Ingénierie des Systèmes d'Information
Vol. 31, No. 10, October, 2026, pp. 3331-3343
Journal homepage: http://iieta.org/journals/isi

Evaluation of Support Vector Machines and K-Nearest Neighbors for Soil Classification under the Unified Soil Classification System

Saja A. Hussein[1*], Ihsan Al-Abboodi[1,2], Ammar S. Dawood[1]

[1] Department of Civil Engineering, College of Engineering, University of Basrah, Basrah 61001, Iraq
[2] Department of Civil Engineering, Shatt Al-Arab University College, Basrah 61001, Iraq

Corresponding Author Email: saja.abdulkadhim@uobasrah.edu.iq

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/isi.311008

Received: 10 April 2026
Revised: 21 August 2026
Accepted: 4 September 2026
Available online: 31 October 2026

Keywords:
soil classification, Unified Soil Classification System, machine learning, Support Vector Machine, K-Nearest Neighbors, Atterberg limits, grain-size distribution

ABSTRACT

Reliable soil classification is essential to geotechnical investigation and design, yet conventional classification depends on laboratory measurements and the interpretation of grain-size distribution and Atterberg limits. This study evaluates Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) classifiers for automated soil classification under the Unified Soil Classification System (USCS). A dataset comprising 478 soil samples compiled from published studies, site-investigation reports, and an open-access repository was used to represent seven USCS classes: clayey gravel (GC), silty sand (SM), clayey sand (SC), silt of low plasticity (ML), clay of low plasticity (CL), elastic silt (MH), and fat clay (CH). Eleven classifier variants, including six SVM and five KNN models, were assessed using 12 combinations of routinely available soil parameters. Model performance was examined using 90:10 and 80:20 training–test splits, with five-fold cross-validation applied to the training data. The highest test accuracy in the 90:10 split was 97.9%, achieved by Cubic SVM with feature group G_4 and Fine KNN with G_2. In the 80:20 split, Cubic SVM achieved 96.9% accuracy with G_2 and G_4, while Fine KNN reached 94.8% with G_8. The results further showed that combinations including the percentage passing the No. 200 Sieve and Atterberg limits generally provided better classification performance than feature sets lacking plasticity information. These findings indicate that SVM and KNN can serve as supplementary tools for USCS classification when the input parameters fall within the range represented by the dataset. However, the limited dataset size and geographic coverage, together with the use of a single random seed, restrict claims regarding broader generalizability. Further validation using independent datasets from different geological and geographical settings is therefore necessary.

1. INTRODUCTION

Soil classification is an essential part of geotechnical engineering and forms the basis for designing foundations, retaining walls, embankments, pavements, and earth dams [1]. Soil type defines the strength, compressibility, permeability, and swell potential values that go into bearing capacity, settlement, and slope stability calculations [2]. Any errors in soil classification can lead to problems in structural design and eventually can cause expensive failures or project delays and thus make a solid and reproducible soil characterization a must [2].

The Unified Soil Classification System (USCS) developed by Casagrande and standardized by ASTM D2487 [3] is the most popular system for soil classification. It classifies soil types according to their grain size and Atterberg limits (the liquid limit (LL), plastic limit (PL), and plasticity index (PI)) [4]. Even though the USCS is widely used, the system is based on the laboratory sieve analysis and Atterberg limit tests, which are time consuming, resource intensive and prone to subjective errors [5]. This makes the USCS less desirable for larger-scale projects and initial rapid surveys.

Driven by increased computational power and large geotechnical datasets, the application of machine learning has been explored as a supplement to laboratory tests [6]. Machine learning algorithms can detect complex non-linear dependencies between soil parameters and make predictions with no assumption of physical models [7]. Machine learning algorithms have already been used in the area of geotechnical engineering for prediction of shear strength [8], settlement analysis, slope stability [2], soil improvement techniques [9], etc., making machine learning algorithms an excellent choice for solving various geotechnical engineering problems.

Among the supervised machine learning (ML) algorithms, the Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) algorithms are most commonly used in various studies. SVM finds the separating hyperplane with maximum margin between class boundaries within a higher-dimensional feature space [10]. With kernels (linear, polynomial, and Gaussian/radial basis function (RBF)), SVM deals with non-linearly separable cases well and has been shown to be efficient in remote sensing data classification and land cover mapping [11, 12]. KNN is a non-parametric instance-based learning algorithm that predicts the class of a data point based

on the majority vote of its KNN in a feature space [13]. The algorithm's performance depends on the choice of *k* and distance metric [14], but due to its simplicity, it can be used as a baseline to compare the effectiveness of the proposed complex ML model.

As the application of ML to geotechnical issues continues to be explored, there are still some research gaps concerning the ML-based classification of soils using the USCS. First, previous studies have mostly focused on one particular ML algorithm or a set of a few input features, and thus the models in the same category cannot be compared with one another for a specific scenario [15]. Second, it remains unknown as to which small number of commonly available geotechnical parameters of the soil should be used to achieve an accurate classification. Third, the performance of an ML-based classifier depends on the ratio between training and testing data, an aspect which was rarely explored in past research and which is important when the dataset is small. The current study seeks to tackle these research gaps by evaluating various versions of SVM and KNN algorithms with respect to twelve possible input features and two training–testing splits based on some typical laboratory parameters as required for the USCS system.

From a literature review, it was evident that different researchers had different approaches and reported different results. For example, Chala and Ray [15] reported the use of Gradient Boosting Classifier, Random Forest (RF), Logistic Regression, KNN, SVM, and Artificial Neural Network (ANN) for classification of soils using cone penetration test (CPT) data with an accuracy of 93% in the best case where only the CPT cone resistance and sleeve friction were used as the feature variables but these are not commonly used soil parameters that are not used by the practicing geotechnical engineer. Noi and Kappas [11], as well as Adugna et al. [12] applied SVM, KNN, and RF to classify lands in their study for remote sensing applications; they noted that the SVM had a good accuracy even for a small set of training data. Ahmad et al. [16] used Decision Tree Classifier, Naive Bayes, and SVM for the assessment of seismic soil liquefaction; Saad et al. [9] used an ensemble-based method for the prediction of soil improvement, and comparisons of the different methods were made in other literature reviews. None of these studies, however, (i) explored the variation in the performance of ML classifiers depending on the choice of the set of input features used as inputs to the ML model for many combinations of variables (i.e., representing different stages in a site investigation) nor (ii) made a comparative evaluation of different SVM and KNN algorithms under similar scenarios when used for the classification of soils with reference to the USCS system nor (iii) evaluated the effect of the ratio of training/testing data on the classification accuracy of an ML algorithm. All these research gaps in the past literature are being addressed in the present work.

The purpose of this paper is to report the performance of the SVM and KNN classifiers for soil classification using the USCS in comparative benchmarking using controlled and systematic experiments. In particular, this paper does not offer any new variation to SVM and KNN algorithms; but the proposed contribution of this paper is in the form of a comparative framework.

In this study, eleven different classifier variants were tested in terms of input variables and split ratio in two ways that yielded significant knowledge for soil scientists and practicing engineers on how best to select parameters (of routinely available soil parameters) for reliable automated soil classification. The results from this study demonstrate that it is sufficient to use three parameters (in terms of moisture content, percent passing the No. 200 Sieve, and Atterberg limits) as input parameters of SVM for accurate soil classification (with 95.8% accuracy), while KNN requires only two (in terms of percent passing the No. 200 and No. 4 Sieve) to obtain high classification accuracy (93% for KNN-13). Furthermore, the results indicate that the performance of both SVM and KNN classifiers is comparable across the two evaluated training–test split ratios (90:10 and 80:20). This benchmarking exercise is important in applied ML for civil and geotechnical engineering [6, 7, 9]:

(1). To create and evaluate the performance of multiple SVM and KNN models for soil classification using a set of input parameters as per USCS.

(2). To compare the performance of these models for 12 different sets of input combinations to determine the key parameters that influence the accuracy of the predicted classes.

(3). To investigate the effect of different test on the total data ratio (10% and 20% test data) on the performance and robustness of the developed models.

(4). To compare the computed classification accuracy of the proposed models with previously published results to demonstrate the suitability and reliability of the model.

2. MATERIALS AND METHODS

2.1 Database development and description

The selection of a representative and broad dataset is arguably the most significant consideration in an ML-based study. Therefore, in this study, a database of 478 soil samples was constructed based on published geotechnical site investigation reports, open literature, journal datasets, and laboratory data, following standard practice [6, 9]. These datasets are mainly the results of standard laboratory tests (i.e., grain size by sieve analysis ASTM D422 and Atterberg limits ASTM D4318). There are no field/in-situ data in the database.

