Research Article

Application of Gradient Boosting Machine Learning Algorithms to Predict Uniaxial Compressive Strength of Soft Sedimentary Rocks at Thar Coalfield

Niaz Muhammad Shahani,1,2 Muhammad Kamran,3 Xigui Zheng,1,2,4,5 Cancan Liu,1,2 and Xiaowei Guo1,2

1School of Mines, China University of Mining and Technology, Xuzhou 221116, Jiangsu Province, China
2Key Laboratory of Deep Coal Resources Mining, Ministry of Education of China, School of Mines, China University of Mining & Technology, Xuzhou 221116, China
3Bandung Institute of Technology, Indonesia
4School of Mines and Civil Engineering, Liupanshui Normal University, Liupanshui 553004, China
5Guizhou Guineng Investment Co., Ltd., Liupanshui 553600, China

Correspondence should be addressed to Xigui Zheng; fs16020006@cumt.edu.cn

Received 18 August 2021; Revised 6 October 2021; Accepted 22 October 2021; Published 1 November 2021

Academic Editor: Zhoujing Ye

Copyright © 2021 Niaz Muhammad Shahani et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The uniaxial compressive strength (UCS) of rock is one of the essential data in engineering planning and design. Correctly testing UCS of rock to ensure its accuracy and authenticity is a prerequisite for ensuring the design of any rock engineering project. UCS of rock has a broad range of applications in mining, geotechnical, petroleum, geomembranes, and other fields of engineering. The application of the gradient boosting machine learning algorithms has been rarely used, especially for UCS prediction, and has performed well, based on the relevant literature of the study. In this study, four gradient boosting machine learning algorithms, namely, gradient boosted regression (GBR), Catboost, light gradient boosting machine (LightGBM), and extreme gradient boosting (XGBoost), were developed to predict the UCS in MPa of soft sedimentary rocks of the Block-IX at Thar Coalfield, Pakistan, using four input variables such as wet density ($\rho_w$) in g/cm$^3$; moisture in %; dry density ($\rho_d$) in g/cm$^3$; and Brazilian tensile strength (BTS) in MPa. Then, 106-point dataset was allocated identically for each algorithm into 70% for the training phase and 30% for the testing phase. According to the results, the XGBoost algorithm outperformed the GBR, Catboost, and LightGBM with coefficient of correlation ($R^2$) = 0.99, mean absolute error (MAE) = 0.00062, mean square error (MSE) = 0.0000006, and root mean square error (RMSE = 0.00079 in the training phase and $R^2$ = 0.99, MAE = 0.00054, MSE = 0.0000005, and RMSE = 0.00069 in the testing phase. The sensitivity analysis showed that BTS and $\rho_w$ are positively correlated, and the moisture and $\rho_d$ are negatively correlated with the UCS. Therefore, in this study, the XGBoost algorithm was shown to be the most accurate algorithm among all the investigated four algorithms for UCS prediction of soft sedimentary rocks of the Block-IX at Thar Coalfield, Pakistan.

1. Introduction

The uniaxial compressive strength (UCS) of rock is one of the important data in engineering planning and design. Correctly testing UCS of rock to ensure its accuracy and authenticity is a prerequisite for ensuring the design of any rock engineering project. UCS of rock has a wide range of applications in mining, geotechnical, petroleum, geomembranes, and other fields of engineering. The study of rock mechanical properties is the basis for innovative advances associated with energy supply. The significance of rock mechanics is acknowledged in the advancement of natural assets, for example, the protection of energy sources (petroleum products: oil, coal, and natural gas), as well as the
protection of the surrounding rock environment. In addition, protective waste removal and hydroelectric energy assets require further examination of rocks and soils, which are dependent on the mechanical properties of rocks [1, 2]. Unconfined UCS of rocks plays a significant role in tackling issues associated with rock mechanics and the design of geostuctures [3–6]. Various researchers have proposed empirical equations in order to estimate the UCS of different rock mass databases as tabulated in Table 1.

Some researchers have employed indirect testing methods, i.e., multiple linear regression (MLR) [17, 22, 23], artificial neural network (ANN) [23, 24], adaptive neuro-fuzzy inference system (ANFIS) [25], and other machine learning algorithms to estimate the accuracy and reliability of rock data [26–28], rather than using direct tests recommended by international standards, which are considered time-consuming, expensive, and unreliable [29, 30]. While these frameworks are appropriate, fast, and favorable techniques to tackle difficult problems, in most cases, they are simply capable of understanding the complex interactions among variables to estimate an objective and do not provide any intuition about the interrelationships among predictors and a return value [31]. Abidi et al. predicted the UCS of intact rocks with 196 data points, i.e., sandstone, marl, limestone, and conglomerate, using the intelligent ANN method and MLR. According to the results of their study, the ANN model performed better than the MLR model in terms of correlation coefficient ($R^2$), mean absolute error (MAE), and root mean square error (RMSE) [23]. Abidi et al. employed both ANN and ANFIS models to predict UCS of 136 sandstone rock samples because laboratory testing methods are time-consuming and complex. Based on the results of several performance metrics, i.e., $R^2$, RMSE, and variance accounted for (VAF), the ANFIS model showed the best performance over ANN [32]. Ceryan et al. adopted soft computing methods, i.e., extreme learning machine (ELM) and minimax probability machine regression (MPMR), to predict the UCS of volcanic rocks. In addition, the least-square support vector machine (LS-SVM) model was incorporated to compare its performance with the proposed soft computing models. Thus, ELM and MPMR outperformed the LS-SVM [33]. Ceryan et al. developed the fuzzy interface system (FIS), ANN, and LV-SVM to predict the UCS of rocks. According to their results, LV-SVM performed better than the other developed models [34]. Neurogenetic and multiple regression (MR) methods were developed by Monjezi et al. to estimate the UCS of rocks. The neurogenetic approach was best applied compared to MR [35]. Aboutaleb et al. predicted the UCS of carbonate rocks using nondestructive tests, namely, simple regression analysis (SRA), multiple regression analysis (MRA), ANN, and support vector regression (SVR). As reported by the SVR model, it performed best among all the models studied [36]. Ghasemi et al. developed a soft computing (model tree) approach to predict the UCS of carbonate rocks, which proved to have the best performance [37]. Barzegar et al. used ensemble tree-based machine learning approaches such as “random forest (RF), M5 model tree, and multivariate adaptive regression splines (Mars)” for predicting UCS of travertine rocks. In parallel, an ensemble committee-based ANN model was developed to correlate the returns of proposed models to determine the predicted UCS in accuracy. According to the results, Mars outperformed all studied models [38]. Matin et al. used RF for variable selection and UCS prediction, and the proposed RF for UCS prediction was satisfactory [39]. Zhong et al. predicted rock mechanical property, i.e., UCS with an established XGBoost model of machine learning, which proved to have the best performance in results [40].

