Journal of Civil Engineering (IEB), 51 (2) (2023) 99-119

Journal of
Civil Engineering
IEB

# Use of SPT and depth for estimating shear-wave velocity using optimised machine learning models

Mehedi Ahmed Ansary[1] and Mushfika Ansary[2]

*[1]Department of Civil Engineering,*
*Bangladesh University of Engineering and Technology, Dhaka, Bangladesh*
*[3]Department of Architecture,*
*University of Asia-Pacific, Dhaka, Bangladesh*

Received 29 October 2023

**Abstract**

In this study, the shear-wave velocity of soil is modeled using seven cutting-edge machine learning procedures, comprising decision trees, multilayer perceptron artificial neural networks, random forests, ridge regression, support vector regressors, K-Nearest Neighbour, and extremely gradient boosting. The hyper-parameters of these algorithms are optimized utilizing the randomized search cross-validation (RSCV) algorithm. The mean average error, root mean square error, and R-squared values are applied as evaluation indicators to assess the efficiency of optimized machine learning procedures on a dataset with 9335 data. The comparison shows that the RSCV approach is effective in the hyper-parameter tuning and that the optimized machine learning procedures have tremendous prospect to evaluate the shear-wave velocity of soil. Among the seven OMLs used for the testing dataset, SVR and MLP display relatively acceptable performance (R2 = 0.7220 and 0.7216, respectively). Both RF (R2 = 0.7183) and XGB (R2 = 0.7138) exhibit performance that is moderately satisfactory. It has been also found that the two input parameters SPT-N value and depth are almost equally important. SVR and MLP efficiency is compared to that of the current models. It is found that the OML models such as SVR and MLP outperform the existing models.

*Keywords:* Shear-wave velocity (VS), SPT-N value, optimized machine learning methods, randomized search cross-validation (RSCV) algorithm.

## 1. Introduction

An essential component of the application of geotechnical and seismic engineering is the determination of the shear wave velocities ($V_S$) of soils. In ground response analysis, which assesses the dynamic behavior of soils during earthquakes, this relationship is crucial. Because it significantly affects the ground motion amplification, the top 30-m soil layer's seismic shear wave velocity is recognized as a crucial element in earthquake engineering. Additionally, investigations on analysis of liquefaction, soil layering, including locationspecific subsurface modeling all makes use of the shear wave velocity of geomaterials. Since direct measurement of shear-wave velocity is costly and necessitates the use of cutting-edge equipment in an environment free from traffic and industrial noise, many researchers have developed correlation equations of $V_S$ with various soil indices over the years in different parts of the world (Ohta and Goto, 1978; Lee, 1992; Kuo et al., 2011; Pérez-Santisteban et al., 2016; Sil and Haloi, 2017; Thokchom et al., 2017; Lu and Hwang, 2020; Bandyopadhyay et al., 2021; Tasmiah and Ansary, 2023). Depth-based correlations equations have been developed earlier by many researchers' such as Boore and Joyner (1997), Klimis et al., (1999), Wang and Wang (2016), and others.

Also, a number of researchers (Cornou et al., 2016; Daag et al., 2022; Sil and Haloi, 2017; Thokchom et al., 2017; Lu and Hwang, 2020; Bandyopadhyay et al., 2021) from various fields of study have developed an empirical link between the penetration resistance, or N value from SPT, and shear-wave velocity, over the years. However, the N-value by itself cannot adequately describe the shear wave velocity. There have been many methods (Ohta and Goto, 1978; Lee, 1992; Kuo et al., 2011; Lu and Hwang, 2020; Bandyopadhyay et al., 2021) proposed to improve such mathematical models by using additional elements like soil type and depth derived from the ground surface.

The first multivariable analysis method was put forth in the study by Ohta and Goto (1978) as an alternative to empirical equations. Numerous researchers (Ohta and Goto, 1978; Lee, 1992; Kuo et al., 2011; *Pérez*-Santisteban et al., 2016; Tasmiah and Ansary, 2023) used multiple regression analysis and took into account depth and N-based regression equations. If the kind of soil and the impact of geology are originally explored, Ohta and Goto (1978) and Lee (1992) both found that "depth" rather than the N-value is the crucial parameter in a correlation equation. Effective overburden pressure was introduced into the calculation by Chapman et al., (2006). According to Kuo et al., (2011) study, the regression model should be chosen using the maximum coefficient of correlation $R^2$ between Vs as well as N or depth.

Kim et al., (2020) used artificial neural networks (ANN) to predict the SPT-N value at the non-drilling investigation points through patterns which is studied by multi-layer perceptron and error back-propagation algorithms using the minimum geotechnical data. Motahari et al., (2022) used SPT-N value results collected from north-east Iran to establish relationships for estimating the relative density in a sandy soil through artificial neural network and statistical analysis. Utilizing a k-Nearest Neighbor (k-NN) machine learning algorithm, Galupino and Dungca (2022) developed a novel method for forecasting typical SPT-N values for each Barangay/Zone of Phillipines. Latitude, longitude, and depth of the borehole were used as input parameters, while the SPT-N values were used as an output parameter. Hossain et al., (2022) used machine learning models like Multiple Linear Regression (MLR), Support Vector Regression (SVR), and Artificial Neural Network (ANN) algorithms to approximate the angle of internal friction of silty sand (SM) of Bangladesh by using SPT-N values, the grain size analysis results, the depth of sample collection.

It is observed from the above studies that very limited researches have been undertaken using machine learning models to estimate a few soil parameters from SPT-N values and other soil properties. None has used machine learning techniques to estimate shear-wave velocity of soils from SPT-N values and depth. Only However, there are still a number of issues that require correct resolution before applying the ML techniques for estimating shear-wave velocity of soils from SPT-N values and depth, such as (1) The viability of other methods has not been extensively investigated, and only a small number of sophisticated ML algorithms have been used in soil density estimate, (2) before using ML algorithms on datasets, it is essential to correctly adjust their hyper-parameters and (3) there is still a need for an organized, thorough comparison of the existing ML techniques. The efficacy of modern ML algorithms when used to estimate shear-wave velocity of soils from SPT-N values and depth may differ significantly, necessitating further research.

This paper employs seven optimized machine learning (OML) techniques in a comparative manner, utilizing the programming language Python, to fill a gap in the existing papers concerning the soil unit weight. The decision tree (DT), K-Nearest Neighbour (KNN), extreme gradient boosting (XGB), multilayer perceptron artificial neural network (MLPANN), RIDGE regression (RIDGE), random forest (RF), and support vector machine (SVM) are the seven ML algorithms are used for this purpose. In this study, a recently collected 378 SPT-PSlog collocated points having 9335 datasets (6160 for sand and 3175 for clay) were used. The depth of the soil sample collected (D), SPT-N values (N) are the input features of the algorithms and will be discussed in the data processing section.

Mean absolute error (MAE), root mean square error (RMSE), and R-squared value ($R^2$) are used as performance indicators for evaluating the efficiency of the seven ML methods. Investigations on the comparative significance of important input variables for shear-wave velocity of soils were also conducted. The limitations of many existing methods are resolved by the current study, which can more effectively estimate the shear-wave velocity of soils than is currently done.

## 2. Methodology

The shear-wave velocity of soils and its affecting variables are investigated in this study using six ML algorithms. The hyper-parameters of these seven algorithms are optimized utilizing the randomized search cross-validation (RSCV) algorithm. The seven ML algorithms and RSCV are briefly described in this section.

Table 1
Correlations developed between shear-wave velocity of soils and SPT-N values and depth of soil in the past years

| Author(s) | No of Data Point | Soil Type | Equation | $R^2$ |
|---|---|---|---|---|
| Ohta and Goto (1978) | 300 | All | $V_s = 61.62N^{0.254}D^{0.222}$ | 0.6724 |
| Lee (1989) | 88 | CL/All | $V_s = 74.44N^{0.16}D^{0.25}$ | 0.78 |
| | | CL/Keelung | $V_s = 71.52N^{0.08}D^{0.29}$ | 0.83 |
| | | CL/Tanshuei | $V_s = 58.56N^{0.13}D^{0.37}$ | 0.92 |
| | | ML/All | $V_s = 73.70N^{0.14}D^{0.26}$ | 0.88 |
| | | SM/All | $V_s = 57.97N^{-0.01}D^{0.46}$ | 0.86 |
| Lee (1992) | 126 | SM | $V_s = 76.16N^{0.076}D^{0.313}$ | 0.776 |
| | | | $V_s = 68.77N^{0.075}(D+1)^{0.340}$ | 0.779 |
| | 265 | CL | $V_s = 95.72N^{0.124}D^{0.210}$ | 0.785 |
| | | | $V_s = 86.10N^{0.116}(D+1)^{0.244}$ | 0.788 |
| | 100 | ML | $V_s = 90.57N^{0.140}D^{0.205}$ | 0.829 |
| | | | $V_s = 82.79N^{0.134}(D+1)^{0.233}$ | 0.830 |
| | 365 | CL/ML | $V_s = 93.54N^{0.125}D^{0.213}$ | 0.798 |
| | | | $V_s = 84.53N^{0.118}(D+1)^{0.246}$ | 0.801 |
| Kuo et al. (2011) | 719 | Sand | $V_s = 93.11N^{0.242}D^{0.136}$ | 0.671 |
| | | Clay/Silt | $V_s = 114.55N^{0.168}D^{0.143}$ | 0.685 |
| Pérez-Santisteban et al. (2016) | 500 | All | $V_s = 71.05N^{0.259}D^{0.382}$ | 0.760 |
| Tasmiah & Ansary (2022) | 9335 | All | $V_s = 63.07D^{0.360}N^{0.120}$ | 0.7171 |
| | 6160 | Sand | $V_s = 59.61D^{0.321}N^{0.165}$ | 0.6844 |
| | 3175 | Clay | $V_s = 63.29D^{0.383}N^{0.111}$ | 0.7428 |

Table 2
Summary of the dataset

| | D (m) | SPT-N Value | $V_s$ (m/s) |
|---|---|---|---|
| Count | 9335 | 9335 | 9335 |
| Mean | 21.2 | 30.4 | 272.9 |
| STD | 13.0 | 24.2 | 113.3 |
| Min | 1.5 | 1 | 15 |
| Max | 79.5 | 100 | 850 |

## *2.1 Decision Tree (DT)*

A supervised machine learning technique which may be utilized to both classification and regression applications is the decision tree (DT) model. It is a predictive model that, in an effort to produce precise predictions, splits the data recursively depending on feature values to create a tree-like structure. A decision tree model works as follows: (a) Tree structure: The decision tree model consists of nodes and edges. Each edge in the graph reflects a potential value or range of values for each node's corresponding feature or attribute. The tree has a root node, and several internal and leaf nodes that fork from it. (b) Feature selection: A choice is made on the basis of a specific characteristic or attribute at every internal node of the tree. Typically, the decision is a binary split, which divides the information into two subsets according to a selected threshold or criteria.