The dataset comprises seven soil classes under the USCS: clayey gravel (GC), silty sand (SM), and clayey sand (SC) among coarse-grained soils, and silt of low plasticity (ML), clay of low plasticity (CL), elastic silt (MH), and fat clay (CH) among fine-grained soils. The database was assembled to ensure that the number of samples in each soil class did not differ substantially, thereby minimizing potential bias in the ML classifier. These data include the following parameters: percentage passing No. 4 Sieve, percentage passing No. 200 Sieve, gravel content (%), sand content (%), and fines content (%), LL, PL, and PI, which is generally sufficient information required to determine the USCS [3].

The summary of the statistical properties of the database is presented in Table 1, in terms of mean, standard deviation, and range of each variable. It can be seen that high standard deviations exist for some of the parameters, such as Gravel, Sand, and Fine content, which suggests a high scatter in the values of these soil properties. High scatter in the values is required for developing an unbiased machine learning model that can classify various types of soils.

2.1.1 Data sources and composition of the 478 soil samples

To improve the transparency and reproducibility of the present study, Table 2 summarizes the specific references and

data sources from which the 478 soil samples were compiled, together with the approximate number of samples obtained from each source. Because the primary sources served different research objectives, their raw geotechnical parameters were systematically extracted and reclassified according to the USCS specifically for the purpose of this study.

The four sources collectively span most of the major USCS classes considered in this study (GC, SM, SC, ML, CL, MH, CH) and a wide range of index properties.

Table 1. The database used in the study

Variable	Mean	Standard Deviation	Minimum	Median	Maximum
Passing No. 4 Sieve	92.75	11.28	42	98	100
Passing No. 200 Sieve	66.33	22.58	12.3	68	100
Gravel	7.252	11.28	0	2	58
Sand	26.42	17.7	0	26	78.3
Fine	66.33	22.58	12.3	68	100
Liquid limit (LL)	50.03	14.38	0	51	89.9
Plastic limit (PL)	25.54	8.19	0	24.15	56.94
Plasticity index (PI)	24.58	11.54	0	23	54

Table 2. Breakdown of the 478 soil samples by data source

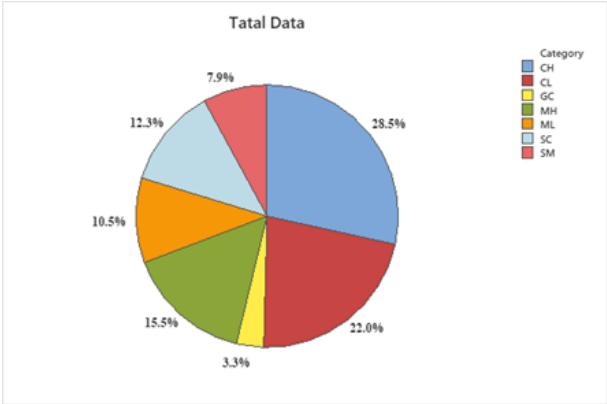
Source	Reference	Category / Description	Number of Samples
1	[10]	Construction management soil classification study	235
2	[17]	Machine learning soil classification study	31
3	[18]	Regional geographic information system (GIS) spatial interpolation study (Kirkuk, Iraq)	25
4	[19]	Open-access Kaggle repository (ConcreteSet)	187
		Total	478

2.2 Data preprocessing and exploratory analysis

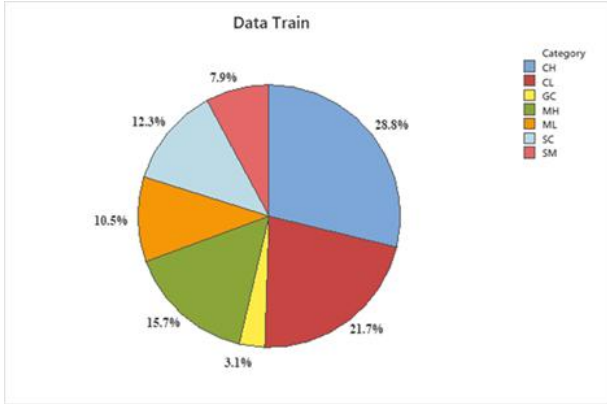
Before training the model, an exploratory data analysis (EDA) was performed to analyze the data structure, distribution, and correlation. To check the generalization capacity of the models, the dataset was split into training and testing datasets. In this study, two scenarios were considered, i.e., 90% training and 10% testing (N_{train} 430, N_{test} 48) and 80% training and 20% testing (N_{train} 382, N_{test} 96). The holdout split was performed using MATLAB's default random number generator with rng(0) (seed = 0) to evaluate the model performance for different scenarios, which is often the case in geotechnical problems [20]. The proportion of classes in the total and testing data was maintained to evaluate the performance of the model, as shown in the pie charts of the total, training, and testing data (Figures 1(a), 1(b), and 1(c)), whereas Figure 1(d) shows the number of each class used in the study.

Recognizing that some of these inputs are not independent, it should also be noted that Gravel (%), Sand (%), and Fine (%) by nature are dependent on each other because their sum equals 100%. In addition, it should also be noted that the PI can be calculated from LL and PL (PI = LL − PL); therefore, there is a degree of dependency between the LL, PL, and PI.

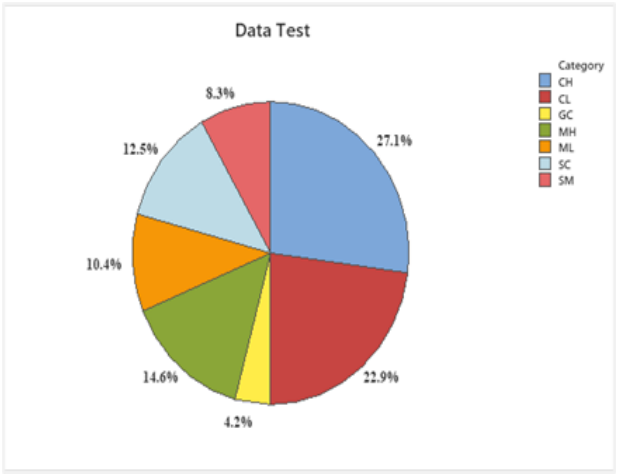
These dependencies were accounted for when selecting the twelve input feature groups to be utilized in this work. For example, in order to avoid these dependencies (or collinearities), if Gravel, Sand, and Fine are included in an input variable group, then all three sieve passing percentages will not be included. Similarly, if the PI is used as an input, it is not combined with both LL and PL at the same time. The input groups have been designed to ensure that the combination of input variables used to estimate the classification of a soil is practically significant and that the results are not merely due to the multicollinearity between variables. It should be noted, however, that although multicollinearity is primarily a concern for linear regression-type models, the presence of redundant variables can still influence the classifiers used in this study: in KNN, redundant features contribute directly to the Euclidean distance calculation and may distort the identification of nearest neighbors, while in kernel-based SVM the presence of correlated features changes the effective geometry of the kernel-induced feature space. The twelve input groups were therefore designed to minimize redundancy, rather than to assume that redundancy has no effect.



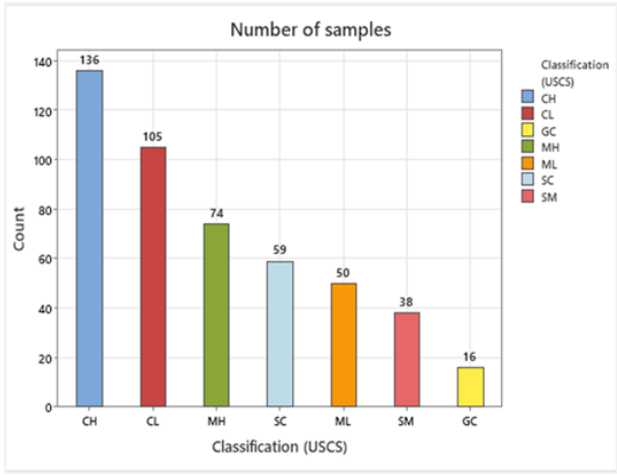
(a) Pie chart of classification (Unified Soil Classification System)



(b) Pie chart of training data



(c) Pie chart of test data



(d) Number of each soil class used in the study

Figure 1. Class distributions in the (a) complete dataset, (b) training set, and (c) test set, and (d) number of samples in each soil class

