MACHINE LEARNING FOR SUBSURFACE DATA ANALYSIS: APPLICATIONS IN OUTLIER DETECTION, SIGNAL SYNTHESIS AND CORE & COMPLETION DATA ANALYSIS

A Thesis

by

OSOGBA OGHENEKARO JEFFERSON

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Siddharth Misra,
Committee Members,	Bryan Maggard
	Sun Yuefeng
Head of Department,	Jeff Spath

December 2020

Major Subject: Petroleum Engineering

ABSTRACT

Application of machine learning has become prominent in many fields and has captured the imaginations of various industries. The development of data driven algorithms and the ongoing digitization of subsurface geological measurements provide a world of opportunities to maximize the exploration and production of resources such as oil, gas, coal and geothermal energy. The current proliferation of data, democratization of state-of-the-art processing technology and computation power provide an avenue for both large and small industry players to maximize the use of their data to run more economic and efficient operations. The aim of this thesis is to discuss the development of robust data-driven methods and their effectiveness in providing insightful information about subsurface properties. The study opens with a brief overview of the current literature regarding application of data driven methods in the oil and gas industry.

Outlier detection can be a strenuous task when data preprocessing for purposes of data- driven modeling. The thesis presents the efficacy of unsupervised outlier detection algorithms for various practical cases by comparing the performance of four outlier detection algorithms using appropriate metrics. Three case were created simulating: noisy measurements, measurements from washout formation and measurements from formations with several thin shale layers. It was observed that the Isolation Forest based model is efficient in detecting a wide range of outlier types with a balanced accuracy score of 0.88, 0.93 and 0.96 for the respective cases, while the DBSCAN based model was effective at detecting outliers due to noisy measurement with balanced accuracy score of 0.93.

NMR measurements provide a wealth of geological information for petrophysical analysis and can be key in accurately characterizing a reservoir, however they are expensive and technically challenging to deploy, it has been shown in research that machine learning models can be effective in synthesizing some log data. However, predicting an NMR distribution where each depth is represented by several bins poses a different challenge. In this study, a Random Forest model was used for predicting the NMR T1 distribution in a well using relatively inexpensive and readily available well logs with an r2 score and corrected Mean absolute percentage error of 0.14 and 0.84. The predictions fall within the margin of error and an index was proposed to evaluate the reliability of each prediction based on a quantile regression forest to provide the user more information on the accuracy of the prediction when no data is available to test the model as will be the case in real world application. Using this method engineers and geologist can obtain NMR derived information from a well when no NMR tool has been run with a measure of reliability for each predicted sample/depth.

Identifying sweet spots in unconventional formations can be the difference between an economically viable well and a money pit, in this study clustering techniques in conjunction with feature extraction methods were used to identify potential sweet spots in the Sycamore formation, elemental analysis of the clusters identified the carbonate concentration in sycamore siltstones as the key marker for porosity. This provided information as to why some layers had more production potential than the others. Machine learning algorithms were also used to identify key parameters that affect the productivity of an unconventional well using data from a simulation software. 11 completion parameters (lateral spacing, area (areal spacing), total vertical depth, lateral length, stages, perforation cluster, sand intensity, fluid intensity, pay thickness, fracture ½ length and fracture conductivity lateral length) were used to predict the EUR and IP90 using a random forest model and the normalized mean decrease in impurity was used to identify the key parameter. The lateral length was identified as the key parameter for estimated ultimate recovery and perforation clusters the key parameter for higher IP90 with a normalized mean decrease in impurity of 0.73 and 0.88 respectively.

Machine learning methods can be integrated to optimize numerous industry workflows and therefore has huge potential in the oil and gas industry. It has found wide applications in automating mundane tasks like outlier detection, synthesizing pseudo-data when true data is not available and providing more information on technical operation for sound decision making.

DEDICATION

I would like to dedicate this thesis to the Almighty God for his guidance, protection and favor throughout my Master's program and my lovely Mother, Ruth Osarogue Osogba.

ACKNOWLEDGEMENTS

The author wishes to thank the following individuals who have contributed immensely to this research project:

Dr. Siddharth Misra for the rigorous training, guidance and attention to detail that made the work in this thesis come to fruition,

Dr. Sun Yuefeng and Dr. Bryan Maggard for serving as members of my thesis advisory committee.

I also wish to thank my friends and colleagues, as well as the department faculty and staff (special thanks to Mrs. Eleanor Schuler) for making my time at Texas A&M University a great experience.

Finally, I would like to thank my Mother and my family for all their support, kindness and love throughout my Master's program.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis (or) dissertation committee consisting of Dr. Siddharth Misra, Dr. Sun Yuefeng, and Dr. Bryan Maggard of the Harold Vance Department of Petroleum Engineering at Texas A&M University.

Funding Sources

This graduate study was supported by a research assistantship from the Texas Engineering Experiment Station and the Department of Petroleum Engineering at Texas A&M University, both parts of the start-up funding of the projects of Dr. Siddharth Misra.

This work was made possible mainly by the start-up funding of the projects of Dr. Siddharth Misra, whose work on the subject was also partially funded by the U.S. Department of Energy.

The contents and results of this work are solely the responsibility of the student and his advisory committee, and do not necessarily represent the official views of the funding sources cited above.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

Research Motivation

Machine learning has risen to prominence in the 21st century. Given the substantial increase in the amount of data and computational power available, machine learning methods have shown their ability to provide valuable insights into our data.

Machine learning has seen significant success across many industries. For example, outlier detection models are applied in detecting fraudulent transactions in the finance industry, classification and clustering algorithms are used in building recommendation systems in the e-commerce industry, and regression models are used in a variety of industries for different types of forecasting. Considering the levels of success that these models have seen and their diversity of application it is only reasonable that this method should be applied in the oil and gas industry.

Subsurface analysis and geological interpretation is challenging, given that the subsurface itself is complex and requires data from various sources (seismic, well logs, and core data, to name a few) and a combination of empirical and theoretical analysis to approximate certain parameters required for the successful and economic exploration and exploitation of mineral reserves. Equations based on these approaches have seen success so far. However, could a data driven approach to the analysis of the data be better, faster, or more economical? This is the question that is currently being asked by a number of researchers in the oil and gas industry. Given the success of data driven approaches in other industries and the wealth of data available in the industry, it is no surprise that there is a significant amount of attention being paid to the application of machine learning algorithms in petroleum engineering.

Several authors have published works that have shown with varying degree of success the application of machine learning methods in petroleum engineering and some will be discussed in chapter 2.

In this work I will be exploring the use of machine learning models and concepts in analyzing data obtained from a borehole (well logs). These logs provide valuable information that are used for oil, gas, water, mineral and geothermal exploration as well as environmental and geotechnical studies. They are usually available in most drilled well, increasing the applicability of this study.

Overview of Machine Learning

Machine learning can be seen as the use of algorithms or a series of algorithms to identify patterns/trends in data. Several machine learning models which would be discussed later in this work have different ways of identifying these patterns and a good understanding of the algorithm and data are key to a successful machine learning application.

Machine Learning Terminology

The following section lists terminology commonly used when dealing with topics revolving around Machine Learning applications:

- **Feature**: A feature/feature variable is a property or attribute of the phenomenon being observed/investigated, e.g. if we are examining house prices, one of the features might be the neighborhood a house is situated. It is commonly denoted using **X**. Other names for feature are **predictor** variable, **independent** variable etc.
- **Target**: The target is the property itself that seeks to be predicted, using the previous example the house price would be the target. It is commonly denoted using **y**. Another name for target is the dependent variable.
- **Sample** and **Dataset**: A sample/datapoint refers to a single row in the feature vector and/or target vector e.g. using the house analogy we can observe a single house in neighborhood A with price $X. Another name for sample is **instance.** The dataset refers to all the samples available. A sample (*s*) is member of the entire dataset (*D*)
- $\boldsymbol{s} \in \boldsymbol{D}$**Algorithm**: An algorithm is typically defined as specific set of instruction given to a computer to achieve a task. In machine learning, an algorithm is a process to find an equation or set of equations that describes certain statistical patterns and relationships in a dataset. There are several machine learning algorithms which have unique behaviors and underlying principles. A good

understanding of this machine learning algorithms is key to building a good and interpretable model.

- **Fitting**: Fitting is the process whereby the machine learning algorithm "learns" the relationship or patterns within the dataset, it could be between the feature and target as in supervised learning or just the feature as in unsupervised learning. When a machine learning algorithm is fit with a dataset, the product is a machine learning **model**. Fitting is also referred to as training.

Machine Learning Algorithms

Machine learning algorithms can be broadly classified into two groups: **supervised** and **unsupervised**.

