter 2008, Deng et al. 2012, Deng and Kutter 2012, and Hakhamaneshi et al. 2012) and the shake table experiments were conducted in University of California at San Diego (Antonellis et al. 2015) and the National Technical University of Athens in Greece (Drosos et al. 2012, Anastasopoulos et al. 2013, and Tsatsis and Anastasopoulos 2015). Details of these experiments, including types of soils, foundations, structures, and ground motions, number of shaking events, raw data, and meta-data, are available in a database (Gavras et al. 2020). A summary of processed data from these experiments in terms of meaningful engineering parameters and the relationships among them are published in Gajan et al. (2021).

Input features. Input features to machine learning algorithms considered in this study include three rocking system parameters (critical contact area ratio (A/A_c), slenderness ratio (h/B), and rocking coefficient (C_r)), two earthquake ground motion parameters (peak ground acceleration (a_{max}) and Arias intensity of ground motion (I_a)), and a binary feature (called Type) that separates foundations on sandy soils from clayey soils. A/A_c is conceptually a factor of safety for rocking with respect to vertical loading (where A is the total base area of the footing when in full contact with the soil and A_c is the minimum footing contact area required to support the applied vertical load, V). The moment capacity (M_{ult}) of a rocking foundation has been shown to correlate with A/A_c through the following relationship (Gajan and Kutter 2008),

$$M_{ult} = \frac{V \cdot B}{2} \cdot \left[1 - \frac{A_c}{A}\right]$$

where, B is the width of the footing in the direction of shaking. The rocking coefficient of a soil-foundation-structure system, C_r, is the ratio of ultimate rocking moment capacity of the foundation to the weight (vertical load on foundation) of the structure normalized by the effective height (h) of the structure (Deng et al. 2012),

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

These six input features have been selected based on their close relationships with rocking induced settlement in experiments presented in Gajan et al. (2021). The selection of input features is also justified by "feature importance" values obtained from random forest regression model (presented later). The key statistical values of all six input features are presented in Table 1. As can be seen from Table 1, experimental results utilized in this study cover a wide range of rocking system capacity parameters and ground motion demand parameters.

Table 1. Details of input features used in machine learning algorithms

Input feature	A/A_c	h/B	C_r	a_{max} (g)	I_a (m/s)	Type*
Range of values	1.9 - 17.1	1.2 - 2.83	0.08 - 0.36	0.04 - 1.28	0.03 - 26.4	0 or 1
Mean value	8.17	1.89	0.24	0.43	2.31	n/a
Standard deviation	4.27	0.53	0.08	0.26	4.37	n/a
*Type: Sand or clay binary feature (0 or 1)						

Performance parameter. Rocking induced total permanent settlement (S) beneath the foundation is the performance parameter (prediction parameter for machine learning models) considered in this study and is obtained from settlement time-history of the base center point of the footing at the end of the shake for each experiment. Settlement is normalized by the width of the footing to obtain a nondimensional parameter called normalized settlement (NS = S/B) and to make comparisons meaningful across different experiments. The data from the database for 140 individual experiments have been processed to obtain the variation of NS due to foundation rocking with I_a of earthquake for different clusters (C_r) of rocking systems and for sandy soils and clays. The experimental data utilized in this study is shown in Figure 1 (Gajan et al. 2021). As can be seen from Figure 1, though NS seems to have some correlations with different parameters such as I_a, C_r, and sand and clay foundations, the variation in the data is so significant that simple, commonly used statistical models cannot be used to correlate NS with input features with reasonable accuracy.

