processes

MDPI

Article

Bottomhole Pressure Prediction of Carbonate Reservoirs Using XGBoost

Hao Sun, Qiang Luo *, Zhaohui Xia, Yunbo Li and Yang Yu

Research Institute of Petroleum Exploration and Development, PetroChina, Beijing 100083, China;
sunhao19982023@163.com (H.S.)
* Correspondence: riped_lq@petrochina.com.cn; Tel.: +86-158-8206-9972

Abstract: The bottomhole pressure is one of the key parameters for oilfield development and decision-making. However, due to factors such as cost and equipment failure, bottomhole pressure data is often lacking. In this paper, we established a GA-XGBoost model to predict the bottomhole pressure in carbonate reservoirs. Firstly, a total of 413 datasets, including daily oil production, daily water production, daily gas production, daily liquid production, daily gas injection rate, gas–oil ratio, and bottomhole pressure, were collected from 14 wells through numerical simulation. The production data were then subjected to standardized preprocessing and dimensionality reduction using a principal component analysis. The data were then split into training, testing, and validation sets with a ratio of 7:2:1. A prediction model for the bottomhole pressure in carbonate reservoirs based on XGBoost was developed. The model parameters were optimized using a genetic algorithm, and the average adjusted R-squared score from the cross-validation was used as the optimization metric. The model achieved an adjusted R-squared score of 0.99 and a root-mean-square error of 0.0015 on the training set, an adjusted R-squared score of 0.84 and a root-mean-square error of 0.0564 on the testing set, and an adjusted R-squared score of 0.69 and a root-mean-square error of 0.0721 on the validation set. The results demonstrated that in the case of fewer data variables, the GA-XGBoost model had a high accuracy and good generalization performance, and its performance was superior to other models. Through this method, it is possible to quickly predict the bottomhole pressure data of carbonate rocks while saving measurement costs.

Keywords: carbonate reservoirs; bottomhole pressure; machine learning; XGBoost; data driven

check for updates

Citation: Sun, H.; Luo, Q.; Xia, Z.; Li, Y.; Yu, Y. Bottomhole Pressure Prediction of Carbonate Reservoirs Using XGBoost. *Processes* **2024**, *12*, 125. https://doi.org/10.3390/pr12010125

Academic Editors: Tao Zhang, Zheng Sun, Dong Feng, Hung Vo Thanh and Wen Zhao

Received: 13 November 2023
Revised: 22 December 2023
Accepted: 27 December 2023
Published: 3 January 2024

1. Introduction

The bottomhole pressure is one of the key parameters for oilfield development and decision-making. The difference between the bottomhole pressure and the formation pressure largely determines the oil production rate of a well. Traditional methods of obtaining the bottomhole pressure include installing downhole pressure sensors for measurement, a well-testing analysis, or establishing numerical simulation models. The installation of pressure gauges for all wells is costly and carries risks such as equipment lifespan and failures, leading to expensive workovers for equipment replacement and production delays. Well-testing methods also require well shutdowns, which affect economic benefits and involve complex operations. Numerical simulation methods are time-consuming, labor-intensive, and resource-demanding. Therefore, it is crucial to develop a real-time pressure prediction model that saves time, effort, and costs in order to continuously and accurately obtain the bottomhole pressure without shutting down the wells. The traditional correlation prediction methods for forecasting the bottomhole pressure rely on the establishment of empirical and semiempirical (mechanistic) models. Studies by Duns and Ros [1], Hagedorn and Brown [2], Orkiszewski [3], and Beggs and Brill [4] have focused on empirical models, while Mukherjee and Brill [5], Ansari et al. [6], Chokshi et al. [7], and Gomez et al. [8] have researched semi-empirical (mechanistic) models. However, these methods are largely based on a series of specific conditions. Therefore, when applied beyond these conditions, the effectiveness of these methods deteriorates.

Artificial intelligence (AI) has a strong adaptability and flexibility in various environments and datasets, and is widely used in fields such as medicine, transportation, the Internet, and energy. In the application of petroleum engineering, AI is often used to solve problems related to data, optimization, identification, and knowledge integration [9,10]. AI technology can more accurately predict bottomhole pressure by learning patterns and correlations from a large quantity of historical data (as shown in Table 1), especially in complex reservoir conditions and dynamic oilfield development processes. Awadalla and Yousef (2016) [11] used backpropagation algorithm to find the optimal parameters of a feedforward neural network (FFNN) for predicting the bottomhole flowing pressure of vertical oil wells. Mahshid and Suren (2019) [12] predicted the bottomhole flowing pressure of coalbed methane wells using time data, linear regression, and neural-network-based methods. Ahmadi and Chen (2019) [13] compared various artificial neural network methods for predicting the bottomhole pressure of vertical wells, and the results showed that a hybrid genetic algorithm and particle swarm optimization (HGAPSO) was highly accurate. Nait Amar et al. (2018) [14] demonstrated that a hybrid model based on artificial neural network (ANN) and grey wolf optimization (GWO) (ANN-GWO) performed better than other hybrid methods such as genetic algorithm (GA), particle swarm optimization (PSO), or using only a BPNN (backpropagation neural network). Nait Amar and Zeraibi (2020) [15] proposed a hybrid model based on support vector regression (SVR) and the firefly algorithm (FFA) for predicting the bottomhole pressure (BHP) in vertical wells under multiphase flow conditions. Rathnayake et al. (2022) [16] compared multiple linear regression, linear mixed effects, and gradient boosting regression tree (XGBoost) methods for predicting the flowing bottomhole pressure (FBHP) in gas wells, and the results showed that the XGBoost method had the best prediction results.

Table 1. Summary of artificial intelligence algorithms for bottomhole pressure prediction.

Reference	Artificial Intelligence Algorithm	Application
[14]	A hybrid model based on support vector regression (SVR) and the firefly algorithm (FFA)	Vertical wells with multiphase flow
[15]	Gradient boosting regression trees (XGBoost)	Gas wells
[17]	Feedforward neural network (FFNN)	Vertical oil wells
[18]	Neural network	Coalbed methane wells
[19]	Hybrid genetic algorithm and particle swarm optimization (HGAPSO)	Vertical oil wells
[20]	Artificial neural network (ANN) and grey wolf optimization (GWO) (ANN-GWO)	Vertical wells with multiphase flow

Based on previous research, this study proposes a well bottom-pressure prediction model based on XGBoost. The XGBoost algorithm [21] is an ensemble learning algorithm based on gradient boosting trees, which has been widely applied in various fields such as healthcare [22] and telecommunications [23] and has been introduced in several articles regarding its application in the petroleum field. For example, Pan et al. applied XGBoost to predict porosity in well logging [24]; Markovic et al. applied XGBoost to predict water saturation in rock physics [25]; Zhong et al. used XGBoost to generate pseudodensity logging data [26]; Gu et al. applied it to predict permeability in geology [27]; Al Mudhafar et al. applied XGBoost to the reservoir lithology classification of carbonate rocks [28]; Zhang et al. applied it to predict hydrocarbon gases in petrochemistry [29]; Dong et al. used XGBoost for reservoir production prediction [30]; and Wang et al. applied XGBoost to predict reservoir production in highly heterogeneous sandstone [31]. These applications have demonstrated that XGBoost is an efficient, superior, and reliable algorithm. Compared with traditional machine learning algorithms, XGBoost exhibits a higher accuracy, greater flexibility, and is less prone to overfitting, demonstrating significant advantages in large-scale, efficient, and accurate problems. However, there is almost no application of XGBoost in well bottom-pressure prediction in carbonate reservoirs.