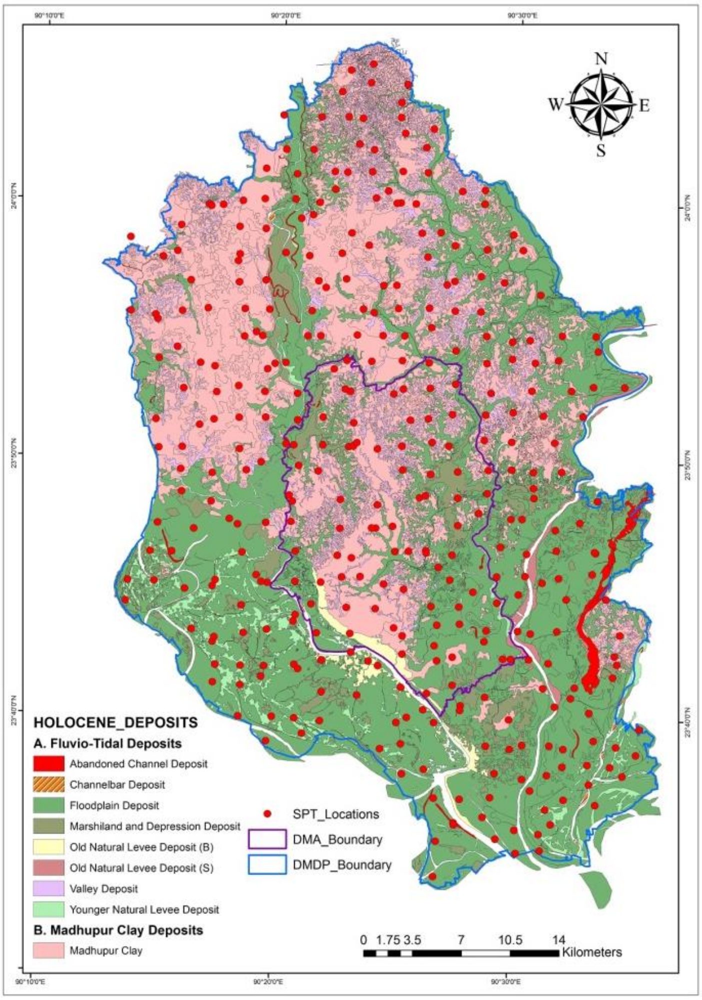

Fig. 1. Locations of SPT/PSLOG tests performed in the DMDP area along with the geology.

The objective is to identify the splits that optimize the target variable's homogeneity or purity within each subset. (c) Recursive splitting: The splitting process continues recursively, creating child nodes for each split. The splitting criteria depend on the specific algorithm and objective. Common criteria include minimizing impurity measures such as entropy or Gini impurity for classification tasks, or minimizing MSE or MAE for regression tasks. (d) Leaf nodes and predictions: Once the splitting process reaches a stopping condition, usually based on a maximum tree depth or a minimum number of samples per leaf, the remaining nodes become leaf nodes. Each leaf node represents a predicted value or class label. For classification, the majority class label within the leaf is often used, while for regression, it can be the target variable's mean or median value for the leaf. (e) Prediction process: To make predictions, a new data point traverses the decision tree by following the split conditions at each internal node until it reaches a leaf node. The predicted value or class label associated with that leaf node is then assigned as the final prediction.

Key characteristics and considerations of the DT models are: (a) Interpretable: Decision trees provide human-readable rules that can be easily understood and visualized. The splits and decisions made by the tree can be interpreted to gain insights into the associations between the features and the desired outcome. (b) Non-linear relationships: Complex non-linear associations between the features and the target factor can be identified through decision trees. (c) Overfitting: Decision trees have the tendency to overfit the training data if not properly controlled. Techniques like pruning, setting a maximum depth, or using regularization parameters can help prevent overfitting and improve generalization to unseen data. (d) Feature importance: Decision trees can provide information about the importance or relevance of different features in predicting the target variable. Features that are closer to the root of the tree and appear in multiple splits are considered more important. (e) Handling missing values: Any values that are missing in the data can be handled by decision trees by assigning them to the most appropriate split based on the available information.

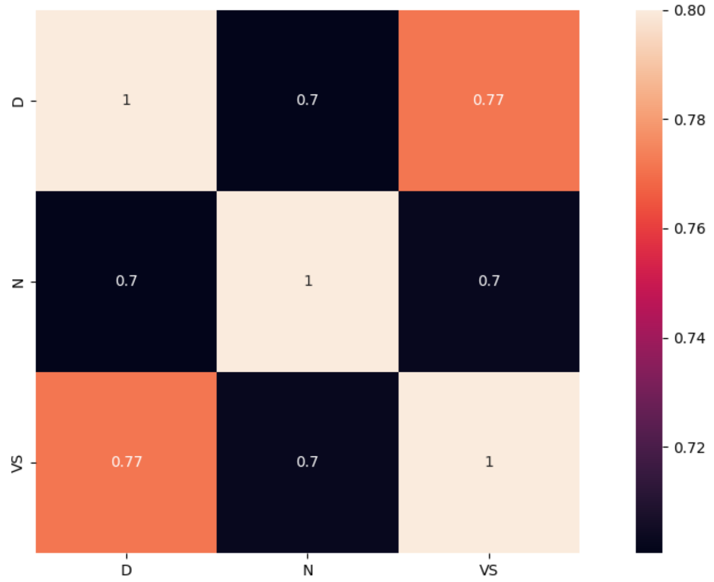

Fig. 2. Correlation matrix of one output variable and two input variables

Decision trees are widely used in various domains due to their interpretability, ease of handling, and ability to handle both categorical and numerical features. However, they may not perform as well as more complex models on certain datasets, and their performance can be affected by outliers and imbalanced classes. It's worth noting that decision trees can be extended and improved through ensemble techniques such as random forests, and gradient boosting, which integrate multiple decision trees to achieve better predictive performance.

Table 3
Hyper-parameter tuning

| ML algorithms | Optimum value |
|---|---|
| DT | 'criterion': 'poisson', 'max_depth': 3, 'max_features': 7, 'min_samples_leaf': 7 |
| XGB | subsample': 1.0, 'reg_lambda': 0, 'reg_alpha': 0.5, 'n_estimators': 200, 'max_depth': 6, 'learning_rate': 0.1, 'colsample_bytree': 0.8 |
| KNN | 'weights': 'uniform', 'p': 2, 'n_neighbors': 11 |
| MLP | 'activation': 'relu', 'alpha': 0.015701864044243653, 'hidden_layer_sizes': 83, 'learning_rate': 'adaptive', 'max_iter': 430, 'solver': 'lbfgs' |
| RF | max_depth=6, max_features=None, max_leaf_nodes=9, n_estimators=50 |
| RIDGE | 'solver': 'cholesky', 'alpha': 0.20565123083486536 |
| SVR | 'kernel': 'rbf', 'gamma': 1, 'C': 1 |

## 2.2 *Extreme Gradient Boosting (XGB)*

XGB is a standard machine learning technique known for its efficiency and efficiency in both regression and classification tasks. It uses a more sophisticated version of the gradient boosting architecture that utilizes an optimized algorithm and various regularization techniques to improve model accuracy and generalization.

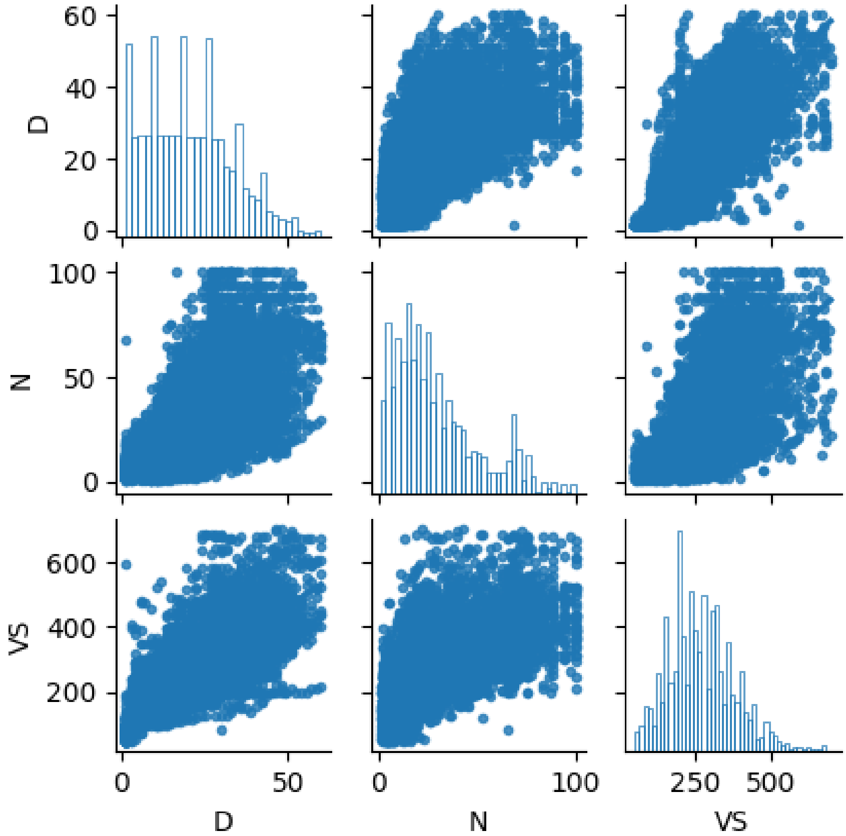

Fig. 3. Pair panel of input and output variables.