Supervised Algorithms

Supervised machine learning algorithms seek to "learn" the relationship between the feature and target. It does this by creating a function that maps the inputs (feature variables) to the output (target variable). In supervised learning a feature and target dataset are **required** to fit/train the model. The subsequent model is then used to predict on an unknown/unseen feature dataset with no corresponding target.

Supervised algorithms are classified as either **regression** or **classification**. The major difference between regression and classification is that in regression the target dataset is a set of **continuous variables** e.g. house prices ($75,234, $94,456, $589,456,

$254,189, etc.) whereas in classification the target dataset is a set of **discrete variables** e.g. house type (bungalow, story-building, apartment). Supervised learning algorithms are usually capable of handling both regression and classification tasks, typically the algorithm would be designed to handle one case and modified to handle the other. Examples of supervised learning algorithms: Linear regression, Support Vector Machine, Random Forest, etc.

Unsupervised Algorithms

Unsupervised machine learning algorithms seeks to “learn” patterns within a dataset without any user assigned label/target i.e. as the name implies without or with little supervision. In unsupervised learning the input is the feature dataset (or simple the dataset) and the output would depend on the algorithm.

Unsupervised algorithms can be used for **clustering**, **probability density estimation, outlier detection** and **dimensionality reduction**. Unsupervised learning algorithms are powerful tools and can be used for fraud detection, for example, your bank will instantly alert you if you make purchase outside your usual “pattern” this is usually done using unsupervised machine learning algorithms, it also used in e-commerce sites for customer-centric recommendations, for example, after shopping a while in your preferred site you can see “similar customers bought” icons, this is usually done using unsupervised algorithms.

Typical Machine Learning Workflow

Each machine learning based study will have a workflow and would be different from another based on the expected outcomes. However, most machine learning workflows, complex or simple will follow a particular schema. This schema is illustrated in Figure I-1 and component parts will be explained in subsequent sections.

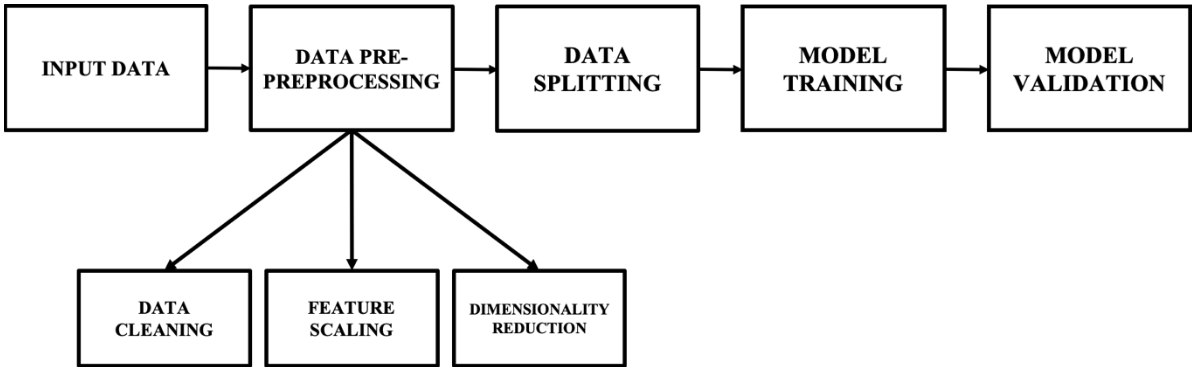

Figure I-1: Typical Machine Learning Workflow

Data Preprocessing

Data pre-processing is a broad term with respect to machine learning and data analytics and generally refers to all the steps taken to prepare a dataset (input data) to be properly fit by an algorithm. It is a key step and significantly affects model results accuracy and should be taken seriously. Some important pre-processing steps are outline below:

- **Data cleaning**: This term refers to operations performed to the dataset to make a dataset mathematically ready for fitting, most dataset in their original form would be impossible to fit for several reasons e.g. presence of non-numerical characters, missing data, poorly filled dataset. Steps taken to remedy this situation are referred to as data cleaning and include but not limited to filling missing data with user accepted value (mean, median, interpolated values etc.), replacing non-numerical values with numerical values (e.g. Yes/No to 0/1) etc. In most cases, when a data is "clean" there are still several steps needed to be taken on dataset to build an optimum model.
- **Feature Scaling**: is a key operation in pre-processing as the dataset will have several features from different source and magnitudes. This different magnitude will have an adverse effect on most popular machine learning algorithms, feature scaling attempts to "level the playing field" and transforms the features in the dataset to the same or near same magnitudes which would in most cases will lead to a better and more accurate model. Feature scaling is explained more in subsequent section
- **Dimensionality Reduction**: refers to the transformation of the input data set from a high dimensional space (large number of feature vectors) to a low dimensional space (small number of feature vectors). The goal of the user when performing dimensionality reduction is to reduce this dimensional space while retaining as much information from the dataset as possible. The need for dimensionality reduction arises because for distance-based algorithms (algorithms that measure

the distance between samples), in high dimensions (n << p) the concept of distance becomes distorted and does not work very well. The most common dimensionality reduction methods are **feature extraction** and **feature selection**.

- **Feature extraction** involves deriving a new feature set from the original feature set by performing a mathematical operation on the initial set, feature extraction does not necessarily lead to dimensionality reduction but can be used for it. It is popularly used in image processing, pattern recognition and signal processing. Some feature extraction methods are: Principal Component Analysis (PCA), auto-encoding, etc.
- **Feature selection** involves selecting the subset of the original feature set that is most relevant to the machine learning task, some popular methods used for feature selection are: F-test, Chi-square test, mutual information etc. Some of which are discussed later in this work.

Data Splitting

Data Splitting is an important step in the machine learning workflow as it provides a dataset for which the model can be tested, it is most relevant to supervised learning. The common convention is to split the dataset into a 70:30 ratio, with 70% of the dataset used for the training the model (train dataset) and 30% of the dataset used for testing the accuracy of the model. The split is done randomly to remove as much bias as is possible.

Model Validation and Evaluation

The model (supervised learning) is validated by comparing the predicted results and actual results using a metric, the selection of this metric is key as model is only as accurate as the metric you used. Popular metrics used for regression are root mean squared error (RMSE), mean absolute error (MAE), mean and absolute percentage error (MAPE), for classification evaluation F1-score, accuracy score, precision and recall are used. Some of which are discussed in length in subsequent section.

CHAPTER II

LITERATURE REVIEW: MACHINE LEARNING APPLICATION IN OIL AND GAS

Several authors have applied several machine learning methods in analyzing and interpreting the data source from the oil and gas industry. This research work cuts across key areas in all area of the oil and gas industry. Some of these bodies of work will be discussed in the chapter.

Machine Learning Application in Drilling

Machine learning methods have been applied successfully in handling several drilling related problems and a select few will be discussed in this section.

Pollock et al. [1] used a combination of supervised, unsupervised and reinforcement methods to create a model that can be used to automatically set tool alignment and force control during directional drilling. Using historical data (bit depth, hole depth, hook load, weight on bit, differential pressure, total pump output and other extraneous variables) from 14 horizontal wells in Appalachia and Permian basin. The wells were selected on the basis that their trajectory matched well with the planned well trajectory. A hierarchical clustering model was used to identify closely related features while a GAN (generative adversarial network and LSTM (long short-term memory) was used in identify the sliding section during drilling. The data is passed into a neural network and the output is compared with the action taken by the directional driller. During the training process this output is iteratively updated until the error between the neural networks output and drillers action is minimized. When the model was applied on a test dataset (new/unseen to the model) the differential pressure and rotary torque normalized percentage error was dropped to 0.21% and 2.7% respectively.

Zhao et al. [2] developed a machine learning based system that is used to detect precursors of drilling events (severe vibration, stuck pipe, fluid loss, sudden equivalent circulating density change) with emphasis on stick-slip vibrations using a feature dataset comprising of surface data, wellbore geometry data, lithology and several downhole measurements. A hierarchical clustering model is first used to identify trends such as: stable, ramp up, step up, pulse down, ramp down, step down and pulse up in the processed time-series drilling data. Using the change in drilling condition from one trend to the other the author concluded that this method can automatically inform drillers when an unusual drilling event occurs.

Zhong et al. [3] applied several classification methods (support vector machine, artificial neural network, random forest and gradient boosting) to identify coals beds using MWD (measurement while drilling) and LWD (logging while drilling) measurements. The dataset was obtained from 6 wells in the Surat basin in Australia. The author concluded that machine learning methods can accurately predict coal pay zones which can consequently reduce drilling down time and reduced cost related to coring or density log coal bed identification while drilling, the author also recommended the use of Neural Network or Random forest for multi well application.