One major issue faced by machine learning models, including XGBoost, is the close relationship between the model's performance and its hyperparameters. Using default parameter values may not fully harness the powerful effects of XGBoost. Conventional grid-search algorithms require manual parameter tuning, which consumes a lot of time and effort and makes finding the global optimal value challenging. The enhanced elite strategy genetic algorithm is an improved genetic algorithm that can more easily find the global optimal solution. Therefore, this paper proposes the use of an improved genetic algorithm to optimize the hyperparameters of XGBoost in order to enhance the prediction accuracy and generalization performance of the XGBoost model. The data used in this study were production data collected from a numerical model fitted to the historical data of the X oilfield. Prior to establishing the XGBoost-based prediction model for carbonate reservoir bottomhole pressure, various preprocessing steps such as data correlation analysis and data dimensionality reduction were conducted. The optimized XGBoost model with adjusted hyperparameters was compared with ten other machine learning models, namely, support vector machines, neural networks, stochastic gradient descent, linear regression, ridge regression, decision trees, random forests, gradient boosting trees, ExtraTrees, and Adaboost, to evaluate the superior predictive performance of XGBoost. These models were selected due to their frequent utilization in machine learning problems. The novelty of this work compared to other studies in the field lies in the following aspects: (1) The X oilfield is a matrix-pore-type carbonate reservoir with localized fractures and dissolution cavities, as well as high-permeability streaks and impermeable interlayers, exhibiting an extremely strong heterogeneity. The uneven pressure distribution during the development process poses a significant challenge for pressure prediction. The use of numerical simulation for pressure prediction requires a great deal of work, and there is currently no suitable empirical formula for pressure prediction in this type of oilfield. Using XGBoost through data-driven methods for pressure prediction is a relatively feasible approach. (2) Previous research often involves collecting extensive geological, development, engineering, well logging, and seismic data to achieve optimal machine learning results. However, due to the inconvenience of a practical application caused by the need for extensive data collection, this work utilizes only six easily collectible production data variables (daily oil production, daily water production, daily gas production, daily liquid production, daily gas injection, and gas–oil ratio) to test the performance of the model in the case of limited data variables. The utilization of a smaller set of variables in model training also facilitates the rapid implementation of real-time bottomhole pressure prediction in the field. (3) The model compares the pressure prediction performance of ten different machine learning algorithms with that of GA-XGBoost, demonstrating the superior accuracy and generalization ability of XGBoost in pressure prediction problems with limited data variables. This workflow also has certain limitations, as the data used are derived from numerical simulations. Although the model has undergone historical fitting, there are still errors compared to actual production data.

2. Principles and Methods

2.1. The Principles of the XGBoost Method

XGBoost [21] is an ensemble algorithm that combines multiple weak classifiers into a strong classifier based on the decision tree algorithm. It enhances the accuracy of predictive models by constructing multiple CART decision trees. The final prediction is obtained by summing the predictions of each trained decision tree in each round of training.

The principle of the XGBoost algorithm is as follows [32]:

$$\hat{y}_i = \sum_{t=1}^{n} f_t(x_i), f_t \in F \tag{1}$$

In the equation, n represents the number of trees in the model. f_t denotes a function in the function space F. $\hat{y}_i$ represents the predicted value, and x_i represents the i-th input data. F represents the set of all possible CART models.

At each iteration, the model remains unchanged, and a new function is added to the existing model. Each function corresponds to a tree, and the newly generated tree is used to fit the residual of the previous prediction. The iterative process is as follows:

$$\begin{cases} \widehat{y}_i^{(0)} = 0 \\ \widehat{y}_i^{(1)} = f_1(x_i) = \widehat{y}_i^{(0)} + f_1(x_i) \\ \widehat{y}_i^{(t)} = \widehat{y}_i^{(t-1)} + f_t(x_i) \end{cases} \tag{2}$$

The objective function of XGBoost is as follows [33]:

$$\mathrm{X_{obj}} = \sum_{i=1}^{n} l(y, \widehat{y}) + \sum_{k=1}^{K} \Omega(f_k) \tag{3}$$

$$\Omega(f_k) = \gamma T + \lambda \frac{1}{2} \sum_{j=1}^{T} \omega_j^2 \tag{4}$$

In Equation (3), $\sum_{i=1}^{n} l(y, \widehat{y})$ is used to measure the discrepancy between predicted scores and true scores, and $\sum_{k=1}^{K} \Omega(f_k)$ is the regularization term that effectively prevents overfitting. In Equation (4), T represents the number of leaf nodes, ω represents the score of each leaf node, γ is used to control the number of leaf nodes, and λ ensures that the scores of leaf nodes are not too large.

XGBoost has several main parameters that need to be adjusted: the number of sub-models, the learning rate, the maximum depth, the minimum child weight, γ, the subsample rate, and the column sampling rate. These hyperparameters affect the final performance of the model but in different ways. Therefore, it is necessary to optimize these hyperparameters.

2.2. Genetic Algorithm

Genetic algorithms (GAs) have been proven to be superior to traditional methods when performing a global search within a complex search space [34]. Usually starting from a group of randomly generated candidates embedded with potential solutions, a GA creates a new generation in each iteration by transferring crucial information such as survival of the fittest, crossover, and mutation operations onto promising candidates selected based on the probability biased towards relatively fitter agents [35].

The enhanced elite retention strategy for genetic algorithm (SEGA) takes advantage of the best individuals (called elites) generated during evolution, which are not subjected to selection, crossover, and mutation but directly copied to the next generation. The main feature of this operation is that elites produced during evolution will not be lost or damaged due to pairing crossover and other operations, significantly improving the ability to reach global convergence.

This paper utilizes the SEGA (Sequential Evolutionary Grid Algorithm) to continuously optimize the parameters of XGBoost and achieve the maximum cross-validation adjusted R-squared score, thereby accurately predicting the bottomhole pressure. The SEGA initially generates a batch of XGBoost hyperparameter initial values within a specified range and trains them with production data from the reservoir to obtain a score for the bottomhole pressure prediction. A higher score indicates better model performance. Based on the objective of maximizing the score, the elite preservation strategy is employed to generate the next generation of superior XGBoost hyperparameter populations, which are then trained to obtain scores. This process continues until the maximum evolution generation or the optimization objective reaches a predetermined threshold. The model training process is illustrated in Figure 1. By utilizing this method, the SEGA automatically searches for the optimal parameters of the XGBoost model. Compared to conventional grid-search methods, it eliminates the need for exhaustive parameter exploration, resulting in faster optimization speed, lower computational cost, and significant time and effort savings.

