Data-Driven Prediction of D-Exponent in Drilling Operations Using XGBoost and LightGBM Algorithms

Faramarz shahkoomahalli[1*] - Ataollah amiri[2] - H.Gheybparvar[3]
[1] Institute of Petroleum Engineering, School of Chemical Engineering, College of Engineering, University of Tehran, Tehran, Iran
[2] Institute of Petroleum Engineering, School of Chemical Engineering, College of Engineering, University of Tehran, Tehran, Iran
[3] PGFK CO. Offshore Drilling Operation Manager

Email : f.shahkoomahalli76@gmail.com

ABSTRACT

Accurate prediction of key parameters in drilling operationsparticularly the D-Exponent plays a crucial role in optimizing processes and reducing operational costs. ***.*** *In this study, real-time drilling data from two directional wells. including parameters such as Weight on Bit (WOB), Equivalent Circulating Density (ECD), Rotary Speed (RPM), and other relevant variables were used to evaluate the performance of two machine learning algorithms: XGBoost and LightGBM. To train and evaluate the models, 80% of the dataset was used for training and the remaining 20% for testing. The results demonstrated that both models are capable of accurately predicting the D-Exponent value. Specifically, the XGBoost model achieved an accuracy of 98% with a Mean Absolute Error (MAE) of 0.073, while the LightGBM model reached an accuracy of 97% with an MAE of 0.074. These methods not only enable real-time prediction but also support drilling engineers in making timely and effective decisions to enhance operational efficiency and reduce costs. The findings of this study suggest that integrating artificial intelligence and machine learning into drilling processes can significantly improve productivity, enhance safety, and minimize operational risks in the oil and gas industry.*

Keywords: *Machine learning, Artificial intelligence, Extreme gradient boosting, LightGBM ,D-Exponent, Ensemble Learning*

1. INTRODUCTION

Drilling oil and gas wells is one of the most complex and costly phases in the energy industry, requiring precise management of operational parameters and a deep understanding of subsurface geological conditions. This process involves drilling to significant depths through formations with varying characteristics, where any error in predicting subsurface conditions can lead to increased costs, project delays, and even serious safety risks such as blowouts and wellbore collapse [1]. In this context, the D- exponent has emerged as a key indicator for analyzing and predicting drilling conditions. This parameter normalizes the rate of penetration (ROP) relative to operational parameters, enabling real-time analysis of subsurface pressure and geological variations. Utilizing the D-exponent allows drilling engineers to more rapidly detect pressure changes and formation transitions, facilitating timely and effective decision-making. Given the numerous complexities involved in drilling operations, the use of modern approaches such as machine learning models and big data analytics has gained traction in recent years. These advanced techniques offer the potential for more accurate and optimized predictions of key drilling parameters, ultimately helping to reduce operational risks.

The D-exponent was first introduced by Jorden and Shirley in 1966, with the objective of normalizing the rate of penetration relative to key operational parameters such as Weight on Bit (WOB), Rotary Speed (RPM), and Bit Diameter. This normalization helps isolate the effects of formation properties from those of operational variables. The original formula for calculating the D-exponent is given as follows: [2]

$$D = \frac{log_{10}\left(\frac{ROP}{60}\right)}{log_{10}\left(\frac{WOB}{RPM}\right)} \quad (1)$$

Where:

- D : the D-exponent.
- ROP : the rate of penetration, measured in feet per hour (ft/hr).
- WOB: the weight on bit, measured in pounds (lbf).
- RPM: the rotary speed of the bit, measured in revolutions per minute (reround/min).

