
PREDICTION MODEL FOR SOIL WATER RETENTION BEHAVIOUR OF UNSATURATED SOILS WITH DIFFERENT VOLUMETRIC WATER CONTENT

**Abdullahi Karaye¹, Gambo Haruna Yunusa^{1,*}, Abubakar Jibrin Shuaib¹, Hafiz
Ibrahim², Ibrahim Khalil Umar³**

¹Bayero University Kano

²Ahmadu Bello University Kano

³Federal University of Technology, Babura, Nigeria

Email: askaraye@gmail.com, ghyunusa@gmail.com*, sadiqjibrin11@gmail.com, sirhafiz01@gmail.com,
ikumar.civil@futb.edu.ng

***Corresponding author:** Gambo Haruna Yunusa

DOI: <https://doi.org/10.67203/sp.ICESID.2026>

ABSTRACT

Unsaturated soils exhibit complex hydro-mechanical behaviour that is strongly influenced by their water content and soil properties. Matric suction is an important parameter governing the behaviour of unsaturated soils, particularly fine-grained and expansive soils, and its determination can require considerable experimental effort. This study therefore developed and evaluated machine-learning models for predicting the matric suction of unsaturated soils using readily available soil and hydraulic properties. A dataset comprising 2,550 soil records was used, with saturated volumetric water content (θ), air-entry value, bulk density (ρ_b), porosity (n), natural moisture content (w), plasticity index (PI), dry density (ρ_d), D_{60} , and coefficient of permeability (k) adopted as predictor variables. Seven machine-learning models, namely Extreme Gradient Boosting (XGBoost), Linear Regression, Ridge Regression, Lasso Regression, Random Forest, Support Vector Regression (SVR), and Decision Tree, were evaluated using training R^2 , test RMSE, and cross-validation mean R^2 . XGBoost demonstrated the best overall predictive performance, achieving a training R^2 of 0.9985, test RMSE of 144.3092 kPa, and cross-validation mean R^2 of 0.8712, followed by Random Forest with a training R^2 of 0.9829, test RMSE of 142.8048 kPa, and cross-validation mean R^2 of 0.8611. Although Random Forest recorded a slightly lower test RMSE, XGBoost exhibited the highest training and cross-validation R^2 values. Overall, the results demonstrate the potential of machine-learning techniques, particularly XGBoost, for predicting matric suction of unsaturated soils from selected soil and hydraulic properties.

Keywords: Matric suction; Unsaturated Soils; Machine Learning Models; XGBoost

1. INTRODUCTION

Most of the soils around us are Unsaturated, with various particle composition of gravel, sand, silt or clay. These soils are expansive which often subjects them to complex hydro-mechanical circles as they contain a lot of moisture. These characteristics possessed by these types of soil, lead to disorders on structures built or constructed on or within them. Clay soils possess a complex behaviour as a result of the strong multi-physical processes taking place. Initially unsaturated, the compacted clay experiences hydration from the saturated host rock. During this process, it tends to expand and develops swelling stresses (Dieudonne *et al*, 2014). Unsaturated soils were found to be four phase soils, namely; the air phase, water phase, soil skeleton, and contractile skin phase. The air and water phase flows while the soil skeleton, and contractile skin phase come to equilibrium under imposed loads (Pereira and Fredlund, 1997). The most significant work during that period was Bishop's effective stress relationship, whereby changes in effective stress attributed to the unsaturated nature of a soil are cast as the product of matric suction in Bishop's effective stress equation: $[\sigma' = (\sigma - U_a) + \chi(U_a - U_w)]$. Effective stress cast in the form of this equation has several difficulties. Looking back, some of the difficulties may not have risen only because of flaws with Bishop's conception, but rather misinterpretation of test results by other investigators (Bishop, 1963; Kanzari *et al*. 2012). A theoretical difficulty in the equation is that it predicts zero effective

stress for any soil in a dry state under no external stress. There is a transitory pore-size range between micro- and macro-structure which is not affected by the magnitude of the compaction pressure (Miller *et al.* 2002). As such, the mechanical loading influences only the macro-structure without affecting the micro-structure (Gao and Sun, 2017). The relationship between pore water suction and water content as presented in a soil water characteristic curve is one of the fundamental relationships used to describe unsaturated behavior of a soil (Nowamooz and Masrouri, 2010). The water retention behavior of unsaturated soils is presented using the soil water retention curve (SWRC). Soil water characteristics curve is a conceptual and interpretative tool by which behavior of unsaturated soils can be understood (Fredlund *et.al*, 2001). This fundamental curve links the variations of the liquid phase potential energy, usually characterized by suction, to the variations of the saturation state, usually characterized by the water content or the degree of saturation (Simon *et.al*, 2011). Suction is inversely proportional to the water content in a soil. Suctions generally increase as the soil de-saturates. Increasing suction generally result in high resistance to flow and increases in effective stress. Soil Water Retention Curve of soils is the tool used to present the hydro-mechanical behavior of soils to highlight both the effect of suction on the change in water and total volumes and the effect of deformation with respect to the water retention capability (Salager *et.al*, 2010). The Soil Water Retention Curve of soils depends on several factors, such as soil type, soil structure, temperature, salt content and stress states etc. These factors can be classified into two groups: internal and external factors. Internal factors are those related to the soil type and soil structure, such as mineralogical composition, pore size distribution and bonding (that is, chemo-physical properties of the soil phases). Typical external factors are stress states and temperature (Khalili and Selvadurai, 2003). For the same type of soil, the effect of stress states on the SWRC can be attributed to the effect of the density change (Gao and Sun, 2017). It was reported that the SWRC expressed by the relation between the suction and degree of saturation depends mainly upon the current density and not directly upon the stress state (Romero and Gens 1999, Gao and Sun, 2017). Although the influence of structure on the behavior of compacted clays has been recognized, only a few water retention models have taken it explicitly into account. Indeed classical approaches for modeling the water retention behavior are based on parameters to be fitted using experimental data (Levasseur, 2012). An understanding of the water retention features is an essential aspect in the field of unsaturated soil behavior modeling. Concomitantly, the trend in modern unsaturated soil mechanics is to quantify, with increasing accuracy, the soil water retention properties of soil in terms of; a fitting equation, the influence of soil compaction conditions, hysteresis modeling and the effect of suction on density (Nuth and Laloui, 2008). Several elastoplastic constitutive models for unsaturated soils have been proposed. The stress-state variables employed in early models are the net stress (the difference between the total stress and pore-air pressure) and the suction (the difference between the pore-air pressure and pore-water pressure), while the strain state variables are the soil skeleton strains. In these models, the influence of un-saturation is expressed only in terms of the suction. The degree of saturation is computed from the suction separately using the so-called soil water retention curve (SWRC), and thus hydraulic hysteresis is not directly incorporated into the constitutive model (Sun *et al*, 2007). The prediction quality of the degree of saturation will thus depend on the accuracy of the void ratio evolving with suction. The proposed water retention model consequently needs to be coupled with a stress-strain model for unsaturated soils. An appropriate constitutive mechanical framework shall be elastoplastic and account for the suction effects on the stress-strain response (D'Onza *et al.* 2010; Nuth *et al.* 2010; Simon *et al*, 2013). Language based modeling approaches foster flexibility and performance by adopting the same host language business and code coordination. Jupyter notebooks with their ability to describe both imperative and declarative coding in a unique format, allowing for selection of the best of two approaches. They still maintain a clear separation between application and coordination layers but still providing a unified interface to both aspects. (Colonnelli *et al*, 2021). Jupyter notebook possess the potential of expressing complex distributed workflows and workflow management system targeting cloud high performance computing (Cloud-HPC) infrastructures and extension stack to implement a proposed methodology (Cantalupo *et.al*, 2022).