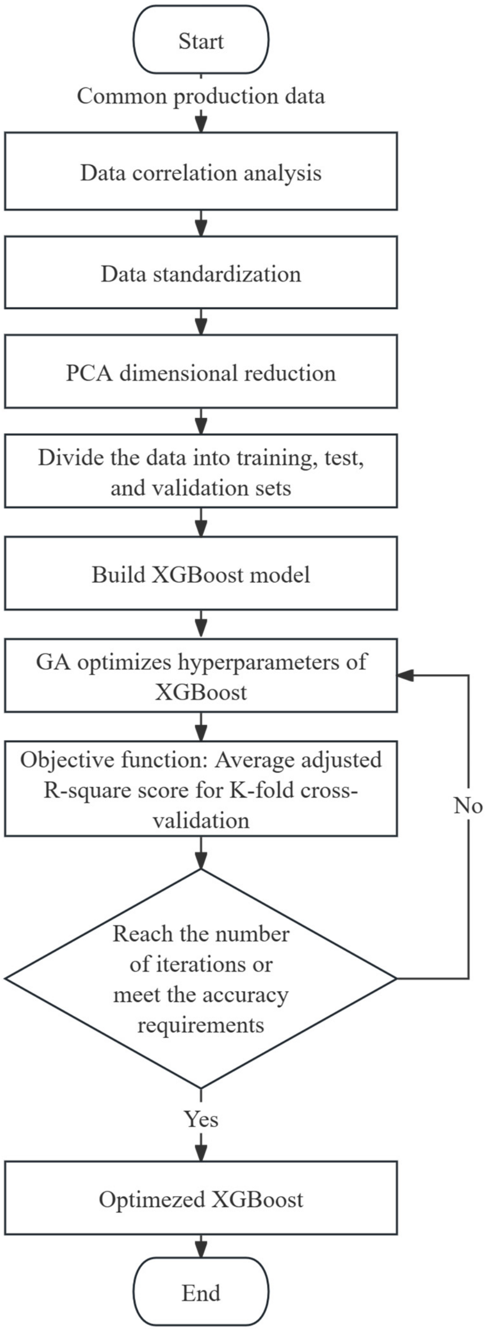

Start
Common production data
Data correlation analysis
Data standardization
PCA dimensional reduction
Divide the data into training, test, and validation sets
Build XGBoost model
GA optimizes hyperparameters of XGBoost
Objective function: Average adjusted R-square score for K-fold cross-validation
Reach the number of iterations or meet the accuracy requirements
No
Yes
Optimezed XGBoost
End

Figure 1. GA-XGBoost model flow diagram.

3. Model Verification and Application

The numerical simulation, through historical fitting, can represent, to some extent, the actual conditions of the target oilfield. The numerical model of the target oilfield had a grid size of 300 × 300 × 20 m, with a total of 239,000 effective grid cells. The boundaries were closed on all sides, and there were 8 components in the model. The model simulation parameters are shown in Table 2. Based on the actual production data and pressure test data, the fitting of a fixed oil volume and single well pressure was carried out, and the overall fitting effect was good.

Table 2. Model parameters.

Parameters	Average Reservoir Thickness per Well, m	Average Permeability, mD	Average Porosity, %	Average Oil Saturation, %	Average Production per Well, bbl
Values	225	0.19678039	12.5	78.4	30,000

Common production data (daily oil production, daily water production, daily gas production, daily liquid production, daily gas injection rate, gas–oil ratio, and bottomhole pressure) were collected from the numerical model as the dataset. The bottomhole flowing pressure was treated as the target value, while the remaining data served as input values.

A total of 413 datasets were collected from 14 wells. It can be observed from Figure 2, the distribution of the bottomhole pressure was uneven, mainly concentrated in the area of 5500 psi to 8300 psi, with the most data distributed around 7000 psi.

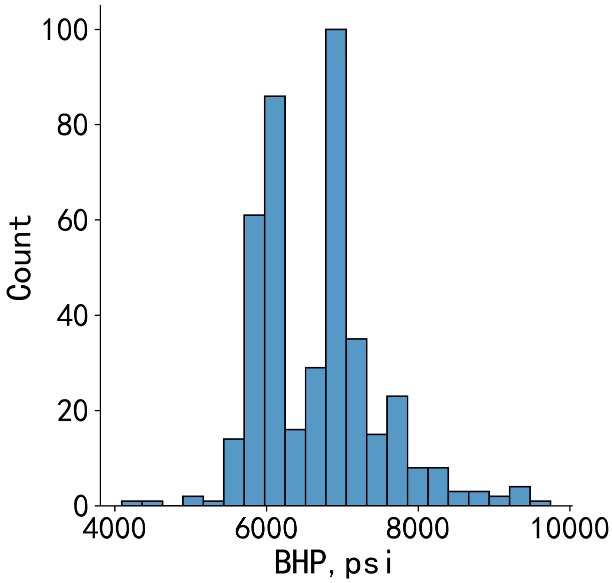


Figure 2. Data distribution histogram of bottomhole pressures.

3.1. Data Standardization

Due to the inconsistent dimensions of the original data, some of which may vary significantly, models tend to give more weight to features with larger scales during training, thus affecting the accuracy of the model. In addition, machine learning models are designed for data that follow a Gaussian distribution. If the data do not conform to a normal distribution, it will also affect the accuracy of the model. Therefore, data standardization is generally performed before training the model.

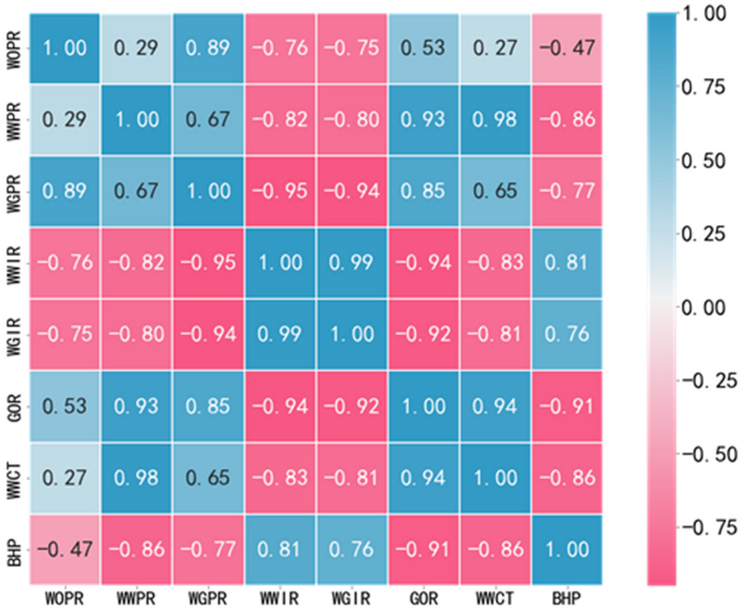
This article used the standard deviation normalization method (standardScale) to make the processed data conform to the standard normal distribution, with a mean of 0 and a standard deviation of 1. The formula is as follows:

$$x^* = \frac{x - \mu}{\sigma} \tag{5}$$

where x^* represents the standardized data, x represents the original data, μ represents the mean of the data, and σ represents the standard deviation of the data.

3.2. Dimension Reduction by Principal Component Analysis

First, the Pearson algorithm [36] was employed to calculate the correlation coefficients between each pair of features, which were then represented using a heatmap as shown in Figure 3. WOPR refers to the oil production rate of a well, WWPR refers to the water production rate of a well, WGPR refers to the gas production rate of a well, WWIR refers to the water injection rate of a well, WGIR refers to the gas injection rate of a well, GOR refers to the gas–oil ratio of a well, WWCT refers to the water cut of a well, and BHP refers to the bottomhole pressure of a well. In the figure, the values of the cells where rows intersect represent the correlation between the variables represented by rows and columns. If it is a positive number, it indicates a positive correlation between these two variables, while a negative number indicates a negative correlation between these two variables. The absolute value of the correlation coefficient ranged from 0.8 to 1.0, indicating a strong correlation between features. It can be observed that variables such as the gas–oil ratio and water cut exhibited a high correlation with the bottomhole flowing pressure (−0.91 and −0.86). Additionally, there were also high correlations among the variables themselves (e.g., the correlation between WWCT and WWPR was 0.98), which could potentially interfere with the predictive performance of the model.