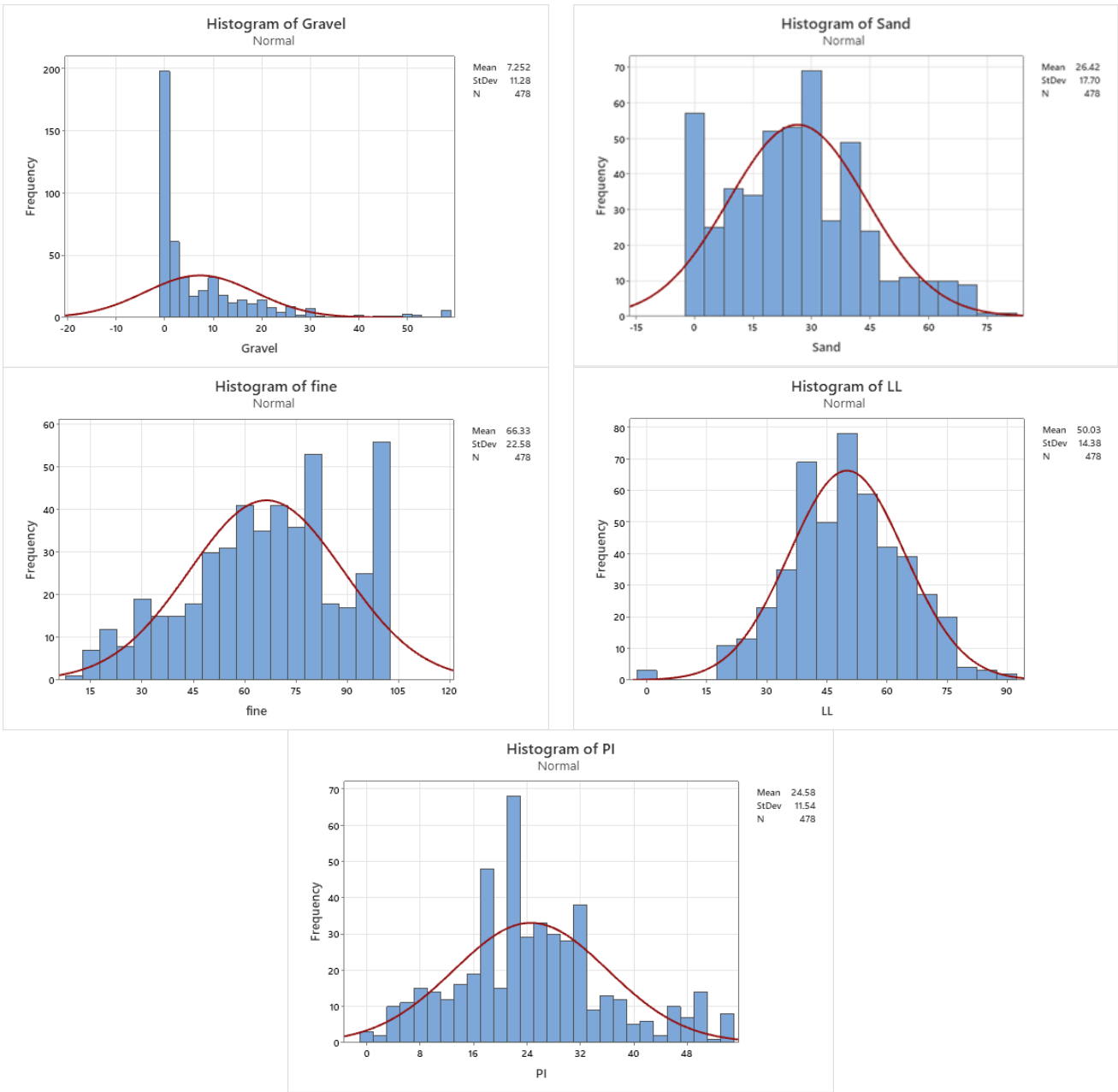


Figure 2. Histograms of the primary soil variables used in this study

Visualizations were key to the EDA process. Histograms for the primary variables (Gravel, Sand, Fine, LL, PI) were generated to examine their frequency distributions (Figure 2). For instance, the histogram for 'Gravel' shows a strong right skew, with most samples having low gravel content, while 'Fine' and 'LL' exhibit distributions that are closer to normal. A boxplot (Figure 3) was used to visualize the distribution, median, quartiles, and potential outliers for each variable simultaneously. This plot clearly shows the wide range of values for 'Sand' and 'Fine' content, reinforcing the statistical findings in Table 1.

Additionally, a correlation matrix was produced to explore relationships between variables (Figure 4), highlighting correlations that fit expectations for soil materials. Consistent with the mathematical definition, the correlation coefficient of -1.00 between 'Gravel (%)' and 'Passing No. 4 Sieve (%)' arises from the inverse relationship between these two variables (i.e., Gravel (%) = 100% - Passing No. 4 Sieve (%)). Similarly, the large negative correlation between 'Gravel (%)' and 'Passing No. 200 Sieve (%)' is logical, as the former indicates materials that, by definition, should contain fewer fine-grained particles.

The correlation between 'Fine' content and 'Passing No. 200 Sieve' is, by construction, exactly 1.00 in the curated dataset, because these two variables are operationally defined as the same physical quantity (the percentage of soil particles smaller than 0.075 mm).

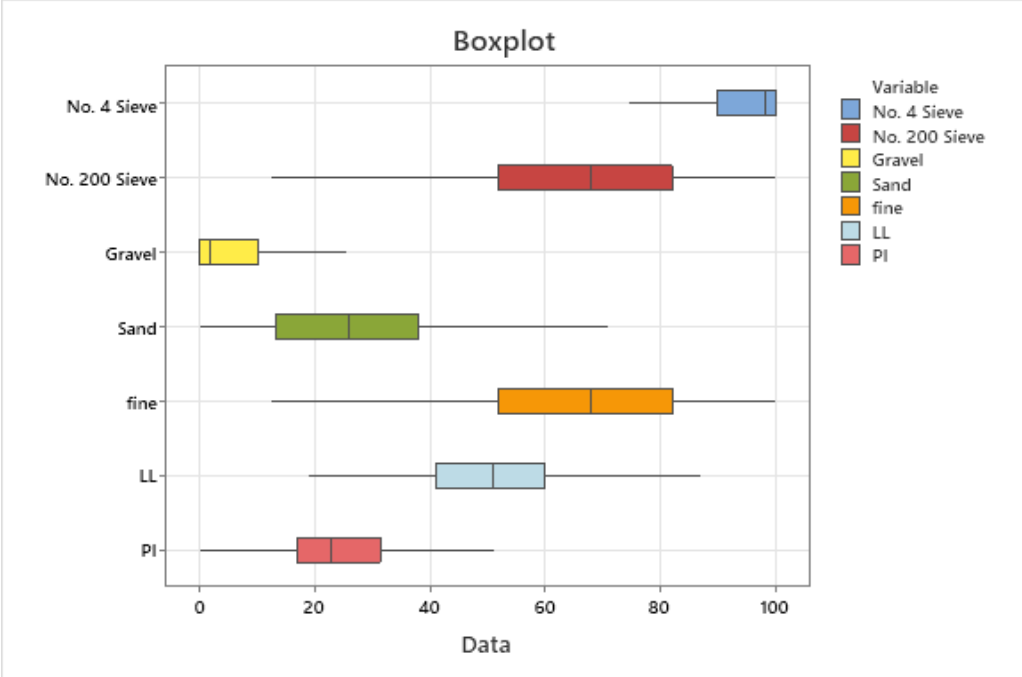


Figure 3. Boxplot of the percentage passing the No. 4 sieve, percentage passing the No. 200 Sieve, gravel content, sand content, fines content, liquid limit (LL), and plasticity index (PI)

