A stacked generalization ensemble model for optimization and prediction of the gas well rate of penetration: a case study in Xinjiang

Naipeng Liu1,2,3 · Hui Gao4 · Zhen Zhao5 · Yule Hu4 · Longchen Duan4

Received: 26 August 2021 / Accepted: 25 November 2021 / Published online: 14 December 2021
© The Author(s) 2021

Abstract
In gas drilling operations, the rate of penetration (ROP) parameter has an important influence on drilling costs. Prediction of ROP can optimize the drilling operational parameters and reduce its overall cost. To predict ROP with satisfactory precision, a stacked generalization ensemble model is developed in this paper. Drilling data were collected from a shale gas survey well in Xinjiang, northwestern China. First, Pearson correlation analysis is used for feature selection. Then, a Savitzky-Golay smoothing filter is used to reduce noise in the dataset. In the next stage, we propose a stacked generalization ensemble model that combines six machine learning models: support vector regression (SVR), extremely randomized trees (ET), random forest (RF), gradient boosting machine (GB), light gradient boosting machine (LightGBM) and extreme gradient boosting (XGB). The stacked model generates meta-data from the five models (SVR, ET, RF, GB, LightGBM) to compute ROP predictions using an XGB model. Then, the leave-one-out method is used to verify modeling performance. The performance of the stacked model is better than each single model, with \( R^2 = 0.9568 \) and root mean square error = 0.4853 m/h achieved on the testing dataset. Hence, the proposed approach will be useful in optimizing gas drilling. Finally, the particle swarm optimization (PSO) algorithm is used to optimize the relevant ROP parameters.

Keywords Rate of penetration · Stacked model · Optimization · Machine learning

Introduction
Drilling optimization is important, as it can reduce the overall costs of drilling. Increasing the rate of penetration (ROP) is one optimization method. The ROP is affected by various interconnected factors (operational parameters, drill bit characteristics, and formation properties). The prediction of ROP is difficult, and many scholars have spent a lot of effort researching ROP models.

In the early years, some physical and mathematical models were established to predict ROP (Maurer 1962; Bingham 1965; Bourgoyne and Young 1974; Warren 1987; Hareland and Rampersad 1994; Motahhari 2008; Motahhari et al. 2010). The Bourgoyne and Young model considers a variety of influencing factors—mechanical parameters, hydraulic parameters, and formation parameters—and has been widely used in the prediction of ROP (Bahari and Baradaran Seyed 2007; Bahari et al. 2009; Hua 2010; Rahimzadeh et al. 2011; Nascimento et al. 2015; Ahmed and Ibrahim 2019). Soares et al. (2016) compared Hareland and Rampersad’s model (Hareland and Rampersad 1994) and Motahhari’s model...
The factors influencing ROP are complex, and physical models have difficulty in integrating them all to accurately predict the ROP. In recent years, machine learning techniques have developed rapidly and are widely used to predict ROP. Hegde et al. (2017) showed that data-driven models were more accurate than physical models (Bingham 1965; Hareland and Rampersad 1994; Motahhari et al. 2010) in ROP prediction in every formation. These data-driven models include linear regression, random forest (RF) (Breiman 2001), and ensemble models (Hegde 2016).

Artificial neural networks (ANNs) are widely used in ROP prediction and have good performance (Arabjamaloei and Shadizadeh 2011; Arabjamaloei et al. 2011; Arabjamaloei and Karimi Dehkordi 2012; Kahraman 2016; Bezminabadi et al. 2017; Anemangely et al. 2018; Elkatatny 2018; Abbas et al. 2019b, a; Sabah et al. 2019; Ashrafi et al. 2019; Diaz et al. 2019; Elkatatny et al. 2020; Zhao et al. 2020; Qian et al. 2021). To improve the performance of ANNs, some algorithms are used to optimize ANN models. Basarir et al. (2014) used an adaptive neuro-fuzzy inference system to predict ROP. Shi et al. (2016) used a typical extreme learning machine and an efficient learning model called upper-layer-solution-aware to predict ROP. Eskandarian et al. (2017) combined RF and monotone multi-layer perceptron models to predict ROP. Anemangely et al. (2018) used a hybrid model composed of a multi-layer perceptron neural network (MLP) together with either a particle swarm optimization (PSO) algorithm or a cuckoo optimization algorithm to predict ROP. Ashrafi et al. (2019) developed and trained eight hybrid ANNs using four evolutionary algorithms: a genetic algorithm, PSO, a biogeography-based optimizer and the imperialist competitive algorithm. Gan et al. (2019a) proposed a novel two-level method that contains a formation drillability fusion submodel established by using the Nadaboost extreme learning machine algorithm and an ROP model established by a neural network with a radial basis function. Elkatatny (2019) developed a new ROP model using an ANN combined with the self-adaptive differential evaluation technique.