According to the above literature, the application of the gradient boosting machine learning algorithms has been rarely used for UCS prediction. In this work, we proposed an innovative adaptation of four gradient boosting machine learning algorithms such as gradient boosted regression (GBR), Catboost, light gradient boosting machine (LightGBM), and extreme gradient boosting (XGBoost) that improves the handling of concept drift. The database employed in this work has been collected from soft sedimentary rocks of the Block-IX at Thar Coalfield, Pakistan. Previously, researchers considered wet density ($\rho_w$) [41]; moisture [41]; dry density ($\rho_d$) [42]; and Brazilian tensile strength (BTS) [43] as input variables for evaluating the UCS. Based on the literature, the four significant input parameters $\rho_w$ in g/cm$^3$; moisture in %; dry $\rho_d$ in g/cm$^3$; and BTS in MPa are used for each developed algorithm. Moreover, 106-point dataset was allocated identically for each algorithm into 70% for the training phase and 30% for the testing phase. Figure 1 illustrates the flowchart of this study.

2. Overview of the Study Area and Dataset

Thar Coalfield is the 7th world’s biggest coal mine in the Sindh Province of Pakistan [44]. Thar Coal is demarcated to be 175.5 billion tons of lignite or brown coal which can be utilized for the fuel and generation of electricity purpose. As shown in Figure 2, the Thar Coalfield is divided into 12 different blocks. Open-pit and underground mining methods can be used to extract coal from the area. Specifically, Block-II of the 12 blocks is fully developed for the open-pit mining method by Sindh Engro Coal Mining Company (SECMC), while some areas of Block-1 and Block-IX are being prepared for the underground mining method. The coal seam thickness of Block-IX at Thar Coalfield is about 12 m with an inclination of 0° to 7°, and the top-bottom plate is siltstone-claystone to claystone. However, for the first time in the history of Pakistan, Shahani et al. recommended the use of mechanized longwall top coal caving (LTCC) method in the Thar Coalfield, Pakistan [44,45]. At the same time, accurate determination of the mechanical properties of the Thar Coalfield, especially the UCS, plays an important role in providing a good understanding of the behavior of the roof and ground prior to mining operations. In this study, 106 samples of soft sedimentary rocks were initially collected randomly from Block-IX at Thar, as shown in Figure 2 of the yellow-colored area. The collected rock samples were then arranged and subdivided according to the recommended standards of the...
International Society of Rock Mechanics (ISRM) [46] and the American Society for Testing Methods (ASTM) [47] in order to maintain the same core size and geological and geometric characteristics. The experimental work on the studied rock samples was carried out in the laboratory of the Department of Mining Engineering, Mehran University of Engineering and Technology (MUET), to determine the physical and mechanical properties like $\rho_w$ (g/cm$^3$); moisture (%) $\rho_d$ (g/cm$^3$); BTS (MPa); and UCS (MPa). In Figure 3(a), a universal testing machine (UTM) was used to perform the UCS as shown in Figure 3(b) and a BTS as shown in Figure 3(c). Table 2 illustrates the entire dataset employed in the study. Table 3 denotes the statistical distribution of the UCS dataset.