XGB model works as follows: (a) Gradient Boosting Framework: The foundation of XGB is the gradient boosting framework, which creates a powerful ensemble model by integrating a number of weak predictive models (usually decision trees). The goal of gradient boosting is to continuously train models that minimize the errors produced by the past models. (b) Optimization Algorithm: XGB employs a highly optimized algorithm to efficiently build the ensemble of weak models. The algorithm leverages parallel processing and tree pruning techniques to decrease memory utilization and gear up the training process. (c) Regularization Techniques: XGB incorporates several regularization procedures to avoid overfitting and enhance the quality of generalization. Regularization methods include shrinkage (learning rate), this regulates how much each tree contributes to the total forecast, which add penalties to the model's complexity. (d) Tree Construction: XGB uses decision trees as base learners. It constructs trees in a greedy manner by iteratively splitting the data based on specific criteria, such as reducing the loss or maximizing the information gain. The tree construction process is guided by optimization objectives and constraints to find the best splits and create trees that capture important patterns in the data. (e) Feature Importance: XGB provides a measure of feature importance, which indicates the relative importance of each input feature in the prediction process. Based on how often a feature is utilized to divide the data among all the ensemble trees, feature significance scores are determined. (f) Hyperparameter Tuning: A variety of hyperparameters are available in XGB that can be adjusted to enhance the efficiency of the model.

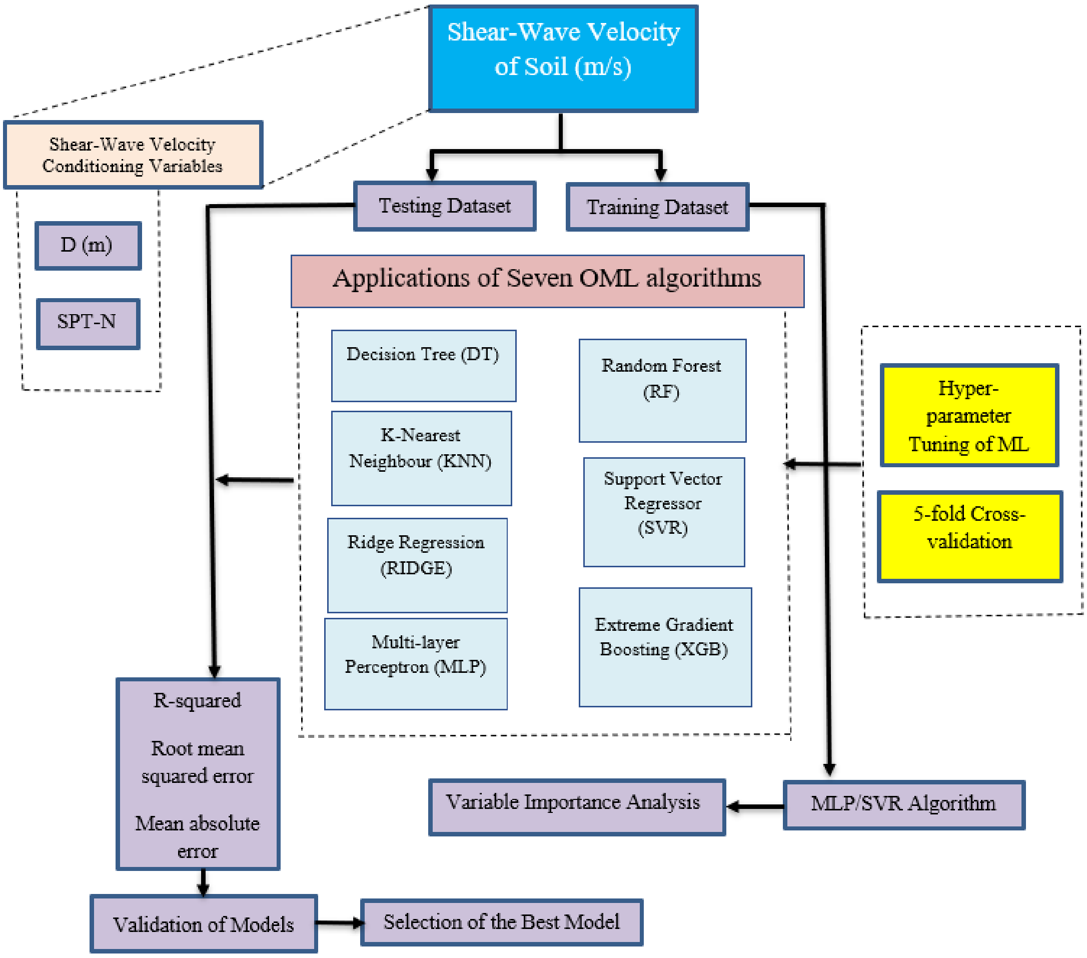
Shear-Wave Velocity of Soil (m/s)
Shear-Wave Velocity Conditioning Variables
D (m)
SPT-N
Testing Dataset
Training Dataset
Applications of Seven OML algorithms
Decision Tree (DT)
Random Forest (RF)
K-Nearest Neighbour (KNN)
Support Vector Regressor (SVR)
Ridge Regression (RIDGE)
Extreme Gradient Boosting (XGB)
Multi-layer Perceptron (MLP)
Hyper-parameter Tuning of ML
5-fold Cross-validation
R-squared
Root mean squared error
Mean absolute error
Variable Importance Analysis
MLP/SVR Algorithm
Validation of Models
Selection of the Best Model

Fig. 4. Methodological flowchart of this study.

XGB offers a comprehensive hyperparameters that can be tweaked to optimize the model's performance. Hyperparameters govern various aspects of the algorithm, such as the learning rate, regularization factors, tree depth, subsampling ratio, etc. XGB has gained popularity for its exceptional performance in machine learning competitions and real-world applications. It is capable of handling large datasets, capturing complex relationships, and providing accurate predictions. However, proper hyperparameter tuning and careful validation are important to achieve optimal results and prevent overfitting.

### *2.3 K-Nearest Neighbour (KNN)*

A non-parametric supervised learning approach used for classification and regression applications is the k-nearest neighbors (KNN) model. Based on the similarity between a data point and its k nearest neighbors in the training data, this straightforward and understandable method provides predictions. Following are the several steps of the KNN algorithm: (a) Load the training data: The KNN algorithm starts by loading the labeled training data, which consists of input feature vectors and their corresponding target values. (b) Select the number of neighbors (k): The parameter k represents the number of neighbors to consider when making predictions. It is typically chosen based on experimentation and validation. (c) Calculate distances: The algorithm determines the distance between each test point and every other point in the training data for a particular test data point. While there are many other distance metrics, some of the most popular ones are Euclidean distance, Manhattan distance, and Minkowski distance. (d) Find the k nearest neighbors: The k training samples with the closest distances to the test point are chosen by the algorithm. This k nearest neighbors will contribute to the prediction for the test point. (e) Make predictions: The predicted value for the test point in regression tasks is often the average or weighted average of its k nearest neighbors' target values. For classification tasks, the predicted class is determined by the majority class among the k nearest neighbors. (f) Evaluate and repeat: Performance indicators like accuracy, mean squared error (MSE), and others can be used to assess the algorithm. To determine the optimal model configuration, the procedure can be repeated with other k values or distance measures. Key characteristics and considerations of the KNN algorithm are: (a) Non-parametric: Because it makes no assumptions about the distribution of the underlying data, KNN is a non-parametric algorithm. (b) Lazy learning: KNN is considered a lazy learning algorithm since it does not explicitly build a model during training. Instead, it simply stores the training data for reference during prediction. (c) Feature scaling: It is important to scale the input features before applying KNN, as features with larger scales can dominate the distance calculations. (d) Curse of dimensionality: KNN can suffer from the curse of dimensionality, where the performance deteriorates as the number of dimensions (features) increases. In high-dimensional spaces, the concept of distance becomes less meaningful.