Several other machine learning methods, such as support vector regression (SVR) (Vapnik 1995) and ensemble methods such as RF (Breiman 2001) and gradient boosting machines (GB) (Friedman 2001) have also been applied to the prediction of ROP. Bodaghi et al. (2015) proposed an SVR model of ROP that was optimized by a cuckoo search algorithm and genetic algorithm. Ansari et al. (2017) proposed a committee support vector regression improved by an imperialist competitive algorithm. Gan et al. (2019b) proposed a support vector regression with iterative local search and a stochastic inertia weight bat algorithm method. Ahmed et al. (2019) compared four computational intelligent techniques: ANN, ELM, SVR and least-squares SVR. The results show that all four computational intelligent techniques had acceptable accuracy, with the best model being the least-squares SVR. Hegde et al. (2015) used trees, bagging and RF to predict the ROP during drilling. Mantha and Samuel (2016) presented an algorithm that can choose the best of four models—neural networks, SVR, RF or GBM—for use in different strata. They can be effectively employed independently of location or formation. Hegde and Gray (2017) increased ROP by changing surface parameters on a rig that used RF algorithms. Hegde and Gray (2018) built ROP, torque on bit and mechanical specific energy models using a data-driven approach with an RF algorithm. Soares and Gray (2019) studied the real-time predictive capabilities of analytics and machine learning ROP models in a continuous learning environment. It was found that by shortening the retraining interval (defined by the length or number of data points), the performance of analysis models and ML models can be improved.

The drilling model above is suitable for the well-studied in their paper. However, a single-ROP model may not be effective for other wells, as it may fall into a local optimal value. Therefore, to improve the accuracy and generalization of the ROP predictive model, this paper uses a stacked generalization ensemble model. This approach combines six models: SVR, extremely randomized trees (ET), RF, light gradient boosting machine (LightGBM), GB and extreme gradient boosting (XGB). The stacked model includes a two-layer structure. The first layer generates meta-data from the SVR, ET, RF, LightGBM and GB models, and the second layer uses the XGB model to make the final prediction. Then, the PSO algorithm is used to optimize the drilling parameters that are effective influences on the ROP.

**Data collection**

The data collected from a shale gas survey well drilled in Xinjiang, northwestern China. Table 1 shows the drilling rig model and main auxiliary equipment. The purpose of this well was to determine the shale’s stratigraphic sequence, thickness, structure and organic geochemical characteristics, and to evaluate its storage performance, rock mechanical parameters and gas-bearing properties.

**Feature selection**

The database consisted of 2383 data points covering 2369 m of drilling depths (10–2379 m). It included the parameters of ROP, depth, weight of bit (WOB), stand pip pressure...
(SPP), rotary speed (RPM), mud weight (MW), temperature (T), flow rate (Q), bit diameter, equivalent circulating densities (ECD), funnel viscosity (FV), solid content, filter loss (FL), formation pressure gradient, and porosity. As redundant features in the data will affect the performance of the model, some features with lower correlations need to be excluded. Figure 1 shows the values of the Pearson correlations between feature variables. The Pearson correlations

| Table 1 | Drilling rig model and main auxiliary equipment |
|---------|-----------------------------------------------|
| Equipment | Model          | Load(kN) | Power(kW) | Remarks |
| Rig      | ZJ30/1700 J   | 1700     | –         | –       |
| Derrick  | JJ180/38      | 1800     | –         | –       |
| Mud pump | SL3NB1300     | –        | 960       | 2 sets  |
| Drilling fluid tank | TZSG-100 | –        | –         | 100m³   |
| Power system | GV12V190PZL-3 | –         | 1000      | 3 sets  |
| Solids control system | Vibrating screen, desilter, desander, centrifuge | 1 set |