**Table 1: Empirical equations proposed by the researchers for estimating UCS.**

<table>
<thead>
<tr>
<th>S. no</th>
<th>Country of origin</th>
<th>Rock type</th>
<th>No. of data</th>
<th>$R^2$</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Turkey</td>
<td>Sedimentary</td>
<td>18</td>
<td>0.82</td>
<td>[7]</td>
</tr>
<tr>
<td>2</td>
<td>USA</td>
<td>Mixed</td>
<td>86</td>
<td>0.81</td>
<td>[8]</td>
</tr>
<tr>
<td>3</td>
<td>Spain</td>
<td>Mixed</td>
<td>194</td>
<td>0.81</td>
<td>[9]</td>
</tr>
<tr>
<td>4</td>
<td>Japan and Indonesia</td>
<td>Mixed</td>
<td>33</td>
<td>0.77</td>
<td>[10]</td>
</tr>
<tr>
<td>5</td>
<td>Turkey</td>
<td>Sedimentary</td>
<td>150</td>
<td>0.65</td>
<td>[12]</td>
</tr>
<tr>
<td>6</td>
<td>India</td>
<td>Sedimentary</td>
<td>13</td>
<td>0.87</td>
<td>[13]</td>
</tr>
<tr>
<td>7</td>
<td>Iran</td>
<td>Sedimentary</td>
<td>40</td>
<td>0.94</td>
<td>[14]</td>
</tr>
<tr>
<td>8</td>
<td>Israel and USA</td>
<td>Mixed</td>
<td>7</td>
<td>0.96</td>
<td>[15]</td>
</tr>
<tr>
<td>9</td>
<td>UK, France, and Denmark</td>
<td>Sedimentary</td>
<td>7</td>
<td>0.9</td>
<td>[16]</td>
</tr>
<tr>
<td>10</td>
<td>England and Turkey</td>
<td>Mixed</td>
<td>44</td>
<td>0.11</td>
<td>[17]</td>
</tr>
<tr>
<td>11</td>
<td>Greece</td>
<td>Metamorphic</td>
<td>32</td>
<td>0.82</td>
<td>[18]</td>
</tr>
<tr>
<td>12</td>
<td>Spain</td>
<td>Sedimentary</td>
<td>99</td>
<td>0.94</td>
<td>[19]</td>
</tr>
<tr>
<td>13</td>
<td>Turkey</td>
<td>Sedimentary</td>
<td>19</td>
<td>0.78</td>
<td>[20]</td>
</tr>
<tr>
<td>14</td>
<td>Portugal</td>
<td>Igneous</td>
<td>9</td>
<td>0.72</td>
<td>[21]</td>
</tr>
<tr>
<td>15</td>
<td>UK, France, and Denmark</td>
<td>Sedimentary</td>
<td>7</td>
<td>0.87</td>
<td>[16]</td>
</tr>
<tr>
<td>16</td>
<td>United Kingdom</td>
<td>Igneous</td>
<td>171</td>
<td>0.53</td>
<td>[21]</td>
</tr>
</tbody>
</table>

**Figure 1: Flowchart of the study.**

Future forecasting
Recognize the best framework
Evaluation of the frameworks
Data Splitting
Training Dataset
Testing Dataset
UCS Dataset
GBR, Catboost, LightGBM, XGboost

Acknowledgments
This research was supported by the National Natural Science Foundation of China (Grant No. 51804210) and the Natural Science Foundation of Hunan Province (Grant No. 2021JJ50077).
12 different blocks of the Thar Coalfield

**Figure 2**: Location map of study area (Block-IX) of Thar Coalfield, Pakistan.

**Figure 3**: (a) Universal testing machine (UTM), (b) deformed rock core specimen for Brazilian tensile strength test, and (c) deformed rock core specimen for UCS test.

**Table 2**: Brief summary of UCS dataset.

<table>
<thead>
<tr>
<th>Serial no.</th>
<th>$\rho_w$ (g/cm$^3$)</th>
<th>Moisture (%)</th>
<th>$\rho_d$ (g/cm$^3$)</th>
<th>BTS (MPa)</th>
<th>UCS (MPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.132</td>
<td>11.78747</td>
<td>1.90719</td>
<td>0.305</td>
<td>0.404</td>
</tr>
<tr>
<td>2</td>
<td>2.013</td>
<td>15.07577</td>
<td>1.749282</td>
<td>0.217402</td>
<td>0.491</td>
</tr>
<tr>
<td>3</td>
<td>2.112</td>
<td>14.08254</td>
<td>1.851291</td>
<td>0.259</td>
<td>0.579</td>
</tr>
<tr>
<td>4</td>
<td>2.119</td>
<td>17.0949</td>
<td>1.809643</td>
<td>0.394188</td>
<td>0.485</td>
</tr>
<tr>
<td>5</td>
<td>2.034</td>
<td>15.97925</td>
<td>1.759547</td>
<td>0.336</td>
<td>0.52</td>
</tr>
<tr>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
<td>. . .</td>
</tr>
<tr>
<td>102</td>
<td>2.003</td>
<td>20.67989</td>
<td>1.659763</td>
<td>0.502</td>
<td>3.35</td>
</tr>
<tr>
<td>103</td>
<td>1.823</td>
<td>14.17769</td>
<td>1.596634</td>
<td>0.452715</td>
<td>2.46</td>
</tr>
<tr>
<td>104</td>
<td>1.964</td>
<td>20.07648</td>
<td>1.635624</td>
<td>0.358732</td>
<td>3.448</td>
</tr>
<tr>
<td>105</td>
<td>1.993</td>
<td>16.44345</td>
<td>1.71156</td>
<td>0.516271</td>
<td>3.549</td>
</tr>
<tr>
<td>106</td>
<td>1.97</td>
<td>12.14405</td>
<td>1.756669</td>
<td>0.627239</td>
<td>3.28</td>
</tr>
</tbody>
</table>
Figure 4 represents the pairwise distribution of various features and UCS. It can be seen that moisture and BTS are moderately correlated to the UCS whereas \( \rho_m \) and \( \rho_d \) are negatively correlated to the UCS. It is important to mention that each feature does not correlate well with UCS independently; hence all the features together are evaluated to predict the UCS.