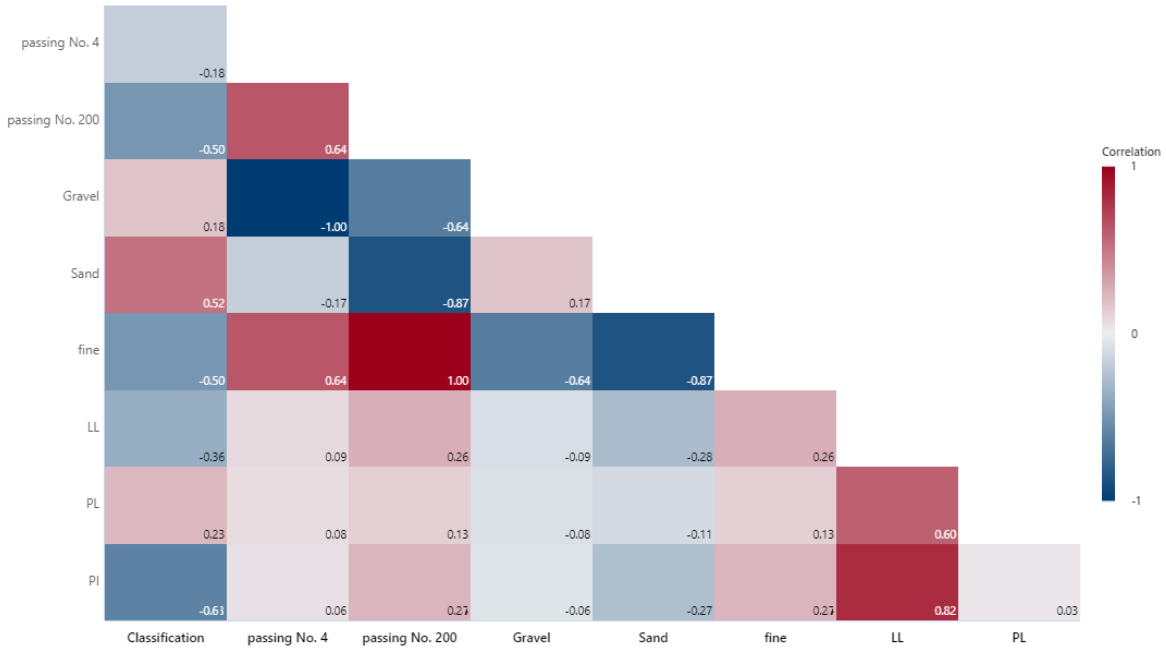


Figure 4. A correlation matrix plot for the data of this study
Note: LL = liquid limit; PL = plastic limit; PI = plasticity index.

2.3 Machine learning models

This study focuses on two powerful and widely used supervised machine learning algorithms for classification: SVM and KNN.

2.3.1 Support Vector Machine

SVM is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane that categorizes new examples. For a two-class learning task, the goal of SVM is to find the hyperplane that maximizes the margin between the two classes, where the margin is defined as the distance between the hyperplane and the nearest data points from either class, known as support vectors [21]. For data that is not linearly separable, SVM uses the "kernel trick" to map the input data into a higher-dimensional feature space where a linear separation is possible. This study evaluated several SVM variants available in the MATLAB Classification Learner, each employing a different kernel function:

•**Linear SVM**: Uses a linear kernel, suitable for linearly separable data.

•**Quadratic and Cubic SVM**: Employ polynomial kernels of the second and third degree, respectively, allowing for more complex, curved decision boundaries.

•**Gaussian SVM (Medium Gaussian)**: Employ an RBF kernel $K(x, x') = \exp(-\|x - x'\|^2 / (2\sigma^2))$, with the kernel scale σ set to the median of pairwise distances of the training samples (MATLAB's Classification Learner default value for 'Medium Gaussian'). A 'medium' bandwidth of σ yields a compromise between a small σ (risking overfitting) and a large σ (risking underfitting).

•**Kernel SVM**: A general term for SVMs using kernel functions, often referring to the RBF kernel in this context.

The versatility of these kernels makes SVM a powerful tool for the complex, multi-class problem of soil classification [12].

2.3.2 K-Nearest Neighbors

KNN is a non-parametric, instance-based learning algorithm where the function is only approximated locally and all computation is deferred until classification [13]. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its 'k' nearest neighbors. The choice of 'k' is a critical hyperparameter; a small 'k' can make the model sensitive to noise, while a large 'k' can be computationally expensive and may oversmooth the decision boundary [14]. The second critical parameter is the distance metric, which is often Euclidean. The article considers several variants of KNN algorithms:

•**Fine KNN (k = 1), Medium KNN (k = 10), and Coarse KNN (k = 100)**: The only difference between these methods is the value of k, the number of neighbors employed. Fine KNN (k = 1) gives low-bias (but high-variance) decision boundaries; Medium KNN (k = 10) gives a compromise; and Coarse KNN (k = 100) gives more regularized boundaries. They all use the Euclidean distance.

•**Cosine KNN**: uses cosine similarity for distance measurement, which is useful when the magnitude of feature vectors does not matter.

•**Weighted KNN**: weights the contributions of the neighbors, so that the closer neighbors will have more weight in the average compared to the further neighbors. The weighted average can often provide better accuracy than a simple majority vote.

KNN is relatively simple and easy to understand. Hence, it is used for benchmarking against more complex algorithms like SVM [18].

2.4 Model implementation and evaluation

All SVM and KNN models were trained using the Classification Learner app in MATLAB R2024a.

For the SVM classifier, the "One-vs-One" approach was used for every kernel type. SVM models used a box constraint (regularization parameter C) of 1.0 for all kernel types, which is the default value in the MATLAB app as well as a common starting point for this value in the literature [20]. In Polynomial SVM models (i.e., Quadratic and Cubic), the polynomial order was 2 and 3, respectively, and no scaling was used. All predictor variables were z-score standardized (zero mean, unit variance) prior to training; this standardization is applied automatically by the MATLAB Classification Learner app and is essential for distance-based methods (KNN) and kernel-based methods (SVM), which are sensitive to differences in the scales of the input variables.

For KNN, the following values for the k-neighbor number were used: 1 for Fine KNN, 10 for Medium KNN, 100 for Coarse KNN, 10 for Cosine KNN (cosine distance), and 10 for Weighted KNN (inverse distance weighting). The values for the hyperparameters selected from the app followed the benchmarks found in other published studies [14, 20] and were the same for all variable groups to provide an equal comparison between variable groups and models.

There were 12 variable groups, G_1 to G_{12}, as shown in Table 3, to investigate the effects of variable sets on classification performance.

These groups include a full variable set (e.g., G_3 and G_4) as well as a reduced variable set, which may be obtained during the initial stages of an investigation (e.g., G_5 and G_6), and is helpful in determining the minimum number of variables for acceptable classification.

Table 3. Groups of variables used for model training

Group	Variables
G_1	Passing No. 4 Sieve, Passing No. 200 Sieve, LL, PI
G_2	Passing No. 4 Sieve, Passing No. 200 Sieve, LL, PL
G_3	Gravel, Sand, Fine, LL, PI
G_4	Gravel, Sand, Fine, LL, PL
G_5	Passing No. 4 Sieve, Passing No. 200 Sieve, LL
G_6	Passing No. 4 Sieve, Passing No. 200 Sieve, PI
G_7	Passing No. 4 Sieve, Passing No. 200 Sieve, PL
G_8	Passing No. 200 Sieve, LL, PI
G_9	Passing No. 200 Sieve, LL, PL
G_{10}	Gravel, Sand, Fine, LL
G_{11}	Gravel, Sand, Fine, PI
G_{12}	Gravel, Sand, PL

Note: LL = liquid limit; PL = plastic limit; PI = plasticity index.

Primarily, the models were assessed on the metric of classification accuracy, which is the number of correct predictions out of the total predictions. In the results tables shown, "Validation Accuracy" and "Test Accuracy" are the two different terms that appear. The "Validation Accuracy" was computed by performing 5-fold cross-validation on the training data. The training data is split into five parts, and the model is trained on four parts while validated on the fifth part.

This is done five times to provide a more robust estimate of the out-of-sample performance of the fixed-configuration models [22]. The hyperparameters were set a priori to the MATLAB Classification Learner default values. The final "Test Accuracy" was computed on the test set (10% or 20% of the total data), which has not been used to train or validate the model and is the closest representation of how the model will perform on new data drawn from a similar population.

The test-set sizes in this work are 48 and 96 observations (for the 10% and 20% splits, respectively), so small differences in results may not be statistically significant. For a binomial distribution, the 95% confidence interval for a proportion with an n of 96 would be ±4.3% for a proportion of 0.95; therefore, any differences smaller than ±4% should be considered in context.

Because of this, the results discussion does not dwell on specific comparisons of model results but rather highlights broader patterns across groups of models. When a larger and independent test set becomes available, statistical significance tests such as McNemar's test could be applied to provide greater certainty to the pairwise comparisons of models.

3. RESULTS

In this part, the results of the applied machine learning models for the classification of soils are shown. The results are categorized according to the ratio of the split data set and the categories of the input variables.

3.1 Model performance with 10% test data (48 samples)

First, model evaluation was done on a split of 90% training with 10% testing. The performance of different SVM variants (six in total) and KNN variants (five in total) was evaluated using the first four variable sets (G_1 to G_4) as an example. The detailed accuracy results of both the validation and testing phases are displayed in Table 4.