Table 4
Statistical analyses of seven ML algorithms for all soils

| ML algorithms | | MAE | RMSE | $R^2$ | Ranking | Clay $R^2$ | Sand $R^2$ |
|---|---|---|---|---|---|---|---|
| DT | Training | 0.1915 | 0.2521 | 0.7079 | 7 | 0.7250 | 0.8019 |
| | Testing | 0.1916 | 0.2458 | 0.6838 | 7 | 0.7581 | 0.5730 |
| XGB | Training | 0.1812 | 0.2382 | 0.7394 | 4 | 0.7567 | 0.7109 |
| | Testing | 0.1842 | 0.2338 | 0.7138 | 4 | 0.7741 | 0.7103 |
| KNN | Training | 0.1761 | 0.2313 | 0.7542 | 1 | 0.7740 | 0.7314 |
| | Testing | 0.1874 | 0.2406 | 0.6970 | 6 | 0.7637 | 0.6856 |
| MLP | Training | 0.1812 | 0.2387 | 0.7381 | 5 | 0.7564 | 0.7033 |
| | Testing | 0.1809 | 0.2306 | 0.7216 | 2 | 0.7782 | 0.7204 |
| RF | Training | 0.1776 | 0.2335 | 0.7495 | 2 | 0.8036 | 0.7539 |
| | Testing | 0.1815 | 0.2319 | 0.7183 | 3 | 0.7602 | 0.6917 |
| RIDGE | Training | 0.1881 | 0.2478 | 0.7178 | 6 | 0.7363 | 0.6813 |
| | Testing | 0.1833 | 0.2342 | 0.7128 | 5 | 0.7657 | 0.6964 |
| SVR | Training | 0.1803 | 0.2379 | 0.7401 | 3 | 0.7582 | 0.7139 |
| | Testing | 0.1806 | 0.2304 | 0.7220 | 1 | 0.7796 | 0.6973 |

KNN is a versatile algorithm that can be used in various domains and is particularly useful when the decision boundaries are non-linear or when the training data has complex patterns. However, as the algorithm needs to calculate distances to all training points for each prediction, it can be computationally demanding for large datasets.

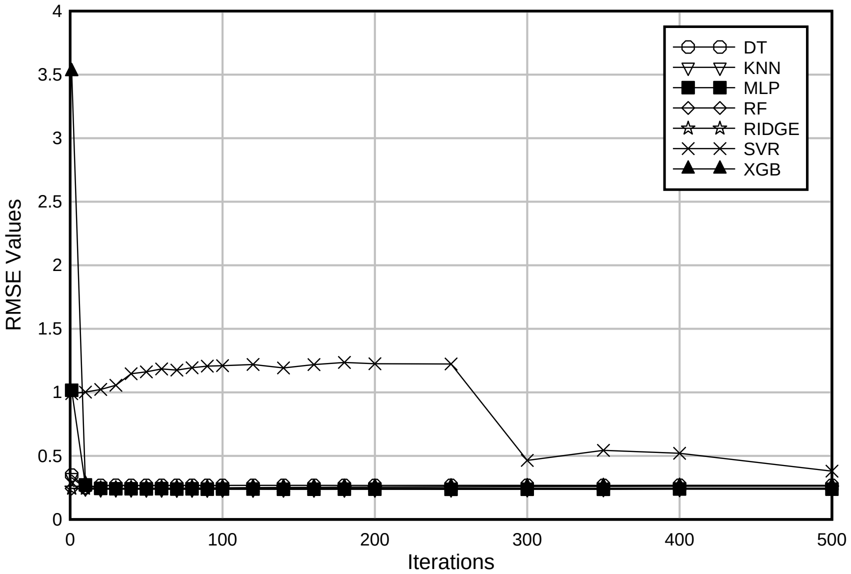

Fig. 5. RMSE values along with iterations on the training dataset.

### 2.4 Multilayer Perceptron (MLP) Artificial Neural Network

MLP artificial neural networks are a common tool for classification and regression among other machine learning problems. It draws inspiration from the design and operation of the human brain. The MLP is made up of numerous layers of neurons, which are interconnected nodes. Usually, there are three different types of layers. (a) A set of features or attributes may be received by the input layer as input data. (b) Layers between the input and output layers are considered hidden layers. Each neuron in a hidden layer takes information from the layer below and processes it before sending the results to the layer above it. The network can learn intricate representations and patterns in the input thanks to the hidden layers. (c) The network's ultimate output is produced by the output layer.

The type of task determines how many neurons are present in the output layer. For example, in a regression task, there is typically a single neuron for predicting a continuous value, while in a classification task; there is one neuron per class for predicting class probabilities. The neurons in an MLP are connected by weighted connections, which determine the strength and importance of the information flowing between neurons. Each neuron generates an output by applying an activation function to the weighted sum of its inputs. In order to reduce the discrepancy between the projected outputs and the actual targets, the MLP modifies the weights of the connections during training. The weights are iteratively updated based on the computed error in order to do this using an optimization approach like gradient descent. Finding the ideal collection of weights is the goal in order to reduce prediction errors and increase the network's capacity to generalize to new inputs. A non-linear function, such as the hyperbolic tangent (tanh) function, rectified linear unit (ReLU) function, sigmoid function, may be employed as the activation function in an MLP. The network can learn intricate connections between the inputs and outputs thanks to non-linear activation functions. MLPs are known for their ability to approximate complex functions and learn non-linear forms in the data.

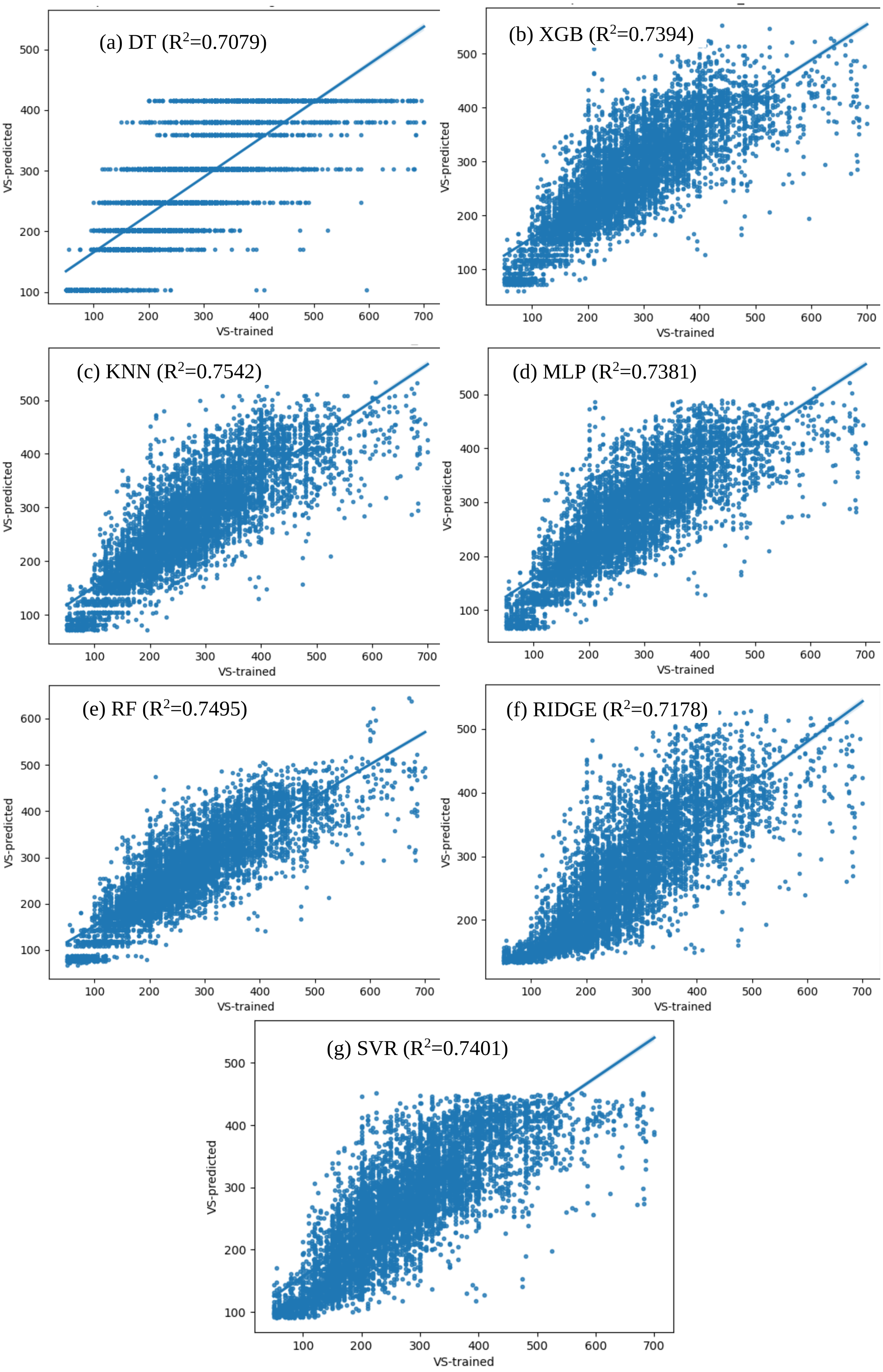

Fig. 6. Regression plots for the training set of seven OML algorithms.

Table 5
Performance evaluation of the tested values versus earlier models and two ML algorithms

| Model Name | $R^2$ |
|---|---|
| Ohta & Goto (1978) | 0.6802 |
| Pérez-Santisteban et al. (2016) | 0.7048 |
| Tasmiah & Ansary (2022) | 0.7157 |
| Present Study (SVR) | 0.7331 |
| Present Study (MLP) | 0.7458 |

However, they can be susceptible to overfitting if not suitably regularized or if the network architecture is not appropriately designed. MLPs have become popular in various domains due to their flexibility, scalability, and effectiveness in handling complex datasets. They have been effectively used in a variety of machine learning applications, including time series analysis, natural language processing, image identification, and many more.