2. METHODOLOGY

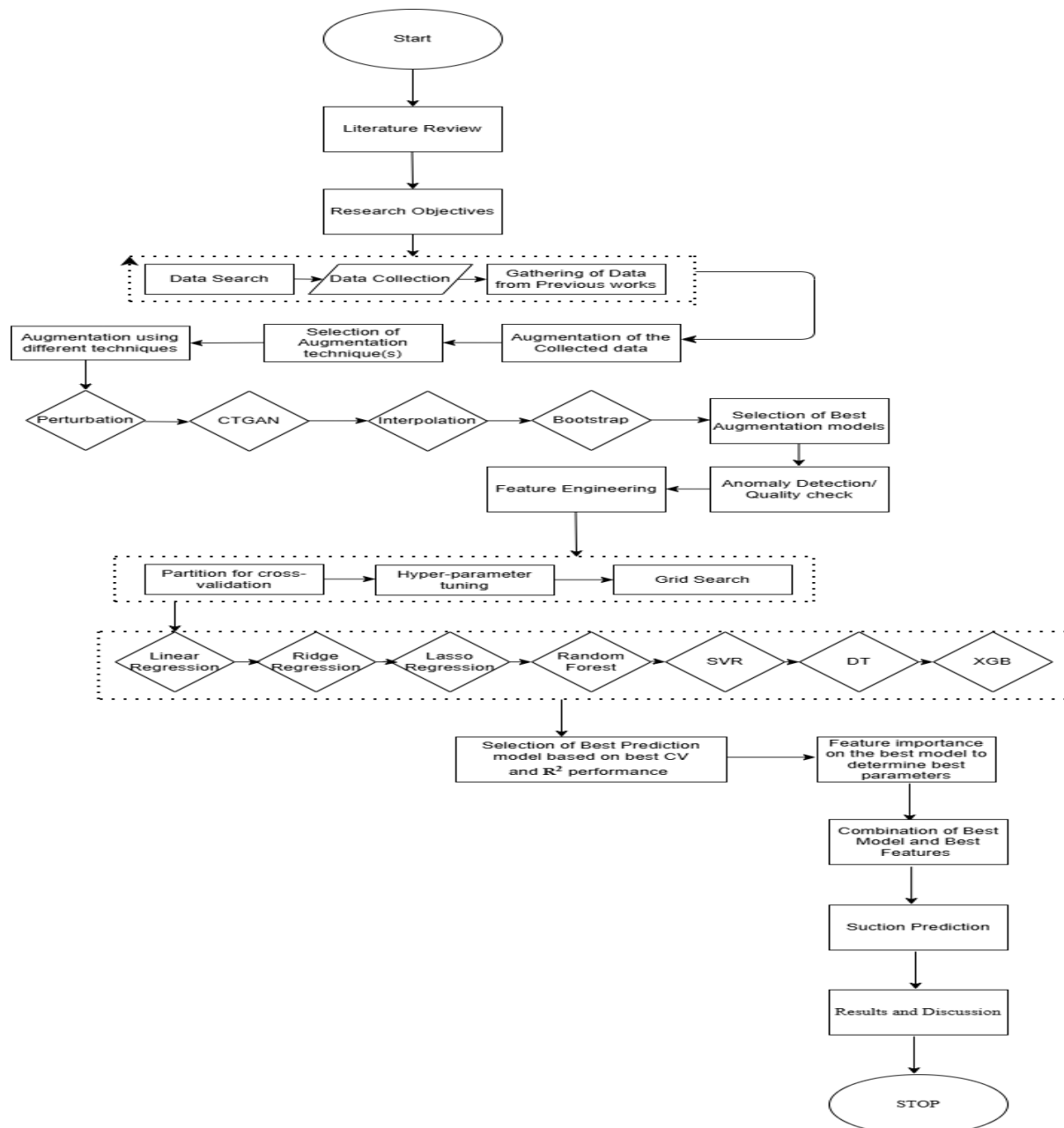


Figure 2.1: Methodology Flowchart

2.1 ORGANIZATION OF DATA

The soil parameters to be used for Matric suction prediction were first organized. Database were collected through previous work of Eberemu, 2008 The data obtained was just 50 dataset which quantity was unfit for machine learning research of this kind, as such the data was sequentially

augmented via the Google Colab version of *Jupyter Notebook environment* of Python. The augmentation process was achieved by;

- Identifying the data
- Applying Augmentation model library (Bootstrap, Perturbation, CTGAN, interpolation)
- Generating synthetic data
- Combining original and generated data

After identifying the original data, the augmentation libraries were employed in order to generate more data for the suction prediction. The following augmentation methods/ libraries were employed; CTGAN (Conditional tabular generative adversarial network), Perturbation, Bootstrap and Interpolation.

2.2 Anomaly Detection test (AD): After the augmentation process, the resulting data (i.e the augmented) was then validated by comparing it with the original data for feature importance. It is conducted in order to evaluate how well the detector generalizes to synthetic variations. From the AD conducted, it was observed that, the augmented data by Interpolation gives the best synthetic data followed by the Bootstrap technique.

2.3 Feature Engineering: Feature engineering is the process of generating, selecting, or transforming raw data into features that improve the performance of machine learning models. It involves domain knowledge, creativity, and data manipulation to craft features that capture relevant patterns and relationships in the data (Trost et.al, 2025). It improves Model Performance, as the engineered features can make simple models and outperform complex ones with poor features (Huang et.al,2023). Feature Engineering captures domain knowledge, in that it incorporates insight about the problem in the data. It also addresses missing values and outliers for model compatibility (Bjorneld et.al,2023). In this research work, Feature Engineering was the first stage of the suction prediction process. The best augmented data adopted (i.e Data from Interpolation), was explored to determine the statistical properties and feature correlations, which was sorted by absolute correlation. After exploring the data;

1. Interaction features were generated
2. Polynomial features were also generated for top correlated values
3. Finally, the new Engineered features were produced

Engineered features were then determined after the model studied the relevant patterns and relationships in the data. The engineered features generated were; Dry density to Natural moisture content ratio ($P_d:w$), Bulk density to Dry density ratio ($P_b:P_d$), Natural moisture content to Volumetric water ($w:\Theta$), Dry density squared (P_d^2), Natural moisture content squared (w^2), moisture level number, density level number, and Log of hydraulic conductivity (Log K). These new features were then combined with the existing features of the data. These combined features were used to train the models for prediction of the matric suction. The models are;