	WOPR	WWPR	WGPR	WWIR	WGIR	GOR	WWCT	BHP
WOPR	1. 00	0. 29	0. 89	−0. 76	−0. 75	0. 53	0. 27	−0. 47
WWPR	0. 29	1. 00	0. 67	−0. 82	−0. 80	0. 93	0. 98	−0. 86
WGPR	0. 89	0. 67	1. 00	−0. 95	−0. 94	0. 85	0. 65	−0. 77
WWIR	−0. 76	−0. 82	−0. 95	1. 00	0. 99	−0. 94	−0. 83	0. 81
WGIR	−0. 75	−0. 80	−0. 94	0. 99	1. 00	−0. 92	−0. 81	0. 76
GOR	0. 53	0. 93	0. 85	−0. 94	−0. 92	1. 00	0. 94	−0. 91
WWCT	0. 27	0. 98	0. 65	−0. 83	−0. 81	0. 94	1. 00	−0. 86
BHP	−0. 47	−0. 86	−0. 77	0. 81	0. 76	−0. 91	−0. 86	1. 00

Figure 3. Heatmap for correlation analysis of data.

A principal component analysis (PCA) [37] is a multivariate statistical algorithm used to assess the correlation between multiple variables. It employs an orthogonal transformation to convert a set of potentially correlated variables into a set of linearly uncorrelated variables called principal components. The PCA algorithm can transform the original data into several principal components while preserving as much of the original information as possible, with each principal component being mutually independent. A PCA serves the purpose of dimensionality reduction by extracting and synthesizing relevant information while reducing the interference of redundant features and accelerating the model training process.

Through the PCA algorithm, the original seven features were mapped and linearly combined to construct new features known as principal components. The importance of the principal components can be evaluated based on their contribution to the variance. A higher variance contribution indicates greater importance. We used Python to call the PCA algorithm of the scikit learn framework and reduce the data to five dimensions based on the criterion of variance contribution rate greater than 95%. The variance contribution rates of the five principal components are shown in Table 3. From Table 3, it can be observed that the first principal component contributed the most, accounting for 52.15% of the variance, while the fifth principal component contributed the least, with a variance contribution rate of 3.65%.

Table 3. Principal component analysis data features.

Features	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
Variance contribution rate	0.52153446	0.19678039	0.11851757	0.08397276	0.03653219

The correlation heatmap of the five principal components after dimensionality reduction is presented in Figure 4. It can be seen that the absolute correlation values of the principal components after dimensionality reduction were all less than 0.5, and the principal components were independent of each other, which was conducive to the performance of the model.

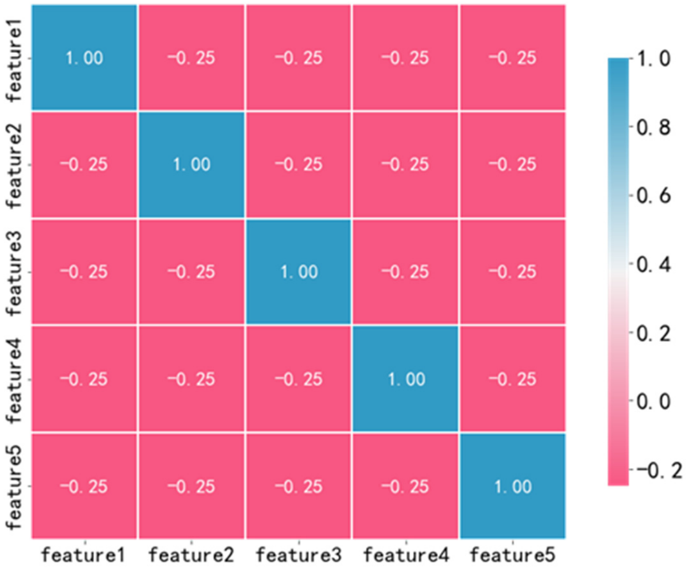


Figure 4. Correlation analysis after dimensionality reduction.

3.3. Model Training

The corresponding parameters of the genetic algorithm used in this study are presented in Table 4. The number of generations was set to 50, as it is generally believed that if convergence is not achieved by the 50th generation, it is necessary to investigate potential issues with the data or the model, as further increasing the generations may not yield significant improvements. The population size has a certain impact on the final results, with larger populations having a higher likelihood of finding the optimal solution. However, this also increases computational complexity. To ensure a balance between accuracy and efficiency, the population size was set to 20. The significance of the convergence threshold is that if the difference between the current generation and the best values from previous generations is smaller than this threshold, it indicates that the optimization of the objective has reached a stagnation state. If the evolution stagnates beyond the maximum limit set, the algorithm will stop running.

Table 4. The corresponding parameters of the genetic algorithm used in this paper.

Parameters	Parameter Range
Number of generations	50
Population size	20
Threshold for determining stagnation in single-objective optimization	1×10^{-6}
Maximum threshold value for evolution stagnation counter	10

The data were randomly split into training, testing, and validation sets with a 7:2:1 ratio, with the number of samples in each set presented in Table 5.

Table 5. Number of sample sets.

Data Set	Training Set	Test Set	Verification Set
Number	296	75	42

Using the SEGA, the parameters were iteratively adjusted to maximize the indicator of the average adjusted R-squared score obtained through cross-validation on the training set. Cross-validation [38–41] is a practical method employed in the machine learning training process, which involves dividing the data samples into smaller subsets. By performing a K-fold cross-validation, it simulates various possible datasets, enabling a thorough testing of the model's performance. This approach effectively utilizes the data to evaluate the model's performance and reduces issues such as overfitting and selection bias, thereby enhancing the model's generalization capability. Based on the empirical formula provided below, a cross-validation fold number of 6 was selected.

$$K \approx \log(n) \tag{6}$$

where K represents the number of cross-validation folds and n represents the size of the dataset. By randomly partitioning the data, each subset contains a mix of best- and worst-case scenarios. The cross-validated score obtained by averaging the scores from each subset may appear lower compared to the conventional training set score. However, it provides a more comprehensive evaluation of the model's performance across different datasets, making it a suitable objective function for optimization purposes.

The R-squared coefficient of determination indicates the goodness-of-fit of the trend line. Given a series of true values (y_i) and their corresponding predicted values, R-squared is defined as follows:

$$R^2 = 1 - \frac{\sum_i (\widehat{y}_i - y_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{7}$$

The R-squared value is in the range (−∞,1] and represents the proportion of variance explained by the model. It provides an intuitive understanding of the model's performance. As R-squared approaches 1, the model's performance improves. But as the number of independent variables increases, the R-squared value becomes larger, and using the adjusted R-squared value to evaluate the fitting effect of the model is more accurate. The adjusted R-squared takes into account both the sample size n and the number of independent variables k. Therefore, it reflects the modelling accuracy.

$$\text{Adjusted } R^2 = 1 - \left(1 - R^2\right)\frac{(n-1)}{(n-k-1)} \tag{8}$$

RMSE (root-mean-square Error) is simultaneously used as a supplementary validation metric to assess the performance of the model. The RMSE measures the square root of the ratio between the squared deviations of the predicted values and the observed values, divided by the number of observations, n.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(Y_i - f(x_i))^2} \tag{9}$$

The RMSE quantifies the deviation between predicted values and actual values, and it is particularly sensitive to outliers in the data. A smaller RMSE value indicates higher model performance.

