A Coarse-to-Fine Approach for Intelligent Logging Lithology Identification with Extremely Randomized Trees

Yunxin Xie¹ · Chenyang Zhu² · Runshan Hu² · Zhengwei Zhu¹

Received: 29 April 2020 / Accepted: 27 July 2020 / Published online: 12 August 2020
© The Author(s) 2020

Abstract Lithology identification is vital for reservoir exploration and petroleum engineering. Recently, there has been growing interest in using an intelligent logging approach for lithology classification. Machine learning has emerged as a powerful tool in inferring lithology types with the logging curves. However, well logs are susceptible to logging parameter manual entry, borehole conditions and tool calibrations. Most studies in the field of lithology classification with machine learning approaches have focused only on improving the prediction accuracy of classifiers. Also, a model trained in one location is not reusable in a new location due to different data distributions. In this paper, a unified framework is provided for training a multi-class lithology classification model for a data set with outlier data. In this paper, a coarse-to-fine framework that combines outlier detection, multi-class classification with an extremely randomized tree-based classifier is proposed to solve these issues. An unsupervised learning approach is used to detect the outliers in the data set. Then a coarse-to-fine inference procedure is used to infer the lithology class with an extremely randomized tree classifier. Two real-world data sets of well-logging are used to demonstrate the effectiveness of the proposed framework. Comparisons are conducted with some baseline machine learning classifiers, namely random forest, gradient tree boosting, and xgboosting. Results show that the proposed framework has higher prediction accuracy in sandstones compared with other approaches.

Keywords Lithology classification · Ensemble methods · Outlier detection

---

✉ Chenyang Zhu
c.zhu@soton.ac.uk

¹ Changzhou University, Changzhou, China

² University of Southampton, Southampton, UK

1 Introduction

The role of lithology identification in mineral exploration and petroleum exploration has received increased attention across several disciplines in recent years. As the basis of reservoir characteristics research and geological modeling, lithology identification provides a reliable basis for measuring the spatial distribution of the mineral area (Rider 1986). Lithology identification is used in fields such as reservoir characterization, reservoir evaluation, and reservoir modeling in petroleum development and engineering. Thus it is vital to understand the lithology of the target layer in the geology and petroleum engineering industries.

At present, approaches such as gravity, well-logging, seismic, remote sensing, electromagnetic and geophysics, have been used in lithology identification. Well-logging is one of the most common practices for lithology identification in petroleum exploration. The geological information carried by well-logging data is an essential source for determining gas reserves and making gas exploration plans. There have been studies focusing on building statistical models for lithology identification from domain knowledge (Busch et al. 1987; Porter et al. 1969). However, the work involves much human work and the proposed model is subject to change due to different well-logging data distributions in different areas. In recent years, researchers have shown an increased interest in using artificial intelligence to predict lithology classes automatically with computer-aided tools in well-logging and drilling technologies. The machine learning algorithms, such as support vector machine, neural network, random forest (RF) and gradient tree boosting (GTB), not only reduce data analysis work for domain experts but also improve the lithology classification accuracy.

These machine learning-based approaches to lithology identification attempt to train a multi-class classifiers model based on a large amount of labeled well-logging data with logging curves such as natural gamma (GR) and compensated neutron log (CNL). Then the model can be used to predict the lithology classes with data that have the same feature space. This supervised learning approach generates a function that maps the feature space to a specific lithology class. However, the well logs are susceptible to logging parameter mistakes during manual entry, borehole conditions and tool calibrations. In addition, the logging measurements are affected by the gas effect in gas reservoirs. Under different gas saturations, the same lithology could have different distributions of logging curves. As different areas have different gas saturations, a model trained in one location might not be reusable in a new location due to different logging distributions. Thus a unified framework is needed to combine the outlier detection and the multi-class classifier and improve the prediction accuracy. Our previous work also concludes that although ensemble methods could help to improve the prediction accuracy compared with the other machine learning approaches, the sandstone classes are challenging to classify (Xie et al. 2018).

In this paper, a coarse-to-fine framework that combines outlier detection, multi-classification with an extremely randomized tree-based classifier is proposed to solve these issues. There are three main contributions in this paper. Firstly, the unsupervised learning approach, the local outlier factor (LOF) algorithm is used to identify outlier data. The parameter of the number of neighbors is tuned to get the best prediction result. Then a coarse-to-fine inference procedure is used to infer the lithology classes. The

samples are classified into general lithology classes first. Then the samples classified as sandstone class are classified into specific subclasses, such as pebbly coarse sandstones, coarse sandstones, medium sandstones and fine sandstones. Two extremely randomized tree classifiers are trained for the coarse and fine targets. Parameter tuning is used to get the optimal parameter setting for coarse and fine classification tasks. Finally, the framework is applied to two real-world well-logging data sets from the Daniudui gas field (DGF) and the Hangjinqi gas field (HGF). Results show that the LOF helps to improve prediction accuracy. Moreover, the performance of the proposed coarse-to-fine procedure with the optimized ensemble tree framework is compared with three other benchmarks, namely RF, GTB and xgboosting. Results show that the proposed framework helps us gain a higher prediction accuracy in classifying lithology classes, especially sandstone classes.

