Having Deep Investigation on Predicting Unconfined Compressive Strength by Decision Tree in Hybrid and Individual Approaches

Qingqing Zhang, Lei Wang*, Hongmei Gu
Information Technology and Cultural Management Institute,
Hebei Institute of Communications, Shijiazhuang 051430, Hebei, China

***Abstract*—In the field of geotechnical engineering Rocks' unconfined compressive strength (UCS) is an important variable that plays a significant part in civil engineering projects like foundation design, mining, and tunneling. These projects' stability and safety depend on how accurately UCS predicts the future. In this study, machine learning (ML) techniques are applied to forecast UCS for soil-stabilizer combinations. This study aims to build complex and highly accurate predictive models using the robust Decision Tree (DT) as a primary ML tool. These models show relationships between UCS considering a variety of intrinsic soil properties, including dispersion, plasticity, linear particle size shrinkage, and the kind of and number of stabilizing additives. Furthermore, this paper integrates two meta-heuristic algorithms: the Population-based vortex search algorithm (PVS) and the Arithmetic optimizer algorithm (AOA) to enhance the precision of models. These algorithms work in tandem to bolster the accuracy of predictive models. This study has subjected models to rigorous validation by analyzing UCS samples from different soil types, drawing from historical stabilization test results. This study unveils three noteworthy models: DTAO, DTPB, and an independent DT model. Each model provides invaluable insights that support the meticulous projection of UCS for soil-stabilizer blends. Notably, the DTAO model stands out with exceptional performance metrics. With an R^2 value of 0.998 and an impressively low RMSE of 1.242, it showcases precision and reliability. These findings not only underscore the accuracy of the DTAO model but also emphasize its effectiveness in predicting soil stabilization outcomes.**

Keywords*—Unconfined compressive strength; machine learning; decision tree; population-based vortex search algorithm; arithmetic optimizer algorithm

I. INTRODUCTION

Geotechnical Engineering (GE) is no exception to the widespread adoption of ML techniques in several industries. Applications of ML at GE cover a broad variety of responsibilities, from landslip from detection to prediction of material properties. Creating predictive models for GE issues has demonstrated astounding success and cutting-edge technology. It is becoming increasingly clear that there is a growing need for more accurate predictive models to address various geotechnical challenges in various domains. Continuous improvement and advancement are necessary to enable the wide-scale adoption of such models, providing more precise and practical answers to various geotechnical problems [1], [2], [3].

Compaction of loose soils is a critical component of engineering projects because it increases the weight per unit area of engineering structures like earth dams and highway embankments. This compaction process improves soil endurance, boosts load-bearing capacity, and stabilizes embankment slopes to prevent settlement problems, going beyond merely increasing soil strength [4], [5]. Additionally, compaction has various advantages, including increases in density, volume, permeability, waterproofing, and porosity. Together, these improvements raise the soil's general quality and increase its capacity to support structural loads. UCS is a critical component of geomechanical modeling, particularly in investigating mechanical rock behavior [6]. The ultimate compressive stress, or UCS, that a rock can withstand when subjected to controlled, uniaxial loading before undergoing failure. Rock mechanics, which combines theoretical concepts with real-world applications, illuminates how rocks respond to various stress situations [7], [8]. In areas like the production of solid materials and wellbore stability, particularly in the petroleum operations context, the effects of rock failure have wide-ranging effects. In drilling operations, guiding bit hydraulics, determining the ideal mud weight, controlling costs, and improving drilling efficiency overall, the availability of UCS data from subsurface formations is of utmost importance [9], [10], [11]. Due to its crucial role in resolving geotechnical issues, the assessment of UCS is a fundamental pillar in rock engineering [12]. Direct UCS measurement is performed using the Unconfined Compression Test (UCT), a standardised procedure that has the support of both the American Society for Testing and Substances ($ASTM$) and the International Society for Rock Mechanics ($ISRM$) [13]. However, performing UCS tests in a lab requires carefully prepared core samples in addition to time and resources. When collaborating with brittle, thinly bedded, or severely fractured rock formations, it can be challenging to meet these requirements. Many researchers recommend indirect approaches for UCS prediction in response to these difficulties. These techniques, which include the Brazilian tensile strength test (BTS), the point load index test ($Is(50)$), and the ultrasonic test (Vp), provide quick, affordable, and transportable substitutes for UCS testing. Combined with engineering knowledge, these correlated index tests offer applicable initial UCS evaluations. Direct mechanical rock property evaluation is based on experiments performed on extracted core samples, which offer insights into actual stress conditions and mechanical characteristics. These tests use

various techniques, including point load tests, scratch tests, Schmidt hammer tests, and uniaxial and triaxial compressive strength tests. Together, these methods create the benchmark for identifying property [14], [15], [16]. To secure representative core samples, however, obtaining a continuous UCS profile along wellbores presents difficulties that include high costs and labor-intensive procedures. Indirect approaches have been developed to get around this restriction, bridging the gap by proving relationships between petrophysical well-log data and rock characteristics. UCS has significantly impacted foundation design, slope stability analysis, and structural integrity. Its significance extends beyond rocks to encompass a variety of materials, including industrial wastes and soils. In stabilized materials, UCS is crucial, impacting both the aesthetics and performance of pavements [17]. However, determining a material's UCS requires takconsidering several factors, including physicochemical characteristics, the kind of cementitious admixtures used, and curing time. These factors call for carefully thought-out laboratory experiments and specialized tools. The pursuit of accuracy, reflected in the specimens' sizes, makes these tests reliable [18], [19].

This study centers around predicting critical soil properties, precisely the UCS outcomes, using a ML approach. Given the challenges in obtaining experimental data, this study focuses on improving the DT algorithm's performance. To overcome this challenge, a combination of two algorithms is employed, including the PVS and the AOA. The research highlights the significant positive impact of optimizing the planning and building of UCS buildings for the infrastructure industry. Through the compilation of a large-scale UCS dataset, this study enables comparison analyses to evaluate the efficacy of the suggested framework. The study's conclusions provide useful insights into accurately predicting UCS in civil engineering projects. This approach revolves around forecasting UCS by incorporating the DT algorithm into the ML strategy. This addresses the complexity of acquiring empirical UCS data by optimizing the DT model's parameters through the integration of PVS and AOA algorithms. In essence, this research provides practical guidance and essential information for tackling UCS prediction, a crucial aspect of soil behavior in civil engineering.

II. Literature Review

The aforementioned challenges with the fundamental sample preparation required for standard testing and the limitations of statistical models may be addressed with the help of certain predictive ML models. Meulenkamp and Grima [20], for example, used Artificial Neural Networks (ANN) to predict UCS and discovered that ANN in 194 rock samples had better generalization than statistical models. Sonmez et al. [21] used regression and fuzzy inference systems (FIS) to predict UCS and elasticity modulus simultaneously, observing that regression performed better for elasticity modulus and that FIS had a good prediction accuracy for UCS. Regression and fuzzy models were used by Gokceoglu and Zorlu [22] to forecast the elasticity modulus and UCS in difficult rocks. Regression and a neural network were used by Dehghan et al. [23] to predict these values; regression was not as successful as the neural network. To estimate UCS for three different rock types, Mishra and Basu [24] employed multiple regression and FIS, proving the superiority of these methods over simple regression. The predicted accuracy of the UCS estimate has been substantially enhanced by several groundbreaking research.

III. Materials and Methodology

A. Data Gathering

In order to ensure the accuracy of predicting the UCS of rocks, it is crucial to compile a comprehensive dataset that includes relevant input variables. This endeavor involves carefully considering various factors and has been undertaken with unwavering commitment. As part of this pursuit, the data is meticulously divided into 70% as the training set, 15% as the validation set, and 15% as the testing set. It has been repeatedly demonstrated and scientifically verified that this allocation improves the performance of prediction models [25]. The prediction of UCS is accomplished using a DT model, which effectively leverages the predictive capabilities inherent in the aforementioned variables. These input parameters, accompanied by their detailed definitions and measurement units, serve as critical components in developing predictive models [26]. The subsequent section delineates the process of collecting data, providing an in-depth overview of each input variable:

1) Bulk Density (BD): Bulk Density quantifies the mass of a rock specimen per unit volume, measured in kilograms per cubic meter (kg/m^3). This parameter offers valuable insights into the rock's density and compactness.

2) Brazilian Tensile Strength (BTS): Brazilian Tensile Strength (BTS) evaluates a rock's resistance to tensile forces, expressed in megapascals (MPa). It is determined through the Brazilian Tensile Strength Test, wherein a rock specimen undergoes diametrical compression until failure occurs.