### *2.5 Random Forest (RF)*

An ensemble learning technique called RF combines the predictions of multiple decision trees to get predictions that are more accurate. Both classification and regression tasks can be accomplished with this flexible and effective technique. RF method works as follows: (a) Ensemble Learning: Random Forest belongs to the family of ensemble learning methods, which combine multiple individual models to make collective predictions. In the case of Random Forest, the individual models are decision trees. (b) Decision Trees: Decision trees are predictive models that learn a series of hierarchical if-else rules based on the features of the data. Each decision tree makes predictions by following a track from the root node to a leaf node, where the final prediction is made. (c) Randomness and Diversity: Random Forest introduces randomness and diversity into the modeling process. Randomness is introduced by randomly selecting subsets of the original data for training each decision tree (bootstrap aggregating or bagging). Diversity is achieved by arbitrarily picking a subset of features for each split in the decision tree. (d) Voting and Aggregation: When making predictions, every decision tree in the Random Forest independently forecasts the target variable. For classification tasks, the class with the majority of votes among the trees is selected as the final prediction. For regression tasks, the average or median of the predicted values from all the trees is taken as the final prediction. (e) Feature Importance: Random Forest provides a measure of feature importance, indicating the comparative significance of each input feature in making predictions. The importance is calculated based on how much each feature contributes to the reduction of impurity or variance across all the decision trees. (f) Hyper-parameter Tuning: There are several hyper-parameters in Random Forest that can be tweaked to enhance efficiency. The maximum depth of each tree, the quantity of trees in the forest, the amount of features taken into account for each split, etc. are some significant hyper-parameters. Random Forest is known for its robustness, scalability, and capacity for handling high-dimensional data. It is less prone to over-fitting compared to individual decision trees and often yields better performance in terms of accuracy. However, like any algorithm, proper hyper-parameter tuning and careful validation are crucial to achieve optimal results.

### *2.6 Ridge regression*

The Ridge method, also known as Ridge regression or Tikhonov regularization, is a regularization technique used in linear regression models. It helps to address the issue of multi-collinearity (high correlation between features) and reduce the impact of less important features on the model. Ridge method works as follows: (a) Objective Function: The objective function for linear regression includes a penalty term thanks to the Ridge technique. The objective function attempts to reduce the quantity of squared residuals, which gauges the difference between expected and observed values. The penalty term is the L2 norm (squared values) of the coefficients multiplied by a regularization parameter (lambda or alpha). (b) L2 Regularization: The L2 norm penalty in the Ridge method is the sum of the squared values of the coefficients.

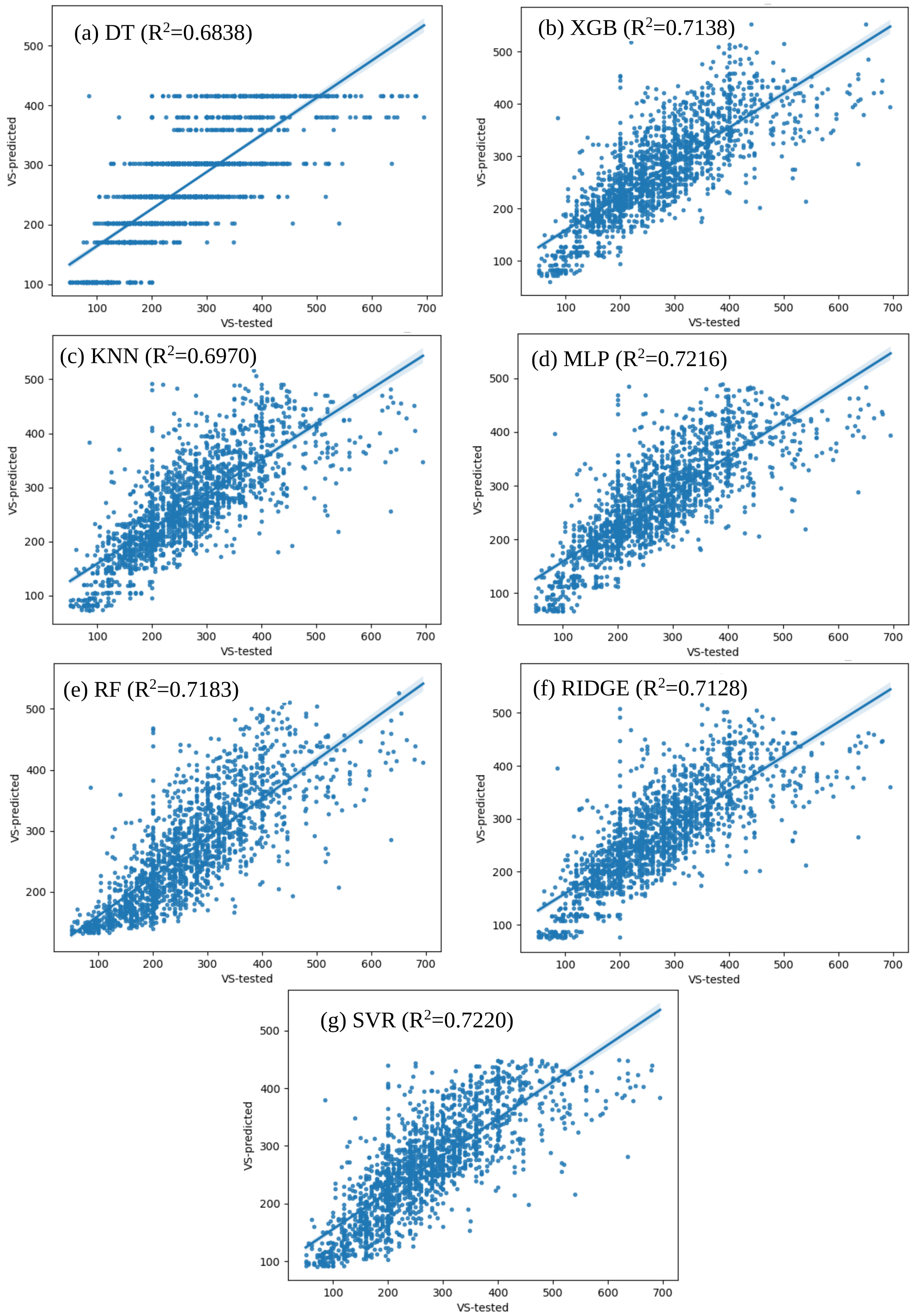

Fig. 7. Regression plots for the testing set of seven OML algorithms.

This penalty term discourages extreme values of the coefficients and encourages smaller, more spread-out values. It aids in managing the model's complexity and lessens the effects of multi-collinearity. (c) Shrinkage of Coefficients: The Ridge method shrinks the coefficients towards zero, reducing their magnitudes. The amount of shrinkage is controlled by the regularization parameter. A larger regularization parameter results in more aggressive shrinkage and smaller coefficients. (d) Multi-collinearity Handling: Ridge regression is particularly useful when dealing with datasets that have multi-collinearity, where features are highly correlated. By shrinking the coefficients, Ridge regression reduces the impact of highly correlated features and prevents them from dominating the model. (e) Bias-Variance Tradeoff: The Ridge method helps in striking a balance between bias and variance in the model. Increasing the regularization parameter increases the bias of the model but reduces its variance, while decreasing the regularization parameter has the opposite effect. Proper tuning of the regularization parameter is important to find the right balance for the given dataset. (f) Hyper-parameter Tuning: Ridge regression involves tuning the regularization parameter (lambda or alpha) to optimize the model's efficiency. Cross-validation techniques can be utilized to evaluate different values of the regularization parameter and select the optimal one.

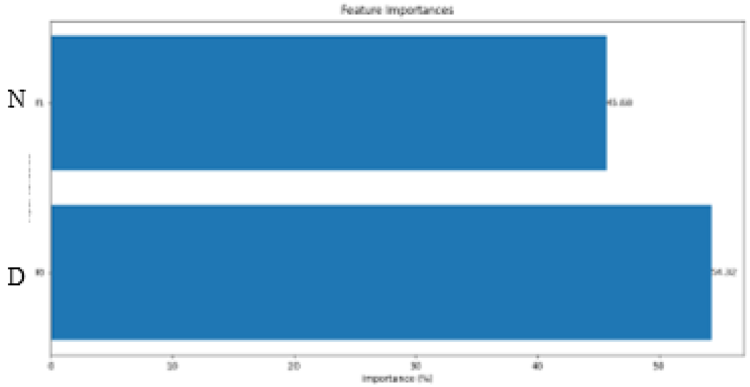

Fig. 8. Relative variable importance for shear-wave velocity of soils.

The Ridge method is widely used in various domains to handle multi-collinearity and improve the stability and generalization of linear regression models. By shrinking the coefficients, in the presence of strongly linked predictors, it aids in determining and ranking the most important features.

## 2.7 *Support Vector Machine (SVM)*

Notable supervised machine learning methods for classification and regression include Support Vector Machine (SVM). It is particularly effective in handling complex datasets with clear margin or separation between different classes. A SVM model works as follows: (a) Basic Concept: In a high-dimensional feature space, SVM seeks to identify the best hyperplane for classifying the data points. In binary classification, SVM seeks to find a hyperplane that maximizes the margin between each class's nearest data points. Support vectors are utilized to express these nearby data points. (b) Feature Space and Hyperplane: The feature space refers to the transformed space where the input data points are mapped using a kernel function. A hyperplane which most effectively divides the data points has been identified by SVM. In two dimensions, the hyperplane is a line, while in higher dimensions, it becomes a hyperplane. (c) Margin and Support Vectors: The margin is the gap between the nearest data points for each class and the hyperplane. SVM seeks to increase this margin, as a larger margin usually implies better generalization and robustness to new data. The data points represent the support vectors that lie on the margin or are misclassified. These points influence the position and orientation of the hyperplane. (d) Linear and Non-linear Separation: SVM can handle both linearly separable and non-linearly separable data. For linear separation, a linear kernel (e.g., the linear function) is used to create a linear decision boundary. For non-linear separation, SVM utilizes kernel functions (for instance, a polynomial or a radial basis function) to translate the data into a space with more dimensions, where a linear separation is possible. (e) Training Process: Given a labeled training dataset, SVM determines the optimal hyperplane by solving an optimization problem. The optimization problem involves identifying the hyperplane that increases the margin while minimizing the classification errors. The solution is obtained by solving a quadratic programming problem or through convex optimization techniques. (f) Prediction: Once the optimal hyperplane is determined, SVM can predict the class label of new, unseen data points by evaluating which side of the hyperplane they fall on. Key characteristics and considerations of the SVM models are: (a) Versatility: SVM can handle both linear and non-linear classification tasks. (b) Robustness: SVM is less prone to overfitting due to the margin maximization objective. (c) Kernel functions: The choice of kernel function can significantly impact SVM's performance and ability to handle complex datasets. (d) Model complexity: The complexity of the SVM model depends on the number of support vectors, which affects training and prediction time. SVM is frequently utilized in many fields, including image classification, text classification, and bioinformatics. Effectively separate classes and handling of high-dimensional data makes it a valuable tool in machine learning.

