Please fill in the name of the event you are preparing this manuscript for.	2021 SPE Middle East Oil & Gas Show and Conference	
Please fill in your 6-digit SPE manuscript number.	SPE-204705-MS	
Please fill in your manuscript title.	Artificial Intelligence Aided Geologic Facies Classification in Complex Carbonate Reservoirs	
Please fill in your author name(s) and company affiliation.		
Given Name	Surname	Company
Yanhui	Zhang	KAUST
Ibrahim	Hoteit	KAUST
Klemens	Katterbauer	Saudi Aramco
Alberto	Marsala	Saudi Aramco

This template is provided to give authors a basic shell for preparing your manuscript for submittal to an SPE meeting or event. Styles have been included (Head1, Head2, Para, FigCaption, etc) to give you an idea of how your finalized paper will look before it is published by SPE. All manuscripts submitted to SPE will be extracted from this template and tagged into an XML format; SPE's standardized styles and fonts will be used when laying out the final manuscript. Links will be added to your manuscript for references, tables, and equations. Figures and tables should be placed directly after the first paragraph they are mentioned in. The technical content of your paper WILL NOT be changed. Please start your manuscript below.

Abstract

Facies classification for complex reservoirs is an important step in characterizing reservoir heterogeneity and determining reservoir properties and fluid flow patterns. Predicting rock facies automatically and reliably from well log and associated reservoir measurements is therefore essential in order to obtain accurate reservoir characterization for field development in a timely manner.

In this study, we present an artificial intelligence (AI) aided rock facies classification framework for complex reservoirs based on well log measurements. We generalize the AI-aided classification workflow into five major steps including data collection, preprocessing, feature engineering, model learning cycle, and model prediction. In particular, we automate the process of facies classification focusing on the use of a deep learning technique, convolutional neural network, which has shown outstanding performance in many scientific applications involving pattern recognition and classification. For performance analysis, we also compare the developed model with a support vector machine approach.

We examine the AI-aided workflow on a large open dataset acquired from a real complex reservoir in Alberta. The dataset contains a collection of well-log measurements over a couple of thousands of wells. The experimental results demonstrate the high efficiency and scalability of the developed framework for automatic facies classification with reasonable accuracy. This is particularly useful when quick facies prediction is necessary to support real-time decision making. The AI-aided framework framework is easily implementable and expandable to other reservoir applications.

Introduction

Facies classification plays an important role in both the exploration and the development of petroleum reservoirs. It is common that the variability and continuity of petrophysical properties of a reservoir is dominated by the facies heterogeneity. In reservoir characterization, facies modeling is an essential component to capture the geological complexity and to constrain the distributions of reservoir porosity and permeability so as to narrow down the range of uncertainty (Pyrch and Deutsch, 2014). Moreover, the distribution of facies often has a dramatic impact on the fluid flow patterns, which makes it an important

type of model parameter for history matching and production optimization, especially in mature complex reservoirs (Liu and Oliver, 2005; Zhang et al., 2014; Canchumuni et al., 2019).

The most direct way to identify the facies is to study the core samples acquired from the borehole. But there is typically only a limited number of core samples extracted in few well locations in practice due to technical and financial reasons. Conventionally, facies classification are carried out by coupling the analysis of available core samples with the interpretation of a suite of wellbore logs such as gamma ray, density, and resistivity logging (Dubois et al., 2007). In general, these types of well logs provide indirect but valuable information about rock composition, fluid content, and sedimentary structure of a reservoir. For manual classification, geologists and reservoir engineers identify and assign facies labels to certain depth intervals based on the interpretation of well logs and cores as well as their expertise and insights. However, the process of manual classification often incurs many challenges such as heavy workload of data analysis, low efficiency, and human subjectivity. It is a time consuming process and can become impractical particularly when large-dimensional data are involved. Therefore, developing a more powerful and automated classification approach is necessary to achieve reasonable characterization of subsurface reservoirs in a timely manner.

