Formation Lithology Classification: Insights into Machine Learning Methods

Conference Paper · September 2019

DOI: 10.2118/196096-MS

4 authors, including:

Mohamed Ibrahim Mohamed
Colorado School of Mines
21 PUBLICATIONS **49** CITATIONS

SEE PROFILE

Mohamed Salah Mohamed
Abu Dhabi National Oil Company
31 PUBLICATIONS **55** CITATIONS

SEE PROFILE

SPE-196096-MS

Formation Lithology Classification: Insights into Machine Learning Methods

Ibrahim Mohamed Mohamed, Colorado School of Mines; Salah Mohamed, Khalda Petroleum; Ibrahim Mazher and Pieprzica Chester, Apache Corp.

This paper was prepared for presentation at the SPE Annual Technical Conference and Exhibition held in Calgary, Alberta, Canada, 30 Sep - 2 October 2019.

Abstract

With the recent tremendous development in algorithms, computations power and availability of the enormous amount of data, the implementation of machine learning approach has spurred the interest in oil and gas industry and brings the data science and analytics into the forefront of our future energy. The idea of using automated algorithms to determine the rock facies is not new. However, the recent advancement in machine learning methods encourages to further research and revisit the supervised classification tasks, discuss the methodological limits and further improve machine learning approach and classification algorithms in rock facies classification from well-logging measurements. This paper demonstrates training different machine learning algorithms to classify and predict the geological facies using well logs data. Previous and recent research was done using supervised learning to predict the geological facies.

This paper compares the results from the supervised learning algorithms, unsupervised learning algorithms as well as a neural network machine learning algorithm. We further propose an integrated approach to dataset processing and feature selection. The well logs data used in this paper are for wells in the Anadarko Basin, Kansas. The dataset is divided into training, testing and evaluating wells used for testing the model. The objective is to evaluate the algorithms and limitations of each algorithm. We speculate that a simple supervised learning algorithm can yield score higher than neural network algorithm depending on the model parameter selected. Analysis for the parameter selection was done for all the models, and the optimum parameter was used for the corresponding classifier.

Our proposed neural network algorithm results score slightly higher than the supervised learning classifiers when evaluated with the cross-validation test data. It is concluded that it is important to calculate the accuracy within the adjacent layers as there are no definite boundaries between the layers. Our results indicate that calculating the accuracy of prediction with taking account the adjacent layers, yield higher accuracy than calculating accuracy within each point. The proposed feed-forward neural network classifier trains using backpropagation (gradient descent) provides accuracy within adjacent layers of 88%. Our integrated approach of data processing along with the neural network classifier provides more satisfactory results for the classification and prediction problem. Our finding indicates that utilizing simple supervised learning with an optimum model parameter yield comparable scores as a complex neural network classifier.

Introduction

Lithology classification using well logs using machine learning has been a subject of intense speculation for the past few years. The idea of using automated algorithms to determine the facies is not new. Indeed, many research has been done to take advantage of the supervised classification task and unsupervised classification to predict the geological facies using well logs data.

Facies are used by petroleum engineers and geologists to group rocks with similar characteristics. The classification based on well log data is the base of reservoir parameter calculations, and provide valuable information for geological studies in such fields as sedimentary facies and the environment. This characterization helps to identify formations that yield hydrocarbons and water. This classification is rather subjective as it depends on the attributes the interrupter chooses to be used for the classification. Biological properties, for instance, can be used to classify the rock. Rock granulometry and mineralogy are important characteristics that can be used for the classification. Well logs are the main source of information for defining the facies and their properties. By measuring the electrical responses as well as the nuclear radiations of the drilled section, we can infer properties about its rock matrix and fluid content. Apart from the valuable information obtained from the lithology interpretation, it has practical value in reserve calculations at the early exploration stage and detailed reservoir description at the development stage (Xie et al. 2018).

Lithologies can be determined through two ways. First, is through the observations from the cuttings obtained while drilling. Second is from lab studies through the observation and analysis of core samples taken from the underground formations. However, the latter method can provide different interpretations (Akinyokun et al. 2009) and, coring wells for analysis is an expensive and time-consuming operation (CITE SPWLA). While, well log data can provide general petrophysical parameters and is also used as a tool for sedimentologists and reservoir engineers (Serra and Abbott 1982), the data obtained from field acquisition could be highly sampled and in large amounts, which may burden the geologist who interprets the lithology within certain time constraints (Horrocks et al. 2015).

In practice, the integration of local geological knowledge, physical models, and personal experience to reduce large seismic and well log data into low-dimensional models of the Earth are used. Nonetheless, these simplified geological and physical assumptions are not always valid in practice, making the inferred model highly uncertain and biased. Many mathematical and artificial intelligence methods have been used to predict lithology based on a large volume of well log data. Prominent among these methods: Naïve Bayes (NB) classifiers (Li and Anderson-Sprecher 2006), Artificial Neural Networks (ANNs) (Qi and Carr 2006; Dubois et al. 2007; Al-Anazi and Gates 2010; Gifford and Agah 2010; Wang and Carr 2012a), Fuzzy logic (Hsieh et al. 2005), and Support Vector Machines (SVMs) (Al-Anazi and Gates 2010; Wang and Carr 2012b).

Machine learning is based on parallel processing used for approximation, clustering and pattern recognition purposes in large multivariant data sets (A. Abdulraheem et al. 2009; S. Elkatatny et al. 2016). Machine learning can explore the hidden connections among different physical quantities. Classifying field data into groups is one of the main branches of the popular field of machine learning. Classification via clustering is where the input data are first grouped into different clusters (log facies) which are subsequently assigned to particular lithofacies. This method has the benefit of unknown lithofacies becoming apparent rather than being silently misclassified. Among the vast panel of methods, this project is focusing on supervised learning and unsupervised learning methods. Supervised methods are learning from already labeled data. Those algorithms can discover abstract representations, understand the data and predict labels. Regression and classification are two main approaches for the supervised learning method, while clustering and dimensionality reduction are two main approaches for the unsupervised learning method.

SVM formulations are achieved by feature selection based on fuzzy logic from well logs. Al-Anazi and Gates (2010) speculate that SVMs yield results better than the probabilistic neural networks. Using

radial basis function kernel can also improve the classification results and decrease the misclassification rate error (Sebtosheikh et al. 2015). Random forest technique has been applied to geochemical and geological data for lithology classification. Prior research claims that random forest outperforms other machine learning algorithms and generates reliable first-pass predictions for practical geological mapping applications (Cracknell and Reading 2012; Cracknell and Reading 2014; Harris and Grunsky 2015). Back-Propagation, a neural network algorithm, has also been applied to lithology identification using well log data (Rogers et al. 1992). The neural network technique has also been compared with the alternating conditional expectations (ACE) method, and the ACE achieved a better performance (Rafik and Kamel 2017).

Multiple studies focus on improving the classification accuracy of different machine learning algorithms while comparing the different machine learning algorithms has been overlooked. In this study, we compare five machine learning algorithms, four of which are classification approaches; Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest – Decision Tree, one neural network approach namely; MLP Feed Forward Neural Network and one clustering unsupervised learning method; K-means. These algorithms were used to to identify lithology of well log data from Chase (Hugoton) and Council Grove gas (Panoma) fields in Southwest Kansas which are considered separate fields for regulatory purposes in Kansas.

Experimental results show that the MLP algorithms compete well with both the supervised methods, which were reported to perform well on wireline logs, while the unsupervised classifier, performed moderately but has low computational complexity. The algorithms for the five machine learning methods are explained in detail in a later section.

Dataset

The dataset used in this project is for eight wells in the Council Grove Dry Gas reservoir in Southwest Kansas (Figure 1). Hugoton and Panoma fields are located in the Hugoton Embayment, the shallow portion of the Anadarko Basin, and asymmetric foreland basin associated with the Ouachita Marathon Orogeny. By Wolfcamp time, the Pennsylvanian aged Anadarko was nearly filled, and slopes on the Kansas shelf were extremely low (Dubois et al. 2007). Due to the glacially driven eustatic sea level changes a fourth order marine continental (carbonate-siliciclastic) sedimentary cycles were formed on the Kansas shelf during Wolfcamp. The field is considered the largest Field in North America with an estimated ultimate recovery (EUR) of 75 TCF (Sorenson 2005). The field was discovered in 1922, but the development only started after 1940. The estimated gas reserve is 34 TCF (963 billion m^3) of original 50 TCF (1,416 billion m^3) produced in both Kansas and Oklahoma combined. Current annual production is about 300 BCF/year.