### *2.8 Randomized Search Cross-Validation (RSCV)*

The term "Randomized Search CV" refers to cross-validation. It is a method for selecting models and tweaking hyperparameters in machine learning. Hyperparameters are settings made by the user prior to training a machine learning model rather than ones that are learned from the data. The rate of learning, the quantity of hidden layers in a neural network, or the regularization strength is a few examples of hyperparameters. To determine the ideal set of hyperparameters for a particular model, Randomized Search CV combines cross-validation and random sampling of hyperparameters. It operates by selecting a subset of hyperparameter combinations at random from a predetermined search space and assessing their effectiveness using cross-validation. Here is a detailed explanation of how Randomized Search CV operates: (a) Establish a search area: Indicate the range of values or distributions from which to sample the hyperparameters, (b) Randomly sample hyperparameter combinations: Pick a selection of hyperparameter combinations at random from the search space, (c) Evaluate each combination: Utilizing each combination of hyperparameters, develop and test the model. Typically, k-fold cross-validation is used for this, where the data is divided into k subsets (folds), the model is trained and assessed k times, and each time, a different fold is used as the validation set, (d) Choose the optimal combination: The performance metric (such as accuracy, precision, or recall) achieved during the cross-validation procedure should be used to determine the optimum hyperparameter combination (e) Train the model again: On the complete training dataset, train the model using the optimal combination of hyperparameters. Randomized Search CV rapidly explores a wide variety of hyperparameter combinations without analyzing all potential possibilities by using random sampling as opposed to an exhaustive grid search. This makes it appropriate in situations where the hyperparameter search space is huge or when there are not enough processing resources. In general, Randomized Search CV aids in automating the hyperparameter tuning process, enabling the choice of ideal hyperparameters for a machine learning model.

## 3. Preparations of data and interpretation

In order to evaluate the shear-wave velocity of soils through a comparison analysis, seven ML procedures are utilized to the dataset of 9335 instances of SPT-N and corresponding shear-wave velocity values. These are collected from 378 collocated SPT and PS logs for shear-wave velocity within the DMDP area of Bangladesh as shown in Figure 1. It should be noted that the dataset is fairly thorough and contains a wide range of metrics that are important for figuring out the shear-wave velocity of soils.

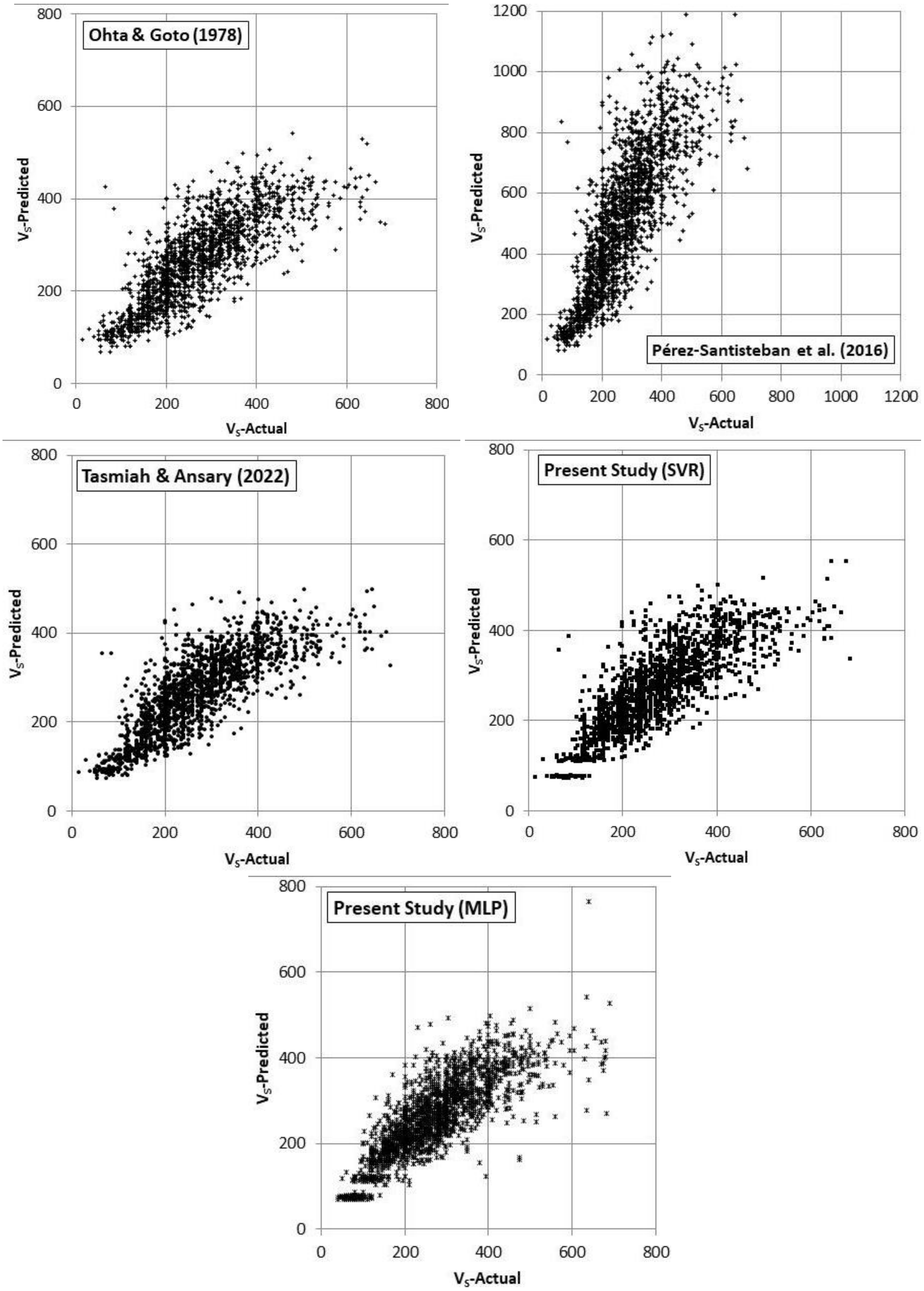

Fig. 9. Relation between actual and predicted value of shear-wave velocity of soils.

For determining shear-wave velocity of soils, it is critical to choose the factors that will have the biggest impact. As input characteristics, two variables are chosen, including depth (D), SPT-N values (N) are nominated as input parameters (see Table 2). Figure 2 shows the correlation matrix for the affecting variables as a heat map and Figure 3 shows the affecting variables as a pair panel. The correlation matrix displays the correlation coefficient between the variables, while the pair panel displays the histogram of individual variable and scatter plots between two variables.

### 3.1 Data splitting and cross-validation

To preserve the model's capacity to simplify while addressing the overfitting issue, in this work, around 80% of the instances are considered in the training set and 20% of specimens are allotted to the testing set utilizing arbitrary selection. It should be mentioned that before performing any modeling, we have normalized the dataset. The objective of normalization is to convert the dataset's values to a mutual scale without affecting variations in the value ranges.

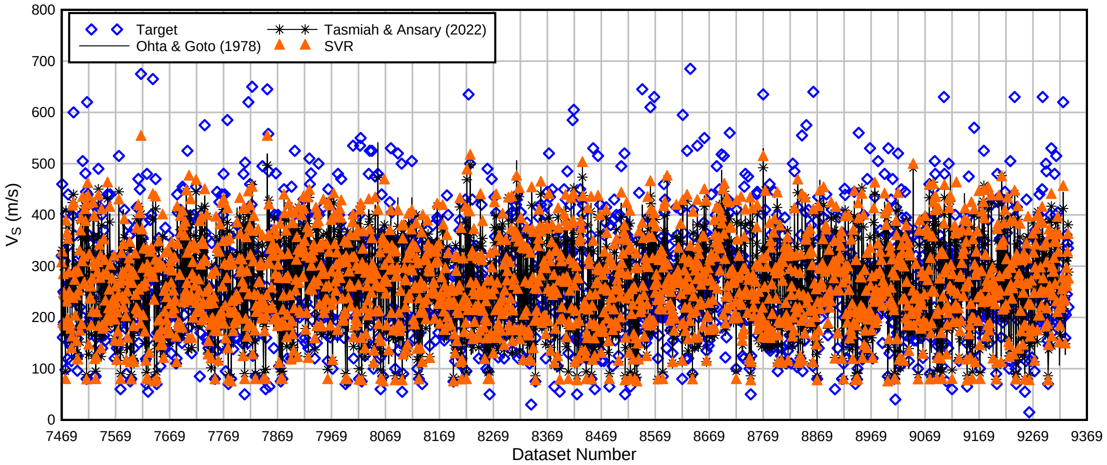

Fig. 10. Measured and predicted shear-wave velocity of soils at testing stages.

