Earth Science Informatics (2023) 16:2285–2305
https://doi.org/10.1007/s12145-023-01014-7

RESEARCH

A semi-supervised coarse-to-fine approach with bayesian optimization for lithology identification

Yunxin Xie[1] · Liangyu Jin[1] · Chenyang Zhu[2] · Siyu Wu[1]

Received: 29 January 2023 / Accepted: 12 April 2023 / Published online: 7 June 2023

Abstract

Lithology identification is critical in the interpretation of well-logging data for petroleum exploration and development. However, the limited availability of labeled well-logging data for machine learning model training can lead to compromised accuracy in lithology classification models. Here, we propose a semi-supervised lithology identification model to overcome this challenge. Our framework consists of Bayesian optimization for tuning ensemble algorithms, including random forest, gradient boosting decision tree, extremely randomized trees, and adaptive boosting, to establish a high-quality baseline model for semi-supervised learning. We also employ a self-training strategy to increase the number of labeled samples in the training set and use the predicted label with the highest confidence as a pseudo-label to reduce the accumulation of deviation caused by incorrect pseudo-labels. Our semi-supervised coarse-to-fine framework improves rock classification accuracy, particularly for sandstone. Testing our model on well-logging data from two real regions, we found that the ExtraRF-based semi-supervised model in the HGF area performs the best, with a maximum classification accuracy of 91.6%, which is 5% higher than the original coarse-to-fine model without using Bayesian optimization and pseudo-labeling techniques.

Keywords Lithology classification · Ensemble methods · Semi-supervised learning · Bayesian optimization · Pseudo labels

Introduction

Lithology identification is a fundamental step in the interpretation of well-logs, serving as a basis for reservoir and basin evaluation. In petroleum exploration, well-logging is the predominant method used for lithology identification. This approach utilizes logging data to derive geological information, enabling the determination of petroleum and natural gas reserves and the formulation of exploration plans for oil and gas. Traditional techniques employed to identify lithology involve cross-mapping, statistical methods, and imaging logging, which entail manual estimation of rock type. However, such methods are known to suffer from low efficiency, high cost, and significant dependence on the expertise of evaluators, leading to subjective results and hindering accuracy improvement. Therefore, there exists an urgent need to develop cost-effective and objective methodologies for lithology identification.

In recent years, the utilization of machine learning techniques for identifying lithology from logging data has become ubiquitous. Multiple research studies have demonstrated that supervised machine learning algorithms, such as neural networks, support vector machines, and ensemble methods, have proven to be beneficial for multi-class lithology identification. The deep learning models' characteristics make them an ideal tool for the lithology identification of rock images with high-dimensional feature spaces. Jiang et al 2021 proposed geological constraints Imamverdiyev and Sukhostat 2019 as features and then used recurrent neural network (RNN) to train the lithology identification

Communicated by: H. Babaie

Yunxin Xie, Liangyu Jin , Siyu Wu contributed to this work equally.

✉ Chenyang Zhu
cz4g16@soton.ac.uk

Yunxin Xie
xieyx@cczu.edu.cn

Liangyu Jin
s21040820014@smail.cczu.edu.cn

Siyu Wu
s21040857011@smail.cczu.edu.cn

[1] School of Petroleum and Natural Gas Engineering, Changzhou University, Changzhou, Jiangsu 213000, China

[2] School of Electronics and Computer Science, University of Southampton, Southampton, Southampton SO17 1BJ, UK

model. Liu et al (2021) introduced an image-based 3D-CNN lithology identification model by combining hyperspectral remote sensing images with the 1D-CNN model proposed by Imamverdiyev. Imamverdiyev and Sukhostat (2019). Li et al (2022) processed rock images with data enhancement techniques and trained them with convolutional neural network (CNN) models, which resulted in better performance than the Fast R-CNN model without data processing (Xu et al, 2021).

In addition, for low-dimensional logging data, ensemble algorithms are deemed more appropriate. Xie et al (2018), and Sun et al (2019) compared ensemble algorithms, artificial neural networks, and traditional supervised learning models. The results show that the performance of lithology identification of the ensemble algorithm is better than that of the artificial neural network when the feature space is limited. Furthermore, Xie et al (2019) optimized Boosting algorithms of ensemble methods based on the logging data, resulting in improved lithology identification accuracy, but the identification of sandstones is still challenging. According to this challenge, (Xie et al, 2020) proposed a coarse-to-fine lithology identification supervised learning framework for sandstone, which combined outlier detection with ensemble algorithms to improve the accuracy of sandstone identification.

Despite the usefulness of supervised learning algorithms in building accurate classification models for lithology identification, they require a vast amount of labeled data for model training, which is laborious and time-consuming. Semi-supervised learning has emerged as a promising approach to address the challenge of acquiring a large volume of labeled training data. By utilizing both labeled and unlabeled data, semi-supervised learning algorithms can achieve outstanding performance in building classifiers with limited labeled data. Consistency regularization, self-training methods, and entropy minimization are the most commonly used semi-supervised learning approaches in practical applications Ouali et al (2020); Kim2021 (2021).

Researchers have recently applied semi-supervised learning to lithology identification with deep learning and machine learning. (Zhou et al, 2021) utilized self-training of semi-supervised learning combined with cross-domain transfer learning to solve the problem of lithology identification under the logging data distribution discrepancy. (Li et al, 2019) proposed a GAN model with a semi-supervised learning approach, namely the SGAN-G model. The proposed model achieves significant results based on logging curves. However, the training processes require 2,000 labeled data and 900,000 unlabeled data. It is not effortless to collect such a large scale of experimental data. In addition, this method only specifically identifies shales and sandstones, which has limited practicality. Li et al (2023) proposed a CE-SGAN model based on pseudo-labeling and the GAN algorithm. The semi-supervised model test in the DGF-HGF areas and Hugoton-Panoma fields, and the results only reached 88.68% and 68.83%, respectively, which results limited by the number of training logging data. Deep learning is a popular way to realize intelligent classification, but the number and dimensions of training data restrict them. According to the case of the small number of training data and low feature dimension, the ensemble algorithms are more suitable Xie et al (2018).

Active learning incorporation with semi-supervised learning is widely applied to lithology identification, assuming the same distribution of logging data. (Ren et al, 2023) improved the naive Bayes model as the baseline for generating pseudo labels. In the iterative semi-supervised learning process, active learning enhanced the baseline model's performance to improve the quality of the pseudo labels. The proposed method achieves excellent results in experimental areas. (Hong et al, 2022) aimed to reduce the expert cost of labeling core data. Active learning experts label the experimental data and combine it with semi-supervised learning. Such a framework saves many expert costs while ensuring accuracy. The supervised learning model is commonly used as a baseline to realize the semi-supervised learning framework. Li et al 2020 proposed a semi-supervised learning algorithm LapSVM by optimizing the SVM baseline model with the data regularization term. This method selected pseudo-labels of unlabeled data based on feature similarity, strengthening the classification model's ability. However, despite these advances, these semi-supervised learning approaches aim to strengthen the data and ignore the optimization of the baseline model. Ouali et al 2020 showed that the high-quality supervised baseline model is essential for the accuracy of semi-supervised learning.

The performance of a model heavily relies on the selection of its parameters. Therefore, it is necessary to utilize parameter optimization algorithms to achieve high-quality baseline models. Local optimization techniques such as grid search and random search are commonly used for parameter optimization. The grid search algorithm has commonly been adopted to tune the rock classification model Xie et al (2018, 2019); Zou et al (2021), but this method relies on brute force. It traverses all parameters group of the parameter space to optimize, which is highly costly in the calculation process (Liashchynskyi2019, 2019). On the other hand, random search randomly selects parameters from the parameter space, making it faster but less effective in achieving the best results Liashchynskyi2019 2012, 2019. Liashchynskyi2019 2019 compared grid search, random search, and genetic algorithm. The result showed that the genetic method is more appropriate for tuning too many parameters in the ample search space. However, local optimization techniques limit the probability of finding better quality parameters, and global optimization algorithms are necessary. Commonly used global optimization methods include particle swarm optimization (PSO), genetic algorithm (GA), differential evolution optimization, and Bayesian optimization. These methods have different applicable scenarios. Ren et al 2023 applied the PSO strategy to enhance the performance of the Fuzzy ID3 rock classification model and achieve high accuracy. PSO and GA have similar principles and are suitable for large search spaces to avoid getting trapped into local optima, but they are time-consuming Hassan et al 2005. Therefore, to guarantee optimal global parameters more efficiently, Bayesian optimization is chosen to update the model parameters. Sun et al (2020) proposed the Bayesian parameter optimization method to tune the parameters of the gradient promotion algorithm and compared it with the differential evolution optimization method Saporetti et al (2019). The results showed that Bayesian optimization can optimize the model parameters faster and improve the classification accuracy.

In this paper, we present a semi-supervised coarse-to-fine framework for lithology classification using Bayesian optimization, addressing the issue of accurately identifying multiple lithologies, particularly sandstone, with limited labeled training data. Our contributions focus on two key aspects. Firstly, we apply Bayesian optimization to tune the ensemble algorithm's parameters more efficiently, improving the baseline model's quality and reducing the pseudo-label error rate. Secondly, we employ self-training's pseudo-labels to expand the training set, aiming to enhance the coarse-to-fine lithology identification model's accuracy. We apply the semi-supervised framework to two actual areas of the Daniudi Gas Field (DGF) and the Hangjinqi Gas Field (HGF) and compare four ensemble methods, namely RF, ExtraRF, GBDT, and AdaBoost. Results demonstrate that Bayesian optimization and pseudo-labels can improve the lithology identification prediction accuracy. Furthermore, the semi-supervised framework, based on a Bayesian optimized extremely random tree model, achieves the best performance.