3) Dry Density (DD): Dry Density signifies the mass of a rock specimen per unit volume when completely dry, akin to Bulk Density. It is also measured in kilograms per cubic meter (kg/m^3) and characterizes the rock's density under dry conditions.

4) P-Wave Velocity (Vp): P-Wave Velocity (Vp) gauges the speed at which compressional ($P-wave$) seismic waves propagate through a rock specimen, expressed in meters per second (m/s). This parameter provides valuable insights into the rock's elastic properties and structural integrity.

5) Saturated Rock name (SRn): Saturated Rock Name (SRn) denotes a rock's compressive strength when saturated with water, measured in megapascals (MPa). Understanding how water impacts a rock's strength characteristics is essential, and SRn plays a vital role in this context.

6) Point load index (Is (50)): Point Load Index ($Is\ (50)$) is determined through the Point Load Index Test, which assesses a rock's strength under concentrated loads. It is quantified in megapascals (MPa) and provides insights into the rock's durability [27].

UCS represents the ultimate objective within this dataset. It signifies the maximum axial load a rock specimen can endure without lateral confinement, and it serves as the variable to be predicted using the other input parameters [28]. The successful prediction of UCS in rocks is contingent upon the quality and comprehensiveness of the dataset, encompassing the input

variables outlined in Table I. Diligent data collection and preprocessing are indispensable stages in constructing accurate predictive models, which contribute significantly to both rock mechanics and geotechnical engineering. Fig. 1 visually portrays the distribution relationship between the input variables and UCS, facilitating the assessment of how changes in one input parameter positively or negatively affect UCS concerning the other inputs [29].

TABLE I. DATA PROPERTIES OF UCS AND INPUT

Features	Dataset Components						
	BD (kg/m3)	***BTS (MPa)***	***DD (kg/m3)***	***Vp (m/s)***	***SRn (MPa)***	***Is(50) (MPa)***	***UCS (MPa)***
Min	0.00	0.00	0.00	1247.00	0.00	0.10	5.50
Max	3.54	4.20	3011.00	6440.00	45.40	6.07	108.68
Mean	0.97	0.83	1669.73	4092.11	23.75	2.51	47.93
St.Dev.	1.276067	1.226962	1308.101	1722.211	18.84653	1.568194	26.84946

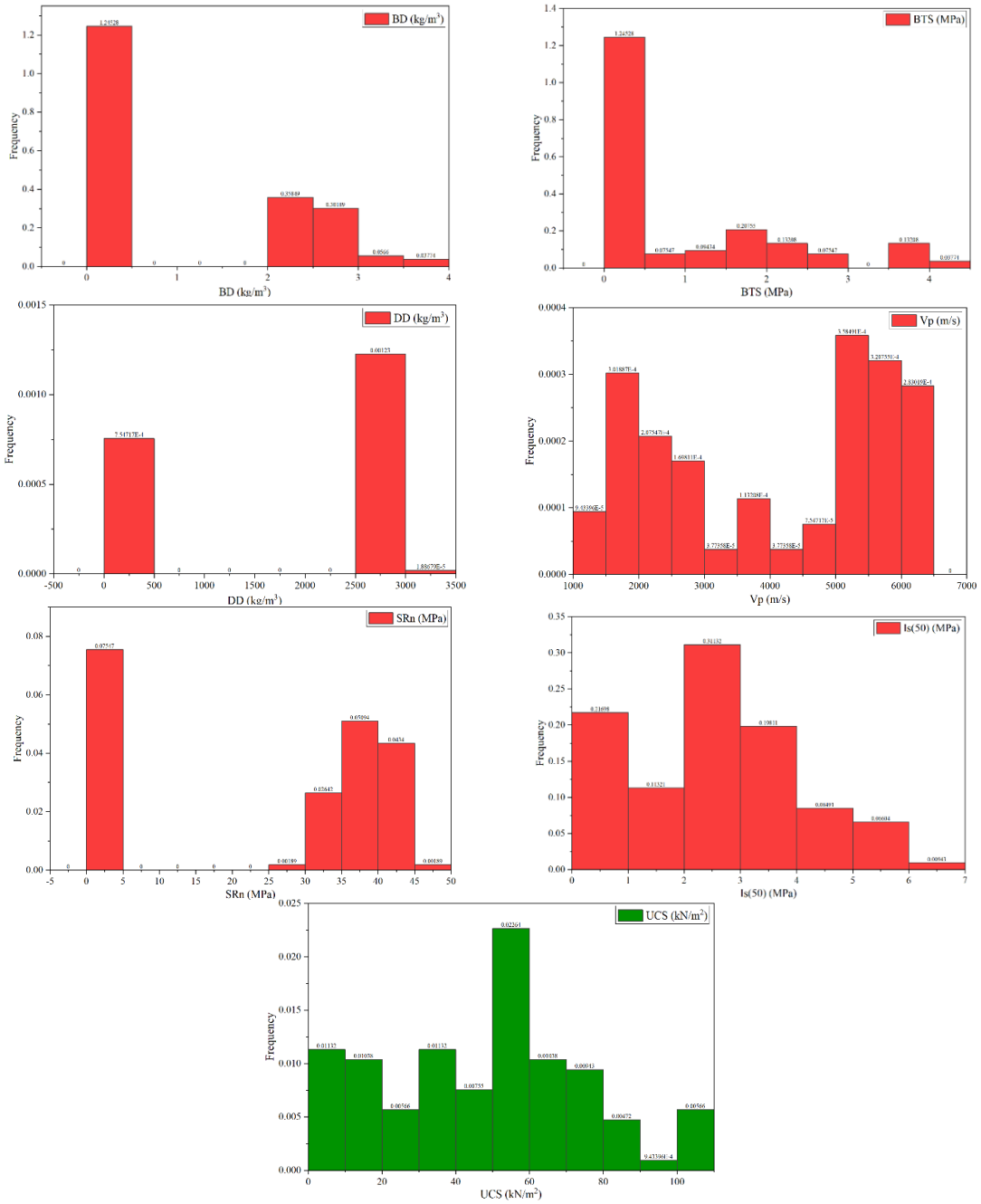

Fig. 1. The input and output distribution plot.

B. Decision Tree (DT)

The DT is a popular supervised learning method for handling regression and classification problems. Because of the tree's split structure or hierarchy, regression analysis may still be used to forecast the expected outcome based on independent variables in cases when a precise category grouping or classification is lacking [30], [31]. The model in Fig. 2 illustrates a basic DT with two continuous variables, x_1 and x_2, whose values are all between 0 and 1, and one binary target variable, Y, with values that are either 0 or 1. Additionally, as Fig. 3 illustrates, the structure may be seen as a segmented geographic region. Dividing the sample space into distinct, well-defined, and comprehensive subspaces is a typical analytical paradigm. Each of these sections is linked to a particular leaf node, which denotes the result of a series of subsequent decision-making processes. In a DT, each record has a single segment that acts as its home, known as a leaf node. The main objective of utilizing DTs for analysis is to find the most effective model that can precisely divide all available data into discrete segments [32].

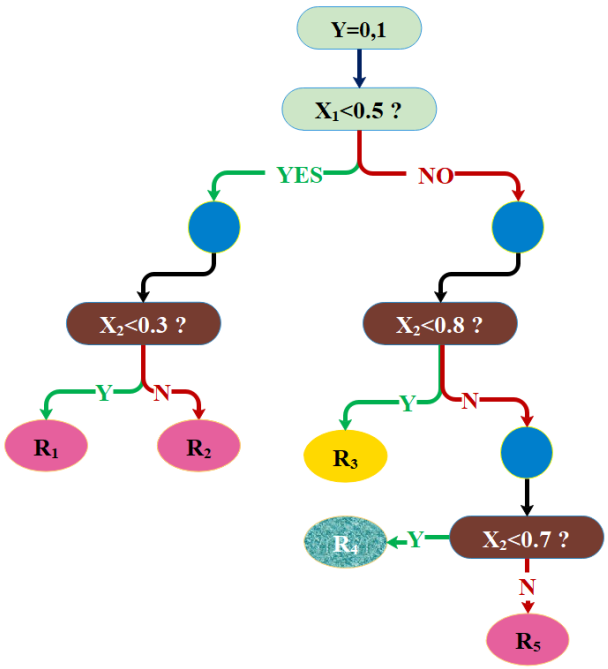

Fig. 2. Y is a binary target variable used in this example DT [33].