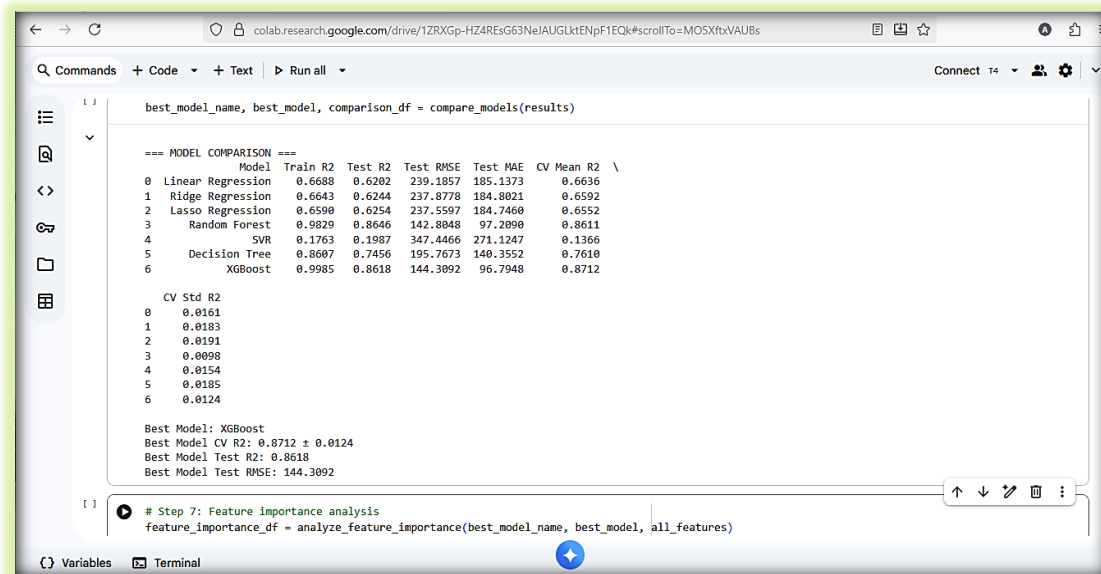
- i. Linear Regression
- ii. Ridge Regression
- iii. Lasso Regression
- iv. Random Forest (RF)
- v. Support Vector Regression (SVR)
- vi. Decision Tree
- vii. Extreme gradient boosting (XGBoost)

The feature Engineering continued by setting hyper parameter grids;

4. The top features and the engineered features were prepared
5. The data were then Split into training and testing, and the random state was also set.
6. The features were then scaled

2.4 Hyperparameter tuning: Hyperparameter tuning is the process of systematically searching for the optimal set of hyperparameters that maximize the model's performance on a given metric, such as accuracy, F1-score, or loss minimization (Belcic and Stryker, 2025). This involves defining a search space (the range of possible values for each hyperparameter), evaluating model performance for different combinations, and selecting the best configuration. Tuning is typically performed using cross-validation to ensure the model generalizes well to unseen data (Elgeldawi et.al, 2021). These hyperparameters control the model's behaviour and impacts its performance significantly (Sci-kit learn.org). Hyper parameter employed in this work was carried out in the following sequence;

1. Models to be used for the prediction of matric suction were first defined, where; Linear regression, Ridge regression, Lasso regression, decision tree, support vector regression (SVR), and XGboost were used
2. Parameter grids were then specified, which were; n-estimators, max depth, learning rate, and subsample (for RF), rbf, C value and gamma scale (for SVR), alpha value (for ridge and lasso regressions), and random state (for decision tree and XGboost models).
3. Scoring metrics were then chosen and set. The chosen scoring metrics were; R^2 , RMSE, and MAE.
4. Cross validation for the models training was also set, where a CV value of 5 folds was set.
5. Hyperparameter grids were then set for the models comparison and selection, where the best model was selected from the array of models.
6. Feature importance analysis was then carried out to determine the best parameters for the best model



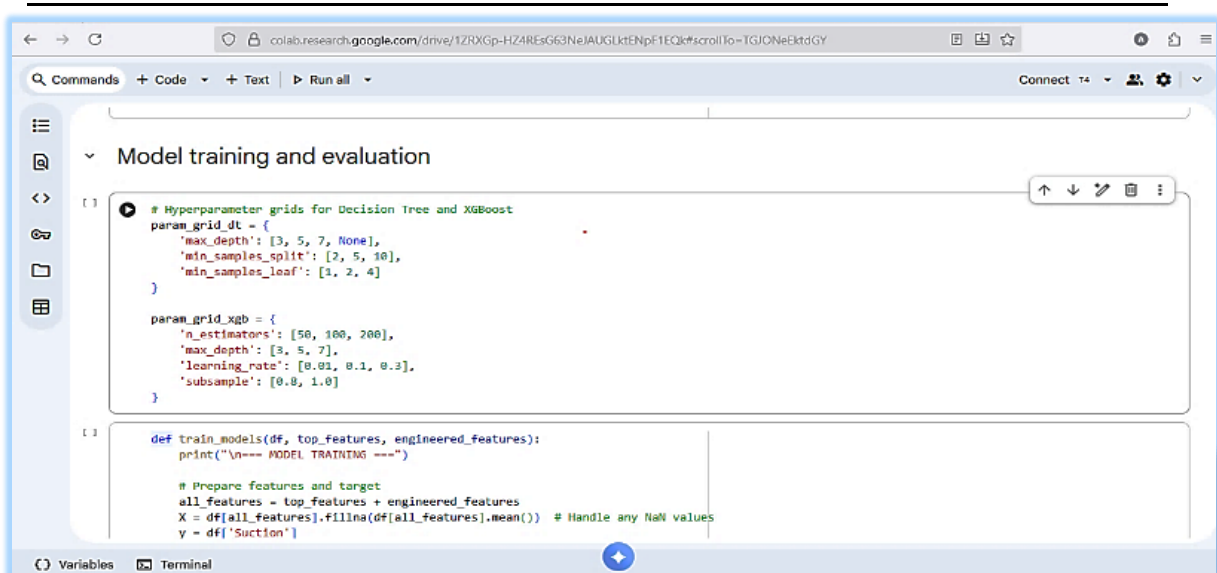
```
best_model_name, best_model, comparison_df = compare_models(results)

=== MODEL COMPARISON ===
      Model  Train R2  Test R2  Test RMSE  Test MAE  CV Mean R2 \
0  Linear Regression    0.6688    0.6202    239.1857    185.1373    0.6636
1  Ridge Regression     0.6643    0.6244    237.8778    184.8021    0.6592
2  Lasso Regression     0.6590    0.6254    237.5597    184.7460    0.6552
3  Random Forest        0.9829    0.8646    142.8048     97.2090    0.8611
4  SVR                  0.1763    0.1987    347.4466    271.1247    0.1366
5  Decision Tree        0.8607    0.7456    195.7673    140.3552    0.7610
6  XGBoost              0.9985    0.8618    144.3092     96.7948    0.8712

CV Std R2
0    0.0161
1    0.0183
2    0.0191
3    0.0098
4    0.0154
5    0.0185
6    0.0124

Best Model: XGBoost
Best Model CV R2: 0.8712 ± 0.0124
Best Model Test R2: 0.8618
Best Model Test RMSE: 144.3092

# Step 7: Feature importance analysis
feature_importance_df = analyze_feature_importance(best_model_name, best_model, all_features)
```



```
# Hyperparameter grids for Decision Tree and XGBoost
param_grid_dt = {
    'max_depth': [3, 5, 7, None],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

param_grid_xgb = {
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 5, 7],
    'learning_rate': [0.01, 0.1, 0.3],
    'subsample': [0.8, 1.0]
}

def train_models(df, top_features, engineered_features):
    print("\n--- MODEL TRAINING ---")

    # Prepare features and target
    all_features = top_features + engineered_features
    X = df[all_features].fillna(df[all_features].mean()) # Handle any NaN values
    y = df['Suction']
```

Figure 2.1: Setting of Hyperparameter grids

Figure 2.2: Grids Search

After executing these steps, the training process was executed and the fine-tuned parameters were determined. In this, the common parameters that makes the best of the trained models during the Hyper-parameter tuning were selected and are used together with the best determined model for the

prediction. The best model determined was the Extreme gradient boost (XGboost) model, and the best/ fine-tuned parameters are; gravimetric water content Θ , Air entry value(Υ_a), hydraulic gradient (k), bulk density (p_b), 'D60', and natural moisture content (w). These parameters were used together with the XGboost model to achieve the aim of this research.