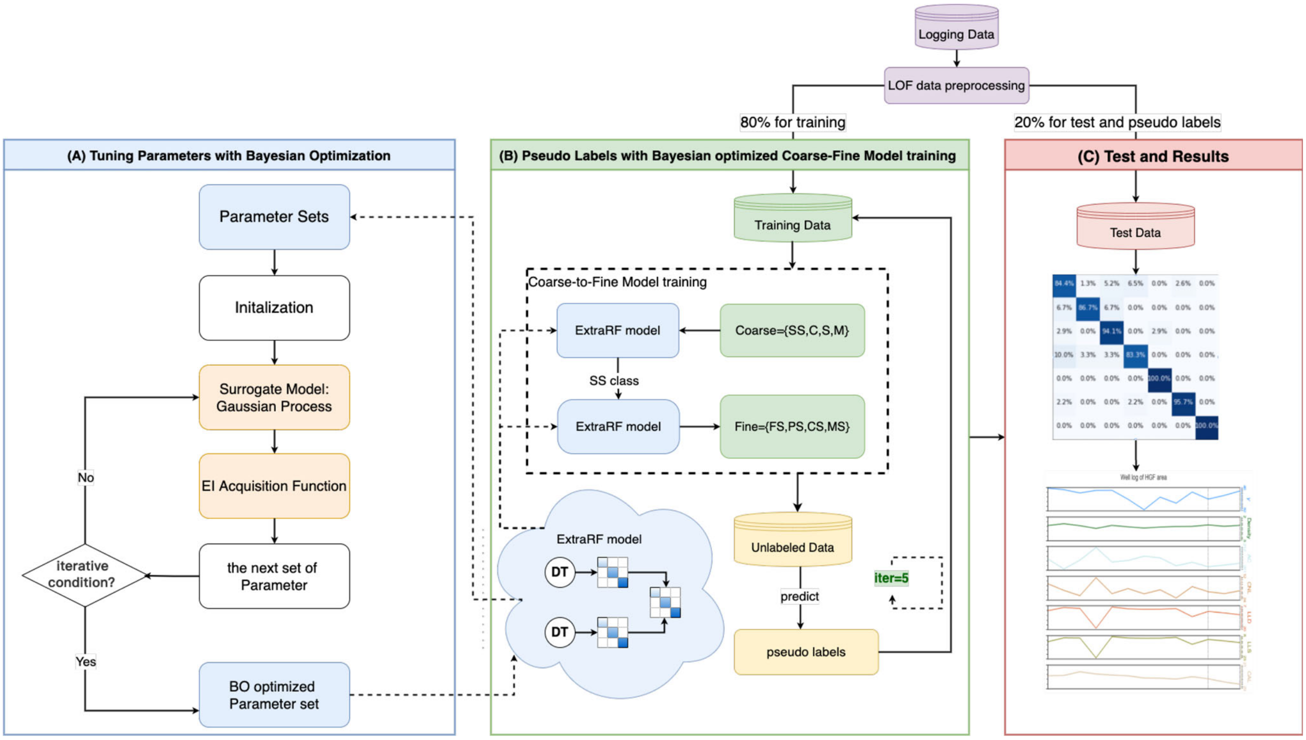
Methodology

Framework overview

This paper proposed a semi-supervised framework based on a Bayesian-optimized coarse-to-fine approach for lithology identification. The workflow of the semi-supervised framework is displayed in Fig. 1. Logging data were collected from various wells within the same region and preprocessed using the LOF algorithm to eliminate outliers. The processed logging data were then divided into training (80%) and test (20%) samples, as shown in parts (B) and (C) of Fig. 1, respectively. In this study, each sample was represented by a feature vector X_i consisting of well-logging measurements at the corresponding depth. We also assigned a label y_i to each sample to indicate its rock type. The y_i label was comprised of two components: a general rock class from Y_{coarse} and a fine sandstone class from Y_{fine}. The Y_{coarse} class was defined as coarse labels, which included sandstone (SS), carbonate (CR), coal (C), siltstone (S), and mudstone (M). Furthermore, the sandstone (SS) class was subdivided into fine sandstone (FS), medium sandstone (MS), coarse sandstone (CS), and pebbled sandstone (PS). The fine sandstone (FS), medium sandstone (MS), coarse sandstone (CS), and pebbled sandstone (PS) were defined as fine labels from the Y_{fine} class.

In order to enhance the accuracy of the baseline model, it is essential to optimize the parameter sets of the ensemble method, such as the ExtraRF algorithm, prior to training. Part (A) of Fig. 1 shows the workflow of the Bayesian optimization algorithm. Bayesian optimization is a global optimization scheme based on Bayes' theorem, which is suitable for optimizing black-box functions with complex or unknown objective functions. The Bayesian optimization mainly includes two core parts: the surrogate model and the acquisition function. Firstly, the ensemble algorithm is replaced with the surrogate model and the prior distribution of the surrogate model is initialized. In the second step, the sample point to be collected is selected based on the acquisition function, and the surrogate is updated by evaluating the sample points to obtain the posterior distribution, which is repeated until the maximum number of iterations is reached. Finally, the Bayesian-optimized ExtraRF algorithm is delivered to part (B) of Fig. 1 for semi-supervised coarse-to-fine framework training.

Part (B) of Fig. 1 illustrates the training process of a semi-supervised coarse-to-fine model that involves labeled and unlabeled data sets. In each iteration, the Bayesian-optimized coarse-to-fine model is trained on the labeled data and used as the initial model. The initial model is then used to predict the labels of the unlabeled data, and the prediction results with high confidence are selected as pseudo-labels. To select the predicted labels with high confidence, Ouali et al (2020) pseudo-label selection rule is followed, which calculates the actual probability of each predicted label and selects the top 10 results with a predicted probability above 95%. The training process of the coarse-to-fine model is divided into two parts. In the first part, the coarse model is trained, which maps the labeled training data with fine labels to coarse labels. The ensemble model is then trained with the coarse-labeled data. In the second part, only the fine-labeled data is used to train the ensemble model for the fine model's training. Finally, the trained model is tested with the test data, and the partial results of the logging curves and the confusion matrices are displayed in part (C) of Fig. 1.



84.4%	1.3%	5.2%	6.5%	0.0%	2.6%	0.0%
6.7%	86.7%	6.7%	0.0%	0.0%	0.0%	0.0%
2.9%	0.0%	94.1%	0.0%	2.9%	0.0%	0.0%
10.0%	3.3%	3.3%	83.3%	0.0%	0.0%	0.0%
0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%
2.2%	0.0%	0.0%	2.2%	0.0%	95.7%	0.0%
0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%

Fig. 1 A semi-supervised coarse-to-fine framework with Bayesian optimization for multi-class lithology identification.(A)the flowchart of Bayesian optimization.(B) the training flow of the semi-supervised coarse-to-fine model.(C) testing model and display the partial results

Bayesian optimization

Snoek et al (2012) first proposed to apply the Bayesian optimization method to the parameter tuning of machine learning algorithms, aiming to find the best parameter group of the model through the Bayesian optimization method. Prior to the advent of Bayesian optimization, grid search and random search Liashchynskyi2019 (2012) were the dominant algorithms used for parameter tuning. However, these methods were computationally demanding, rendering them unsuitable for fine-tuning complex ensemble models.

Bayesian optimization is a sequential algorithm comprising two fundamental components. The initial stage involves substituting the objective function with ensemble algorithms through the use of a surrogate model. The prior distribution of the surrogate model is established by constructing the sampling function as a Gaussian process. In the second stage, the maximum acquisition function is computed to select the optimal sampling point, which leads to the updating of the posterior distribution of the surrogate model.

Rasmussen et al (2003) applied the Gaussian processes to a Bayesian optimization framework as a surrogate model. A Gaussian process is a stochastic process that extends a multivariate Gaussian distribution. Any finite random variables follow multivariate Gaussian distribution, and any random variable follows a one-dimensional Gaussian distribution. The multivariate Gaussian distribution is defined by the mean vector μ and the covariance matrix C, the extension to the Gaussian process is defined by the mean function $\mu(X) = E(F(X))$ and the covariance function $c(X, X') = E((F(X) - \mu(X))(F(X') - \mu(X'))$. Defined as Eq. 1 Rasmussen et al 2005.

$$F \sim GP\left(\mu\left(X\right), c\left(X, X'\right)\right), \tag{1}$$

In this paper, the notation $X = (X_1, X_2, ..., X_n)$ in Eq. 1 pertains to the amalgamation of various parameter sets of the ensemble algorithm. Further, $F = F(X) = [F(X_1), F(X_2), ..., F(X_n)]$ denotes each set of parameters that corresponds to the accuracy of the ensemble algorithm.

The acquisition function is an important component of Bayesian optimization, which plays a vital role in striking a balance between exploitation and exploration. Specifically, it aims to achieve a balance between local optima and potential solutions with large uncertainties obtained via global exploration. The acquisition function utilizes the prior distribution of the Gaussian process to achieve this objective. Getting the data sets before t iteration $D_t = (X_1, f(X_1), (X_2, f(X_2), ..., (X_t, f(X_t))$ is used to compute the maximum acquisition function $\alpha_t(X)$ to select the next parameter set X_{t+1} to be sampled. Define as Eq. 2 Brochu et al 2010.

$$X_{t+1} = \arg\max_{X \in D_t} \alpha_t (X), \tag{2}$$

In this study, we examine the three most commonly utilized acquisition functions: the maximum improvement probability method (PI) based on improvement, the expected improvement method (EI), and the upper confidence limit based on optimization (GP-UCB) Brochu et al (2010). As reported by Gan et al (2010), the EI function is the predominant acquisition function utilized in practical applications. The basic principle underlying the EI function is to determine the degree of improvement in the expected value at points exceeding the current optimal point through calculation, and subsequently selecting the point with the most substantial improvement as the subsequent sampling point. This approach emphasizes exploration by favoring and selecting regions with a slight mean and significant variance.

Combined with the Gaussian process and the EI acquisition function mentioned above, it is assumed that the parameter tuning process of iterative Bayesian optimization in iteration t is:

1. The data set obtained through the previous (t-1) rounds of iteration $D_{t-1} = (X_1, f(X_1)), (X_2, f(X_2)), ...,$ $(X_{t-1}, f(X_{t-1}))$ obtain the prior distribution of the Gaussian process (Brochu et al, 2010). Wherein $\mu(X_{1:t-1})$ is the mean vector made up of the expectation of $f(X_i)$, as shown as $\mu(X_{1:t-1}) = (E(f(X_1)), E(f(X_2)), ..., E(f(X_{t-1})))$, $i = 1, 2, ..., (t-1)$.

$$f(X_{1:t-1}) \sim GP(\mu(X_{1:t-1}), C_{1:t-1}), \tag{3}$$

In accordance with Eq. 3, the symbol $C_{1:t}$ represents the covariance matrix. The calculation formula for $C_{1:t}$ is demonstrated in Eq. 4, wherein $c(X_i, X_j) = E((f(X_i) - \mu(X_i))(f(X_j) - \mu(X_j)))$ denotes the covariance function of (X_i, X_j), where i and j are integer indices within the range of 1 to (t-1).

$$C_{1:t} = \begin{pmatrix} c(X_1, X_1) & \cdots & c(X_1, X_{t-1}) \\ \vdots & \vdots & \vdots \\ c(X_{t-1}, X_1) & \cdots & c(X_{t-1}, X_{t-1}) \end{pmatrix}, \tag{4}$$

2. The maximum acquisition function EI(X_{t-1}), namely Eq. 6 (Brochu et al, 2010), is calculated by using the data set D_{t-1} from the previous (t-1) iteration to obtain the next round of iteration sampling parameter set X_t, namely Eq. 5. Inclusive f^+_{t-1} in Eq. 6 is the maximum prediction accuracy in the previous (t-1) iteration. Moreover, Φ, ϕ is the standard normal distribution's distribution function and probability density function.

$$X_t = \arg\max_{X \in D_{t-1}} EI(X_{t-1}), \tag{5}$$

$$EI(X_{t-1}) = \left(\mu(X_{t-1}) - f^+_{t-1}\right) \Phi\left(\frac{\mu(X_{t-1}) - f^+_{t-1}}{\sigma(X_{t-1})}\right) + \sigma(X_{t-1}) \phi\left(\frac{\mu(X_{t-1}) - f^+_{t-1}}{\sigma(X_{t-1})}\right), \tag{6}$$

According to the Gaussian process, $f(X)$ corresponding to any point $X \in D_{t-1}$ follows a one-dimensional Gaussian distribution, namely Eq. 7. Therein $\mu(X_{1:t-1}) = E(f(X_{t-1})$, and $\sigma^2(X_{t-1}) = c(X_{t-1}, X_{t-1})$.

$$f(X_{t-1}) \sim N\left(\mu(X_{t-1}), \sigma^2(X_{t-1})\right), \tag{7}$$

3. To obtain a new data set D_t for the upcoming iteration (t+1), the optimal point $(X_t, f(X_t))$ is included in the previously collected data set, D_{t-1}. The updated Gaussian process is then used to generate the posterior distribution, which serves as the prior distribution in the upcoming iteration. This process is expressed by Eq. 8, wherein each variable's calculation formula remains the same as that of Eq. 3, with the exception of the inclusion of the data set collected in round t.

$$f(X_{1:t}) \sim GP(\mu(X_{1:t}), C_{1:t}), \tag{8}$$

Ensemble methods

Ensemble algorithms are known for their superior generalization capabilities in comparison to traditional classifiers. They enhance the predictive accuracy of models by combining several weak learners with lower prediction accuracy to form a robust learner Dietterich et al (2000). Boosting and Bagging are two significant categories of ensemble algorithms. Boosting algorithms reduce the ensemble classifier's bias by considering the influence of the preceding weak classifiers. The principle of Bagging algorithms involves repeatedly sampling the sample data and training each weak classifier independently to diminish the integrated algorithm's variance. This study has employed four classic ensemble algorithms, namely Adaptive Boosting (AdaBoost) Hastie et al 2009, Gradient Boosting Decision Tree (GBDT) Friedman2001 2001, Random Forest (RF) Breiman2001 2001, and Extremely Randomized Trees (ExtraRF) Geurts et al 2006 for comparative trials. The experimental results have shown that the ExtraRF algorithm outperforms the other algorithms.