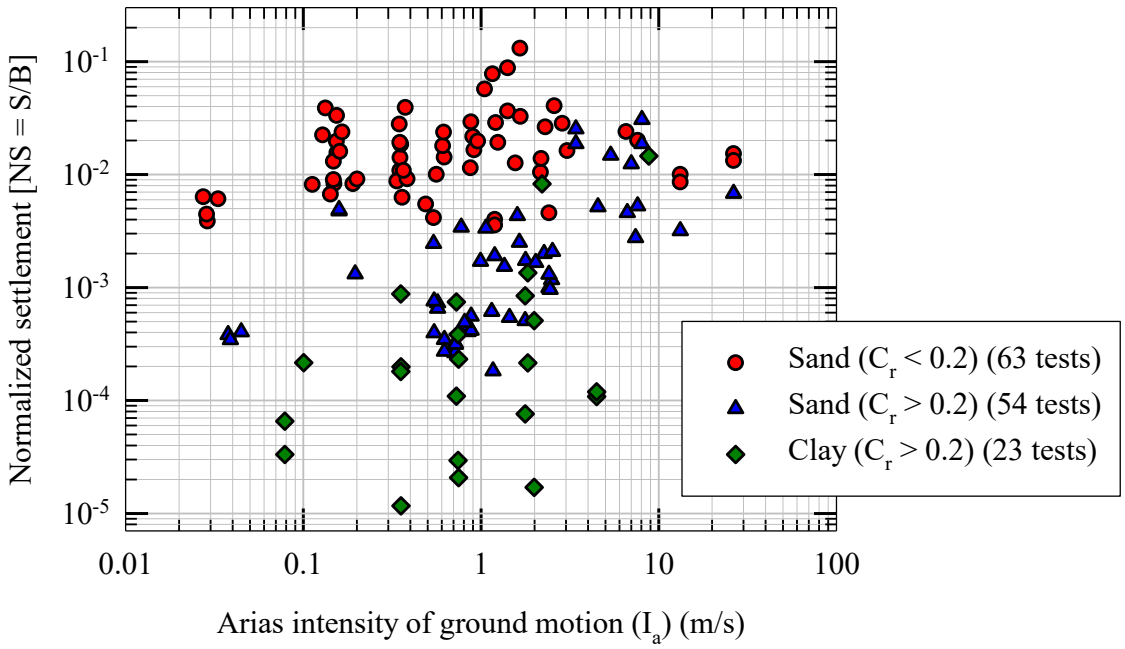

Figure 1. Variation of normalized permanent settlement of foundation with Arias intensity of earthquake and rocking coefficient of foundation (Gajan et al. 2021)

Feature transformation and normalization. As can be seen from Figure 1, the variation (numerical range) in NS and I_a is relatively high, and hence the data is plotted in log – log space.

For this reason, these two parameters (NS and I_a) are transformed to logarithmic values (base 10) before training and testing phases of machine learning models. In addition, in order to make reliable predictions using models developed by different machine learning algorithms, all the input feature data are normalized so that each input feature value varies between 0.0 and 1.0.

MACHINE LEARNING (ML) ALGORITHMS

K-nearest neighbors regression (KNN). The inductive bias of KNN algorithm is based on the assumption that when input data points are plotted (as vectors) in multi-dimensional input feature space, the data points that lie closer together share similar properties (i.e., similar output or prediction values). If there are hidden relationships between multiple input features and the prediction variable (NS), the KNN model is able to learn those relationships through training data. In this study the Euclidean distance measure in n-dimensional space is used to calculate the distance between two data instances (n = 6 in this study; one for each input feature). In KNN, the entire training data is stored in the model during training phase. When a test data instance is fed as input, the KNN algorithm runs through the entire training dataset and finds k number of training instances that are closer to the test instance. The distance-weighted KNN regression model weighs the output values by the inverse of their distance from the test data point to its nearest neighbors and produces an output that is the weighted average of the output values of the nearest neighbors (i.e. closer neighbors of a test data point will have a greater influence on the output than neighbors that are further away). The optimum value of k of KNN model is found to be in between 3 and 6 for the problem considered and k = 3 is used for all cases presented in this paper, except in hyperparameter tuning phase of models (presented later).