Significant research has been devoted to the development of automated workflows for facies classification over the past few decades. Machine learning (ML) as a subfield of AI enables computers to learn from available data without being explicitly programmed, and has therefore been increasingly studied and applied to streamline data management and analysis processes in oil industry. Various supervised ML algorithms have been proposed to expedite the facies classification with better accuracy (Dubois et al., 2007; Hall and Hall, 2017; Dell'Aversana, 2019; Silva et al., 2020). Artificial neural network (ANN) is a popular ML technique that has long been applied for facies classification (Baldwin et al., 1990; Rogers et al., 1992; Saggaf and Nebrija, 2000). Bhatt and Helle (2002) proposed to identify facies from well logs using an ANN approach, in which the neural networks were combined in ensembles and modular systems to improve the generalization performance of ANN beyond the training data. Bhattacharya and Mishra (2018) presented some successful applications of Bayesian network theory and random forest for facies and fracture prediction in heterogeneous reservoirs. Wang et al. (2014) applied support vector machine (SVM) classifier to recognize shale lithofacies from conventional logs where the relationship is complex and nonlinear. Their experimental results indicated reliable performance of SVM classifier with high cross-validation accuracy and stability. Silva et al. (2020) conducted a comparative study on petrofacies classification of a carbonate reservoir using five different ML algorithms including decision tree, random forest, gradient boosting, K-nearest neighbors, and naïve Bayes. The performances of these methods were analyzed based on a comprehensive set of evaluation metrics.

Recently, deep learning (DL) methods have drawn growing attention for facies classification by their ability to extract high-level features and learn more robust mapping functions (Jaikla et al., 2019; Wei et al., 2019). Imamverdiyev and Sukhostat (2019) presented an application of deep convolutional neural network (CNN) for facies classification using well logs. Their results indicated better performance of the deep CNN model compared to other tested DL models like recurrent neural network and long short-term memory models. For supervised learning algorithms, the adequacy of training or labeled data is one prerequisite to obtain reliable performances. Training on small dataset may lead to overfitted model with poor generalization performance. However, it is not uncommon in practical applications that the labeled data are limited while the unlabeled data are dominant, for example, due to the scarcity of core samples or in the case of seismic facies classification (Ross and Cole, 2017). To deal with this situation, semi-supervised learning methods have been proposed in which both labeled and unlabeled data are utilized in the training process in order to achieve better performances (Dunham et al., 2020).

In this study, we introduce a general-purpose facies classification workflow with a focus on the use of a deep learning technique of CNN. We examine the developed workflow on a large open dataset which includes a collection of well-log measurements over a couple of thousands of wells. We also compare the developed CNN model with an SVM approach.

AI-aided facies classification workflow

To build an AI-aided facies classification system, we follow a systematic data-driven modeling process that has been well established with broad applications and practice of ML techniques in various science fields. Figure 1 depicts a schematic flowchart of automated facies classification using ML algorithms. It generally consists of the following major steps:

1. Collecting sufficient amount of relevant data for the problem being considered is a prerequisite to obtain a reasonable result while applying ML algorithms for any learning tasks. In this study, a set of well log measurements obtained at certain depth intervals at a few well locations is used for the facies classification task.
2. In most cases, the raw data gathered in the first step are not in good shape to be used directly by ML algorithms. Preprocessing is generally necessary to transform the raw data into a more suitable form. For instance, this step may involve missing data imputation, handling of duplicates and outliers, and so on.
3. Feature engineering is one of the most important and involved steps in a ML framework. A feature refers to a numeric representation of an aspect of the original dataset. Feature engineering is the process of extracting and transforming the original data into features that better represent the underlying problem and are more suitable for a ML model. Well-determined features can ease the difficulty of the subsequent learning process and may result in improved model accuracy on unseen or future data.
4. In the learning step, the input features are fed to one or an ensemble of different ML classifiers (depending on specific applications), which are trained typically to optimize a specific objective function to achieve a good fit to the data as well as reliable predictive capability. The learning step is often an iterative process involving modeling, training, evaluation, and tuning substeps. In practice, it is common to compare multiple ML algorithms and choose the best performing model based on given model evaluation metrics, since no single ML classifier is superior in all circumstances without any assumptions.
5. Once the trained ML model achieves satisfying performance, the final goal is to deploy the learned model to make predictions on unseen or new data.

Figure 1—Schematic flowchart of automated facies classification using ML algorithms.

McMurray-Wabiskaw dataset

The McMurray-Wabiskaw dataset used in this study was made available for facies prediction by Gosses (2019), which originally came from a large open digital database published by Alberta Geological Survey's Alberta Energy Regulator (Wynne et al., 1994). The McMurray-Wabiskaw oil sands deposit is the largest oil sands deposit in Alberta. The released electronic data were one of the most significant products of the project that the Alberta Geological Survey initiated to map McMurray Formation and the overlaying Wabiskaw Member of the Clearwater Formation systematically in the Athabasca oil sands area. The original dataset contains well-log measurements on 2193 wells, including 750 wells that have core analyses. The log measurement spacing was 0.25 m.

