Evaluation of machine learning methods for formation lithology identification: a comparison of tuning processes and model performances

Abstract

Identification of underground formation lithology from well log data is an important task in petroleum exploration and engineering. Recently, several computational algorithms have been used for lithology identification to improve the prediction accuracy. In this paper, we evaluate five typical machine learning methods, namely the Naïve Bayes, Support Vector Machine, Artificial Neural Network, Random Forest and Gradient Tree Boosting, for formation lithology identification using data from the Daniudui gas field and the Hangjinqi gas field. The input to each model consists of features selected from different well log data samples. To determine the best model to classify the lithology type, this study used validation curve to determine the parameter search range and adopted the hyper-parameter optimization method to obtain the best parameter set for each model. The performance of each classifier is also evaluated using 5-fold cross validation. The results suggest that ensemble methods are good algorithm choices for supervised classification of lithology using well log data. The Gradient Tree Boosting classifier is robust to overfitting because it grows trees sequentially by adjusting the weight of the training data distribution to minimize a loss function. The random forest classifier is also a suitable option. An evaluation matrix showed that the Gradient Tree Boosting and Random Forest classifiers have lower prediction errors compared with the other three models. Although all the models have difficulties in distinguishing sandstone classes, the Gradient Tree Boosting performs well on this task compared with the other four methods. Moreover, the classification accuracy is remarkably similar across the lithology classes for both the Random Forest and Gradient Tree Boosting models.

Key Words: lithology identification; supervised learning; gradient boosting; tuning parameter

1. **Introduction**

Geophysical well log data have advantageous characteristics such as high vertical resolution, good continuity and convenient data acquisition. Therefore, they are an important material resource for underground rock information. Lithology classification based on well log data is the basis of reservoir parameter calculations, and provide the foundation for geological research studies in such fields as sedimentary facies and the environment. Apart from the significance in formation evaluation and geological analysis, lithology interpretation also has practical value in reserve calculation at exploration stage and detailed reservoir description at development stage. Two ways to determine the lithologies and lithofacies are to make inferences from the cuttings obtained during drilling operations and through observation and analysis of the core samples taken from the underground formations (Salehi and Honarvar, 2014). However, these two approaches are not always reliable because different geologists may provide different interpretations ([Akinyokun, et al.,2009](#Akinyokun2009)). Given the constraints on the sample data, the trend has been toward the use of well log data, which can serve not only to predict general petrophysical parameters but also as a tool for sedimentologists and reservoir engineers ([Serra and Abbott, 1982](#Serra1982)). However, the well log data could be highly sampled and numerous, which can burden the geologist who must integrate the data with their workflow and interpret the lithology within certain time constraints ([Horrocks, et al., 2015](#Horrocks2015)).

Since the introduction of well logs, many mathematical methods have been used to predict lithology based on well log data ([Delfiner, et al.,1987](#Delfiner1987)). In recent years, using computer technology to automatically predict lithology is becoming an important aspect in well logging and drilling technologies. These computer technologies assist the geologists to avoid the unnecessary data analysis work and improve the lithology identification accuracy. Given the approaches that can identify different grain size of clastic rock with better accuracy, geologists can build better quantitative reservoir evaluation models of different grain size, which can also improve the precision of reservoir evaluation.

Several machine learning techniques have been introduced to lithology classification and identification, including the support vector machine, neural network and random forest classifiers. The support vector machine classification formulation is achieved using features selected based on fuzzy logic from well logs. This approach performs better than do probabilistic neural networks ([Al-Anazi, et al.,2010](#anLH2013)). Adopting the radial basis function kernel also improve the classification accuracy because it has been found to yield the minimum misclassification rate error (Sebtosheikh, et al.,2015). Sebtosheikh also concluded that it is beneficial to implement the normalized polynomial kernel function by using the optimum values obtained from a grid search technique to predict the lithology (Sebtosheikh and Salehi, 2015). The neural network technique has also been applied to lithology identification and recognition with well log data ([Rogers, et al.,1992](#Rogers1992)) by using the back-propagation method to obtain the patterns to identify the lithology. The neural network technique has also been compared with the alternating conditional expectations (ACE) method, and the ACE achieved a better performance ([Rafik, et al.,2016](#Rafik2016)). The random forest technique has mainly been applied to geological and geochemical data for lithology identification, and it outperforms other machine learning algorithms in this area ([Harris, et al.,2015](#Harris2015), [Cracknell, et al.,2012](#Cracknell2012), [Cracknell, et al.,2014](#Cracknell2014)). In addition to its classification performance, the random forest approach is also able to generate reliable first-pass predictions for practical geological mapping applications ([Cracknell, et al.,2014](#Cracknell2014)). Moreover, research has been conducted by adapting multi-agent machine learning and classifier combinations to learn rock facies sequences from wireline well log data with relatively high accuracy ([Gifford, et al.,2010](#Gifford2010)).

Existing research mainly focuses on improving the classification accuracy of different machine learning algorithms. However, different lithology features are used for the classification and comparisons between these studies are difficult. Less work has been done to compare the performance of different machine learning models. This study conducts a rigorous comparison of five machine learning algorithms, namely Naïve Bayes (NB), Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF) and Gradient Tree Boosting (GTB). These algorithms are fed with the well log data from two gas fields in the Ordos Basin to identify lithology classes. We explore the search range of parameters and apply the grid search technique to find the best parameter set for lithology identification. The results are evaluated based on the metrics of precision, recall, F1-score and classification accuracy. The performance of different algorithms is examined based on the selected features of the well log data. Ensemble methods are reported to perform well in identifying lithology classes with small feature set, especially the sandstone class that other algorithms have difficulties to distinguish.

1. **Methodology**
   1. Machine Learning

Machine learning has become increasingly popular because it represents a major improvement in computers, allowing them to learn rather than being programmed explicitly. In the past decade, machine learning has been used in areas such as natural language processing, speech recognition, stock prediction, and many others. Machine learning algorithms produce general hypotheses from externally supplied instances and make predictions on future instances ([Kotsiantis,2007](#Kotsiantis2007)). Supervised learning is a machine learning task that generates a prediction model from labelled training data. Each training data example consists of a set of features and a desired output value. The supervised learning algorithm analyses the training data and generates a model that can be used to predict new examples from the same type of feature vector. Supervised learning can be divided into two major categories, namely, supervised regression and supervised classification. In this study, we used a labelled dataset with a feature vector consisting of seven features from several wells with which we want to perform lithology identification. The five supervised classification approaches are used to solve this multiclass classification problem.

* 1. Approaches
     1. Naïve Bayes

Naïve Bayes (NB) is a classification technique based on Bayes’ theorem under the assumption that the features are conditionally independent given a class label ([Sammut, 2011](#Sammut2011)).Assuming that there are *m* classes and *n* features, according to Bayes’ theorem, the probability that a new example belongs to class *i* is given by (1). With the naïve independence assumption that , (1) can be simplified to (2). Because is a constant, we can derive (3). We can use maximum a posteriori estimation to estimate and . For multiclass classification, we can compute the probability that the sample represents each class and pick the class with the largest probability. However, the NB algorithm depends on the assumption that features are independent from each other. In real situations, the features may have specific relations with each other. Therefore, the NB is usually used as a baseline for classification problems.

* + 1. Support Vector Machine

The SVM is a supervised machine learning algorithm that constructs a hyperplane or set of hyperplanes to distinguish between instances of different classes ([Cortes, et al.,1995](#Cortes1995)). Assuming that the margin of a hyperplane is the distance of the closest instance to the hyperplane, the SVM model should have hyperplanes that divide the classes as widely as possible. The goal of the SVM is then to maximize the margin. Given the training dataset with a feature vector of dimension , the classification is given by , where with being the number of training data. A linear classifier is given in (4) where is the hyperplane normal weight vector. A rescaled hyperplane classifier satisfies (5) and the weight vector is determined from (6) ([Kecman, 2005](#Kecman2005)). The Lagrange function in (7) can be used to solve (6), which arrives at and . Here, are the Lagrange multipliers for each inequality constraint.

.

In cases of non-separable data, a slack variable is added to each example; then, (5) becomes (8). A penalty constant C is added to the , forming (9). The penalty constant C weights the importance of misclassifying each training example. For smaller values of C, the optimization will choose a large-margin hyperplane that might misclassify more points. In practice, the parameter C is typically varied over a wide range of values and the optimal performance is assessed using a separate validation set or through cross-validation to verify the performance using only one training set ([Shawe-Taylor, et al.,2004)](#Shawe2004).

Kernel functions allow the SVM to separate non-linearly separable support vectors using a linear hyperplane ([Hsu, et al.,2002](#Hsu2002)). The radial basis function (RBF) kernel is a commonly used kernel function for training nonlinear SVM classifiers ([Chang, et al.,2010](#ChangYW2010)). Assume that the feature map is , such that . Then, the RBF kernel is shown in (10). Here, we use to simplify the equation. By applying the Lagrange function, we can obtain (11), where is the solution to the dual problem in (12), Finally, we arrive at the prediction, given by (13).