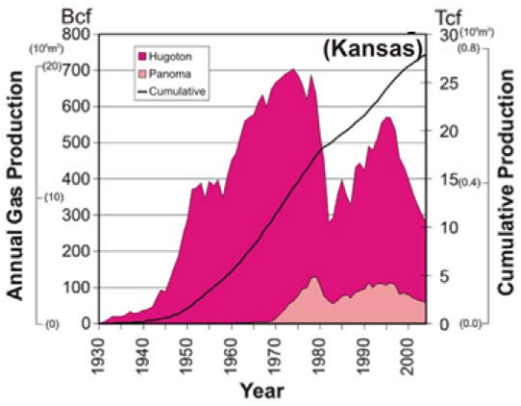

Figure 1—Production history of Hugoton and Panoma wells per 640 acres. (Modified from Dubois et al. (2006)).

There are total 12,000 drilled wells through an area of 6,200 mi^2 where each well produces 2.8 BCF. Figure 2 shows the production history of wells per 640 acres: Kansas Hugoton-2, Panoma-1, and Oklahoma

Hugoton-1. The eight wells used in this study have continuous core and facies are manually assigned. Locations of the eight wells are shown in Figure 3. Variables sampled at half-foot (0.15 m) intervals are then plotted against depth.

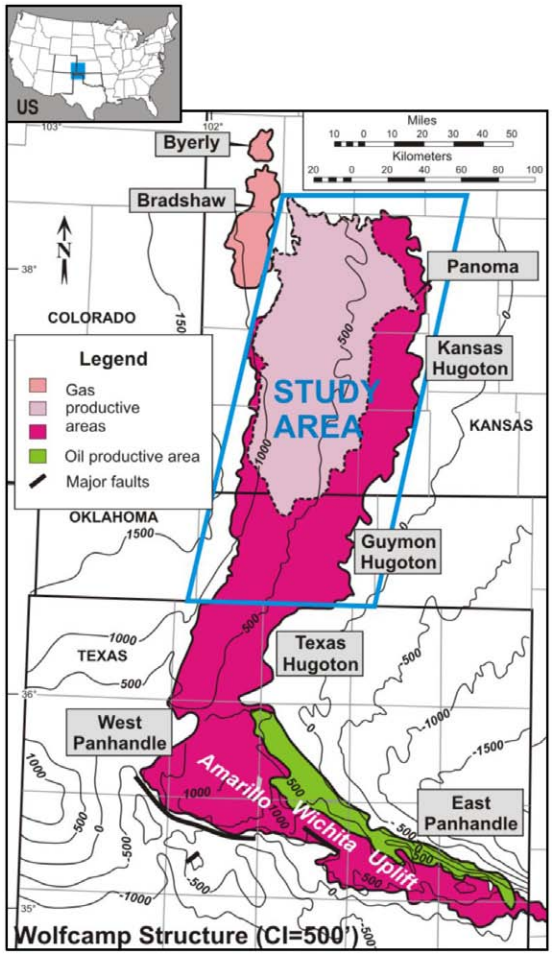

Figure 2—Study area Permian, Wolfcamp gas and oil fields. (Modified from Pippin (1970) and Sorenson (2005))

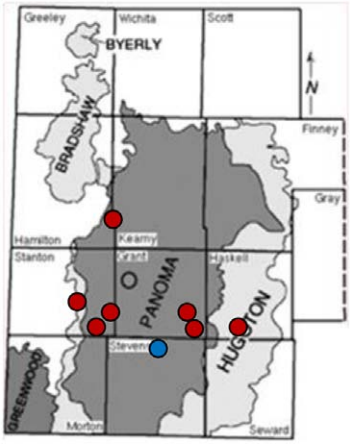

Figure 3—Eight well logs were used for the study. Red dots for "Training wells" and blue dots for "Blind wells" (Modified from Dubois et al. (2006)).

The core study indicates the existence of nine discrete facies (classes of rocks). Three are of continental origin, coarse sandstone (1), coarse siltstone (2) and fine siltstone (3), and six are of marine origin, marine

shaley siltstone (4), carbonate mudstone (5), wackestone (6), fine-crystalline dolomite (7), packstone (8), and bafflestone (9). Figure 4 shows the sediment composition diagram. The diagram shows the range of sedimentary rock types represented as mixtures of three components: calcium and magnesium carbonates, clay minerals, and silica (silicon dioxide). The argillaceous rocks are derived from the lithification of clay-rich muds, and they are given in blue circle which includes coarse siltstone (2) and fine siltstone (3) and, marine shaley siltstone (3). While limestones and dolomites rocks are presented with the red circle that includes fine-crystalline dolomite (7), a green circle includes carbonate mudstone (5), wackestone (6), packstone (8), and bafflestone (9) and lastly, quartz-rich rocks as coarse sandstone (1) are given by an orange circle.

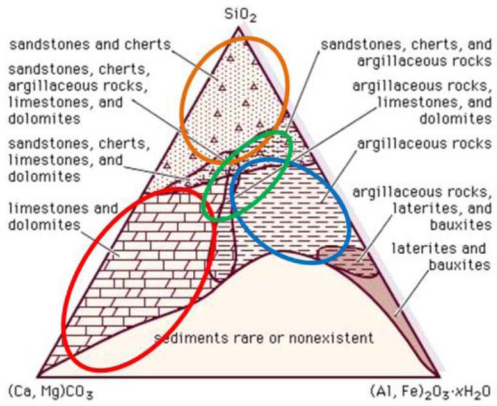

Figure 4—Sediment composition triangle. (Modified from (Schwab et al. 2018)).

Physical and chemical properties of the rocks are measured by the mean of the petrophysical logging tools that are lowered into the wellbore and record the measurements at half-foot increments. The tools lowered into wellbore can measure physical properties such as porosity, gamma radiation, resistivity, and photoelectric. These properties are often recorded and used to provide insights into the facies classification, rock properties, and fluid properties. Each combination of responses measured by the petrophysical logging tool is a characteristic of a specific type of rock, fluid, or rock property. The wireline logs available across the eight wells are gamma ray, deep induction log, average neutron-density porosity, neutron-density porosity difference, and photoelectric density log.

Methodology

Machine Learning is a field of computer science that gives computers the ability to learn without being explicitly programmed (Samuel, 1988); it's applied for prediction of unknown values through the generalization of known values. Machine learning is subdivided into three main categories; unsupervised learning, supervised learning, and reinforcement learning. Each category is used to solve different types of problems and scores performance differently. In the past decade, machine learning has been used in areas such as speech recognition, stock prediction, etc. Machine learning algorithms produce general hypotheses from externally supplied instances and make predictions on future instances (Kotsiantis et al. 2007). Supervised learning generates a prediction model from labeled input called training data. Each training data consists of a set of features and the desired output value. The supervised learning algorithm analyzes the training data and generates a model that can be used to predict new examples from the same type of feature vector.

Clustering is an unsupervised machine learning method that learns to group from the data itself and doesn't require training data. The groups consist of samples with similar features, which can be considered as distinct geological facies. Hyperparameters are tunable parameters that are used to optimize the performance

of the given algorithm to the data used to train it and generalize beyond it to make correct predictions on new data.

K–Nearest Neighbors (KNN) – Classification

*k*NN is a simple supervised learning approach. When training a *K*NN model, the model memorizes the locations of all points and their values. For predictions, the model takes the input data and calculates the *k* nearest points to that input. In a classification problem, the majority class of the nearest points is the prediction and in regression, the average of the nearest points is the prediction. This approach may seem simplistic but with a large amount of data, it can be quite effective. The model performance can be changed by adjusting the hyperparameter *k*.