Fig. 1 Pearson correlations between feature variables

| Table 2 | Overview of drilling data |
|---------|----------------------------|
| Statistic | Depth (m) | ROP (m/h) | WOB (kN) | SPP (MPa) | RPM (r/min) | MW (g/cm³) | T (℃) | Q (m³/min) | Bit diameter (mm) | ECD (g/cm³) | Porosity | Solid content (%) | FL (mL) |
| Count    | 2383      | 2383      | 2383      | 2383      | 2383        | 2383       | 2383   | 2383       | 2383              | 2383       | 2383     | 2383            | 2383    |
| Mean     | 1200.53   | 2.87      | 57.80     | 5.29      | 1.15        | 30.00      | 1.98   | 314.27     | 1.16              | 1.16       | 1.53     | 0.25             | 4.11    |
| Standard deviation | 687.41    | 2.62      | 52.18     | 2.19      | 0.07        | 2.93       | 0.29   | 69.42      | 0.07              | 0.07       | 0.53     | 0.06             | 0.98    |
| Minimum  | 10        | 0.26      | 0         | 0         | 30          | 1.02       | 20     | 0.647      | 216               | 1          | 1        | 0.1              | 3.3     |
| Median   | 1201      | 2.07      | 48.2      | 5.4       | 35          | 1.14       | 30     | 1.878      | 311.1             | 1.15       | 1.31     | 0.2              | 3.7     |
| Maximum  | 2389      | 28.3      | 202.3     | 11.5      | 120         | 1.3        | 36     | 2.807      | 444.5             | 1.31       | 12.83    | 0.4              | 7       |
of formation pressure gradient and FV with ROP are very low, so these two features were excluded. An overview of the drilling data is shown in Table 2.

**Noise reduction**

Data measured is always affected by noise. Even in the best-controlled drilling conditions, errors in measurements are at least approximately 5% (Orr 1998; Redman 1998). Noisy data can affect the machine learning process, increase learning time and reduce performance (Garcia et al. 2015). In this research, we used the SG smoothing filter (Savitzky and Golay 1964). The SG filter can smooth a signal without degrading its original properties too much. It is based on the least-squares principle of the polynomial smoothing algorithm. The SG technique is widely used in drilling data preprocessing (Anemangely et al. 2018; Ashrafi et al. 2019; Sabah et al. 2019). According to the characteristics of the data, we filtered and smoothed the following data: ROP, WOB, SPP, MW, temperature, Q, ECD and porosity. Figure 2 compares the data measured in the well (blue) with the data that was denoised using the SG filter (red). It can be seen from the figure that the SG filter has removed the abnormal points and smoothed the curve.

**Data normalization**

Data normalization is important in machine learning. The numerical magnitudes of each drilling variable are different; therefore, if the data is not normalized, the training time will be longer and the performance of the data-driven model will be poor. It is necessary to normalize raw drilling data to eliminate this effect on the results of the machine learning algorithm. After the raw data have been normalized, each indicator has the same order of magnitude, making it suitable for comprehensive and comparative evaluations. In this paper, we used min–max normalization to normalize the raw drilling data (Sun et al. 2019), as shown in (1).

\[
\hat{x}_i = \frac{x_i - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{1}
\]

where \(x_{\text{max}}\) is the maximum value and \(x_{\text{min}}\) is the minimum value. After normalization, all data are in the interval [0, 1].

**Method**

In this section, six machine learning models are introduced according to their particular architecture: SVR, RF, ET, GB, LightGBM and XGB. Moreover, the proposed technique—the stacked generalization ensemble model—is comprehensively investigated. Then, the PSO algorithm is introduced to optimize the drilling parameters.