The SVM algorithm is often used to handle binary classification problems. To extend the SVM to make it suitable for multi-class classification, a one-against-one method is suggested by Hsu, in which a binary model is first created for all the expected classes, and the prediction is based on the highest confidence score of the binary model on that class compared to all the other classes ([Hsu, et al.,2002](#Hsu2002)).

* + 1. Artificial Neural Network (ANN)

An ANN is a computer model that mimics the workings of the human brain to learn from examples and obtain solutions to complex decision and classification problems ([An, 2013](#anLH2013)). Given a training dataset with a feature vector of dimension *n*, which is classified by , an ANN uses a complex and non-linear form of hypotheses to interpret the predictions of new examples. Here, is a weight assigned to each and is the bias for the function. Assuming that one neuron takes the input , the output is as given by (14). Here, is a sigmoid activation function for each neuron. The ANN algorithm training process is shown in [Fig. 1](#Fig1). A forward propagation is taken to calculate the output value of the network. Here, denotes the layer of neural network, denotes the total weighted sum of the inputs to neuron in layer, and denotes the output of after the activation function. After the forward propagation step, a hypothesis is calculated based on the weights and activation functions. We then calculate the loss function with the goal of minimizing the loss function to train the network.

Because the loss functionis a non-convex function, optimization algorithms such as Stochastic Gradient Descent (SGD) or Adaptive Moment Estimation (Adam) can be used to update the weights of each neuron to reduce the local error. The SGD updates and, as shown in (15), where is the learning rate. The back-propagation algorithm can provide an efficient way to compute the partial derivatives to update the weights. A penalty for the local error of each layer is assigned to the neurons in the previous layer, and the weights are updated based on the penalty . To avoid becoming stuck in a local minimum during optimization, momentum is used to accelerate the SGD in the relevant direction ([Qian, 1999](#Qian1999)). Adam is another approach that computes adaptive learning rates for each parameter ([Kingma and Ba, 2014](#Kingma2014)). In this study, we compared different optimization approaches for classifying lithology.

%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
\documentclass{article}
\usepackage{algorithm}
\usepackage[noend]{algpseudocode}
\pagestyle{empty}
\renewcommand{\thealgorithm}{1}
\begin{document}
\begin{algorithm}
\caption{Artificial Neural Network Training Algorithm, modified from (Reed, 1999)}\label{euclid}
\begin{algorithmic}[1]
\Procedure{Froward Propogation}{}
\State $z^{l+1}=w^l \alpha^l+b^l$
\State $\alpha^{l+1}=f(z^{l+1})$
\EndProcedure
\Procedure{Calculate Loss Function}{}
\State $J(w,b)=\frac{1}{2} \|h_{w,b}(x)-y\|^2$
\EndProcedure
\Procedure{Backpropogation}{}
\State $calculate~partial ~derivatives~ of~ output~ layer$
\State $\delta_l=\frac{\partial}{\partial z_i^l}\frac{1}{2} \|h_{w,b}(x)-y\|^2=-(y_i-a_i^l)f'(z_i^l)$
\State $calculate ~partial~ derivatives~ of~ hidden~ layers~and~update~weights$
\State $for ~j=l-1;j>=2;j--$
\State $~~~\delta^j=((w^l)^\top \delta^{(j+1)})f'(z^j)$
\State $~~~\triangledown_{w^j} J(w,b)=\delta^{(j+1)} (\alpha^j)^\top $
\State $~~~\triangledown_{b^j} J(w,b)=\delta^{(j+1)} $
\State $end ~for$
\EndProcedure
\end{algorithmic}
\end{algorithm}
\end{document}

**Fig. 1.** Artificial neural network training algorithm, taken from ([Reed, 1999](#Reed1999)) with different presentations

* + 1. Ensemble Methods

Ensemble methods are learning algorithms that construct a set of classifiers and classify new datasets by taking a weighted vote of their predictions ([Dietterich, 2000](#Dietterich2000)). There are two categories of ensemble methods: averaging methods and boosting methods. Averaging methods independently build several estimators and average their predictions. Because the voted prediction’s variance is reduced, its performance is better than that of a single base estimator. The RF is an averaging ensemble method introduced by Breiman based on the concept of a decision tree ([Breiman, 2001](#Breiman2001)). In contrast, boosting methods build estimators sequentially by reducing the bias of the combined estimator at each step, combining several weak models to produce a single powerful model. GTB is a boosting ensemble method that produces a prediction model using a collection of weak decision tree models ([Friedman,2002](#Friedman2002)). This paper compares the performance of these two approaches with other machine learning techniques.

The decision tree is one of the most intuitive models for classification problems. It can break down a complex decision-making process into a collection of simpler decisions ([Safavian and Landgrebe, 2002](#Safavian2002)). To build a decision tree, the feature that best divides the training data would be the root node of the tree. A feature that best divides the training data is one where the information gain and the entropy are higher than those of all the other features. For the remaining features, the same process is repeated recursively by splitting on the feature that has the highest remaining information gain and adding that as a child node of the current node. However, decision trees can occasionally cause overfitting issues ([Aha, et al., 1998](#Aha1998)).

The Random Forest is used to improve the prediction accuracy of decision trees. It combine the bagging technique ([Breiman,1996](#Breiman1996)) with the random feature selection technique ([Ho, 1998](#Ho1998)). The RF algorithm works as illustrated in [Fig. 2](#Fig2). The training dataset , with a feature vector of dimension , is classified by . The RF takes a random sample from the dataset and recursively constructs splits based on randomly selected features until the tree is as large as desired. Then, the remaining data are dropped down the tree to obtain the classes of leaves. This process is repeated several times to build the forest. Then, the training data is fed into the RF model and each instance is assigned to a final category based on the majority vote over the forest. Here, denotes the instance that is fed into each decision tree. The function represents the probability of the predicted class based on instance for each tree, and the function indicates the probability of the predicted class based on the random forest. The final category is determined by the larger probability of this observation. Because the sampled data and the selected features are randomly chosen from the dataset, each tree is independent, which leads to a low variance.

%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
\documentclass{article}
\usepackage{algorithm}
\usepackage[noend]{algpseudocode}
\pagestyle{empty}
\renewcommand{\thealgorithm}{2}
\begin{document}
\begin{algorithm}
\caption{Random Forest Training Algorithm}\label{euclid}
\begin{algorithmic}[1]

\State$for~i~from~1~to~M$
\State $~~x \in x_i$
\State $~~forest=forest\bigcup RandomizedTree(x)$
\State$end~for$
\State $Assign~each~instance~to~a~final~category~based~on~a~majority~vote~over~forest$
$~~P(c|v)=\frac{1}{T}\sum_{t=1}^{T}P_t(c|v)$

\Function{RandomizedTree(x)}{}
\State $F \in Features(x)$
\State $while~Tree.size<MaxTreeSize$
\State$~~~f \in F \wedge f=MaxInformationGain(x)$
\State$~~~Tree.splitOn(f)$
\State$return~Tree$
\EndFunction
\end{algorithmic}
\end{algorithm}
\end{document}

**Fig. 2.** Random forest training algorithm

Distinct from the RF, the GTB algorithm sequentially generates base models. This emphasizes the instances that are difficult to estimate through the process of building better models. While building a new model, samples that are misclassified or incorrectly estimated in the previous base model will have a greater chance of being selected or assigned a higher weight. As a result, the new model will better predict these samples compared with the previous model. A loss function is usually used to measure the difference between the predicted and true values, which indicates how well the model fits the data. For the boosting process, a stepwise modelling approach can be used to add new base models that best reduce the loss function without changing the models that have already been sequentially added. Friedman proposed this generic gradient boosting method ([Friedman,2001](#Friedman2001)) and a refined generalized boosting of decision trees that uses a fixed-size regression tree as the base model to improve model quality ([Friedman,2002](#Friedman2002)). The algorithm works as shown in [Fig. 3](#Fig3). The training dataset with a feature vector of dimension *n* is classified by . is the loss function and is used to minimize the expected value of the loss function. The model is initialized with a constant value , and a gradient descent process is applied to minimize the loss function. Assume there are iterations. For each descent step, is calculated to minimize . GTB partitions the input space into joint regions and predicts a constant value in each region . In each step, the multiplier is calculated to minimize the loss function and the model is updated with this multiplier. However, overfitting can occur when the number of iterations is too large, which leads to poor generalization ability. The model becomes overfit to the training data, causing a reduction in the prediction accuracy on unseen data. Friedman’s algorithm adds a “shrinkage” parameter to control the learning rate of the procedure ([Friedman,2002](#Friedman2002)).