However, considering the effect of bit diameter on the rate of penetration and weight on bit modified the original formula by incorporating the bit diameter. The revised D-exponent formula is expressed as follows: [3]

$$d_c = \frac{log_{10}\left(\frac{ROP}{RPM * 60}\right)}{log_{10}\left(\frac{12 * WOB}{D_b}\right)} \quad (2)$$

Where D_b is the bit diameter (in inches). This modification improved the accuracy of subsurface pressure prediction and enhanced the practical applicability of the D-exponent. The corrected formula allows drilling engineers to numerically analyze variations in the rate of penetration and use it to estimate pore pressure and assess geological formation conditions. The D-exponent serves as an indicator for identifying abnormal formation pressures in geological strata. Unexpected changes in the D-exponent value may signal overpressure or underpressure zones, which—if not detected in time—can lead to serious incidents such as well blowouts or borehole collapse. Theoretically, a negative value for the d-exponent is neither physical nor acceptable, as the d-exponent is an empirical indicator used to estimate formation hardness based on the rate of penetration (ROP), weight on bit (WOB), rotary speed (RPM), and bit diameter. Under normal drilling conditions, this value is expected to be positive. However, in practice, negative d-exponent values may be observed due to the following reasons:

- Errors in Input Data
- Drilling in Very Soft Formations

In specific cases involving the drilling of very soft formations with low bit weight and high penetration rates, the RPM may become so high that the logarithmic ratio in the d-exponent formula becomes unexpectedly negative. Additionally, drilling through hard or compact formations requires more force to be applied from the bit to the formation in order to achieve effective cutting. This increased force results in a higher Weight on Bit (WOB). Since WOB appears in the denominator of the d-exponent formula, a significant increase in WOB can reduce the logarithmic ratio, potentially leading to a negative d-exponent value which is typically unexpected and indicates the need to re-examine the input data. However, such a situation is also uncommon and, in most cases, the input data should be thoroughly reviewed and validated. Therefore, accurate prediction and trend analysis of the D-exponent during drilling operations is of critical importance. Recent advances in artificial intelligence and machine learning have enabled the analysis of large-scale and complex drilling data [4]. Machine learning models have demonstrated the ability to identify nonlinear and intricate patterns within the data, thereby providing more accurate predictions. By utilizing real-time drilling parameters, these models can offer more precise estimations of the D-exponent and subsurface pressures, supporting drilling engineers in making more informed decisions to optimize operations.

2. XGBOOST ALGORITHM

XGBoost, short for eXtreme Gradient Boosting, is an optimized and distributed library for gradient boosting that is designed for high performance, flexibility, and portability. This algorithm is an improved version of the traditional Gradient Boosted Decision Tree (GBDT) and can be effectively applied to both classification and regression problems. XGBoost consists of two main components: the decision tree and the gradient boosting algorithm. A decision tree is a method used for project risk assessment and feasibility evaluation. In this approach, a tree-like structure is constructed to estimate the probability that the Net Present Value (NPV) is equal to or greater than zero, based on predefined probabilities for various scenarios. This technique employs an intuitive form of probabilistic analysis and is called a "decision tree" due to the branching structure that resembles a natural tree. In machine learning, decision trees are foundational algorithms used for both classification and regression tasks. When used for classification, the model is referred to as a classification tree, and when applied to regression problems, it is known as a regression tree.[5]

The Boosting algorithm is a type of ensemble learning method aimed at converting a collection of weak learners into a single strong learner. The core idea is to iteratively train several weak models by adjusting the weights of training samples at each step, and then combine these models linearly to construct a more powerful predictive model. The Gradient Boosting algorithm is a specific type of boosting technique in which, at each training iteration, the model attempts to minimize the residual error from the previous stage.[7]

The core idea of Boosting in the XGBoost algorithm is to initially construct a simple, low-accuracy CART model (Classification and Regression Tree). This base model is then iteratively improved: at each step, a new model is trained to evaluate and correct the errors of the previous model, resulting in a progressively more accurate overall model.