2 Related Work

Extensive research has shown that deep learning approaches could help classify lithology classes with well-logging data. Zhong et al. (2019) built a four-layer neural network to identify coals in a well at Surat Basin with logging-while-drilling data. Although their work delivered an overall accuracy of 96%, they were conducting binary classification. Our work mainly focuses on multi-class classification, which might result in lower prediction accuracy. Zhu et al. (2018) utilized the wavelet-decomposition approach to convert the lithology identification task into a supervised image recognition task. A convolution neural network (CNN) was then used to train the model to classify lithology classes. Chen et al. (2020) developed deep learning-based lithology classification models with drilling string vibration data. The data were preprocessed with noise reduction and time-frequency transformation. Then the CNN combined with Mobilenet (Howard et al. 2017) and ResNet (He et al. 2016) was used to train the model to predict complex formation lithologies such as fine gravel sandstone, fine sandstone and mudstone. However, the model, trained with deep learning algorithms, required a high-dimension feature space, while the feature space of well-logging data is limited. Thus, ensemble methods would perform better in classifying lithology classes when the feature space is limited (Xie et al. 2018).

Several studies have attempted to compare the performances of machine learning algorithms for lithology classification. Deng et al. (2019) compared three machine learning approaches, namely artificial neural network, support vector machine and RF for carbonate vuggy facies classification. Results showed that RF has the best classification accuracy in vug-size classification. Xie et al. (2018) evaluated five machine learning approaches, namely artificial intelligent network, support vector machine, naive Bayes, RF and GTB for lithology classification. Results showed that the ensemble methods perform better than the traditional ones when the feature space is limited. Built on their work, Dev and Eden (2018) applied AdaBoost and LogitBoost with random tree-based learners, achieving higher performance metrics. Xie et al. (2019) applied regularization on GTB and xgboosting and stacked the classifiers to improve the classification accuracy. Tewari and Dwivedi (2020) also showed that the heterogeneous ensemble methods, namely voting and stacking, could improve the prediction

accuracy for mudstone lithofacies in a Kansas oil-field area. Ao et al. (2019b) proposed a linear random forest (LRF) algorithm for better logging regression modeling with limited samples. Results showed that the LRF is robust, but subject to data errors. Their work mainly focused on regression results on formation properties. Our work proposes to use outlier detection to exclude data errors. Ao et al. (2019a) also proposed a pruning random forest (PRF) to identify sand-body from seismic attributes in the western Bohai Sea of China. Results show that PRF has better predictive performance and robustness. Also, several researchers have explored the improvement of ensemble methods for higher prediction accuracy. Asante-Okyerere et al. (2019) proposed a gradient boost model that is based on the Gaussian mixture model for lithofacies identification in South Yellow Sea's southern Basin. The proposed model has better prediction accuracy in classifying mudstone and siltstone compared with the classic gradient boosting approach. Based on the GTB algorithm, Saporetti et al. (2019) propose to use differential evolution to select the optimal parameter set to train the boosting model. Based on the RF algorithm, Ao et al. (2018) proposed to use mean-shift iterations to gather samples to local maximum points of the probability density. Then the RF algorithm was applied to the generated prototype similarity space for better classification results.

3 Methodology

3.1 Framework Overview

In this paper, a coarse-to-fine framework is proposed for multi-class lithology classification. As shown in Fig. 1, the raw well-logging data collected from different wells in the same area are passed into the LOF to exclude data points that are not measured correctly due to borehole conditions, tool calibrations or human errors. Then the processed data are randomly split into training samples and test samples. Each sample is a feature vector x_i formed with the values of logging curves at the same depth. Also, each sample is assigned with a label $y_i \in \Omega$ that indicates the corresponding lithology type at the same depth based on domain knowledge. Here Ω is used to present coarse or fine lithology targets. For example, Ω_c is defined as the general type of lithology classes, which includes sandstone (SS), carbonate rock (CR), coal (C), siltstone (S) and mudstone (M). So $\Omega_c = \{SS, CR, C, S, M\}$. Then Ω_f is defined as concrete lithology classes of sandstones, which includes fine sandstone (FS), medium sandstone (MS), coarse sandstones (CS) and pebbly coarse sandstone (PS). So $\Omega_f = \{FS, MS, CS, PS\}$.