%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
%FontSize=10
%TeXFontSize=10
\documentclass{article}
\usepackage{algorithm}
\usepackage[noend]{algpseudocode}
\pagestyle{empty}
\renewcommand{\thealgorithm}{3}
\begin{document}
\begin{algorithm}
\caption{Generic Gradient Boosting and Gradient Tree Boosting Algorithm， modified from (Friedman, 2001) and (Friedman, 2002)}\label{euclid}
\begin{algorithmic}[1]
\State$F_0(x)=argmin_\gamma \sum_{i=1}^{N}L(y_i,\gamma)$
\State$for~m~from~1~to~M$
\State$~~z_{im}=-[ \frac{\partial L(y_i,F(x_i))}{\partial F(x_i)}]_{F(x)=F_{m-1}(x)}~~~for~i=1,...,n$
\State$~~h_m(x)=\sum_{j=1}^{J_m}b_{jm}I(x\in R_{jm})$
\State$~~\gamma_m=argmin_\gamma \sum_{i=1}^{N}L(y_i,F_{m-1}(x_i)+\gamma)$
\State$~~F_m(x)=F_{m-1}(x)+\rho \gamma_m~(x\in R_{jm})$
\State$end~for$
\end{algorithmic}
\end{algorithm}
\end{document}

**Fig. 3.** Gradient tree boosting algorithms, taken from ([Friedman, 2001](#Friedman2001)) and ([Friedman, 2002](#Friedman2002)), respectively with different presentation

* 1. Evaluation

Estimating the accuracy of a classifier can help in predicting its future accuracy ([Kohavi,1995](#Kohavi1995)). Bias and variance always exist in supervised machine-learning algorithms because a training dataset cannot usually include all possible examples. The main reason to use cross validation to evaluate a supervised model is to trade off bias for low variance and remove the impact of the pattern of the current dataset. This provides the selected supervised model with a more generalized data pattern with low bias and low variance, which causes the model to perform better in practice. For this case study, we want to construct a generalizable supervised model that can handle lithology classification; however, we have data from only several wells in a single area. To improve model usability, a 5-fold-cross-validation method is used to reduce the possibility of data overfitting and provide insight about how the model generalizes an independent dataset.

Using the cross-validation method, the original well log is randomly partitioned into five equally sized samples. This cross-validation process is repeated five times; each time, the nth sample is selected as the test data, and the remaining samples are selected as the training data. The five results from the entire process can then be averaged to produce an estimation. Here, a matrix of *precision*, *recall* and *F1* score is evaluated for the accuracy of the model. *Precision* is used to define the number of relevant items selected (the ratio of true positives () to all predicted positives ()). *Recall* is used to define the number of relevant items retrieved by the supervised model (the ratio of true positives () to all actual positives). The score, commonly used in information retrieval, measures the accuracy of both precision and recall. A good retrieval algorithm should maximize both precision and recall simultaneously. Hence, the is also be considered to be an evaluation metric.

* 1. Scikit-learn

Scikit-learn is a Python module that integrates a wide range of machine learning algorithms for both supervised and unsupervised problems ([Pedregosa, et al.,2011](#Pedregosa2011)). This case study implements the NB classifier and uses “SVC”, “RandomForestClassifier”, “MLPClassifier”, “GradientBoostingClassifier” from Scikit-learn module to create the supervised learning classifiers. In addition, pipelines for automating machine learning workflows are also used in this case study. Scikit-learn chains together a linear sequence of data transforms in a modeling process that can be evaluated.

1. **Case study**

Two tectono-stratigraphic sequences are present in the Paleozoic strata of the Ordos Basin, an Upper Paleozoic sequence containing terrestrial clastics and coals and Lower Paleozoic marine to non-marine sediments ([Dai, et al.,2005](#Dai2005)). The Upper Paleozoic reservoirs, which have relatively low porosity and permeability, are formed in a fluvial-deltaic depositional environment ([Fu, et al.,2013](#Fu2013)). The target formations are the Carboniferous Taiyuan, Permian Shanxi and Xiashihezi formations, which are the main gas-bearing formations. The well log data and core analysis reports used in this study were obtained from five wells in the Daniudui gas field (DGF), which is located in the eastern portion of the Yishan Slope of the Ordos Basin, and from seven wells in the Hangjinqi gas field (HGF), which is located in the north Ordos Basin. The core analyses report that the major lithology types include sandstone, mudstone, coal and small amounts of microcrystalline and micrite carbonate rock.

It has been proven that the sandstone reservoir of the Upper Paleozoic strata in the Ordos Basin is favorable for gas accumulation ([Li, et al.,2005](#Li2005), [Yang, et al.,2012](#Yang2012)). Therefore, it is important to study and further understand sandstone reservoirs. The grain size distribution of clastic rock provides vast amounts of information on sedimentary environments through geological time. The clastic rocks in this study are divided into pebbly sandstone (＞1 mm grain size), coarse sandstone (0.5 ~1 mm grain size), medium sandstone (0.25-0.5 mm grain size), fine sandstone (0.01 - 0.25 mm grain size), siltstone (0.005 - 0.05 mm grain size) and mudstone (＜0.005 mm grain size), according to the detrital grain size classification of the oil industry standard of China ([Zhu,2008](#Zhu2008)). In this study, carbonate rock (CR), coal (C), pebbly sandstone (PS), coarse sandstone (CS), medium sandstone (MS), fine sandstone (FS), siltstone (S) and mudstone (M) are the eight target lithology classes to be identified.

A total of 2,153 log readings corresponding to the exact values of the core depth, and seven log parameters are used to identify the lithology. The seven log parameters are the gamma ray log (GR), acoustic log (AC), density log (DEN), compensated neutron log (CNL), deep latero log (LLD), shallow latero log (LLS) and caliper log (CAL) ([Table 1](#Table1)). The samples are randomly split into a training set and a test set. The training set is used to develop the model to perform lithology classification and the test set is applied to the trained model to estimate how well the model has been trained and to obtain model properties such as classification errors, precision and recall.

**Table 1**

Wireline logs used in this article

|  |  |  |
| --- | --- | --- |
| Parameter | Nomenclature | Unit |
| GR | gamma ray log | API |
| AC | acoustic log | μs/m |
| DEN | density log | g/cm3 |
| CNL | compensated neutron log | % |
| LLD | deep latero log | Ωm |
| LLS | shallow latero log | Ωm |
| CAL | caliper log | cm |

1. **Methods**
   1. Pre-processing

915 log readings from five wells in the DGF and 1,238 log readings from seven wells in the HGF are collected to perform the evaluation. Each data point is preprocessed with seven features: GR, AC, DEN, CNL, LLD, LLS and CAL. These collected data are transformed to a matrix of rows for the number of instances and seven columns containing the described features; one column contains the labeled type of rock. The data is randomly split with 80% used as the training set and 20% used as the testing set. Finally, there are 732 instances of training data and 183 instances of test data from the DGF, while there are 990 instances of training data and 248 instances of test data for the HGF. We use the data from the DGF to determine the parameters for each model and evaluate the machine learning methods using the datasets from both areas.

* 1. Classification model training

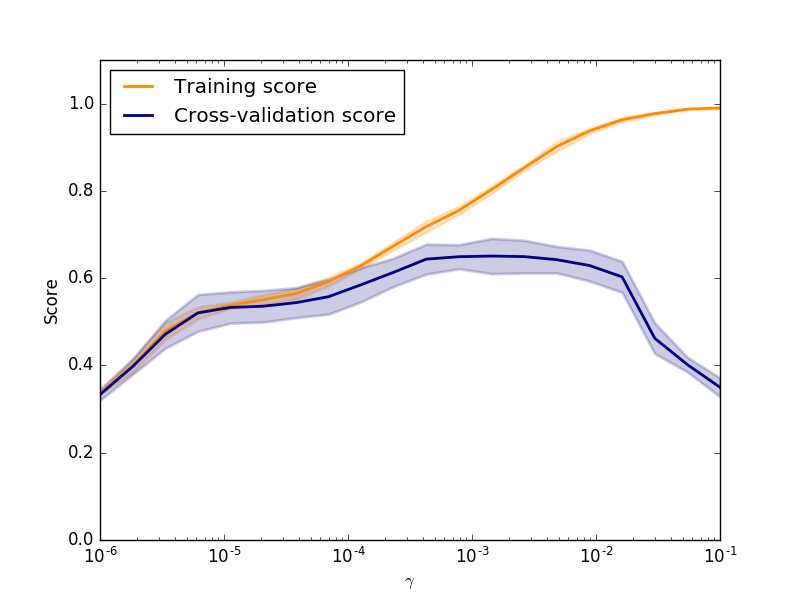
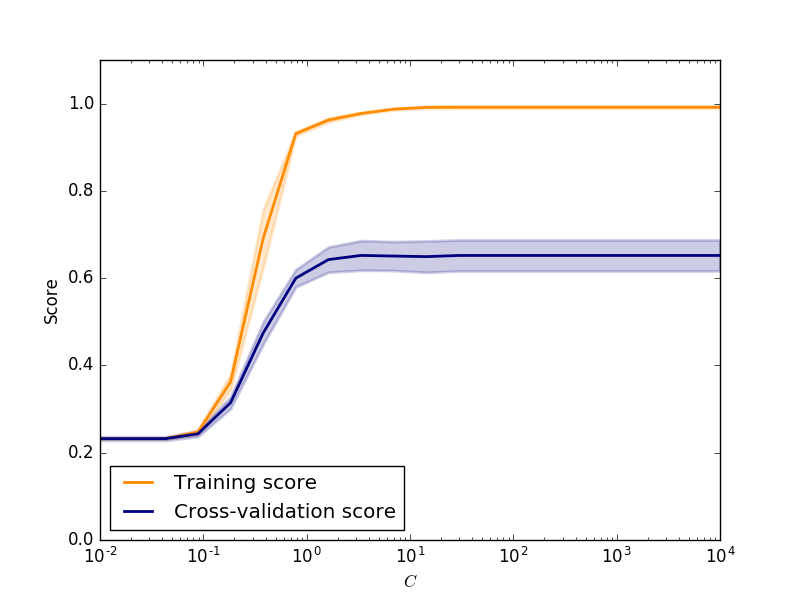
Appropriate parameter values are essential if the supervised models mentioned above are to obtain high classification performances on unseen examples. A robust parameter selection process (tuning) is used in this case study to obtain improved models that represent the data characteristics. Tuning is a process that uses a performance metric to rank the classifiers with different parameters to optimize a parameter for each specific model. Here, accuracy is used as a metric to measure the classification performance and grid search is used as a general method to optimize the parameter that could best train the model. For each model, the parameters that require tuning are given in [Table 2](#table2).