3. Methods

3.1. Gradient Boosted Regressor (GBR). The gradient boosted regression (GBR) tree incorporates the weak learners; i.e., the learner algorithms moderately perform well as compared to random algorithms, into a strong learner in a repetitive technique [48]. In contrast to the bagging technique, the boosted mechanism produces the base frameworks consecutively. The robustness of the prediction model is enhanced by generating several frameworks in a series by focusing priority on this learning information that is implicated to evaluate. In the boosting mechanism, the previous base frameworks that are inappropriate to estimate are frequently developed in the training datasets compared to those models that are precisely evaluated. Each supplementary base framed is directed to rectify the errors produced by its preceding base framework. The existence of the boosting mechanism is from the response of Schapire to Keran’s inquiry [49, 50] (Kearns): Is a combination of a weak learner an alternative to distinguish strong learner? Weak Learner is defined as the algorithm that is working well as compared to random approximation; a strong base framework is a more authentic classification or regression algorithm that is inconsistent that is efficiently corresponding with the problem. The response to such an inquiry is very significant. The evaluation of a weak framework is often unchallenging as compared with a strong framework. Schapire demonstrated that the response to Kearns’ inquiry is “Yes” manifesting that many weak frameworks are combined into a high and single robust framework.

The main difference between boosted and bagging mechanism is that in boosting method the training datasets are systematically resampled in order to anticipate the most convenient instruction for each succeeding framework. The modified dissemination in each stage of training is contingent on the error generated by the preceding framework. In contrast, in the bagging mechanism, each trail is consistently designated to generate a training dataset, and the uncertainty of designating a single trail is divergent for the boosting mechanism. The trails that are incorrectly evaluated or misclassified have more probability to be set with higher weights. Hence, each freshly developed framework emphasizes the trail that has been incorrectly evaluated or misclassified by succeeding frameworks.

Boosting arranges the secondary frameworks that reduce a certain loss function averaged over the learning datasets, i.e., mean absolute error or squared mean error. The loss function calculates the aggregate of the forecasted value that varies from the measured value. A forward stage-wise modeling technique is one of the estimated solutions to the problem. This modeling technique consecutively adjoins new base frameworks without replacing the coefficient and parameter of the model that has previously been attached. With reference to the regression models, the boosting mechanism is a configuration of “functional gradient descent.” Functional gradient descent is an optimization mechanism that reduces a certain loss function by attaching a base framework at each stage that minimizes well the loss function. Figure 5 shows the description of GBR employed in this study [51, 52].

Friedman suggested an improvement to the gradient boosted regression models by employing a predetermined amount of regression tree to the base framework. The modified framework enhances the performance of the Friedman model [53]. In order to predict the UCS of sedimentary rocks, the modified version of gradient boosted regression has been used. Table 4 illustrates the execution of a GBR framework in python. Consider that the leaves number for each tree is \( l \), each tree splitting the input space into \( l \) separate territory \( T_{lp} \), \( T_{2lp} \), \ldots \, \ldots \, \ldots \, T_{lp} \) and forecasting a constant value \( k_{lp} \) for territory \( T_{lp} \). The gradient boosting regression tree is given by

\[
f_p(a) = \sum_{l=1}^{l} k_{lp} F(a \in T_{lp}),
\]

where \( F(a \in T_{lp}) = \begin{cases} 1, & \text{if } a \in T_{lp} \\ 0, & \text{otherwise} \end{cases} \)

By employing a regression tree to reinstate \( f_p(a) \) in the generic gradient boosting mechanism, the framework gradient descent stage size and updating equation are given by equation (2) and (3), respectively.

\[
f_p(a) = f_{p-1}(a) + \rho_p g_p(a).
\]

\[
\rho_p = \arg\min_{\rho} \sum_{l=1}^{l} M(b_{i, f_{p-1}(a_i)} + \rho g_p(a_i)).
\]

Hence, equations (2) and (3) fit as equations (4) and (5):

\[
f_p(a) = f_{p-1}(a) + \sum_{l=1}^{l} \rho_p k_{lp} F(a \in T_{lp}).
\]

\[
\rho_p = \arg\min_{\rho} \sum_{l=1}^{l} M(b_{i, f_{p-1}(a_i)} + \sum_{l=1}^{l} \rho_p k_{lp} F(a \in T_{lp})).
\]

By employing discrete ideal \( \rho_{lp} \) for each territory \( T_{lp} \), \( k_{lp} \) are supposed to be detached. The updated framework equations (6) and (7) are given by

\[
f_p(a) = f_{p-1}(a) + \sum_{l=1}^{l} \rho_p F(a \in T_{lp}).
\]

\[
\rho_p = \arg\min_{\rho} \sum_{l=1}^{l} M(b_{i, f_{p-1}(a_i)} + \sum_{l=1}^{l} \rho_p F(a \in T_{lp})).
\]

The framework overfitting can be restrained through managing the gradient boosting repetitions number,
more competent, appraising the beneficiation of each tree by $J \in (0, 1)$. Hence, the updated model is given by

$$f_p(a) = f_{p-1}(a) + L \sum_{l=1}^{L} \rho_p F(a \in T_{lp}).$$  \hfill (8)

3.2. Catboost. Catboost is a type of gradient boosting algorithm that is succeeded in recent times by Dorogush et al. [54]. Catboost tackles both the regression and classification problems and has been publicized in a new free open-source multiplatform gradient boosting library [54, 55]. Decision trees are employed as a base weak learner in the Catboost algorithm and gradient boosting to sequentially fit on the decision tree. In order to enhance the execution of the Catboost algorithm and to avoid overfitting, the inconsistent permutation of the learning information of the gradient is employed [54].