**Support vector regression**

The support vector machine (SVM) approach was developed by Vapnik and collaborators at Bell Laboratories (Vapnik and Chervonenkis 1964; Vapnik 1995). The SVM principle is based on statistical learning theory and structural minimization (Bello et al. 2016). The SVR model can be defined as the following (Fletcher 2013):

\[
f(x) = \sum_{i=1}^{M} (\alpha_i - \alpha_i^*) K(x_i, x) + b \tag{2}
\]

where \(\alpha_i\) and \(\alpha_i^*\) are Lagrange multipliers and \(K(x_i, x)\) is kernel function. The kernel function can transform low-dimensional nonlinear problems to high-dimensional programming linear problems. There are different kernels (linear kernel, polynomial kernel, radial basis function kernel, etc.) for performing tasks in high-dimensional feature spaces (Zhong et al. 2019).

**Random forest**

The RF is an extended variant of bagging (Bheiman 1996). Bagging based on bootstrap sampling is the most famous parallel integrated learning methods (Efron and Tibshirani 1993). Bagging is a combination of bootstrapping and aggregating. Bootstrap sampling was performed on a data set of \(N\) samples to obtain dataset \(D_1\). The training model was repeated on \(D_1\) for \(M\) times to obtain \(M\) models, then the variance of the model could be reduced by averaging.

The base learner of RF is a decision tree. The selection of random attributes is further increased in the training process of the decision tree. RF is simple, easy to implement, has low computational overhead, and has powerful performance in many tasks. It is known as the method that represents integrated learning technology. The diversity of basic learners in RF not only comes from sample interference but also attribute interference. So the performance can be improved by increasing the divergence between each learner (Zhou 2016).

**Extremely randomized trees**

Also known as extra-trees (ET; Geurts et al. 2006), extremely randomized trees are tree-based randomization ensembles that combine the attribute randomization of a random subspace with a totally random selection of the cut-point. Compared to RF, it splits nodes by completely randomly selecting tangent points, and uses the entire learning sample to grow the tree. It can reduce the variance while slightly increasing the bias at the same time.
Fig. 2 Measured and denoised data from the drilled well
Gradient boosting machine

The GB (Friedman 2001) is a model with iterative stage addition that can be regarded as minimizing the steepest descent of a given loss function. GB is another ensemble method that can build a more powerful model by merging multiple decision trees. Unlike the RF method, GB uses a continuous method to construct trees, where each tree can reduce the error of the previous tree. There is no randomization in a GB regression tree, but strong pre-pruning is used. GB trees usually use trees with small depths so that the model occupies less memory and the prediction speed is faster.

Light gradient boosting machine

LightGBM (Ke et al. 2017) is a GB comprising reduce data dimensionality adaptation algorithm, as well as speed up process system. LightGBM is optimized by using: A decision tree algorithm based on a histogram, a leaf-wise leaf growth strategy with depth limitation, acceleration of histogram difference, and direct support for categorical features. It supports high-efficiency parallel training and has the advantages of fast training speed, low memory consumption, good accuracy and distributed support, and can rapidly process massive datasets.

Extreme gradient boosting

XGB (Chen and Guestrin 2016) is designed to be a highly scalable and accurate tree-boosting system. XGB incorporates a set of decision trees to build a powerful regression model. This large-scale machine learning method can easily and automatically apply multi-threaded parallelism to shorten execution times. In contrast to the GB model, XGB uses second-order Taylor expansion of the loss function. Additionally, the depth of the tree and the weight of the leaf nodes are part of the XGB objective function. It can reduce the iteration process and enhance the performance of the tree. A step-by-step decision tree growth technique is implemented to reduce model complexity.