3. RESULTS AND DISCUSSION

This research was conducted in order to predict the maximum matric suction up to which an unsaturated soil can attain using some of the soil's properties by the best model which is to be determined from the grid search to be conducted in the work. This was achieved by using the following soil properties; Saturated volumetric water content (Θ), air entry value, Bulk density (P_b), porosity (n), natural moisture content (w), Plasticity index ($P.I$), Dry density (P_d)and D_{60} particle size parameter, and coefficient of permeability(k). These soil parameters were used as inputs, and the matric suction was the Class (target). The models used are described in Appendix B. The data presented in Table 4.1 are the first five rows of the initial/rough data obtained from previous works of Eberemu, 2008. These data are as displayed during the EDA, prior to data augmentation.

Table 3.1: Some of Data obtained from the previous works

S/N	p_d	W	p_b	P.I	Θ	K	N	D60	Υ_a	Suction
1	1.58	16.5	27.65	24.0	0.313	3.41e-9	0.395	0.13	26	100
2	1.67	14	25.05	24.0	0.225	1.32e-9	0.360	0.13	27	1500
3	1.74	13.7	25.58	24.0	0.252	8.09e-10	0.333	0.13	70	500
4	1.84	10.9	21.90	24.0	0.233	1.10e-10	0.295	0.13	30	1000
5	1.58	17	28.44	20.5	0.329	3.57e-9	0.375	0.07	27	100

3.1 DATA AUGMENTATION

The soil data obtained from the initial works of Eberemu,2008 were only 50 dataset. These seem unfit for machine learning, and data of this kind are very scarce due to the limited availability of the apparatus and deep technical experience to conduct such laboratory test, this is one of the challenges that lead to the idea of coming up with this research.

As such to compliment for machine learning, the data was carefully increased to a quantity that will fit the machine learning process by generating **realistic synthetic data** via the Google Colab version of *Jupyter Notebook environment* of Python. The data was first explored by the EDA function (exploratory data analysis), where the data was fully identified; The data was found to possess the following characteristics:

Table 3.2: EDA of Obtained data.

Statistics	p _d	W	p _b	P.I	Θ	K	N	D ₆₀	□a	Suction
Mean	1.671	17.2111	30.221	16.940	0.294	7.85e-10	0.3276	0.0814	43.42	594.60
STD	0.092	2.7374	3.3610	4.7238	0.035	2.83e-9	0.0387	0.0294	22.13	494.74
Min	1.520	10.900	21.896	9.3000	0.225	1.77e-11	0.242	0.058	26.00	30.00
25%	1.620	15.325	28.143	10.550	0.269	8.87e-11	0.2968	0.0650	30.00	100.00
50%	1.6600	17.450	30.353	17.400	0.295	1.59e-10	0.3305	0.0675	34.00	500.00
75%	1.740	19.000	32.591	20.500	0.322	3.96e-10	0.3590	0.1150	51.50	1000.00
Max	1.860	22.500	35.796	24.000	0.362	2.02e-8	0.3950	0.1300	130.00	1500.00

The techniques/ models used for data augmentation are;

3.11 CTGAN; This is the conditional tabular generative adversarial network, and it was adopted in this research to augment our data as it possesses the characteristics of generating synthetic data in tabular form (Adiputra and Wanchai, 2024). This technique generates 2550 dataset inclusive of our original obtained 50 dataset. Table 4.3 shows the last 10 rows of the 2550 dataset as augmented via this technique; the original data was in the first 50 rows of the gross data. As such, these in table 4.3 are among the newly generated synthetic data.

Table 3.3: Synthetic Data generated from the CTGAN augmentation.

S/N	Pd	W	Pb	P.I	O	K	N	D60	Ya	Suction
1	1.595024	23.74394	25.24218	23.7793	0.305585	-6.5E+08	0.392384	0.069602	38	822
2	1.748228	19.87633	25.45922	10.55828	0.224964	-8.2E+08	0.397609	0.067371	191	20
3	1.55489	18.22767	28.95152	15.73044	0.320516	-6.7E+08	0.368871	0.072973	35	315
4	1.539816	19.60095	27.84347	22.74472	0.287332	-1.3E+08	0.267411	0.069937	36	634
5	1.684196	21.45238	30.45064	18.93844	0.343418	3.76E+08	0.340046	0.066238	38	270
6	1.804752	22.05911	25.98905	6.722476	0.284142	-2E+08	0.38597	0.128827	47	1949
7	1.664946	25.30323	33.51952	11.32379	0.307012	-1.1E+09	0.29389	0.069431	40	480
8	1.660393	16.8659	28.79407	5.391353	0.249366	-1.4E+09	0.381103	0.0596	66	381
9	1.635402	18.97353	32.38551	19.26509	0.321297	2.23E+08	0.408059	0.071388	25	1901
10	1.65171	21.58423	31.28827	20.86861	0.284379	-1E+09	0.33686	0.078649	29	338
11	1.690247	21.70114	33.75695	20.09929	0.355583	-1.4E+09	0.255096	0.080705	31	1378

From the resulting data obtained via this augmentation technique, it was observed to possess the following characteristics;

Table 3.4: Properties of Data generated from the CTGAN augmentation.

Statistical Properties	p_d	W	p_b	P.I	Θ	K	N	D_{60}	α	Suction
Mean	1.6708	17.2111	30.2260	16.9299	0.2954	4.5698E8	0.3277	0.0815	43.1239	575.8094
STD	0.0749	2.2188	2.7184	4.6595	0.0284	2.6316E9	0.0317	0.0238	18.0171	392.4089
Min	1.5200	10.9000	21.8960	9.3000	0.2250	1.77E-8	0.2420	0.0580	26.0000	30.0000
25%	1.6126	15.7163	28.4275	13.1894	0.2770	1.1681E-10	0.3047	0.0648	30.6758	234.0921
50%	1.6614	17.2637	30.3853	17.0185	0.2958	2.3086E-10	0.3294	0.0694	35.9692	500.0000
75%	1.7226	18.7647	32.2386	20.7756	0.3166	4.6489E-10	0.3530	0.0985	50.7785	860.9097
Max	1.8600	22.5000	35.7960	24.0000	0.3620	2.32E10	0.3950	0.1300	130.0000	1500.00

From the data properties and the last 11 rows of the data, it can be observed that the data possess outliers as in case of the k values (coefficient of permeability), this augmentation model produces negative values (e.g -6.5E8), and some of the k values were seen to be in positive exponents (e.g 2.32E10) which are unrealistic, and impractical. These makes the outcomes from this technique unacceptable, and other techniques explored. As such in any data augmentation containing coefficient of permeability (k) values, CTGAN should be avoided as an Augmentation technique.