To evaluate the error at each tree structure, the algorithm uses the derivative of the loss function, which is defined as follows: [6]

$$g_i = \partial_{y_i^{(t-1)}} l\left(y_i, p_i^{(t-1)}\right); G_j = \sum_{i \in I_j} g_i \quad (3)$$

$$h_i = \partial^2_{y_i^{(t-1)}} l\left(y_i, p_i^{(t-1)}\right); H_j = \sum_{i \in I_j} h_i \quad (4)$$

Where gi and hi are the gradients obtained from the first and second derivatives based on the error of the previous structure, and Gj and Hj are the gradient values in each j-th leaf by adding the gradient in each set of index data points, yi is the predictor of each sample and pi is the predicted result. The best tree structure that has a minimum objective function value which is generally divided into 2 forms (training loss and regularization).

$$f_{obj} = L + \Omega \quad (5)$$

$$L = \sum_i (y_i, p_i)^2 \quad (6)$$

$$\Omega = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (7)$$

Here, L represents the training loss, which is typically calculated using the Mean Squared Error (MSE) and reflects the model's performance on the training data. Ω is the regularization term, used to control model complexity and prevent overfitting.γ (Gamma) is a parameter associated with the number of leaves in the tree. λ (Lambda) is the regularization parameter for leaf weights in regression models, commonly referred to as reg_lambda. W_j is the weight assigned to the j-th leaf. T denotes the total number of leaves in the tree.

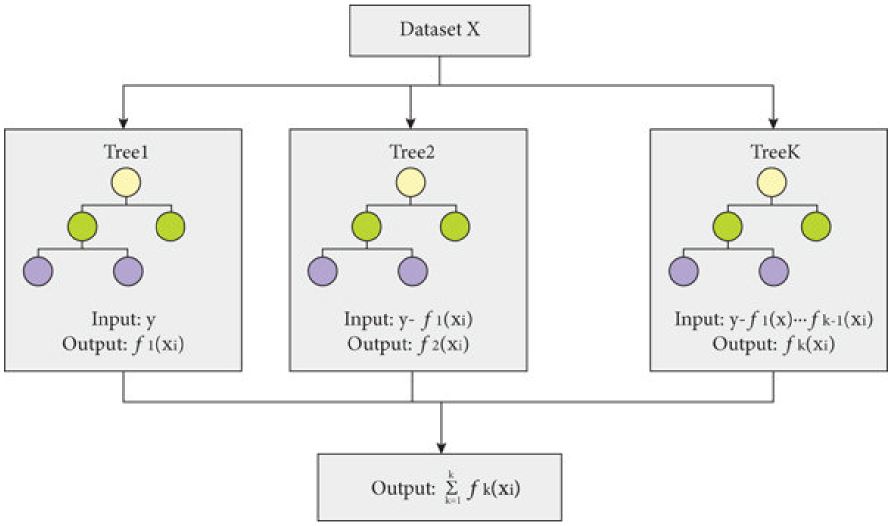

Fig.1. *Flow chart of XGBoost framework [5]*

3. LIGHTGBM ALGORITHM

LightGBM is a tree-based learning algorithm and a framework for Gradient Boosting. It is designed with a focus on high scalability and improved efficiency, offering several advantages:

- Reduced memory usage
- Higher training speed and efficiency
- Improved prediction accuracy

This algorithm incorporates two innovative techniques that significantly enhance its speed compared to other boosting algorithms: [9]

- Gradient-Based One-Side Sampling (GOSS)
- Exclusive Feature Bundling (EFB)

These techniques are key to LightGBM's superior performance in large-scale learning tasks. Figure 2 presents a schematic representation of the LightGBM algorithm. Also Table 1 presents a comparison between the two algorithms selected for this study.