The concept of extreme random trees was initially proposed by Geurts et al (2006), who constructed them in a top-down structure from decision trees similar to other tree-based ensemble algorithms. Since each decision tree collectively determines the ensemble algorithm's generalization ability, the generalization ability of each decision tree and the difference among decision trees will impact the ensemble algorithm's performance. An extreme random tree selects cut points randomly to divide nodes, thereby enhancing the randomness of node partitioning and the difference among decision trees. In contrast, Random Forest employs the random method of replacement to obtain the training set, resulting in repeated samples in each selected training set and leading to overfitting. The extreme random tree addresses this problem of repeated sampling. The training set of each decision tree is composed of all experimental data to improve sample utilization and reduce prediction bias.

Experiments

Data sets

The Ordos Basin is a prominent and extensive superimposed petroliferous basin, with its proven reserves of natural gas, coal-bed gas, and coal, ranking first in China. This study employs data collected from two distinct gas fields situated in the northern region of the Ordos Basin, China. The first field, known as the Daniudi Gas Field (DGF), is located in the Tabamiao area on the northern slope of Yi-shan, while the second field, the Hangjinqi Gas Field (HGF), is located in the northern bulge of the Islamic Union. The geological composition of the DGF region is predominantly quartz sandstone and sandstone, with smaller amounts of mudstone, carbonate, and coal. Conversely, the HGF region comprises primarily of lithic sandstone, coarse sandstone, mudstone, among others.

This paper collected 867 and 1,238 logging data from seven DGF and HGF wells and created two datasets. We used the two datasets to train the proposed model, respectively. We show the complete training process of each area in Algorithm 1 and Algorithm 2. During the training process, we randomly split the data into two datasets, which take 80% as the training set and 20% as the test set. Table 1 shows the lithology classes and quantities collected in the two areas. The coarse lithologies were mudstone (M), siltstone (S), coal rock (C), and sandstone (SS), which were further divided into pebbled coarse sandstone (PS), coarse sandstone (CS), fine sandstone (FS), and medium sandstone (MS). In total, we collected 867 logging data in the DGF area, comprising 114 CS data, 132 FS data, 120 PS data, 211 MS data, 53 S data, 133 M data, and 104 C data, and collected 1238 logging data from the HGF, including 207 CS data, 146 FS data, 370 PS data, 206 MS data, 47 S data, 248 M data, and 14 C data. Seven logging curves, namely Acoustic log (AC), Calliper log (CAL), Compensated Neutron log (CNL), Density log (DEN), Gamma ray log (GR), deep formation resistivity (LLD), and shallow formation resistivity (LLS), were used as the feature space to classify lithology. Each sample was composed of a 7-dimensional feature vector and a corresponding lithology class as the label. We tested the performance of the lithology identification model using log data from another well in each area, specifically 140 log data of well D17 in the DGF area and 166 log data of well J66 in the HGF area.

Coarse-to-fine framework based on bayesian optimization

In this paper, we split two progresses to describe the training process of the proposed semi-supervised model. The first implementation process is the coarse-to-fine model based on Bayesian optimization, as shown in Algorithm 1 and depicted in Section 3.2. The second implementation process shown in Algorithm 2 is the optimized coarse-to-fine framework based on pseudo-labeling, depicted in Section 3.3.

Algorithm 1 illustrate the coarse-to-fine model based on Bayesian optimization. Given the logging data of one area (X,Y) shown in Table 1. The coarse classes were mudstone (M), siltstone (S), coal (C), and sandstone (SS), defined as $Y_{coarse} = \{SS, C, S, M\}$. The sandstone (SS) could be refined into pebbly coarse sandstone (PS), coarse sandstone (CS), fine sandstone (FS), and medium sandstone (MS). The fine labels were defined as $Y_{fine} = \{FS, CS, PS, MS\}$.

To avoid the impact of inaccurate data on the model accuracy, we preprocessed the collected logging data using outlier detection before the experiment. As Xie et al. Xie et al (2021) demonstrated in their experiments with well-logging data from the same area, the LOF outlier detection algorithm could effectively deal with outlier data and improve the model's accuracy. Therefore, we used the LOF algorithm to preprocess the experimental data (X,Y) as (X_{lof},Y_{lof}) before training begins. And then we randomly split the collected data from one area into 80% training set (X_{train},Y_{train}) and 20% test set (X_{test},Y_{test}). Subsequently, Bayesian optimization was employed to optimize the parameters of the ensemble models, with the search range of parameters illustrated in Table 2. The iterative process of Bayesian optimization consists of two main parts. Firstly, the surrogate model $(\mathcal{GP})$ is used to replace the ensemble model $(\mathcal{F})$. The prior distribution of the surrogate model is used to build the sampling function as the Gaussian process. Secondly, the next optimal sampling point (x_i) is selected by calculating the maximum

Table 1 The special simple groups

	Classes	Descriptions	Number of DGF	Number of HGF
Coarse classes	M	mudstone	133	248
	S	siltstone	53	47
	C	coal	104	14
Fine classes	PS	pebbly coarse sandstone	120	370
(sandstone)	CS	coarse sandstone	114	207
	FS	fine sandstone	132	146
	MS	medium sandstone	211	206

acquisition function ($max\mathcal{EI}(x{\in}D_{i-1})$) and updating the posterior distribution of the surrogate model by the next set of sampling $\{D_{i-1},(x_i,y_i)\}$.

In our study, we bifurcated the training process of the model into two distinct stages. Firstly, we performed coarse model training by leveraging the labels ($\mathcal{MAP}(Y_{train})$) of the training set and mapping them onto the coarse-labeled data set (X_{train},Y_{coarse}). The Bayesian optimized ensemble model was trained using this coarse-labeled data set to obtain the Coarse model. Subsequently, for fine model training, we trained the Fine model using the fine-labeled data set (X_{fine},Y_{fine}) obtained from the training set. During the testing phase, we employed the Coarse model to predict the outcome for X_{test}, generating $Pred_c$. We selected the SS from $Pred_c$ to test the performance of the fine model.

Algorithm 1 Coarse-to-Fine framework Based on Bayesian optimized Ensemble model

```
Input : Logging Data(X,Y)={$(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$}
        where Y=($Y_{coarse}$
        ,$Y_{fine}$)=({SS,C,S,M},{FS,MS,PS,CS})
($X_{lof}$,$Y_{lof}$) $\leftarrow$ LOF(X,Y) ;
($X_{train}$,$Y_{train}$),($X_{test}$,$Y_{test}$) $\leftarrow$ TrainTestSplit($X_{lof}$,$Y_{lof}$,0.8);
procedure Bayesian optimization;
for $i$ from 1 to $\mathcal{N}$ do
    Input : $\mathcal{GP}$$\leftarrow$Surrogate Model, $\mathcal{EI}$ $\leftarrow$ Acquisition
            Function
    $D_{i-1}$={$(x_1,y_1),(x_2,y_2),...,(x_{i-1},y_{i-1})$};
    Use $\mathcal{GP}$ surrogate ensemble model as $\mathcal{F}$;
        $\mathcal{F}(D_i) \sim \mathcal{GP}$;
    $x_i \leftarrow max\mathcal{EI}(x \in D_{i-1})$;
    The optimized ensemble model $y_i$;
        $y_i \leftarrow \mathcal{F}(x_{i-1})+\delta$;
    $D_i \leftarrow \{D_{i-1},(x_i,y_i)\}$ ;
end
return Bayesian-optimized Ensemble model ;
end procedure;
procedure Training Bayesian-optimized Coarse-to-Fine
framework ;
 $Y_{coarse} \leftarrow \mathcal{MAP}(Y_{train})$;
 Coarse$\leftarrow$ Train($X_{train}$,$Y_{coarse}$);
 Select Label=$Y_{fine}$ from $\{X_{train},Y_{train}\}$;
 Fine$\leftarrow$ Train($X_{fine}$,$Y_{fine}$);
end procedure;
procedure Test Coarse-to-Fine framework;
 Select Label=$Y_{fine}$ from $\{X_{test},Y_{test}\}$ as $test_{fine}$;
 $Pred_c \leftarrow Test_{Coarse}(test_{fine})$;
 if $Pred_c$=SS then
     $Pred_f \leftarrow Test_{Fine}(test_{fine})$;
 end
 Calculate the accuracy;
end procedure
```

Pseudo labels

In the realm of semi-supervised learning, numerous methods have been proposed and investigated over the years. Among these, the most prominent and time-honored approaches include consistency regularization, proxy-label methods, and entropy minimization. Proxy-label methods, in particular, utilize a trained model on the labeled set to generate additional training examples by labeling instances from the unlabeled set based on certain heuristics. Self-training strategy, which is one of the proxy-label methods, falls under this category. In addition, entropy minimization serves as a viable method for expanding the training set, compelling the model to produce confident predictions while minimizing the entropy of the predictions. Furthermore, consistency training can be considered a form of proxy-label method, wherein the labels are predicted by calculating the distance of the outputs Ouali et al 2020.

In this study, we opted to employ self-training methods to realize semi-supervised learning, with pseudo-label serving as a key strategy Amini et al 2021. To begin with, we trained the Bayesian optimized coarse-to-fine model using labeled data as the baseline model. In each iteration, the current coarse-to-fine model was employed to predict the unlabeled data, and the actual probability of each predicted label was computed. The top 10 results with a predicted probability exceeding 95% were selected as the pseudo labels with high confidence. Subsequently, we augmented the labeled data set with the pseudo labels and their corresponding unlabeled data, thereby forming a new labeled data set to serve as the training data for the next round of iteration. This process was continued until reaching the termination condition of the iteration that we set.

The implementation process of the semi-supervised coarse-to-fine framework is depicted in Algorithm 2. To conduct

Table 2 Ensemble algorithms tuning parameter

Ensemble Methods	Tuned Parameters	Search Range	BO Parameters setting
RF	Max depth	[5,40]	18
	Max features	[1,7]	2
	The number of estimators	[50,300]	200
	Min samples split	[2,8]	3
GBDT	Max depth	[5,40]	28
	Max features	[1,7]	3
	The number of estimators	[50,300]	240
	Learning rate	[0,0.8]	0.50
ExtraRF	Max depth	[5,40]	39
	Max features	[1,7]	4
	The number of estimators	[50,300]	220
	Min samples split	[2,8]	3
AdaBoost	Max features	[1,7]	4
	The number of estimators	[50,300]	180
	Learning rate	[0,0.8]	0.78

experiments with the semi-supervised model, we excluded the labels of the test data set, utilizing it instead as the unlabeled data set (X_u). The semi-supervised learning process involved five iterations. In each iteration, the coarse-labeled model ($\mathcal{H}_c$) was initially employed to train the unlabeled data, thereby generating coarse labels. The top ten high-confidence labels (95%) were selected as the coarse pseudo labels ($\mathcal{PL}_c$). Next, the sandstone was selected from the predicted coarse labels, and the fine-labeled model was trained on the sandstone ($\mathcal{H}_f$). We predicted and selected the top ten high-confidence fine labels as the fine pseudo labels ($\mathcal{PL}_f$). Finally, the pseudo labels were merged into the training data set ($\{\mathcal{S}, (X_{PL}, \mathcal{PL})\}$).