3.12 Perturbation; is a data augmentation technique that involves introducing small, controlled modifications (i.e perturbations) by applying small random changes to the original data in order to generate synthetic variations (Hyeyung et.al, 2024). These perturbations can include adding noise (for example, Gaussian or salt-and-pepper), slight shifts in values, or transformations that mimic real-world variability (Satou and Mitkiy, 2025). This technique generates 550 dataset inclusive of our original obtained 50 dataset. Table 4.5 shows the next 10 rows after the 500th row of dataset as augmented via this technique; the original data was in the first 50 rows of the gross data. As such, these in table 4.5 are among the newly generated synthetic data.

Table 3.5: Data generated from the Perturbation augmentation.

S/N	Pd	W	pb	P.I	O	K	N	D60	Ya	Suction
1	1.5784	16.5663	27.2995	23.8051	0.3125	2.3782E+8	0.4	0.1317	24.7443	100
2	1.6613	13.9007	24.6901	24.2895	0.2250	1.8204E+8	0.3585	0.1278	27.2902	1500
3	1.7357	13.6106	25.7540	23.7792	0.2520	2.5280E+8	0.3346	0.1295	71.2053	500
4	1.8429	11.0233	21.9624	23.6897	0.2312	6.0520E+7	0.2952	0.1311	32.2454	1000
5	1.5673	16.7625	28.2958	20.3313	0.3325	9.0036E+7	0.3715	0.0688	27.1943	100
6	1.5466	21.2363	34.4586	20.2106	0.3351	1.8763E+8	0.3876	0.0691	40.5062	500
7	1.6525	14.9195	26.7177	20.2366	0.2952	6.7349E+7	0.3472	0.0696	26.4263	100
8	1.6315	18.81545	32.6609	20.9254	0.2821	2.5460E+8	0.3584	0.0680	28.1957	1000
9	1.7013	14.3641	26.2798	20.3987	0.2875	1.7672E+8	0.3304	0.0715	69.0464	500
10	1.6690	18.6389	32.4016	20.9141	0.2935	2.0506E+7	0.3387	0.0690	61.4554	1000

From the table above, it can be observed that majority of the values of hydraulic conductivity are in positive exponents, and this makes them outliers as hydraulic conductivity signifies the speed of water through the soil pores. This parameter is practically very low almost negligible, as such this discards the data from this augmentation technique. Also, this augmentation technique produces the least quantity of synthetic data, which is scanty for machine learning works.

3.13 Bootstrap technique: Bootstrap augmentation involves generating new datasets of the same size as the original by repeatedly sampling subsets of the data with replacement. Bootstrap is also good in generating larger datasets while maintaining the original data distribution (Kohavi and Longhatham,2023). This technique generates 1000 dataset inclusive of our original obtained 50 datasets. Table 4.6 shows the last 10 rows of the dataset as augmented via this technique; the original data was in the first 50 rows of the gross data. As such, these in table 4.6 are among the newly generated synthetic data.

Table 3.6: Data generated from the Bootstrap augmentation

S/N	Pd	W	pb	P.I	O	K	N	D60	Ya	Suction
1	1.61	19.8	33.488	9.3	0.242	1.51E-10	0.315	0.058	40	1000
2	1.79	12.8	24.702	14.3	0.236	3.20E-10	0.266	0.065	40	1500
3	1.8	14.8	28.44	14.3	0.277	7.30E-11	0.263	0.065	34	500
4	1.71	17.2	31.122	14.3	0.267	1.17E-10	0.299	0.065	33	1500
5	1.75	16.5	30.625	20.5	0.236	6.15E-11	0.308	0.07	35	1500
6	1.56	21	34.32	20.5	0.336	1.89E-10	0.383	0.07	40	500
7	1.6	20.1	33.76	9.3	0.323	1.36E-10	0.319	0.058	70	1000
8	1.86	12.9	25.854	24	0.307	1.04E-10	0.287	0.13	27	30
9	1.68	18.1	32.088	9.3	0.293	1.20E-10	0.285	0.058	130	1500
10	1.56	21	34.32	20.5	0.336	1.89E-10	0.383	0.07	40	500
11	1.75	16.5	30.625	20.5	0.236	6.15E-11	0.308	0.07	35	1500

From table 3.6, it can be observed that the generated synthetic data conforms well to the trend of the original data, including the k values which often possess many outliers. This is because Bootstrap possesses the capability of generating larger datasets while maintaining the original data distribution. This fits one of the requirements for this research, but this augmentation technique was able to produce only 1000 datasets including the original 50 datasets, which looks inadequate for machine learning and shows less reliability during anomaly detection of the produced synthetic data. This leads this work to exploring more augmentation technique.

3.1.4 Interpolation: This technique was employed for augmentation in this research work due to the fact that; Interpolation augmentation in Python possess several advantages over other augmentation techniques such as; handling missing values, flexibility (as in linear, polynomial and spline), improved model performance and efficient computation. This technique generates 2550 dataset inclusive of our original obtained 50 datasets. Table 4.7 shows the next 10 rows after the 2000th row of dataset as augmented via this technique; the original data was in the first 50 rows of the gross data. As such, these in table 4.7 are among the newly generated synthetic data.

Table 3.7: Data generated from the Interpolation augmentation

S/N	pd	W	Pb	P.I	O	K	N	D60	Ya	Suction
1	1.647	15.805	27.444	17.553	0.2748	1.13E-9	0.340	0.0984	55.5282	500
2	1.654	18.670	32.517	15.150	0.3152	1.35E-10	0.325	0.0656	29.6456	154.8531
3	1.761	15.638	29.118	15.596	0.2690	9.27E-11	0.312	0.0678	41.2437	718.9069
4	1.822	15.079	29.275	24.000	0.3168	9.73E-11	0.301	0.1300	27.4895	165.2705
5	1.711	15.705	28.496	22.537	0.2695	5.04E-10	0.335	0.1049	67.4935	708.873
6	1.634	19.038	32.739	19.319	0.2931	1.43E-10	0.349	0.0690	39.5236	904.7644
7	1.663	16.908	29.645	19.125	0.2810	5.48E-10	0.337	0.1061	42.6218	232.6433
8	1.600	17.030	28.853	19.544	0.3194	3.04E-09	0.363	0.0692	27.9247	315.7676
9	1.710	17.700	31.977	10.579	0.3204	6.53E-11	0.278	0.0642	36.6113	221.8617
10	1.657	18.500	32.287	16.419	0.3375	1.48E-10	0.330	0.0792	69.0760	500

From table 3.7, it can be observed that the generated synthetic data conforms well to the trend of the original data, including the k values which often possess many outliers, this is as a result of the model's characteristic of creating new instances that lie between the originals in the data space, helping to expand datasets, reduce overfitting, and improve model generalization. It is particularly effective for handling class imbalance. This fits one of the requirements for this research. The synthetic data obtained from this augmentation technique was adopted in this work, despite conforming to the original data, it also shows the least anomaly as show in the figures below.