Machine learning algorithms that build mathematical models based on training samples can be used to predict the labels of test samples with the same feature vector. In this paper, the supervised learning algorithm is used for multi-class lithology classification. The training samples are used in two training tasks as shown in Fig. 1, namely coarse target model training and fine target model training. For coarse target model training, the labels of training samples with concrete lithology labels $y_i \in \Omega_f$ are mapped to the coarse label SS. For fine target model training, only those training samples with the concrete lithology labels $y_i \in \Omega_f$ are used to train the model.

Fig. 1 A coarse-to-fine framework for multi-class lithology classification

In this paper, a randomized ensemble tree classifier is used to train the classified model. However, several parameters need to be tuned for the randomized ensemble tree classifier in different areas for the lithology classifier. The cross-validation technique can be applied to evaluate the classifier with different parameter sets. Then the optimal parameter set can be obtained based on the bias and variance scores of the models. Also, overfitting could be prevented with a cross-validation technique. Using the N -fold cross-validation technique, the training samples are partitioned to N sub-samples with equal size. Then for each validation procedure, the n th samples are selected as the test data and the rest of the samples are selected as training data. The model is trained with different parameter sets. This validation procedure is repeated N times. The final accuracy score is calculated by averaging the accuracy score for each procedure. The parameter set that has the best accuracy score is proved to be the optimal set that suits the corresponding data distributions.

Lastly, 20% of the test samples are fed into the generated coarse target model. The samples whose labels are predicted as *SS* are fed into the generated fine target model. The logging curves and the confusion matrixes then could be used to compare the performance between different models with the optimal parameter set.

3.2 Outlier Detection

Outlier detection aims to find the data samples that deviate from the distribution of the majority of the data (Ben-Gal 2005). These data samples might result from mistakes or contamination of manual entry for the logging parameters. Based on the dimension of the feature space, the outliers could be *univariate* or *multivariate*. Univariate outliers are those found in a single feature space, while multivariate outliers are those found in n -feature space. In the case study, a multivariate outlier is built to exclude divergent data samples that might be caused by tool calibrations or manual entry mistakes.

Breunig et al. (2000) proposed using an LOF to locate the anomalous data points by estimating the local deviation of the given data points with respect to its neighbors. An LOF is an unsupervised learning approach which uses the logging curves as feature values to filter out the outlier data. Given the training samples $(x_0, x_1, \dots, x_n)$ with size n , the LOF first defines the $kD(x_i)$ as the distance of x_i to the k th nearest neighbor. Based on $kD(x_i)$, the LOF uses $N_k(X_i)$ to denote the samples within $kD(x_i)$ distance. Then, the reachability distance between two data samples $RD(x_i, x_j)$ is defined as Eq. (1), where $D(x_i, x_j)$ denotes the distance between x_i and x_j . So the reachability distance between x_i and x_j is defined as the maximum of the distance between x_i and x_j , and the distance of x_j to the k th nearest neighbors.

$$RD(x_i, x_j) = \max\{kD(x_j), D(x_i, x_j)\}. \quad (1)$$

In light of reachability distance, the local reachability density of x_i is defined as Eq. (2), which denotes the inverse of average distance at which x_i could be reached from the neighbors.

$$LRD_k(x_i) = \frac{|N_k(x_i)|}{\sum_{x_j \in N_k(x_i)} RD(x_i, x_j)}. \quad (2)$$

Then the local reachability density of x_i could be used to compare with those of the neighbors using the LOF defined in Eq. (3). The value of $LOF_k(x_i)$ then can be used to infer whether x_i is similar to its neighbors. If $LOF_k(x_i)$ has a value of approximately 1, then it means x_i is similar to its neighbors. If the value is less than 1, then the sample is considered to be an inlier sample. If the value is larger than 1, then the sample is considered to be an outlier sample. The LOF technique could be used to detect outlier data samples effectively.

$$LOF_k(x_i) = \frac{\sum_{x_j \in N_k(x_i)} \frac{LRD_k(x_j)}{LRD_k(x_i)}}{|N_k(x_i)|}. \quad (3)$$

3.3 Extremely Randomized Trees

In supervised learning, variance and bias always exist as a model trained from samples cannot incorporate the data patterns precisely. Ensemble methods are used to generate a set of weak learners and combine the prediction results of these weak learners into the final result. Ensemble methods are proved to not only decrease the variance but also improve the prediction accuracy (Dietterich 2000).