Algorithm 2 Bayesian optimized Ensemble model Coarse-to-fine framework With Pseudo Labels

```
input : $X_u$ ←Unlabeled Data , $\mathcal{S}$=($X_{train}$,$Y_{train}$)
        $\mathcal{H}_c \leftarrow Coarse\ Model, \mathcal{H}_f \leftarrow Fine\ Model$
procedure Pseudo Labels;
for $iter$ from 0 to 5: do
    Train $\mathcal{H}_c$ on $\mathcal{S}$;
    $prob_c$← PredictProb $\mathcal{H}_c$ on $X_u$;
    if $prob_c$=$\mathcal{OVER}$(0.95)∩ $\mathcal{MAX}$[0:10] then
        $X_c \in X_u$;
        $\mathcal{PL}_c \leftarrow \mathcal{H}_c$.predict($X_c$)
    end
    Select $X_f \in X_u$ where $\mathcal{H}_c$.predict($X_u$)=SS;
    Train $\mathcal{H}_f$ on $\mathcal{S}$;
    if $prob_f$=$\mathcal{OVER}$(0.95)∩ $\mathcal{MAX}$[0:10] then
        $\mathcal{PL}_f \leftarrow \mathcal{H}_f$.predict($X_f$)
    end
    $X_{PL} \leftarrow X_c \cup X_f$;
    $\mathcal{PL} \leftarrow \mathcal{PL}_c \cup \mathcal{PL}_f$;
    $\mathcal{S} \leftarrow \{\mathcal{S}, (X_{PL}, \mathcal{PL})\}$
end
end procedure
```

Result analysis

To evaluate the efficacy of the Bayesian optimization and pseudo labels method in enhancing the classification performance of the coarse-to-fine model, we conducted an ablation study to present the findings. To ensure the reliability of the results and to examine the impact of different ensemble algorithms on the coarse-to-fine model, we applied four ensemble methods, namely ExtraRF, RF, GBDT, and AdaBoost, in the ablation study.

We utilized precision, recall, f1-score, prediction accuracy, and confusion matrices to evaluate the results of the ablation study. Precision, recall, f1-score, and prediction accuracy were determined using TP, FP, TN, and FN, where P (Positive) and N (Negative) denote whether the predicted class is positive or negative, and T (True) and F (False) represent whether the predicted result is accurate or not.

The precision, recall, F1-score, and accuracy are commonly used performance metrics to evaluate the effectiveness of a model. Precision measures the ratio of correctly predicted positive instances (TP) to the total predicted positive instances (TP+FP), which represents the model's accuracy and is defined as Eq. 9. Recall, on the other hand, is the ratio of correctly predicted positive instances (TP) to all actual positive instances (TP+FN), representing the proportion of retrieved positive instances. Its definition is shown in Eq. 10. F1-score is a metric that balances the precision and recall rates of a model and can be expressed as the harmonic mean of the two values, defined as Eq. 11. Finally, accuracy is the percentage of correct predictions out of the total prediction samples, which can be defined as Eq. 12.

$$precision = \frac{TP}{TP + FP}, \tag{9}$$

$$recall = \frac{TP}{TP + FN}, \tag{10}$$

$$f1 - score = \frac{2}{\frac{1}{recall} + \frac{1}{precision}}, \tag{11}$$

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN}, \tag{12}$$

To validate the generalizability of our results, logging data from two distinct regions, namely DGF and HGF, were selected as the experimental data. The results of the confusion matrices can be found in Fig. 4,5,6,7,8,9,10, and 11 in Appendix A while the results of other measurements are shown in Tables 3,4,5,6,7,8,9, and 10 of Appendix B. By observing the tables and figures, we can conclude that the ExtraRF-based model achieves the highest classification accuracy, with 88.2% in the DGF area and 91.6% in the HGF area. As mentioned earlier, the total number of training data in the HGF area is higher than in DGF, which leads to better prediction accuracy in HGF.

Ablation study

Table 3 in Appendix B presents a summary of the ExtraRF-based coarse-to-fine model's precision, recall, f1-score, and prediction accuracy in the HGF area. The most notable observation in the table is the improvement in accuracy. The model in Table 3(b) achieved 91.6%, an increase of 4.9% from 86.7% in the baseline model shown in Table 3(a) in Appendix B. Table 3(c) and 3(d) in Appendix B show that their accuracy is 87.6% and 88%, respectively. The accuracy results indicate that the Bayesian optimization and pseudo labels method can enhance the model's multi-classification ability and are considerably better than using them individually.

Apart from accuracy, precision, recall, and f1-score for each rock class also show similar positive outcomes, as demonstrated in Table 3 in Appendix B. Table 3(a) in Appendix B shows that the classification precision of C and S reached 100%, while CS has only 76%. Only C has a recall of 100%, and the classes of MS, M, and S are less than the precision. The lowest recall of CS is only 78%. Precision and recall jointly affected the f1-score, so the lowest f1-score of CS is 77%, the highest of C is 100%, and the other classes are all around 80%-90%. Turning to Table 3(c) in Appendix B, it is evident that CS has the most significant effect. The precision and recall of CS improve from 76% and 78% to 94% and 89%, respectively, an increase of more than 10%. The precision of MS also increased from 87% to 94% by 7%. Meanwhile, the recall of MS also improved by 4%. Examining the experimental evidence in Table 3(d) of B, the recall of S is significantly improved from 82% to 91%, balancing the difference between precision and recall, resulting in an increase in the f1-score. Overall, the comparison shows that the Bayesian optimization or pseudo labels method can increase the efficiency of the multi-classification model.

Interestingly, the combination of the Bayesian optimization parameter tuning and pseudo labels method has proven to be effective in improving the model's classification ability. A comparison of Table 3(b) and Table 3(c) in B reveals an 8% increase in the recall of CS, resulting in a 3% increase in the f1-score. The precision and recall of M have also increased from 83% and 93% to 93% and 95%, respectively. Additionally, the precision and recall of PS have increased from 82% and 94% to 90% and 96%. By contrasting Table 3(b) and Table 3(d) in Appendix B, it is observed that the precision of CS, FS, MS, and PS has been improved. CS and FS have increased from about 80% to approximately 90%, resulting in an improvement of nearly 10%. MS and PS have enhanced from 87% and 88% to 90% by 3% and 2%, respectively. Furthermore, the recall of CS and MS has risen from 76% and 82% to 97% and 90%, respectively, resulting in a remarkable increase in the f1-score of CS from 79% to 94% by 15%. These results further demonstrate that the combination of Bayesian optimization parameter tuning and pseudo labels method can improve the identification ability of the multi-classification model. This conclusion has been verified by testing the other three ensemble methods in the HGF area, and the experimental results are presented in Table 4-Table 6 of Appendix B.

Table 4 in Appendix B reveals that the model in Table 4(b) outperforms the baseline model with an accuracy of 86.2%, a 4% increase. The models in Table 4(c) and Table 4(d) also exhibit higher accuracy than the baseline, with an increase of 3.1% and 1.4%, respectively, but still fall short of the Table 4(b) model. A comparison of Table 4(a) and Table 4(b) in Appendix B illustrates an 8% improvement in the recall of class M, contributing to a 6% increase in the f1-score. The precision and recall of CS and PS also improve, resulting in a 2% increase in their f1-scores. Notably, FS exhibits the most remarkable improvement, with precision increasing nearly 10%, from 74% to 83%, and recall improving more than 20%, from 71% to 92%. A comparison of Table 4(b) and Table 4(c) in Appendix B also shows a significant improvement in FS, with recall increasing around 20% from 70% to 92%. Similarly, the recall of FS and CS also increases by 17% and 7%, respectively, when comparing Table 4(d) and Table 4(b) in Appendix B.

From the data presented in Table 5 of Appendix B, it is evident that the accuracy of the Table 5(b) model in Appendix B is higher at 85.3% compared to the 81.8% achieved by the Table 5(a) model in Appendix B. The accuracy of Table 5(c) and Table 5(d) models in Appendix B are 84.9% and 84.5%, respectively, which are both lower than that of Table 5(b) in Appendix B but higher than Table 5(a) in Appendix B. This result aligns well with previous studies. Comparing Table 5(a) and Table 5(b) in Appendix B, we observe a significant increase in the recall of S from 50% to 90% by 40%, while the precision of FS and C improved from 70% and 50% to 95% and 100% by 25% and 50%. The recall and precision of PS increased from 81% and 79% to 95% and 83%, respectively.

Lastly, the AdaBoost-based coarse-to-fine model is presented in Table 6 of Appendix B. An analysis of Table 6 of Appendix B leads to the same conclusion as above, with the highest accuracy being 85.3%, which is 4% higher than the base model. By optimizing the AdaBoost parameter using Bayesian optimization and incorporating pseudo labels, the recall of FS and S, which was around 60% in Table 6(a) of Appendix B, improved to nearly 90%.

To substantiate the generalizability of our findings, we conducted similar experiments in the DGF region. The results of these experiments are presented in Table 7-Table 10 in Appendix B. However, due to the limited availability of experimental data in DGF, the accuracy of the models is lower compared to HGF. A comparison of Table 7(a) and Table 7(b) in Appendix B shows that the ExtraRF-based model achieved a maximum accuracy of 88.2%, a significant improvement of 6.2% over the baseline model. However, the baseline model performed poorly in classifying CS and PS, which was significantly improved by employing Bayesian optimization and pseudo labels, especially for PS, which increased to about 90%. The RF-based model shown in Table 8 of Appendix B achieved an accuracy of 85.7% in Table 8(b), an improvement of 4.3% over the Table 8(a) model in Appendix B. Comparing the two models revealed a significant increase in the precision and recall of CS, MS, and PS. The accuracy of the GBDT-based model and AdaBoost-based model increased to 83.1% and 83.9% in Table 9(b) and Table 10(b) in Appendix B, respectively. Similarly, a close examination of Table 9 and Table 10 in Appendix B suggests that the Bayesian optimization and pseudo labels method is highly beneficial in improving the identification ability of fine classes. Overall, our experiments indicate that the ExtraRF-based coarse-to-fine framework with Bayesian optimization and pseudo labels has impressive capabilities for lithology multi-classification, especially for fine classes.

Confusion matrices

In this section, we present the confusion matrices for the coarse-to-fine framework based on the untuned parameter ensemble classifier, as depicted in Fig. 4,5,6,7,8,9,10, and 11 of Appendix A. Overall, the results suggest that the classification ability of the model, particularly for fine classes, is not strong. For example, in the HGF area, Fig. 4,5,6, and 7 in Appendix A illustrate these relationships. Fig. 4(a) in Appendix newinlinkApp1app1A reveals that the baseline model has an excellent classification effect of 100% for C; however, there is still potential for improvement in the classification ability of other classes. Comparing Fig. 4(a) and Figure 4(b) in Appendix A shows that the prediction accuracy of CS, MS, and M has increased by nearly 20%, 11.2%, and 6.9%, respectively. However, in contrast, there is a significant decrease in the accuracy of the C and S. There are two main reasons for this result. One is that the imbalance of the collected training data will enable the model to learn insufficiently for the small number of rock categories in the training process. Table 1 displays that the DGF area only has 53 S data, which is 6.1% of the DGF area's total data. The HGF area contains 47 logging data of S and 14 logging data of C, which only account for 3.7% and 1.1% of the training data in the HGF area, respectively. The other reason is the accuracy deviation caused by the inevitable accumulation of errors in the iterative process of pseudo-label technology Ouali et al (2020). During training, the cumulative error of pseudo-labels mixing the logging data with a small proportion of categories may cause the accuracy of this class to decrease.