The raw dataset is preprocessed to remove missing values and some types of well logs with minor amount of data entries, so that the processed dataset contains common well-log types for each well at every depth. We use four geophysical well logs as independent features including density porosity (DPHI), gamma ray (GR), induction deep resistivity (ILD), and neutron porosity (NPHI). There are six different types of facies: (1) sand, (2) shaly sand, (3) sandy shale, (4) shale, (5) coal, and (6) cemented sand.

ML algorithms

In this study, we focus on the use of a CNN technique to perform the facies classification task. To analyse its performance, we also compare it with a commonly used SVM approach.

CNN

The architecture of the proposed CNN model is schematized in Figure . It contains an input layer, three one-dimensional (1D) convolutional layers, a max-pooling layer, two full-connected (dense) layers and an output layer.

Figure 2—The overall architecture of the proposed CNN model.

The input vector consists of four features derived from the log measurements of DPHI, GR, ILD, and NPHI, which is firstly propagated through a sequence of convolutional and pooling layers to extract useful latent features. The convolutional layers are obtained by the multiplication of input data from the previous

layer and predefined filters (kernels) whose parameters will be learned from the training process. The filters scan through the input data volume and can detect a variety of latent patterns depending on the data and filter number. As the network goes deeper, more complicated or high-level features can be extracted. The same size of the filter, 2×1 (height $\times$ width), is used for each convolutional layer. The filter number is designed to increase gradually as 32, 64, and 128 for each convolutional layer in sequential order. The pooling layer is used to reduce the spatial dimensions of input convolutional layers and decrease the chance of overfitting. Here we use a max-pooling layer with a 2×1 filter that outputs the maximum value of each area of the input data scanned by the filter.

The high-level features produced at the end of the convolutional section will be firstly flattened to a long column vector and then pass through a fully connected section. There are two fully connected layers containing 500 and 200 units, respectively. A rectified linear unit (ReLU) activation function is adopted for the convolutional and fully connected layers. Lastly, the fully connected section is connected to an output layer (fully connected, with the dimensions of 6×1) that uses a softmax activation function to compute the predictive probabilities for all facies classes.

In addition, batch normalization is implemented after each convolutional and fully connected layers to normalize the output of a previous layer by re-centering and re-scaling. The batch normalization has been shown to be effective in improving the stability of a neural network and accelerating the training process. To find the best training parameters (network weights and biases), a cross-entropy loss function is considered to measure the difference between the prediction and the true label (facies class), which is minimized by a stochastic gradient descent method named as Adam optimizer.

SVM

SVM is a powerful ML algorithm which has been widely used in a variety of applications including classification, regression, and clustering (Mehne and Mirjalili, 2020). The main idea of SVM classification is to find a separating hyperplane (decision boundary) that maximizes the margin while ensuring classification accuracy. The margin refers to the distance between the decision boundary and the data points (so-called support vectors) that are closest to this boundary in the feature domain. Usually, learning models with large margins tend to have a lower generalization error while models with small margins are more prone to overfitting (Raschka, 2015). SVMs can achieve optimal performance for linearly separable classification problems. In the case of nonlinear classification problems, SVMs can perform a nonlinear classification in an efficient way through the so-called kernel trick. The rationale behind the trick is to map the input sample points into high-dimensional feature space through kernel functions, from where one can find a hyperplane to separate the samples.

In this study, we train an SVM for the facies classification using the scikit-learn library (Pedregosa et al., 2011), which provides a highly optimized implementation of the algorithm.

Experimental results and discussion

The proposed facies classification workflow is implemented with a set of highly efficient and well-developed ML libraries and platforms including Tensorflow and Keras. In the preprocessed McMurray-Wabiskaw dataset, there are data from 1230 wells containing more than 0.3 million samples.

The input feature vector is constructed using the log measurements of DPHI, GR, ILD, and NPHI. Figure shows a scatter matrix plot generated to visualize the pattern between different log measurements and the distribution of the well logs associated with corresponding facies. The diagonal panes depict the statistical distribution of each well log with regard to different facies classes. The off-diagonal panes disclose pairwise relationships between different well logs. The plot reflects one typical feature of facies classification problems that the facies clusters are partially overlapped in the feature space. Therefore, it can be difficult to separate one facies class from another either in a manual or an automatic approach. This challenge motivates the use of more advanced ML techniques for improved and reliable prediction ability.

Figure 3—Scatter matrix plot of considered well logs.

The dataset is randomly splitted into training (90%) and test (10%) sets. To enhance the performance of the ML models, feature scaling is implemented. Specifically, the features are standardized so that they have a zero mean and a unit variance. The constructed CNN model is trained by minibatch gradient descent, wherein the batch size is set to 2048.