The optimization process is illustrated in Figure 5, where the average score of the population improves from below 0.9350 to above 0.9525. The final average score of the population is comparable to that of the elites. The model optimization process terminates after the 18th generation.

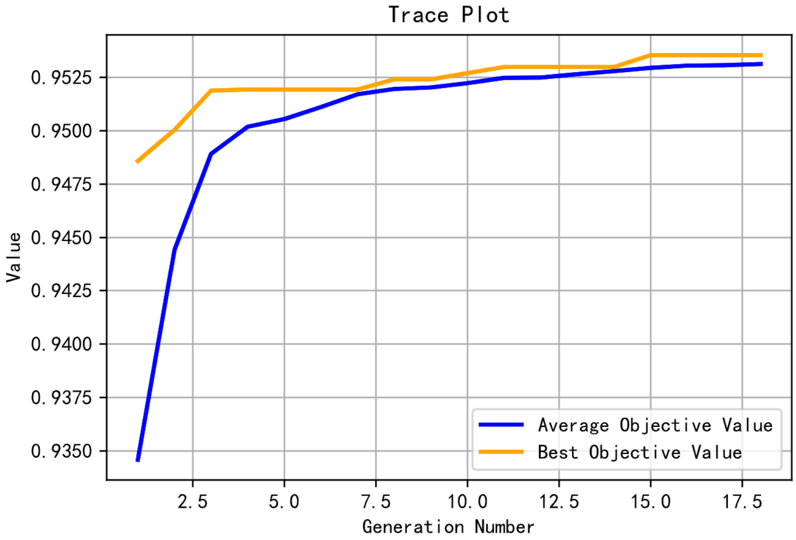


Figure 5. Genetic algorithm optimization process.

After optimization using the genetic algorithm, the optimal values of the XGBoost parameters were found and are shown in Table 6. The average adjusted R-squared score from the cross-validation was 0.9536.

Table 6. The parameters of the XGBoost model established in this paper.

Parameters	Parameter Value
n_estimators	6401
learning_rate	0.0338
max_depth	4
min_child_weight	6.5425
gamma	0.0233
subsample	0.9483
colsample_bytree	0.6000

To verify the effectiveness of XGBoost, we compared it with other common machine learning models, including support vector machines (SVM), neural networks, stochastic gradient descent (SGD), linear regression, ridge regression, decision trees, random forests, gradient boosting trees, ExtraTrees, and Adaboost. Support vector machines can handle the interaction of nonlinear features but are sensitive to missing data. Artificial neural networks can fully approximate complex nonlinear relationships but require a large number of parameters. The parameter update speed of random gradient descent is fast, but it may converge to local optima. Linear regression is easy to implement, but it cannot fit nonlinear data. Although ridge regression is less prone to overfitting compared to linear regression, it is also unable to handle nonlinear data. Decision trees are easy to understand and interpret, but they are prone to overfitting. Random forest performs well and can handle high-dimensional data, but it may overfit in some noisy problems. Gradient boosting decision trees have a good generalization ability but perform poorly on high-dimensional sparse datasets. ExtraTrees is a variant of random forests with better generalization performance at times. The Adaboost algorithm is a typical boosting algorithm that is not prone to overfitting, but imbalanced data can lead to decreased accuracy and sensitivity to outliers. Table 7 shows the parameters used by each model. These models were chosen because they have performed well in previous applications. Neural networks [13] are used for predicting bottomhole pressure in vertical multiphase flow oil wells, with a prediction error of no more than 10%. A support vector machine [42] was used for comparison in predicting bottomhole flow pressure in multiphase flow, as it represents a commonly used ML method and was therefore classified as the benchmark method in FBHP prediction. Linear regression [16] has been used to predict the bottomhole flow pressure of coalbed methane wells, with high accuracy in single-well prediction problems. Other models [43–49] have performed well in similar application fields.

Table 7. The parameters used by each model.

Model	Parameters
SVM	C: 1; epsilon: 0.1; gamma: 0.167
ANN	single hidden_layer_sizes: 100; learning_rate_init: 0.003
SGD	penalty: l2; alpha: 0.0001
Linear regression	fit_intercept: True; normalize: False
Ridge regression	alpha: 1
Decision trees	min_samples_split: 2; min_samples_leaf: 1
Random forests	n_estimators: 100; min_samples_split: 2; min_samples_leaf: 1
Gradient boosting trees	learning_rate: 0.1; n_estimators: 100; alpha: 0.9; max_depth: 3
ExtraTrees	n_estimators: 100; min_samples_split: 2; min_samples_leaf: 1
Adaboost	n_estimators: 100; learning_rate: 1

4. Results Analysis and Comparison

4.1. Results Analysis

The optimized XGBoost model had an adjusted R-squared score of 0.99 and a root-mean-square error of 0.0015 on the training set; the adjusted R-squared score on the test set was 0.84, with a root-mean-square error of 0.0564; the adjusted R-squared score of the validation set was 0.68, with a root-mean-square error of 0.0721. We drew a scatter plot with the true value as X and the predicted value as Y in Figure 6, where the comparison between the true and predicted values of the model on the training, testing, and validation sets is shown. A scattered distribution above the grey dashed line indicates that the predicted value is greater than the true value, a scattered distribution below the grey dashed line indicates that the predicted value is equal to the true value, and a scattered distribution below the grey dashed line indicates that the predicted value is less than the true value. The closer the scatter points are to the grey dashed line, the closer the predicted value is to the true value. It can be seen that the points formed by the predicted values and the true values of the model in the training set are basically distributed near the line with a slope of one, indicating that the predicted values were relatively close to the true values. For unseen data, the model still performed well in both the test and validation sets, with predicted values close to actual values, indicating good generalization performance.

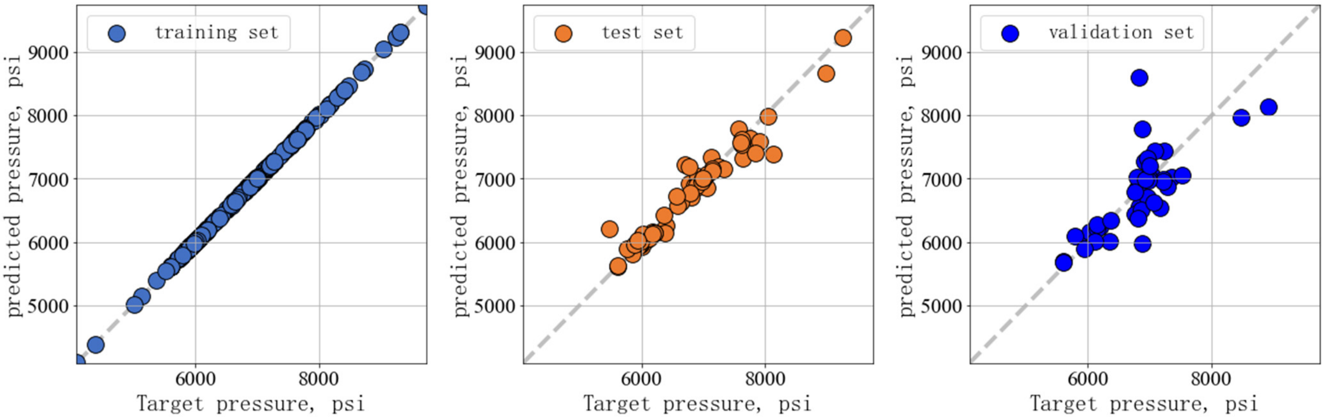


Figure 6. Predicted versus true values of the XGBoost model.