Ensemble methods have two main categories, namely sequential ensemble methods and parallel ensemble methods. Sequential ensemble methods generate the prediction model sequentially by refining the weak learners with the ability to improve the prediction accuracy with lower bias and variance. The typical algorithms of sequential ensemble methods are AdaBoost (Hastie et al. 2009), GTB (Friedman 2002) and xgboosting (Chen and Guestrin 2016). Parallel ensemble methods generate independent weak learners and average their predictions. RF (Liaw and Wiener 2002) and

extremely randomized trees (Geurts et al. 2006) are typical algorithms for the parallel ensemble methods.

Both RF and extremely randomized trees are built on decision trees. The decision tree helps to divide the complicated classification problem into a set of decisions made by features. In each step of constructing a node of the decision tree, the feature that best classifies the training data would be the node. The steps are repeated recursively until all features are used to generate the decision tree. The Gini impurity is usually used to select the feature for the node. Given the feature space $\mathcal{F} = \{f_1, f_2, \dots, f_J\}$ with size J , one of the nodes in the decision tree would be $n \in \mathcal{F}$. Assume that the samples of size N in node n are divided into K classes of samples, and each sample is of size M_i . Then the Gini impurity of node n is shown in Eq. (4). For each step, the decision tree would search for the feature that causes the greatest reduction in $I(n)$ as the node. However, when the decision tree is built following this procedure, the final model would be overfitting to the model and lose generality for the test data.

$$I(n) = 1 - \sum_{i=1}^K \left(\frac{M_i}{N} \right)^2. \quad (4)$$

Both RF and extremely randomized trees introduce randomness into the model to avoid overfitting. Both algorithms generate a set of independent decision trees based on the features. Then the bagging technique is used to summarize the results of the set of decision trees and generate the final result. Here Algorithm 1 is used to illustrate the procedure. When training the data with the extremely randomized tree algorithm, an ensemble tree set is initialized as empty. Then, to build each decision tree T_i , a subset of X is used to get a decision tree with randomly selected features. The samples of X are drawn without replacement. Moreover, in the step shown in line 9 of Algorithm 1, the decision tree selects a random split to divide the parent node into two random child nodes. After creating M decision trees, the extremely randomized tree is built. During the test process, the x is fed into ET . For each tree T_i in ET , a prediction is y_i by each decision tree. Then the final decision is made by averaging the result of all the decision trees. The RF algorithm works with almost the same procedure except for line 8 and line 9 in Algorithm 1. However, for the RF, the samples that train the independent decision trees are drawn with replacement, so repetition of sample is allowed in the same decision tree. Also, the decision tree would select the best split to convert the parent node into child nodes based on the Gini impurity.

3.4 The Coarse-to-Fine Approach for Lithology Classification

Algorithm 2 shows our proposed coarse-to-fine approach for multi-class lithology classification. Given the training data (X, Ω) , the LOF is applied to exclude the abnormal data points. The processed data are obtained as (X_o, y_o) . Then the data are split into training samples (X_{train}, y_{train}) and test samples (X_{test}, y_{test}) randomly, where the training samples own 80%. Initially, labels of training data are mapped to coarse labels. Then the extremely randomized tree model is trained on the data X_{train}, y_c as the coarse target model clf_c . Also, samples whose labels are in the set

---

Algorithm 1 Extremely Randomized Trees Algorithm, modified from Geurts et al. (2006)

```

1: Given  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  with feature space  $\mathcal{F}$  where  $x_i \in X$  and  $y_i \in \Omega$ .
2: Given the number of decision trees  $M$  and max depth of each tree  $max\_depth$ 
3: procedure TRAIN( $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ )
4:    $ET \leftarrow \{\}$ 
5:   for  $i$  from 1 to  $M$  do
6:      $T_i \in X \rightarrow \Omega$ 
7:     while ( $dodepth(T) < max\_depth$ )
8:       Randomly select  $X_i \subset X$  without replacement
9:       Randomly select feature  $f \in \mathcal{F}$ 
10:      Use  $f$  as the node to construct tree
11:    end while
12:     $ET = ET \cup \{T_i\}$ 
13:  end for
14:  return  $ET$ 
15: end procedure
16: procedure TEST( $x$ )
17:  for  $i$  from 1 to  $M$  do
18:    Select decision tree  $T_i$  from  $ET$ 
19:     $y_i \leftarrow T_i(x)$ 
20:  end for
21:   $y = \frac{\sum_{i=1}^M y_i}{M}$ 
22:  return  $y$ 
23: end procedure