Support Vector Machine (SVM) – Classification

Support Vector Machines (SVMs) are one of the popular methods used in machine learning. For a 2-dimensional, 2-class classification problem, SVMs find the line that maximizes the separation between the points of each class. The distance between the line and the nearest point classified one way, or the other is the margin. Many lines can separate the sets of points, but the goal is to find the line with the largest margin. The points closest to the separating line are support vectors.

The output from SVM is the label of a particular point as $w_T x+b$. As shown in Figure 5 three lines are obtained, one where the label is unknown (0) at the line equidistant from the two classes, and two lines where we know the labels are one class or the other (1, −1). The three lines are $w^T x+b=0$, $w^T x+b=1$, and $w^T x+b=-1$. If we select one support vector from each class, x_1, x_{-1}, the margin is the difference between the two points.

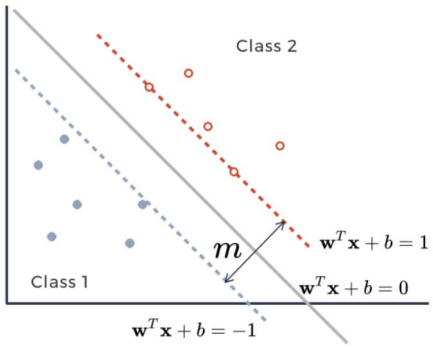

Figure 5—Geometric display of SVM's separating line where m is the margin.

We can state the two support vectors as,

$$w^T x_1 + b = 1$$
$$w^T x_{-1} + b = -1$$

subtracting the two equations gives,

$$w^T\left(x_1 - x_{-1}\right) = 2$$

dividing both sides by ||w|| results in,

$$margin = \frac{2}{||w||}$$

by minimizing the reciprocal of the margin, we obtain the objective function as

$$\min_w \frac{1}{2}||w||^2$$
$$s.t. \quad y_i\left(w^T x_i + b\right) \geq 1 \quad \forall\, i,$$

where $y_i \in [1, -1]$ determines the class point *i,* belongs to.

A hard margin does not allow any room for error, and this leads to overfitting the data. Soft margin allows some error by using a parameter, *C*. This will allow the model a certain amount of error based on the wrong points it classifies. We determine the error for a point using the distance to its class line and we use the *C* value as a weight to multiply by the sum of all the distances of the wrong points. A higher value for *C* will result in a "harder" margin and a lower value will result in a "softer" margin. Thus, the updated objective function is as follows

$$\min_{w} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$
$$s.t. \quad y_i\left(w^T x_i + b\right) \geq 1 - \xi_i, \xi_i \geq 0 \quad \forall\, i,$$

where ξ_i is the error for point *i* and *C* is the penalty tuning parameter.

Random Forest – Decision Tree

Random forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Similar to a forest is a group of trees, a random forest model is an aggregation of several decision trees. Each decision tree is trained on a random subset of the data to create the random forest model. To use the model, the random forest aggregates the prediction from each of the trees. The model selects the most common class, and for regression, it selects the average of the resulting predictions. Similar to the previous two classifiers, Random Forest takes an input of $n_{estimators}$ which represent the number of trees in the forest.

MLP Feed Forward Neural Network

A multilayer perceptron (MLP) is a class of feedforward artificial neural network consisting of at least three layers of nodes: an input layer, a hidden layer, and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function.

A feed-forward neural network (FFNN) connections flows in one direction and that direction only. This means that neurons in one layer will produce an output that is an input to neurons in the next layer. Data is fed forward and never returns to a previous layer. A fully connected network is one where the neurons in one layer are each connected to every neuron in the previous and the next layers, as shown in Figure 6.

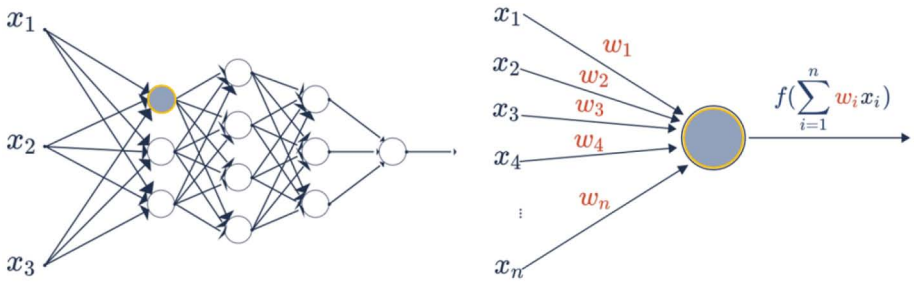

Figure 6—Example of a fully connected feedforward neural network (left) and neuron calculating its output (right).

K-means

K-means is an unsupervised method that has a hyperparameter *k* which is the number of clusters to group the data into. *K*-means determines cluster centroids, μ, that solve the objective,

$$\min_{\mu} \sum_{i=1}^{k} \sum_{x_j \in C_i} ||x_j - \mu_i \quad ||2_2$$

where μ_i is the centroid of cluster i and C_i is the subset of points that belong to cluster i. K-means minimizes the Euclidean distance between each point and the centroid of its labeled cluster using the following algorithm.

Input: Features from data
Output: Centroids of the k clusters
1 Initialize cluster centroids (randomly or using a particular strategy)
2 **while** cluster centroids update not converge **do**
3 Determine cluster for each point based on the distance to centroids
4 Update centroids' location as the mean of points in the cluster
5 **end**

Data Processing

The statistical distribution of the input variables is shown in Figure 7f. This is used to count the number of times each facies appear in the dataset. It can be observed that dolomite yields the lowest occurrence, and coarse siltstone has the highest occurrence. As explained later in the discussion section, the distribution of facies affects the accuracy calculations.

Figure 7b indicates that mudstone and wackstome (Facies 5 and 6) are characterized by a relatively high (and variable) resistivity. Figure 7e indicates that bafflestone (Facies 9) is characterized by a relatively high (and variable) photoelectric effect. Figure 7c indicates that shaley sandstone and dolomite (Facies 3 and Facies 7) are characterized by a relatively high gamma ray.

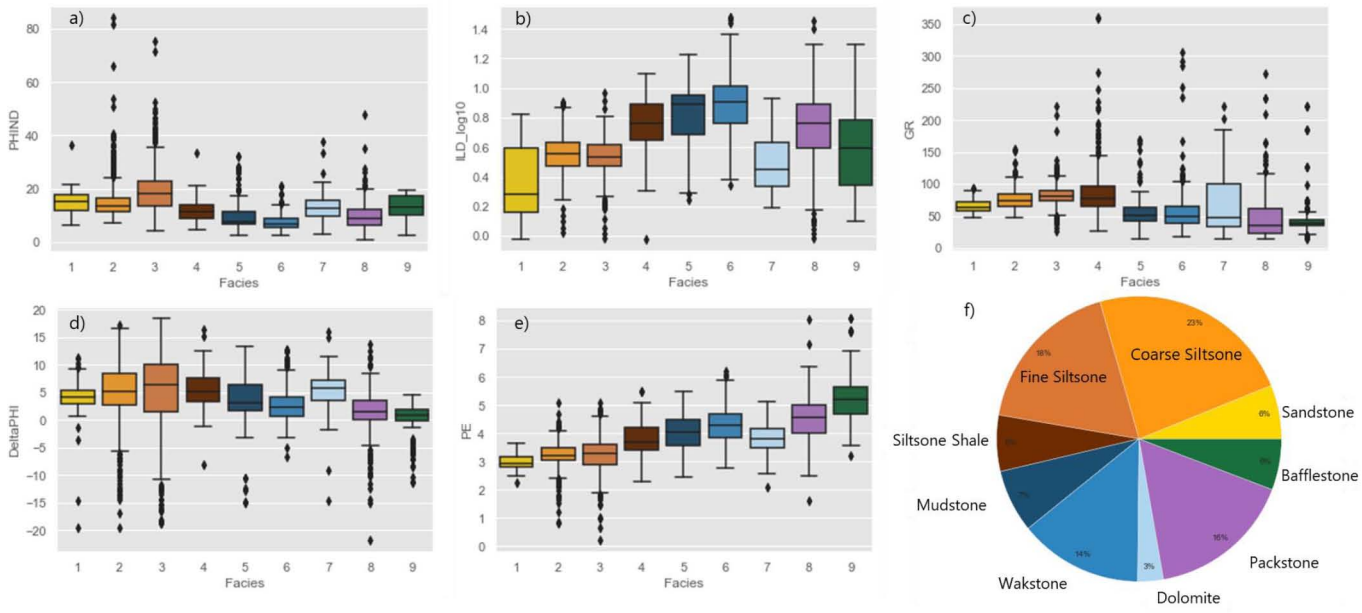

Figure 7—Distribution of PHIND (a), ILD_Log10 (b), GR (c), DeltaPHI (d) and PE (e) across the facies. f) distribution of data by facies.