Conversely, comparing Fig. 4(c) and 4(a) in Appendix A demonstrates that the former had a more significant effect on fine classes. The accuracy of CS and MS has significantly improved, but the improvement amplitude is lower than that of Fig. 4(b) of Appendix A. Furthermore, comparing Fig. 4(a) and Figure 4(d) in Appendix A reveals that the prediction accuracy of PS, MS, and S classes has also increased, with S experiencing a sharp rise to 90.9% from 81.8%. Nevertheless, the accuracy of other classes, except for FS, did not show a noticeable improvement. Specifically, PS only improved by 1%, and FS increased by 6% from 82.8% to 88.9%.

In the HGF area, the other three ensemble algorithms also yield similar conclusions, albeit with lower accuracy than the ExtraRF-based model. Fig. 5 of Appendix A presents the results of the RF-based model. Comparison of Fig. 5(a) and Figure 5(b) in Appendix A reveals that the accuracy of FS and M improved most significantly, with FS increasing from 71.4% to 92.3% by 20.9% and M from 82.9% to 94.9% by 12%. Fig. 5(c) in Appendix A shows that the accuracy of CS and S can be improved by 9.2% and 12.2% compared with Fig. 5(a) in Appendix A, while Fig. 5(d) in Appendix A shows that S increased by 11.1%. Further analysis of Fig. 6(a) and Fig. 6(b) in Appendix A shows that PS increased from 81.1% to 94.5% by 13.4%, and CS from 85.3% to 92.5% by 7.2%. S rose steeply from 50% to 90%, a significant increase of 40%. Figure 7(a) and Fig. 7(b) in Appendix A exhibit several differences. There is an upward trend in the accuracy of MS, FS, and S, with MS increasing by 11.1%, PS by 22.5% from 65.5% to 88%, and S leaping by 26.2% from 57.1%.

Additionally, similar experimental findings are obtained in DGF. The confusion matrix of the ExtraRF-based model is demonstrated in Fig. 8 in Appendix A. In particular, Fig. 8(b) shows that PS reaches a peak of 92.9%, an increase of nearly 25% compared to the model shown in Fig. 8(a) of Appendix A. Furthermore, the accuracy of CS increased by 16.7%, while M and C achieved 100% accuracy. By analyzing the results from Figs. 9,10, and 11 in Appendix A, we are more confident in the classification ability of the proposed model. The model exhibits analogous classification accuracy across different datasets and for each class. Overall, our results indicate that the combination of Bayesian optimization and pseudo labels with a coarse-to-fine framework can effectively enhance the performance of lithology identification, particularly for fine classes.

Well-logging and testing

In this study, we have applied our lithology identification model to two new wells, namely well J66 located in the HGF area and well D17 situated in the DGF area, with the aim of validating its ability. Subsequently, we have compared the model's predictions with the actual lithology, and have presented the outcomes along with log curves in Figs. 2 and 3. Our analysis of the aforementioned figures reveals that both wells, D17 in the DGF area and J66 in the HGF area, achieved a lithology identification accuracy of 98.6% and 95.2%, respectively, using our model. Our test results demonstrate that our model can effectively solve the challenge of lithology identification in intelligent logging wells.

Conclusions

This paper presents a semi-supervised Coarse-to-Fine approach with Bayesian optimization for lithology identification. Unlabeled data was utilized to improve the accuracy of lithology prediction. The results suggest that combining the coarse-to-fine framework with semi-supervised learning can leverage unlabeled data to enhance model accuracy, particularly for sandstone classes. Additionally, the study demonstrates that compared to traditional parameter optimization methods, Bayesian optimization can improve optimization speed and achieve global parameter optimization.

However, the practical implications of this research may be influenced by factors such as geographic location and the amount of available data. For this experiment, logging data from HGF and DGF were selected, and high accuracy in lithology identification was obtained. The proposed model is expected to perform similarly in other areas. However, the distribution of logging data may vary across different regions, which may cause the accuracy of the lithology identification model to decrease, making it difficult to generalize into a universal model. Furthermore, the model was trained on a limited number of logging data (over 2000 logging data from two areas) and only used seven logging curves as data characteristics for each logging data. This limitation could impact the diversity of model selection and the classification accuracy.

The distinction in logging data distributions may result in decreased accuracy and difficulty in establishing a universal model. Future research directions can explore the combination of transfer learning and semi-supervised learning for domain-adaptive lithology identification to address this issue. The expected outcome of such research would be to ensure classification accuracy and establish a universal lithology identification model.

Author Contributions Yunxin Xie and Chenyang Zhu conceived and designed the experiments; Yunxin Xie, Liangyu Jin and Chenyang Zhu wrote the original paper. Siyu Wu performed the experiments and draw the figures; Yunxin Xie and Chenyang Zhu analyzed the data; Liangyu Jin reviewed the paper.

Funding This work is supported by Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (No.22KJB170007, No.22KJB520012).

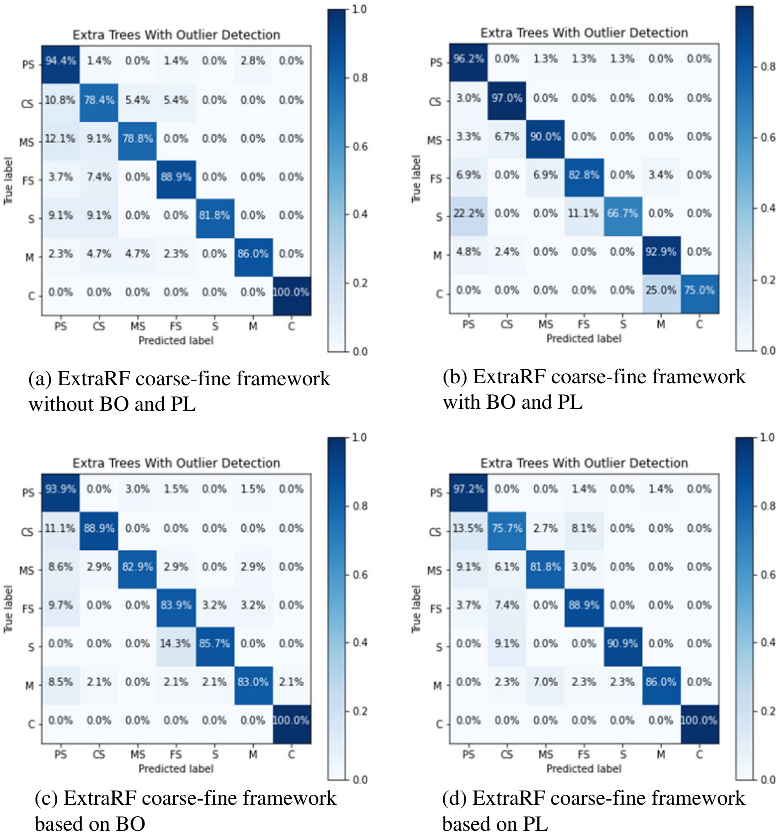
Availability of data and materials The data could be accessed from Appendix A of this paper: Xie, Yunxin, et al. "Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances." Journal of Petroleum Science and Engineering 160 (2018): 182-193.

Declarations

Competing Interest No conflict of interest exits in the submission of this manuscript.

Appendix A

Fig. 2 Logging Curves and Prediction Results of ExtraRF coarse-to-fine framework based on BO and Pseudo Labels in Well J66 Data of HGF



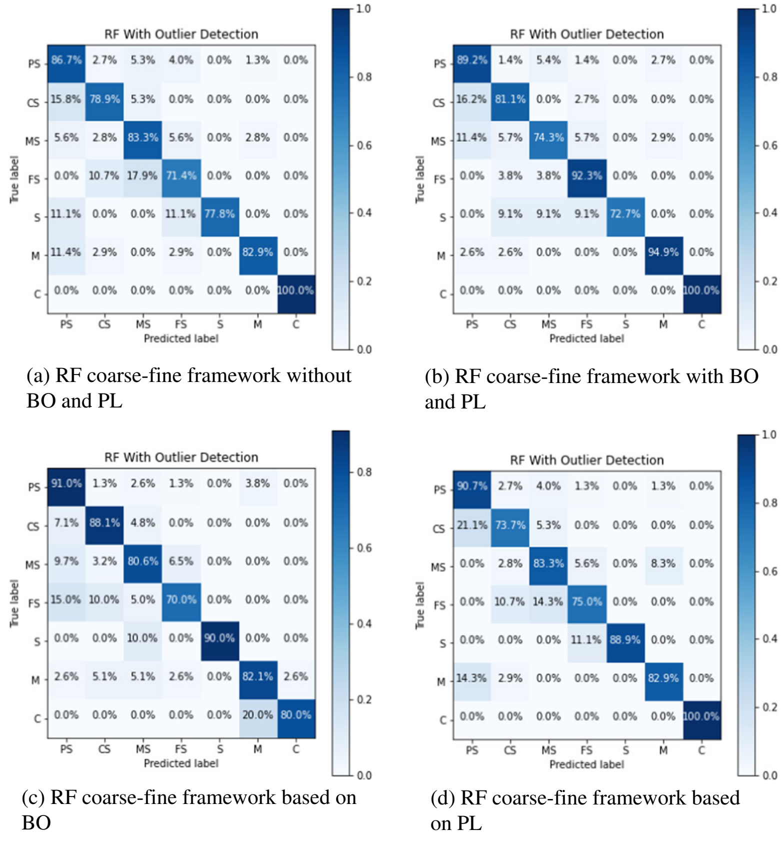
(a) ExtraRF coarse-fine framework without BO and PL

(b) ExtraRF coarse-fine framework with BO and PL

(c) ExtraRF coarse-fine framework based on BO

(d) ExtraRF coarse-fine framework based on PL

Fig. 3 Logging Curves and Prediction Results of ExtraRF coarse-to-fine framework based on BO and Pseudo Labels in Well D17 Data of DGF



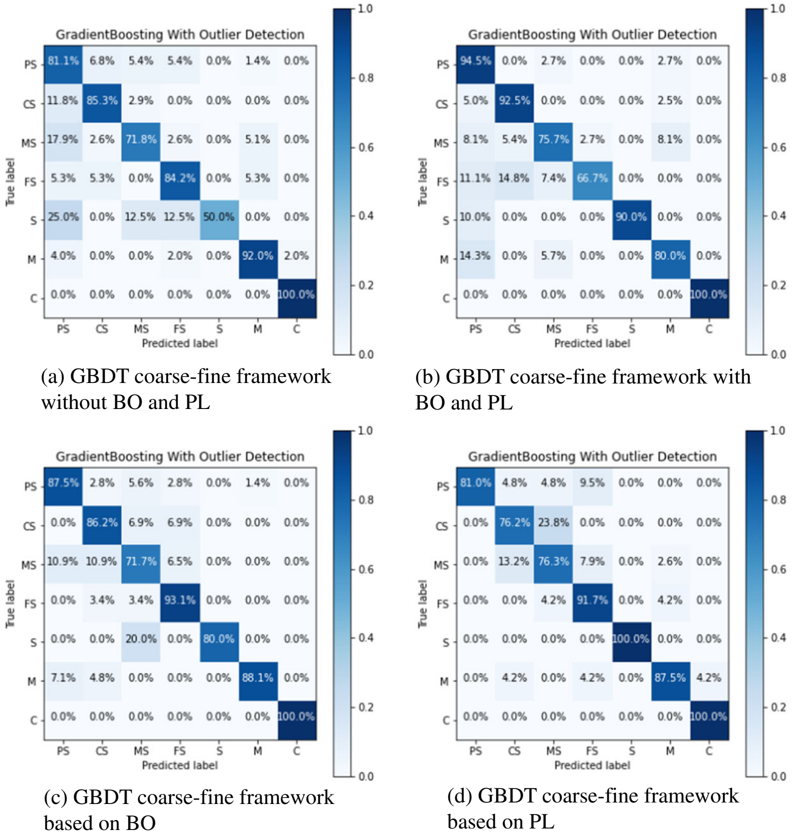
(a) RF coarse-fine framework without BO and PL