Stacked generalization ensemble model

Stacked generalization (Wolpert 1992; Breiman 1996; Wolpert and Macready 1996) is the most popular meta-learning algorithm (Wolpert and Macready 1996; Sill et al. 2009; Zhang et al. 2010; Dou et al. 2020). Algorithm 1 shows the pseudo-code of the stacked generalization ensemble algorithm. Stacked generalization refers to any scheme for feeding information from one set of generalizers to another before forming the final guess. Stacked generalization can reduce the deviation of the generalizer from the provided learning set. The stacked generalization error is comprised of a term that depends on the generalization error of the individual learners and another term that contains all the correlations between learners, and is defined as:

\[ E = \bar{E} - \bar{A} \]  

where \( E \) is the generalization errors of the individual learners, which depend on the errors of the individual learners \( E_i \) and the combined strategy algorithm; and \( \bar{A} \) is the ambiguities, which depend on the correlation between the individual learners \( A_i \) and the combined strategy algorithm (Krogh and Vedelsby 1995). From (3), it is obvious that increasing the ambiguity and decreasing individual generalization errors will improve the overall generalization. The errors of individual learners are small and the correlation between them is low, so the stacked model will be more accurate than individual learners. If a good algorithm is used to combine different individual learners to minimize the generalization error of the individual learners and maximize the ambiguities, then we can obtain the best stacked model.
The stacked algorithm structure of this article, shown in Fig. 3, uses the original dataset to train the primary learner. To avoid overfitting, cross-validation is used to train the primary learner. Then, the output of the primary learner is used as the new input features and the corresponding original tag is used as the new tag. In the first level, there are five learners: RF, ET, SVR, LightGBM and GB. First, fivefold cross-validation is used to train the training set, which generates new training data from each fold and then synthesizes a new training dataset from each single model. The new test dataset is the average of each fold model’s value, which is predicted by the original test dataset. In the second level, the new training dataset is trained by XGB and then the final predicted value is obtained.

**Algorithm 1: Stacked**

**Input:** Training dataset $D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$; First learning algorithm $L_1, L_2, ..., L_T$; Second learning algorithm $L$.

1: for $t = 1, 2, ..., T$ do
2: \[ h_t = L_t(D); \]
3: end for
4: $D' = \emptyset$
5: for $i = 1, 2, ..., m$ do
6: \[ \text{for } t = 1, 2, ..., T \text{ do} \]
7: \[ z_{it} = h_t(x_i); \]
8: \[ \text{end for} \]
9: \[ D' = D' \cup \{(z_{1i}, z_{2i}, ..., z_{Ti}), y_i\}; \]
10: end for
11: $h' = L(D')$

**Output:** $H(x) = h'(h_1(x), h_2(x), ..., h_T(x))$

The stacked algorithm structure of this article, shown in Fig. 3, uses the original dataset to train the primary learner. To avoid overfitting, cross-validation is used to train the primary learner. Then, the output of the primary learner is used as the new input features and the corresponding original tag is used as the new tag. In the first level, there are five learners: RF, ET, SVR, LightGBM and GB. First, fivefold cross-validation is used to train the training set, which generates new training data from each fold and then synthesizes a new training dataset from each single model. The new test dataset is the average of each fold model’s value, which is predicted by the original test dataset. In the second level, the new training dataset is trained by XGB and then the final predicted value is obtained.

**Particle swarm optimization**

After the drilling rate model is built, we use algorithms to optimize the effective parameters and obtain the optimal ROP. In the oil and gas industry, many algorithms are used for difficult optimization problems, including gene expression programming, genetic programming, PSO, cuckoo optimization algorithm, biogeography-based optimizer, and imperialist competitive algorithm, social spider optimization, sine cosine algorithm, multi-verse optimization and moth flame optimization (Bodaghi et al. 2015; Hajirezaie et al. 2015, 2017a, b; Anemangely et al. 2018; Ashrafi et al. 2019; Sabah et al. 2019; Zhou et al. 2021). PSO resembles a school of flying birds and is an extremely simple and effective algorithm (Kennedy and Eberhart 1995). It requires only simple mathematical operators and is computationally inexpensive in terms of both memory requirements and speed. Hence, we use PSO to optimize ROP. The process for implementing PSO is as in Algorithm 2. In PSO, the particles are placed in the search space of the function and each particle evaluates the objective function at its current position. These particles move via cooperation and competition between the particles themselves. The position of the moved particle is expressed by the following equation:

$$\begin{align*}
\vec{v}_i &\leftarrow w \cdot \vec{v}_i + c_1 \cdot rand_1() \cdot (\vec{p}_i - \vec{x}_i) + c_2 \cdot rand_2() \cdot (\vec{p}_g - \vec{x}_i) \\
\vec{x}_i &\leftarrow \vec{x}_i + \vec{v}_i
\end{align*}$$