Scatter matrix cross plot was drawn to visualize the change of the features with the facies, as shown in Figure 8. It is a useful tool to visualize how two properties vary with rock type and give insights if there is any direct correlation between them. Each pane in the plot shows the relation between two features and the color change corresponds to each facies. Figure 8 shows that the petrophysical properties measured by wireline logging tools are fuzzy and it is not clear what direct relationships exist between the measurements and facies types. This is where machine learning will prove useful (Dubois et al. 2007). For each portion of the facies continuum, there is a wide range of responses for each measured petrophysical property and thus feature

spaces of each property have significant overlap between each facies. The petrophysical properties recorded at half-foot intervals are actually the weighted average of a much larger interval. Borehole environments as borehole size, mud properties, pressure, temperature, and other environmental conditions affect the logging tool responses.

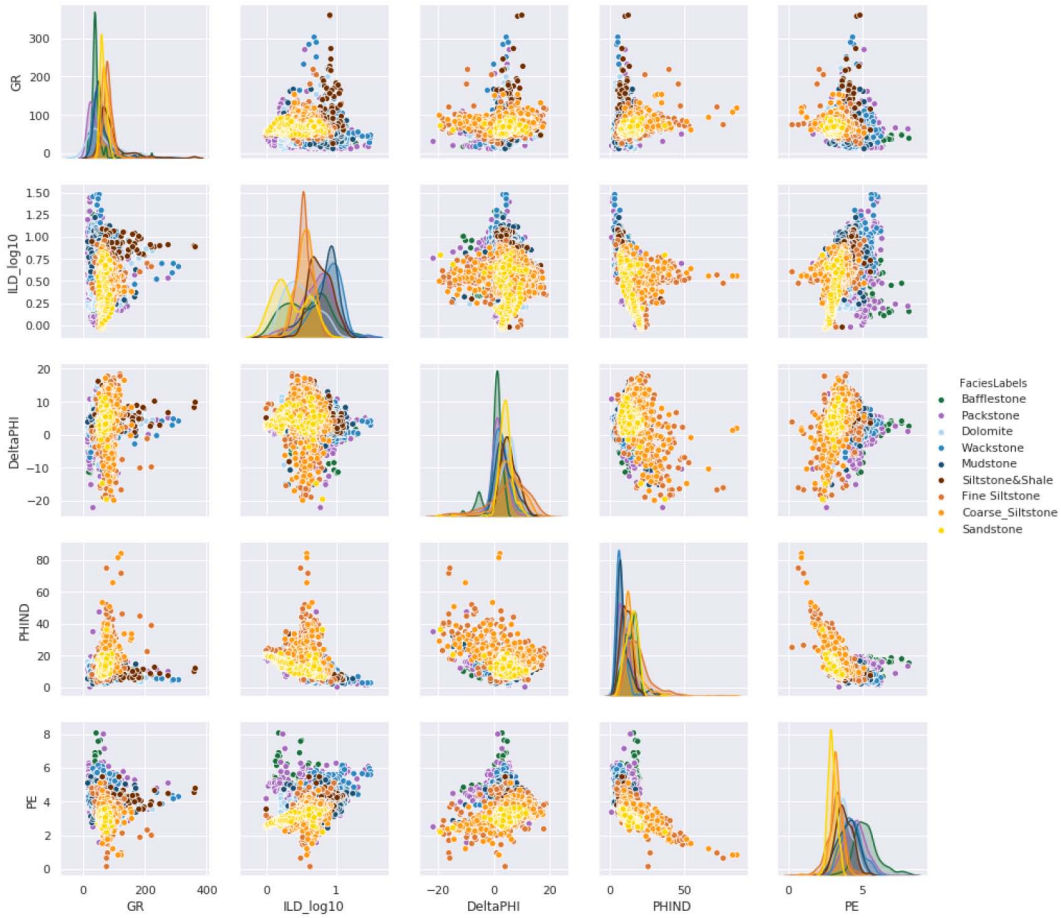

Figure 8—Cross-plot of data features, PE (photoelectric effect), PHIND (porosity), DeltaPHI (neutron-density porosity difference), ILD_log10 (resistivity), and GR (gamma ray). Facies represented by colored dots form overlapping clouds and are difficult to separate with parametrically defined boundaries.

There is a total of eight wells, where we used seven wells as training wells, and one well is used as a test or blind well for evaluating the model. To standardize the data, the "sklearn.preprocessing" were used to transform the raw features vectors into a representation that is more suitable. This grants that the features are standard normally distributed; Gaussian with zero mean and unit variance. We also used "train test split" to split training data into training and test data. The test data are used as cross-validation data to tune the parameter of the model. The test data is used to evaluate the algorithm and is not used for training the network. For this project, 20% of the available data points were used for testing, and the rest were used for training. Evaluating the accuracy of a classifier can help in predicting its future accuracy (Kohavi 1995). Bias and variance are often observed in supervised machine learning. Here comes the role of cross-validation where the bias is traded off for low variance and thus, removes the impact of the pattern of the current dataset (Xie et al. 2018). Cross-validation was used in this study to reduce the possibility of data overfitting and provide insight into how the model generalizes an independent dataset.

Accuracy and Model Evaluation

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.

The recall is the ratio of true positives t_p to the summation of true positives t_p and the number of false negatives f_n. It is used to define the number of relevant items retrieved by the supervised model. While precision is the ratio of true positives t_p to the summation of true positives t_p and the number of false negatives f_p. It is used to define the number of relevant items selected. F1-score provides the harmonic mean of precision and the recall. It is used to measures the accuracy of both precision and recall. A good algorithm should maximize both precision and recall simultaneously.

Results and Discussion

Training using SVM Classifier

SVM is capable of performing multi-class classification on a dataset. This requires two parameters namely; Gamma (γ) and C. The parameter C is a regularization factor and tells the classifier how much we want to avoid misclassifying training examples. A large value of C will try to correctly classify more examples from the training set, but if C is too large, it may overfit the data and fail to generalize when classifying new data. If C is too small, then the model will not be good at fitting outliers and will have a large error on the training set. The Gamma (γ) parameter describes the size of the radial basis functions, which is how far away two vectors in the feature space need to be considered close. SVM classifier takes as input two arrays: an array "X" of size [n samples, n features] holding the training samples, and an array "y" of class labels. After fitting the model, it can then be used to predict the new values.

The model was then evaluated using the testing data. We used the confusion matrix as shown in Table 1 to obtain the F1-score, precision, and recall, as shown in Table 2. Using the default parameters for SVM, the F1-score was 49% and the accuracy of the facies classification which is the number of correct classifications divided by the number of classifications was calculated and was found to be 49%. Noting that there is no definite boundaries between the layers, it is possible that the adjacent layers may affect the predictions, thus it's important to calculate the accuracy within the adjacent layers. In this algorithm, the accuracy within adjacent layers was found to be 84%.

Table 1—Confusion matrix using SVM classifier and default values of Gamma (γ) and C.

	S.S.	C. Siltstone	F. Siltstone	Sh. Siltstone	Mudstone	Wackstone	Dolomite	Packstone	Bafflestone
S.S.	18	20	1	0	0	0	0	0	0
CoarseSiltstone	0	91	35	0	0	1	0	3	0
Fine Siltstone	0	22	5	0	0	0	0	3	0
Shaley Siltstone	0	15	6	0	0	4	0	8	0
Mudstone	0	4	4	1	0	39	0	12	0
Wackstone	0	14	2	0	0	32	0	24	0
Dolomite	0	4	1	0	0	0	0	12	0
Packstone	0	9	2	0	0	16	0	60	1
Bafflestone	0	0	0	0	0	1	0	22	11

Table 2—Precision, recall, and F1-scores for 5-fold-cross-validation using SVM classifier and default values of Gamma (γ) and C.