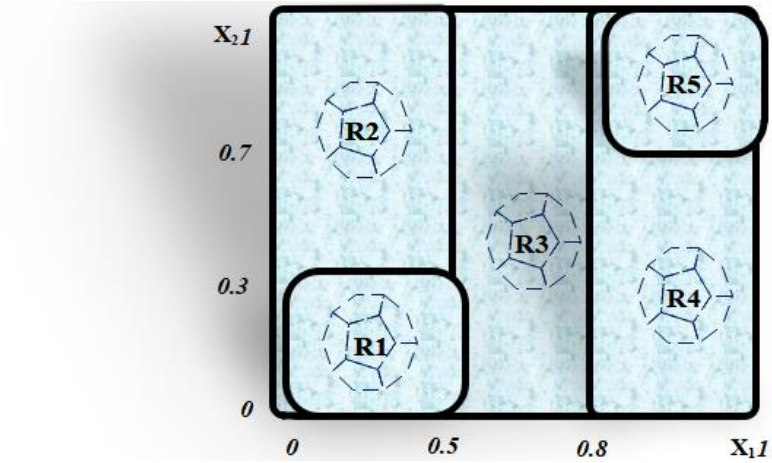

Fig. 3. Sample space view with DT [33].

Nodes and branches are the basic components of a DT model, and splitting, stopping, and pruning are essential steps in its development [34].

C. Population-Based Vortex Search Algorithm (PVS)

As a metaheuristic relying on a solitary explanation, the Vortex Search algorithm demonstrates effective exploitation capabilities, allowing it to execute rapidly [35]. The Gaussian distribution centered on a single point is used by the Vortex Search (VS) method to produce fresh candidate solutions. Nevertheless, despite attempts to promote variety in the search area, this may result in early convergence in some situations. On the other hand, population-based tactics perform better during the search space's exploration stage, when it's necessary to

conduct a detailed analysis of unknown regions. These methods create new points based on the understanding gained from each iteration's previous points [36]. A community-driven VS algorithm known as PVS is presented in this study. There are 3 stages to the algorithm: a) initialization, b) the primary phase, and c) another phase.

1) Initializing: Control parameters including population size ($psize$), vortex size ($vsize$), termination condition, and likelihood of mutation (η_m) are specified during the algorithm's initialization phase. The entire quantity of potential fixes generated in a single iteration is represented by the value of $psize$, which is divided equally by halved to produce the value of $vsize$, which is equal to $psize/2$. The value of $vsize$ determines how many candidate solutions (CS) are generated in the initial stage and the subsequent stage generates the remaining CSs from ($vsize + 1$) to $psize$ [37]. The maximum number of function evaluations ($maxFEs$) serves as the method's halting criterion, and the polynomial mutation used in the second phase of the procedure utilizes the probability value η_m. Additionally, μ_0 and q_0 are calculated using, respectively, Eq. (1) and Eq. (2).

$$\mu_0^i = \frac{upper_i + lower_i}{2} \tag{1}$$

$$q_0^i = \sigma_0^i = \frac{\max(upper_i) - \min(lower_i)}{2} \tag{2}$$

2) First phase: The first iteration of this phase involves randomly generating the entire population of size $p$size. Merely 50% of the population ($vsize$) is created at random in the subsequent iterations. After this phase, updating the focal point (μ) is the optimal course of action. In this phase, the Gaussian distribution is used to generate half of the inhabitants, in accordance with Eq. (3) from the original VS algorithm. A population-based technique with selection pressure is used to update half of the population, while the other half is subjected to the best-center-oriented exploitation procedure. Overflowing solutions are recast into the designated range by applying Eq. (4).

$$s_i^t(x_i^t \Big| \mu_t, v) = ((2\pi)^d |v|)^{-\left(\frac{1}{2}\right)} e^{(-1/2\left(x_i^t - \mu_t\right)^T v^{-1}(x_i^t - \mu_t)} \tag{3}$$

$$\begin{aligned} s_i(lower_i \vee s_i) upper_i \rightarrow s_i \\ = rand \times (upper_i - lower_i) \\ + lower_i \end{aligned} \tag{4}$$

Although it is not explicitly specified, the starting population is generated by using the starting centering point (μ_0) in the original VS algorithm. The midpoints of the population are selected at random after the initial population. This is where a modification to the VS algorithm is proposed, leading to variations of the PVS method. PVS_a designates the version in which μ_0 was present in the starting population, whereas PVS_b designates the version in which it was absent. The computed center point μ_0 is used as the initial potential fix POP (1) in the population during the first iteration of PVS_a, and the remainder $psize - 1$ candidate solutions POP (2: $psize$) are created at random. On the other hand, $psize$ candidate solutions POP (1: $psize$) are generated at random to form the initial population of PVS_b.

3) Second phase: By relying on interactions between potential solutions throughout the course of iterations to modify their locations during the search, population-based algorithms vary from single-solution-based algorithms. The basic strategy involves expressing the local or global experiences of potential solutions in a vector format to enable information sharing, while the updating method may differ based on the specific algorithm employed. Potential solutions are matched by the PVS algorithm using a proportional selection mechanism. This approach incorporates the ABC algorithm's onlooker bee phase's location update procedure, with some modifications made specifically to address minimization issues. Eq. (5) is employed to calculate the selection pb for each candidate solution.

$$\begin{aligned} pb_i &= csum_i / csum_{psize} \\ csum_i &= \sum_{j=1}^{i} normp_i \ \ and \\ normp_i &= p_i / \sum_{i=1}^{psize} p_i \ \ and \\ p_i &= 0.9 \times \left(max\{\vec{f}\} - f_i\right) + 0.1 \end{aligned} \tag{5}$$

f symbolizes the suitability value of the ith solution and $max\{\vec{f}\}$ p_i signifies the $ith's$ scaled fitness value solution for minimization, which was achieved by converting the values of the objective function, which range from maximization to minimization. denotes the population's greatest fitness value at this time. The function $normp$ is used to calculate the probability values obtained by normalizing the p values in the 0.5– 1 range. The cumulative sum vector of norm p values is denoted by $csum$. The remaining 50% of the populace, for each solution CS_i where i ranges from $vsize + 1$ to $psize$, using the prob vector, an arbitrary neighbor solution is chosen from each of the population's solutions. To produce a new solution (CS_{new}), the amount of a randomly chosen dimension is updated using Eq. (6). The acquired dimension value is next subjected to an Eq. (7) check for limit exceedance.

$$\begin{aligned} CS_{new} = CS_{current} \ then \ CS_{new}^i \\ = CS_{current}^i + (CS_{current}^i \\ - CS_{neigbour}^i) \times (r - 0.5) \times 2 \end{aligned} \tag{6}$$

$$CS_{new} = \begin{cases} lower_i, & CS_{new}^i < lower_i \\ CS_{new}^i, & lower_i \le CS_{new}^i \le upper_i \\ upper_i, & CS_{new}^i > upper_i \end{cases} \tag{7}$$

The usefulness of the new solution CS_{new} is calculated with a random number r ranging from 0.5 to 1, and subsequently contrasted with the current solution's fitness value $CS_{current}$. If the If CS_{new} has a higher fitness level than $CS_{current}$, the former takes the place of the latter. On the other hand, a mutant solution

CS_{mutant} is produced by the polynomial mutation in accordance with Eq. (8) if CS_{new} is not superior to $CS_{current}$.

$$CS_{mutant} = CS_{current} + \delta_q \times (upper - lower)$$

$$\delta_q = \begin{cases} \left[\begin{matrix} 2r+(1-2r) \\ (1-\delta_1)^{\eta_{m+1}} \end{matrix}\right]^{\frac{1}{\eta_{m+1}}}, & if\ r \le 0.5 \\ 1 - \left[\begin{matrix} 2(1-r)+2(r-0.5) \\ (1-\delta_2)^{\eta_{m+1}} \end{matrix}\right]^{\frac{1}{\eta_{m+1}}}, & otherwise \end{cases} \quad (8)$$

$$\delta_1 = \frac{CS_{current} - lower}{upper - lower}$$

$$\delta_2 = \frac{upper - CS_{current}}{upper - lower}$$

In this instance, a random number rnd is created for each dimension between 0.5 and 1. Further processing is applied if rnd is less than the η_m value, which is calculated in this section by dividing 1 by the dimensionality of the problem. The research suggests that the polynomial mutation operator is the best method for overcoming the problem of preventing local peaks and maintaining variety in the search space, which is a major roadblock for metaheuristics. After the answer is warped using a polynomial probability distribution, the polynomial mutation operator generates a perturbation effect. Next, a contrast is drawn between $CS_{current}$ and CS_{mutant} using a greedy selection process. After completing this process, the best option discovered is applied to update the center point (μ).