```

Algorithm 2 The Coarse-to-fine Approach for Lithology Classification

---

```

1: Given  $(X, \Omega) = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  where  $x_i \in X$  and  $y_i \in \Omega$ .
2:  $\Omega_c = \{SS, CR, C, S, N\}$ ,  $\Omega_f = \{FS, MS, CS, PS\}$ ,  $\mathcal{G} \in \Omega \rightarrow \Omega_c$ 
3:  $(X_o, y_o) \leftarrow LOF(X, \Omega)$ 
4:  $(X_{train}, y_{train}), (X_{test}, y_{test}) \leftarrow train\_test\_split(X_o, y_o, 0.8)$ 
5:  $y_c \leftarrow \mathcal{G}(y_{train})$ 
6:  $clf_c \leftarrow Train(X_{train}, y_c)$ 
7: Select  $(X_f, y_f)$  from  $(X_{train}, y_{train})$  such that  $\forall y \in y_f \Rightarrow y \in \Omega_f$ 
8:  $clf_f \leftarrow Train(X_f, y_f)$ 
9:  $y\_predc \leftarrow clf_c.Test(X_{test})$ 
10: Select  $(X_t, y_t)$  from  $(X_{test}, y_{test})$  where  $clf_c.Test(X_t) = SS$ 
11:  $y\_predf \leftarrow clf_f.Test(X_t)$ 
12: Calculate accuracy of the model

```

Ω_f are extracted to train the fine target model clf_f . The test data are fed into the coarse target model clf_c and get the corresponding predictions $y\_predict$. Then the test samples whose labels are predicted as SS are applied to the fine target model clf_f to get the concrete lithology classes. Hyperparameter tuning with cross-validation is used in the model training process. Detailed descriptions are provided in the experiment section.

4 Experiments

To examine the proposed framework, extensive experiments were performed to compare the performance of the framework with some benchmarks by using well-logging data collected from multiple wells in DGF and HGF. In this section, we describe how the feature space and the corresponding labels for the data set were used to train the model. Then the performance of outlier detection is presented by comparing the classification accuracy of using and not using an LOF. Next, the model training and hyperparameter tuning process is presented. Finally a brief analysis of our framework compared with other benchmarks with confusion matrixes and logging curves is used to demonstrate the advantages of our framework.

4.1 Experiment Settings

Seven exploratory wells from DGF and seven exploratory wells from HGF were selected for model training and result validation. In the experiments, seven lithology classes, namely coarse sandstone (CS), medium sandstone (MS), fine sandstone (FS), pebbly coarse sandstone (PS), mudstone (M), siltstone (S) and coal (C), were the lithology classes to be classified. Seven logging curves shown in Table 1 were used as the feature space to classify lithology. Each sample in the well had a 7-dimension feature vector and a corresponding lithology class as the label. The data sets from DGF and HGF were used to validate our approach. In DGF, six wells that contain 896 well-logging samples were used to train and validate the model. Well ‘D17’ in DGF with 19 continuous well-logging data samples was used to present the predicted lithology classes with logging curves. In HGF, six wells that contain 1,225 well-logging samples were used to train the model in HGF. Well ‘J66’ with 13 geologically continuous samples was used to present the predicted lithology classes. DGF and HGF samples were from separate areas geospatially, thus the proposed framework could be used to train the model twice to fit the data patterns in these two areas.

Take the procedure in DGF as an example. The data were preprocessed with the LOF algorithm. The number of neighbors and the value of contamination were chosen based on the validation score. The performance of the LOF was compared with two other baselines, namely the classifier without outlier detection and the classifier with isolation forest outlier detection. Then the data were split into a training set and a test set. 80% of the whole data set was taken as the training data and 20% was taken as the test data. For the training set, fivefold cross-validation was used to tune the parameters of the extremely randomized tree classifier. The optimal parameter set was chosen for both areas. Finally, the performance of the proposed framework was compared to three other baselines, namely RF classifier, GTB classifier and xgboosting classifier. The confusion matrixes of test data were provided to compare the prediction accuracy of different classifiers. Also, the predicted lithology classes with logging curves were presented in well ‘D17’. The same procedure was repeated in HGF to validate that our proposed approach could be applied to different areas with independent geological patterns.