	Precision	Recall	F1-Score
S.S.	1	0.46	0.63
Coarse Siltstone	0.51	0.7	0.59
Fine Siltstone	0.54	0.7	0.61
Shaley Siltstone	0	0	0
Mudstone	0	0	0
Wackstone	0.34	0.44	0.39
Dolomite	0	0	0
Packstone	0.42	0.68	0.52
Bafflestone	0.92	0.32	0.48
Avg.	0.49	0.49	0.49

Next, we focused on the model parameter optimization to obtain the highest accuracy and F1-scores. In this part, we train the model with different values for the Gamma (*γ)* and the *C*. A range from 0.0001 to 100 for Gamma (*γ)* and 0.01 to 10000 for *C* were evaluated. The results from the loop are shown in Figure 9. We picked the parameters with the highest cross validation accuracy. In this project, the optimal *C* and Gamma (*γ)* were found to be 100 and 0.01 respectively.

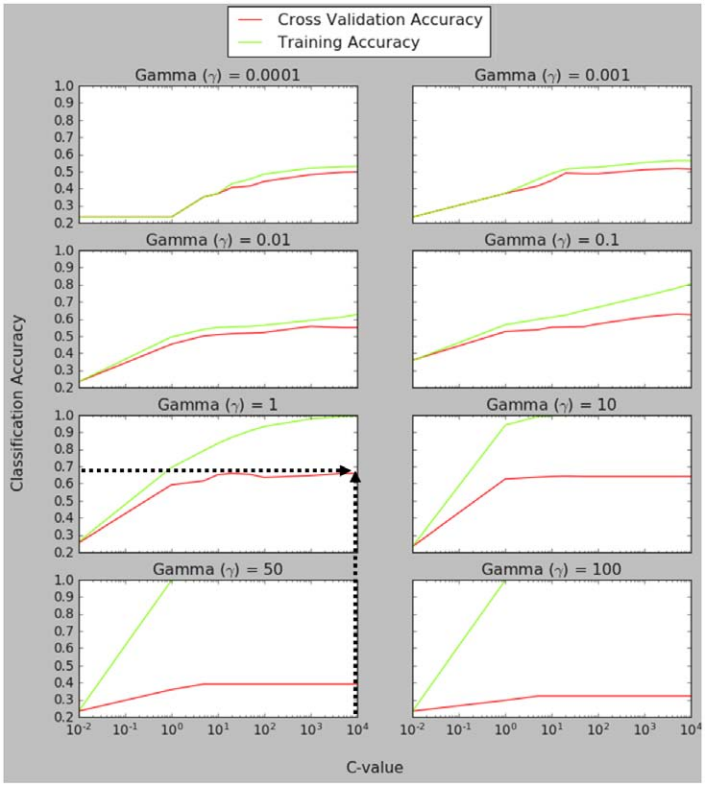

Figure 9—Searching for optimum cross-validation accuracy with different values of Gamma (*γ*) and *C*.

The model was evaluated with the new parameters, and F1-score of 66% was obtained. The new Facies accuracy is 65%, and the adjacent facies accuracy is 84%. The model was then evaluated with the blind test data. The F1-score obtained was 49%, the new Facies accuracy is 49%, and the adjacent facies accuracy is 89%.

Table 3—Confusion matrix using SVM classifier with optimum Gamma (γ) and C values.

	S.S.	C. Siltstone	F. Siltstone	Sh. Siltstone	Mudstone	Wackstone	Dolomite	Packstone	Bafflestone
S.S.	27	7	4	0	0	0	0	1	0
CoarseSiltstone	5	104	12	5	0	2	0	2	0
Fine Siltstone	1	20	57	1	2	2	0	0	1
Shaley Siltstone	0	4	1	21	1	3	0	3	0
Mudstone	0	3	3	7	27	14	0	6	0
Wackstone	0	5	5	3	8	43	0	6	2
Dolomite	0	2	1	0	2	0	8	3	1
Packstone	1	1	6	4	7	12	1	50	6
Bafflestone	0	0	1	0	0	1	0	2	30

Table 4—Precision, recall, and F1-scores using SVM classifier with optimum Gamma (γ) and C values.

	Precision	Recall	F1-Score
S.S.	0.79	0.69	0.74
Coarse Siltstone	0.71	0.8	0.75
Fine Siltstone	0.63	0.68	0.66
Shaley Siltstone	0.51	0.64	0.57
Mudstone	0.57	0.45	0.5
Wackstone	0.56	0.6	0.58
Dolomite	0.89	0.47	0.62
Packstone	0.68	0.57	0.62
Bafflestone	0.75	0.88	0.81
Avg.	0.66	0.66	0.66

Training using the KNN Classifier

For this classifier, we tested the model for different k-values to obtain the optimal F1-score. A nested loop was made to test multiple k-values, and the F1-scores for training and the validation data were plotted, as shown in Figure 10. An optimum value for k was observed to be equal to 68.

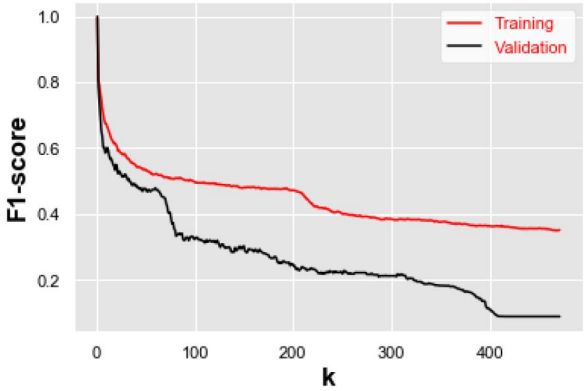

Figure 10—F1-scores for Different k values

The model was then evaluated using the testing data. We used the confusion matrix, as shown in Table 5 to obtain the F1-score, precision, and recall, as shown in Table 6. Using k equals to 68 the F1-score was

found to be 65%. Also, the accuracy of the facies classification which is the number of correct classifications divided by the number of classifications was calculated and was found to be 70%, and accuracy within the adjacent layers was found to be 86%. The model was then evaluated with the blind test data. The F1-score obtained was 46%, the new facies accuracy is 46%, and the adjacent facies accuracy is 89%. Confusion matrix table and F1-scores table for the blind well are shown in Table 7 and Table 8, respectively.

Table 5—Confusion matrix using KNN classifier with optimum k=68.

	S.S.	C. Siltstone	F. Siltstone	Sh. Siltstone	Mudstone	Wackstone	Dolomite	Packstone	Bafflestone
S.S.	28	7	3	0	0	0	0	1	0
CoarseSiltstone	2	97	23	5	0	1	1	1	0
Fine Siltstone	1	27	49	1	2	3	0	0	1
Shaley Siltstone	0	1	0	23	1	5	0	3	0
Mudstone	0	2	3	8	26	12	0	9	0
Wackstone	1	5	4	3	5	45	0	8	1
Dolomite	0	0	0	1	1	0	9	5	1
Packstone	1	3	5	2	4	12	1	55	5
Bafflestone	0	0	0	0	0	0	0	3	31

Table 6—Precision, recall, and F1-scores using KNN classifier with optimum k=68.

	Precision	Recall	F1-Score
S.S.	0.85	0.72	0.78
Coarse Siltstone	0.68	0.75	0.71
Fine Siltstone	0.56	0.58	0.57
Shaley Siltstone	0.53	0.7	0.61
Mudstone	0.67	0.43	0.53
Wackstone	0.58	0.62	0.6
Dolomite	0.82	0.53	0.64
Packstone	0.65	0.62	0.64
Bafflestone	0.79	0.91	0.85
Avg.	0.66	0.66	0.65

Table 7—Prediction results for Blind well: confusion matrix using KNN classifier with optimum k=68.