(b) RF coarse-fine framework with BO and PL

(c) RF coarse-fine framework based on BO

(d) RF coarse-fine framework based on PL

Fig. 4 Confusion matrices on the HGF test set with ExtraRF



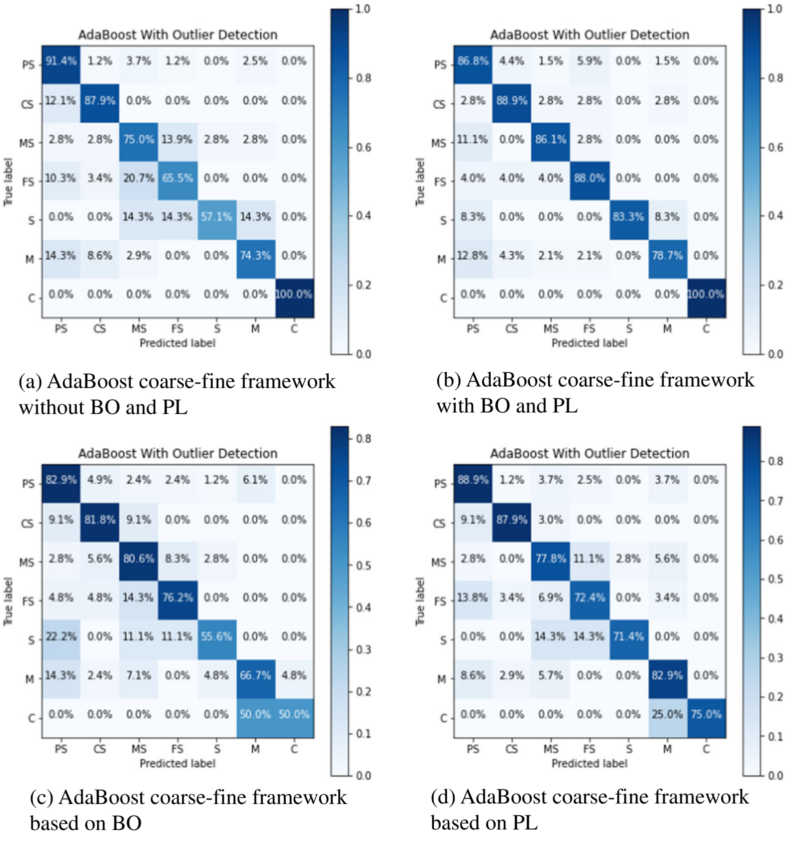
(a) GBDT coarse-fine framework without BO and PL

(b) GBDT coarse-fine framework with BO and PL

(c) GBDT coarse-fine framework based on BO

(d) GBDT coarse-fine framework based on PL

Fig. 5 Confusion matrices on the HGF test set with RF



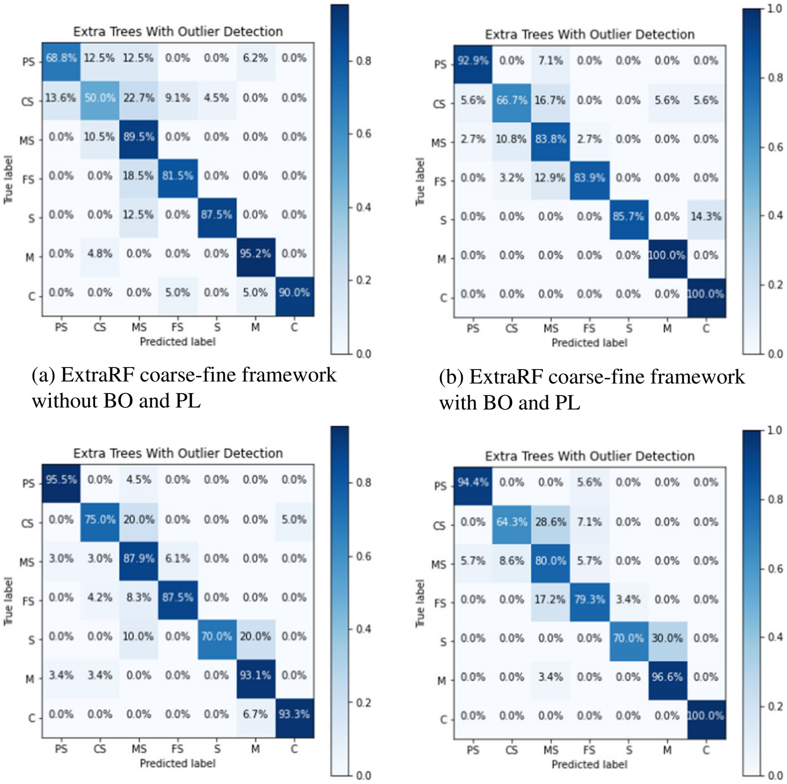
(a) AdaBoost coarse-fine framework without BO and PL

(b) AdaBoost coarse-fine framework with BO and PL

(c) AdaBoost coarse-fine framework based on BO

(d) AdaBoost coarse-fine framework based on PL

Fig. 6 Confusion matrices on the HGF test set with GBDT



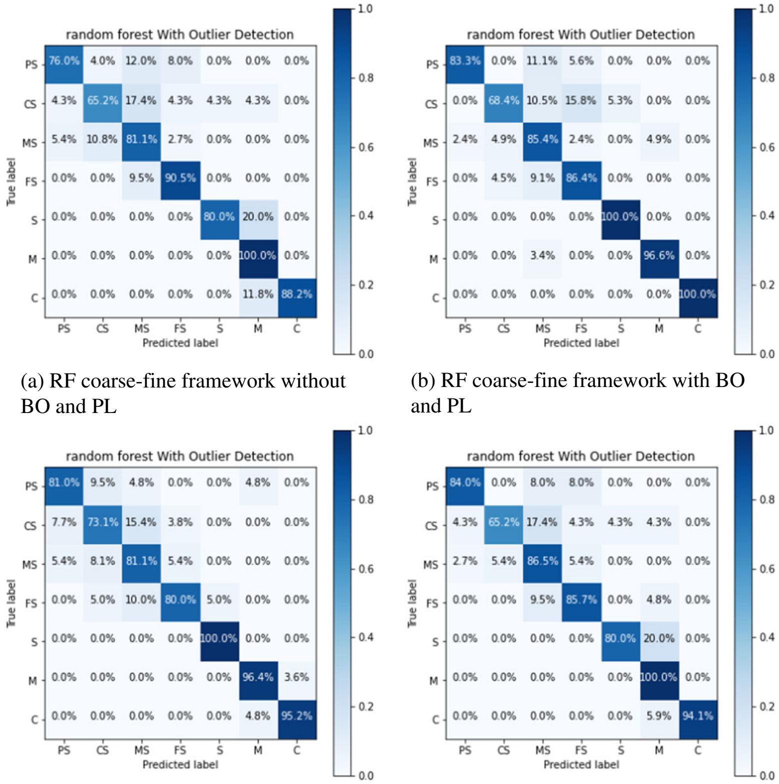
(a) ExtraRF coarse-fine framework without BO and PL

(b) ExtraRF coarse-fine framework with BO and PL

(c) ExtraRF coarse-fine framework based on BO

(d) ExtraRF coarse-fine framework based on PL

Fig. 7 Confusion matrices on the HGF test set with AdaBoost



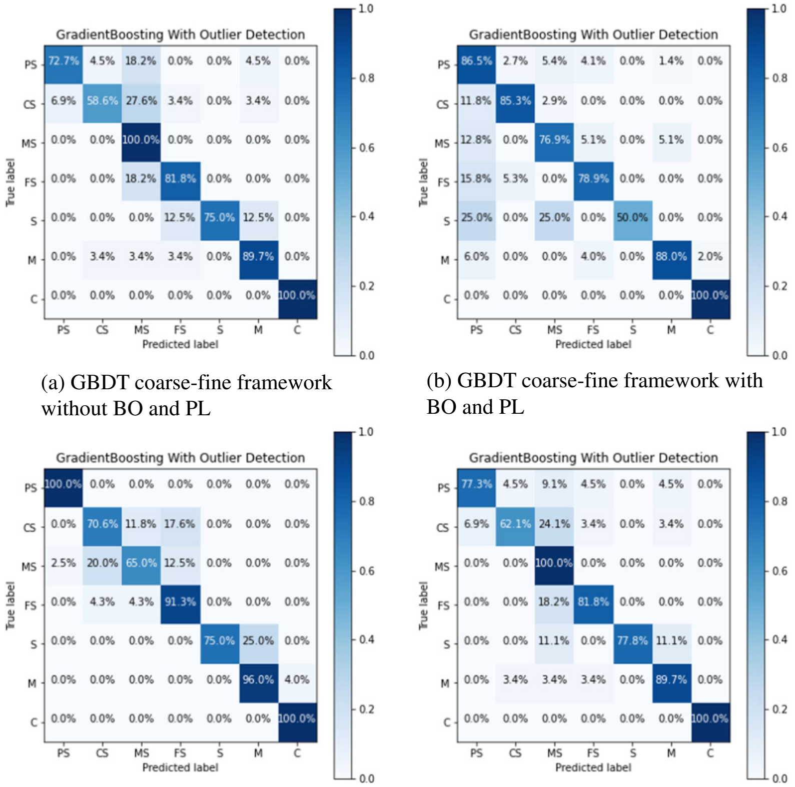
(a) RF coarse-fine framework without BO and PL

(b) RF coarse-fine framework with BO and PL

(c) RF coarse-fine framework based on BO

(d) RF coarse-fine framework based on PL

Fig. 8 Confusion matrices on the DGF test set with ExtraRF



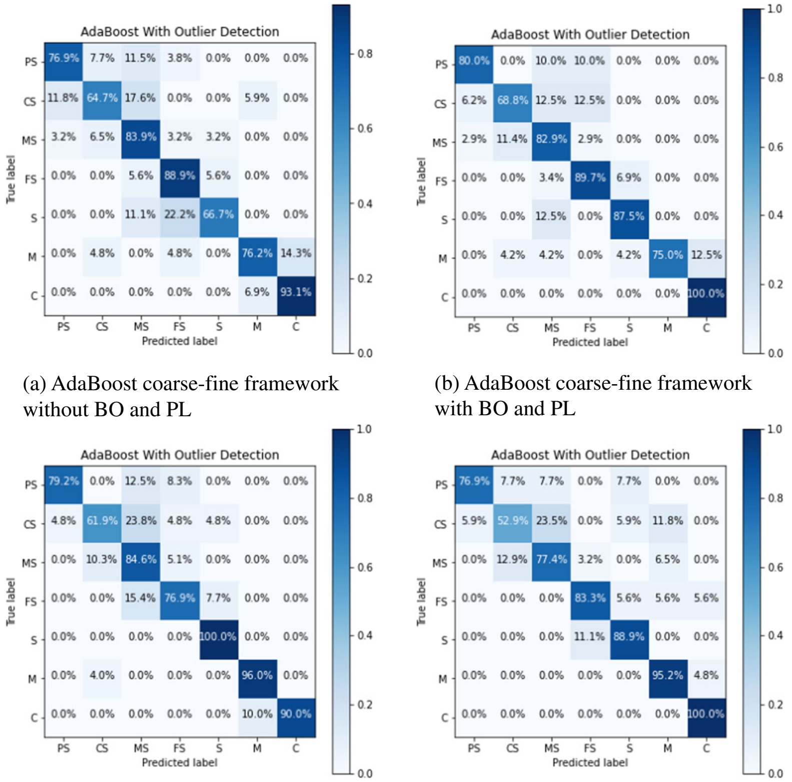
(a) GBDT coarse-fine framework without BO and PL