After the current generation is finished, the radius size for the following generation is decreased by calculating Eq. (9). As long as the PVS algorithm completes the greatest number of function evaluations, it keeps running. First, $vsize$ solutions inside the lowered radius are repeated, and in the second phase, random data is added to the answers, which comprise the remaining 50% of the population.

$$r_t = \sigma_0 \times \frac{1}{x} \times \Gamma(x, a_t)$$
$$where\ \ a_t = \frac{(MaxFEs - Fes)}{MaxFEs} \quad (9)$$
$$then\ \ if\ \ (a_t \le 0) a_t = 0.1$$

D. Arithmetic Optimizer Algorithm (AOA)

This algorithm was proposed by Abualigah, employing some mathematical operators and formulas in 2020 [38]. The AOA algorithm begins with a random population of solutions. At each iteration, the objective value for each solution is computed. This algorithm has two control parameters called M and \, which need to be updated before updating the solution position in the following:

$$M(i) = Min + i \times \left(\frac{Max - Min}{I}\right) \quad (10)$$

The function's value at the current iteration is shown by $M(i)$, the maximum iteration is shown by I, and the $minimum$ and $maximum$ values for the bounds of M are shown by Min and Max.

$$P(i) = 1 - \left(\frac{i}{I}\right)^{\frac{1}{c}} \quad (11)$$

Here, the coefficient of the mathematical optimizer probability (P) determines the function's value at the ith iteration. After updating the P and M, it also generates a random number called r_3 to switch between exploitation and exploration. The search makes use of Eq. (12).

$$x_{i,,j}(t+1) =$$
$$\begin{cases} \frac{best(x_j)}{(P+\alpha)} \times (ub_j - lb_j) \times M + lb_j & if\ \ r_1 < 0.5\ (a) \\ best(x_j) \times P \times (ub_j - lb_j) \times M + lb_j & if\ \ otherwise\ (b) \end{cases} \quad (12)$$

Furthermore, Eq. (13) is utilized for exploitation:

$$x_{i,,j}(t+1) =$$
$$\begin{cases} best(x_j) - P \times (ub_j - lb_j) \times M + lb_j & if\ \ r_2 < 0.5\ (a) \\ best(x_j) + P \times (ub_j - lb_j) \times M + lb_j & if\ \ otherwise\ (b) \end{cases} \quad (13)$$

It should be noted that section 2.3 mentions the AOA parameters, which are the same as $DAOA's$. Additionally, the AOA flowchart is shown in Fig. 4.

E. Performance Evaluation Methods

This article assesses the models using several metrics, such as the previously stated Mean Absolute Relative Error ($MARE$), Correlation Coefficient (R^2), Mean Square Error (MSE), Normalized Root Mean Squared Error ($NRMSE$), and Root Mean Square Error ($RMSE$). Excellent performance of the algorithm during the phases of training, validation, and testing is indicated by a high R^2 value. On the other hand, lower $RMSE$ and MSE values are preferable because they exhibit reduced model inaccuracy. Eq. (14) to (18) are used to calculate these metrics.

Coefficient of Correlation

$$R^2 = \left(\frac{\sum_{i=1}^{W}(h_i - \bar{h})(q_i - \bar{q})}{\sqrt{[\sum_{i=1}^{W}(h_i - h)^2][\sum_{i=1}^{W}(q_i - \bar{q})^2]}}\right)^2 \quad (14)$$

Root Mean Square Error

$$RMSE = \sqrt{\frac{1}{W}\sum_{i=1}^{W}(q_i - h_i)^2} \quad (15)$$

Mean Square Error

$$MSE = \frac{1}{W}\sum_{i=1}^{w} q_i^2 \quad (16)$$

Mean Absolute Relative Error

$$MARE = \frac{1}{W}\sum_{i}^{w} \frac{|q_i - h_i|}{|\bar{q} - \bar{h}|} \quad (17)$$

Normalized Root Mean Squared Error

$$NRMSE = \frac{RMSE}{q_i - \bar{q}} \quad (18)$$

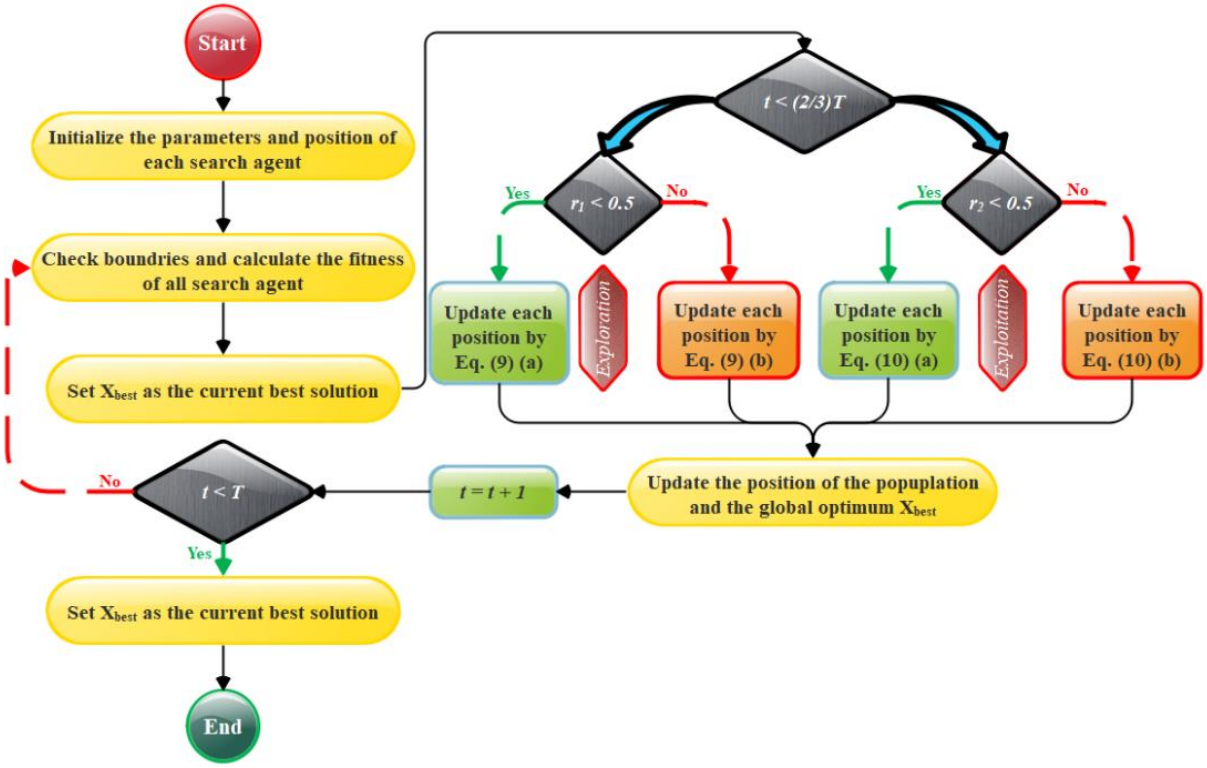

Fig. 4. The flowchart of AOA.

The anticipated and experimental values are denoted by the variables h_i and q_i in these equations, respectively. The mean values of the expected and experimental samples are denoted by the symbols $\bar{h}$ and $\bar{q}$, respectively. Alternatively, W indicates how many samples are being examined.

The study employed three models, namely DT, $DTAO$, and $DTPB$, for the prediction of UCS. These models underwent evaluation using experimental measurements in Table II, with the evaluation process divided into four phases: training (70%), validation (15%), testing (15%), and overall assessment (100%), ensuring an unbiased evaluation. To comprehensively assess and compare the algorithms, five statistical metrics were utilized, including $NRMSE$, MSE, R^2, $RMSE$, and $MARE$. The primary metric for assessing model performance was R^2, indicating how effectively the independent variable accounts for variance in the dependent variable.