3.2 ANOMALY DETECTION

In order to confirm the goodness of the synthetic data, anomaly detection test was performed. Anomaly detection is defined as the process of identifying data points, events, or patterns that deviate significantly from the expected behaviour within a dataset (Moacir et. al, 2025). It is conducted in order to evaluate how well the detector generalizes to synthetic variations. Anomaly detection was carried out using the **feature distribution comparison technique**. Feature distribution comparison anomaly detection is an unsupervised or semi-supervised machine learning technique for identifying outliers or anomalies in datasets by quantifying the dissimilarity between the probability distributions of features in observed data instances and those derived from a reference model of normal behaviour (Fan et.al, 2024). This Anomaly detection technique was first used in research conducted by Gu et.al,2005. The anomaly detection was conducted on the two best techniques that produces the most realistic synthetic data, namely; Bootstrap technique and Interpolation technique.

3.3 Feature Engineering

In this, engineered features were then determined after the model studied the relevant patterns and relationships in the data. The new engineered features generated were; Dry density to Natural moisture content ratio ($P_d:w$), Bulk density to Dry density ratio ($P_b:P_d$), Natural moisture content to Volumetric water ($w:\Theta$), Dry density squared (P_d^2), Natural moisture content squared (w^2), moisture level number, density level number, and Log of hydraulic conductivity (Log K). These new features were then combined with the existing features of the data. These combined features were used to train the models for prediction of the matric suction.

3.4 Hyperparameter tuning: The hyper-parameter technique used in this work is **Grid Search** and the following are the feature importance of the respective parameters (both original and engineered features) of some of the models used;

Table 3.8: Hyper-parameter Grids for the Models Used

S/N	Models	Hyper-parameter grids
1	Ridge Regression	Ridge(alpha=1.0)
2	Lasso Regression	Lasso(alpha=1.0)
3	Random Forest	Random Forest Regressor (n estimators=100, random state=42)
4	SVR	SVR (kernel=rbf, C=1.0, gamma=scale)
5	Decision Tree	Decision Tree Regressor (random state=42)
6	XGBoost	'n_estimators': [50, 100, 200] Max_depth': [3, 5, 7], Learning_rate': [0.01, 0.1, 0.3], Subsample': [0.8, 1.0]

From table 3.8 above, the parameter grids assigned to each of the used models during the hyper parameter tuning can be observed.

Table 3.9: Random Forest Feature Importance

Model	Feature	Feature importance
Random Forest	Θ	0.2130
	Ψ_a	0.1199
	P_b	0.0224
	D60	0.0532
	W	0.0058
	$P_d:w$	0.0125
	$P_b:P_d$	0.0075
	$W:\Theta$	0.4026
	P_d^2	0.1015
	W^2	0.0060
	K	0.0257
	Log k	0.0284
	Moisture level No.	0.0005
	Density level No.	0.00010

Table 3.91: Decision Tree Feature Importance

Model	Feature	Feature importance
Decision Tree	Θ	0.2298
	Ψ_a	0.1112
	P_b	0.0136
	D60	0.0400
	W	0.0000
	$P_d:w$	0.0020
	$P_b:P_d$	0.0022
	$W: \Theta$	0.4547
	P_d^2	0.1053
	W^2	0.0103
	K	0.0114
	Log k	0.0196
	Moisture level number	0.0000
	Density level number	0.0000

Table 3.92: XGBoost Feature Importance

Model	Feature	Feature importance
XGBoost	Θ	0.0963
	Ψ_a	0.0685
	P_b	0.0352
	D60	0.0547
	W	0.0261
	$P_d:w$	0.0184
	$P_b:P_d$	0.0220
	$W:\Theta$	0.4231
	P_d^2	0.1221
	W^2	0.0000
	K	0.0441
	Log k	0.0000
	Moisture level number	0.0000
	Density level number	0.0896

From Tables 3.90 through Table 3.92, It can be observed that, hyper-parameter tuning was conducted on the combined parameters (both original and engineered features) on each of the best models in order to determine the best parameters for each prediction model (fine-tuned parameters). Models' comparison was also conducted, to determine the best model to be used for the matric suction prediction. The best model was selected based on **cross validation (CV) performance and generalization**. The table below shows the models with the metrics used for comparison.

Table 3.93: Model Comparison

S/N	Model	Train R^2	Test R^2	Test RMSE	Test MAE	CV Mean R^2	CV Std R^2
1	Linear Regression	0.6688	0.6202	239.1857	185.1373	0.6636	0.0161
2	Ridge Regression	0.6643	0.6244	237.8778	184.8021	0.6592	0.0183
3	Lasso Regression	0.6590	0.6254	237.5597	184.7460	0.6552	0.0191
4	Random Forest	0.9829	0.8646	142.8048	97.2090	0.8611	0.0098
5	SVR	0.1763	0.1987	347.4466	271.1247	0.1366	0.0154
6	Decision Tree	0.8607	0.7456	195.7673	140.3552	0.7610	0.0185
7	XGBoost	0.9985	0.8618	144.3092	96.7948	0.8712	0.0124

From the above table of model comparison, it was determined that the best model for the matric suction prediction was; XGBoost model. This is because it possesses the best comparison metrics as follows;

Best Model CV R^2 : 0.8712 ± 0.0124

Best Model Test R^2 : 0.8618

Best Model Test RMSE: 144.3092

Also, XGB stands to be the best model as it gave the lowest Test RMSE with the best training R^2 (0.9985) and the highest cross-validation (CV) R^2 value. These characteristics possessed by the model are acceptable requirements of a Strong model, this is as per the work of Khoshvaght *et.al*, (2025) and Chicco *et al.* (2021).

But, in order to rigorously compare the performance of Random Forest and XGBoost, a 5×2-fold paired t-test (Dietterich, 1998) was conducted using R^2 as the evaluation metric. The test yielded a p-value of (0.78), indicating that the performance difference between the two models is not statistically significant. Therefore, while XGBoost was selected as the final model based on marginally higher cross-validation scores, Random Forest remains equally competitive.

Feature importance analysis was then conducted on this best model. And the followings table displays the feature importance of the respective parameters used in the model.

Table 3.94: Feature importance Analysis on XGBoost model

S/N	Feature	Importance
1	Θ	0.096350
2	Υ_a	0.068466
3	P_b	0.035157
4	D60	0.054653
5	W	0.026056
6	$P_d:w$	0.018460
7	$P_b:P_d$	0.022000
8	$W:\Theta$	0.423051
9	P_d^2	0.122120
10	W^2	0.000000
11	K	0.044110
12	Log k	0.000000
13	Moisture level number	0.000000
14	Density level number	0.000000

From the conducted feature importance analysis on the best model, it is determined that the following predictors are most required by the model in matric suction prediction;

1. Saturated volumetric water content (Θ)
2. Air entry value (Υ_a)
3. Hydraulic Conductivity ('k)
4. Bulk Density (P_b)
5. D_{60} particle size parameter (D60)
6. Natural Moisture Content (w)

After the feature importance analysis on the best model, the best model and parameters were deployed. The work then Proceeds to the matric suction prediction, where another set of 10 unseen data (Data containing only the predictors without the class value i.e matric suction) were used to predict matric suction. Majority of these data are synthetic, as such it possess limitations;

1. Synthetic data lacks real-world noise (sensor glitches, typos, outliers) and complex feature interactions.
2. They build splits based on fake, clean thresholds.
3. These models rely on the correlation structure between features
4. Synthetic data that comes from generative adversarial networks (GAN), it often memorizes the training set's surface patterns without capturing the full probability distribution.
5. Synthetic data rarely generates realistic edge cases.