The predictive power of seven OML algorithms is assessed in this study using K-fold cross-validation on the same data. The data can be subjected to cross-validation techniques to reduce the likelihood of overfitting and bias during selection in the ML approaches. The data is split into K equal-sized subsets for the K-fold CV. The single surviving subset of the K subsets is employed as the testing data, while the K - 1 subsets are utilized as training data. Then, this procedure is carried out K times using various subsets as the testing subset. In order to evaluate OML algorithms on a small sample of data, CV is a resampling approach. The 5-fold CV is the most common CV, which has been utilized in this study.

### 3.2 Measures of performance

The shear-wave velocity of soils (VS) is investigated between actual and estimated values using mean absolute error (MAE), root mean square error (RMSE), and R-squared value ($R^2$) to show the accuracy of seven OML algorithms' estimation. The mean absolute error is the mean absolute error between actual and predicted values. The most often used metric for assessing models is the mean squared error. Here, the difference between actual values and anticipated values is squared, and the mean of those values is computed for each data point. The MSE can be a helpful statistic to employ when the dataset contains unforeseen values, either very high or low values. However, the MSE can either overstate or underestimate how awful the prediction is when dealing with noisy data, i.e., when the data are not completely dependable. The RMSE is described as a root of MSE. A statistical parameter R-squared may be used to assess the accuracy of the fit which represents how narrowly an algorithm resembles the real data points.

## 4. Analysis findings

This section discusses hyper-parameter tuning, an evaluation of seven ML methods for estimating the shear-wave velocity of soils, and the significance of influencing variables. Figure 4 depicts the process for shear-wave velocity of soils using OML approaches. Training and testing datasets are created from the initial dataset. Seven cutting-edge ML algorithms are optimized after being trained on the training dataset. Then, the OML models are utilized to the test dataset in order to compare their results.

### *4.1 Hyper-parameter tuning results*

The hyper-parameters of every ML method which have been obtained through the randomized search cross-validation (RSCV) algorithm are shown in Table 3, along with their tuned values. Figure 5 displays the evolution of the root mean squared error value over the training dataset's iterations. Figure 5 shows that hyper-parameter adjustment, especially for MLP, SVR and XGB, has a significant impact on how well ML algorithms perform.

### *4.2 Review and comparison of six machine learning models*

### *4.2.1 Training dataset results*

On the training data of 7468 shear-wave velocity of soils and corresponding SPT-N and depth, seven OML algorithms are used. The regression graphs for each of these techniques are displayed in Figure 6. Among the seven OMLs, KNN exhibits a relative better performance ($R^2$ = 0.7542). The performance of RF ($R^2$ = 0.7495), SVR ($R^2$ = 0.7401), XGB ($R^2$ = 0.7394) and MLP ($R^2$ = 0.7381) are next. These are followed by RIDGE ($R^2$ = 0.7178) and DT ($R^2$ = 0.7079) showing relatively mediocre performances.

### *4.2.2 Testing dataset results*

The shear-wave velocities of soils in the testing dataset are now estimated using seven OML models that were trained in the former section. Seven OML approaches' performance on 1867 samples from the testing data—where no training procedure was applied—is assessed. Figure 7 presents the regression graphs for the test set of seven OML models. The ranking of OML technique performance for the testing dataset is different from that for the training dataset, as shown by a comparison of Figures 6 and 7, and R-squared values also vary between the training and testing datasets. On the testing dataset, SVR and MLP display acceptable performance ($R^2$ = 0.7220) and ($R^2$ = 0.7216), respectively. The performance of RF ($R^2$ = 0.7183), XGB ($R^2$ = 0.7138), and RIDGE ($R^2$ = 0.7128) are next. These are followed by KNN ($R^2$ = 0.6970) and DT ($R^2$ = 0.6838) showing mediocre performances.

### *4.2.3 Results comparison*

Each OML algorithm's mean absolute error, root mean squared error, and R-squared values for training and testing datasets are presented in Table 4. Each technique's performance ranking is also displayed. According to Table 4, KNN and DT, which are ranked first and last among OML models for training data, respectively obtain the uppermost and lowermost R-squared values. Similarly for the testing datasets, SVR and DT are categorized first and last among OML algorithms for testing datasets, respectively. For testing datasets, MLP is ranked 2nd and for the training datasets RF is ranked 2nd. The efficiency of OML procedures on the testing data is more significant to be taken into account as a utilization of each OML algorithm meanwhile the testing data may be seen as an illustration of an actual situation. Considering depth and SPT-N value as independent variables and shear-wave velocity of soil as the dependent variable, a multiple linear regression equation has been developed recently by Tasmiah and Ansary (2022), where obtained $R^2$ was 0.7171 for all soils. Using SVR and MLP ML algorithms, this $R^2$ value has been increased to 0.7220 and 0.7216, respectively. Similarly, for sandy soil (6160 data), the obtained $R^2$ by Tasmiah and Ansary (2022) was 0.6844. In this study through using MLP and XGB ML algorithms, $R^2$ value has been increased to 0.7204 and 0.7103, respectively. For clay soil (3175 data), the obtained $R^2$ by Tasmiah and Ansary (2022) was 0.7428. In this study through using SVR and MLP ML algorithms, $R^2$ value has been increased to 0.7796 and 0.7782, respectively.

### *4.3 Results of variable importance*

The SVR exhibits the top relative efficiency in case of testing data, according to the comparison. In order to evaluate the significance of influencing parameters for the shear-wave velocity estimation of soils, SVR is used. The normalized values for variable importance are displayed in Figure 8. According to Figure 8, depth is the factor that has the biggest impact on estimating the shear-wave velocity of soils (score = 0.5432). The importance value of SPT-N value is 0.4568. That means the two input parameters are almost equally important.

### *4.4 Comparison with existing correlations and ML algorithms*

In this section, previously established shear-wave velocity of soils versus SPT-N value correlations developed by Ohta and Goto (1978), Pérez-Santisteban et al., (2016) and Tasmiah and Ansary (2022) for all soil types have been used to predict shear-wave velocity of soils. Five predicted shear-wave velocity of soils from the above three correlations and two other predicted using SVR and MLP algorithms are then compared with the actual shear-wave velocity of soils (1869 tested values) in Figure 9. Table 4 presents the performance evaluation of the actual values with the three earlier models and the two ML algorithms (SVR and MLP). Figure 10 presents the actual and predicted shear-wave velocity of soils in the testing stages.

Figure 11 displays the residuals plots of the five methods - three existing correlations and two ML algorithms. Figure 11b's negative residuals show that Pérez-Santisteban et al., (2016)'s model over predicts the shear-wave velocity of soils of the dataset. In contrast, the other methods such as Ohta and Goto (1978), Tasmiah and Ansary (2022), SVR and MLP ML algorithms show balanced accuracies, meaning predictions made by these models are relatively impartial and equally distributed above and below the observed values. These show that these four models are doing a fair job of capturing the fundamental patterns and trends in the data.

### *4.5 Discussion*

This study's main advantage is its comparison and proposal of seven improved machine learning (ML) techniques for estimating the shear-wave velocity of soils. The following components of this study add to our understanding of the estimation of shear-wave velocity of soils and other geotechnical engineering fields: For regression problems in geotechnical engineering, (i) the optimized ML approaches are extremely promising, (ii) the stability and resilience of regression procedures may be effectively explored using MAE, RMSE, and R-squared values, (iii) the strategy described in this work holds great promise for expanded use in other geotechnical engineering fields where regression issues are regularly encountered, and (iv) a few guidelines have been given for forecasting the shear-wave velocity of soils by applying ML algorithms. If more data can be collected, the efficiency of the suggested optimized ML models can be enhanced.