Bhowmik et al. [4] compared the use of two learning models (Random Forest and Radial basis function) coupled with genetic algorithm for riser design automation and the traditional manual optimization for riser optimization configuration. The author concluded that the machine learning based meta models performed better in terms of computational time and cost as compared to the traditional manual optimization. They also noted that the Random Forest model performed best of all.

Machine Learning Application in Reservoir Engineering and Petrophysics

Machine learning methods have been applied successfully in handling several reservoir engineering and petrophysics related questions or problems and a select few would be discussed in this section.

Wu et al. [5] proposed a method for locating kerogen/organic matter and pore in SEM images of shale samples using a Random Forest classifier on features engineered from SEM Images, the author concluded that this method which had an F1 score of 0.9 on a validation dataset was more reliable and robust method when compared to popular methods of threshold and object-based segmentation for locating pores and organic matter.

Jiabo et al. [6] developed a model using several machine learning models: Linear Regression, Partial Least Squares, LASSO (Linear Absolute Shrinkage Selector Operator), ElasticNet, MARS (Multivariate Adaptive Regression Splines) and Neural Networks with feature set comprising of resistivity, neutron-density, gamma ray, caliper, photoelectric factor and synthetic lithological logs in predicting compressional and shear velocity for geomechanical characterization of shale wells. In both cases the Neural Network outperformed the other algorithms and produced the most accurate model.

Li et al. [7] proposed a method for predicting NMR T2 distribution using variable auto encoders with mineralogy and fluid saturation logs as features from a well located in the Bakken formation. It was observed that with hydrocarbon containing pores having sizes ranging from 9 to 2349.9 nm corresponding to bin 2 and 3 having high R2 score, the model prediction had. An average R2 score of 0.78.

Gaganis et al. [8] used a feed-forward neural network to predict the phase equilibrium coefficients k_i as an alternative to the various phase split approaches that require large computational power and multiple iterations. The authors did this by "treating the phase-split problem as a function learning one" and obtaining an accurate approximation of the function by using the neural network to provide a mapped function.

Onwuchekwa et al. [9] used a host of machine learning algorithms: K Nearest Neighbors, Support Vector Regression, Kernel Ridge Regression, Random Forest, Adaptive Boosting and Collaborative filtering to predict reservoir fluid properties, a feature set consisting of initial reservoir pressure, saturation pressure, solution gas oil ratio, formation volume factor, condensate gas ratio, API gravity, gas gravity, saturated oil viscosity and dead oil viscosity from 296 oil and 72 gas reservoirs in the Niger Delta and used them to predict formation volume factor, oil viscosity and condensate gas ratio. The author concluded that all techniques performed comparably or better than the industry standard of Standing and Vasquez-Beggs correlation in predicting oil formation volume factor, for oil viscosity the Random Forest and Adaptive Boosting gave comparable results with Beggs-Robinson correlation and did not require dead oil viscosity, although the model performed “not as good” in predicting condensate gas ratio, the author hypothesized this to the limited amount of data from the gas reservoirs compared to the oil reservoirs. **Son et al.** [10] used an ElasticNet model to predict fluid saturation from NMR 1D T1-T2 on core samples from the Meramec and successfully compared the result with fluid saturations gotten from a T1-T2 2-D map.

Machine Learning Application in Production Engineering

Machine learning methods have been applied successfully in handling several production related questions or problems.

Cao et al. [11] proposed a data-driven method for predicting production flowrates, they considered 2 cases, one involving predicting future flow rates from an existing well and another involving predicting flow rates from a new well. A neural network was used for the prediction with production rate history and tubing head pressure used for model training in case 1 and the production history combined with geological properties, tubing head pressures from surrounding wells used for model training in case 2. The method provides more detail than the conventional decline curve analysis when forecasting and is less cumbersome than the reservoir simulation techniques. The author argues that this should not be replacement for these methods but a way to validate existing forecasts.

Ounsakul et al. [12] proposed the use of machine learning methods in artificial selection, the authors gathered an initial feature set of 50 variables: well parameters, production conditions, fluid properties, reservoir parameters, surface facilities, probability factors, supplier factors and HSE (health, safety and environment) consideration. Only samples that met the required threshold for cost/barrel were selected and the different artificial methods (gas lift, beam pump, ESP, PCP) were targets. Three algorithms: Naïve Bayes, Decision tree and Neural Network were used, and the decision tree had the highest accuracy with a reported accuracy, precision, recall and F1 score of 0.94.

Machine Learning Application in Midstream, Downstream and Facilities

Patel et al. [13] used clustering techniques to improves the efficiency of time-consuming nature of advanced process control in oil fields with hundreds of wells, using a case study of over 300 wells in a large conventional oil field in Saudi Arabia, the authors were able to cluster wells with similar features (well parameters) into groups and perform conventional APC (advance process control) methods on those wells. The authors recommended this method for advanced process control in fields with multiple wells given the success of the study.

Omrani et al. [14] were able to use machine learning models PCA (principal component analysis) for dimensionality reduction and Artificial Neural network to classify slug flows in wells and subsea risers using flow data from multiphase flow simulator.

The current literature on the application of machine learning and data driven methods in petroleum engineering and mineral exploitation in general is vast and the above mentioned research works are just select materials in the vast amount of research available on the topic.

CHAPTER III

EFFICACY OF UNSUPERVISED OUTLIER DETECTION METHODS ON SUBSURFACE DATA

In this chapter, I will discuss outlier detection methods for subsurface data. Outlier detection is an important step for a number of petrophysical and production related analysis because the presence of outliers can adversely affect the results of such analysis. Here I will discuss outlier types, importance, the limitations of univariate outlier detection techniques and the advantages of using unsupervised learning to identify outliers in multivariate data. This chapter will

- provide an overview of unsupervised outlier detection methods
- introduce four outlier detection algorithms
- compare the performance of the different outlier detection algorithms on different configuration of data

Introduction

Outliers are datapoints (samples) that are significantly different from the general trend of the dataset. A sample is considered as an outlier when its attributes do not represent the behavior of the phenomenon/process in comparison with most of the samples in a dataset. Outliers are indicative of issues in data gathering/measurement process or rare events in the mechanism that generated the data. Identification and removal of outliers is an important step prior to building a data-driven model. Outliers skew the descriptive statistics used by data analysis and machine learning algorithms which are required to build a data-driven model. A model developed on data containing outliers may not accurately represent the normal behavior of data because the learned model contains unrepresentative patterns due to the presence outliers. Outliers in a dataset affect the predictive accuracy and generalization capability of the created model. In the context of this work, outliers can be broadly categorized into three types: point/global, contextual, and collective outliers [15].

Point/global outliers refer to individual datapoint or sample that significantly deviates from the overall distribution of the entire dataset or from the distribution of certain combination of features. These outliers exist at the tail end of a distribution and largely vary from the mean of the distribution e.g. subsurface depths where gamma ray reading spike above 2000gApi or well producing at an average rate of 200bbl/day having a recorded production of 1500bbl/day on a given day should be considered outliers. From an event perspective, getting the winning ticket in a national lottery is an example of a point outlier.

Contextual/conditional outliers are points that deviate significantly from the data points within a specific context; e.g. a large gamma ray reading in sandstone due to an increase in potassium-rich minerals (feldspar). Snow in summer is an example of contextual outlier, snow in most US north-eastern states is not necessarily an outlier but when it occurs in June, in the context of seasons it becomes an outlier, same is the case with the gamma ray reading. High gamma ray readings are not necessarily outliers but when a high gamma ray reading occurs in sandstone or coal bed that point can be labelled as an outlier. Points labeled as contextual outliers are valid outliers only for a specific context; a change in the context (e.g. snowing in January and high gamma ray reading in shale) will result in a similar point to be considered as an inlier.

Collective outliers are small cluster of data which as a whole deviate significantly from the entire dataset; e.g. log measurements from regions affected by borehole washout. For example, it is not rare that people move from one residence to the next; however, when an entire neighborhood relocates at the same time, it will be considered as a collective outlier. As regards to subsurface characterization, outliers in well logs and subsurface measurements occur due to wellbore conditions, logging tool deployment, and physical characteristics of the geological formations. For example, washed out zones in the wellbore and borehole rugosity significantly affects the readings of shallow-sensing logs, such as density, sonic, and photoelectric factor (PEF) logs, resulting in outlier response. Along with wellbore conditions, uncommon beds and sudden change in formation properties in the formation also result in outlier behavior of the subsurface measurements.