Support vector regression (SVR). The objective of original support vector machine (SVM) algorithm for classification is to find a hyperplane in an n-dimensional input space that distinctly separates the data points. The margin (ϵ) is the distance from the hyperplane to the closest data point (support vector) in n-dimensional space. The major difference between SVM and SVR is that whereas SVM classification algorithm tries to fit the largest possible margin between different classes while limiting margin violations of data points, SVR algorithm tries to fit as many training data instances as possible within the margin while limiting margin violations of data points. The inductive bias of SVR algorithm is that it assumes that the data points that have distinct characteristics tend to be separated by wide margins. The major parameter of SVR algorithm is the kernel function that is used to map (transform) the data instances (input data) into n-dimensional space. The radial basis function (RBF) kernel is used in this study as it is the most widely used kernel in SVR and it has the flexibility of dealing with nonlinear data (Geron 2019). Since complex, nonlinear data cannot be accommodated perfectly by a hyperplane and a margin, a wiggle room is allowed for the margin. A hyperparameter called the cost parameter or penalty parameter (C) defines the magnitude of this wiggle room across all dimensions in input space. The optimum value of C of SVR model is found to be 20 for the problem considered and C = 20 (with ϵ = 0.1) is used for all cases presented in this paper, except in hyperparameter tuning phase (discussed further in hyperparameter tuning section).

Random forest regression (RFR). The decision tree regression (DTR) algorithm uses supervised learning technique to build a tree-like data structure by employing a top-down, greedy search through the space of possible branches using information gain as a measure of reduction in uncertainty in training data (reduction in entropy in data). A random forest regression (RFR) model is an ensemble model that fits multiple base DTR models each on random subsets of the

original training dataset and using random input features, and then aggregate their individual predictions on testing dataset by averaging to form a final prediction. The major hyperparameter of DTR model is the maximum depth of the tree, and the optimum value for maximum depth of the tree is found to be 6 for the data analyzed (presented later). In RFR model, each individual base DTR model may have a higher bias than if it were trained on the original training dataset, but introduction of randomness and aggregation reduces both bias and variance. In other words, the introduction of randomness in input training data (random subset of training data, random input features and random size individual trees) reduces the uncertainty in final predictions on test data. For optimum performance of RFR model, the maximum number of random input features to consider when considering a split is limited to 2 (a hyperparameter) (randomly chosen 2 features out of total 6 features) and the optimum value for maximum number of trees in the ensemble is found to be 100 (described further in hyperparameter tuning section).

RESULTS AND DISCUSSION

Initial evaluation of ML models. Altogether 140 experimental data (instances) are extracted from the above-mentioned database and are randomly split into two groups for the purpose of initial training (supervised learning) and testing (validation) of ML models. A 70%– 30% split is used to randomly create a training dataset (98 instances) and a testing dataset (42 instances) using the train-test-split function available in "scikit-learn" library of modules in Python (https://scikit-learn.org/stable/). In addition to the three nonlinear ML models described above, for the purpose of comparisons, a multivariate linear regression (MLR) ML model is also developed using the same input features and supervised learning technique. As MLR model is a linear, parametric model, it is used as the baseline model for comparison of model performances. All four ML models are first trained using the same training dataset (98 instances) and then tested using previously unseen testing dataset (42 instances).

The variation of predicted NS with their corresponding experimental NS values on top of 1:1 prediction-to-experiment comparison lines during initial testing phase are presented in Figure 2 for all four ML models. Also included in Figure 2 are the mean absolute errors (MAE) and mean absolute percentage errors (MAPE) in predictions of each model. As can be seen from Figure 2, the three nonlinear, nonparametric ML models developed in this study (KNN, SVR, and RFR) consistently outperform the linear, parametric baseline MLR model in terms of MAE and MAPE in predictions. For example, the accuracy of predictions in terms of MAPE of SVR, KNN and RFR models is improved by about 27% to 36% when compared to that of MLR model (0.7), while the accuracy of all three nonlinear models in terms of MAE is improved by 20% to 30% when compared to that of MLR model (0.01). The predictions of SVR model show more variation (outliers), especially for relatively smaller values of settlement. However, as will be seen later, the SVR model performs better (increased accuracy and reduced variance) when combined with KNN model using stacked generalization as a stacked ensemble model. The results of this stacked model is presented in the summary plots in the last section.