Table 1 Logging curves with abbreviation and descriptions

Logging parameter (unit)	Abbreviation	Descriptions
Gamma-ray (API)	GR	Natural gamma
Acoustic ($\mu\text{s/m}$)	AC	Sonic delta T
Density (g/cm^3)	DEN	Formation's bulk density
Compensated neutron (%)	CNL	Thermal and epithermal neutron
Deep lateral (Ωm)	LLD	Deep formation resistivity
Shallow lateral (Ωm)	LLS	Shallow formation resistivity
Caliper (cm)	CAL	The diameter and shape of the formation

4.2 Outlier Detection

For the LOF outlier detection technique, two parameters needed to be tuned to get the optimal parameter set to fit the regions where the training data were the most concentrated. The first parameter was the number of k -nearest neighbors. The LOF measures the density deviation of the samples regarding its neighbors, so the number of neighbors highly affects the performance of outlier detection. The second parameter was the amount of contamination of the data set, which determines the proportion of outliers in the data set. Figure 2 shows the validation curves of different parameter settings with fivefold cross-validation. Here the accuracy shown in Eq. (5) is used as the evaluation matrixes. The notation y is used to denote the true labels and $\hat{y}$ as the predicted labels. The data were first preprocessed with the LOF with a different parameter set and trained with the extremely randomized tree classifier for the coarse lithology targets. As can be seen from the figure, the average accuracy of testing sets achieved the highest score when the number of neighbors was set to 3 and the contamination of the data set was set to 0.05. This parameter set was then used as the parameter for LOF outlier detection.

$$\text{accuracy}(y, \hat{y}) = \frac{\sum_{i=0}^{N-1} 1(\hat{y}_i = y_i)}{N}. \quad (5)$$

Table 2 shows the comparison of accuracy scores for the data preprocessed with LOF and two other baselines, namely the classifier without outlier detection and the classifier with isolation forest outlier detection. The data were preprocessed with different outlier detection techniques, then the same classifier was applied to train the model with the coarse target. The test set was used to compare the performance of different outlier detection techniques. Results show that the model with the data preprocessed with the LOF achieved the highest prediction accuracy of 97.4%. The model without outlier detection or using the isolation forest outlier detection achieved lower prediction accuracy of 96.4%. The results show that the LOF was sufficient to improve the prediction accuracy for lithology classification.

Fig. 2 Validation curves to compare performance of LOF with different parameter sets

Table 2 Comparison between different outlier detection approaches

Algorithm	No outlier detection	LOF	Isolation forest
Accuracy	0.964	0.974	0.964

4.3 Model Training and Parameter Tuning

Suitable parameters of supervised models are vital for the classifiers to achieve high prediction accuracy. Hyperparameters could be used to select the parameters of the supervised model from a search range with some evaluation matrixes. This section mainly focuses on improving the robustness of the extremely randomized tree by tuning the parameters of the model with fivefold cross-validation. Accuracy is used as the evaluation index.

Table 3 shows the parameters needed to be tuned for the extremely randomized tree model with the coarse target data set and fine target data set. However, the trained model should not be too tightly fit to the training set as it will lose its predictive ability for the test data. Overfitting usually occurs when the model is too fitted to the training data. The validation scores generated from the cross-validation results could help in selecting the optimal parameter set that does not cause model overfitting or underfitting. Figure 3 shows the validation scores of the different parameter sets of extremely randomized trees with the fine target data set in DGF and HGF. The validation scores of the model with the coarse target data set have similar patterns. As can be observed, the performance of the randomized tree model was not greatly

Table 3 Hyperparameter tuning for extremely randomized trees model

Training data	Tuned parameter	Search range	Optimal parameter setting
Fine target	Number of trees	[50, 300]	200
	Number of features	[1, 7]	4
	Maximum depth of each tree	[5, 40]	20
	Minimal samples to split on feature	[2, 8]	3
Coarse target	Number of trees	[50, 300]	200
	Number of features	[1, 7]	4
	Maximum depth of each tree	[5, 40]	25
	Minimal samples to split on feature	[2, 8]	3

affected by the number of trees or the maximum number of features to consider when splitting the data with random features. However, the maximum depth of each tree and the minimal number of samples required to split the node did affect the predictive accuracy considerably. Both the training score and the cross-validation score were low when the maximum depth of the tree was lower than 15. When the maximum depth reached 15, the training score held at 1 and the cross-validation score vibrated. Also, the cross-validation score decreased with the increase of the minimal number of samples required to split the node. Based on the cross-validation scores, hyperparameter tuning with grid-search was applied to find the optimal parameter set within the search range. The optimal parameter set for the fine target and coarse target is provided in Table 3. The optimal parameter set was then used to train clf_c and clf_f in Algorithm 2.

4.4 Result Analysis

In this section, the effectiveness of the proposed coarse-to-fine framework is examined against the baseline classifiers, namely RF, GTB and xgboosting. All the classifiers were trained with data preprocessed with the LOF outlier detection technique. As mentioned earlier, 20% of the data in DGF and HGF were used as the test data. After training the model with 80% of the data, the model was fed with 180 samples from the DGF area and 245 samples from the HGF area to get the predicted lithology classes. The results are exhibited in Table 4, Figs. 4 and 5.