Outlier handling refers to all the steps taken to negate the adverse effect of outliers in a dataset. After detecting the outliers in a dataset, how they are handled depends on the immediate use of the dataset. Outliers can be removed, replaced or transformed depending on the type of dataset and its use. Outlier handling is particularly important as outliers could enhance or mask relevant statistical characteristics of the dataset. For instance, outliers in weather data could be early signs of a weather disaster, ignoring this could have catastrophic consequences, outliers in real time MWD (measurement while drilling) could be early signs of a kick. However, before consider handling outliers they must first be detected. In this chapter, I will apply four unsupervised outlier detection techniques (ODTs) on various original and synthetic log datasets. Following that, a comparative study of these unsupervised techniques for purposes of log-based subsurface characterization.

Overview of Outlier Detection Models

Outlier detection methods detect anomalous observations/samples that do not fit the typical/normal statistical distribution of a dataset. Simple methods for outlier detection use statistical tools, such as boxplot and z-score which are based on univariate analysis.

A boxplot is a standardized way of representing the distribution of samples corresponding to various features using boxes and whiskers. The boxes represent the inter-quartile range of the data while the whiskers represent a multiple of the first and third quartile of the variable, any datapoint/sample outside these limits is considered an outlier. The next simple statistical tool for outlier detection is the Z-score, which indicates how far a datapoint/sample is from its mean for a specific feature. A Z-score of 1 means the sample point is 1 standard deviation away from its mean. Typically, Z-score values greater than or less than +3 or -3 respectively are considered outliers. However, those values can be changed depending on the preference of the user. Z-score is expressed as:

$$\mathbf{Z - score = \frac{x_i - \bar{x}}{\sigma}} \qquad \textbf{Equation III-1}$$

where,

$x_i = sample\ instance$

$\bar{x} = mean\ of\ distribution$

$\sigma = standard\ deviation\ of\ distribution$

Outlier detection based on simple statistical tools generally assume that the data has a normal distribution and do not consider the correlation between features in a multivariate dataset. Advanced outlier detection methods based on machine learning (ML) can handle correlated multivariate dataset, detect abnormalities within them, and does not assume a normal distribution of the dataset [53]. Well logs and subsurface measurements are sensing heterogenous geological mixtures with lots of complexity in terms of the distributions of minerals and fluids; consequently, these measurements generally do not exhibit Gaussian distribution and generally exhibit considerable correlations within the features. Data-driven outlier detection techniques built using machine learning are more robust in detecting outliers as compared to simple statistical tools.

Outliers in dataset can be detected either using supervised or unsupervised ML technique. In supervised ODT, outlier detection is treated as a classification problem. The model is trained on dataset with samples pre-labelled as either normal data or outliers. The model then learns to assign labels to the samples in a new unlabeled dataset as either inliers or outliers based on what was “learned” from the training dataset. Supervised ODT is robust when the model is exposed to a large, statistically diverse training set (i.e. dataset that contains every possible instance of normal and outlier samples), whose samples are accurately labelled as normal or outlier. Unfortunately, this is difficult, time consuming and sometimes impossible to obtain as it requires significant human expertise in labeling and expensive data acquisition to obtain the large dataset. On the contrary, unsupervised

ODT overcomes the requirement of labelled dataset. Unsupervised ODT assumes: (1) number of outliers is much smaller than the normal samples and (2) outliers do not follow the overall ‘trend’ in the dataset.

Figure III-1 shows various outlier detection models currently in use and their mode of operation. Both supervised and unsupervised outlier detection techniques are used in different industries. For instance, in credit fraud detection neural networks are trained on all known fraudulent and legitimate transactions, and every new transaction is assigned a fraudulent or legitimate label by the model based on the information from the train dataset. It could also be trained in an unsupervised manner by flagging transactions that are dissimilar from what is normally encountered. In medical diagnosis, outlier detection techniques are used in early detection and diagnosis of certain diseases by analyzing the patient data (e.g. blood pressure, heart rate, insulin level etc.) to find patients for whom the measurements deviate significantly from the normal conditions. Zengyou et al. [16] used a cluster based local outlier factor algorithm to detect malignant breast cancer by training their model on features related to breast cancer. Outlier detection techniques are also used in detecting irregularities in the heart functioning by analyzing the measurements from an ECG (Echo Cardiogram) for purposes of early diagnosis of certain heart diseases. In the oil and gas industry, Chaudhary et al. [17] was able to improve the performance of the SEPD (Stretched Exponential Production Decline) model by detecting and removing outliers from production data by using the Local Outlier Factor method. In another oil and gas application, Luis et al. [18] used one-class support vector machine (SVM) to detect possible operational issues in offshore turbomachinery, such as pumps, compressors, by detecting anomalous signals from their sensors. When implementing an unsupervised outlier detection model, a prior knowledge of the expected fraction of outliers improves the accuracy of outlier detection. In many real-world applications, these values are known. For example, in the medical field, there is a good estimate of the fraction of people who contract a certain rare disease. Unfortunately, when working with well log dataset this fraction is not necessarily known as they depend on several factors (operating conditions during logging, type of formation etc.). This is an additional challenge in applying unsupervised outlier detection algorithm on well log data.

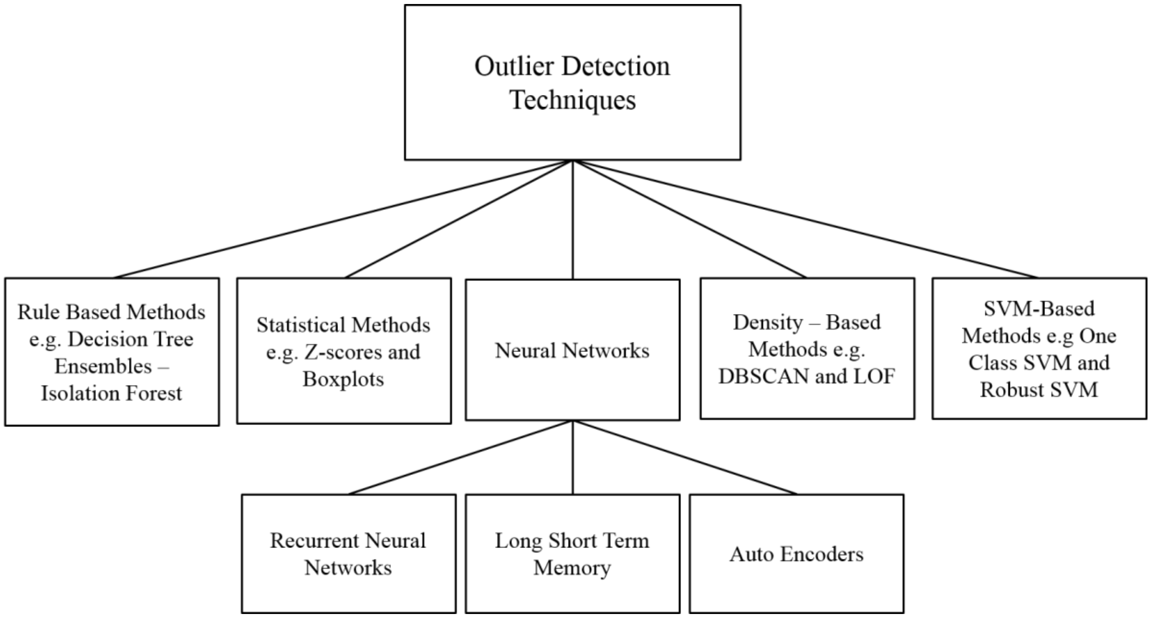
Outlier Detection Techniques
Rule Based Methods e.g. Decision Tree Ensembles – Isolation Forest
Statistical Methods e.g. Z-scores and Boxplots
Neural Networks
Density – Based Methods e.g. DBSCAN and LOF
SVM-Based Methods e.g One Class SVM and Robust SVM
Recurrent Neural Networks
Long Short Term Memory
Auto Encoders

Figure III-1: Popular methods used for Outlier detection

Machine Learning Based Outlier Detection Algorithms

In this chapter, the performance of four unsupervised outlier detection algorithms were compared on their ability to identify in well logs the formation depths that exhibit anomalous or outlier log responses. The algorithm was used in an unsupervised manner without minimal tuning. Each formation depth is considered a sample and the different well logs are considered features. An unsupervised outlier detection algorithm processes the feature matrix corresponding to the available samples that contain both normal and anomalous behavior to detect depths that exhibit outlier behavior. Unsupervised outlier detection algorithms detect anomalous behavior either based on distance, density, decision boundary, or affinity that are used to quantify the relationships governing the inlier and outlier samples. In the next section four unsupervised algorithms will be introduced namely isolation forest (IF), one-class SVM (OCSVM), local outlier factor (LOF), and density-based spatial clustering of applications with noise (DBSCAN).