Table 4. Accuracy results from MATLAB learner for Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) models (10% test split)

		G_1		G_2		G_3		G_4	
Model	**Classifier**	**Validation Accuracy (%)**	**Test Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**
SVM	Quadratic SVM	92.1	95.8	90.7	93.8	92.3	95.8	91.9	95.8
	Cubic SVM	91.9	95.8	92.8	95.8	92.3	95.8	93.3	**97.9**
	Linear SVM	91.6	91.7	90.5	93.8	91.6	91.7	91.2	91.7
	Efficient linear SVM	90	89.6	88.8	91.7	96	93.8	96	93.8
	Medium Gaussian	89.1	93.8	90.9	95.8	88.8	93.8	89.8	93.8
	Kernel SVM	89.5	93.8	90.9	95.8	89.1	95.8	91.4	91.7
KNN	Weighted KNN	89.1	89.6	91.6	95.8	87.9	87.5	91.9	95.8
	Fine KNN	88.4	95.8	90.9	97.9	88.4	91.7	91.9	91.7
	Medium KNN	80.7	83.3	84.4	85.4	82.1	85.4	84.4	87.5
	Cubic KNN	80	81.2	85.1	83.3	80.9	83.3	83.5	87.5
	Cosine KNN	78.1	79.2	81.4	81.2	81.4	83.3	82.6	89.6

Table 5. Accuracy results from the MATLAB learner for Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) models (20% test split, G_1–G_4)

		G_1		G_2		G_3		G_4	
Model	**Classifier**	**Validation Accuracy (%)**	**Test Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**	**Validation Accuracy (%)**	**Test Accuracy (%)**
SVM	Quadratic SVM	91.6	94.8	92.1	93.8	91.9	94.8	89.8	92.7
	Cubic SVM	91.9	93.8	91.9	**96.9**	92.4	92.7	91.9	**96.9**
	Linear SVM	89.3	91.7	90.1	92.7	90.6	92.7	90.3	90.6
	Efficient linear SVM	90.1	88.5	86.4	88.5	96.3	92.7	95.8	92.7
	Medium Gaussian	87.2	93.8	90.3	95.8	88.2	94.8	88.5	94.8
	Kernel SVM	88.7	91.7	90.6	94.8	89	91.7	90.3	**96.9**
KNN	Weighted KNN	89.8	91.7	92.1	95.8	89.3	90.6	92.1	95.8
	Fine KNN	88	92.7	91.9	94.8	88	91.7	91.9	93.8
	Medium KNN	79.3	83.3	82.5	88.5	80.4	84.4	81.9	89.6
	Cubic KNN	78.8	83.3	81.2	88.5	78.5	84.4	81.7	89.6
	Cosine KNN	77.2	82.3	81.4	82.3	77.5	83.3	78.3	85.4

From the results, the following points are noted: The best accuracy for Cubic SVM is 97.9%, which is for the variable group G_4 (Gravel, Sand, Fine, LL, PL). This means that the prediction using the particle size distribution along with both Atterberg limits is the best. The best accuracy for Fine KNN is also 97.9%, which is for the variable group G_2 (Passing No. 4, Passing No. 200, LL, PL). This implies that the best prediction using KNN is with the direct results of sieve analysis along with Atterberg limits. Quadratic and Cubic SVM are generally better than Linear SVM. Fine and Weighted KNN are generally better than Medium and Cubic KNN. Cosine KNN is the worst, especially for the variable group G_1.

3.2 Model performance with 20% test data (96 samples)

In order to measure the models' performance with a larger test set and analyze their robustness, the analysis was recommenced with an 80% training and 20% testing split. This stage included all twelve variable groups (G_1 to G_{12}), offering a more comprehensive examination of feature importance.

3.2.1 Performance of groups G_1 to G_4

The performance of the models on the first four variable groups with the 20% test split is summarized in Table 5. A summary of the best-performing models for each group is presented in Figure 5.

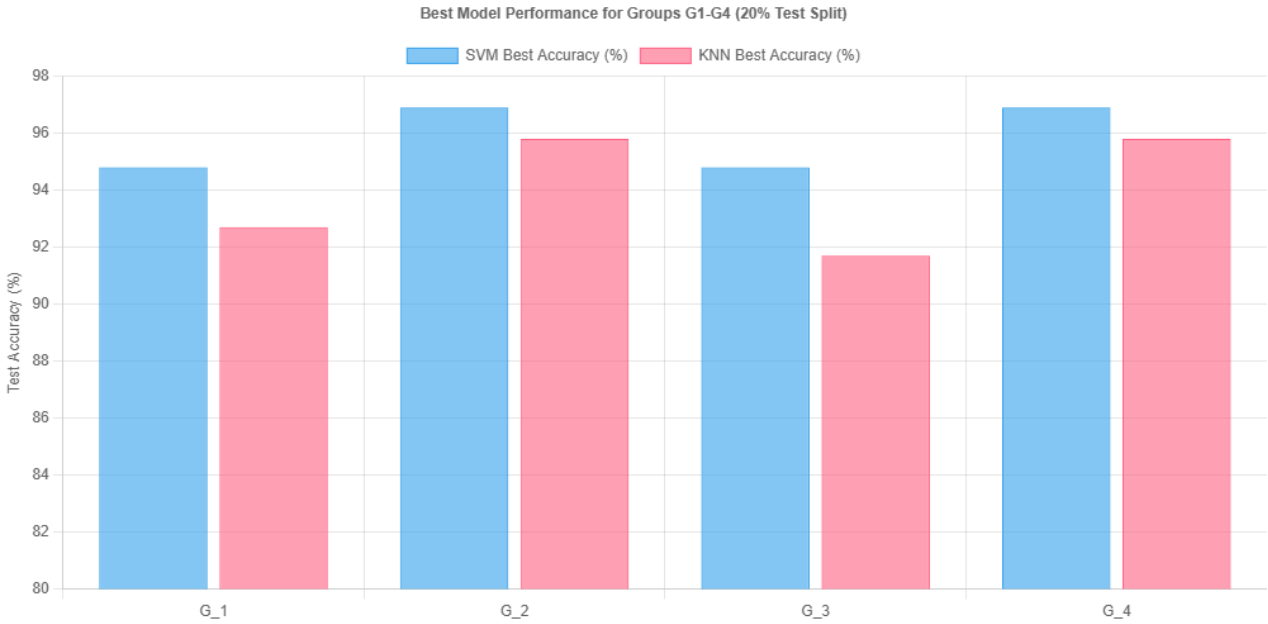


Figure 5. Best test accuracy for Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) models across groups G_1 to G_4 (20% test split)

With the larger test set, the Cubic SVM model again demonstrated exceptional performance, achieving the highest test accuracy of 96.9% for both groups G_2 and G_4. The Kernel SVM model also reached 96.9% accuracy for group G4. This reinforces the finding that combining particle size data (either from sieves or as Gravel/Sand/Fine percentages) with Atterberg limits (LL and PL) provides the most robust feature set for classification. The KNN models, while still performing well, did not reach the peak accuracies of the SVM models in this test split. The Weighted KNN model showed strong and consistent performance, achieving 95.8% accuracy for groups G_2 and G_4.

3.2.2 Performance of groups G_5 to G_{12}

The second phase of the evaluation presents the classification accuracy of the models on variable groups (G_5 to G_{12}). In this phase, the 20% test set (96 samples) was used to evaluate the effect of the absence of one or more influential parameters. The classification results of the top two performing SVM and KNN models for each variable group are given in Table 6.

The accuracy of the models drastically reduces for variable groups. For G_5 (excluding plasticity), the best SVM (Medium Gaussian) and KNN (Fine KNN) classifiers resulted in a test accuracy of only 78.1 and 77.1, respectively. For G_6 (excluding LL), the test accuracy of the Cubic SVM and Weighted KNN models was found to be 70.8 and 79.2, respectively. These results underscore the critical importance of including both particle size distribution and plasticity characteristics for accurate USCS classification.

For other variable groups, the classification results were found to be significantly improved. G_8 and G_9, which include a combination of Passing No. 200 Sieve and Atterberg limits, show excellent results. For G_8, the test accuracy of the Fine KNN and Cubic SVM models was found to be 94.8 and 91.7, respectively.

For G_9, the Cubic SVM and Weighted KNN models resulted in test accuracies of 93.8 and 92.7, respectively. These results are similar to the results obtained with full sets of features in G_1 to G_4. Therefore, these sets of features are important combinations for classification. Figure 6 shows the confusion matrices for the test data of the feature groups G_5–G_9 by the best KNN models. We can observe the density of correctly classified samples along the diagonal for groups G_8 and G_9. This indicates that the classification performance is better when the Passing No. 200 Sieve data are combined with the Atterberg limits. On the other hand, the overlap of classes is high for groups G_5 and G_6. This indicates that the predictive performance decreased when some important soil classification features were missing.

SVM with a Cubic kernel was always the winner among the groups from G_5 through G_{12}. Within the KNN family, Fine and Weighted KNN always performed better and more consistently than other models in this family, i.e., Medium, Coarse, and Cosine KNN. This is because USCS class boundaries in the feature space tend to form a rather compact and locally clustered structure. Therefore, voting in favor of low-k nearest neighbors with distance weighting was better than that for broad voting.