The Catboost algorithm aims to minimize the forecasting relocation that happens during the learning stage. The dissemination relocation is the removal of $F(y_i) | (y_i)$ with $y_i$ being a learning sample, with relation to $F(y) | (y_i)$ for a test sample $y$. At the learning stage, gradient boosting employs the same sample for the calculation of gradient and the framework that reduces that gradient. The concept of Catboost is to establish $j \ldots n$, the base framework for individual $P$ boosting repetition. The $ith$ framework of the $mth$ repetition is learnt of initial $ith$ samples of the permutation and is applicable to calculate the gradient of the $j+1$ sample for $p+I$ repetition. Succeeding, to be unrestrained of the start arbitrary permutation, this technique employs $s$ reciprocated arbitrary permutation. A distinguish framework is constructed per repetition that manages all permutations and frameworks. Symmetric trees work as a foundation for the frameworks. The trees are extended by growing all the leaf nodes level-wise by using the same splitting criteria.

The mechanism introduced in the Catboost algorithm is to calculate the contemporary character that is identical to the one imitated for establishing the network. Thus, for a specified arbitrary permutation of the samples, the data sample $< i$ is used to calculating the character value for each

<table>
<thead>
<tr>
<th>Table 3: Statistical distribution of UCS dataset.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_w$ (g/cm$^3$)</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Standard deviation</td>
</tr>
<tr>
<td>Minimum</td>
</tr>
<tr>
<td>Maximum</td>
</tr>
</tbody>
</table>

Figure 4: Pairwise correlation between the input variables and output in the original dataset.
sample \(i\). Later on, various permutations are executed and the acquired character value for each sample is averaged. Catboost is a large-scale and comprehensive library that is composed of various elements as standard boosting, GPU learning and consists of tenfold hyperparameters optimization to modify to several practicable investigation circumstances. Standard gradient boosting is also a part of the Catboost algorithm. Figure 6 depicts the description of Catboost employed in this study.

It should be noted that the training capability of the Catboost algorithm is supervised by its framework hyperparameter, namely, the number of iterations, learning rate, maximum depth, etc. The determination of the best hyperparameters for a model is a difficult, time-consuming, challenging, and tedious task contingent on the user's skills and expertise. Table 5 illustrates the execution of a Catboost framework in python employed in this study.

3.3. Light Gradient Boosting Machine (LightGBM). Light Gradient Boosting Machine, abbreviated as LightGBM, is an open-source gradient boosting machine learning framework by Microsoft that uses a decision tree as a based training algorithm [56]. LightGBM inserts consecutive element value buckets into discrete bins with higher efficiency and faster training speed. It employs the histogram-based algorithms [57,58] in order to enhance the learning stage, minimize memory consumption, and incorporate an updated communication network to improve training uniformity, known as parallel voting decision tree machine learning algorithm. The learning information is split into multiple trees, the local voting technique is implemented to choose the top-\(k\) element, and the globing voting technique is acquired in each repetition. LightGBM manipulates the leaf-wise strategy to perceive a leaf with maximum splitter gain as shown in Figure 7. It can be used for classification, regression, sorting, and numerous machine learning projects. LightGBM builds a sophisticated tree compared with the level-wise distribution technique by the leaf-wise distribution technique, which is the principal component for executing the algorithm with higher efficiency. Even so, it gives rise to overfitting; however, it can be prohibited by using the max-depth element in LightGBM.

LightGBM [56] is a comprehensive library that executes gradient boosting and intends some modification. The execution of gradient boosting has been mainly concentrated on building a computationally systematic algorithm. The library comprises tenfold training hyperparameters that authenticate this framework to implement in the diverse scenario. LightGBM execution also suggests advanced features both in CPU and in GPU; it can operate as gradient boosting and has multifold combinations including bootstrap subsampling and column randomization. LightGBM is broadly featured by Gradient-based One-Side Sampling and Exclusive Feature Bundling. Gradient-based One-Side Sampling is a subsample mechanism employed to build the learning information as a base tree in the ensemble. In the Adaboost machine learning algorithm, the objective of this mechanism is to enhance the importance of the sample with a higher probability, which is associated with the sample having a higher gradient. When the Gradient-based One-Side Sampling is implemented, the learning information for the base learner is formulated on the top fraction of the sample with a higher gradient (a) plus an arbitrary instance fraction (b) reacquired from the samples with a smaller gradient. To remunerate for the variation of measured dissemination, the sample in the lower gradient category is arranged together and weighed up by \((1 - x)/y\) when
calculating the data gain. On the other hand, the Exclusive Feature Bundling mechanism accumulates the sparse element into a single element. This can be accomplished without preventing any knowledge when those elements do not consist of a nonzero number coincidently. The mechanism anticipates supplementary learning rate gain. Table 6 illustrates the execution of a LightGBM framework in python employed in this study.

3.4. Extreme Gradient Boosting (XGBoost). Extreme gradient boosting (XGBoost) is one of the significant types of

<table>
<thead>
<tr>
<th>Description</th>
<th>Python script</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Import the python packages</strong></td>
<td>from sklearn.ensemble import GradientBoostingRegressor from sklearn.model_selection import cross_val_score, RepeatedKFold Framework = GradientBoostingRegressor() # Interpretation of the evaluation mechanism cv = RepeatedKFold(n_splits = 10, n_repeats = 3, random_state = 1) # Framework assessment n_scores = cross_val_score(Framework, x_train, UCS_train, scoring = 'neg_mean_absolute_error', cv = cv, n_jobs = -1) # Performance Report print('MAE: %.8f (%.8f)' % (mean(n_scores), std(n_scores))) # fit the framework on training dataset Framework.fit(x, UCS)</td>
</tr>
</tbody>
</table>

| Training dataset | Train_dataset_Forecasting = Framework.predict(x_train) |
| Testing dataset | Test_dataset_Forecasting = Framework.predict(x_test) |

Table 4: Executing a GBR framework in python employing the gradient boost regressor.