(b) GBDT coarse-fine framework with BO and PL

(c) GBDT coarse-fine framework based on BO

(d) GBDT coarse-fine framework based on PL

Fig. 9 Confusion matrices on the DGF test set with RF



(a) AdaBoost coarse-fine framework without BO and PL

(b) AdaBoost coarse-fine framework with BO and PL

(c) AdaBoost coarse-fine framework based on BO

(d) AdaBoost coarse-fine framework based on PL

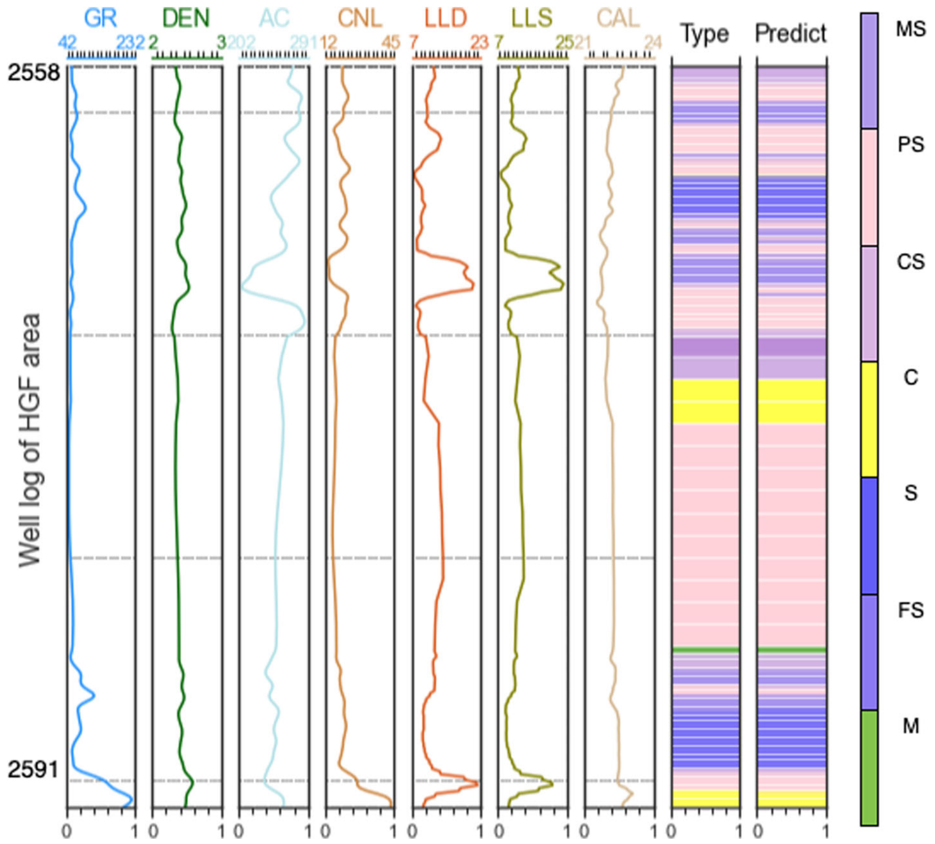


Fig. 10 Confusion matrices on the DGF test set with GBDT

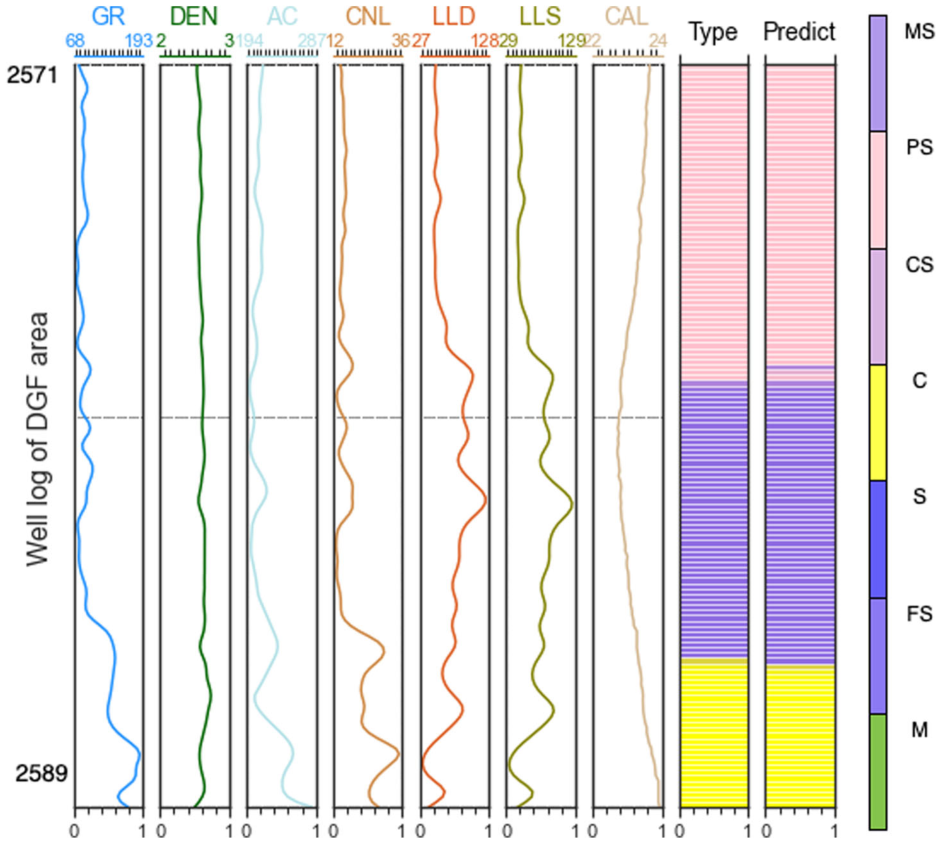


Fig. 11 Confusion matrices on the DGF test set with AdaBoost s

Appendix B

Table 3 Precision, recall, F1-score and prediction accuarcy for ExtraRF coarse-to-fine classifier on the HGF dataset

class	precision	recall	f1	acc
(a) ExtraRF coarse-fine framework without BO and PL				
C	**1**	**1**	**1**	0.867
CS	0.76	0.78	0.77	
FS	0.86	0.89	0.87	
M	0.95	0.86	0.9	
MS	0.87	0.79	0.83	
PS	0.86	0.94	0.9	
S	**1**	0.82	0.9	
(b) ExtraRF coarse-fine framework with BO and PL				
C	**1**	0.75	0.86	**0.916**
CS	0.91	**0.97**	**0.94**	
FS	0.92	0.83	0.87	
M	0.93	0.95	0.94	
MS	0.9	0.9	0.9	
PS	0.9	0.96	0.93	
S	0.86	0.67	0.75	
(c) ExtraRF coarse-fine framework based on BO				
C	0.75	**1**	0.86	0.876
CS	**0.94**	0.89	**0.91**	
FS	0.87	0.84	0.85	
M	0.93	0.83	0.88	
MS	**0.94**	0.83	0.88	
PS	0.82	0.94	0.87	
S	0.75	0.86	0.8	
(d) ExtraRF coarse-ne framework based on PL				
C	**1**	**1**	**1**	0.88
CS	0.82	0.76	0.79	
FS	0.8	0.89	0.84	
M	0.97	0.86	0.91	
MS	0.87	0.82	0.84	
PS	0.88	0.97	0.93	
S	0.91	0.91	0.91	

Table 4 Precision, recall, F1-score and prediction accuarcy for RF coarse-to-fine classier on the HGF dataset

class	precision	recall	f1	acc
(a) RF coarse-fine framework without BO and PL				
C	**1**	**1**	**1**	0.822
CS	0.81	0.79	0.8	
FS	0.74	0.71	0.73	
M	0.94	0.83	0.88	
MS	0.73	0.83	0.78	
PS	0.83	0.87	0.85	
S	**1**	0.78	0.88	
(b) RF coarse-fine framework with BO and PL				
C	**1**	**1**	**1**	**0.862**
CS	0.83	0.81	0.82	
FS	0.83	0.92	0.87	
M	0.93	0.95	0.94	
MS	0.81	0.74	0.78	
PS	0.86	0.89	0.87	
S	**1**	0.73	0.84	
(c) RF coarse-fine framework based on BO				
C	0.8	0.8	0.8	0.853
CS	0.86	0.88	0.87	
FS	0.78	0.7	0.74	
M	0.89	0.82	0.85	
MS	0.76	0.81	0.78	
PS	0.88	**0.91**	0.89	
S	**1**	0.9	**0.95**	
(d) RF coarse-fine framework based on PL				
C	**1**	**1**	**1**	0.836
CS	0.8	0.74	0.77	
FS	0.84	0.75	0.79	
M	0.88	0.83	0.85	
MS	0.77	0.83	0.8	
PS	0.84	0.91	0.87	
S	**1**	0.89	0.94	

Table 5 Precision, recall, F1-score and prediction accuarcy for GBDT coarse-to-fine classifier on the HGF dataset

class	precision	recall	f1	acc
(a) GBDT coarse-fine framework without BO and PL				
C	0.5	**1**	0.67	0.818
CS	0.81	0.85	0.83	
FS	0.7	0.84	0.76	
M	0.92	0.92	**0.92**	
MS	0.82	0.72	0.77	
PS	0.79	0.81	0.8	
S	**1**	0.5	0.67	
(b) GBDT coarse-fine framework with BO and PL				
C	**1**	**1**	**1**	**0.853**
CS	0.86	0.93	0.89	
FS	0.95	0.67	0.78	
M	0.82	0.8	0.81	
MS	0.82	0.76	0.79	
PS	0.83	0.95	0.88	
S	**1**	0.9	0.95	
(c) GBDT coarse-fine framework based on BO				
C	**1**	**1**	**1**	0.849
CS	0.71	0.86	0.78	
FS	0.79	0.93	0.86	
M	0.97	0.88	0.93	
MS	0.8	0.72	0.76	
PS	0.89	0.88	0.88	
S	**1**	0.8	0.89	
(d) GBDT coarse-fine framework based on PL				
C	0.95	**1**	0.97	0.845
CS	0.67	0.76	0.71	
FS	0.79	0.92	0.85	
M	0.91	0.84	0.87	
MS	0.84	0.91	0.87	
PS	**1**	0.81	0.89	
S	**1**	**1**	**1**	

Table 6 Precision, recall, F1-score and prediction accuarcy for AdaBoost coarse-to-fine classifier on the HGF dataset

class	precision	recall	f1	acc
(a) AdaBoost coarse-fine frame-work without BO and PL				
C	**1**	**1**	**1**	0.813
CS	0.83	0.88	0.85	
FS	0.73	0.66	0.69	
M	0.87	0.74	0.8	
MS	0.71	0.75	0.73	
PS	0.85	0.91	0.88	
S	0.8	0.57	0.67	
(b) AdaBoost coarse-fine frame-work with BO and PL				
C	**1**	**1**	**1**	**0.853**
CS	0.84	0.89	0.86	
FS	0.76	0.88	0.81	
M	0.93	0.79	0.85	
MS	0.89	0.86	0.87	
PS	0.82	0.87	0.84	
S	**1**	0.83	0.91	
(c) AdaBoost coarse-fine frame-work based on BO				
C	**1**	0.5	0.67	0.827
CS	0.69	0.89	0.77	
FS	0.8	0.64	0.71	
M	0.95	0.8	**0.87**	
MS	0.81	0.81	0.81	
PS	0.82	**0.91**	0.86	
S	**1**	0.75	0.86	
(d) AdaBoost coarse-fine frame-work with BO and PL				
C	**1**	0.75	0.86	0.831
CS	0.91	0.88	**0.89**	
FS	0.75	0.72	0.74	
M	0.81	0.83	0.82	
MS	0.76	0.78	0.77	
PS	0.87	**0.89**	0.88	
S	0.83	0.71	0.77	