Interestingly, in groups G_7, G_8, and G_{12}, the top KNN performed slightly better than the top SVM. Geotechnically speaking, groups G_8 and G_{12} contain the Passing No. 200 Sieve as well as Atterberg limits, which are essentially the two axes in the plasticity chart. As a result, the two-dimensional space used by the model is the same as that of the plasticity chart in the USCS, where the USCS class boundary can be simply determined by relatively easy criteria such as LL = 50 and A-line PI = 0.73(LL–20). Therefore, low-k nearest neighbors in KNN can capture the local neighborhood well enough so that the model can make an accurate prediction.

Geotechnically speaking again, the reason the accuracy of the model remained approximately the same between G_7 and G_8 and between G_{11} and G_{12} when additional sieve-passing percentages or derived Gravel/Sand/Fine percentages were added is that only two of the added values (the Passing No. 200 Sieve and the Atterberg Limits) are enough to determine the class.

The confusion matrices of the top models (Figure 7) allow us to identify more granular information beyond that provided by accuracy values. In the case of full feature groups (G_1 to G_4), all the USCS classes have extremely high precision. Most misclassifications are between USCS categories that are closely related: for example, between SM and SC, or ML and CL. This type of confusion between boundaries could be expected since the USCS classification of fine-grained soils (and thus ML or CL) is based on plasticity (i.e., on the position relative to the A-line), while the classification of coarse-grained soils (and thus SM or SC) is based mainly on fine contents and the position of these values relative to the A-line. It is therefore expected that these classes would sometimes be confused if samples had fine contents and plasticity values close to the A-line boundary in Figure 2.

On the contrary, in the case of partial feature groups that lack plasticity information (G_5 and G_6), misclassifications between the M and C groups of fine-grained USCS classes increase significantly, in accordance with results showing that Atterberg limits play an important role in soil classification.

Table 6. Top accuracy results from MATLAB learner for Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) models (20% test split, G_5–G_{12})

Group	SVM (Top Model)			KNN (Top Model)		
	Classifier	Validation Accuracy (%)	Test Accuracy (%)	Classifier	Validation Accuracy (%)	Test Accuracy (%)
G_5	Medium Gaussian	72	78.1	Fine KNN	75.4	77.1
G_6	Cubic SVM	73.3	70.8	Weighted KNN	79.6	79.2
G_7	Kernel SVM	76.4	80.2	Weighted KNN	83.2	88.5
G_8	Cubic SVM	91.1	91.7	Fine KNN	89.8	94.8
G_9	Cubic SVM	91.1	93.8	Weighted KNN	90.8	92.7
G_{10}	Cubic SVM	72.3	80.2	Weighted KNN	74.1	83.3
G_{11}	Kernel SVM	76.4	77.1	Fine KNN	78.8	80.2
G_{12}	Cubic SVM	77.2	83.3	Weighted KNN	84.3	88.5

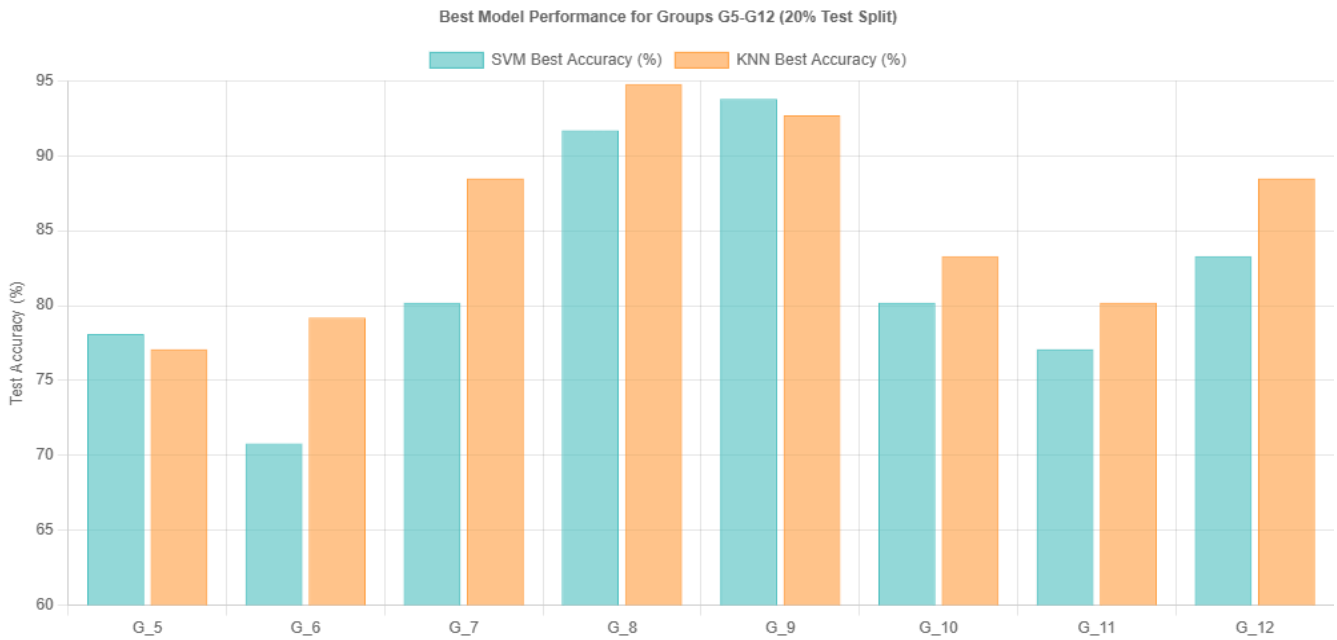


Figure 6. Best test accuracy for Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) models across groups G_5 to G_{12} (20% test split)