Figure 6: Description of Catboost employed in this study.
ensemble learning algorithms in machine learning methods [59]. It comprises ordinary regression and classification trees by incorporating analytical boosting techniques. Boosting enhances the precision estimation of the framework by building various trees as a substitution to establish a targeted tree and later on adding them to determine a coherent forecasting model [60]. XGBoost initiates the tree by sequentially retaining the residuals of past trees as influences on the consequence tree. Due to this, the consequence tree establishes comprehensive forecasting by generating the error of past trees. At the stage when the loss function is reduced, the sequential framework structure relation can be segmented as a gradient descent type that improves the prediction by adding an extra tree at each step to lower the reduction [61]. Tree growth ends when a preestablished number of the most extraordinary tree is gained, or when the error of the training stage cannot be augmented to a predicted sequential tree number. The performance promptness and evaluation accuracy of the gradient boosting can be significantly increased by attaching arbitrary investigation. Particularly for each tree in a symmetry, an arbitrary training data subsample is considered from the entire training dataset, excluding the replacement. Instead of the entire sample, this arbitrarily described subsample is then utilized to fit the tree and is ascertained to update the framework. XGBoost is a state-of-the-art redistributed gradient boosting algorithm that can manage and achieve up-to-date forecasting demonstration [62]. The second-order evaluation of the loss function is employed in the XGBoost algorithm which is expeditious and speedy as compared to the conventional gradient boosting algorithms. It has been extensively adopted to mine gene coupling features. Figure 8 depicts the description of XGBoost employed in this study.

Consider \( \bar{u}_i \) as the forecasted result of \( i \)th number of data for which the characteristics vector is \( V_i \). \( E \) shows the estimators number for each estimator \( f_k \) (with \( k \) from 1 to \( E \)) corresponding to individual tree anatomy, and \( u_i^0 \) depicts the preliminary assumption that is the average of the measured characteristics in the learning information. The following equation implements various expansion functions to forecast the results:

\[
\bar{u}_i = u_i^0 + \eta \sum_{k=1}^{E} f_k(V_i), \tag{9}
\]

whereas the \( \eta \) parameter is the learning rate that is adjoined to improve the model implementation to enhance the model, to execute rhythmically while linking the latest tree and confront overfitting.

Concerning Eq. 9, at \( k \)th state, a character \( k \) is connected to the model and the \( k \)th forecasted value \( u_i^{(k)} \) is implemented from the forecasted value at the preceding state \( u_i^{(k-1)} \) and the augmented \( f_k \) of the character of the attached \( k \)th character is depicted in

\[
u_i^{(k)} = u_i^{(k-1)} + \eta f_k, \tag{10}\]

whereas \( f_k \) shows the weight of leaves that are developed by reducing the objective function of the \( k \)th tree represented by

\[
\text{obj} = \gamma N + \sum_{a=1}^{N} G_a \omega_a + \frac{1}{2} (H_a + \lambda) \omega_a^2, \tag{11}\]
whereas N designates the leaves of the kth tree and \( \omega_a \) represent the weight of the leaves from 1 to N, and \( \gamma \) and \( \lambda \) are the uniformity features that are utilized to implement the coherence to the anatomy to evade the overfitting in the model. The parameters \( H_a \) and \( G_a \) are the aggregate of entire information attached with data \( a \) leaf of the prior and subsequent loss function gradient, respectively.

In sequence to build the kth tree, a single leaf is split into different numeration leaves. Eq. 12 indicates the anatomy of utilizing the gain parameters. Consider \( C_R \) and \( D_R \) depict the interdependent right leaf, and \( C_L \) and \( D_L \) are interdependent left leaf containing the divergence. At the time, the gain parameter is close by zero, and the diverging benchmark is traditionally assumed. The \( \gamma \) and \( \lambda \) are the uniformity features that are peripheristically susceptible on the gain feature; i.e., higher the regularization parameter will decrease the gain parameter which in result will avoid the convolution of the ex cruciating of the leaf. But it will decrease the ability of the framework in order to apt to the training dataset.

\[
gain = \frac{1}{2} \left[ \frac{C_L^2}{D_L + \lambda} + \frac{C_R^2}{D_R + \lambda} + \frac{(C_L + C_R)^2}{D_L + D_R + 2\lambda} \right]. \tag{12}
\]

XGBoost is a widely used machine algorithm that assembles an articulated and logical accomplishment of gradient boosting machine learning algorithm. A numerical value is the outcome of the regression prediction model problems. XGBoost can be practiced promptly to probabilistic regression frameworks. The ensembles are established from the decision tree model. Ensembles successively link trees and adjust the forecasting imprecision models. These types of ensemble machine learning techniques are known as boosting. The frameworks are established by implementing any arbitrary gradient descent optimization technique and distinct loss function. When the model is implemented, the gradient loss function is decreased, and therefore this mechanism is recognized as "gradient boosting." Figure 8 illustrates the execution of an XGBoost framework in python employed in this study. XGBoost framework performed well on the UCS datasets as compared to GBR, Catboost, and LightGBM with the same parameter \( n\text{-splits} = 10, n\text{-repeats} = 3, \) random_state = 1 (all the other parameters were used as a default in python). However, in order to further enhance the performance of the XGBoost framework, gbm_param_grid was further implemented in this study.