**Table 2**

Tuned parameters for each supervised model and tuned optimum parameter value

|  |  |  |  |
| --- | --- | --- | --- |
| Supervised model | Tuned parameters | Search range | Optimal parameter setting |
| SVM | Penalty parameter of the error term (C)  Kernel coefficient for ‘RBF’ () | 0.1-1000  0.0001-0.01 | 1000  0.0001 |
| ANN | learning rate  maximum number of learning iterations (max iter)  solver for weight optimization (solver) | 0.0001-0.001  200-1000  SGD, Adam | 0.001  1000  Adam |
| RF | The minimum number of samples required to split an internal node (min samples split)  The maximum depth of the tree (max depth)  The minimum number of samples required at a leaf node (min samples leaf)  The number of trees in the forest (n estimators) | 0-10  5-20  0-20  100-1000 | 2  12  1  150 |
| GTB | learning rate to shrink the contribution of each tree  The minimum number of samples required at a leaf node (min samples leaf)  maximum depth of the individual tree (max depth)  The number of boosting stages (n estimators)  The minimum number of samples required to split an internal node (min samples split) | 0.001-0.8  5-50  10-100  300-900  5-50 | 0.3  20  50  900  25 |

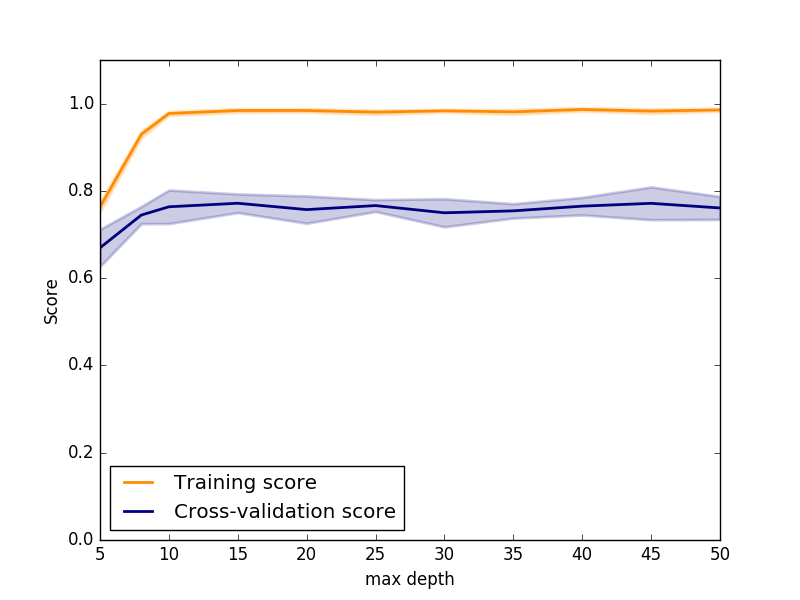
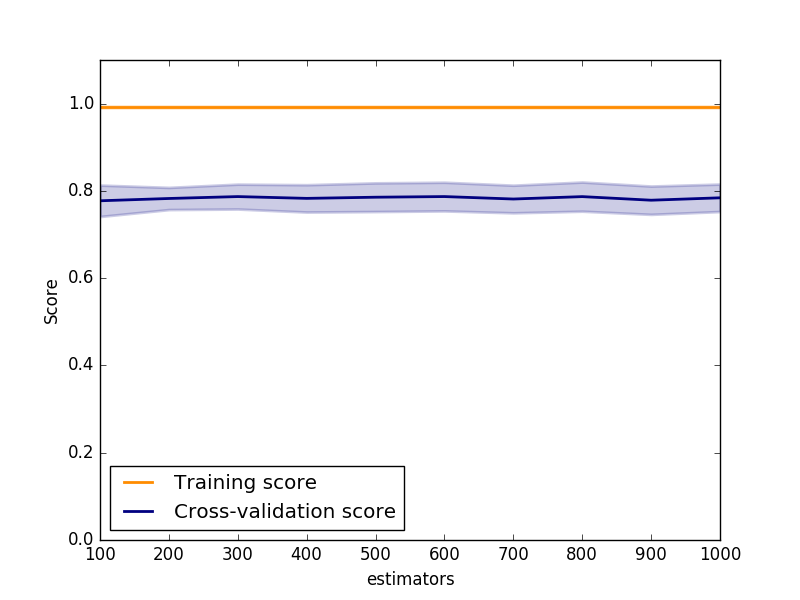
A validation score indicates whether the estimator is overfitting or underfitting for some hyperparameter values. To narrow the range of different parameters, we use a validation curve from a 5-fold-cross-validation of different models to determine the proper search range. When the cross-validation increases with the training score, the model is underfitting, and vice versa. [Fig. 4](#fig4) shows the validation curve for the SVM classifier with the parameters C and . [Fig. 5](#fig5) shows the validation curve for the RF classifier with the parameters related to learning trees. The training and the cross-validation scores do not change dramatically as the parameter values change; thus, the range must be determined through the tuning process. [Fig. 6](#fig6) shows the validation curve for the GTB classifier with the parameters related to learning trees and the learning rate. The range of parameters related to the learning trees should be tuned. The determined search range of the parameters of each model is shown in [Table 2](#table2).



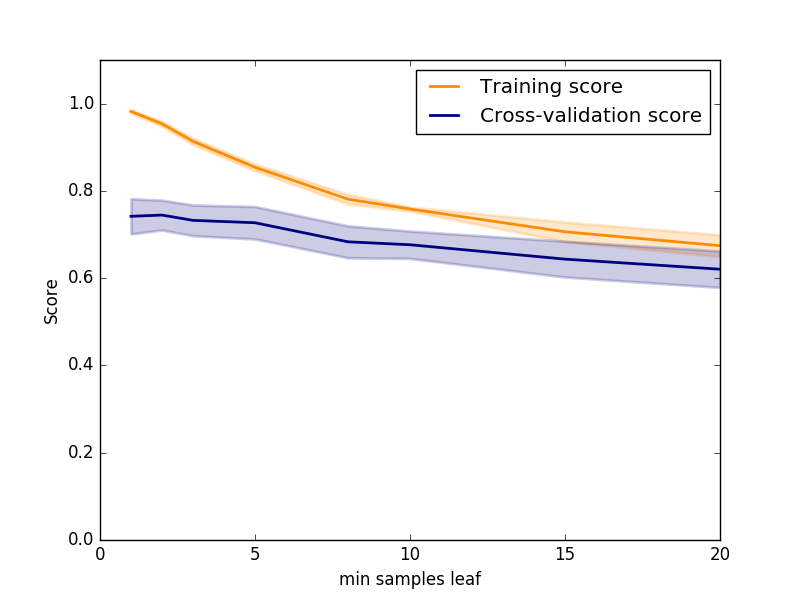
1. (b)

**Fig. 4.** Validation curve for the SVM classifier: (a) the penalty parameter of the error term (C);

(b) the Kernel coefficient for ‘RBF’ ()