Importance of input features. The implementation of RFR model in scikit-learn (in Python) has a function called "feature importances", which outputs relative importance of each input feature. This function measures a feature's importance by looking at how much the tree nodes that use that feature reduce uncertainty in data on average. The function computes this score for each feature after training, then it normalizes the results so that the sum of all feature importances is equal to 1.0. To investigate feature importance, twenty different random forest

models are developed by randomly selecting the training data (using the random state variable) and the feature importance values are obtained for each random forest. The average values of the feature importance along with their standard deviations (as error bars) are presented in Figure 3. As can be seen from Figure 3, rocking coefficient (C_r) has the highest score for feature importance, followed by h/B and A/A_c. This indicates that rocking system capacity parameters contribute more to model predictions than the earthquake intensity parameters and type of soil (in other words, the settlement is more sensitive to rocking foundation capacity parameters than to earthquake demand parameters). The other three input features also have feature importance scores greater than 8% to 13% each, indicating that none of the input features are redundant.

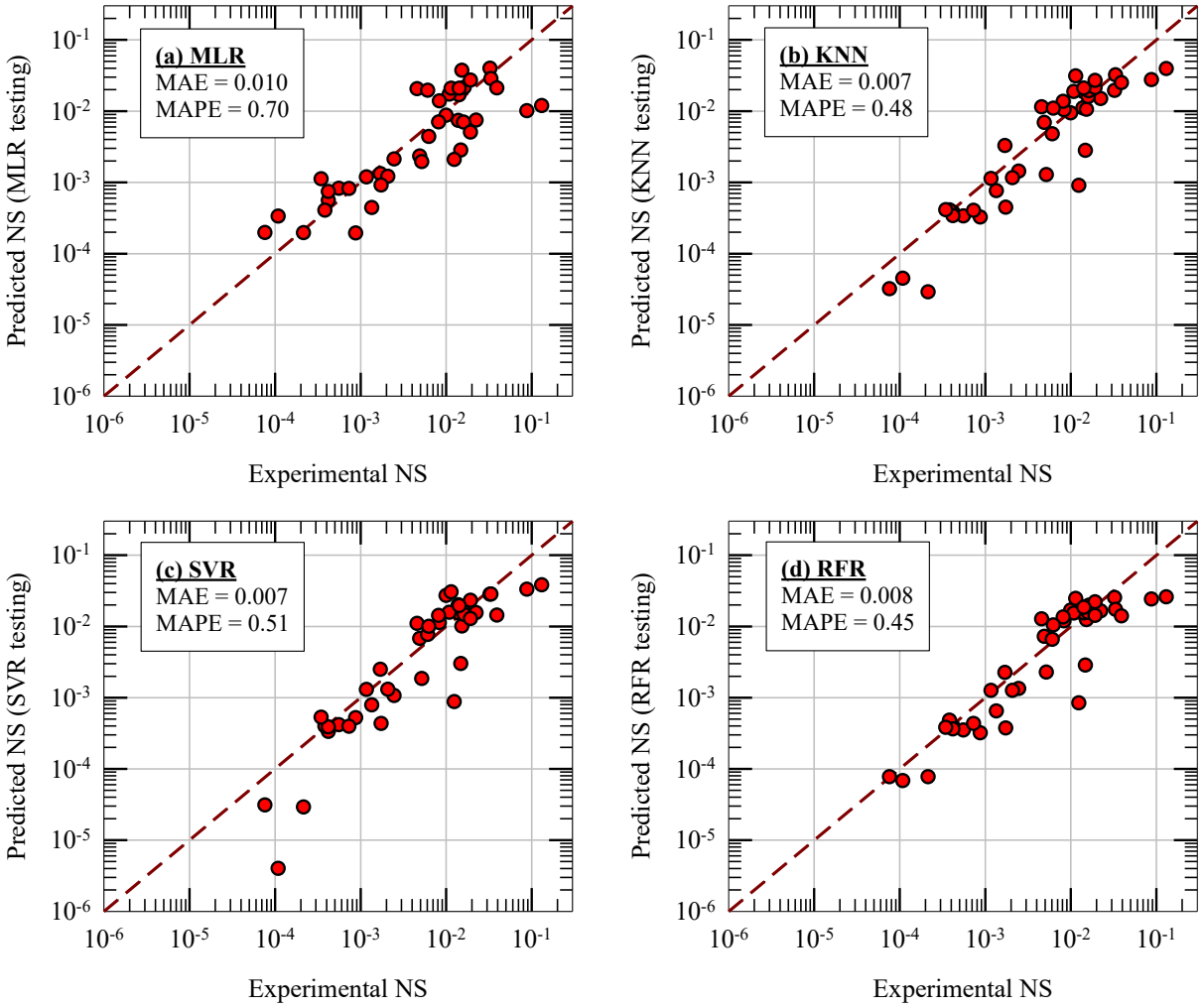

Figure 2. Comparisons of model predictions of normalized settlement (NS) with experimental results for different ML models during initial testing phase

Repeated k-fold cross validation of ML models. The k-fold cross validation is a technique for evaluating the performance of ML models on multiple, different train-test splits of dataset. It provides a measure of how good the model performance is both in terms of accuracy and variance (commonly referred to as bias-variance trade-off). In this study, the dataset is randomly shuffled and split into different groups of train-test sets for 5-fold cross validation (k = 5). In a 5-