(4)
where \( \vec{v}_i \) is the rate of positional change, \( w \) is the inertial weight, \( c_1 \) and \( c_2 \) are two positive constants, \( \text{rand}_1() \) and \( \text{rand}_2() \) are two random functions in the range \([0,1]\), \( \vec{p}_i \) is the previous personal best position, \( \vec{p}_g \) is the best position among all particles, and \( \vec{x}_i \) is the current position (Shi and Eberhart 1998). We then evaluate the objective function at its current position and update \( \vec{p}_i \) and \( \vec{p}_g \). We keep updating the particle position and evaluation repeatedly until the end condition is met. Eventually, the swarm as a whole, like a flock of birds collectively foraging for food, is likely to move close to an optimum of the fitness function (Poli et al. 2007).

**Algorithm 2: PSO**

1. Initialize a population array of particles with random positions and velocities in the search space.
2. Evaluate each particle and get the global optimum.
3. \textbf{loop}
4. Change the velocity and position of the particle
5. For each particle, evaluate the desired optimization fitness function
6. Update the optimal positions of particles and swarm
7. If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations), exit loop.
8. \textbf{end loop}

### Results and discussion

#### Model performance

We use the leave-one-out method to verify model performance. The whole dataset is split, with 80\% (1986 points) used as the training dataset and 20\% (477 points) used as the testing dataset. We use \( R^2 \), root mean square error (RMSE) and Mean Absolute Percentage Error (MAPE) to evaluate the performance of the model. The closer the \( R^2 \) score is to 1, the better the goodness of fit of the model. It is expressed as:

| Hyperparameters         | Search space                               | Best   | Training \( R^2 \) | Testing \( R^2 \) | Training RMSE (m/h) | Testing RMSE (m/h) |
|-------------------------|-------------------------------------------|--------|--------------------|-------------------|--------------------|-------------------|
| SVR                     | kernel {'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'} | 'rbf'  | 0.6070             | 0.5574            | 1.3031             | 1.5540            |
|                         | C \{0.1,1,10,100,1000,10,000\}             | 1000   |                    |                   |                    |                   |
| RF                      | n_estimators \{10,100,200,400,600,800\}   | 200    | 0.9853             | 0.8718            | 0.2521             | 0.8321            |
|                         | max_depth [3,30]                           | 20     |                    |                   |                    |                   |
|                         | min_samples_split [2,5]                   | 2      |                    |                   |                    |                   |
| ET                      | n_estimators \{10,100,200,300,400,600,800\} | 300    | 0.9997             | 0.9268            | 0.0338             | 0.6320            |
|                         | max_depth [3,30]                           | 18     |                    |                   |                    |                   |
|                         | min_samples_split [2,5]                   | 2      |                    |                   |                    |                   |
| GB                      | n_estimators \{10,100,200,400,600,800\}   | 800    | 0.9854             | 0.8663            | 0.2509             | 0.8541            |
|                         | max_depth [3,30]                           | 3      |                    |                   |                    |                   |
|                         | learning_rate \{0.1,0.01,0.001,0.0001\}   | 0.1    |                    |                   |                    |                   |
| LightGBM                | n_estimators \{10,100,200,400,600,800\}   | 200    | 0.9787             | 0.8777            | 0.3033             | 0.8168            |
|                         | max_depth [3,30]                           | 15     |                    |                   |                    |                   |
|                         | learning_rate \{0.1,0.01,0.001,0.0001\}   | 0.1    |                    |                   |                    |                   |
| XGB                     | n_estimators \{10,100,200,300,400,600,800\} | 300    | 0.9222             | 0.8059            | 0.5799             | 1.0292            |
|                         | gamma \{0.0,1,0.001,0.0001,0.00001\}       | 0.0001 |                    |                   |                    |                   |
| Stacked                 |                                           | 0.9879 | 0.9568             | 0.2284            | 0.4853             |
Fig. 4 Cross-plot of predicted denoised values versus ROP values in the training and testing datasets

\[ R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y_i})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} , \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \]  

where \( y_i \) is the predicted values, \( x_i \) is actual values. The RMSE is a measure of the spread of actual \( x \) values around the average of predicted \( y \) values. The smaller the RMSE value is, the higher the prediction accuracy. It is expressed as:

\[ \text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \hat{y_i})^2}{n}} \]