Table 7 Precision, recall, F1-score and prediction accuarcy for ExtraRF coarse-to-fine classier on the DGF dataset

class	precision	recall	f1	acc
(a) ExtraRF coarse-fine frame-work without BO and PL				
C	**1**	0.9	**0.95**	0.82
CS	0.61	0.5	0.55	
FS	0.88	0.81	0.85	
M	0.91	**0.95**	0.93	
MS	0.72	0.89	0.8	
PS	0.79	0.69	0.73	
S	0.88	0.88	0.88	
(b) ExtraRF coarse-fine frame-work with BO and PL				
C	0.9	**1**	0.95	**0.882**
CS	0.71	0.67	0.69	
FS	0.96	0.84	0.9	
M	0.96	**1**	**0.98**	
MS	0.79	0.84	0.82	
PS	0.87	0.93	0.9	
S	**1**	0.86	0.92	
(c) ExtraRF coarse-fine frame-work based on BO				
C	0.93	0.93	**0.93**	0.876
CS	0.83	0.75	0.79	
FS	0.91	0.84	0.87	
M	0.9	0.93	0.92	
MS	0.78	0.88	0.83	
PS	0.91	**0.95**	**0.93**	
S	**1**	0.7	0.82	
(d) AdaBoost coarse-fine frame-work with BO and PL				
C	**1**	**1**	**1**	0.857
CS	0.75	0.64	0.69	
FS	0.85	0.79	0.82	
M	0.9	0.97	0.93	
MS	0.74	0.8	0.77	
PS	0.89	0.94	0.92	
S	0.88	0.7	0.78	

Table 8 Precision, recall, F1-score and prediction accuarcy for ExtraRF coarse-to-fine classier on the DGF dataset

class	precision	recall	f1	acc
(a) RF coarse-fine framework without BO and PL				
C	**1**	0.88	**0.94**	0.814
CS	0.75	0.65	0.7	
FS	0.76	0.9	0.83	
M	0.75	**1**	0.86	
MS	0.77	0.81	0.79	
PS	0.86	0.76	0.81	
S	0.89	0.8	0.84	
(b) RF coarse-fine framework with BO and PL				
C	**1**	**1**	**1**	**0.857**
CS	0.81	0.68	0.74	
FS	0.76	0.83	0.79	
M	0.88	0.97	0.92	
MS	0.83	0.85	0.84	
PS	0.94	0.83	0.88	
S	0.89	**1**	0.94	
(c) RF coarse-fine framework based on BO				
C	**0.95**	0.95	**0.95**	0.851
CS	0.76	0.73	0.75	
FS	0.84	0.8	0.82	
M	0.93	0.96	**0.95**	
MS	0.81	0.81	0.81	
PS	0.81	0.81	0.81	
S	0.75	**1**	0.86	
(d) RF coarse-fine framework based on PL				
C	**1**	0.94	**0.97**	0.845
CS	0.88	0.65	0.75	
FS	0.75	0.86	0.8	
M	0.75	**1**	0.86	
MS	0.8	0.86	0.83	
PS	0.91	0.84	0.87	
S	0.89	0.8	0.84	

Table 9 Precision, recall, F1-score and prediction accuarcy for GBDT coarse-to-fine classier on the DGF dataset

class	precision	recall	f1	acc
(a) RF coarse-fine framework without BO and PL				
C	**1**	**1**	**1**	0.807
CS	0.89	0.59	0.71	
FS	0.86	0.82	0.84	
M	0.81	0.9	0.85	
MS	0.63	**1**	0.77	
PS	0.89	0.73	0.8	
S	**1**	0.67	0.8	
(b) GBDT coarse-fine framework with BO and PL				
C	0.5	**1**	0.67	**0.831**
CS	0.91	0.85	0.88	
FS	0.68	0.79	0.73	
M	0.94	0.88	**0.91**	
MS	0.81	0.77	0.79	
PS	0.79	0.86	0.83	
S	**1**	0.5	0.67	
(c) GBDT coarse-fine framework based on BO				
C	0.95	**1**	**0.98**	0.826
CS	0.57	0.71	0.63	
FS	0.72	0.91	0.81	
M	0.8	0.96	0.87	
MS	0.9	0.65	0.75	
PS	0.95	**1**	0.97	
S	**1**	0.75	0.86	
(d) GBDT coarse-fine framework based on PL				
C	**1**	**1**	**1**	0.826
CS	0.9	0.62	0.73	
FS	0.86	0.82	0.84	
M	0.81	0.9	0.85	
MS	0.66	**1**	0.79	
PS	0.89	0.77	0.83	
S	**1**	0.78	0.88	

Table 10 Precision, recall, F1-score and prediction accuarcy for AdaBoost coarse-to-fine classier on the DGF dataset

class	precision	recall	f1	acc
(a) AdaBoost coarse-fine frame-work without BO and PL				
C	**0.9**	**0.93**	**0.92**	0.795
CS	0.69	0.65	0.67	
FS	0.76	0.84	0.8	
M	0.8	0.73	0.76	
MS	0.74	0.84	0.79	
PS	0.87	0.77	0.82	
S	0.75	0.67	0.71	
(b) AdaBoost coarse-fine frame-work with BO and PL				
C	0.88	**1**	**0.94**	**0.839**
CS	0.69	0.69	0.69	
FS	0.84	0.9	0.87	
M	**1**	0.75	0.86	
MS	0.81	0.83	0.82	
PS	0.89	0.8	0.84	
S	0.64	0.88	0.74	
(c)AdaBoost coarse-fine frame-work based on BO				
C	**1**	0.9	0.95	0.832
CS	0.72	0.62	0.67	
FS	0.8	0.77	0.78	
M	0.96	0.96	**0.96**	
MS	0.73	0.85	0.79	
PS	0.95	0.79	0.86	
S	0.73	**1**	0.84	
(d) AdaBoost coarse-fine frame-work based on PL				
C	0.94	1	**0.97**	0.826
CS	0.6	0.53	0.56	
FS	0.88	0.79	0.83	
M	0.8	**0.91**	0.85	
MS	0.8	0.77	0.79	
PS	**0.95**	0.77	0.85	
S	0.67	0.89	0.76	

References

Jiang C, Zhang D, Chen S (2021) Lithology identification from well-log curves via neural networks with additional geologic constraint. Geophysics 86(5):85–100

Imamverdiyev Y, Sukhostat L (2019) Lithological facies classification using deep convolutional neural network. Journal of Petroleum Science and Engineering 174:216–228

Liu H, Wu K, Xu H, Xu Y (2021) Lithology classification using tasi thermal infrared hyperspectral data with convolutional neural networks. Remote Sensing 13(16):3117

Li D, Zhao J, Liu Z (2022) A novel method of multitype hybrid rock lithology classification based on convolutional neural networks. Sensors 22(4):1574

Xu Z, Ma W, Lin P, Shi H, Pan D, Liu T (2021) Deep learning of rock images for intelligent lithology identification. Computers & Geosciences 154:104799

Xie Y, Zhu C, Zhou W, Li Z, Liu X, Tu M (2018) Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances. Journal of Petroleum Science and Engineering 160:182–193

Sun J, Li Q, Chen M, Ren L, Huang G, Li C, Zhang Z (2019) Optimization of models for a rapid identification of lithology while drilling-a win-win strategy based on machine learning. Journal of Petroleum Science and Engineering 176:321–341

Xie Y, Zhu C, Lu Y, Zhu Z (2019) Towards optimization of boosting models for formation lithology identification. Mathematical Problems in Engineering 2019

Xie Q, Luong M-T, Hovy E, Le QV (2020) Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 10687–10698

Ouali Y, Hudelot C, Tami M (2020) n overview of deep semi-supervised learning. arXiv preprint arXiv:2006.05278

Kim G (2021) Recent deep semi-supervised learning approaches and related works. arXiv preprint arXiv:2106.11528

Zhou K, Li S, Liu J, Zhou X, Geng Z (2021) Sequential data-driven cross-domain lithology identification under logging data distribution discrepancy. Measurement Science and Technology 32(12):125122

Li G, Qiao Y, Zheng Y, Li Y, Wu W (2019) Semi-supervised learning based on generative adversarial network and its applied to lithology recognition. IEEE Access 7:67428–67437

Zhao F, Yang Y, Kang J, Li X (2023) Ce-sgan: Classification enhancement semi-supervised generative adversarial network for lithology identification. Geoenergy Science and Engineering 223:211562

Ren Q, Zhang H, Zhang D, Zhao X, Yan L, Rui J, Zeng F, Zhu X (2022) A framework of active learning and semi-supervised learning for ithology identification based on improved naive bayes. Expert Systems with Applications 202:117278

Hong Z, Yao J, Li K, Hu G (2022) Conjunction of active and semi-supervised learning for wireline logs-based automatic lithology identification. IEEE Geoscience and Remote Sensing Letters 19:1–5

Li Z, Kang Y, Feng D, Wang X-M, Lv W, Chang J, Zheng WX (202) Semi-supervised learning for lithology identification using laplacian support vector machine. Journal of Petroleum Science and Engineering 195, 107510

Zou Y, Chen Y, Deng H (2021) Gradient boosting decision tree for lithology identification with well logs: a case study of zhaoxian gold deposit, shandong peninsula, china. Natural Resources Research 30(5):3197–3217

Liashchynskyi P (2019) Grid search, random search, genetic algorithm a big comparison for nas arXiv preprint arXiv:1912.06059

Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. Journal of machine learning research 13(2)

Ren Q, Zhang H, Zhang D, Zhao X (2023) Lithology identification using principal component analysis and particle swarm optimization fuzzy decision tree. Journal of Petroleum Science and Engineering 220:111233

Hassan R, Cohanim B, De Weck O, Venter G (2005) A comparison of particle swarm optimization and the genetic algorithm. In: 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, p 1897

Sun Z, Jiang B, Li X, Li J, Xiao K (2020) A data-driven approach for lithology identification based on parameter-optimized ensemble learning. Energies 13(15):3903

Saporetti CM, da Fonseca LG, Pereira E (2019) A lithology identification approach based on machine learning with evolutionary parameter tuning. IEEE Geoscience and Remote Sensing Letters 16(12):1819–1823

Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. Advances in neural information processing systems 25

Rasmussen CE (2003) Gaussian processes in machine learning. In: Summer School on Machine Learning, pp 63–71. Springer

Rasmussen C, Williams C (2005) Gaussian processes for machine learning. In: GAUSSIAN PROCESSES FOR MACHINE LEARNING. Adaptive Computation and Machine Learning, pp 1–247

Brochu E, Cora VM, De Freitas N (2010) A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. arXiv preprint arXiv:1012.2599

Gan W, Ji Z, Liang Y (2021) Acquisition functions in bayesian optimization. In: 2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), pp 129–135

Dietterich TG (2000) Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems, pp 1–15. Springer

Hastie T, Rosset S, Zhu J, Zou H (2009) Multi-class adaboost. Statistics and its. Interface 2(3):349–360

Friedman JH (2001) Greedy function approximation a gradient boosting machine. Annals of statistics, 1189–1232

Breiman L (2001) Random forests. Machine learning 45(1):5–32

Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized tree. Machine learning 63(1):3–42

Xie Y, Zhu C, Hu R, Zhu Z (2021) A coarse-to-fine approach for intelligent logging lithology identification with extremely randomized trees. Mathematical Geosciences 53(5):859–876

Amini M-R, Feofanov V, Pauletto L, Devijver E, Maximov Y (2022) Self-training A survey. arXiv preprint arXiv:2202.12040

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.