The following table shows a comparison of some of the predicted and original matric suction values.

Table 3.95: Comparison between Obtained and the Predicted matric suction.

S/N	P _d	W	P _b	P.I	Θ	K	N	D ₆₀	Υ _a	Obtained Ψ	XGB Predicted Ψ
1	1.60	16.79	28.54	21.94	0.293	1.53E-09	0.374	0.0947	44.24	335.40	390.74
2	1.68	15.15	27.10	18.15	0.295	2.63E-10	0.326	0.0675	44.66	183.88	230.23
3	1.74	14.80	27.56	22.63	0.246	5.16E-10	0.323	0.1065	56.29	891.79	721.29
4	1.65	19.14	33.25	14.41	0.344	8.80E-11	0.324	0.0657	79.89	500	498.62
5	1.63	19.18	32.80	18.62	0.288	1.55E-10	0.350	0.0680	36.73	1000	966.31
6	1.58	16.5	27.65	24	0.313	3.41E-09	0.395	0.13	26	100	109.22
7	1.67	14	25.05	24	0.225	1.32E-09	0.36	0.13	27	1500	1493.35
8	1.74	13.7	25.58	24	0.252	8.09E-10	0.333	0.13	70	500	547.13
9	1.84	10.9	21.90	24	0.233	1.10E-10	0.295	0.13	30	1000	998.91
10	1.58	17	28.44	20.5	0.329	3.57E-09	0.375	0.07	27	100	98.21

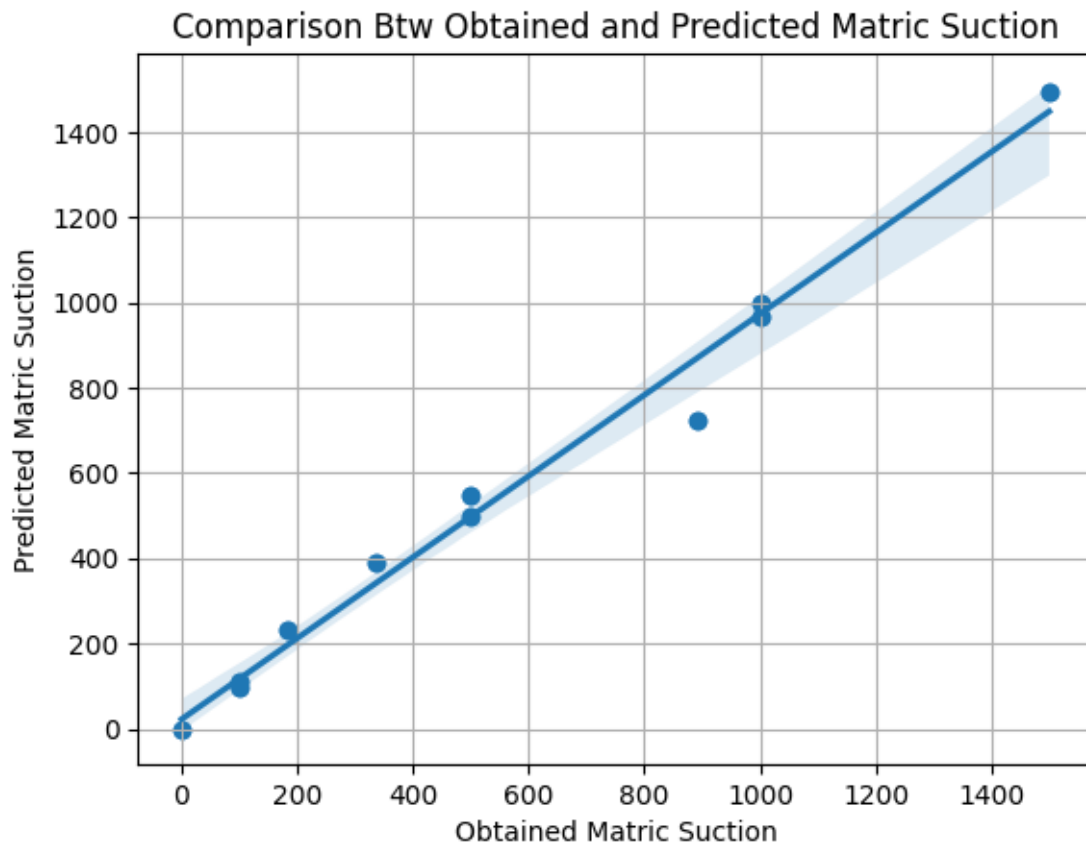


Figure 3c: Comparison Between Predicted and Obtained Suction Values

From table 4.93, the Test R^2 is **0.8618** and the CV Mean is **0.8712**. These scores are almost identical to each other (with a difference of ~ 0.01). If it were overfitting, the Test/CV scores would plummet (that is, Test $R^2 < 0.70$), which they do not.



```

import numpy as np
from sklearn.model_selection import ShuffleSplit
from sklearn.ensemble import RandomForestRegressor
from xgboost import XGBRegressor
from sklearn.metrics import r2_score
from scipy.stats import t

def fivex2_paired_ttest(X, y, model1, model2, metric=r2_score, n_repeats=5, random_state=42):
    """
    Performs the 5x2-fold paired t-test (Dietterich, 1998).

    Parameters:
    - X, y: Your features and target (numpy arrays or DataFrames).
    - model1, model2: Scikit-learn style estimators (e.g., RandomForest, XGBoost).
    - metric: Scoring function (default is r2_score).

    Returns:
    - t_stat: The t-statistic.
    - p_value: Two-tailed p-value.
    - mean_diff: Average difference (model1_score - model2_score). Positive means model1 is better.
    """
    # Ensure numpy arrays for easy indexing
    X = np.array(X)

```

Figure 3d: The P-Test Analysis

From figure 3c above, it can be observed that a comparison of obtained matric suction with matric suction values predicted from XGboost model is depicted, and the values used are real data, among the first 50 obtained from the work of Eberemu, 2008. 60% of the data was used for training, 30% for testing and 10% for vaditation. Based on this, the model can therefore be judged as very good.

4. CONCLUSION

Data obtained in the tests were fit to the models described. An optimization routine was used to fit the parametric models to the measured data by altering the fitted parameters iteratively. The maximum accuracy was achieved. The sum of the model capability is defined by some assigned metrics (i.e RMSE, R^2 , and CV). From the results obtained, it was observed that six independent variables were used as inputs, after optimization and one dependent variable (Target Matric Suction).