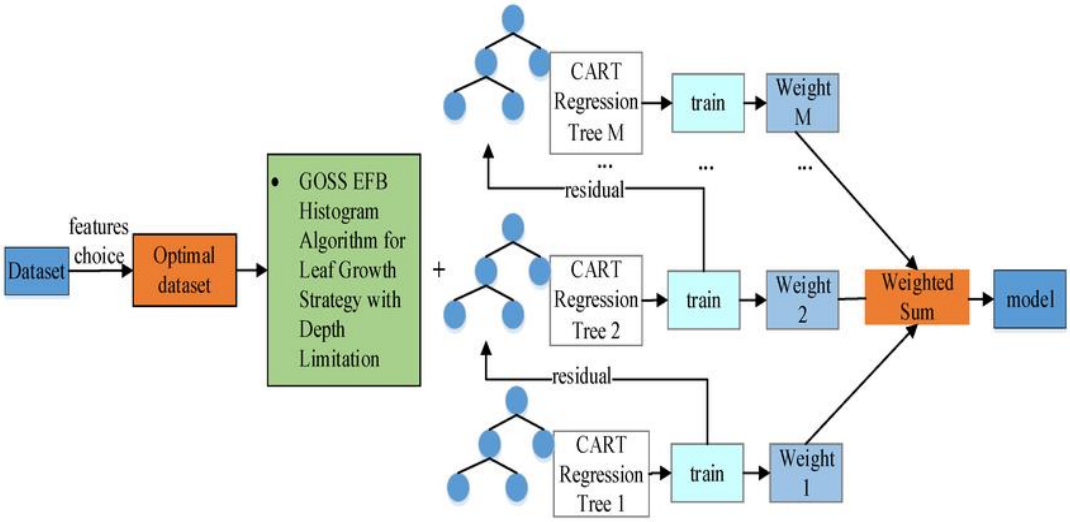

Fig.2. *Flow chart of LightGBM framework [8]*

***Table 1**. Comparison of XGBoost and LightGBM Algorithms*

Feature	XGBoost	LightGBM
Training speed	slower	faster
Memory consumption	higher	lower
Support of large datasets	good	Very good
Sensivity to overfitting	lower	higher

4. DATA PROCESSING AND CALCULATIONS

4.1. Data Analysis

The data used in this study were obtained from drilling operations of two directional wells, numbered 5 and 35, located in one of the oil fields in Iran. A total of 8,300 drilling data points were collected via mud logging from these two wells within a specific block, forming the dataset used in the analysis. Figure 3 provides a summary of the statistical analysis of the dataset.

	depth	hookload	rpm_bit	flow_in	mudweight_in	mudweight_out	ecd	wob	torque	d_exponent
count	8300.000000	8300.000000	8300.000000	8300.000000	8300.000000	8300.000000	8300.000000	8300.000000	8300.000000	8300.000000
mean	2093.000000	191.911711	109.811084	1380.710630	82.210365	82.325320	83.137423	13.486217	4.347164	1.186184
std	1198.097427	53.911171	79.114607	1083.935525	18.300055	18.186491	17.977218	7.825785	3.674308	0.697119
min	11.000000	54.300000	-96.000000	137.240000	47.430000	-44.220000	38.250000	0.000000	-151.000000	-0.743000
25%	1055.750000	152.100000	77.000000	354.545000	71.907500	71.090000	74.200000	7.700000	2.360000	0.948000
50%	2093.000000	184.750000	119.000000	866.183000	77.315000	77.410000	77.410000	12.100000	4.260000	1.395000
75%	3130.250000	227.625000	160.000000	2112.883250	81.160000	81.470000	81.620000	17.825000	6.000000	1.635000
max	4175.000000	345.700000	917.000000	3678.345000	149.710000	141.090000	149.710000	53.500000	36.000000	2.968000

Fig.3. *Summary of sample data statistical analysis*

4.2. Data Normalization

Initially, missing and outlier data were removed from the dataset. Given that the variables exhibit a wide range of variation, they can have a disproportionate impact on the overall model results, which is statistically and analytically unreasonable. Therefore, to address the scale differences among various. evaluation indicators, it is necessary to normalize each feature In this study, normalization was performed using two standard techniques: MinMaxScaler and StandardScaler. These methods transform the values of each feature into a range of [0, 1], ensuring consistency in scale across all input variables.

$$X^* = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{8}$$

$$X_{norm} = \sqrt{\frac{\sum(X_i - \bar{X})}{n}} \tag{9}$$

In these equations, X^* represents the data normalized using MinMaxScaler, X_{norm} denotes the data normalized using StandardScaler, X_{max} and X_{min} are the maximum and minimum values of the data, respectively, $\bar{X}$ is the mean of the data, and n is the total number of data points.

4.3. Training and Testing Data Selection

In this study, a total of 8,300 data points from two wells were used. Eighty percent (80%) of the data were allocated for training, while the remaining twenty percent (20%) were used for testing, in order to evaluate the model's accuracy in predicting the d-exponent.