3.5. Performance Indices. In order to accurately and approximately assess the performance of machine learning algorithms, various researchers have employed different evaluation criteria, i.e., coefficient of correlation (\( R^2 \)) [63], mean absolute error (MAE), mean square error (MSE) [64], and root mean square error (RMSE) [65]. The higher value of \( R^2 \) and the lower MAE, MSE, and RMSE indicate that the predicted target values perform better in each proposed model. In this study, four estimation criteria are employed to narrate the association between measured and predicted UCS values, i.e., \( R^2 \), MAE, MSE, and RMSE as shown in Table 8.

4. Results and Discussion

UCS plays a significant role in the stability of geostuctures. In this study, the use of novel machine learning methods to estimate the accuracy of the UCS is necessary. This is because the accuracy of the studied UCS can help in designing any type of rock engineering project. Therefore, this study aims to incorporate these gradient boosting machine learning algorithms, namely, GBR, Catboost, LightGBM, and XGBoost to predict the UCS of soft sedimentary rocks of Block-IX at Thar Coalfield, Pakistan, by using four input variables, i.e., \( \rho_w \) (g/cm\(^3\)); moisture (%); \( \rho_d \) (g/cm\(^3\)); and BTS (MPa). Then, the 106-point dataset for each model was identically assigned as 70% for the training phase to train the model and 30% for the testing phase to test the model. In the end, the ultimate output of predicted UCS is as follows.

**Table 6:** Executing a lightgbm algorithm in python employing the lightgbm regressor.

<table>
<thead>
<tr>
<th>Description</th>
<th>Python script</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import the python packages</td>
<td>from lightgbm import LGBMRegressor()</td>
</tr>
<tr>
<td></td>
<td>from sklearn.model_selection import cross_val_score, RepeatedKFold</td>
</tr>
<tr>
<td></td>
<td>Framework = LGBMRegressor()</td>
</tr>
<tr>
<td></td>
<td># Interpretation of the evaluation Mechanism</td>
</tr>
<tr>
<td></td>
<td>cv = RepeatedKFold(n_splits = 10, n_repeats = 3, random_state = 1)</td>
</tr>
<tr>
<td></td>
<td># Framework assessment</td>
</tr>
<tr>
<td></td>
<td>n_scores = cross_val_score(Framework, x_train, UCS_train,</td>
</tr>
<tr>
<td></td>
<td>scoring = 'neg_mean_absolute_error', cv = cv, n_jobs = -1, error_score = 'raise')</td>
</tr>
<tr>
<td></td>
<td># Performance Report</td>
</tr>
<tr>
<td></td>
<td>print('MAE: %.8f (%.8f)' % (mean(n_scores), std(n_scores)))</td>
</tr>
<tr>
<td></td>
<td># fit the framework on training dataset</td>
</tr>
<tr>
<td></td>
<td>Framework.fit(x, UCS)</td>
</tr>
</tbody>
</table>

**Figure 8:** Description of XGBoost employed in this study.
Figures 9(a) to 9(d) and 10(a) to 10(d) show the scatter plots of the predicted UCS as opposed to the measured UCS in the training and testing phases, respectively, employing various algorithms. In Figures 9(a) to 9(d), the correlation coefficient ($R^2$) values of the GBR, Catboost, LightGBM, and XGBoost are 0.97, 0.95, 0.40, and 0.99 in the training phase, respectively. At the same time, Figures 10(a) to 10(d) demonstrate the $R^2$ values of the GBR, Catboost, LightGBM, and XGBoost are 0.94, 0.91, 0.39, and 0.99 in the testing phase, respectively. Figures 11 and 12 reveal the performance plots of predicted UCS as opposed to measured UCS in the training phase. Figure 12 reveals the performance plots of predicted UCS as opposed to measured UCS in the testing phase.

Figure 13 shows the various performance indices of proposed algorithms such as $R^2$, MAE, MSE, and RMSE. The performance indices can tell us the evaluation of the developed model by which we can distinguish the most suitable model. In this study, the performance indices of the proposed algorithms are tabulated in Table 9.

Taylor diagram describes a concise statistical description of how well the frameworks match their standard deviation and correlation. The following equation [66] represents the formula of the Taylor diagram:

$$ R = \frac{1/P \sum_P (r_n - \bar{r})(f_n - \bar{f})}{\sigma_r \sigma_f} \quad (13) $$

where $R$ represents a correlation, $P$ is the discrete point number, $r_n$ and $f_n$ are two vectors, $\sigma_r$ and $\sigma_f$ show the standard deviation of $r$ and $f$, and $r_n$ and $f_n$ depict the average value of vectors $r$ and $f$, respectively.

Figure 14 shows the association among the standard deviation, RMSE, and correlation of predicted UCS and measured UCS of GBR, Catboost, LightGBM, and XGBoost algorithms, respectively, from Figure 13. According to the results, the UCS of the XGBoost prediction algorithm was highly correlated with the measured UCS compared to the other studied algorithms. In addition, the standard deviation of the XGBoost is nearest to the original value of the standard deviation. As a result, the XGBoost demonstrates high accuracy as compared to the existing publicly available literature [7, 21, 40], and is proved to be the high accurate algorithm among all for UCS prediction. The GBR also has a standard deviation nearer to the original standard deviation but shows the smallest $R^2$ value. Meantime, Catboost and LightGBM have the smallest correlation and are further away from the original standard deviation values.
Table 8: Performance indices employed in this work.