- i. It is therefore concluded that nine (9) soil parameters were adopted as inputs and were used for the model development in predicting the matric suction and suited the aim of this research.
- ii. The small (50 dataset) available data were successfully augmented to a quantity that will provide reliable prediction, preventing pathological overfitting and control estimation variance and the best augmented dataset was also deduced by Anomaly detection.
- iii. The models were successfully generated and validated by the 9 relevant soil parameters that were used as inputs to the model.
- iv. It is concluded that, 6 out of the 9 inputs can fit the target in predicting the Matric Suction, this as per the feature importance analysis conducted.
- v. It is also concluded that the Extreme Gradient Boost Model (XGB) is the best model, among the seven trained models in this work as per the set of hyper parameter grids possessed by the model, as shown in Table 3.8. And as per the Cross-validation metrics comparison of the models in Table 3.9.

ACKNOWLEDGEMENTS

I am most indebted to my supervisor in the person of Ass.Prof. Engr. G.H Yunusa of Civil Engineering Department. He had been enduring and supportive in the course of this work. I am most grateful for his commitment to the work.

I am grateful to the department of civil engineering for making all facilities available for my work. I wish also to thank, Mal. Ahmad Abdulnasir Shu'aib, Dr. Aminu Sulaiman, Engr.Dr. Muttaqa Uba Zango and Dr.Aminu Dar'dau Rafin-Dadi for their machine Learning assistances.

Grateful acknowledgement must also be made to my Beloved Wife and the following relatives and friends for their immense help, prayers and moral support throughout my school days; Amina Shehu Karaye, Engr Abubakar Jibrin, Engr. Dr. Ibrahim Khalil Umar, Musa Shehu Karaye, Nazifi Uba Isa, Abubakar Kumo, Haj. Zulaiha Abdullahi, Ahmad AbdulNasir Shuaib, Engr. Muhammad Umar, Engr. Inuwa Barau Inuwa, Engr. Salma B.Gambo, Bldr. Muhammad Idris etc

More importantly, I thank Almighty Allah for guiding me in all spheres of my life and directing towards this noble profession of Civil Engineering.

5. REFERENCES

- Adiputra, N.M.I., & Wanchai, P. (2024). "CTGAN-ENN: a tabular GAN-based hybrid sampling method for imbalanced and overlapped data in customer churn prediction." *Journal of Big Data*, Volume 11, Issue 1, pages; 1-25
- Adrian O. Eberemu (2008) "Evaluation of Bagasse Ash Treated Lateritic Soil as a suitable Material for Waste Landfill Barrier (Liner) and Cover" An Unpublished Phd Dissertation, Ahmadu Bello University, Zaria, Nigeria.
- Belcic, I. & Stryker C. (2025). What is hyperparameter tuning? IBM. <https://www.ibm.com/think/topics/hyperparameter-tuning>
- Björnelid, O., Carlsson, M., & Löwe, W. (2023). Case study – Feature engineering inspired by domain experts on real world medical data. *Intelligence-Based Medicine*, 8, Article 100110. <https://doi.org/10.1016/j.ibmed.2023.100110>
- Bishop A.W. and Blight G.E. (1963) "Some Aspects of Effective Stress in SATURATED AND PARTLY SATURATED SOILS" *Geotechnique*, Volume 13, Issue 3, Pages;177-197, doi: doi.org/10.1680/geot.1963.13.3.177
- Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7, Article e623. <https://doi.org/10.7717/peerj-cs.623>
- Dieudonne A.C, Levasseur S., Charlier R., Della Vecchia G., C. Jommi (2011) "A Water Retention Model For Compacted Clayey Soils" Research Article, Department of Geosciences and Engineering, TU Delft, Delft, The Netherlands.
- Elgeldawi, E., Sayed, A., Galal, A.R., & Zaki, A.M. (2021). "Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis" *Journal of Informatics*, Volume 8, No. 79.
- Fan, D., Li, M., Han D. (2024). Feature Aggregation Network for Memory-Based Industrial Anomaly Detection. In 2024 IEEE 9th International Conference on Data Science in Cyberspace (DSC) (pp. 677–683). IEEE. <https://doi.org/10.1109/DSC63484.2024.00101>
- Guo, Y., & Birgisson, B. (2025). Analytical prediction of soil–water characteristics curves for granular pavement base course materials. *Journal of Transportation Geotechnics*, Volume 55, 101704. <https://doi.org/10.1016/j.trgeo.2025.101704>
- Huang, H.-N., Chen, H.-M., Lin, W.-W., Huang, C.-J., Chen, Y.-C., Wang, Y.-H., & Yang, C.-T. (2023). Employing feature engineering strategies to improve the performance of machine learning algorithms on echocardiogram dataset. *Digital Health*, 9, Article 20552076231207589. <https://doi.org/10.1177/20552076231207589>
- Hyeyoung Maeng, Eckley I. A., & Fearnhead, P. (2024). Collective Anomaly Detection in High-Dimensional VAR Models. *Statistica Sinica*, Volume33, Issue 2, ISSN: 1603-1627
- Khalili, N., & Selvadurai, A. P. S. (2003). A fully coupled constitutive model for thermo-hydro-mechanical analysis in elastic media with double porosity. *Geophysical Research Letters*, Volume 30, Issue 24, Article 2268. <https://doi.org/10.1029/2003GL018838>

- Khoshvaght, H., Permala, R. R., Razmjou, A., & Khiadani, M. (2025). A critical review on selecting performance evaluation metrics for supervised machine learning models in wastewater quality prediction. *Journal of Environmental Chemical Engineering*, 13(6), Article 119675. <https://doi.org/10.1016/j.jece.2025.119675>
- Krisnanto S., Rifdah S.A., Ananda A.A and Putri N.A (2025) “Analysis of the stress state variables for Unsaturated soils with Compressible and Incompressible soil solids” *Geomate Journal*, Volume 28, Issue 125.
- Nowamooz, H., & Masroufi, F. (2010). Volumetric strains due to changes in suction or stress of an expansive bentonite/silt mixture treated with lime. *Comptes Rendus. Mécanique*, Volume 338, Issue 4, page: 230-240, DOI: [10.1016/j.crme.2010.03.011](https://doi.org/10.1016/j.crme.2010.03.011)
- Nitsch A., Asaad A., Wichtmann T., Emmerich K., and Baille W., (2026) “Swelling Behaviour of Compacted Ca/Mg Bentonite during hydration on different scales, *Journal of Applied Clay Science* (Elsevier), Volume 281, March 2026.
- Osame P.U, Peretomode E., and Hawez H.K.(2025) “Advances in Flow of Water Through Variably Saturated Soils; A Review of Model Approaches and Experimental investigations with use of Sensors, *MDPI Sensors Journal*, Volume 25, Issue 22.
- Satou, H., Mitkiy, A., Collins, E., & Kingston, F. (2025). Geometrically Regularized Transfer Learning with On-Manifold and Off-Manifold Perturbation. *ArXiv*, abs/2505.15191.
- Simon Salager, Marta Rizzi & Lyess Laloui (2011) “An innovative device for determining the soil water retention curve under high suction at different temperatures” volume 6, ISSN:135–142.
- Trost, C. O. W., Žák, S., Schaffer, S., Walch, L., Zitz, J., Klünsner, T., Leitner, H., Exl, L., & Cordill, M. J. (2025). Explainable machine learning and feature engineering applied to nanoindentation data. *Materials & Design*, 253, Article 113897. <https://doi.org/10.1016/j.matdes.2025.113897>