4.4. Hyperparameter Tuning

The XGBoost and LightGBM algorithms include a variety of hyperparameters that must be configured prior to model training. This process is considered a critical phase in model development. The primary goal of initializing these hyperparameters is to tailor the model to the variability in the data it analyzes—in this study, drilling data collected through mud logging. Proper tuning can significantly improve the model's performance in estimating the target parameter, d-exponent. One of the automated methods for hyperparameter optimization is the use of the GridSearchCV (GS) module. This tool systematically evaluates all combinations of hyperparameter values within a defined grid and selects the configuration that yields the best performance. The results for several sample hyperparameter settings of the XGBoost algorithm are presented in Table 2.

Table 2. *Hyperparameter values after tunning using GridSearchCV (GS).*

hyperparameter	default	lightgbm	xgboost
N_estimators	100	1000	800
Max_depth	6	2	6
Reg_lambda	1	0.84	0.84
Learning_rate	1	0.08	0.1
gamma	0	0	0

5. DISCUSSION

To assess the predictive performance of the models on the test dataset, statistical analysis was conducted using three metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and the Coefficient of Determination (R^2 Score). The values of these metrics, which are used to evaluate model accuracy and to identify the best-performing model, are presented in Table 3. In Figures 4 and 5, the measured d-exponent (calculated using

Equation 1) is plotted against the predicted d-exponent obtained from the two machine learning algorithms, XGBoost and LightGBM. In the range of -0.5 to 0, the data are uniformly distributed, while in the 0 to 1 interval, a higher degree of data dispersion is observed. This deviation from actual values is attributed to a lack of sufficient training data in this range. However, in the range of 1 to 3, both models have demonstrated high predictive accuracy and have effectively estimated the target parameter.

Table 3. *MAE , MSE , R^2 of two models*

LIGHTGBM	XGBOOST	
0.074	0.073	MAE
0.014	0.010	MSE
0.97	0.98	R^2

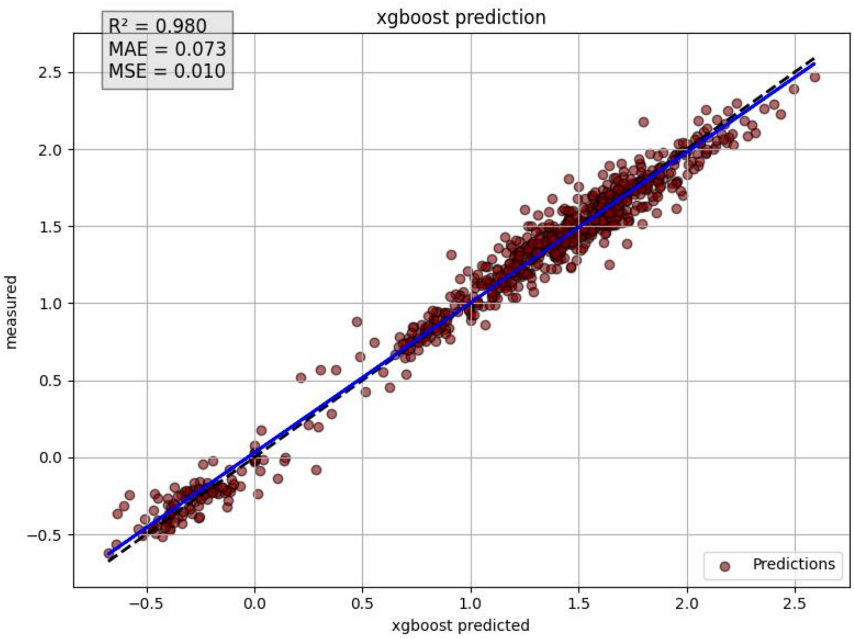

Fig.4. *Predicted* d-exponent *values by the XGBoost model versus actual* d-exponent *values*