TABLE II. THE OUTCOME OF THE MODELS CREATED FOR DT

Model	*Phase*	Index values				
		RMSE	***R^2***	***MSE***	***NRMSE***	***MARE***
DT	*Train*	3.537	0.984	12.518	0.048	0.121
	Validation	5.375	0.958	28.887	0.336	0.095
	Test	5.841	0.972	34.122	0.365	0.103
	All	4.271	0.975	18.250	0.040	0.114
DTAO	*Train*	1.242	0.998	1.543	0.017	0.020
	Validation	3.715	0.988	13.802	0.232	0.052
	Test	4.299	0.989	18.480	0.269	0.072
	All	2.439	0.993	5.950	0.023	0.032
DTPB	*Train*	2.444	0.992	5.971	0.033	0.044
	Validation	5.254	0.985	27.605	0.328	0.066
	Test	5.382	0.981	28.965	0.336	0.085
	All	3.565	0.987	12.708	0.034	0.053

The *DTAO* model demonstrated superior performance during the training phase, boasting the highest R^2 value (0.998) among all models. In contrast, the *DT* model exhibited slightly lower training-phase R^2 values at 0.984. Additionally, *RMSE*, an error indicator, was evaluated in the study, with a range of 1.242 to 5.841. The *DT* model had the largest RMSE, while the *DTAO* model showcased the lowest. In the training phase, the *DT* model had the highest *NRMSE* value (0.048), whereas the *DTAO* model had the lowest (0.017). The *DTAO* model also excelled in terms of *MARE*, with a value of 0.020, while the *DT* model had the highest *MSE* among the models evaluated, with a value of 12.518. Overall, the findings showed that in certain phases, the *DTAO* model performed better than the *DT* and *DTPB* models. But when choosing a model for practical applications, other aspects including computational effectiveness, model complexity, and simplicity of implementation should also be taken into account. However, the results indicate that *AOA* modification improved the *DT* model's prediction of *UCS* substantially.

The performance of hybrid models is efficiently compared using a scatter plot in Fig. 5, which considers two important parameters: R^2 and *RMSE*. R^2 is a robust indicator of agreement, while *RMSE* quantifies the extent of deviation. The plot's central axis acts as a reference point, and the proximity of individual data points to this axis reveals the precision of the models. Notably, the *DTAO* model stands out as a model of exceptional accuracy, as indicated by its data points closely clustering around the central axis, highlighting minimal divergence. In contrast, the *DTPB* and *DT* models exhibit similar performance levels, with their respective data points scattered widely, indicating significant variability.

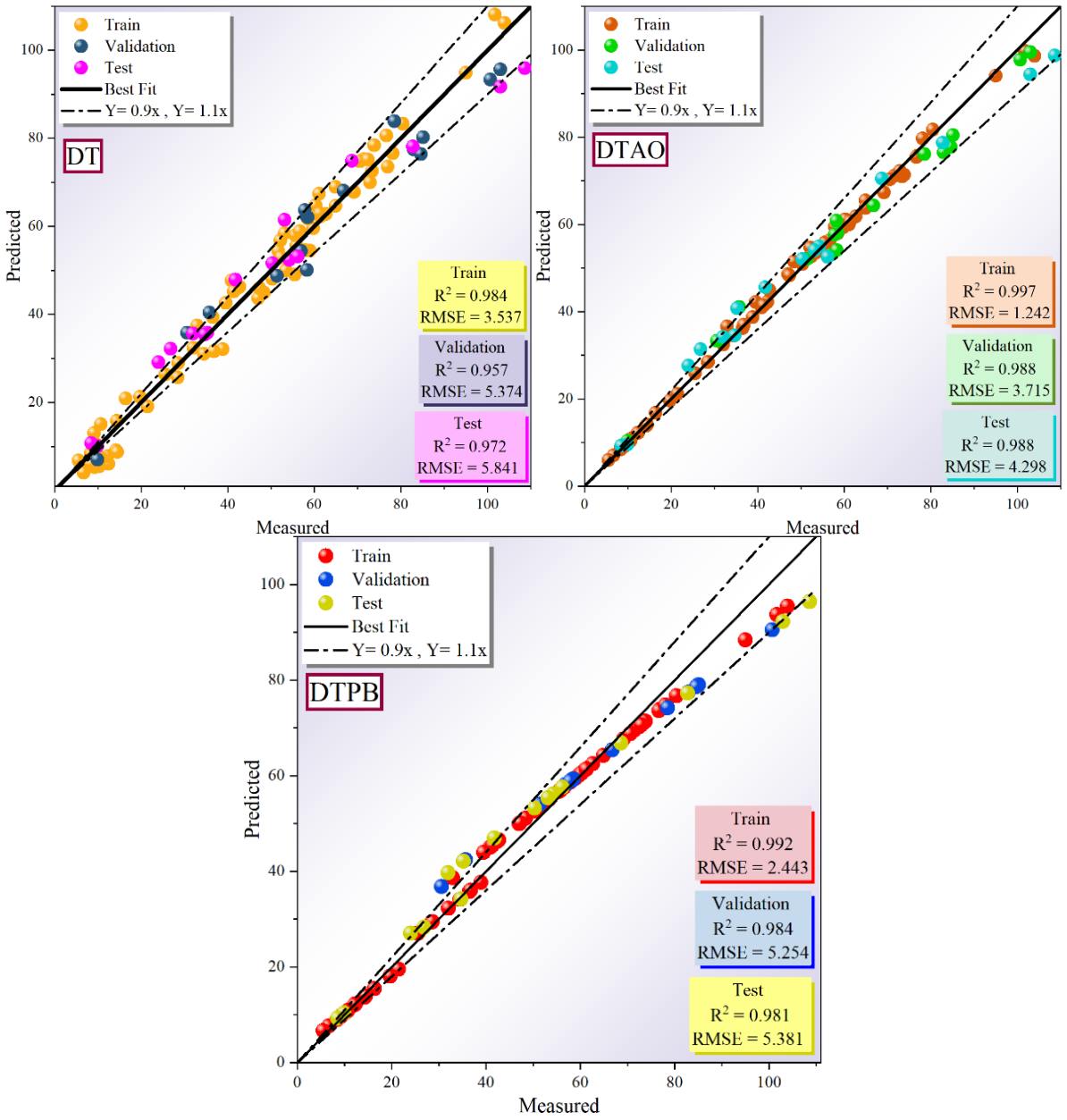

Fig. 5. The hybrid model's produced scatter plot.

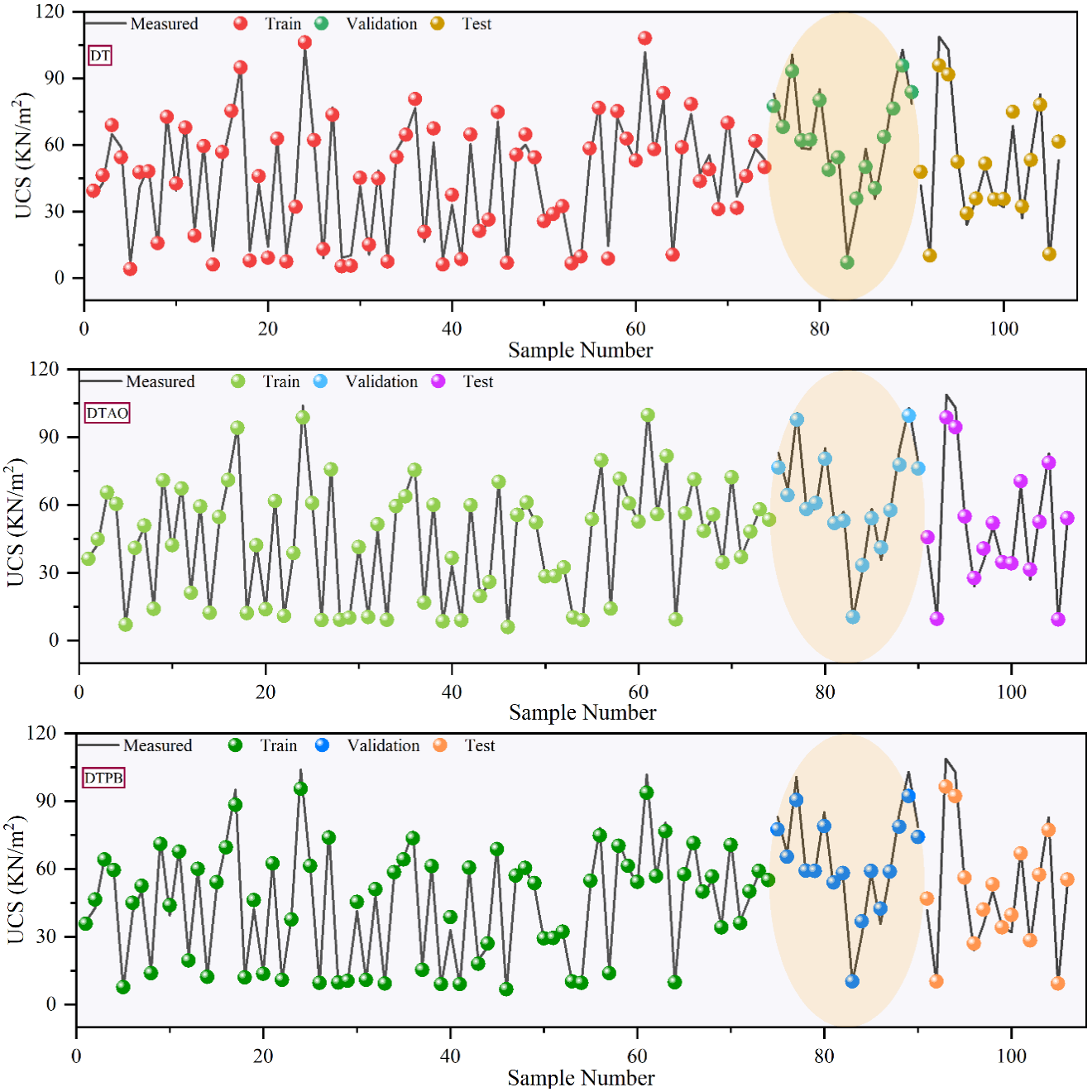

Fig. 6. Comparison between measured and expected values.