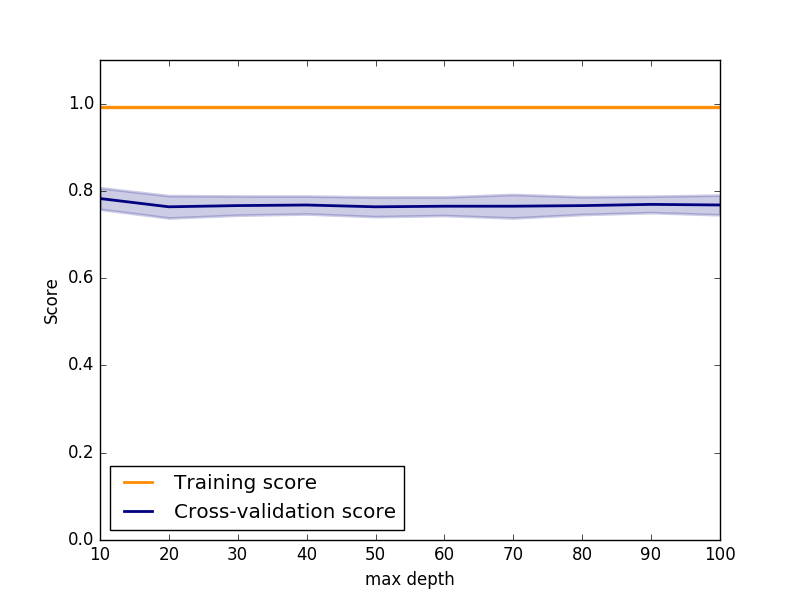
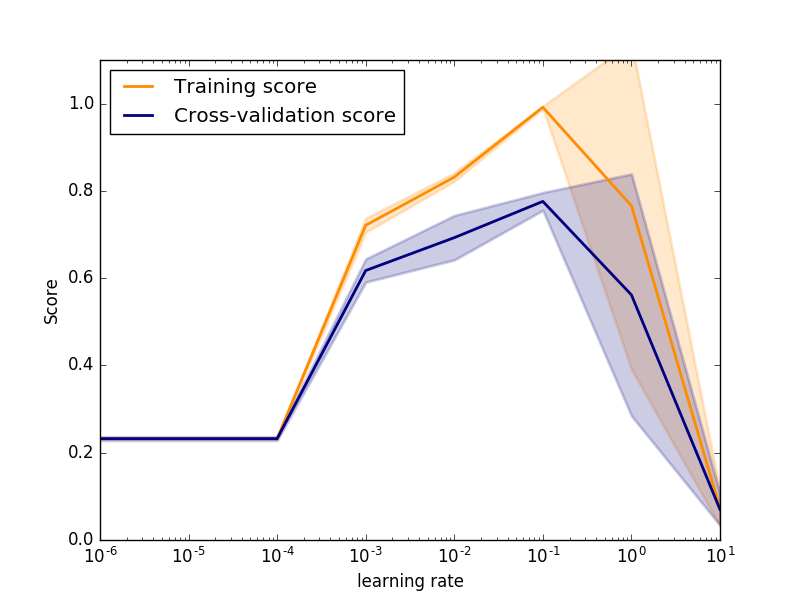
1. (b)



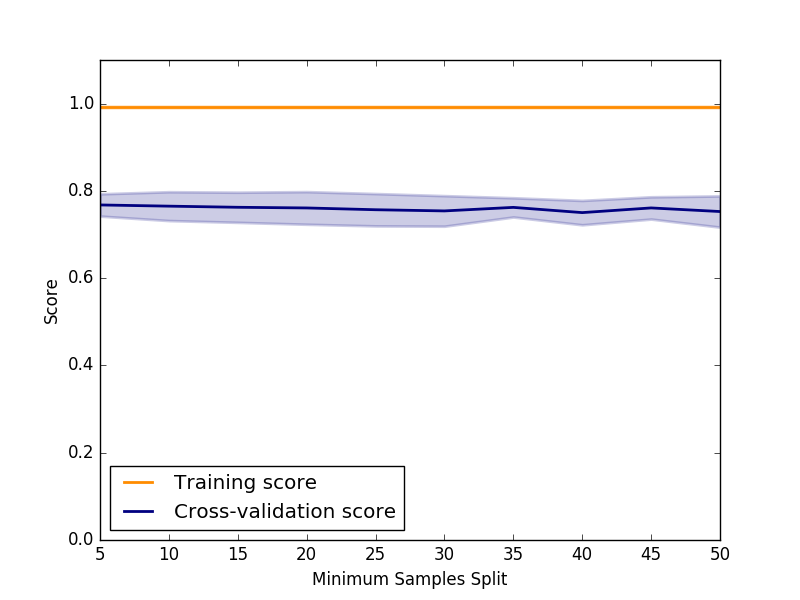
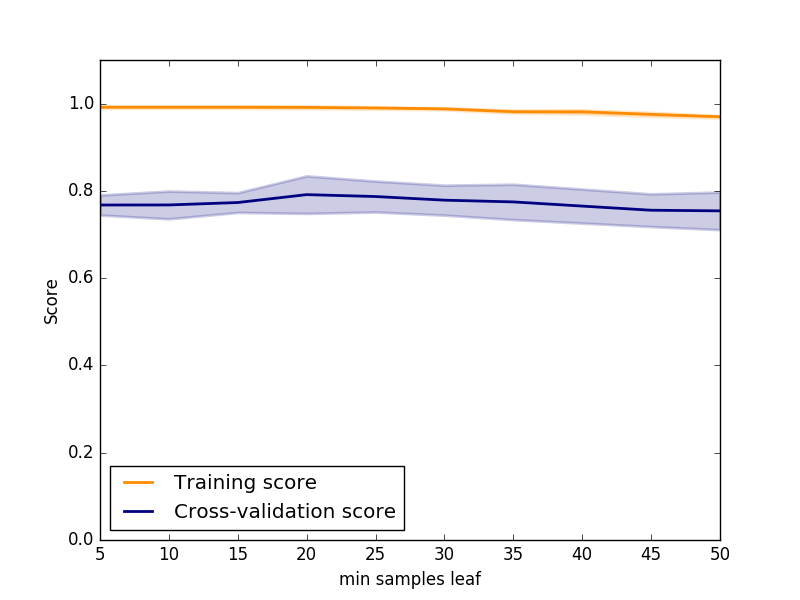
(c) (d)

**Fig. 5.** Validation curve for the RF classifier: (a) the number of trees in the forest (estimators);

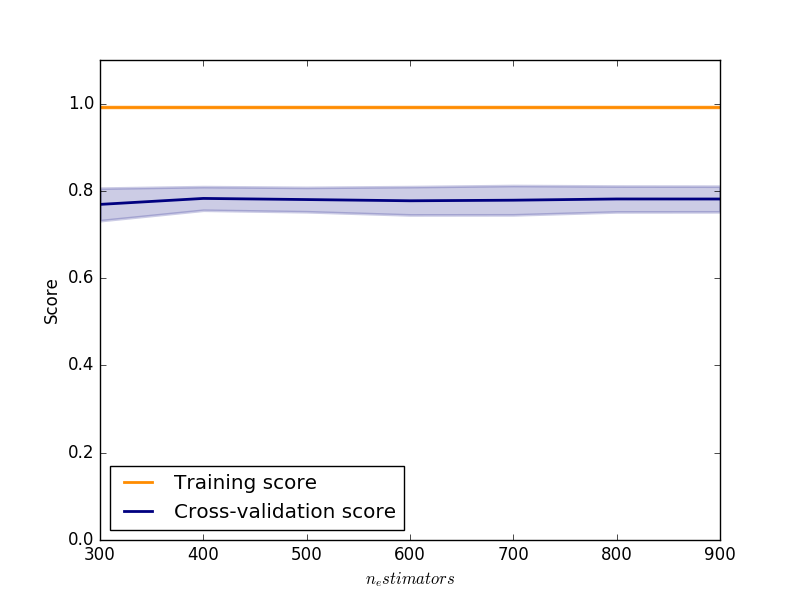
(b) the maximum depth of the tree (max depth); (c) the minimum number of samples required at a leaf node (min samples leaf); (d) the minimum number of samples required to split an internal node (min samples split)



1. (b)



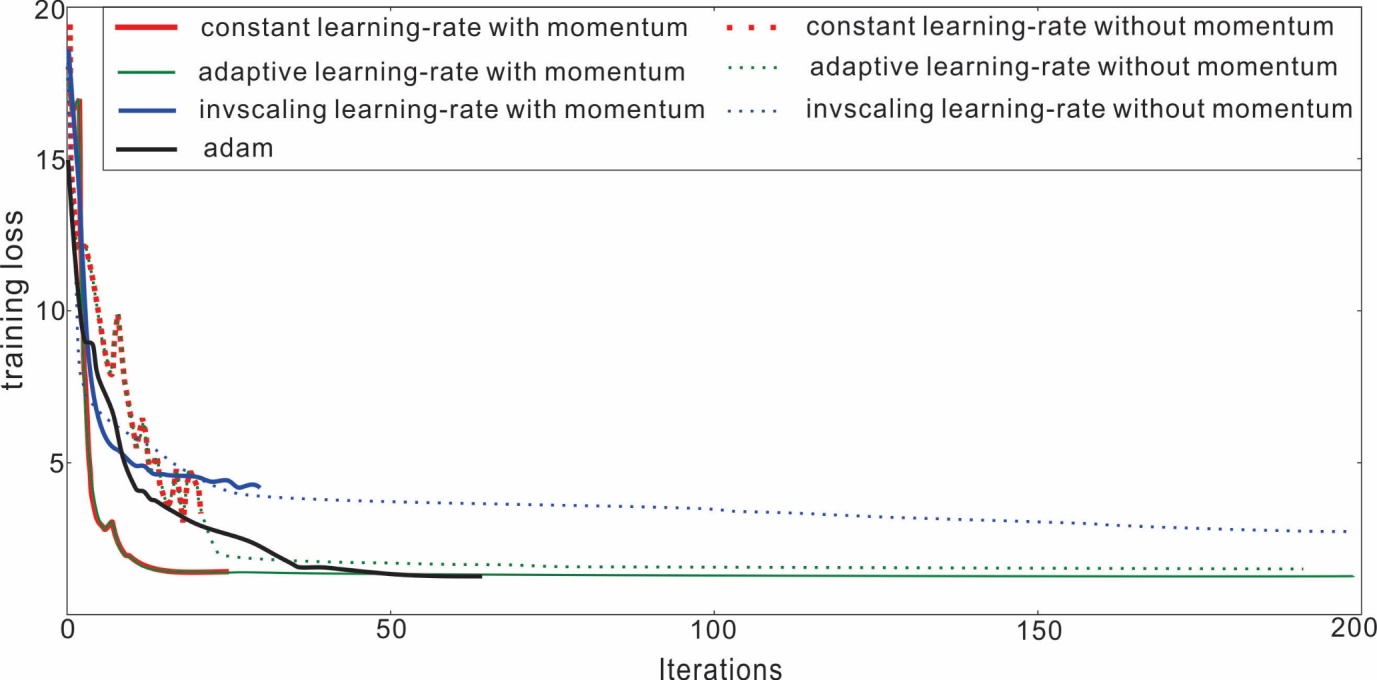
(c) (d)