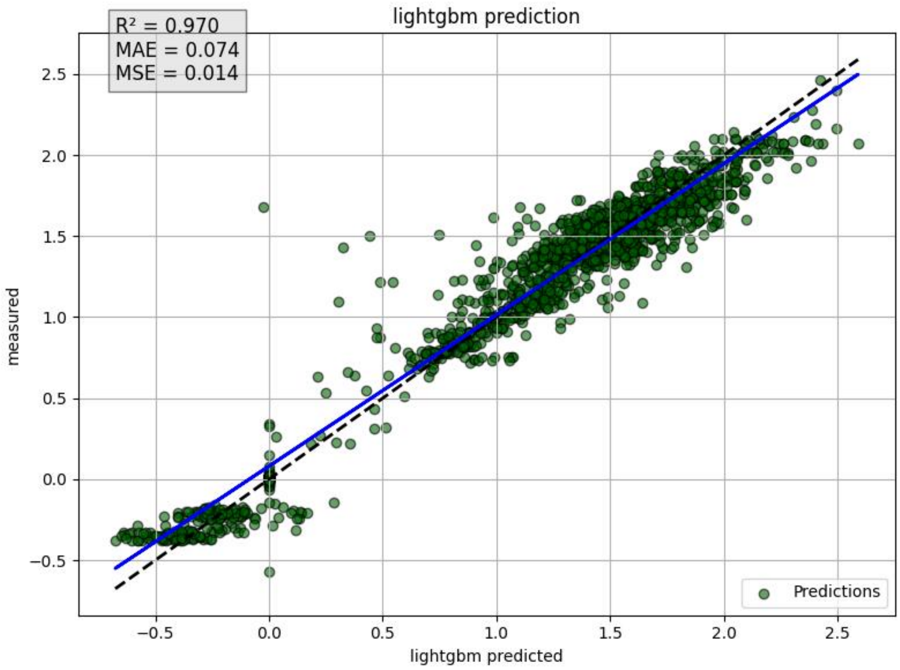

Fig.5. *Predicted d-exponent values by the LightGBM model versus actual d-exponent value*

6. CONCLUSIONS

Drilling engineers use the d-exponent parameter to estimate rock drillability. This parameter is calculated using drilling parameters such as rate of penetration (ROP), bit diameter, rotary speed (RPM), and weight on bit (WOB), along with the lithology of the formation. However, traditional methods for calculating the d-exponent are often time-consuming and prone to computational errors. In this study, drilling data from two directional wells were utilized to predict the d-exponent using two machine learning algorithms: XGBoost and LightGBM. The results demonstrated very high accuracy—97% and 98%, respectively—highlighting the strong performance of both models in predicting the d-exponent. These findings suggest that such models can be effectively used in field applications to reduce computation time and eliminate manual calculation errors.