It can be seen that the validation set has significant fluctuations in the predicted values of the model at 7000 psi, with the distribution of predicted values ranging from 6000 psi to 9000 psi, which may be due to data-driven limitations. Due to the high dependence of data-driven methods on the quality and quantity of data, the prediction error of the model for unseen data such as the validation set is relatively large compared to the training set, which is inevitable. In addition, compared with traditional models, pure data-driven models do not have physical-knowledge constraints, and some predicted values may have significant deviations. In future work, improving XGBoost by incorporating physical-knowledge constraints may lead to better predictive performance.

4.2. Comparison of Models

The performance of the XGBoost model was compared with that of ten other common machine learning models, including support vector machines (SVM), neural networks, stochastic gradient descent (SGD), linear regression, ridge regression, decision trees, random forests, gradient boosting trees, ExtraTrees, and Adaboost. The comparison of adjusted R-squared scores for these models on different datasets is illustrated in Figure 7.

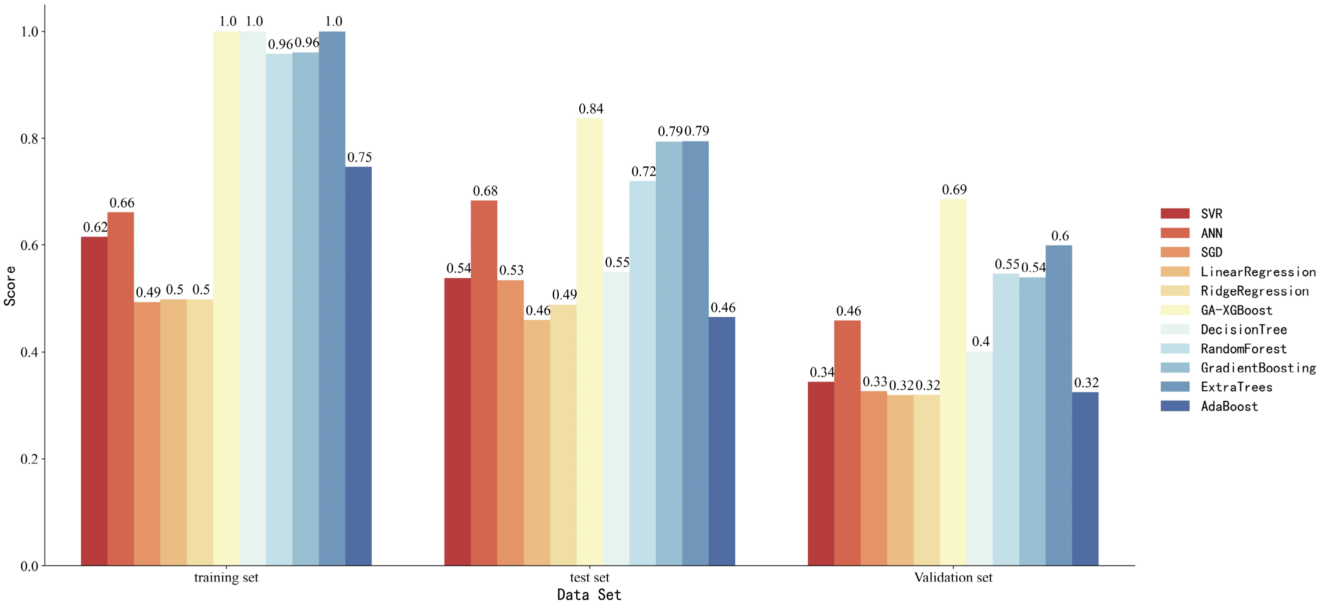


Figure 7. Comparison of Adjusted-R-squared scores for different datasets across models.

Linear regression, ridge regression, and Adaboost models performed poorly, with scores of only 0.5, 0.51, and 0.51 on the training set. This may be due to the complex nonlinear relationship between the bottomhole pressure and input variables in heterogeneous reservoir backgrounds, and these models cannot fit nonlinear data well. Previous studies used support vector regression and neural networks for predicting bottomhole pressure, and both performed well. However, with fewer data variables, support vector regression scored only 0.62 on the training set, while the neural network scored only 0.66 on the training set. It can be observed that tree-based models, such as decision trees and random forests, exhibited superior performance compared to other machine learning algorithms like support vector machines in addressing this particular problem. The training set scores of these tree-based models consistently exceeded 0.9. However, due to their tendency to overfit, the tree-based models yielded lower scores on the testing and validation sets compared to the XGBoost model. The XGBoost model optimized by GA demonstrated excellent performance across all three datasets.

5. Discussion

In this study, we selected the XGBoost algorithm, which exhibits a high accuracy and strong generalization, to establish a bottomhole pressure prediction model for carbonate reservoirs. We used a genetic algorithm with an enhanced elite strategy to optimize XGBoost's hyperparameters, with the objective function being the average adjusted R-squared score of a K-fold cross-validation, and compared its performance with other popular machine learning models. We selected six key variables, such as daily oil and water production, for the bottomhole pressure prediction, enabling a quick on-site implementation with limited data. The aim was to assess whether XGBoost could still perform well under limited data variable conditions. A correlation analysis revealed strong links between variables, potentially impacting the model training effectiveness. We used dimensionality reduction algorithms to convert the data into independent variables, minimizing interference from redundant features. We employed the cross-validation score as the genetic algorithm's optimization target, enhancing data use and the model's ability to generalize. By comparing XGBoost with commonly used machine learning algorithms, we verified the superior accuracy and generalization capabilities of XGBoost.

We established an XGBoost bottomhole pressure prediction model and optimized hyperparameters using SEGA to evaluate the pressure prediction performance of XGBoost under limited data volume. The optimized XGBoost model had an adjusted R-squared score of 0.99 and a root-mean-square error of 0.0015 on the training set; the adjusted R-squared score on the test set was 0.84, with a root-mean-square error of 0.0564; the adjusted R-squared score of the validation set was 0.69, with a root-mean-square error of 0.0721, and the superiority of GA-XGBoost was established through a comparison with other models. The poor performance of the linear regression, ridge regression, and Adaboost models may be due to the complex nonlinear relationship between the bottomhole pressure and input variables in heterogeneous reservoir backgrounds, and these models cannot fit nonlinear data well. Previous studies have used support vector regression and neural networks to predict bottomhole pressure, both of which have achieved good results. However, in the case of limited data variables, the performance of support vector regression and neural networks on the training set was also relatively average. It can be observed that tree-based models, such as decision trees and random forests, exhibited superior performance in solving this specific problem compared to other machine learning algorithms such as support vector machines. However, due to their tendency towards overfitting, tree-based models scored lower on the test and validation sets compared to XGBoost models. XGBoost, optimized by the GA model, demonstrated excellent performance on all three datasets, demonstrating the performance of the model.

6. Conclusions

The GA-XGBoost model demonstrated its ability to accurately estimate pressure values in challenging scenarios characterized by a significant heterogeneity and extreme variations in pressure distribution during the development of carbonate reservoirs. It maintained this capability even in cases with fewer data variables. This advantage of the model eliminates the need to install pressure gauges, resulting in cost savings.