Isolation Forest

Isolation forest (IF) assumes that the outliers will likely lie in sparse regions of the feature space and have more empty space around them than the densely clustered normal/inlier data [19]. Since outliers are in less populated regions of the dataset, it generally takes fewer random partitions to isolate them in a segment/partition, meaning they are more susceptible to isolation [20]. Isolation Forest generates tree-like structures, where the number of partitioning required to isolate an outlier sample in a terminating node is equivalent to the path length from the root node to the terminating node. This path length is averaged over a “forest” of such random trees and is a measure of the normality of a sample (to what degree a sample is an outlier or not), such that anomalies have noticeably shorter path lengths i.e. they are easier to partition and isolate in the feature space. A decision function labels each observation as an inlier or outlier based on the path length of the observation compared to the average path length of all observations. IF requires minimal hyperparameter tuning to obtain reasonable reuslts, has low computation requirements, fast to deploy, and can be parallelized for faster computation. The major hyperparameters for tuning are the amount of contamination of the dataset, number of trees/estimators, maximum number of samples to be used in each tree, and maximum number of sub-sampled features used in each tree. Figure III-2(a) illustrates the outlier detection by the Isolation Forest when applied to a simple two-dimensional dataset containing 110 samples. The green samples represent inliers and the orange samples represent outliers, and the shade of blue in the background is indicative of degree of normality of samples lying the shaded region, where darker blue shades correspond to outliers that are easy to partition or in this case isolate.

One Class Support Vector Machine

One-class support vector machine (OCSVM) is a parametric unsupervised outlier detection algorithm suitable when the data distribution is gaussian or near gaussian with very few cases of the anomalies. OCSVM is based on the support vector machine that finds the support vectors and then separate the data into separate classes using hyperplanes. OCSVM finds a minimal hypersphere in the kernel space (transformed feature space) that circumscribes maximum inliers (normal samples); thereby inferring the normality in the dataset. OCSVM nonlinearly projects the data into a high-dimensional kernel space, and then maximally separates the data from the origin of the kernel space. As a result, OCSVM may be viewed as a regular SVM where all the training data lies in the first class, and the origin is taken as the only member of the second class. Nonetheless, there is a trade-off between maximizing the distance of the hyperplane from the origin and the number of training data points contained in the hypersphere (region separated from the origin by the hyperplane). An optimization routine is used to process the available data to select certain samples as support vectors that parameterize the decision boundary to be used for outlier detection [21]. OCSVM implementation is challenging for high-dimensional data, tends to overfit, and needs careful tuning of the hyperparameters. OCSVM is best suited for outlier detection when the training set is minimally contaminated by outliers without any assumptions on the distribution of the inlying data. Important hyperparameters of OCSVM are the gamma and outlier fraction. The gamma influences the radius of the gaussian hypersphere that separates the inliers from outliers, large values of gamma will result in smaller hypersphere and ‘stricter’ model and vice versa. It acts as the cut-off parameter for the Gaussian hypersphere that governs the separating boundary between inliers and outliers [22]. Outlier fraction defines the percentage of the dataset that is outlier. Outlier fraction helps in creating tighter decision boundary to improve outlier detection. Figure III-2(a), Figure III-2(b) illustrates the working of the one-class SVM where the interfaces of two different shades are few possible decision functions that can be used for outlier detection. Figure III-2(b) illustrates the outlier detection by the OCSVM when applied to a simple two-dimensional dataset containing 110 samples. The purple samples are outliers, and the different contour shades in the background is indicative of degree of normality of samples lying in the shaded region, with inner most contours pointing toward more “normal” samples.

Density Based Clustering Algorithm with Noise Application

DBSCAN is a density-based clustering algorithm which can be used as an unsupervised outlier detection algorithm. The density of a region depends on the number of samples in that region and the proximity of the samples to each other. DBSCAN seeks to find regions of high density in the dataset and define them as inlier clusters. Samples in less dense regions are labelled as outliers. The key idea is that for each sample in the inlier cluster, the neighborhood region of the certain user-defined size (referred to as bandwidth) must contain at least a minimum number of samples, that is, the density in the neighborhood must exceed a user defined threshold [23]. DBSCAN requires the tuning of the following hyperparameters that control the outlier detection process: minimum number of samples required to form a cluster, maximum distance between any two samples in a cluster, and parameter p defining the form of the Minkowski Distance, such that Minkowski distance transforms into Euclidean distance for $p = 2$. The DBSCAN model is particularly effective at detecting point outliers and can also detect collective outliers if they occur at low density regions. However, it is not reliable at detecting contextual outliers. DBSCAN requires significant expertise level in select hyperparameters for optimal performance in terms of outlier detection. Figure III-2

(d) shows the working of a DBSCAN model on a two dimensional dataset with 110 samples with 10 points representing outlier (purple coded) and 100 points represents normal data (green coded).

Local Outlier Factor

Local outlier factor (LOF) is an unsupervised outlier detection algorithm that does not learn a decision function. Simple density-based outlier detection algorithms are not as reliable for outlier detection when the clusters are of varying densities. The Local Outlier Factor mitigates the problem by using relative density and assigns a score to each sample based on its relative density. LOF compares the local density of a sample to the local densities of its K-nearest neighbors to identify regions of similar density and to identify outliers, which have a substantially lower density than their K-nearest neighbors. LOF assigns a score to each sample by computing relative density of each sample as a ratio of the average local reachability density of neighbors to the local reachability density of the sample, and flags the points with low scores as outliers [24]. A sample with LOF score of 3 means the average density of this point's neighbors is about 3 times more than its local density, i.e. the sample is not like its neighbors. LOF score of a sample smaller than one indicates the sample has higher density than neighbors. The number of neighbors (K) sets how many neighbors are considered when computing the LOF score for a sample. In Figure III-2 (c), the LOF is applied to the above-mentioned 2-dimensional dataset containing 110 samples. The radius of the circle encompassing a sample is directly proportional to the LOF score of the sample and by that, the points at the middle of the plot are the outliers in this dataset and the sample points at the ends with the smaller radius circles around the sample points are the normal sample points. A standard value of K=20 is generally used for outlier detection [25]. Like DBSCAN for unsupervised outlier detection, LOF is severely affected by the curse of dimensionality and is computationally intensive when there are large number of samples. LOF needs attentive tuning of the hyperparamters. Due to the local approach, LOF is able to identify outliers in a data set that would not be outliers in another area of the data set. The major hyper parameters for tuning are the number of neighbors to consider for each sample and metric p for measuring the distance, similar to DBSCAN, where the general form of Minkowski distance transforms into Euclidean distance for p=2.

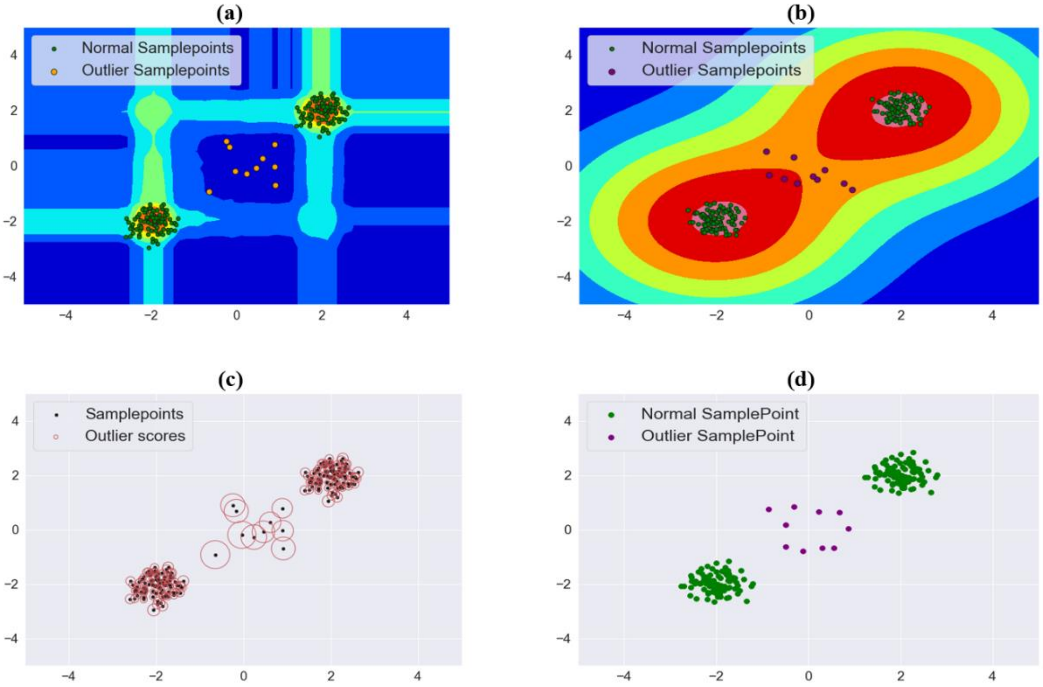

Figure III-2: Application of the proposed outlier detection models with the blue points indicating inliers and red points indicating outliers (a) Isolation Forest – different shades of blue on map signify the decision boundaries with dark values signifying an increasing likelihood of a sample being an outlier (b) One class SVM -- – different shades of blue on map signify the decision boundaries with dark values signifying an increasing likelihood of a sample being an outlier (c) Local Outlier Factor – the radius of the red circle signifies the likelihood of a sample being an outlier (d) DBSCAN

Methodology

Data for this comparative study was obtained from one well, the logs selected for the analysis are gamma ray (GR), density (RHOB), neutron porosity (NPHI), compressional velocity (DTC), deep and shallow resistivity (RT and RXO). 5617 samples are available from a depth interval; of 580 – 5186 ft. The dataset contains log responses from limestone, sandstone, dolostone and shale bed.