Table 1 summaries the evaluation results and comparison of the performance of constructed SVM and CNN classifiers. The evaluation metrics considered here include Accuracy, Precision, Recall and F1 score. Accuracy computes the proportion of correctly classified samples. Precision measures the fraction of true positive instances among the ones classified as positive. Recall measures the fraction of true positive instances among all true positive ones labeled in the data. F1 score is a weighted harmonic mean of precision and recall. From the results outlined in Table 1, both SVM and CNN classifiers after training give comparable performance on the test dataset. We can see that both classifiers provide reasonable prediction on the facies classes of sand, cemented sand and shale, while perform poorly on the rest facies classes. Particularly for the shaly sand and sandy shale, the prediction accuracy is low. This is perhaps understandable if we look back at the facies distribution shown in Figure . These two facies classes are mostly overlapped with other facies, which makes it very difficult to identify by the classifiers.

Table 1—Evaluation results and comparison of the performance of different learning models.

Method	Evaluation metrics	Facies						
		Sand	ShalySand	SandyShale	Shale	Coal	CementedSand	Total
SVM	Accuracy	0.728	0.536	0.213	0.727	0.496	0.774	0.644
	F1	0.686	0.314	0.506	0.716	0.513	0.804	0.646
	Precision	0.649	0.594	0.480	0.705	0.531	0.837	0.651
	Recall	0.728	0.536	0.213	0.727	0.496	0.774	0.644
CNN	Accuracy	0.723	0.371	0.481	0.767	0.516	0.798	0.653
	F1	0.697	0.402	0.492	0.727	0.527	0.812	0.651
	Precision	0.673	0.440	0.502	0.691	0.593	0.825	0.651
	Recall	0.723	0.371	0.481	0.767	0.516	0.798	0.653

Figure shows the normalized confusion matrix obtained with the trained CNN model applied to the test dataset. The confusion matrix provides another way to examine the performance of a classification algorithm. Each row in the matrix denotes the instances in a predicted class and each column denotes the instances in an actual class. The principal diagonal of the matrix gives a better visualization of the prediction behavior that we found in Table 1. In addition, the off-diagonal elements of the matrix disclose the ratio of misclassification for each facies class. For example, we can see that it is difficult for the classifier to separate sandy shale from shaly sand and coal.

Figure 4—Confusion matrix for the trained CNN classifier.

In the experiments presented above, the developed AI-aided workflow may not have been implemented in an optimal way, in the sense that our attention has been focused on evaluating its feasibility and potential for facies classification under general conditions. There are several ways to further improve the performance of the developed workflow, for instance, through feature augmentation and refined systematic tuning of related hyper-parameters (Bestagini et al., 2017; Dell'Aversana, 2019). This will be considered in our future work.

Conclusions

In this study, we introduced a general-purpose AI-aided facies classification workflow with a focus on the use of CNN models based on well log measurements. Owing to the ever-growing computing power

and the easy accessibility of well-developed ML libraries and platforms, it allows an efficient implementation of the AI-aided classification workflow that covers data processing, feature engineering, model training, model evaluation, and model prediction. In particular, the high flexibility of the CNN model enables the developed workflow to be easily implementable and expandable to different reservoir applications under various conditions.

We verified the constructed CNN model with a large open dataset acquired from a real complex reservoir. The experimental results demonstrated the high efficiency of the developed workflow for automatic facies classification with reasonable prediction accuracy. For real-time decision making, the availability of quick and reliable facies prediction framework is particularly useful.