fold cross validation, four folds of data are used for training of ML models and one fold is used for testing, and this process is repeated 5 times. On top of this, the whole process is repeated 3 times with different randomization of the data in each repetition (repeated k-fold cross validation). With this setup, for each model, the repeated 5-fold cross validation tests result in 15 sets of predictions and 15 different values for testing MAE and testing MAPE.

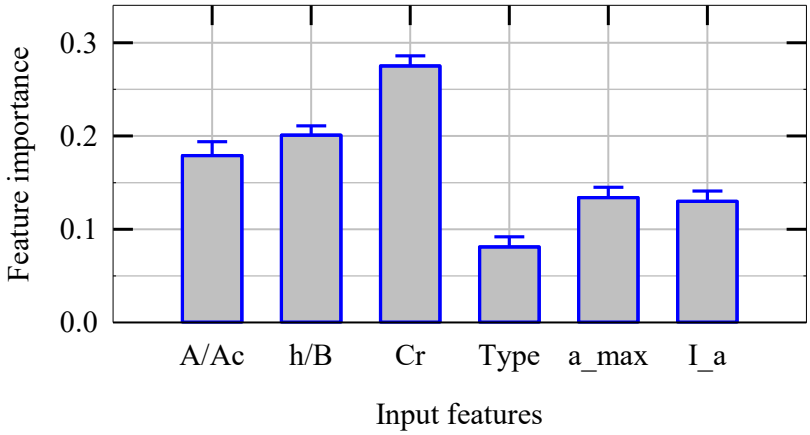

Figure 3. Normalized feature importance scores obtained from multiple RFR models

Hyperparameter tuning. For each ML model, the key hyperparameters are tuned using the average MAPE values obtained in repeated 5-fold cross validation tests performed on the initial training dataset (98 instances). It should be noted that for final evaluation of models (next section), the entire dataset (140 instances) are used in 5-fold cross validation. Hyperparameter tuning is carried out (i) to investigate the sensitivity of model predictions to their hyperparameters, (ii) to determine the optimum values of these parameters for the problem considered and data analyzed, and (iii) to ensure that the models neither overfit nor underfit the training data. The number of nearest neighbors (k), the cost or penalty parameter (C), the maximum depth of the tree, and the number of trees in the forest are varied for KNN, SVR, DTR, and RFR models, respectively, and the corresponding MAPE values of predictions are calculated. The results are presented using the average values of MAPE obtained in repeated 5-fold cross validations of all four nonparametric ML models in Figure 4.