	S.S.	C. Siltstone	F. Siltstone	Sh. Siltstone	Mudstone	Wackstone	Dolomite	Packstone	Bafflestone
S.S.	3	79	2	0	0	5	0	0	0
CoarseSiltstone	6	63	15	0	0	5	0	0	0
Fine Siltstone	0	41	69	0	0	6	0	1	0
Shaley Siltstone	0	0	0	1	0	5	0	1	0
Mudstone	0	1	0	7	0	7	0	4	0
Wackstone	0	1	0	2	3	50	0	15	0
Dolomite	8	1	0	0	0	0	0	8	0
Packstone	0	0	1	2	0	14	0	21	2
Bafflestone	0	0	0	0	0	0	0	0	0

Table 8—Prediction results of Blind well: Precision, recall, and F1-scores using KNN classifier with optimum k=68.

	Precision	Recall	F1-Score
S.S.	0.18	0.03	0.06
Coarse Siltstone	0.34	0.71	0.46
Fine Siltstone	0.79	0.59	0.68
Shaley Siltstone	0.08	0.14	0.11
Mudstone	0	0	0
Wackstone	0.54	0.7	0.61
Dolomite	0	0	0
Packstone	0.42	0.53	0.47
Bafflestone	0	0	0
Avg.	0.46	0.46	0.46

Training using Random Forrest Classifier

Random forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Similar to the previous two classifiers, Random Forest takes an input of $n_{estimators}$ which represents the number of trees in the forest. The optimal $n_{estimators}$ was found to be 52, where the F1-score is 63%. The new facies accuracy is 63%, and the adjacent facies accuracy is 88%. The model was then evaluated with the blind test data. The F1-score obtained was 45%, the new facies accuracy is 45%, and the adjacent facies accuracy is 88%.

Table 9—Prediction results for Blind well: confusion matrix using Random Forrest classifier with optimum $n_{estimator}$=52.

	S.S.	C. Siltstone	F. Siltstone	Sh. Siltstone	Mudstone	Wackstone	Dolomite	Packstone	Bafflestone
S.S.	11	68	1	1	1	5	0	2	0
CoarseSiltstone	7	65	6	2	1	5	0	3	0
Fine Siltstone	1	39	56	5	4	8	0	4	0
Shaley Siltstone	0	0	0	1	3	2	0	1	0
Mudstone	0	0	0	1	0	14	0	4	0
Wackstone	0	0	0	2	6	32	0	30	1
Dolomite	1	0	1	0	0	0	7	8	0
Packstone	0	0	0	0	0	9	0	30	1
Bafflestone	0	0	0	0	0	0	0	0	0

Table 10—Prediction results for Blind well: Precision, recall and F1-scores using Random Forrest classifier $n_{estimator}$=52.

	Precision	Recall	F1-Score
S.S.	0.55	0.12	0.2
Coarse Siltstone	0.38	0.73	0.5
Fine Siltstone	0.88	0.48	0.62
Shaley Siltstone	0.08	0.14	0.11
Mudstone	0	0	0
Wackstone	0.43	0.45	44
Dolomite	1	0.41	58
Packstone	0.37	0.75	49
Bafflestone	0	0	0
Avg.	0.45	0.45	0.45

Neural network Model: Training using MLP Feed Forward Neural Network

MLP Classifier implements a multi-layer perceptron (MLP) algorithm that trains using backpropagation. MLP trains on two arrays: x-array with the size ($n_{samples}$, $n_{features}$). x-array holds the training samples represented as floating point feature vectors. y-array with the size ($n_{samples}$), which holds the target values for the training samples. For this project, we used the rectified linear unit activation function "*relu*" and the solver for weight optimization "*lbfgs optimizer*" which is in the family of quasi-Newton methods. The optimum hidden layer size was found to be 2 hidden layers, each containing 150 and 25 neurons, respectively. The F1-score was found to be 64%, new facies accuracy is 64%, and the adjacent facies accuracy is 88%. The model was then evaluated with the blind test data. The F1-score obtained was 42%, the new facies accuracy is 40%, and the adjacent facies accuracy is 86%.

Table 11—Prediction results for Blind well: confusion matrix using MLP classifier with optimum 2 hidden Layers

	S.S.	C. Siltstone	F. Siltstone	Sh. Siltstone	Mudstone	Wackstone	Dolomite	Packstone	Bafflestone
S.S.	28	11	0	0	0	0	0	0	0
CoarseSiltstone	4	95	20	5	1	2	1	2	0
Fine Siltstone	0	27	51	1	2	2	0	0	1
Shaley Siltstone	0	2	2	23	3	1	0	2	0
Mudstone	0	1	2	5	26	18	0	8	0
Wackstone	0	8	2	4	6	39	1	10	2
Dolomite	0	0	1	0	2	0	9	4	1
Packstone	0	1	5	0	4	15	2	54	7
Bafflestone	0	0	0	0	0	0	0	4	30

Table 12—Prediction results for Blind well: precision, recall, and F1-scores using MLP classifier with optimum 2 hidden Layers.

	Precision	Recall	F1-Score
S.S.	0.75	0.07	0.12
Coarse Siltstone	0.34	0.61	0.44
Fine Siltstone	0.7	0.5	0.58
Shaley Siltstone	0.03	0.14	0.06
Mudstone	0	0	0
Wackstone	0.51	0.56	0.54
Dolomite	0.9	0.53	0.67
Packstone	0.33	0.5	0.4
Bafflestone	0	0	0
Avg.	0.42	0.42	0.42

Unsupervised learning: K-means Classifier

k-mean classifier creates distinct groups of data points such that points within different clusters yield different characteristics while points within the same cluster yield similar characteristics. Data points are grouped into distinct groups that define different lithologies where k is the number of groups. A lower *k* means larger groups with similar characteristics, while a larger *k* value generates smaller groups with more granularity. For this project, we used the "elbow" method to obtain an estimation of *k*-values and number of clusters. The line chart resembles an arm, where the inflection point on the curve before it plateaus is considered a good indication of the optimal number of clusters. As shown in Figure 11 and Figure 12, the elbow plot yield a plateau at the inflection point of 8 or 9. The y-axis represents the within-cluster sum of errors (wcss) which is the summation of each clusters distance between that specific clusters each point against the cluster centroid and can be calculated as follows;

$$WCSS = \sum_{P_i \text{ in Cluster } 1} distance\ (P_i,\ C_1)^2 + \sum_{P_i \text{ in Cluster } 2} distance\ (P_i,\ C_3)^2 + \sum_{P_i \text{ in Cluster } n} distance\ (P_i,\ C_n)^2$$