Fig. 6 depicts a thorough comparison between expected results and actual measurements, neatly broken down into stages of testing, validation, and training. The best state can be found using these expected outcomes as a guide. Examining the *DTAO* model's behavior reveals a slight discrepancy between measured values obtained during the training and testing phases, with the latter typically exhibiting relatively higher values. Similar to the *DTAO* model's projected points, the *DTPB* model's projected points also deviate slightly from the measurements taken, though not as significantly. The *DT* model, in contrast, reveals a more pronounced level of variance and exhibits comparatively diminished efficacy in comparison to the other two models.

Fig. 7 shows a line plot of the error % of the created models. *DTAO* has the lowest error rate, as seen by the graph, with the majority of values falling within the 17% range. More values above 43% and a wider range of error percentages are present for *DT* and *DTPB*. The *DT* and *DTPB* distributions are notably tilted to the right, suggesting that certain data points have notably larger error percentages. This illustrates both the improved accuracy of the *DTAO* and the way the graphs of the generated models' error percentage distributions are shown.

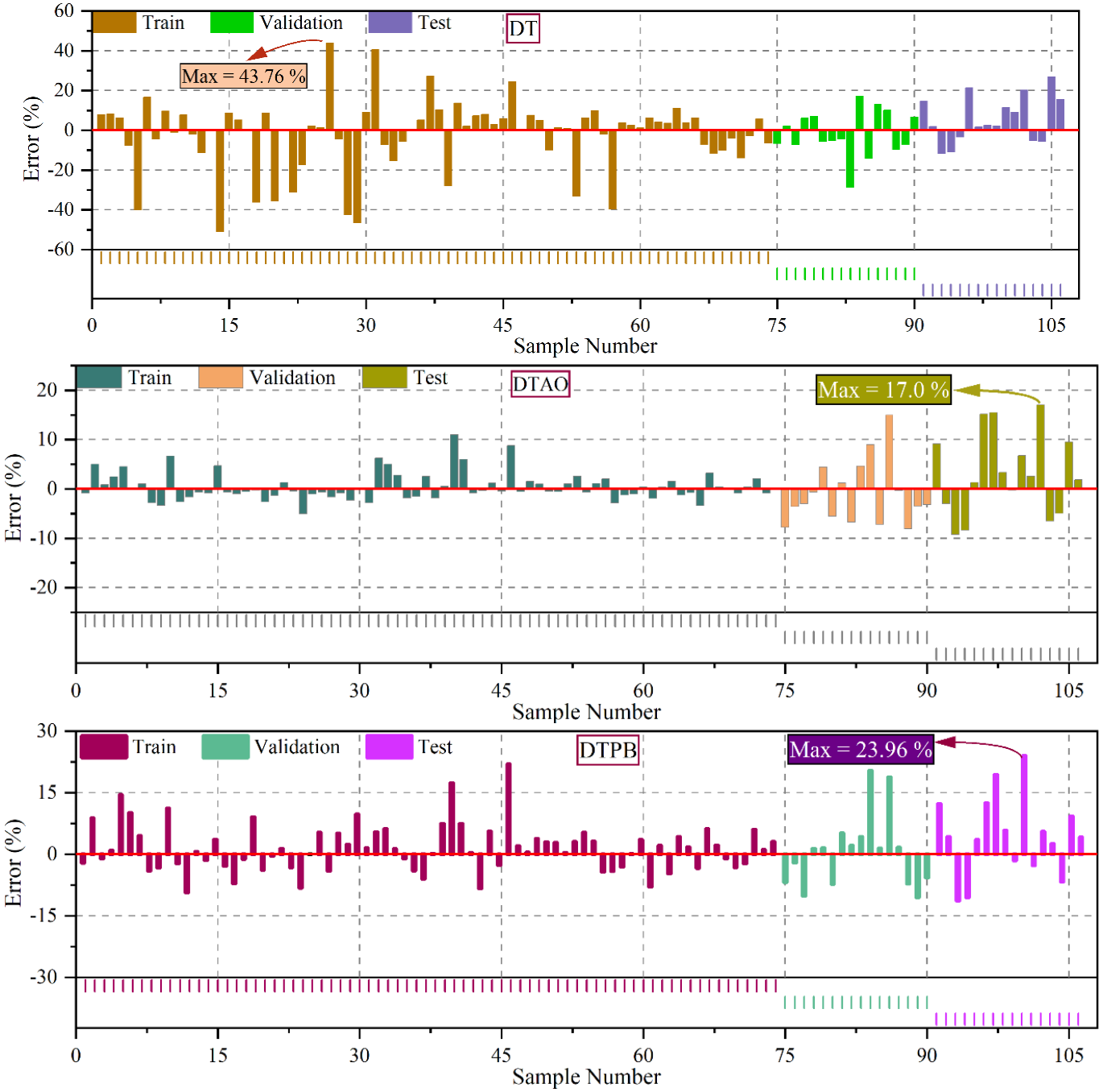

Fig. 7. The line plot determines the error rate percentage for the line models.

Fig. 8 displays an interval map that illustrates the error percentages of the models that were employed in this study. Throughout the training phase, the *DTAO* model demonstrated remarkable performance, maintaining errors below the 20% threshold with a consistent mean error rate of 0%. There was minimal dispersion in the data's normal distribution. The *DT* model, on the other hand, demonstrated dispersion in all phases and a very symmetrical and homogenous normal distribution, despite the error rate being below 20%. On the other hand, out of the three models, the DTPB model showed the most substantial and diversified inconsistencies. An uncommon event in statistical analysis occurred during the assessment step when one outlier data point accounted for more than 20% of the dataset. In contrast to the other two models, the DT model's Gaussian distribution showed better dispersion and a lower frequency of occurrences close to 0.

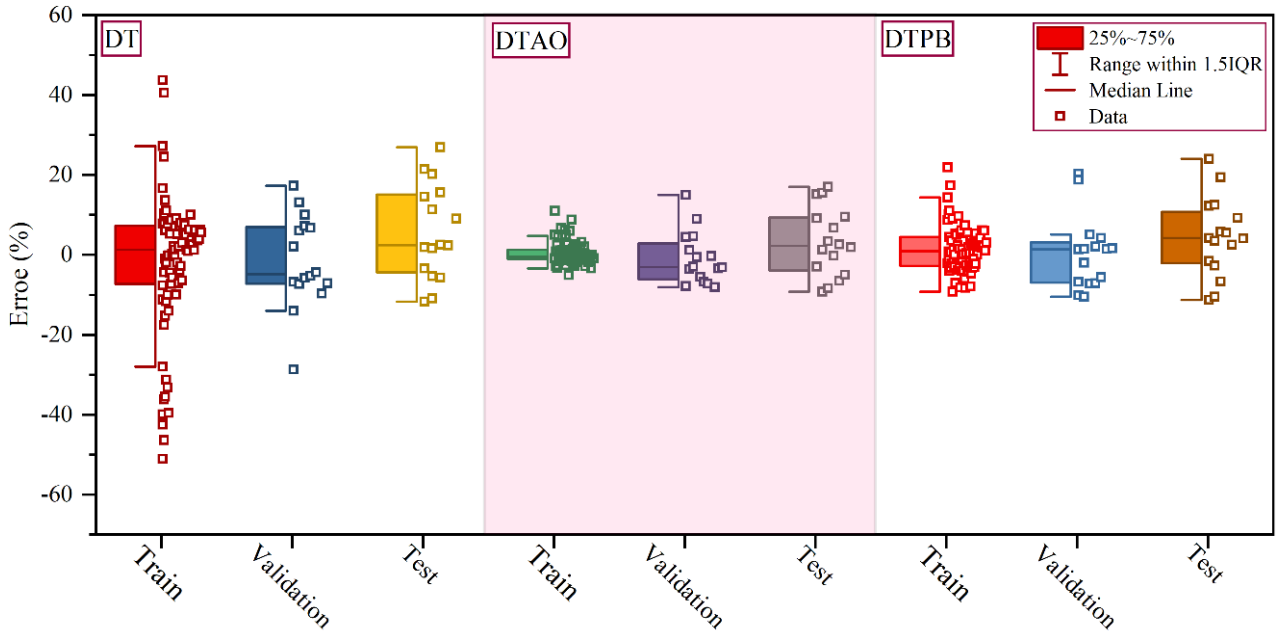

Fig. 8. The normal supply of errors between the intermission plot models.