The machine learning algorithms were implemented using the scikit-learn library (Pedregosa et al. 2011). The tuning hyperparameters of these estimators were chosen using grid-search cross-validation of the training data. The performance and hyperparameters of each single model
Fig. 5 RMSE and $R^2$ values of the different models in the testing dataset

![Graph showing RMSE and $R^2$ values](image)

Fig. 6 Error of modeled versus denoised ROP values in the testing dataset

![Graph showing error of modeled versus denoised ROP values](image)

Fig. 7 Feature importance of ensemble models

(a) RF

(b) ET

(c) GB

(d) LightGBM

![Bar charts showing feature importance](image)
and stacked generalization model are shown in Table 3. The performance of the stacked generalization model is better than those of each single model. In the testing dataset, the $R^2$ value of the stacked generalization model is 0.9568, higher than that of the best single model, ET. Meanwhile, the RMSE of the stacked generalization model is 0.4853 m/h, lower than that of the best model, ET. Figure 4 presents a regression plot of the predicted values versus the denoised values for each single model and stacked generalization model. Figure 5 presents the RMSE and $R^2$ values of the different models in the testing dataset. Figure 6 presents the error of the predicted values versus the denoised values in the testing dataset. The error is expressed as:

$$\text{Error} = y_i - x_i$$

(7)

Obviously, the regression coefficient values of the stacked generalization model are highly desirable, indicating that this model can make good predictions of drilling ROP.

The combination of learners brings three benefits (Dietterich 2000). First of all, from a statistical perspective, there can be more than one hypothesis that get the equal performance at the training set due to the fact that the hypothesis space is large. If we misselect a single learner at this time, we may get a poor generalization performance. Combining multiple learners will reduce this risk. Second, from a computational point of view, the algorithm often fall into a local minimum with poor generalization performance. This problem can be reduced by stacked generalization of multiple learner combinations. Third, from the perspective of representation, the true hypothesis of some learning tasks may not be within the hypothesis space considered by the current learning algorithm. At this time, using a single learner is invalid. By combining multiple learners, it is possible to learn a better approximation due to the expansion of the corresponding hypothesis space.

Learning algorithms that have problems due to statistical factors usually show high variance, while problems caused by calculation factors result in high calculation variance, and representation factors cause high deviations. Therefore, by combining several methods, the learning algorithm can reduce the impacts of variance and bias at the same time (Zhou 2016). Figure 7 shows the importance of the features in the ensemble models, RF, ET, GB and LightGBM. It can be seen that the weight of each feature is different in each model. The top three import features are depth, porosity and FL in RF; depth, ECD and MW in ET; depth, porosity and ECD in GB; Q, WOB and MW in LightGBM. It indicates that the top three features which have greater influence on the prediction of target value of these models is different. The ambiguity between these learners is large and the accuracy of the individual model is high. Hence, the error can be reduced by combining these models. In this study, the XGB can combine different individual learners to

| Parameter | WOB (kN) | RPM (r/min) | Q (m$^3$/min) | Solid content (%) |
|-----------|----------|-------------|----------------|-------------------|
| Search space | [0, 202] | (30,40,60,80,120) | [0.647, 2.807] | [0.1, 0.4] |

Fig. 8 Optimum versus initial values
minimize the generalization error of the individual learners and maximize the ambiguities, so the stacked generalization ensemble model can improve the accuracy.