The testing MAPE of KNN model first decreases as k increases (Figure 4 (a)). Typically and theoretically, the testing MAPE would increase if k became much bigger. This is a typical behavior of the KNN algorithm when tested with previously unseen test data. The minimum MAPE during testing was obtained when k is in between 2 and 4 (the testing MAPE is around 0.9 for this range). Based on this observation and to be consistent with previously published results on a related topic (Gajan 2021), k = 3 is taken as the optimum value of the KNN algorithm for the problem considered. Any value smaller than 3 for k would overfit the training data and much greater values for k would underfit the training data. For SVR model, neither a very small value for C (underfitting the training data) nor a very high value for C (overfitting the training data) is preferred. The MAPE values of SVR model first decreases and then increases as C increases, with the optimum value of C being 20 (Figure 4 (b)).

Similar to the concept in SVR model, for DTR model (Figure 4 (c)), neither a very shallow tree (underfitting the training data) nor a very deep tree (overfitting the training data) is preferred

and an optimum value for maximum depth of the tree is needed. The MAPE values of DTR model first decreases and then increases as the maximum depth of tree increases, with the optimum maximum depth of the tree being between 6 and 9. Based on this observation, maximum depth of tree for optimum performance of DTR model is found to be 6 for the problem considered. For RFR model, as can be seen from Figure 4 (d), the MAPE decreases as number of trees in random forest increases, indicating that the accuracy of the model increases with the number of trees. However, after about 100 trees, the MAPE of the RFR model does not decrease further, indicating that the optimum number of trees required is 100.

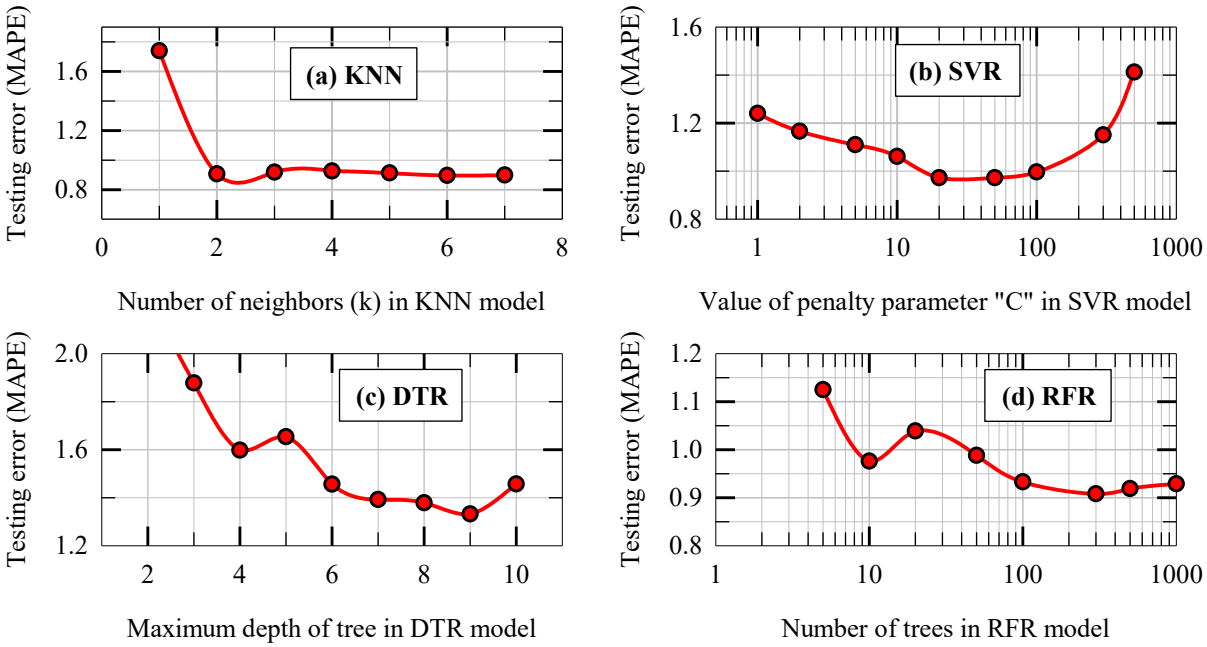

Figure 4. Results of hyperparameter tuning of ML models using 5-fold cross validation