Data Preprocessing

Data preprocessing refers to the transformations applied to data before feeding it to the machine learning algorithm. Primary use of data preprocessing is to convert the raw data into a clean dataset that the machine learning workflow can process. A few data preprocessing tasks include fixing null/nan values, imputing missing values, scaling the features, normalizing samples, removing outliers, encoding the qualitative/nominal categorical features, and data reformatting. Data preprocessing is an important step because a data-driven model built using machine learning is as good as the quality of data processed by the model.

Feature Transformation

Machine learning models tend to be more efficient when the features/attributes are not skewed and have relatively similar distribution and variance. Unfortunately feature vectors can come in many different distributions and are not always normal/gaussian. However certain techniques can be used to transformed to this non gaussian distributions to a gaussian/near-gaussian distribution. The transformed feature is a function of the initial feature, some simple functions used for transformations are logarithm and power (square (2), inverse (-1) and any reasonable real number) some more complex transformation involve more complex functions like the box-cox transformation, quantile transformation. Resistivity measurements range from 10^{-2} ohm-m (brine filled formation) to 10^{3} ohm-m (low porosity formation) and tend to exhibit log-normal distribution. To reduce the right skewness observed in the resistivity data (i.e. mean >> mode), the resistivity measurements is transformed using the logarithmic function (logarithmic transformation). This reduces it skewness and variability and improves the model's predictive performance, as demonstrated in subsequent sections.

Feature Scaling

A dataset generally contains features that significantly vary in magnitudes, units and range. This tends to bias the machine learning methods based on distance, volume, density, and gradients. Without feature scaling, a few features will dominate during the model development. For instance, the features with high magnitudes will weigh in a lot more in the distance calculations than features with low magnitudes, which for example will adversely affect k-nearest neighbor classification/regression and principal component analysis. Feature scaling is an important aspect of data preprocessing that improves the performance of the data-driven models. For methods based on distance, volume and density, it is essential to ensure that the features have similar or near similar scales for improved performance. Data from different logs usually range between different scales.

For example, the RHOB (1.95 – 2.95g/cc) and GR (50 – 250 gAPI) log have vastly different scales.

For purposes of feature scaling, robust scaling method is used, which can be expressed mathematically as:

$$\mathbf{x_{is}} = \frac{\mathbf{x_i - Q_1(x)}}{\mathbf{Q_3(x) - Q_1(x)}} \qquad \textbf{Equation III-2}$$

where x_{is}: scaled feature x for the *i-th* sample; x_i: unscaled feature x for the *i-th* sample; $Q_1(x)$: first quartile of feature x; and $Q_3(x)$: third quartile of feature x. The first and third quartiles represent the median of the lower half and upper half of the data, respectively, which is not influenced by outliers. Robust scaling is performed on the features (logs) because it overcomes the limitations of other scaling methods, like the Standard scaler that assumes the data is normally distributed and the MinMax scaler that assumes that the feature cannot exceed certain values due to physical constraints. Presence of outliers adversely affects the Standard scaler and severely affects the MinMax scaler. Robust scaler overcomes the limitations of the MinMax scaler and Standard scaler by using the first and third quartiles for scaling the features instead of the minimum, mean and maximum values. The use of quartiles ensures that the robust scaler is not sensitive to outliers, whereas the minimum and maximum values used in the MinMax scaler could be the outliers and the mean and standard deviation values used in the Standard scaler is influenced by outliers.

Validation Dataset

3 validation datasets were created containing known organic/synthetic outliers to assess and compare the performances of the mentioned unsupervised outlier detection algorithms. Being unsupervised methods, there is no direct way of quantifying the performances of isolation forest, local outlier factor, DBSCAN, and one-class SVM. Therefore, domain knowledge, physically consistent thresholds, and various synthetic data creation methods is used to assign outlier/inlier label to each sample in the dataset to be processed by the unsupervised outlier detection model.

Dataset #1: Noisy Measurement

Dataset #1 was constructed from the above-mentioned dataset to compare the performance of the four unsupervised outlier detection algorithms in identifying depths where log responses are adversely affected by noise. Noise in well log dataset can adversely affect its geological interpretation as it masks the formation property at those depths. The dataset was acquired in the aforementioned well drilled with a bit of size 7.875" and is comprised of log responses measured at 5617 recorded depths points. Dataset #1 comprise gamma ray (GR), bulk density (RHOB) and compressional velocity (DTC) logs from the dataset for the depths where the borehole diameter is between 7.8" and 8.2". This led to 4037 inliers in Dataset #1. A synthetic noisy log response (200 samples) is then created based on the distribution shape and range of each of the feature vectors, such that each sample represent a valid log response but has no physical relationship to the original dataset.

Following that, synthetic noisy log responses for 200 additional depths were randomly introduced/ “scattered” into the feature matrix to create Dataset #1. Consequently, Dataset #1 contains in total 4237 samples, out of which 200 are outliers. Figure III-3(a) is a 3D scatterplot of Dataset #1, such that the green points are labelled as inlier which represent the recorded well log data from each feature vector (RHOB, GR and DTC) and the purple points are labelled as outliers which represent the synthetic noisy dataset.

Dataset #2: Bad Hole Measurement

Dataset #2 was constructed from the dataset to compare the performance of the four unsupervised outlier detection techniques in detecting depths where the log responses are adversely affected by the large borehole sizes, also referred as bad holes. Like Dataset #1, Dataset #2 comprise GR, RHOB, DTC, deep resistivity (RT), and neutron porosity (NPHI) logs from the dataset for depths where the borehole diameter is between 7.8” and 8.2”. Following that, the depths in the dataset where borehole diameter is greater than 12” were added to Dataset #2 as outliers. Consequently, Dataset #2 contains in total 4128 samples, out of which 91 are outliers and 4037 are inliers. Inliers in Dataset #2 are the same as those in Dataset #1. Comparative study on Dataset #2 involved experiments with four distinct feature subsets sampled from the available features GR, RHOB, DTC, RT, and NPHI logs. The four feature subsets are referred as FS1, FS2, FS3 and FS4, where Feature Set 1 (FS1) contains GR, RHOB and DTC, Feature Set 2 (FS2) contains GR, RHOB and RT, Feature Set 3 (FS3) contains GR, RHOB, DTC and RT, and Feature Set 4 (FS4) contains GR, RHOB, DTC and NPHI. The four feature subsets were used to analyze the effects of features on the performances of the four unsupervised outlier detection algorithms. Figure III-3(b) is a 3D scatterplot of Dataset #2 for the subset FS1, where green points are the inliers which represent well log data measured in the borehole with diameter between 7.8" – 8.2" (gauge/near gauge) and the purple points are the outliers which represent well log data measured in the borehole with diameter greater than 12" (washout). The outlier points should represent points where the well logs reading will be negatively affected by the effect of the larger hole (e.g. limited tool depth of investigation).