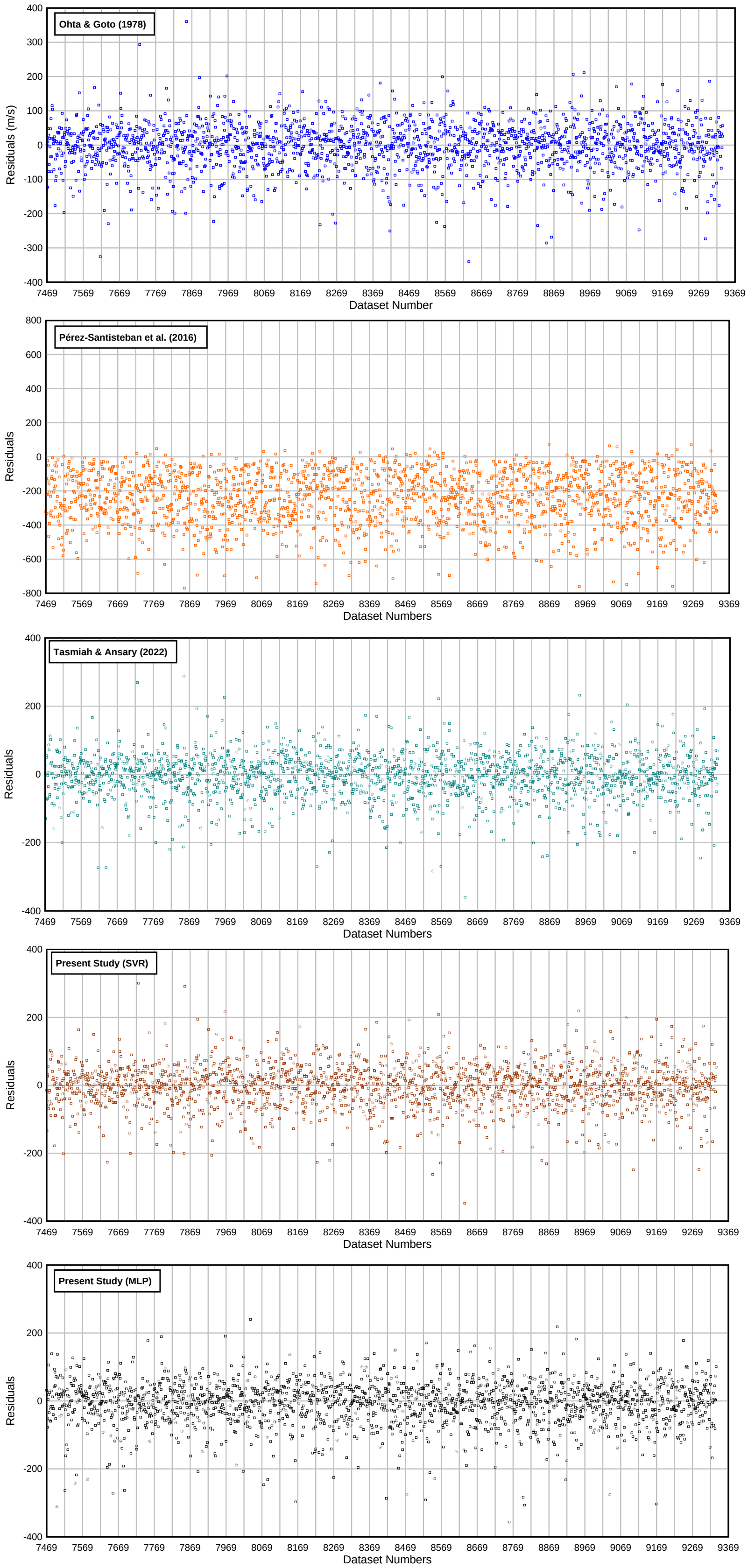

Fig. 11. Relation between residuals and dataset numbers for different predicted shear-wave velocity of soils at testing stages.

## 5. Conclusion and recommendations

When laboratory testing cannot be done, empirical equations are employed to determine the engineering properties of soil. In-situ test findings and soil index features are frequently combined to create empirical relationships that offer cost-effective and non-destructive alternatives. The training dataset that is utilized to construct the algorithm affects how effective it is. In this study, the optimum model for predicting the shear-wave velocity of soils has been identified by a thorough evaluation of seven OML models, comprising DT, KNN, MLP, RF, RIDGE, SVR, and XGB. 9335 pieces of data make up the dataset used by the OML algorithms. As performance measure, a fivefold cross-validation is employed, along with mean average error, root mean square error, and R-squared. Following are some significant deductions:

Randomized search cross-validation (RSCV) algorithm is a useful procedure for tweaking the hyper-parameters of ML models, in line with the optimal scores attained by ML algorithms across iterations.

On the training dataset of 7468 shear-wave velocity of soils and corresponding SPT-N and depth, seven OML algorithms are used. Among the seven OMLs, KNN exhibits a relative better performance ($R^2$ = 0.7542). The performance of RF ($R^2$ = 0.7495), SVR ($R^2$ = 0.7401), XGB ($R^2$ = 0.7394) and MLP ($R^2$ = 0.7381) are next. These are followed by RIDGE ($R^2$ = 0.7178) and DT ($R^2$ = 0.7079) showing relatively mediocre performances. The shear-wave velocities of soils in the testing dataset are also estimated using seven OML algorithms that were trained. Seven OML approaches' performance on 1867 samples from the testing data—where no training procedure was applied—is assessed. The ranking of OML technique performance for the testing dataset is different from that for the training dataset, and that R-squared values also differ between the training and testing data. On the testing dataset, SVR and MLP display acceptable performance ($R^2$ = 0.7220) and ($R^2$ = 0.7216), respectively. The performance of RF ($R^2$ = 0.7183), XGB ($R^2$ = 0.7138), and RIDGE ($R^2$ = 0.7128) are next. These are followed by KNN ($R^2$ = 0.6970) and DT ($R^2$ = 0.6838) showing mediocre performances.

The depth is the factor that has the biggest impact on estimating the shear-wave velocity of soils (score = 0.5432). The importance value of SPT-N value is 0.4568. That means the two input parameters are almost equally important. The efficiency of SVR and MLP is matched with the models of Ohta and Goto (1978), Pérez-Santisteban et al., (2016) and Tasmiah and Ansary (2022) on the testing dataset of the present study. Pérez-Santisteban et al., (2016)'s model over predicts the shear-wave velocity of soils of the dataset. In contrast, the other methods such as Ohta and Goto (1978), Tasmiah and Ansary (2022), SVR and MLP ML algorithms show balanced accuracies, meaning predictions made by these models are relatively impartial and equally distributed above and below the observed values. These show that these four models are doing a fair job of capturing the fundamental patterns and trends in the data.

### References

Bandyopadhyay S, Sengupta A, Reddy GR (2021). Development of correlation between SPT-N value and shear wave velocity and estimation of non-linear seismic site effects for soft deposits in Kolkata city. *Geomech Geoengin* 16:1–19. https://doi.org/10.1080/17486025.2019.1640898

Boore DM, Joyner WB (1997). Site amplifications for generic rock sites, *Bull Seismol Soc Am* 87:327–341. https://doi.org/10.1785/BSSA0870020327

Chapman MC, Martin JR, Olgun CG, Beale JN (2006). Site-Response Models for Charleston, South Carolina, and Vicinity Developed from Shallow Geotechnical Investigations, *Bull Seismol Soc Am* 96:467–489. https://doi.org/10.1785/0120040057

Cornou, C., M. Brax, N. Salloum, M. Rahhal, Farah, Harakeh, J. Harb, D. Y. Abdel-Massih, Armand, Mariscal, C. Voisin, D. Jongmans, P. Bard (2016). Shear-Wave Velocity Structure and Correlation with N-Spt Values in Different Geological Formations in Beirut, Lebanon, *2$^{nd}$ Int. Conf. on Earthquake Engg & Seismology*, Turkey, August, 2016.

Daag, A.S.; Halasan, O.P.C.; Magnaye, A.A.T.; Grutas, R.N.; Solidum, R.U., Jr. (2022). Empirical Correlation between Standard Penetration Resistance (SPT-N) and Shear Wave Velocity (Vs) for Soils in Metro Manila, Philippines, *Appl. Sci.*, 12, 8067. https://doi.org/10.3390/app12168067

Galupino, J., Dungca, J. (2022). Development of a k-Nearest Neighbor (kNN) Machine Learning Model to Estimate the SPT N-Values of Valenzuela City, Philippines, *The 9th AUN/SEED-Net Regional Conference on Natural Disaster (RCND 2021)*. 10.1088/1755-1315/1091/1/012021

Hossain, A., T. Alam, S. Barua & M. R. Rahman (2022). Estimation of shear strength parameter of silty sand from SPT-N60 using machine learning models, *Geomechanics and Geoengineering*, 17:6, 1812-1827, doi: 10.1080/17486025.2021.1975048

Kim, K., Park, H., Goo, T., Kim, H. (2020). A Prediction of N-value Using Artificial Neural Network, *The Journal of Engineering Geology*, 30(4): 457-468. https://doi.org/10.9720/kseg.2020.4.457

Klimis, N.S., Margaris, B.N., Koliopoulos, P.K. (1999). Site-Dependent Amplification Functions and Response Spectra in Greece, *J Earthq Eng* 3:237–270. https://doi.org/10.1080/13632469909350346

Kuo C-H, Wen K-L, Hsieh H-H, et al (2011). Evaluating empirical regression equations for Vs and estimating Vs30 in northeastern Taiwan, *Soil Dyn Earthq Eng* 31:431–439. https://doi.org/10.1016/j.soildyn.2010.09.012

Lee SH-H (1992). Analysis of the Multicollinearity of Regression Equations of Shear Wave Velocities, *Soils Found* 32:205–214. https://doi.org/10.3208/sandf1972.32.205

Lu C-C, Hwang J-H (2020). Correlations between Vs and SPT-N by different borehole measurement methods: effect on seismic site classification, *Bull Earthq Eng* 18:1139–1159. https://doi.org/10.1007/s10518-019-00767-1

Motahari, M.R., Amini, O., Khoshghalb, A. Etemadifar, M., Alali, N. (2022). Investigation of the Geotechnical Properties and Estimation of the Relative Density from the Standard Penetration Test in Sandy Soils (Case Study: North East of Iran), *Geotech Geol Eng* 40, 2425–2442 (2022). https://doi.org/10.1007/s10706-021-02036-y

Ohta Y, Goto N (1978). Empirical shear wave velocity equations in terms of characteristic soil indexes, *Earthq Eng Struct Dyn* 6:167–187. https://doi.org/10.1002/eqe.4290060205

Pérez-Santisteban I., A. M. Martín, A. Carbó, J. M. Ruiz-Fonticiella (2016). Empirical Correlation of Shear Wave Velocity (Vs) with SPT of Soils in Madrid, *first break*, 34: 87-92. 10.3997/2214-4609.201601971

Sil A, Haloi J (2017). Empirical Correlations with Standard Penetration Test (SPT)-N for Estimating Shear Wave Velocity Applicable to Any Region, *Int J Geosynth Ground Eng* 3:22. https://doi.org/10.1007/s40891-017-0099-1

Tasmiah, A., Ansary, M.A. (2023). Development of Correlation Equations Between Shear Wave Velocity and Standard Penetration Test Values for Different Soil Types for DMDP Area of Bangladesh Using Multivariate Analysis, *Indian Geotech J* 53: 665–677 https://doi.org/10.1007/s40098-022-00698-w

Thokchom S, Rastogi BK, Dogra NN, et al (2017) Empirical correlation of SPT blow counts versus shear wave velocity for different types of soils in Dholera, Western India. Nat Hazards 86:1291–1306. https://doi.org/10.1007/s11069-017-2744-3

Wang S-Y, Wang H-Y (2016). Site-dependent shear-wave velocity equations versus depth in California and Japan, *Soil Dyn Earthq Eng* 88:8–14. https://doi.org/10.1016/j.soildyn.2016.05.001