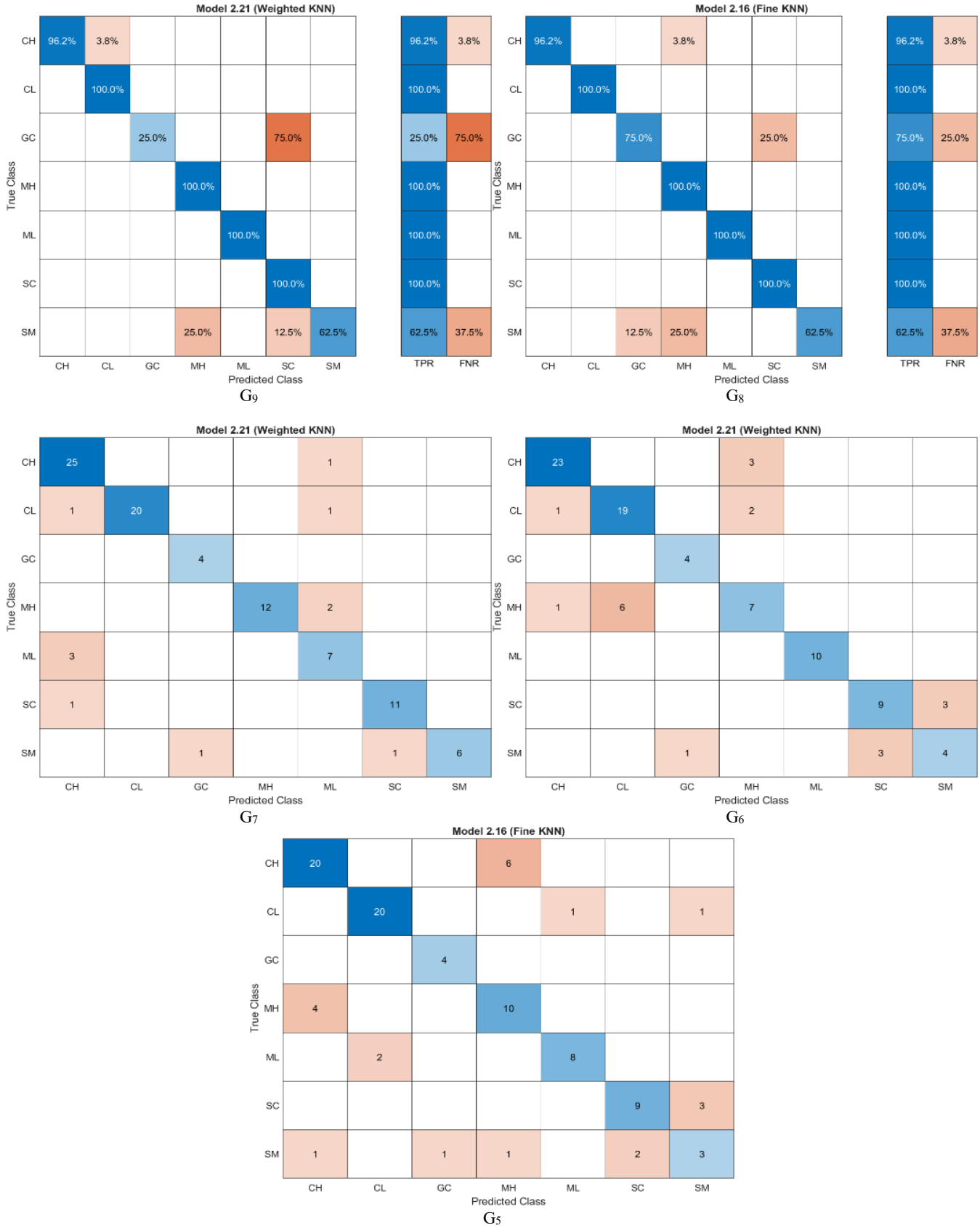


Figure 7. Test data confusion matrices for different feature groups

4. DISCUSSION

This research serves as an instruction manual for the geotechnical community on implementing ML-assisted soil classification on a day-to-day basis. This project didn't seek to present a novel classifier but rather to provide a focused response to a key engineering inquiry: with the range of parameters accessible at various phases of a ground exploration, which classifier is the best to use and which inputs will result in the most effective USCS prediction?

The findings have immediate relevance to field implementation; for instance, if a client requires a USCS description during the desk review or preliminary ground investigation stage, where only sieve and Atterberg information is known, the models trained on groups G_8 or G_9 (94.8% accuracy) could be used for immediate USCS classification. As information from more advanced phases of work is made available, models can be updated accordingly; for example, for groups G_2 or G_4 with Cubic SVM, for a USCS classification accuracy of 97.9%. This methodology indicates that ML models can be incorporated within existing geotechnical practice, which can be performed without requiring extra resources to the laboratory testing regime and using freely accessible software (MATLAB).

In respect to computation cost, the SVM and KNN models presented in this study were trained and validated in MATLAB R2024a with a desktop PC (Intel Core i7, 16 GB RAM). The complete process of training all 11 model variants for all 12 variable groups (with 5-fold cross-validation) took roughly 3–5 min in total. The shortest time required for training an individual model was less than a second (Fine KNN, k = 1), with the longest model training times around 30 seconds (Gaussian SVM, with RBF kernel with the biggest set of variables). Classification of all 96 samples in the testing set for all models was completed in less than 0.1 s. These figures suggest that the proposed approach is a feasible, practical, and useful solution even without access to advanced hardware, such as graphics cards or clusters. It should be noted that computation times are not reported to sub-second precision as it was not a focus of this work; however, it is proposed in further work to report more formally on computational complexities, including, for example, comparisons between the SVM/KNN models and ensemble-based models.

4.1 Comparison of Support Vector Machine and K-Nearest Neighbors performance

A key finding of this study is that Cubic SVM's performance is another interesting point of note In the 10% test split, Cubic SVM achieved a maximum accuracy of 97.9%, while its maximum accuracy in the 20% test split was 96.9%. This implies that the complex, non-linear decision boundaries learned by the cubic kernel fit well the multi-class nature of soil classification. SVM's advantage in determining an optimal separating hyperplane in a high-dimensional feature space has been previously noted [16]. A study on land use/cover classification [11] reported that SVM generated the highest overall accuracy and the lowest sensitivity to training sample size compared to RF and KNN. We also observed that the SVM variants, especially the kernelized ones, offered consistently robust and accurate predictions.

The KNN models also did well, with Weighted KNN and Fine KNN frequently matching the best SVM model's performance. Fine KNN with k = 1 resulted in a maximum accuracy of 97.9% in the 10% test split (i.e., the same as the best SVM model), which shows that, for certain feature sets, the local structure of the data is so unique that it is possible to classify using just a single neighbor. Weighted KNN was also a very strong and stable model, especially in the more restricted variable groups (G_7, G_9, G_{12}), where it frequently outperformed the best SVM model. The study [23] reports the effectiveness of instance-based learners, such as KNN, for geotechnical applications, which we also observe. However, the relatively poor performance of Medium, Cubic and, in particular, Cosine KNN shows that choosing the correct type of KNN and its hyperparameters is important, as also observed by study [14].

4.2 Influence of input variables

The results of variable groups from G5 to G12 are of particular interest since they are directly relevant to the question of which soil variables are the most important for classification. The most striking observation is the significant drop in the accuracy of groups G5 (without P) and G6 (without LL). For group G6, the Cubic SVM model achieved a test accuracy of 70.8%, as reported in Table 6. This result, together with the lower accuracies observed for groups G5 and G6, suggests that retaining complementary particle-size distribution and plasticity-related variables may contribute to improved discrimination among the seven USCS classes represented in the present dataset [1]. In the absence of any plasticity measure (LL or PI), it becomes impossible to distinguish between silts (M) and clays (C) and between low-plasticity (L) and high-plasticity (H) soils.

Equally interesting is the excellent accuracy achieved by the models trained on groups G_8 and G_9, with the highest being 94.8%, which is only marginally lower than the models trained on the full-featured groups. This shows that, with the right selection of variables, it is possible to match the performance of the full variable set. For instance, group G_8 uses the Passing the No. 200 Sieve along with LL and PI. The fact that the models trained on this group achieve such high accuracy suggests that the models are able to learn the classification rules based on these three fundamental variables. This is of great practical importance, because it shows that, as long as the basic data on PSD and Atterberg limits are available, the models will perform well.

A comparison of the use of raw variables [i.e., Passing No. 4 and No. 200 (G_1, G_5, G_8, and G_{11})] versus the variables derived from them (Gravel, Sand, Fine; G_3, G_6, G_9, and G_{12}) did not indicate any advantage of one over the other. The best-performing models in both categories (e.g., G_1 vs. G_3) performed equally well. This indicates that the ML models are able to learn the linear transformations required to translate between these two variable sets. This provides great practical flexibility, as models can be developed to work with either of the two data types. Similarly, a comparison between the use of PI versus PL (when used in conjunction with LL) also showed a minor impact on the performance of the top models. Given that PI is simply calculated from LL and PL, it is not surprising that both data types worked equally well.

4.3 Effect of data splitting

We utilized two separate test set sizes (10% and 20%) for diagnostic purposes rather than strictly for comparison. The fact that the two test set sizes achieved similar accuracies (97.9% and 96.9%, respectively) for the top classifiers and maintained similar rankings of the candidate models is therefore suggestive of, but does not by itself prove, robustness of the trained models to the specific train/test partition used. Furthermore, the 5-fold cross-validation accuracy and the test accuracy differed by only about 2-3% for the most accurate models, which is consistent with the models learning generalizable features rather than memorizing the training data. These observations are, however, based on a single random seed; a more rigorous evaluation using repeated random subsampling or k-fold cross-validation across multiple seeds is required before strong generalization claims can be

made, and this is identified below as an important direction for future work.

However, if samples are taken from significantly different soil environments, we would expect that a different model will need to be trained. We have varied the combinations of input variables as well as the split ratio of the study, but several aspects of robustness were not examined in the present study. In particular, the robustness of the trained classifiers to Gaussian noise added to the inputs and to randomly missing input values was not investigated, although both are common problems in real geotechnical practice (e.g., measurement noise, instrument miscalibration, and disturbed samples). A systematic robustness analysis that includes noise injection and missing-value simulation is therefore an important limitation of the current work and is planned as part of a follow-up study.

Finally, we performed 5-fold cross-validation during the training process to assess the stability of the accuracy; a more rigorous validation would involve a repeated random subsampling scheme combined with multiple random seeds.

4.4 Comparison with existing literature and limitations

The performance of the models in the current study is generally on a par with those of other studies that have explored the use of ML for geotechnical engineering applications. For instance, the study [15] attained accuracy values as high as 93% for soil classification models that rely on CPT results. In contrast, the highest accuracy in the current study, which was based on laboratory test data, was even higher. Similarly, the performance of the two best-performing models (Cubic SVM and Weighted KNN) in the current study was in line with the results obtained by the study [20], which concluded that ensemble and advanced kernel-based models (e.g., extreme gradient boosting (XGBoost) and SVM) can generally perform better than the other models when complex soil properties are predicted.

In terms of limitations, this paper needs to consider the following issues.

One, the dataset we gathered was obtained from published information and may not be consistent with field conditions due to other uncertainties such as test error, non-standardized testing, and spatial variations, which could all produce scatter.