Dataset #3: Shaly Layers with Noisy and Bad Hole Measurements

Dataset #3 was constructed from the onshore dataset to compare the performance of the four unsupervised outlier detection techniques in detecting thin shale layers/beds in the presence of noisy and bad-hole depths. Dataset #3 comprise GR, RHOB, DTC, RT, and NPHI responses from 201 depth points from a sandstone bed, 201 depth points from a limestone bed, 201 depth points from a dolostone bed and 101 depth points from a shale bed of the Onshore Dataset. These 704 depths constitute the inliers. 30 bad-hole depths with borehole diameter greater than 12" and 40 synthetic noisy log responses are the outliers that are combined with the 704 inliers to form Dataset #3. Consequently, Dataset #3 contains in total 774 samples, out of which 70 are outliers. Comparative study on Dataset #3 involved experiments with four distinct feature subsets sampled from the available features GR, RHOB, DTC, RT, and NPHI logs, namely FS1, FS2, FS3 and FS4, like that performed on Dataset #2. Figure III-3(c) is a 3D scatterplot of Dataset #3 for the subset FS1, where green points are the inliers where each point represent well log reading from either sandstone, limestone, dolostone or shale bed and the purple points are outliers where each point either represents a synthetic noisy log data or well log reading from a sample depth with diameter greater than 12".

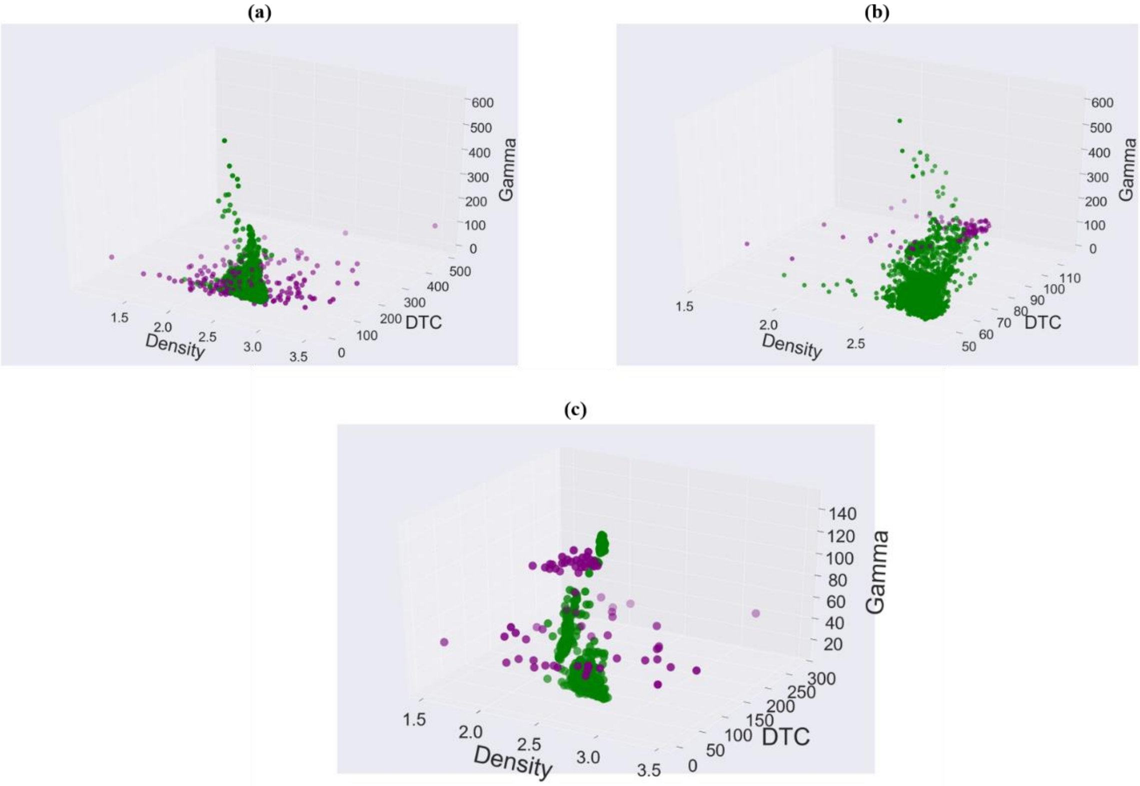

Figure III-3: Scatter plot highlighting the distribution of all created datasets: (a) Dataset #1, (b) Dataset #2, (c) Dataset #3

Metrics for Algorithm Evaluation

The selected unsupervised outlier detection algorithms will process the four above-mentioned datasets and will assign a label (either outlier or inlier) to each depth (sample) in the dataset. Labels are assigned based on the log responses for each depth. In real world application of unsupervised outlier detection, there is no prior information of outliers and outlier labels are present. For purposes of comparative study of the performances of the unsupervised outlier detection algorithms four datasets were created, named Datasets #1, #2, #3, and #4 containing outlier and inlier labels. In evaluating this algorithms metrics/scores employed to evaluate the classification methods will be used. In evaluating a binary classification model each prediction by the model are classified as either true positive, true negative, false positive or false negative. In comparing this algorithm, the outlier detection problem will be treated as binary classification problem (i.e. only two classes can be predicted: inlier or outlier) and therefore this tag can be used. The true positive/negative refer to the number of outlier/inlier samples that are correctly detected as the outlier/inlier by the outlier detection model. On those lines, false positive/negative refer to the number of outlier/inlier samples that are incorrectly detected as the inlier/outlier by the unsupervised outlier detection model. For example, when an outlier is detected as an inlier by the model, it is referred to as a false negative.

The following classification evaluation metrics will be used in this chapter to compare the performance of each algorithms on the datasets, similar metrics were employed by Wu et al, 2019 [52]:

Recall

Recall (also referred to as sensitivity) is the ratio of true positives to the sum of true positives and false negatives. It represents the fraction of outliers in dataset correctly detected as outliers. It is expressed as:

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \qquad \textbf{Equation III-3}$$

It is an important metric but should not be used in isolation as a high recall does not necessarily mean a good outlier detection because of the possibility of large false positives, i.e. actual inliers being detected as outliers. For example, when an outlier detection model detects each data point as an outlier, the recall will be 100% but it is a bad performance.

Specificity

Specificity is the ratio of true negatives to the sum of true negatives and false positives. It represents the fraction of correctly detected inliers by the unsupervised outlier detection model. It is expressed as:

$$\mathbf{Specificity} = \frac{\mathbf{TN}}{\mathbf{TN+FP}} \qquad \textbf{Equation III-4}$$

It is an important metric in this work as it ensures that inliers are not wrongly labeled as outliers. It is used together with recall to evaluate the performance of a model. Ideally, high recall and high specificity is required. A high specificity on its own does not indicate a good performance. For example, if a model detects every data point as an inlier. The specificity will be 100%.

Balanced Accuracy Score

The balanced accuracy score is the arithmetic mean of the specificity and recall, it overcomes the limitation of the recall and specificity by combining both metrics and providing a single metric for evaluating the outlier detection model. It is expressed mathematically as:

$$\mathbf{Balanced\ Accuracy\ Score} = \frac{\mathbf{Recall+Specificity}}{\mathbf{2}} \qquad \textbf{Equation III-5}$$

Its values range from 0 to 1, such that 1 indicates a perfect performing outlier detection model that correctly detects all the inliers and outliers in the dataset. Balanced accuracy score of less than 0.5 indicates that randomly assigned labels will perform better than outlier detection model for identifying either the outlier or the inlier.

Precision

Precision is a measure of the reliability of outlier label assigned by the unsupervised outlier detection algorithm. It represents the fraction of correctly predicted outlier points among all the predicted outliers. It is expressed mathematically as:

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}} \qquad \textbf{Equation III-6}$$

Similar to recall, precision should not be used in isolation to assess the performance. For instance, if a dataset has 1000 outliers and a model detects only one point as an outlier and it happens to be a true outlier, then the precision of the model will be 100%.

F-1 Score

The F-1 score is the harmonic mean of the recall and precision, like the balanced accuracy score it combines both metrics to overcome their singular limitations. It is expressed mathematically as:

$$\mathbf{F_1\ Score} = \frac{\mathbf{2\ \times Precison \times Recall}}{\mathbf{Precision+Recall}} \qquad \textbf{Equation III-7}$$

The values range from 0 – 1, such that F1 score of 1 indicates a perfect prediction and 0 a total failure of the model. If the earlier discussed case is, where the dataset contains 1000 inliers and 100 outliers, and the outlier detection algorithm detects only one outlier which happens to be a “true” outlier, the precision is 1, the recall is 0.01 and the specificity is 1. The balance accuracy score is 0.5. However, the F1-score is around 0.02. F1-score and balanced accuracy score helps to detect a poorly performing outlier detection model.