References

- Baldwin, J., Bateman, R., Wheatley, C., 1990. Application of a neural network to the problem of mineral identification from well logs. *Petrophysics* 31, 279–293.
- Bestagini, P., Lipari, V., Tubaro, S., 2017. A machine learning approach to facies classification using well logs, in: SEG International Exposition and 87th Annual Meeting. pp. 2137–2142. <https://doi.org/10.1190/segam2017-17729805.1>
- Bhatt, A., Helle, H.B., 2002. Determination of facies from well logs using modular neural networks. *Pet. Geosci.* 8, 217–228. <https://doi.org/10.1144/petgeo.8.3.217>
- Bhattacharya, S., Mishra, S., 2018. Applications of machine learning for facies and fracture prediction using Bayesian Network Theory and Random Forest: Case studies from the Appalachian basin, USA. *J. Pet. Sci. Eng.* 170, 1005–1017.
- Canchumuni, S.W.A., Emerick, A.A., Pacheco, M.A.C., 2019. History matching geological facies models based on ensemble smoother and deep generative models. *J. Pet. Sci. Eng.* 177, 941–958. <https://doi.org/10.1016/j.petrol.2019.02.037>
- Dell'Aversana, P., 2019. Comparison of different Machine Learning algorithms for lithofacies classification from well logs. *Boll. di Geofis. Teor. ed Appl.* 60.
- Dubois, M.K., Bohling, G.C., Chakrabarti, S., 2007. Comparison of four approaches to a rock facies classification problem. *Comput. Geosci.* 33, 599–617. <https://doi.org/10.1016/j.cageo.2006.08.011>
- Dunham, M.W., Malcolm, A., Kim Welford, J., 2020. Improved well-log classification using semisupervised label propagation and self-training, with comparisons to popular supervised algorithms. *Geophysics* 85, O1–O15. <https://doi.org/10.1190/geo2019-0238.1>
- Gosses, J., 2019. McMurray-Wabiskaw-preprocessed-datasets. GitHub Repos.
- Hall, M., Hall, B., 2017. Distributed collaborative prediction: Results of the machine learning contest. *Lead. Edge* 36, 267–269. <https://doi.org/10.1190/tle36030267.1>
- Imamverdiyev, Y., Sukhostat, L., 2019. Lithological facies classification using deep convolutional neural network. *J. Pet. Sci. Eng.* 174, 216–228. <https://doi.org/10.1016/j.petrol.2018.11.023>
- Jaikla, C., Devarakota, P., Auchter, N., Sidahmed, M., Espejo, I., 2019. FaciesNet: Machine Learning Applications for Facies Classification in Well Logs, in: Second Workshop on Machine Learning and the Physical Sciences at the 33rd Conference on Neural Information Processing Systems (NeurIPS). pp. 10–12.
- Liu, N., Oliver, D.S., 2005. Ensemble Kalman filter for automatic history matching of geologic facies. *J. Pet. Sci. Eng.* 47, 147–161. <https://doi.org/10.1016/j.petrol.2005.03.006>
- Mehne, S.H.H., Mirjalili, S., 2020. Support vector machine: Applications and improvements using evolutionary algorithms, in: *Evolutionary Machine Learning Techniques*. Springer, pp. 35–50.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., others, 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12, 2825–2830.
- Pyrz, M.J., Deutsch, C. V., 2014. Geostatistical reservoir modeling. Oxford university press.
- Raschka, S., 2015. Python machine learning. Packt publishing ltd.
- Rogers, S.J., Fang, J.H., Karr, C.L., Stanley, D.A., 1992. Determination of lithology from well logs using a neural network. *Am. Assoc. Pet. Geol. Bull.* 76, 731–739.
- Ross, C.P., Cole, D.M., 2017. A comparison of popular neural network facies-classification schemes. *Lead. Edge* 36, 340–349. <https://doi.org/10.1190/tle36040340.1>
- Saggaf, M.M., Nebrija, E.L., 2000. Estimation of lithologies and depositional facies from wire-line logs. *Am. Assoc. Pet. Geol. Bull.* 84, 1633–1646.
- Silva, A.A., Tavares, M.W., Carrasquilla, A., Misságia, R., Ceia, M., 2020. Petrofacies classification using machine learning algorithms. *GEOPHYSICS* 85, WA101–WA113. <https://doi.org/10.1190/geo2019-0439.1>
- Wang, G., Carr, T.R., Ju, Y., Li, C., 2014. Identifying organic-rich Marcellus Shale lithofacies by support vector machine classifier in the Appalachian basin. *Comput. Geosci.* 64, 52–60.

-
- <https://doi.org/10.1016/j.cageo.2013.12.002>
- Wei, Z., Hu, H., Zhou, H. wei, Lau, A., 2019. Characterizing Rock Facies Using Machine Learning Algorithm Based on a Convolutional Neural Network and Data Padding Strategy. *Pure Appl. Geophys.* 176, 3593–3605. <https://doi.org/10.1007/s00024-019-02152-0>
- Wynne, D.A., Attalla, M., Berezniuk, T., Brulotte, M., Cotterill, D.K., Strobl, R., Wightman, D., 1994. Athabasca oil sands data McMurray/Wabiskaw oil sands deposit—Electronic data, Alberta Geologic Survey, Technical report.
- Zhang, Y., Oliver, D.S., Chen, Y., Skaug, H.J., 2014. Data assimilation by use of the iterative ensemble smoother for 2D facies models. *SPE J.* 20, 169–185. <https://doi.org/10.2118/170248-PA>