Second, we focused on SVM and KNN algorithms in order to conduct a more controlled, detailed comparison between different variants of the same family. There are many other families which have also proved effective for geotechnical classification, including, for example, RF, Gradient Boosting (i.e., XGBoost), and ANN [12, 24], which could be considered in subsequent studies. For instance, ensemble (i.e., RF, XGBoost) and kernel-SVM-based classifiers usually outperform single learners for large datasets with many features in a published study [24]; however, it has been demonstrated that, for the feature space size and data size, SVM and KNN are still competitive benchmarks [11].

Third, the dataset size we employed only included one case study, which needs to be validated by others before any conclusions could be drawn to justify its generalization, especially in different regions or countries.

5. CONCLUSIONS

This paper evaluates SVM and KNN classifiers for automatic USCS soil classification, using a data set comprising 478 samples defined by standard soil testing parameters. Twelve sets of independent variable groups and two test splits were examined. The main findings with quantitative results are presented here:

1. Cubic SVM (Group G_4 and 10% split) and Fine KNN (Group G_2 and 10% split) were the top-performing classifiers, with the highest classification accuracies of 97.9%. By contrast, Cubic SVM achieved classification accuracies of 96.9% in Group G_2 and G_4 with the 20% split, demonstrating the performance stability of these models, irrespective of their partition in training and test sets.

2. Input-variable combinations that included Atterberg limits parameters—particularly the Liquid Limit (LL) and Plastic Limit (PL)—alongside the percentage passing the No. 200 sieve were generally associated with higher classification accuracy. For group G6, the Cubic SVM model achieved a test accuracy of 70.8%, as reported in Table 6. In contrast, group G8, consisting of the percentage passing the No. 200 sieve, LL, and Plasticity Index (PI), achieved a maximum test accuracy of 94.8% using Fine KNN. These findings suggest that a carefully selected limited set of input variables may retain substantial discriminative information and may help guide the development of soil classification models in future studies where laboratory data or testing resources are limited.

3. Rankings of classifiers were almost the same for both 10% (48 samples) and 20% (96 samples) test sets, which is consistent with stable relative performance of the candidate models under the two split ratios considered. In addition, 5-fold cross-validation accuracy was close to the holdout test accuracy, differing by only a few percentage points for the top classifiers.

4. It should also be noted that this data set may not represent every geotechnical site where a USCS classification might occur. For instance, the current models may not be able to classify USCS samples from tropical regions (e.g., from laterite soils), organic sites (e.g., peat), or any other cemented soils that may not fit within the ranges of the training data used in this paper. Furthermore, this study only investigated two classifiers: SVM and KNN. Comparisons to other popular machine-learning techniques such as RF and gradient boosting are suggested for a wider comparison in future work.

This study showed promising results in that SVM and KNN models trained using a set of standard laboratory measurements can serve as useful classification tools in routine soil investigations. Further research will seek to validate and improve these models by testing on data sets with more regional variations and different geologic settings and by incorporating more sophisticated machine learning methods, e.g., neural networks and ensemble classifiers, and investigating the impact of parameter uncertainty and sample size in the training data on the model performance.

REFERENCES

[1] Mitchell, J.K., Soga, K., O'Sullivan, C. (2025). Fundamentals of Soil Behavior. John Wiley & Sons.

[2] Huang, F.M., Xiong, H.W., Chen, S.X., et al. (2023). Slope stability prediction based on a long short-term memory neural network: Comparisons with convolutional neural networks, support vector machines and random forest models. International Journal of Coal Science & Technology, 10: 18.

https://doi.org/10.1007/s40789-023-00579-4
[3] ASTM Committee D-18 on Soil and Rock. (2017). Standard practice for classification of soils for engineering purposes (unified soil classification system). ASTM International. https://doi.org/10.1520/D2487-17
[4] O'Kelly, B.C. (2021). Review of recent developments and understanding of Atterberg limits determinations. Geotechnics, 1(1): 59-75. https://doi.org/10.3390/geotechnics1010004
[5] Pendleton, M. (2021). The unified soil classification system: What you should know before a career as a professional geologist. Geological Society of America Abstracts with Programs, 53(4): 363171. https://doi.org/10.1130/abs/2021CD-363171
[6] Yaghoubi, E., Yaghoubi, E., Khamees, A., Vakili, A.H. (2024). A systematic review and meta-analysis of artificial neural network, machine learning, deep learning, and ensemble learning approaches in field of geotechnical engineering. Neural Computing and Applications, 36: 12655-12699. https://doi.org/10.1007/s00521-024-09893-7
[7] Harle, S.M., Wankhade, R.L. (2025). Machine learning techniques for predictive modelling in geotechnical engineering: A succinct review. Discover Civil Engineering, 2: 86. https://doi.org/10.1007/s44290-025-00224-w
[8] Ahmad, M., Al Zubi, M., Almujibah, H., et al. (2025). Improved prediction of soil shear strength using machine learning algorithms: Interpretability analysis using SHapley Additive exPlanations. Frontiers in Earth Science, 13: 1542291. https://doi.org/10.3389/feart.2025.1542291
[9] Saad, A.H., Nahazanan, H., Yusuf, B., et al. (2023). A systematic review of machine learning techniques and applications in soil improvement using green materials. Sustainability, 15(12): 9738. https://doi.org/10.3390/su15129738
[10] Obasi, S.N.N., Pemberton, J., Awe, O.O. (2025). A comparative study of soil classification machine learning models for construction management. International Journal of Construction Management, 25(5): 584-593. https://doi.org/10.1080/15623599.2024.2341500
[11] Noi, P.T., Kappas, M. (2018). Comparison of random forest, k-nearest neighbor, and support vector machine classifiers for land use/cover classification using Sentinel-2 imagery. Sensors, 18(1): 18. https://doi.org/10.3390/s18010018
[12] Adugna, T., Xu, W.B., Fan, J.L. (2022). Comparison of random forest and support vector machine classifiers for regional land cover mapping using coarse resolution FY-3C images. Remote Sensing, 14(3): 574. https://doi.org/10.3390/rs14030574
[13] Cover, T., Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1): 21-27. https://doi.org/10.1109/TIT.1967.1053964
[14] Seman, P., Shoop, S., McGrath, S.P., Rolling, R.S. (2006). Soil strength prediction with k-nearest neighbor method. In Proceedings of the 59th Canadian Geotechnical Conference, pp. 212-218. https://members.cgs.ca/documents/conference2006/Seatosky/S1/0212-218.pdf.
[15] Chala, A.T., Ray, R. (2023). Assessing the performance of machine learning algorithms for soil classification using cone penetration test data. Applied Sciences, 13(9): 5758. https://doi.org/10.3390/app13095758
[16] Ahmad, M., Tang, X.W., Qiu, J.N., Ahmad, F., Gu, W.J. (2021). Application of machine learning algorithms for the evaluation of seismic soil liquefaction potential. Frontiers of Structural and Civil Engineering, 15: 490-505. https://doi.org/10.1007/s11709-020-0669-5
[17] Liu, C.Y., Ku, C.Y., Wu, T.Y., Ku, Y.C. (2024). An advanced soil classification method employing the random forest technique in machine learning. Applied Sciences, 14(16): 7202. https://doi.org/10.3390/app14167202
[18] Shakir, B.Q., Shareef, M.A., Al Nuaimy, Q.A.M. (2025). GIS-based spatial interpolation approaches for soil parameter assessment: A case study from the Taza Khurmatu District, Kirkuk Governorate, Northern Iraq. Iraqi Geological Journal, 58(2C): 117-129. https://doi.org/10.46717/igj.2025.58.2C.7
[19] Kamath, N.S. (2021). ConcreteSet. Kaggle. https://www.kaggle.com/datasets/nanditasureshkamath/concreteset, accessed on Dec. 11, 2025.
[20] Nguyen, Q.H., Ly, H.B., Ho, L.S., et al. (2021). Influence of data splitting on performance of machine learning models in prediction of shear strength of soil. Mathematical Problems in Engineering, 2021(1): 4832864 .https://doi.org/10.1155/2021/4832864
[21] Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20: 273-297. https://doi.org/10.1007/BF00994018
[22] GeeksforGeeks. (2025). How to find the optimal value of K in KNN? GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/how-to-find-the-optimal-value-of-k-in-knn/.
[23] Shah, J., Alam, M., Javed, F., Umar, M., Ahmad, F. (2025). Comparative performance evaluation of machine learning models for predicting the ultimate bearing capacity of shallow foundations on granular soils. Scientific Reports, 15: 36525. https://doi.org/10.1038/s41598-025-13926-z
[24] Almuaythir, S., Zaini, M.S.I., Lodhi, R.H. (2025). Predicting soil compaction parameters in expansive soils using advanced machine learning models: A comparative study. Scientific Reports, 15: 24018. https://doi.org/10.1038/s41598-025-09279-2