As can be observed from Table 4, there were more training samples from the HGF area than from the DGF area. Thus the model performance in HGF was better than in the DGF area, overall. The prediction accuracy of the proposed coarse-to-fine frame-

Fig. 3 Cross-validation scores of different parameter set of extremely randomized trees model

Table 4 Prediction accuracy for different classifiers in DGF and HGF area

Area	Coarse-to-fine framework	Random forest	Gradient tree boosting	XGBoosting
DGF	0.894	0.814	0.789	0.832
HGF	0.911	0.844	0.831	0.809

work was 89.4% in the DGF area and 91.1% in the HGF area, which was the highest value compared with all the other baseline classifiers. Figures 4 and 5 show the confusion matrixes on the DGF test set and HGF test set with four classifiers. From the observations, all the classifiers have good ability to distinguish classes of siltstone (S), mudstone (M) and coal (C). Taking the coarse-to-fine framework as an example, 100% of coal, 96.9% of mudstone and 88.9% of siltstone were classified into the correct class. Moreover, it is the same for other classifiers in both areas. However, in the comprehensive comparison of the confusion matrixes of sandstones, our proposed coarse-to-fine framework improved the prediction accuracy for sandstones significantly. Taking the test data in the DGF area as an example, 15.8% and 10.5% of the PS class would have been misclassified to CS and MS classes with the RF classifier. The prediction accuracy for the CS class is even less than 50% with the RF classifier. The same issues occurred in the GTB classifier and xgboosting classifier. The prediction accuracy of CS was 35.3% with the GTB classifier, and 41.2% of CS would have been misclassified to MS in the xgboosting classifier. For the three baseline classifiers, the CS class was most easily misclassified to the MS class. Likewise, the PS class was prone to be misclassified to CS and MS classes. However, the prediction accuracy for sandstones improved when the proposed coarse-to-fine framework was applied. In the DGF area, the prediction accuracy of the CS class was 64.7%, which was the highest among all the classifiers, and only 29.4% of the CS class was misclassified to the MS

Fig. 4 Confusion matrix on the DGF test set with: **a** extremely randomized trees with coarse-to-fine approach; **b** random forest; **c** gradient tree boosting; **d** XGBoosting

class compared with the over 40% misclassification rate of the other classifiers. In the HGF test set, the same results were obtained; 7.4% and 11.1% of FS class would have been misclassified to PS and CS classes with the RF classifier and GTB classifier. Furthermore, the prediction accuracy of the FS class with the xgboosting classifier was only 63.0%. However, with our proposed framework, the prediction accuracy of the FS class achieved 92.6%. Also, the prediction accuracy of all the sandstone classes was higher than those classified with other baseline classifiers. Based on the observations, it could be concluded that the coarse-to-fine framework not only helps to improve the prediction accuracy for the multi-class lithology identification problem when the feature space is limited but also improve the classification ability for sandstone classes. Also, more well-logging training data could help to improve model performance.

The visualizations of the predicted lithology classes with well-logging curves are provided in Figs. 6 and 7. It can be observed from the figures that the proposed coarse-to-fine framework was able to identify the lithology classes with high accuracy. In

Fig. 5 Confusion matrix on the HGF test set with: **a** extremely randomized trees with coarse-to-fine approach; **b** random forest; **c** gradient tree boosting; **d** XGBoosting

well D17 in the DGF area, the prediction accuracy achieved 100%. In well J66 in the HGF area, all the lithology classes were sandstone classes. One CS class was misclassified to the PS class, and a PS class was misclassified to the MS class. The results indicate that our proposed framework works well for the intelligent logging lithology identification problem. Our framework achieved better performance than the baseline classifiers, and our framework outperformed other classifiers on the sandstone lithology classification.

5 Conclusions

In this paper, a coarse-to-fine framework that integrates outlier detection and extremely randomized trees technique was proposed for intelligent logging lithology identification. The framework mainly addressed three issues in intelligent lithology

Fig. 6 Logging curves and prediction results of coarse-to-fine approach on dataset in well D17 in DGF area

Fig. 7 Logging curves and prediction results of coarse-to-fine approach on dataset in well J66 in HGF area

classification. First, the number of logging curves is limited. Second, during the drilling process, the well logs are susceptible to logging parameter manual entry, borehole conditions, tool calibrations. Third, sandstones classes are difficult to classify with traditional classifiers. In the proposed framework, a local outlier function, or LOF, was first used to exclude the data samples that deviated far from other training samples, then the model was trained with a coarse task and a fine task with extremely randomized trees. Hyperparameter tuning with cross-validation was used to obtain the optimal parameter set for the model. Experiments were conducted in two real-world case studies by comparing the prediction accuracy of our proposed framework with three other baseline classifiers. Results showed that LOF outlier detection helps to

improve the prediction accuracy of the model. The proposed framework outperformed the other three classifiers with prediction accuracy 89.4% and 91.1% in the DGF and HGF areas, respectively. Our proposed framework has the capability to distinguish sandstone classes with high accuracy. Observations also show that more data could help to improve the performance of the model.