(e)

**Fig. 6.** Validation curve for the GTB classifier: (a) the learning rate; (b) the maximum depth of the individual tree (max depth); (c) the minimum number of samples required at a leaf node (min samples leaf); (d) the minimum number of samples required to split an internal node (min samples split); and (e) the number of boosting stages (n estimators)

The performance of an ANN can be evaluated by both how fast the model converges and the training loss of the model. [Fig. 7](#fig7) shows the training loss curve for different NN learning strategies with an initial learning rate of 0.001 for the training data, including the SGD and Adam. The results show that strategies that include momentum converge faster than strategies that do not. A constant learning rate is maintained during the training process, while the inverse scaling learning rate at each time step uses an inverse scaling exponent. The adaptive learning strategy maintains the learning rate as long as the training loss continues to decrease. Among all the optimization strategies, Adam performs the best: it converges quickly and has the lowest training loss. Therefore, we use the Adam optimization strategy to train the ANN.



**Fig. 7.** Training loss curve for different ANN learning strategies with an initial learning rate of 0.001

The tunning process adapts the grid search approach which search through the specified search range to find the parameter set that has the best cross validation score among all the parameter combinations of each classifier. The optimum parameter settings for each classifier based on the 5-fold cross-validation is shown in Table 2.

* 1. Prediction evaluation

To consider the effect of random seeds in the model, we train each model ten times with the training set and test set, splitting them randomly using a different random seed for the two areas. During each run, we train the model with the training set and evaluate the trained model using the test set to obtain the predicted classes. We determined the precision, recall and F1-score of the predicted classes against the true classes for each run. After ten training processes, we averaged the scores to compare the performances of the different models. Two confusion matrixes of the predicted labels and true labels of the lithology classes from the two areas are constructed to show the prediction performance of each model.

1. **Results and Discussion**

This section compares the results of the five machine learning algorithms. Our results show that the prediction accuracy of the NB is below 50% because it assumes that the features are conditionally independent, given the class label. Thus, we do not include the results of the NB in this paper. Initially, the search process and the final results of the hyperparameter space for the best parameter set for each classifier is presented. Next, the precision, recall and F1-score of each lithology class for each classifier with the best hyperparameter trained model is compared. Finally, we compare four confusion matrixes based on the best model of each classifier to clearly present how each of the four classifiers determines the lithology classification.

5.1. Tuning Process

The optimum parameter set for each supervised model is shown in [Table 2](#table2). The GTB classification performance is highly dependent on the learning rate, which shrinks the contribution of each tree. When the learning rate is too large, a significant penalty is imposed on the trees, leading to overfitting, and vice versa. The RF classification performance is highly dependent on the tree structure. The results show that when the depth of the tree is too small, underfitting will occur, while the minimum number of samples required to split an internal node and the minimum number of samples required at a leaf node cause overfitting of the model. A smaller leaf causes the model to be more prone to capture noise in the training data. These results also show that the SVM models with a larger penalty parameter and a smaller kernel coefficient achieve better performance. Larger values of the kernel coefficient cause overfitting, while larger values of the penalty parameter do not cause overfitting.

5.2. Accuracy Metrics

Precision, recall and F1-score are used to evaluate the performance of the four models for the 5-fold-cross-validation. [Table 3](#table3) and [Table 4](#table4) show the precision, recall and F1-score of each lithology class for each model in the DGF and HGF, respectively. Overall, the ensemble methods achieve better classification results compared with the other two approaches. The SVM has the next-best performance, and the NN has the worst performance. Each model has a distinct behavior for the different lithology classes. All the models exhibit relatively high classification accuracy for the C and M classes compared with the other classes. There might be some large difference in the lithology properties of the C and M classes. The precision scores of the GTB and RF are above 80%. Among the classifiers, the GTB has the best performance in distinguishing the sandstone classes; its average precision scores on the sandstone classes in the DGF and HGF are 77.8% and 81.1%, respectively. The RF algorithm has a relative high classification accuracy; its average precision scores on the sandstone classes are 74.7% and 80.9%, respectively. These results indicate that the ensemble methods are more applicable for lithology classification with the available limited well log data, while the GTB performs better in classifying sandstone classes.

**Table 3**

Precision, recall and F1-scores for 5-fold-cross-validation over four classifiers on the DGF dataset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Gradient Tree Boosting | | | Random Forest | | |
| precision | recall | f1-score | precision | recall | f1-score |
| PS | 0.834 | 0.858 | 0.843 | 0.786 | 0.810 | 0.796 |
| CS | 0.645 | 0.569 | 0.599 | 0.696 | 0.612 | 0.649 |
| MS | 0.773 | 0.803 | 0.786 | 0.726 | 0.797 | 0.759 |
| FS | 0.859 | 0.794 | 0.823 | 0.779 | 0.784 | 0.779 |
| S | 0.823 | 0.863 | 0.837 | 0.911 | 0.636 | 0.743 |
| M | 0.834 | 0.912 | 0.870 | 0.804 | 0.883 | 0.838 |
| C | 0.969 | 0.949 | 0.956 | 0.996 | 0.929 | 0.960 |
| CR | 0.879 | 0.843 | 0.859 | 0.930 | 0.948 | 0.937 |
| avg | 0.827 | 0.824 | 0.822 | 0.829 | 0.800 | 0.808 |
|  | Support Vector Machine | | | Neural Network | | |
| precision | recall | f1-score | precision | recall | f1-score |
| PS | 0.777 | 0.808 | 0.791 | 0.567 | 0.456 | 0.492 |
| CS | 0.577 | 0.591 | 0.581 | 0.548 | 0.457 | 0.495 |
| MS | 0.768 | 0.715 | 0.736 | 0.651 | 0.753 | 0.697 |
| FS | 0.723 | 0.776 | 0.746 | 0.611 | 0.738 | 0.659 |
| S | 0.879 | 0.626 | 0.728 | 0.325 | 0.183 | 0.222 |
| M | 0.860 | 0.857 | 0.857 | 0.665 | 0.797 | 0.723 |
| C | 0.732 | 0.977 | 0.835 | 0.890 | 0.888 | 0.884 |
| CR | 0.785 | 0.512 | 0.604 | 0.662 | 0.480 | 0.548 |
| avg | 0.763 | 0.733 | 0.735 | 0.615 | 0.594 | 0.590 |

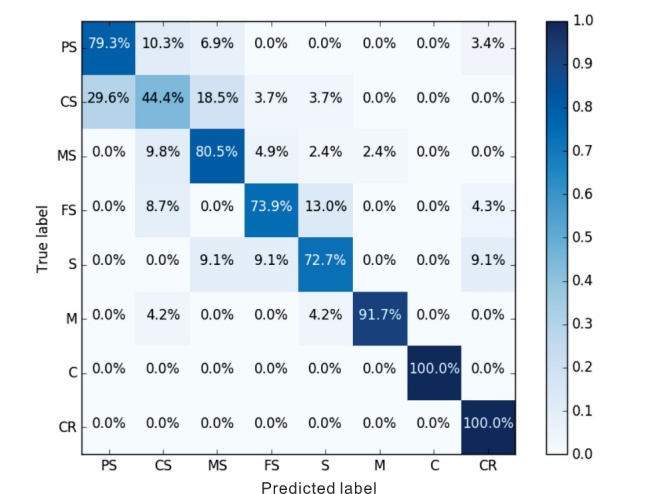
**Table 4**

Precision, recall and F1-scores for 5-fold-cross-validation over four classifiers on the HGF dataset