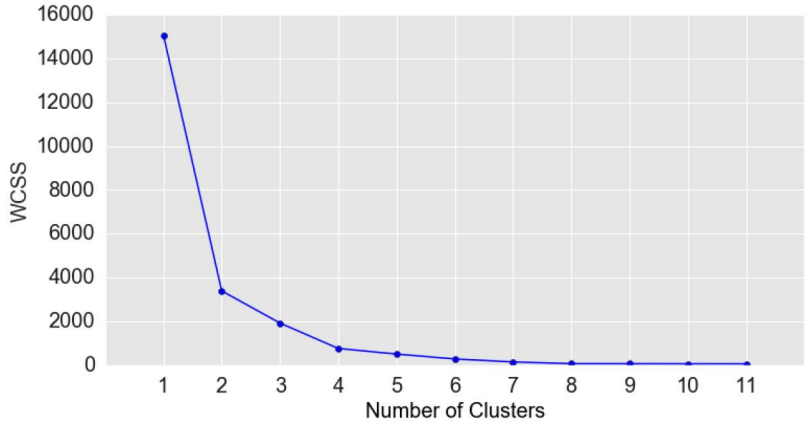

Figure 11—Elbow method for the optimal number of clusters

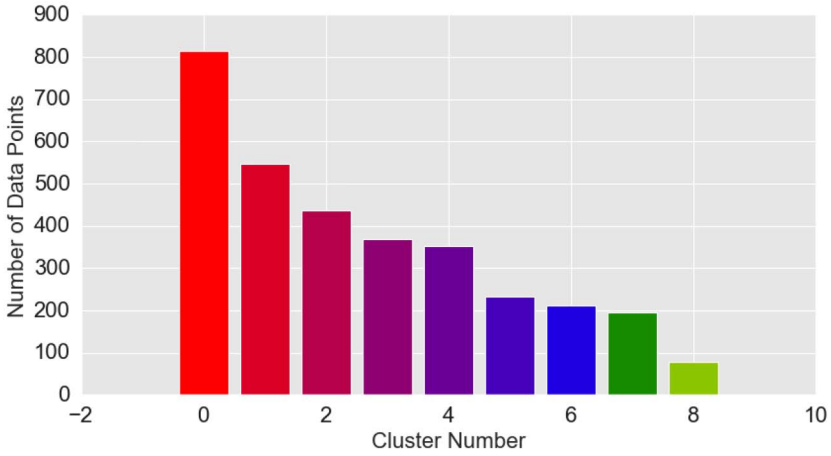

Figure 12—Distribution of data by optimal number of clusters=9

For this classifier, the average silhouette score is used to evaluate the accuracy of the model. It was found that the optimal number of clusters equals nine, with an average silhouette score equals to 83%. Another way to evaluate our clustering performance by using (t-SNE) t-Distributed Stochastic Neighbor Embedding, which is a nonlinear dimensionality reduction technique for embedding high dimension number of data to ultimately visualize the data in a low two-dimensional space with two dimensions (Maiti 2018). Each high dimensional object is modeled by a two- dimensional points in such a way that similar points are modeled by closer points and different points are modeled by distant points with high probability. Figure 13 shows the t-SNE algorithms clusters the raw data into nine classes. The axes are numbers that are generated when the high dimension data is mapped onto two dimensions. The figure shows very well distinct and separated clusters. Only a few points don't agree with clustering algorithm; few yellow points are classified with red and vice-versa and few yellow points are classified with blue.

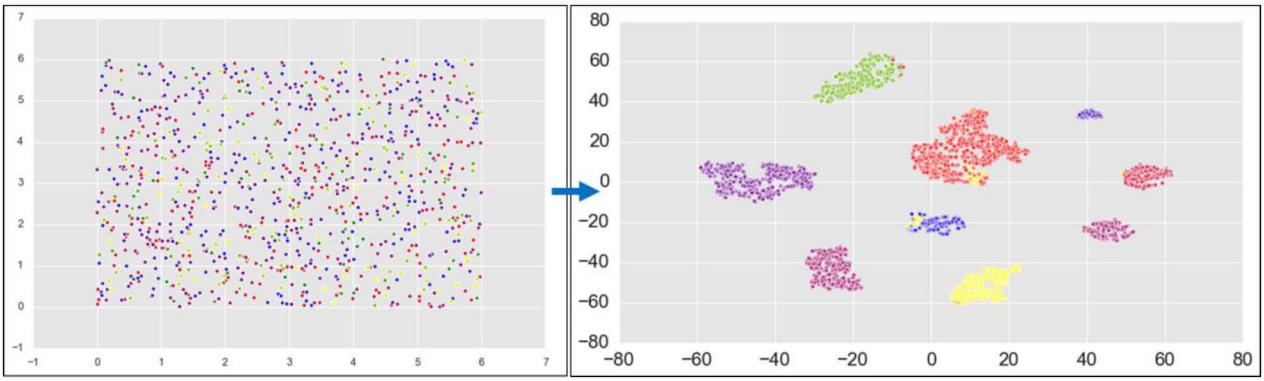

Figure 13—2D visualization of data using the t-SNE algorithm

Summary of Results

Table 12 summarize the results of all classifiers with optimum parameters. Overall, the supervised methods achieve better classification results compared with the other two approaches. MLP has the next-best performance, and the unsupervised learning algorithm has the worst performance.

From the confusion matrices shown above, we conclude that each model has a distinct behavior for the different lithology classes. All the models exhibit relatively high classification accuracy for coarse siltstone, sandstone, and fine siltstone compared with the other classes. It has been observed that coarse siltstone consistently yield the highest F1-score with all the supervised and unsupervised algorithms. This is due to coarse siltstone is more distributed among the well logs. Moreover, sandstone and siltstone yield properties

that are different and distinct from other available lithologies. While mudstone and shaley siltstone yield the lowest F1-scores, due to the scarcity of these two facies among the well logs. It was also observed that the MLP classifier yield the higher F1-score for predicting the dolomite.

Among the classifiers, the SVM has the best performance in distinguishing the sandstone classes; its average precision scores on the sandstone classes is 79%. The KNN algorithm has a relatively high classification accuracy for siltstone; its average precision scores on the siltstone classes is 88%.

KNN classifier failed to classify dolomite correctly; instead, it was classified as wackstone and packstone. Dolomite was successfully classified with an accuracy of 67% using the MLP neural network approach.

These results indicate that simple supervised methods are more applicable for lithology classification with the available limited well log data, while the neural network performs better in classifying dolomite classes and not as accurate in classifying the sandstone and siltstone classes. Figure 14 shows the well logs recorded for the blind test well and the predicted lithology using each classifier.

Table 12—Summary of the scores for different classifiers with optimum parameters. Results are presented in %.

Classifier	Category	Optimum Parameter	Cross-validation Test Data		Blind Well	
			F1-Score	Adj. Facies Accuracy	F1-Score	Adj. Facies Accuracy
SVM	Supervised	Gamma =100, C =0.01	66	84	49	89
KNN	Supervised	k =68	65	86	46	89
Random Forest	Supervised	$n_{estimator}$ =52	63	88	45	88
MLP	Feed-Forward Neural Network	2 Hidden Layers. Neurons 150,25	64	88	42	86
					k-Clusters	
			6	7	8	9
K-means	Unsupervised	Silhouette- Score	74	78	82	83