IV. DISCUSSION

Table III provides a comparative summary of various published articles on the prediction of UCS using different datasets, variables, models, and evaluators, including R^2 and RMSE metrics. Narendra et al. [39] utilized a dataset of 186 samples with variables such as curing period, clay water-cement ratio, cement content, liquid limit, liquidity index, water content, pH, and Na+. They applied a Genetic Programming (GP) model, reaching an R^2 of 0.988 and an RMSE of 1.135. Ceryan et al. [40] worked with a smaller dataset of 56 samples, using variables like Clt, Cly, Fld, Qrz, Qq, Bi, n, ne, Id, Vp, and Vm. Their model of regression (REG) yielded an R^2 of 0.883 and an RMSE of 1.108. A dataset of 93 samples was employed by Majdi and Rezaei [41] and Rezaei et al. [42]. The variables included rock type, Schmidt hardness, density, and porosity. While Rezaei et al. employed a Mamdani fuzzy model and produced an R^2 of 0.943 and an RMSE of 3.2, Majdi and Rezaei utilized an ANN model and obtained an R^2 of 0.972 and an RMSE of 1.113. Mohamad et al. [43] analyzed 160 samples with variables such as rock type, weathering grade, BD, BTS, Is(50), and Vp. Their ANN-PSO model demonstrated high accuracy with an R^2 of 0.982 and an RMSE of 0.077. The present study involved 106 samples and variables including BD, BTS, DD, Vp, SRn, and Is. Using the Decision Tree-Arithmetic Optimizer (DTAO) model, this study achieved a striking R^2 of 0.988 and an RMSE of 1.242, demonstrating comparable accuracy to the best-performing models reviewed.

1) Limitation: While this study demonstrates significant advancements in predicting UCS for soil-stabilizer combinations using ML techniques, several limitations must be acknowledged. Firstly, the dataset size, though reasonably substantial, may still limit the generalizability of the models. A larger and more diverse dataset encompassing a wider range of soil types and conditions would enhance the robustness and applicability of the models. Secondly, the models developed are highly specific to the input variables used in this study. Variables such as particle size distribution, plasticity, and stabilizer type are crucial, but other potentially influential factors like temperature, humidity, and long-term aging effects were not considered. Including these factors could further improve model accuracy and reliability. Thirdly, the study's reliance on historical data means that any inaccuracies or biases in the original data could propagate through the models, affecting their predictions. Rigorous data validation and cleaning procedures are essential to mitigate this risk. Additionally, the integration of meta-heuristic algorithms like the PVS and the AOA, while enhancing model precision, introduces complexity. This complexity may pose challenges for practical implementation and computational efficiency, particularly for large-scale projects. Lastly, the models, though validated against historical data, require further testing in real-world scenarios to confirm their practical effectiveness and reliability in diverse geotechnical applications.

2) Future study: A potential future study stemming from this research could delve into the application of advanced ML techniques beyond DTs for predicting UCS in soil-stabilizer combinations. One avenue could involve exploring ensemble methods such as Random Forests, Gradient Boosting Machines, or Neural Networks to compare their predictive performance with the DT-based models developed in this study. This comparative analysis would provide a more comprehensive understanding of which ML algorithms are most effective for this specific prediction task. Moreover, considering the integration of meta-heuristic algorithms like the PVS and the AOA, a future study could focus on optimizing the parameters and configurations of these meta-heuristic algorithms. This optimization process could enhance the precision and

efficiency of the predictive models, leading to even more accurate UCS forecasts for various soil-stabilizer combinations. Additionally, extending the validation process to include real-world field data from ongoing construction projects or geological surveys could strengthen the practical applicability of the predictive models developed. This extension would involve collaborating with industry partners or governmental agencies to access and analyze relevant datasets, ensuring that the models are validated under diverse and realistic conditions.

TABLE III. THE SUMMARY OF PUBLISHED ARTICLES

Article	Num. of Dataset	Variables	Model	Evaluator	
				R^2	RMSE
Narendra et al. [39]	186	Curing period, clay water-cement ratio, cement content, liquid limit, liquidity index, water content, pH, Na^+	GP	0.9881	135
Ceryan et al. [40]	56	Clt, Cly, Fld, Qrz, Qq, Bi, n, ne, Id, Vp, and Vm	REG	0.8837	1.108
Majdi and Rezaei [41]	93	Rock type, Schmidt hardness, Density, and Porosity	ANN	0.9725	1.113
Rezaei et al. [42]	93	Rock type, Schmidt hardness, Density, and Porosity	Mamdani fuzzy	0.9437	3.2
Mohamad et al. [43]	160	Rock type, Weathering grade, BD, BTS, Is(50), and Vp	ANN-PSO	0.982	0.077
Present study	106	BD, BTS, DD, Vp, SRn, and Is	DTAO	0.988	1.242

V. CONCLUSION

By ML techniques, specifically DT algorithms, this study presents an innovative approach to predict UCS values with a high level of accuracy. This method provides an affordable alternative and drastically cuts down on the amount of time needed for UCS prediction. Using DT techniques, a unique ML model serves as the foundation for the main framework for UCS prediction. To enhance precision and reduce errors, 3 models were developed by combining the AOA and PVS meta-heuristic algorithms, namely DT, $DTAO$, and $DTPB$. These models were put through a rigorous validation process that used lab samples from publically accessible sources for the testing, validation, and training phases. In order to thoroughly assess model performance, various metrics such as $RMSE$, MSE, R^2, $NRMSE$, and $MARE$ were employed. These metrics collectively provide a deep insight into the model's ability to predict UCS accurately and its overall effectiveness in estimation. This research significantly advances the field of soil mechanics by improving understanding of the factors influencing UCS through the application of ML techniques. Consequently, it opens up opportunities for more precise and dependable UCS predictions in various engineering applications. In this investigation, it was demonstrated that the $DTAO$ models achieved the highest R^2 values, while the DT model exhibited the lowest R^2 value, albeit with a marginal difference of only 2.1%. Furthermore, the error indicators revealed that the $DTAO$ models outperformed the DT and $DTPB$ models by demonstrating lower error values. Notably, the $DTAO$ models

Consistently exhibited the last $RMSE$ values across all phases, with a significant difference of 96% and 65% when compared to the DT and $DTPB$ models, respectively. This highlights the exceptional accuracy and reliability of the $DTAO$ models in predicting UCS.

REFERENCES

[1] M. Nguyen Duc, A. Ho Sy, T. Nguyen Ngoc, and T. L. Hoang Thi, "An Artificial Intelligence Approach Based on Multi-layer Perceptron Neural Network and Random Forest for Predicting Maximum Dry Density and Optimum Moisture Content of Soil Material in Quang Ninh Province, Vietnam," in CIGOS 2021, Emerging Technologies and Applications for Green Infrastructure: Proceedings of the 6th International Conference on Geotechnics, Civil Engineering and Structures, Springer, 2022, pp. 1745–1754.

[2] H. A. Shah et al., "Application of Machine Learning Techniques for Predicting Compressive, Splitting Tensile, and Flexural Strengths of Concrete with Metakaolin," Materials, vol. 15, no. 15, p. 5435, Aug. 2022, doi: 10.3390/ma15155435.

[3] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng, "Machine learning basics," Deep learning, pp. 98–164, 2016.

[4] M. F. Randolph and C. P. Wroth, "Analysis of deformation of vertically loaded piles," Journal of the geotechnical engineering division, vol. 104, no. 12, pp. 1465–1488, 1978.

[5] B. T. Pham, "A novel classifier based on composite hyper-cubes on iterated random projections for assessment of landslide susceptibility," Journal of the Geological Society of India, vol. 91, no. 3, pp. 355–362, 2018.

[6] A. Rassoul and K. Mojtaba, "Predicting maximum dry density, optimum moisture content and California bearing ratio (CBR) through soil index using ordinary least squares (OLS) and artificial neural networks (ANNS)," International Journal of Innovative Technology and Exploring Engineering, vol. 5, no. 3, pp. 1–5, 2015.

[7] S.-S. Park, "Unconfined compressive strength and ductility of fiber-reinforced cemented sand," Constr Build Mater, vol. 25, no. 2, pp. 1134–1138, 2011.