Compared to numerical simulation techniques, predicting bottomhole pressure using GA-XGBoost eliminates the need for complex model building and historical fitting operations, while significantly improving computational speed. The use of genetic algorithms for automatic optimization avoids the extensive trial-and-error process required by conventional grid-search methods, saving time and effort.

Predicting the bottomhole pressure through a data-driven approach only requires commonly available production data, enabling a convenient and real-time estimation of bottomhole pressure. This method is more practical and facilitates on-site implementation of real-time bottomhole pressure prediction.

The GA-XGBoost model achieved an adjusted R-squared score of 0.99 and a root-mean-square error (RMSE) of 0.0015 on the training set. On the test set, the model obtained an adjusted R-squared score of 0.84 with an RMSE of 0.0564, while on the validation set, it achieved an adjusted R-squared score of 0.69 with an RMSE of 0.0721. Compared to other machine learning models, the GA-XGBoost model was superior in predicting bottomhole pressure with limited data features, exhibiting a high accuracy and good generalization performance. It achieved excellent results in the bottomhole pressure prediction on the training, test, and validation sets.

In future work, several trends are expected. Firstly, collecting data of various types and larger quantities will allow us to evaluate whether XGBoost's accuracy and generalization can be improved when dealing with massive test datasets, while also assessing the computational speed of XGBoost. Secondly, introducing constraints based on prior physical knowledge during the XGBoost training process may further enhance the model's accuracy, aligning the predicted results more closely with engineers' understanding.

Author Contributions: Conceptualization, H.S. and Q.L.; methodology, Z.X.; software, H.S.; validation, Z.X.; formal analysis, H.S.; investigation, Q.L.; resources, Z.X.; data curation, Y.L.; writing—original draft preparation, H.S.; writing—review and editing, Y.Y.; visualization, Q.L.; supervision, Y.L. and Y.Y.; project administration, Z.X. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Major Science and Technology Special Project of the China National Petroleum Corporation, "Research on Key Technologies for Efficient Production of Overseas Large-scale Carbonate Reservoirs" (Grant No. 2023ZZ19-04).

Data Availability Statement: Data are unavailable due to privacy restrictions.

Conflicts of Interest: Author Zhaohui Xia, Yunbo Li and Yang Yu were employed by the company Research Institute of Petroleum Exploration and Development. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. All authors have read and agreed to the published version of the manuscript.

Nomenclature

ANN	Artificial neural network
BHP	Bottomhole pressure of a well
GA	Genetic algorithm
GOR	Gas–oil ratio of a well
SGD	Stochastic gradient descent
SVR	Support vector regression
WGIR	Gas injection rate of a well
WGPR	Gas production rate of a well
WOPR	Oil production rate of a well
WWIR	Water injection rate of a well
WWCT	Water cut of a well
WWPR	Water production rate of a well
XGBoost	eXtreme Gradient Boosting

References

1. Duns, H.; Ros, N.C.J. *Vertical Flow of Gas and Liquid Mixtures in Wells*; OnePetro: Richardson, TX, USA, 1963.
2. Hagedorn, A.R.; Brown, K.E. Experimental Study of Pressure Gradients Occurring During Continuous Two-Phase Flow in Small-Diameter Vertical Conduits. *J. Pet. Technol.* **1965**, *17*, 475–484. [CrossRef]
3. Orkiszewski, J. Predicting Two-Phase Pressure Drops in Vertical Pipe. *J. Pet. Technol.* **1967**, *19*, 829–838. [CrossRef]
4. Beggs, D.H.; Brill, J.P. An Experimental Study of Two-Phase Flow in Inclined Pipes. *J. Pet. Technol.* **1973**, *25*, 607–617. [CrossRef]
5. Mukherjee, H.; Brill, J.P. Pressure Drop Correlations for Inclined Two-Phase Flow. *J. Energy Resour. Technol.* **1985**, *107*, 549–554. [CrossRef]
6. Ansari, A.M.; Sylvester, N.D.; Sarica, C.; Shoham, O.; Brill, J.P. A Comprehensive Mechanistic Model for Upward Two-Phase Flow in Wellbores. *SPE Prod. Facil.* **1994**, *9*, 143–151. [CrossRef]
7. Corsano, A. Experimental Study and the Development of a Mechanistic Model for Two-Phase Flow through Vertical Tubing. In Proceedings of the SPE Western Regional Meeting, Anchorage, AK, USA, 22–24 May 1996; SPE: Kuala Lumpur, Malaysia, 1996.