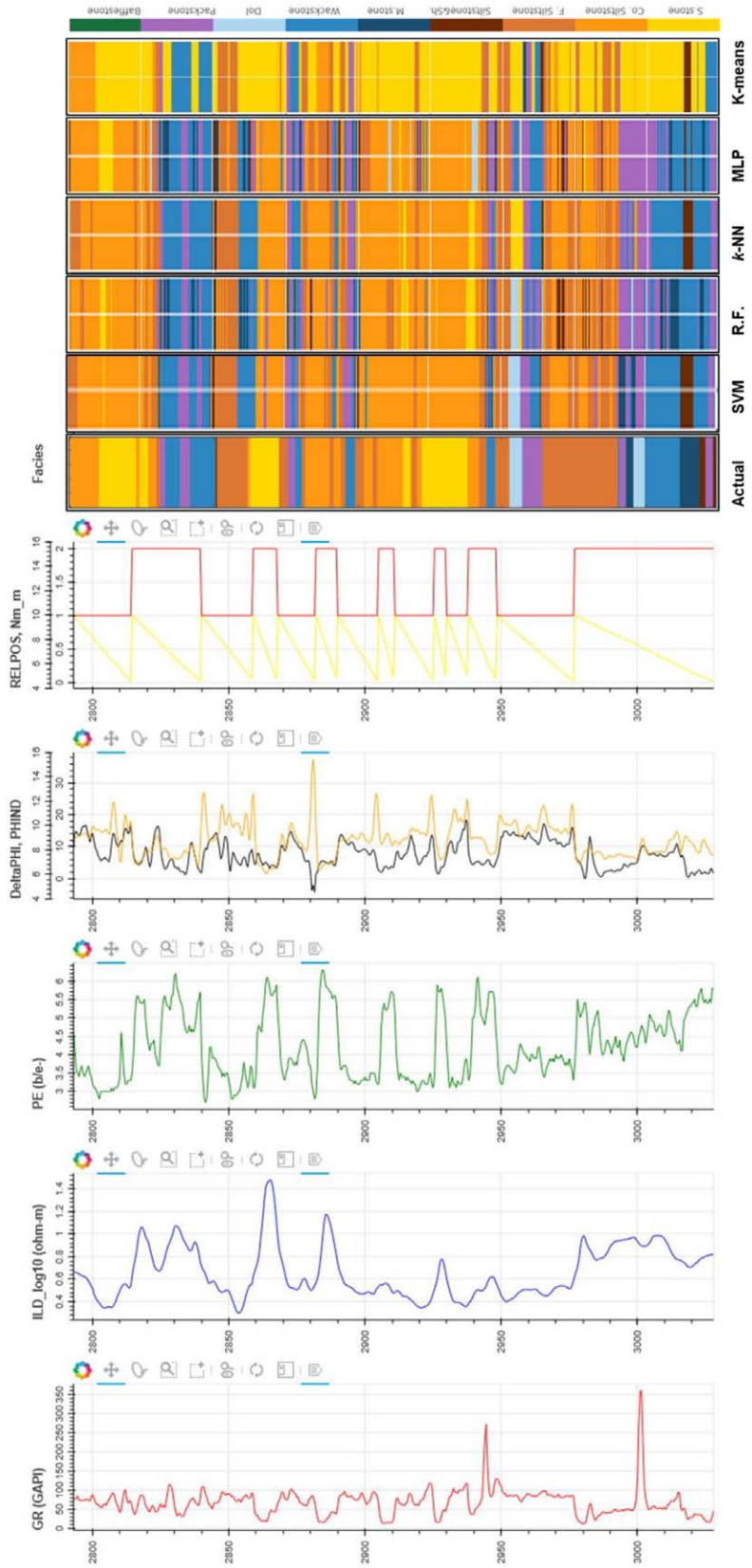

Figure 14—Well log data from left to right; GR (gamma ray), ILD_log10 (resistivity), PE (photoelectric effect), PHIND (porosity), DeltaPHI (neutron-density porosity difference), RELOPS (relative position) and nonmarine-marine indicator (NM_M). Actual Facies from cores and predicted facies using SVM, Random Forrest, K-NN, MLP Feed Forward Neural Network and K-means

Conclusion

Effectiveness of the classifier is based on the model hyperparameters for each classifier. Absolute correctness is important and desirable. However, considering there are no definite boundaries between facies, the fuzzy nature of the facies, the weighted average of the collected data, borehole environments and inherent error in the well log data, being close within adjacent facies (within one facies) is nearly as good as being accurate. Analysis of core samples in the lab can provide different interpretations, and it is an expensive and time-consuming operation. Manual Interpretation of lithology is a tricky, uncertain task that requires multiple days. Having a facies classification model that provide an approximate distribution of facies in relatively less time is another key objective.

- Supervised classified outperformed the unsupervised and the Feed-Forward neural network classifiers. The traditional parametric classifiers yield the highest performance when the features are standard normally distributed and when the optimum hyperparameter is selected.
- SVM is the clear winner in this study, ranking the highest accuracy of absolute correctness and with considering the adjacent facies. Nearest neighbor and random forest classifier have the next-best performance, followed by MLP Feed-Forward Neural Network and the unsupervised learning algorithm has the worst performance.
- None of the classifiers effectively predict the dolomite and the shaley sandstone possibly due in part to their small class size.
- All the models exhibit relatively high classification accuracy for coarse siltstone, sandstone, and fine siltstone compared with the other classes.

Acknowledgment

We very much appreciate the endorsement and the support of this work by Apache Corporation and its North American Unconventional Resource reservoir engineering team in San Antonio and UREP (Unconventional Reservoir Engineering Project) and Computer Science department at Colorado School of Mines. The author also would like to acknowledge the efforts of Nadima Dwihusna, Xiaoyu Zhu, and Saad Elbeleidy during this research.

References

Akinyokun, OC, Enikanselu, PA, Adeyemo, AB et al. 2009. Well log interpretation model for the determination of lithology and fluid contents. *The Pacific Journal of Science and Technology* **10** (1): 507–517.

Al-Anazi, A and Gates, ID. 2010. On the capability of support vector machines to classify lithology from well logs. *Natural Resources Research* **19** (2): 125–139.

A. Abdulraheem, M. Ahmed, A. Vantala, T. Parvez. 2009. Prediction of rock mechanical parameters for hydrocarbon reservoirs using different artificial intelligence techniques. Saudi Arabia Section Technical Symposium, *Society of Petroleum Engineers*

Cracknell, Matthew J and Reading, Anya 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.

Cracknell, MJ and Reading, AM. 2012. Machine Learning for Lithology Classification and Uncertainty Mapping. Proc., AGU Fall Meeting Abstracts.

Dubois, Martin K, Bohling, Geoffrey C, and Chakrabarti, Swapan. 2007. Comparison of four approaches to a rock facies classification problem. *Computers & Geosciences* **33** (5): 599–617.

Dubois, Martin K, Byrnes, Alan P, Bohling, Geoffrey C et al. 2006. Multiscale geologic and petrophysical modeling of the giant hugoton gas field (permian), Kansas and Oklahoma, USA.

Gifford, Christopher M and Agah, Arvin. 2010. Collaborative multi-agent rock facies classification from wireline well log data. *Engineering Applications of Artificial Intelligence* **23** (7): 1158–1172.

Harris, JR and Grunsky, EC. 2015. Predictive lithological mapping of Canada's North using Random Forest classification applied to geophysical and geochemical data. *Computers & geosciences* **80**: 9–25.

Horrocks, Tom, Holden, Eun-Jung, and Wedge, Daniel. 2015. Evaluation of automated lithology classification architectures using highly-sampled wireline logs for coal exploration. *Computers & geosciences* **83**: 209–218.

Hsieh, Bieng-Zih, Lewis, Charles, and Lin, Zsay-Shing. 2005. Lithology identification of aquifers from geophysical well logs and fuzzy logic analysis: Shui-Lin Area, Taiwan. *Computers & Geosciences* **31** (3): 263–275.

Kohavi, Ron. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proc., Ijcai2, 1137–1145.

Kotsiantis, Sotiris B, Zaharakis, I, and Pintelas, P. 2007. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering* **160**: 3–24.

Li, Yumei and Anderson-Sprecher, Richard. 2006. Facies identification from well logs: A comparison of discriminant analysis and naïve Bayes classifier. *Journal of Petroleum Science and Engineering* **53** (3-4): 149–157.

Maiti, Tannistha. Machine learning applied to geyophysical well log data. *Towards Data Science*.

Pippin, Lloyd. 1970. Panhandle-Hugoton field, Texas-Oklahoma-Kansas--the first fifty years.

Qi, Lianshuang and Carr, Timothy R. 2006. Neural network prediction of carbonate lithofacies from well logs, Big Bow and Sand Arroyo Creek fields, Southwest Kansas. *Computers & Geosciences* **32** (7): 947–964.

Rafik, Baouche and Kamel, Baddari. 2017. 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* **26** (3): 763–778.

Rogers, Samuel J, Fang, JH, Karr, CL et al. 1992. Determination of lithology from well logs using a neural network (1). *AAPG bulletin* **76** (5): 731–739.

Samuel, Arthur L. 1988. Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress. In *Computer Games I*, 366–400. Springer.

Schwab, Frederick L., Folk, Robert Louis, Beck, Kevin Charles et al. 2018. Sedimentary rock. *Encyclopædia Britannica*. https://www.britannica.com/science/sedimentary-rock.

Sebtosheikh, Mohammad Ali, Motafakkerfard, R, Riahi, Mohammad-Ali et al. 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** (1): 59–68.

Serra, O t and Abbott, HT. 1982. The contribution of logging data to sedimentology and stratigraphy. *Society of Petroleum Engineers Journal* **22** (01): 117–131.

Sorenson, Raymond P. 2005. A dynamic model for the Permian Panhandle and Hugoton fields, western Anadarko basin. *AAPG bulletin* **89** (7): 921–938.

S. Elkatatny, Z. Tariq, M. Mahmoud. (2016). Real time prediction of drilling fluid rheological properties using Artificial Neural Networks visible mathematical model. *J. Petrol. Sci. Eng.*, **146**, pp. 1202–1210

Wang, Guochang and Carr, Timothy R. 2012a. Marcellus shale lithofacies prediction by multiclass neural network classification in the Appalachian Basin. *Mathematical Geosciences* **44** (8): 975–1004.

Wang, Guochang and Carr, Timothy R. 2012b. Methodology of organic-rich shale lithofacies identification and prediction: A case study from Marcellus Shale in the Appalachian basin. *Computers & Geosciences* **49**: 151–163.

Xie, Yunxin, Zhu, Chenyang, Zhou, Wen et al. 2018. Evaluation of machine learning methods for formation lithology identification: A comparison of tuning processes and model performances. *Journal of Petroleum Science and Engineering* **160**: 182–193.