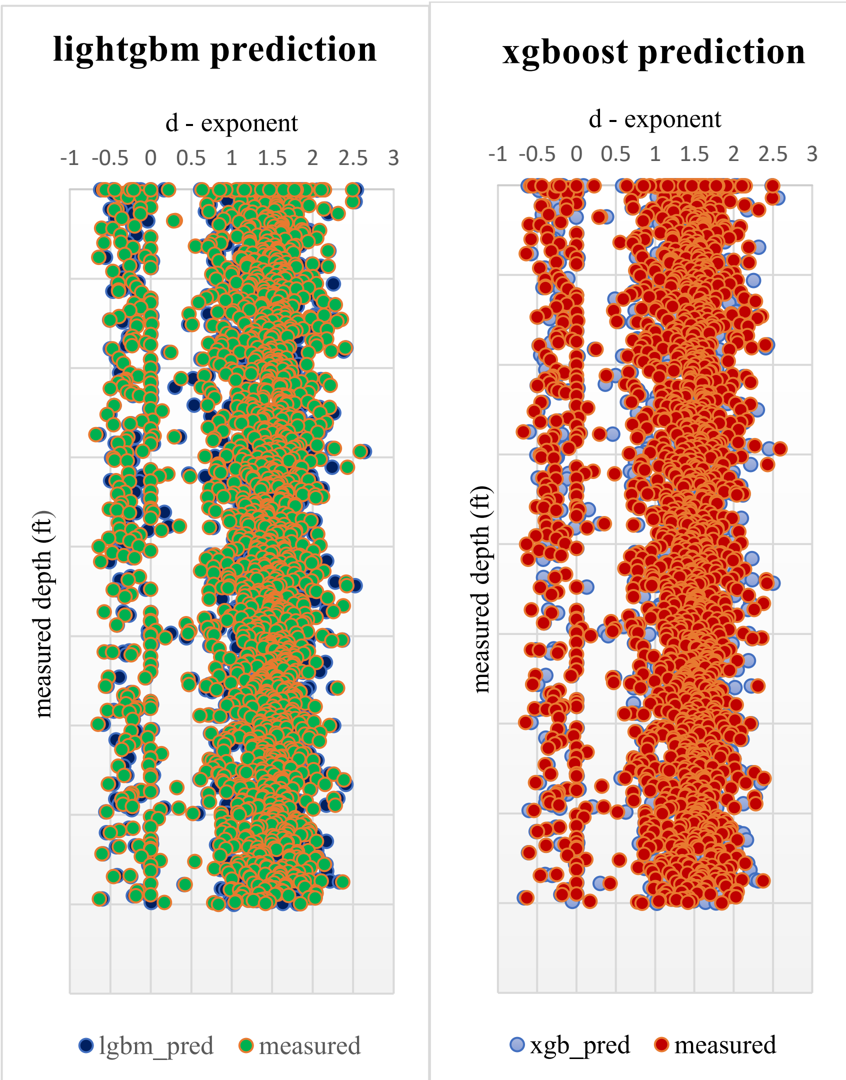

Fig.6. *Comparison of d-exponent values predicted by the proposed model and those measured during drilling, across test data points at corresponding depths, to evaluate the performance of the Jordan model versus the proposed model.*

REFERENCES

[1] *Real-Time Artificial Intelligence-Enhanced Machine Learning for Drilling Parameter Prediction. (2024). SPE Annual Technical Conference and Exhibition (ATCE), Society of Petroleum Engineers. https://onepetro.org/SPEATCE/proceedings abstract/24ATCE/24ATCE/563683.*

[2] *Jorden, J. R., & Shirley, O. J. (1966). Application of drilling performance data to overpressure detection. Journal of Petroleum Technology,* 18*(11), 1387-1394. https://doi.org/10.2118/1387-G.*

[3] *Bourgoyne, A. T., Millheim, K. K., Chenevert, M. E., & Young, F. S. (1986). Applied Drilling Engineering (2nd ed.). Society of Petroleum Engineers.*

[4] *Wang, Y., Zhang, L., & Li, J. (2024). Pore pressure prediction based on conventional well logs and machine learning methods. Journal of Petroleum Science and Engineering, 210, 110345. https://doi.org/10.1016/j.petrol.2024.110345.*

[5] *Zhou, F., Fan, H., Liu, Y., Ye, Y., Diao, H., Wang, Z., Rached, R., Tu, Y., & Davio, E. (2022). Application of XGBoost algorithm in rate of penetration prediction with accuracy. Presented at the*

International Petroleum Technology Conference, Riyadh, Saudi Arabia, 21-23 February. https://doi.org/10.2523/IPTC-22100-MS.

[6] *Yu Ling, Wu Tiejun. Ensemble learning: Boosting algorithm overview [J]. pattern recognitionand artificial intelligence, 2004, 17 (01): 52–59.*

[7] *Friedman J H 2001 Greedy function approximation: a gradient boosting machine Annals of statistics 1189–1232.*

[8] *Liu, J.-J., & Liu, J.-C. (2022). Permeability predictions for tight sandstone reservoir using explainable machine learning and particle swarm optimization. Geofluids, 2022, Article ID 2263329. https://doi.org/10.1155/2022/2263329.*

[9] *Yu Ling, Wu Tiejun. Ensemble learning: Boosting algorithm overview [J]. pattern recognitionand artificial intelligence, 2004, 17 (01): 52–59.*