Acknowledgements Funding was provided by National Natural Science Foundation of China (Grant No. 61772090) and China Scholarship Council (Grant No. 201708060147).

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

- Ao Y, Li H, Zhu L, Ali S, Yang Z (2018) Logging lithology discrimination in the prototype similarity space with random forest. *IEEE Geosci Remote Sens Lett* 16(5):687–691
- Ao Y, Li H, Zhu L, Ali S, Yang Z (2019a) Identifying channel sand-body from multiple seismic attributes with an improved random forest algorithm. *J Pet Sci Eng* 173:781–792
- Ao Y, Li H, Zhu L, Ali S, Yang Z (2019b) The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. *J Pet Sci Eng* 174:776–789
- Asante-Okyerere S, Shen C, Ziggah YY, Rulegeya MM, Zhu X (2019) A novel hybrid technique of integrating gradient-boosted machine and clustering algorithms for lithology classification. *Nat Resour Res* 29:2257–2273
- Ben-Gal I (2005) Outlier detection. In: Maimon O, Rokach L (eds) *Data mining and knowledge discovery handbook*. Springer, Berlin, pp 131–146
- Breunig MM, Kriegel HP, Ng RT, Sander J (2000) LOF: identifying density-based local outliers. In: *Proceedings of the 2000 ACM SIGMOD international conference on management of data*, pp 93–104
- Busch J, Fortney W, Berry L (1987) Determination of lithology from well logs by statistical analysis. *SPE Form Eval* 2(04):412–418
- Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 785–794
- Chen G, Chen M, Hong G, Lu Y, Zhou B, Gao Y (2020) A new method of lithology classification based on convolutional neural network algorithm by utilizing drilling string vibration data. *Energies* 13(4):888
- Deng T, Xu C, Jobe D, Xu R (2019) A comparative study of three supervised machine-learning algorithms for classifying carbonate vuggy facies in the kansas arbuckle formation. *Petrophysics* 60(06):838–853
- Dev VA, Eden MR (2018) Evaluating the boosting approach to machine learning for formation lithology classification. In: Eden MR, Ierapetritou MG, Towler GP (eds) *Computer aided chemical engineering*, vol 44. Elsevier, Amsterdam, pp 1465–1470
- Dietterich TG (2000) Ensemble methods in machine learning. In: *International workshop on multiple classifier systems*. Springer, Berlin, pp 1–15
- Friedman JH (2002) Stochastic gradient boosting. *Comput Stat Data Anal* 38(4):367–378
- Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. *Mach Learn* 63(1):3–42
- Hastie T, Rosset S, Zhu J, Zou H (2009) Multi-class adaboost. *Stat Interface* 2(3):349–360
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
- Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint <https://arxiv.org/abs/1704.04861>*

- Liaw A, Wiener M (2002) Classification and regression by randomForest. *R News* 2(3):18–22
- Porter C, Pickett G, Whitman W (1969) A method of determining rock characteristics for computation of log data; the litho-porosity cross plot. *Log Anal* 10(06):1–19
- Rider MH (1986) *The geological interpretation of well logs*. Wiley, US
- Saporetti CM, da Fonseca LG, Pereira E (2019) A lithology identification approach based on machine learning with evolutionary parameter tuning. *IEEE Geosci Remote Sens Lett* 16(12):1819–1823
- Tewari S, Dwivedi UD (2020) A comparative study of heterogeneous ensemble methods for the identification of geological lithofacies. *J Petrol Explor Prod Technol* 10:1849–1868
- 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. *J Pet Sci Eng* 160:182–193
- Xie Y, Zhu C, Lu Y, Zhu Z (2019) Towards optimization of boosting models for formation lithology identification. *Math Probl Eng* 2019:1–13
- Zhong R, Johnson R, Chen Z, Chand N (2019) Coal identification using neural networks with real-time coalbed methane drilling data. *APPEA J* 59(1):319–327
- Zhu L, Li H, Yang Z, Li C, Ao Y (2018) Intelligent logging lithological interpretation with convolution neural networks. *Petrophysics* 59(06):799–810