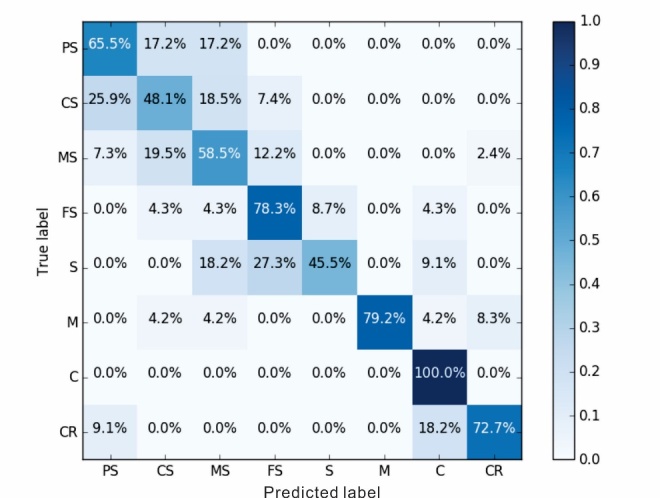
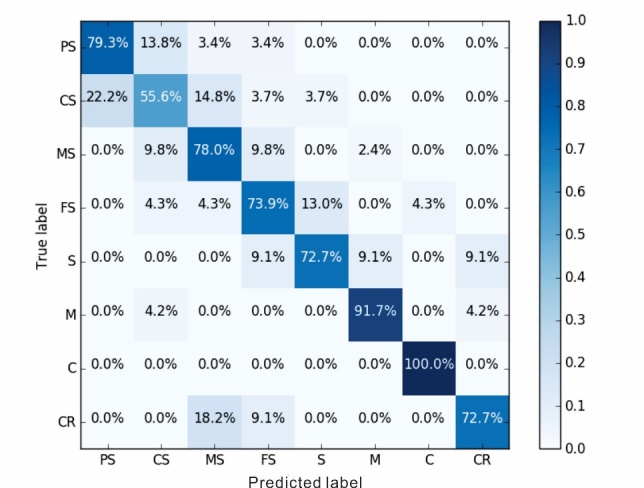
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Gradient Tree Boosting | | | Random Forest | | |
| precision | recall | f1-score | precision | recall | f1-score |
| PS | 0.841 | 0.852 | 0.846 | 0.810 | 0.883 | 0.845 |
| CS | 0.806 | 0.839 | 0.821 | 0.836 | 0.720 | 0.774 |
| MS | 0.822 | 0.790 | 0.803 | 0.794 | 0.798 | 0.793 |
| FS | 0.775 | 0.778 | 0.773 | 0.796 | 0.800 | 0.778 |
| S | 0.813 | 0.714 | 0.738 | 0.919 | 0.785 | 0.836 |
| M | 0.919 | 0.920 | 0.918 | 0.897 | 0.906 | 0.900 |
| C | 0.889 | 0.676 | 0.744 | 0.900 | 0.675 | 0.746 |
| avg | 0.838 | 0.796 | 0.806 | 0.850 | 0.795 | 0.810 |
|  | Support Vector Machine | | | Neural Network | | |
| precision | recall | f1-score | precision | recall | f1-score |
| PS | 0.794 | 0.800 | 0.797 | 0.553 | 0.821 | 0.660 |
| CS | 0.725 | 0.751 | 0.733 | 0.592 | 0.456 | 0.508 |
| MS | 0.727 | 0.704 | 0.714 | 0.460 | 0.407 | 0.428 |
| FS | 0.728 | 0.746 | 0.736 | 0.042 | 0.019 | 0.027 |
| S | 0.802 | 0.694 | 0.726 | 0.466 | 0.238 | 0.296 |
| M | 0.855 | 0.883 | 0.864 | 0.735 | 0.806 | 0.766 |
| C | 0.500 | 0.297 | 0.327 | 0.778 | 0.453 | 0.560 |
| avg | 0.733 | 0.696 | 0.700 | 0.518 | 0.457 | 0.464 |

5.3. Confusion Matrix

[Fig. 8](#fig8) and [Fig. 9](#fig9) present which lithology classes are misclassified to other classes for each model in the DGF and HGF. Overall the M, C and S have the highest prediction accuracy. However, the PS, CS, MS and FS classes might be misclassified to other sandstones. This result may occur in cases where the grain sizes of the sandstone are difficult to determine and the lithology interpretations present possible errors. The ensemble methods achieve the best classification performances: they classify most sandstones into the correct class. Both RF and GTB are powerful tree-based ensemble techniques that provide important measures for the feature variables. From the confusion matrix, we found that these two classifiers achieved similar classification accuracies among the lithology classes. As a result, the choice of model may come down to the use case. In RF, the trees are built independently, while the GTB builds new trees based on previous trees. RF uses fewer iterations , causing faster model training, but is prone to overfitting. While the GTB is robust to overfitting, it requires many boosting stages to perform optimally. Therefore, it is slower compared with the RF classifier. However, because distributed computing has recently become quite sophisticated, a distributed gradient boosting framework such as LightGBM by Microsoft ([Microsoft,2017](#Microsoft2017)), could be used to reduce the training time. In addition, the matrix shows that most of the misclassification cases occur between sandstone cases. The smaller sandstone is misclassified to siltstone, mudstone, coal and carbonate rock, which means there are large differences between sandstones and other lithology classes.

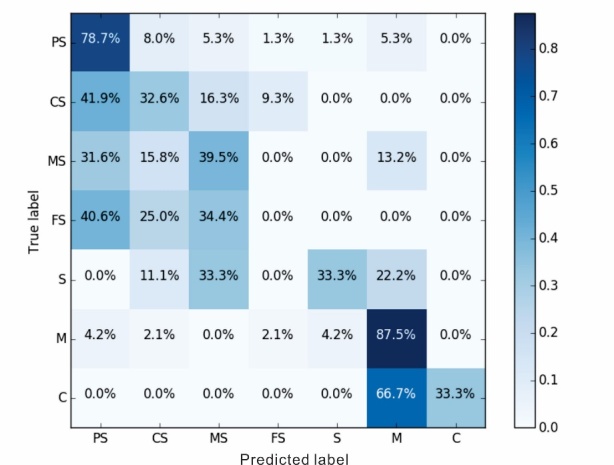
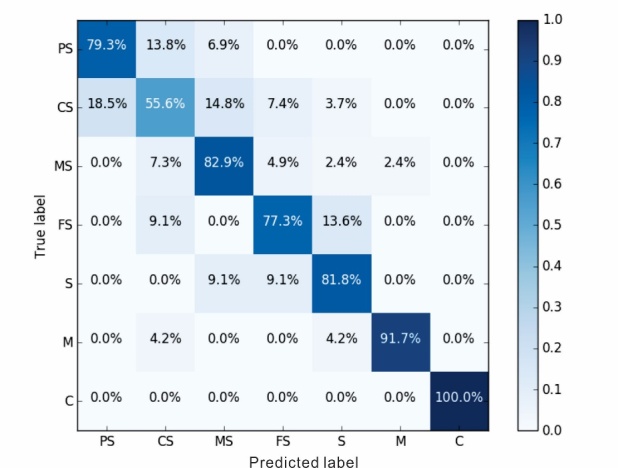


(a) (b)

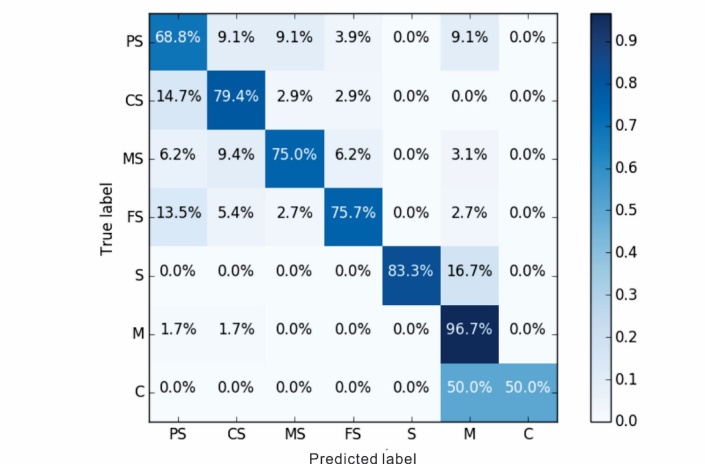
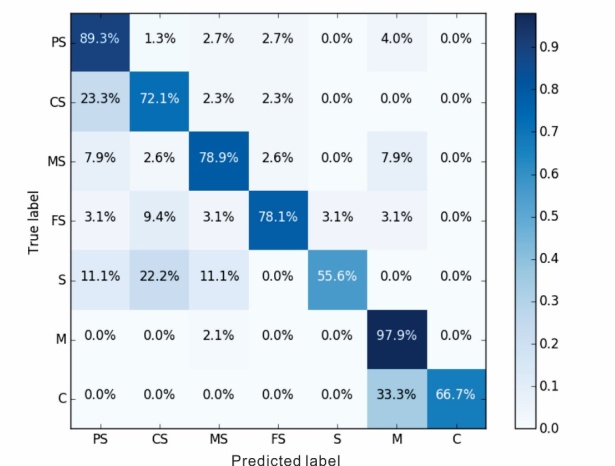


(c) (d)

**Fig. 8.** Confusion matrix plots on the DGF test dataset: (a) Gradient Tree Boosting Model; (b) Neural Network Boosting Model; (c) Random Forest Model; (d) Support Vector Machine Model



(a) (b)



(c) (d)

**Fig. 9.** Confusion matrix plots on the HGF test dataset (a) Gradient Tree Boosting Model; (b) Neural Network Boosting Model; (c) Random Forest Model; (d) Support Vector Machine Model

1. **Conclusion**

In this study, five different machine learning classifiers were applied to the problem of lithology classification using well log data obtained from twelve exploration wells from two gas fields in the Ordos Basin. We determined the required parameters for each model and used validation curves to determine the search range for each parameter. Then, we used a hyperparameter optimization method to optimize the parameter set. To examine the effects of the random seed on the models, all the instances were mixed and then divided randomly ten times into training and testing datasets. Each time, we trained the model using the training and testing datasets and obtained the average precision, recall and F1-score values to evaluate the performance of each classifier.