8. Gomez, L.E.; Shoham, O.; Schmidt, Z.; Chokshi, R.N.; Northug, T. Unified Mechanistic Model for Steady-State Two-Phase Flow: Horizontal to Vertical Upward Flow. *SPE J.* **2000**, *5*, 339–350. [CrossRef]
9. Wang, B. The Latest Application of Artificial Intelligence Technology in Petroleum Engineering Field. *China CIO News* **2018**, *10*, 95.
10. Zhao, X.; Chen, X.; Lan, Z.; Wang, X.; Yao, G. Pore Pressure Prediction Assisted by Machine Learning Models Combined with Interpretations: A Case Study of an HTHP Gas Field, Yinggehai Basin. *Geoenergy Sci. Eng.* **2023**, *229*, 212114. [CrossRef]
11. Awadalla, M.; Yousef, H. Neural Networks for Flow Bottom Hole Pressure Prediction. *Int. J. Electr. Comput. Eng. IJECE* **2016**, *6*, 1839. [CrossRef]
12. Firouzi, M.; Rathnayake, S. *Prediction of the Flowing Bottom-Hole Pressure Using Advanced Data Analytics*; OnePetro: Richardson, TX, USA, 2019.
13. Ahmadi, M.A.; Chen, Z. Machine Learning Models to Predict Bottom Hole Pressure in Multi-phase Flow in Vertical Oil Production Wells. *Can. J. Chem. Eng.* **2019**, *97*, 2928–2940. [CrossRef]
14. Nait Amar, M.; Zeraibi, N.; Redouane, K. Bottom Hole Pressure Estimation Using Hybridization Neural Networks and Grey Wolves Optimization. *Petroleum* **2018**, *4*, 419–429. [CrossRef]
15. Nait Amar, M.; Zeraibi, N. A Combined Support Vector Regression with Firefly Algorithm for Prediction of Bottom Hole Pressure. *SN Appl. Sci.* **2020**, *2*, 23. [CrossRef]
16. Rathnayake, S.; Rajora, A.; Firouzi, M. A Machine Learning-Based Predictive Model for Real-Time Monitoring of Flowing Bottom-Hole Pressure of Gas Wells. *Fuel* **2022**, *317*, 123524. [CrossRef]
17. Jia, D.; Liu, H.; Zhang, J.; Pei, X.; Wang, Q.; Yang, Q. Data-Driven Optimization for Fine Water Injection in a Mature Oil Field. *Pet. Explor. Dev.* **2020**, *47*, 629–636. [CrossRef]
18. Wang, H.; Mu, L.; Shi, F.; Dou, H. Production Prediction at Ultra-High Water Cut Stage via Recurrent Neural Network. *Pet. Explor. Dev.* **2020**, *47*, 1009–1015. [CrossRef]
19. Tariq, Z.; Mahmoud, M.; Abdulraheem, A. *An Artificial Intelligence Approach to Predict the Water Saturation in Carbonate Reservoir Rocks*; OnePetro: Richardson, TX, USA, 2019.
20. Artun, E.; Kulga, B. Selection of Candidate Wells for Re-Fracturing in Tight Gas Sand Reservoirs Using Fuzzy Inference. *Pet. Explor. Dev.* **2020**, *47*, 383–389. [CrossRef]
21. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13 August 2016; pp. 785–794.
22. Ogunleye, A.; Wang, Q.-G. XGBoost Model for Chronic Kidney Disease Diagnosis. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *17*, 2131–2140. [CrossRef]
23. Dhaliwal, S.S.; Nahid, A.-A.; Abbas, R. Effective Intrusion Detection System Using XGBoost. *Information* **2018**, *9*, 149. [CrossRef]
24. Pan, S.; Zheng, Z.; Guo, Z.; Luo, H. An Optimized XGBoost Method for Predicting Reservoir Porosity Using Petrophysical Logs. *J. Pet. Sci. Eng.* **2022**, *208*, 109520. [CrossRef]
25. Markovic, S.; Bryan, J.L.; Rezaee, R.; Turakhanov, A.; Cheremisin, A.; Kantzas, A.; Koroteev, D. Application of XGBoost Model for In-Situ Water Saturation Determination in Canadian Oil-Sands by LF-NMR and Density Data. *Sci. Rep.* **2022**, *12*, 13984. [CrossRef] [PubMed]
26. Zhong, R.; Johnson, R.; Chen, Z. Generating Pseudo Density Log from Drilling and Logging-While-Drilling Data Using Extreme Gradient Boosting (XGBoost). *Int. J. Coal Geol.* **2020**, *220*, 103416. [CrossRef]
27. Gu, Y.; Zhang, D.; Bao, Z. A New Data-Driven Predictor, PSO-XGBoost, Used for Permeability of Tight Sandstone Reservoirs: A Case Study of Member of Chang 4+5, Western Jiyuan Oilfield, Ordos Basin. *J. Pet. Sci. Eng.* **2021**, *199*, 108350. [CrossRef]
28. Al-Mudhafar, W.J.; Abbas, M.A.; Wood, D.A. Performance Evaluation of Boosting Machine Learning Algorithms for Lithofacies Classification in Heterogeneous Carbonate Reservoirs. *Mar. Pet. Geol.* **2022**, *145*, 105886. [CrossRef]
29. Zhang, J.; Sun, Y.; Shang, L.; Feng, Q.; Gong, L.; Wu, K. A Unified Intelligent Model for Estimating the (Gas + n-Alkane) Interfacial Tension Based on the eXtreme Gradient Boosting (XGBoost) Trees. *Fuel* **2020**, *282*, 118783. [CrossRef]
30. Dong, Y.; Qiu, L.; Lu, C.; Song, L.; Ding, Z.; Yu, Y.; Chen, G. A Data-Driven Model for Predicting Initial Productivity of Offshore Directional Well Based on the Physical Constrained eXtreme Gradient Boosting (XGBoost) Trees. *J. Pet. Sci. Eng.* **2022**, *211*, 110176. [CrossRef]
31. Wang, Z.; Tang, H.; Cai, H.; Hou, Y.; Shi, H.; Li, J.; Yang, T.; Feng, Y. Production Prediction and Main Controlling Factors in a Highly Heterogeneous Sandstone Reservoir: Analysis on the Basis of Machine Learning. *Energy Sci. Eng.* **2022**, *10*, 4674–4693. [CrossRef]
32. Zhai, L. XGBoost-Based Water Injection Profile Prediction Method and Its Application. *Pet. Geol. Recovery Effic.* **2022**, *29*, 175–180. [CrossRef]
33. Shi, J.; Xu, H.; Peng, L. Research on Complaint Management System of Manufacturing Industry Based on XGBoost. *Manuf. Autom.* **2023**, *45*, 76–79.
34. Chen, K.-Y. Forecasting Systems Reliability Based on Support Vector Regression with Genetic Algorithms. *Reliab. Eng. Syst. Saf.* **2007**, *92*, 423–432. [CrossRef]
35. Whitley, D. A Genetic Algorithm Tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [CrossRef]

36. Benesty, J.; Chen, J.; Huang, Y.; Cohen, I. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*; Cohen, I., Huang, Y., Chen, J., Benesty, J., Eds.; Springer Topics in Signal Processing; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–4, ISBN 978-3-642-00296-0.
37. Wold, S.; Esbensen, K.; Geladi, P. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
38. Rahimi, M.; Riahi, M.A. Reservoir Facies Classification Based on Random Forest and Geostatistics Methods in an Offshore Oilfield. *J. Appl. Geophys.* **2022**, *201*, 104640. [CrossRef]
39. Wang, G.; Ju, Y.; Carr, T.R.; Li, C.; Cheng, G. *Application of Artificial Intelligence on Black Shale Lithofacies Prediction in Marcellus Shale, Appalachian Basin*; OnePetro: Richardson, TX, USA, 2014.
40. Al-Mudhafar, W.J. *Incorporation of Bootstrapping and Cross-Validation for Efficient Multivariate Facies and Petrophysical Modeling*; OnePetro: Richardson, TX, USA, 2016.
41. Pirrone, M.; Battigelli, A.; Ruvo, L. *Lithofacies Classification of Thin Layered Reservoirs through the Integration of Core Data and Dielectric Dispersion Log Measurements*; OnePetro: Richardson, TX, USA, 2014.
42. Marfo, S.A.; Asante-Okyere, S.; Ziggah, Y.Y. A new flowing bottom hole pressure prediction model using M5 prime decision tree approach. *Model. Earth Syst. Environ.* **2022**, *8*, 2065–2073. [CrossRef]
43. Sun, P.; Huo, S.; He, T. Multiple machine learning models in estimating viscosity of crude oil: Comparisons and optimization for reservoir simulation. *J. Mol. Liq.* **2023**, *384*, 122251. [CrossRef]
44. Fan, P.; Deng, R.; Qiu, J.; Zhao, Z.; Wu, S. Well logging curve reconstruction based on kernel ridge regression. *Arab. J. Geosci.* **2021**, *14*, 1559. [CrossRef]
45. Üneş, F.; Demirci, M.; Taşar, B.; Kaya, Y.Z.; Varçin, H. Modeling of dam reservoir volume using generalized regression neural network, support vector machines and M5 decision tree models. *Appl. Ecol. Environ. Res.* **2019**, *17*, 7043–7055. [CrossRef]
46. Wang, M.; Feng, D.; Li, D.; Wang, J. Reservoir Parameter Prediction Based on the Neural Random Forest Model. *Front. Earth Sci.* **2022**, *10*, 888933. [CrossRef]
47. Qin, Y.; Ye, Z.; Zhang, C. Application of GBDT for division of petroleum reservoirs. *J. Phys. Conf. Ser.* **2020**, *1437*, 012050. [CrossRef]
48. Seyyedattar, M.; Zendehboudi, S.; Butt, S. Relative permeability modeling using extra trees, ANFIS, and hybrid LSSVM–CSA methods. *Nat. Resour. Res.* **2022**, *31*, 571–600. [CrossRef]
49. Busari, G.A.; Lim, D.H. Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance. *Comput. Chem. Eng.* **2021**, *155*, 107513. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.