**Rate of penetration optimization**

The stacked generalization ensemble model obtained in the previous section is incorporated into the optimization algorithm to optimize the parameters that maximize the ROP. Since depth increases during drilling, it was not considered as a decision variable. It makes sense that the parameters cannot be optimized for each meter of penetration (Zhao et al. 2020). Therefore, we selected 23 points, one every 100 m, at which to optimize the ROP parameters to demonstrate the value of the proposed model. The selected parameters based on the formation and drilling design were T, bit diameter, and porosity. The critical parameters for downhole pressure and drilling safety are MW, ECD, and SPP, while FL is the key factor in protecting reservoirs and inhibiting water sensitivity. Therefore, MW, ECD, SPP, and FL cannot be changed arbitrarily. Hence, measured values of the parameters depth, T, bit diameter, porosity, MW, ECD, SPP, and FL are input into the model, while WOB, RPM, Q, and solid content are optimized via PSO to obtain the optimal ROP. A total of 20 particles are set; the first particle is set as the initial measured value of the drilling process, with other particles generated randomly. The search space of the optimized parameters is shown in Table 4. The optimization results are shown in Fig. 8; the ROP is better than the initial value, with the mean ROP increasing by 33.5% from 3.25 to 4.34 m/h.

**Conclusions**

This work presents a stacked generalization ensemble model for predicting the ROP while drilling. The model combines six efficient methods: SVR, RF, ET, GB, LightGBM and XGB. The stacked generalization ensemble model has two levels. In the first level, there are five learners: RF, ET, SVR, LightGBM and GB. New training data are generated through fivefold cross-validation and the original target value is used as the target value. In the second level, the new training dataset is trained by XGB. The ambiguity between these learners of first level is large and the accuracy of the individual model is high. The XGB can combine different individual learners to minimize the generalization error of the individual learners and maximize the ambiguities. The performance of the stacked model is better than each single model, the $R^2$ value is 0.9568 and the RMSE is 0.4853 m/h. The model provides a method for predicting drilling rates with high accuracy, which is beneficial to the optimization of ROP. By optimizing the effective drilling parameters via PSO at the selected 23 points, the mean ROP can be increased by 33.5%.

**Funding** This work was supported by the National Natural Science Foundation of China under Grant 61733016, No.1 Institute Geology and Mineral Resources of Shandong Province under Grant 2020WDW01, the Qinghai Province Key R&D and Transformation Program under Grant 2020-SF-149, the National Key R&D Program of China under Grant 2018YFC0603405.

**Declarations**

**Conflict of interest** On behalf of all the co-authors, the corresponding author states that there is no conflict of interest.

**Data availability** https://github.com/liunaipeng/rop/blob/main/qcrop.csv

**Code availability** https://github.com/liunaipeng/rop/blob/main/qcrop.ipynb

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

**References**

Abbas AK, Rushdi S, Alsaba M, Al Dushaishi MF (2019b) Drilling rate of penetration prediction of high-angled wells using artificial neural networks. J Energy Resour Technol Trans ASME 141:1–11. https://doi.org/10.1115/1.4043699

Abbas AK, Rushdi S, Alsaba M (2019a) Modeling rate of penetration for deviated wells using artificial neural network. In: Society of petroleum engineers - Abu Dhabi international petroleum exhibition and conference 2018, ADIPEC 2018

Ahmed AI, Ibrahim AA (2019) Bourgoyne and young model development review. Int J Eng Sci Res Technol 8:164–174

Ahmed OS, Adeniran AA, Samsuri A (2019) Computational intelligence based prediction of drilling rate of penetration: a comparative study. J Pet Sci Eng 172:1–12. https://doi.org/10.1016/j.petrol.2018.09.027

Al-Abduljabbar A, Elkatatny S, Mahmoud M et al (2019) A robust rate of penetration model for carbonate formation. J Energy Resour Technol Trans ASME 141:1–9. https://doi.org/10.1115/1.4041840

Anemangely M, Ramezanzadeh A, Tokhmechi B et al (2018) Drilling rate prediction from petrophysical logs and mud logging data using an optimized multilayer perceptron neural network. J Geophys Eng 15:1146–1159. https://doi.org/10.1088/1742-2140/aaac5d