<table>
<thead>
<tr>
<th>Performance indices</th>
<th>Definition</th>
<th>Mathematical formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R^2$</td>
<td>$R^2$ is an aggregate bench used to compute how significant the association between measured and predicted UCS values is. MAE shows the mean absolute error, which is a conventionally employed statistical feature that can inspect the real scenario of the predicted UCS value error. MSE as shown by equation (15) is a broadly practiced mathematical standard that shows the appropriate standard deviation of the error relative to the measured and predictive UCS value. RMSE is a statistical measure used to compute the mean of square error in UCS database.</td>
<td>$R^2 = 1 - \frac{\sum_{i=1}^{n} (UCS_{\text{measured}} - UCS_{\text{predicted}})^2}{\sum_{i=1}^{n} (UCS_{\text{measured}} - UCS_{\text{mean}})^2}$</td>
</tr>
<tr>
<td>MAE</td>
<td>$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n}</td>
<td>UCS_{\text{mean}} - UCS_{\text{measured}}</td>
</tr>
<tr>
<td>MSE</td>
<td>$\text{MSE} = \frac{1}{T} \sum_{i=1}^{n} (UCS_{\text{measured}} - UCS_{\text{mean}})^2$</td>
<td></td>
</tr>
<tr>
<td>RMSE</td>
<td>$\text{RMSE} = \sqrt{\frac{1}{2} \sum_{i=1}^{n} (UCS_{\text{measured}} - UCS_{\text{mean}})^2}$</td>
<td></td>
</tr>
</tbody>
</table>

UCS$_{\text{predicted}}$ and UCS$_{\text{mean}}$ represent the predicted values and the mean values, $T$ shows the total number of datasets, and UCS$_{\text{measured}}$ is the measured value.

Figure 9: Scatter plots of predicted UCS as opposed to measured UCS of different models in the training stage: (a) GBR, (b) Catboost, (c) LightGBM, and (d) XGBoost.
Figure 10: Scatter plots of predicted UCS as opposed to measured UCS of different models in the testing stage: (a) GBR, (b) Catboost, (c) LightGBM, and (d) XGBoost.

Figure 11: Continued.
The results of the present study are similar to a previous study in the literature where Zhong et al. [40] also developed the XGBoost algorithm for predicting of rock mechanical properties, i.e., UCS, in their study, which proved to have the best performance in results with maximum $R^2$ and smallest MAE and RMSE.
5. Sensitivity Analysis

Sensitivity analysis is a statistical tool that evaluates how target features are influenced based on changes in input features [67]. In this work, sensitivity analysis has been carried out by implementing "feature importance" module in python programming language on the superior performed XGBoost testing data. The following equation [68] depicts the formula of the feature importance:

\[
\text{XGboostfm}_m = \frac{\sum_{\text{all tree}} \text{fm}_{mn}}{T},
\]

where XGboostfm\(_m\) = the importance of feature \(m\) estimated from all trees in the XGBoost testing dataset, \(\text{fm}_{mn}\) = the feature importance for \(m\) tree in \(n\), and \(T\) indicates the entire number of trees.

From Figure 15, it is clear that BTS and \(\rho_w\) are positively correlated, and the moisture and \(\rho_d\) are negatively correlated.
with the UCS. The feature importance of each input feature is given as:

\[ \text{BTS} = 5.050, \quad \rho_w = -0.078, \quad \text{moisture} = -6.457, \quad \text{and} \quad \rho_d = 3.475. \]

### 6. Conclusion

UCS is one of the significant rock strength parameters and has various applications in mining, geology, geotechnical engineering, etc. An advanced data visualization model has been established in order to depict the research findings of various indirect tests to UCS. This study developed the four gradient boosting machine learning algorithms such as GBR, Catboost, LightGBM, and XGBoost for predicting UCS of soft sedimentary rocks of Block-IX at Thar Coalfield. The application of these developed algorithms has rarely been employed for the prediction of rock mechanical parameters, especially UCS. In order to execute the gradient boosting machine learning algorithms, the four significant input parameters \( \rho_w \) in g/cm\(^3\); moisture in \%; dry \( \rho_d \) in g/cm\(^3\); and \( \text{BTS} \) in MPa are used as input parameter and the corresponding UCS is employed as output parameter. Moreover, 106-point dataset was split identically for each algorithm into 70% for the training phase and 30% for the testing phase.

![Taylor diagram](image)

**Figure 14:** Taylor diagrammatic representation of GBR, XGBoost, Catboost, and LightGBM models in the (a) training phase and (b) testing phase.

![Sensitivity analysis](image)

**Figure 15:** Sensitivity analysis of UCS dataset.
phase. The developed models were executed in python programming language. In this study, the performance of each developed algorithm was computed and as a result, XGBoost outperformed the GBR, Catboost, and LightGBM with $R^2 = 0.99$, MAE = 0.00062, MSE = 0.0000006, and RMSE = 0.00079 in the training phase and $R^2 = 0.99$, MAE = 0.00054, MSE = 0.0000005, and RMSE = 0.00069 in the testing phase. In addition, according to the Taylor diagram, the standard deviation of the XGBoost is nearest to the original value of the standard deviation. Therefore, the XGBoost exhibits high accuracy in the training and testing phases, respectively. In this study, the developed XGBoost algorithm proved to be the best-fit algorithm for predicting the UCS of soft sedimentary rocks of the Block-IX at Thar Coalfield, Pakistan. Future work can be expanded employing different datasets to verify the more accuracy of the proposed algorithms.

**Data Availability**

The data, models, and code generated and/or employed during the study are available from the corresponding author upon request.

**Conflicts of Interest**

The authors declare no potential conflicts of interest.

**Acknowledgments**

This research was supported by the Science and Technology Innovation Project of Guizhou Province (Qiankehe Platform Talent [2019] 5620 to XXZ).

**References**