The results indicate that the ensemble methods perform well for lithology classification using data from well logs compared with other non-linear models when the number of features is small. SVM and ANN are usually used in high dimensional spaces. However, the limited number of features in this case study (seven) reduced their performances. However, the ANN classifier is capable of learning non-linear models. ANN with hidden layers have a non-convex loss function for which more than one local minimum exists, which might lead to different validation accuracies. Therefore, this case study indicates that the models the ANN and SVM create might be too complicated, resulting in overfitting.

Results also show that the challenge for lithology classification is sandstone classification. The ensemble methods could provide useful information to distinguish sandstone classes, because they performed the classification task with relatively high accuracy. The RF algorithm can be trained rapidly, but it is prone to overfitting. The GTB can provide a model with high classification accuracy, which could assist geologists in the presence of missing core or cutting samples. Furthermore, geologists can make use of the classification model to analyze large amount of well log data during geological exploration, which also improves the data analysis efficiency in petroleum geology. Although GTB is robust to overfitting, it is relatively slow to train the model, especially when the data size is huge. To apply GTB machine learning technique in real time data analysis of petroleum geology, more work can be done to increase the training speed of the GTB method, for example, applying new techniques such as LightGBM. Because the sandstone classes are difficult to classify, more features should be explored to improve their classification accuracy using the ensemble methods.

**Acknowledge**

We would like to thank the Huabei branch of Sinopec Group for providing the resources used in this study.

# Reference

An L H., 2013. Neutral network in lithology determination. Journal of Computer Science and Cybernetics. 16(2),59-62.

Akinyokun, O.C., Enikanselu, P.A., Adeyemo, A.B., Adesida, A.,2009. Well Log Interpretation Model for the Determination of Lithology and Fluid Contents. Pacific Journal of Science and Technology 10, 507-517.

Aha, D.W., Breslow, L.A.,1998. Comparing simplification procedures for decision trees on an economics classification task. Naval Research Laboratory, Washington, DC,19pp.

Al-Anazi, A., Gates, I.D.,2010. On the Capability of Support Vector Machines to Classify Lithology from Well Logs. Natural Resources Research 19, 125-139.

Breiman, L.,1996. Bagging predictors. Machine Learning 24, 123-140.

Breiman, L.,2001. Random Forests. Machine Learning 45, 5-32.

Chang Y W, Hsieh C J, Chang K W, et al., 2010. Training and Testing Low-degree Polynomial Data Mappings via Linear SVM. Journal of Machine Learning Research 11(11):1471-1490.

Cortes, C., Vapnik, V.,1995. Support-Vector Networks. Mach. Learn. 20, 273-297.

Cracknell, M., Reading, A.,2012. Machine Learning for Lithology Classification and Uncertainty Mapping. in: AGU Fall Meeting Abstracts, pp. 1511.

Cracknell, M.J., Reading, A.M.,2014. Geological mapping using remote sensing data: A comparison of five machine learning algorithms, their response to variations in the spatial distribution of training data and the use of explicit spatial information. Computers & Geosciences 63, 22-33.

Dai, J., Li, J., Luo, X., Zhang, W., Hu, G., Ma, C., Guo, J., Ge, S.,2005. Stable carbon isotope compositions and source rock geochemistry of the giant gas accumulations in the Ordos Basin, China. Organic Geochemistry 36, 1617-1635.

Delfiner, P. C., Peyret, O., Serra, O., 1987. Automatic determination of lithology from well logs. Spe Formation Evaluation 2(3), 303-310.

Dietterich T G.,2000. Ensemble Methods in Machine Learning. Multiple classifier systems. 1857(1):1-15.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.,2002. SMOTE Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16, 321-357.

Friedman, J.H.,2001. Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics 29, 1189-1232.

Friedman, J.H.,2002. Stochastic gradient boosting. Computational Statistics and Data Analysis 38, 367-378.

Fu, J., Wei, X., Nan, J.,2013. Characteristics and origin of reservoirs of gas fields in the Upper Paleozoic tight sandstone, Ordos Basin. Journal of Palaeogeography 15(4), 529-538[in Chinese].

Gifford, C.M., Agah, A.,2010. Collaborative multi-agent rock facies classification from wireline well log data. Engineering Applications of Artificial Intelligence 23, 1158-1172.

Harris, J.R., Grunsky, E.C.,2015. Predictive lithological mapping of Canada's North using Random Forest classification applied to geophysical and geochemical data. Computers & Geosciences. 80, 9-25.

Ho, T.K.,1998. The Random Subspace Method for Constructing Decision Forests. IEEE Trans. Pattern Anal. Mach. Intell 20, 832-844.

Horrocks, T., Holden, E. J., Wedge, D.,2015. Evaluation of Automated Lithology Classification Architectures using Highly-Sampled Wireline Logs for Coal Exploration. Computers & Geosciences. 83,209-218.

Hsu, C.-W., Lin, C.-J.,2002. A comparison of methods for multiclass support vector machines. Trans. Neur. Netw 13, 415-425.

Kecman, V., 2005. Support Vector Machines – An Introduction. In: Wang, L. (Ed.), Support Vector Machines: Theory and Applications 1. Springer, Berlin, Heidelberg, pp. 1–47 (Chapter 1).

Kingma D P, Ba J.,2014. Adam: A Method for Stochastic Optimization. ICLR, 1-15.

Kotsiantis, S.B.,2007. Supervised Machine Learning: A Review of Classification Techniques. in: Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, IOS Press, pp. 3-24.

Kohavi, R.,1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. in: Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2, Morgan Kaufmann Publishers Inc., Montreal, Quebec, Canada, pp. 1137-1143.

Li, J., Luo X., Shan, X., 2005. Natural gas accumulation in the Upper Paleozoic of Ordos Basin, China. Petroleum Exploration and Development 32(4), 54-59[in Chinese].

"Microsoft/Lightgbm". Microsoft. N.p., 2017. Web. 17 Mar. 2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., and Grisel, O., et al.,2011. Scikit-learn: machine learning in python. Journal of Machine Learning Research 12(10), 2825-2830.

Qian N.,1999. On the momentum term in gradient descent learning algorithms. Neural Netw, 12(1),145-151.

Rafik, B., Kamel, B.,2016. Prediction of permeability and porosity from well log data using the nonparametric regression with multivariate analysis and neural network, Hassi R’Mel Field, Algeria. Egyptian Journal of Petroleum, 1-16.

Reed R D, Marks R J.,1999. Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press, 49-67.

Rogers, S.J., Fang, J., Karr, C., Stanley, D.,1992. Determination of lithology from well logs using a neural network. AAPG bulletin 76, 731-739.

Safavian S R, Landgrebe D.,2002. A survey of decision tree classifier methodology. IEEE Transactions on Systems Man & Cybernetics. 21(3),660-674.

Salehi, S.M., Honarvar, B.,2014. Automatic identification of formation lithology from well log data: a machine learning approach. Journal of Petroleum Science Research 3(2), 73-82.

Sammut, C., 2011. Encyclopedia of machine learning. New York: Springer, pp.713-714.

Serra, O., & Abbott, H. T.,1982. The contribution of logging data to sedimentology and stratigraphy. Society of Petroleum Engineers Journal 22(1), 117-131.

Sebtosheikh, M., Motafakkerfard, R., Riahi, M., Moradi, S., Sabety, N.,2015. Support vector machine method, a new technique for lithology prediction in an Iranian heterogeneous carbonate reservoir using petrophysical well logs. Carbonates and Evaporites 30, 59-68.

Sebtosheikh, M. A., Salehi, A.,2015. Lithology prediction by support vector classifiers using inverted seismic attributes data and petrophysical logs as a new approach and investigation of training data set size effect on its performance in a heterogeneous carbonate reservoir. Journal of Petroleum Science & Engineering, 134, 143-149..

Shawe-Taylor, J., Cristianini, N.,2004. Kernel methods for pattern analysis. Journal of the American Statistical Association 101(476), 1730-1730.

Yang, H., Fu, J., Liu, X., Meng, P.,2012. Accumulation conditions and exploration and development of tight gas in the Upper Paleozoic of the Ordos Basin. Petroleum Exploration and Development 39, 315-324[in Chinese].

Zhu, X.,2008. Sedimentary Petrology (4th edition). Petroleum Industry Press, Beijing, 484pp[in Chinese].