Summary of overall average accuracy and variance of model predictions. Figure 5 presents the results of (a) MAE and (b) MAPE during repeated 5-fold cross validation tests performed on entire dataset for different machine learning models. It should be noted that results obtained for an ensemble stacked model (using stacked generalization technique and linear regression as meta model) combining KNN model and SVR model are also presented in Figure 5 (a) and (b). In each case, results obtained for testing MAE and MAPE are plotted in the form of boxplots. For each machine learning model, the boxplot plots the average (triangles), median (the horizontal line inside the box), 25th and 75th percentile values (bottom and top edges of the box), and 10th and 90th percentile values (bottom and top horizontal lines outside the box).

The average MAE of all four nonlinear, nonparametric ML models (KNN, SVR, RFR, and stacked model) are smaller than that of the baseline MLR model (Figure 5 (a)). The average MAE values of these four models for normalized settlement (S/B) vary between 0.005 and 0.006, indicating that the rocking induced settlement can be predicted within an error limit of 0.5% to 0.6% of the width of the foundation on average. A similar trend is found for MAPE presented in Figure 5 (b), except that SVR model seems to have greater variance in predictions. However, by simply combining SVR model with KNN model (using stacked generalization), the resulting stacked ensemble model reduces both the average and the variance in MAPE quite significantly.

In a previous study that incorporated four input features (Gajan 2021), the average MAPE in predicted normalized settlement is 3.0 or higher for MLR and KNN models. In the present study, six input features are used and two different ensemble methods have been developed (random forest using bagging of multiple decision trees and a stack ensemble model by combining KNN and SVR models), and they show significantly improved performance (the overall average error (MAPE) reduced from about 3.0 to around 0.7).

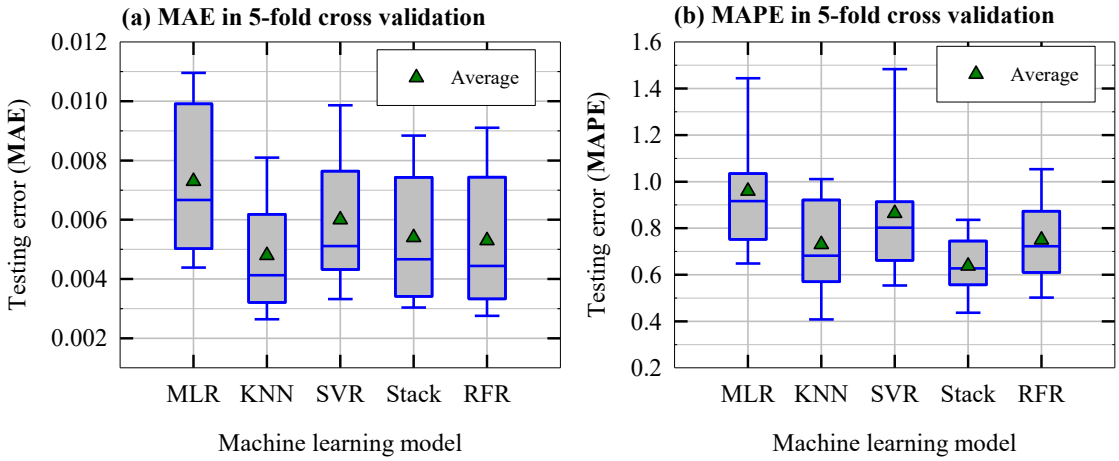

Figure 5. Summary results of 5-fold cross validation of ML models in terms of the average and variance of mean absolute error (MAE) and mean absolute percentage error (MAPE)

CONCLUSIONS

Based on the results presented in this paper, the following major conclusions are derived:

- The nonlinear, nonparametric ML models developed in this study (KNN, Stack (KNN and SVR), and RFR) perform better than a baseline MLR model in terms of accuracy and variance in predictions of rocking induced settlement of shallow foundations. The overall average testing MAPE of these three model predictions are around 0.7, and the models neither underfit nor overfit the training data.
- The overall average mean absolute errors (MAE) in predictions of all four nonlinear ML models are around 0.005 to 0.006, indicating that the rocking induced settlement can be predicted within an average error limit of 0.5% to 0.6% of the width of the footing.
- Based on the feature importance values, it can be concluded that the six chosen input features capture the rocking induced settlement satisfactorily and that none of the input features are redundant. The permanent settlement is more sensitive to rocking foundation capacity parameters than to ground motion intensity parameters.
- The data-driven predictive models developed in this study can be used in combination with other mechanics-based models to complement each other in modeling of rocking induced settlement of shallow foundations in practical applications.

ACKNOWLEDGEMENTS

The author acknowledges the National Science Foundation (NSF) for funding the research presented in this paper through award number CMMI-2138631.

REFERENCES

Anastasopoulos, I., Loli, M., Georgarakos, T., and Drosos, V. (2013). "Shaking table testing of rocking-isolated bridge pier on sand." *J. Earthquake Eng.* 17, 1–32.

Antonellis, G., Gavras, A. G., Panagiotou, M., Kutter, B. L., Guerrini, G., Sander, A. C., and Fox, P. J. (2015). "Shake table test of large-scale bridge columns supported on rocking shallow foundations." *J.Geotech. Geoenviron. Eng.* 141.

Deitel, P., and Deitel, H. (2020). *Introduction to Python for computer science and data science*. Pearson publishing (ISBN-10: 0-13-540467-3; ISBN-13: 978-0-13-540467-6).

Deng, L., and Kutter, B. L. (2012). "Characterization of rocking shallow foundations using centrifuge model tests." *Earthquake Engng Struct. Dyn.* 41, 1043–1060.

Deng, L., Kutter, B. L., and Kunnath, S. K. (2012). "Centrifuge modeling of bridge systems designed for rocking foundations." *J. Geotech. Geoenviron. Eng.* 138, 335–344.

Drosos, V., Georgarakos, T., Loli, M., Anastsopoulos, I., Zarzouras, O., and Gazetas, G. (2012). "Soil-Foundation-Structure Interaction with Mobilization of Bearing Capacity: Experimental Study on Sand." *J.Geotech. Geoenviron*, 138(11), 1369–1386.

Gajan, S. (2021). "Application of machine learning algorithms to performance prediction of rocking shallow foundations during earthquake loading." *Soil Dyn. Earthquake Engng*, 151.

Gajan, S., and Kutter, B. L. (2008). "Capacity, settlement, and energy dissipation of shallow footings subjected to rocking." *J. Geotech. Geoenviron. Eng.* 134, 1129–1141.

Gajan, S., Soundararajan, S., Yang, M., and Akchurin, D. (2021). "Effects of rocking coefficient and critical contact area ratio on the performance of rocking foundations from centrifuge and shake table experimental results." *Soil Dyn. Earthquake Engng*, 141.

Gavras, A., Kutter, B. L., Hakhamaneshi, M., Gajan, S., Tsatsis, A., Sharma, K., Kouno, T., Deng, L., Anastasopoulos, I., and Gazetas, G. (2020). "Database of rocking shallow foundation performance: Dynamic shaking." *Earthquake Spectra*.

Geron, A. (2019). *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.

Hakhamaneshi, M., Kutter, B. L., Deng, L., Hutchinson, T. C., and Liu, W. (2012). "New findings from centrifuge modeling of rocking shallow foundations in clayey ground." in *Proc, Geo-Congress 2012*, 25–29 March, 2012, Oakland, CA.

Tsatsis, A., and Anastasopoulos, I. (2015). "Performance of rocking systems on shallow improved sand: shaking table testing." *Front. Built Environ.* 1:9.