[8] S. K. Das, P. Samui, and A. K. Sabat, "Application of artificial intelligence to maximum dry density and unconfined compressive strength of cement stabilized soil," Geotechnical and Geological Engineering, vol. 29, pp. 329–342, 2011.

[9] A. Hossein Alavi, A. Hossein Gandomi, A. Mollahassani, A. Akbar Heshmati, and A. Rashed, "Modeling of maximum dry density and optimum moisture content of stabilized soil using artificial neural networks," Journal of Plant Nutrition and Soil Science, vol. 173, no. 3, pp. 368–379, 2010.

[10] R. M. Ruffolo and A. Shakoor, "Variability of unconfined compressive strength in relation to a number of test samples," Eng Geol, vol. 108, no. 1–2, pp. 16–23, 2009.

[11] S. Sathyapriya, P. D. Arumairaj, and D. Ranjini, "Prediction of unconfined compressive strength of a stabilized expansive clay soil using ANN and regression analysis (SPSS)," Asian J Res Soc Sci Humanit, vol. 7, no. 2, pp. 109–123, 2017.

[12] C. KS, Y. M. Chew, M. H. Osman, and M. G. SK, "Estimating maximum dry density and optimum moisture content of compacted soils," in International Conference on Advances in Civil and Environmental Engineering, 2015, pp. 1–8.

[13] C. M. O. Nwaiwu and E. O. Mezie, "Prediction of maximum dry unit weight and optimum moisture content for coarse-grained lateritic soils," Soils and Rocks, vol. 44, p. e2021054120, 2021.

[14] S. A. Naeini, B. Naderinia, and E. Izadi, "Unconfined compressive strength of clayey soils stabilized with waterborne polymer," KSCE Journal of Civil Engineering, vol. 16, pp. 943–949, 2012.

[15] M. Ghazavi and M. Roustaie, "The influence of freeze-thaw cycles on the unconfined compressive strength of fiber-reinforced clay," Cold Reg Sci Technol, vol. 61, no. 2–3, pp. 125–131, 2010.

[16] O. Sivrikaya, E. Togrol, and M. Komur, "Determination of unconfined compressive strength by Artificial Neural Network," in 10th National Congress of Soil Mechanics and Foundation Engineering, Istanbul, Turkey, 2004.

[17] M. A. Grima and R. Babuška, "Fuzzy model for the prediction of unconfined compressive strength of rock samples," International Journal of Rock Mechanics and Mining Sciences, vol. 36, no. 3, pp. 339–349, 1999.

[18] B. S. Narendra, P. V Sivapullaiah, S. Suresh, and S. N. Omkar, "Prediction of unconfined compressive strength of soft grounds using computational intelligence techniques: A comparative study," Comput Geotech, vol. 33, no. 3, pp. 196–208, 2006.

[19] Behnam Sedaghat, G. G. Tejani, and S. Kumar, "Predict the Maximum Dry Density of Soil based on Individual and Hybrid Methods of Machine Learning," Advances in Engineering and Intelligence Systems, vol. 002, no. 03, 2023, doi: 10.22034/aeis.2023.414188.1129.

[20] F. Meulenkamp and M. A. Grima, "Application of neural networks for the prediction of the unconfined compressive strength (UCS) from Equotip hardness," International Journal of rock mechanics and mining sciences, vol. 36, no. 1, pp. 29–39, 1999.

[21] H. Sonmez, E. Tuncay, and C. Gokceoglu, "Models to predict the uniaxial compressive strength and the modulus of elasticity for Ankara Agglomerate," International Journal of Rock Mechanics and Mining Sciences, vol. 41, no. 5, pp. 717–729, 2004.

[22] C. Gokceoglu and K. Zorlu, "A fuzzy model to predict the uniaxial compressive strength and the modulus of elasticity of a problematic rock," Eng Appl Artif Intell, vol. 17, no. 1, pp. 61–72, 2004.

[23] S. Dehghan, G. SATTARI, and M. A. ALIABADI, "Prediction of uniaxial compressive strength and modulus of elasticity for Travertine samples using regression and artificial neural networks," Mining Science and Technology (China), vol. 20, pp. 41–46, Jan. 2010, doi: 10.1016/S1674-5264(09)60158-7.

[24] D. A. Mishra and A. Basu, "Estimation of uniaxial compressive strength of rock materials by index tests using regression analysis and fuzzy inference system," Eng Geol, vol. 160, pp. 54–68, 2013.

[25] A. ASTM, "Standard test method of unconfined compressive strength of intact rock core specimens," ASTM Publication, 1986.

[26] Z. T. Bieniawski, "Estimating the strength of rock materials," J South Afr Inst Min Metall, vol. 74, no. 8, pp. 312–320, 1974.

[27] C. Gokceoglu and K. Zorlu, "A fuzzy model to predict the uniaxial compressive strength and the modulus of elasticity of a problematic rock," Eng Appl Artif Intell, vol. 17, no. 1, pp. 61–72, 2004.

[28] F. Meulenkamp, "Improving the prediction of the UCS, by EQUOTIP readings using statistical and neural network models," Memoirs of the Centre for Engineering Geology in the Netherlands, vol. 162, no. 127, pp. 85–101, 1997.

[29] E. Momeni, D. J. Armaghani, M. Hajihassani, and M. F. M. Amin, "Prediction of uniaxial compressive strength of rock samples using hybrid particle swarm optimization-based artificial neural networks," Measurement, vol. 60, pp. 50–63, 2015.

[30] H. I. Erdal, "Two-level and hybrid ensembles of decision trees for high performance concrete compressive strength prediction," Eng Appl Artif Intell, vol. 26, no. 7, pp. 1689–1697, 2013.

[31] A. Ahmad et al., "Prediction of compressive strength of fly ash based concrete using individual and ensemble algorithm," Materials, vol. 14, no. 4, p. 794, 2021.

[32] A. Karbassi, B. Mohebi, S. Rezaee, and P. Lestuzzi, "Damage prediction for regular reinforced concrete buildings using the decision tree algorithm," Comput Struct, vol. 130, pp. 46–56, 2014.

[33] R. Zhou, Y. Tang, H. Li, and Z. Liu, "Predicting the compressive strength of ultra-high-performance concrete using a decision tree machine learning model enhanced by the integration of two optimization meta-heuristic algorithms," Journal of Engineering and Applied Science, vol. 71, no. 1, p. 43, 2024, doi: 10.1186/s44147-023-00350-1.

[34] S. B. Kotsiantis, "Decision trees: a recent overview," Artif Intell Rev, vol. 39, pp. 261–283, 2013.

[35] B. Doğan and T. Ölmez, "A new metaheuristic for numerical function optimization: Vortex Search algorithm," Inf Sci (N Y), vol. 293, pp. 125–145, 2015.

[36] T. Sağ, "PVS: a new population-based vortex search algorithm with boosted exploration capability using polynomial mutation," Neural Comput Appl, vol. 34, no. 20, pp. 18211–18287, 2022.

[37] B. Doğan and T. Ölmez, "Fuzzy clustering of ECG beats using a new metaheuristic approach," in 2nd International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO), 2014, pp. 7–9.

[38] M. Castelli, L. Vanneschi, and S. Silva, "Prediction of high-performance concrete strength using genetic programming with geometric semantic genetic operators," Expert Syst Appl, vol. 40, no. 17, pp. 6856–6862, 2013.

[39] B. S. Narendra, P. V Sivapullaiah, S. Suresh, and S. N. Omkar, "Prediction of unconfined compressive strength of soft grounds using computational intelligence techniques: A comparative study," Comput Geotech, vol. 33, no. 3, pp. 196–208, 2006.

[40] N. Ceryan, U. Okkan, and A. Kesimal, "Prediction of unconfined compressive strength of carbonate rocks using artificial neural networks," Environ Earth Sci, vol. 68, pp. 807–819, 2013.

[41] A. Majdi and M. Rezaei, "Prediction of unconfined compressive strength of rock surrounding a roadway using artificial neural network," Neural Comput Appl, vol. 23, pp. 381–389, 2013.

[42] M. Rezaei, A. Majdi, and M. Monjezi, "An intelligent approach to predict the unconfined compressive strength of rock surrounding access tunnels in longwall coal mining," Neural Comput Appl, vol. 24, pp. 233–241, 2014.

[43] E. T. Mohamad, D. Jahed Armaghani, E. Momeni, and S. V. Alavi Nezhad Khalil Abad, "Prediction of the unconfined compressive strength of soft rocks: a PSO-based ANN approach," Bulletin of Engineering Geology and the Environment, vol. 74, pp. 745–757, 2015.