ROC AUC Score

Each unsupervised outlier detection algorithm implements a specific threshold to determine whether a sample is outlier. ROC curve is a plot of the true positive rate (recall) vs the false positive rate (1 - specificity) at different decision/probability thresholds. When the threshold of an unsupervised outlier detection algorithm is altered, the performance of the unsupervised outlier detection algorithm changes resulting in the ROC curve. For instance, the isolation forest computes the average path length of samples, such that samples with shorter path length are considered more likely to be outliers and are given a higher anomaly score. A threshold is set for the isolation forest by defining the anomaly score beyond which a sample will be considered an outlier. For the isolation forest, the anomaly scores typically range from -1 to 1 with the threshold set at 0 by default, such that negative values (<0) are labelled outliers and positive value (>0) are labelled inliers. For good outlier detection, an unsupervised outlier detection algorithm should have high recall (high TPR) and high specificity (low FPR), meaning the ROC curve should shift towards the top left corner of the plot. As the ROC curve shifts to the left top corner, the area under curve (AUC) tends to 1, which represents a perfect outlier detection for various choices of threshold. An unsupervised outlier detection algorithm is reliable when the recall and specificity are close to 1 and independent of the choice of thresholds, which indicates an AUC of 1. A ROC curve exhibiting a gradient close to 1 and AUC of 0.5 indicates that the unsupervised outlier detection algorithm is performing only as good as randomly selecting certain samples as outliers.

Discussion of Results

Dataset #1

Dataset #1 as earlier explained contains measured 4037 GR, RHOB, DTC and RT responses combined with 200 synthetic noise samples having a total of 4237 sample points. The unsupervised outlier detection model performance is evaluated for three feature subsets referred to as FS1, FS2, and FS2$*$, where FS1 contains GR, RHOB, and DTC; FS2 contains GR, RHOB, and logarithm of RT; and FS2$*$ contains GR, RHOB, and

RT. For the subsets FS1 and FS2∗ of Dataset #1, DBSCAN performs better than the other models, as indicated by the balanced accuracy score. For the subset FS1 of Dataset #1, the DBSCAN correctly labels 176 of the 200 introduced noise samples as outliers and 3962 of the 4037 “normal” data points as inliers; consequently, DBSCAN has a balanced accuracy score and F1 score of 0.93 and 0.78, respectively. For the subset FS2 of Dataset #1, log transform of resistivity negatively impacts the outlier detection performance. Logarithmic transformation of resistivity reduces the variability in the feature. On using deep resistivity (RT) as is (i.e., without logarithmic transformation) in the subset FS2∗, DBSCAN generates similar performance as with the subset FS1 (Table 3.1). All models except isolation forest (IF) are adversely affected by the logarithmic transformation of RT. Visual representation of the performances in terms of balanced accuracy score is shown in Figure III-4(a).

LOF model does not perform well in detecting noise in a well-log dataset. Based on the ROC-AUC score, LOF performs the worst compared with OCSVM and IF in terms of the sensitivity of the accuracies (precisions) of both inlier and outlier detections to the decision thresholds. Based on F1 score, DBSCAN has the highest reliability and accuracy (precision) in outlier detection; however, hyperparameter tuning should be done to improve the precision of DBSCAN because the current F1 score is not close to 1. One reason for low F1 score is that the inlier-outlier imbalance was not addressed. All these evaluation metrics used in this study are simple metrics that can be improved by weighting the metrics to address the effects of imbalance (i.e., the number of positives is one order of magnitude smaller than the number of negatives). F1 score of all the methods can be improved by improving the precision.

Table 1: Results from Dataset #1

		BALANCED ACCURACY SCORE	F1 SCORE
ISOLATION FOREST	FS1	0.84	0.55
	FS2	0.85	0.37
	FS2*	0.88	0.63
ONE CLASS SVM	FS1	0.91	0.57
	FS2	0.81	0.45
	FS2*	0.92	0.59
LOCAL OUTLIER FACTOR	FS1	0.73	0.28
	FS2	0.62	0.18
	FS2*	0.68	0.24
DBSCAN	FS1	0.93	0.78
	FS2	0.66	0.42
	FS2*	0.93	0.76

Dataset #2

Dataset #2 as earlier explained contains 4037 measured GR, RHOB, DT, RXO and RT normal responses combined with 91 samples from depth affected by significant washout. In Dataset #2, model performance is evaluated for five feature subsets: FS1, FS2, FS2$**$, FS3, and FS4. FS1 contains GR, RHOB, and DTC; FS2 contains GR, RHOB, and RT; FS2$**$ contains GR, RHOB, and RXO; FS3 contains GR, RHOB, DTC, and RT; and FS4 contains GR, RHOB, DTC, and NPHI. In each feature set, there are 91 depths (samples) labeled as outliers and 4037 depths labeled as inliers. Isolation forest (IF) performs better than other methods for all the feature sets. DBSCAN and LOF detections are the worst. IF performance for FS2 is worse compared with other feature subsets, because FS2 uses RT, which is a deep-sensing log and is not much affected by the bad holes. Consequently, when RT (deep resistivity) is replaced with RXO (shallow resistivity) in subset FS2$**$, the IF performance significantly improves indicating the need of shallow-sensing logs for better detection of depths where logs are adversely affected by bad holes. Subset FS3 is created by adding DTC (sonic) to FS2. FS3 has four features, such that DTC is extremely sensitive to the effects of bad holes, whereas RT is not sensitive. In doing so, the performance of IF on FS3 is comparable with that on FS1 and much better than that on FS2. This mandates the use of shallow-sensing logs as features for outlier detection. Visual representation of the performances in terms of balanced accuracy score is shown in Figure III-4(b).

Outlier detection performance on Dataset #2 clearly shows that when features that are not strongly affected by hole size (e.g., deep resistivity, RT) are used, the model performance drops, as observed in FS2. On the contrary, when shallow-sensing DTC and RXO are used as features, the model performance improves. I conclude that feature selection plays an important role in determining the performance of ODTs, especially in identifying "contextual outliers." IF model is best in detecting contextual outliers, like the group of log responses affected by bad holes. F1 scores are low because the fraction of actual outliers in the dataset is a small fraction (0.022) of the entire dataset, and contamination levels are not set a priori. Being an unsupervised approach, in the absence of constraints such as contamination level, the model is detecting many original inliers as outliers. Therefore, balanced accuracy score and ROC-AUC score are important evaluation metrics (Table 3.2).

Table 2: Results from Dataset #2

		BALANCED ACCURACY SCORE	F1 SCORE
ISOLATION FOREST	FS1	0.93	0.23
	FS2	0.64	0.11
	FS2*	0.86	0.21
	FS3	0.91	0.22
	FS4	0.93	0.24
ONE CLASS SVM	FS1	0.76	0.22
	FS2	0.6	0.11
	FS2**	0.65	0.14
	FS3	0.74	0.21
	FS4	0.84	0.28
LOCAL OUTLIER FACTOR	FS1	0.38	0.11
	FS2	0.57	0.07
	FS2**	0.56	0.08
	FS3	0.61	0.1
	FS4	0.61	0.09
DBSCAN	FS1	0.58	0.18
	FS2	0.53	0.09
	FS2**	0.56	0.17
	FS3	0.58	0.14
	FS4	0.61	0.18

Dataset #3

Dataset #3 consists of 774 samples with 704 sample points representing sandstone, limestone and dolostone beds and are labelled as inliers combined with 70 samples points of shale which are labelled as outliers in the dataset. Performance on Dataset #3 indicates how well a model detects depths where log responses are affected by either noise or bad hole in a heterogenous formation with thin layers of sparsely occurring rock type (i.e., shale). The objective of this evaluation is to test if the models can detect the noise and bad-hole influenced depths (samples) without picking the rare occurrence of shales as outliers. Outlier methods are designed to pick rare occurrences as outliers; however, a good shale zone even if it occurs rarely should not be labeled as outlier by the unsupervised methods.

Comparative study on Dataset #3 involved experiments with four distinct feature subsets sampled from the available features GR, RHOB, DTC, RT, and NPHI logs, namely, FS1, FS2, FS3, and FS4. FS1 contains GR, RHOB, and DTC; FS2 contains GR, RHOB, and RT; FS3 contains GR, RHOB, DTC, and RT; and FS4 contains GR, RHOB, DTC, and NPHI. In all feature sets, 70 points are known outliers, and 704 are known inliers, comprising sandstone, limestone, dolostone, and shales. Isolation forest (IF) model performs better than the rest for all feature sets. Interestingly, with respect to F1 score, IF underperforms on FS2 compared with the rest, due to lower precision and imbalance in dataset. This also suggests that DTC is important for detecting the bad-hole depths, because FS2 does not contain DTC, unlike the rest (Table 3.3). Visual representation of the performances in terms of balanced accuracy score is shown in Figure III-4 (c)

Table 3: Results from Dataset #3

		BALANCED ACCURACY SCORE	F1 SCORE
ISOLATION FOREST	FS1	0.91	0.81
	FS2	0.96	0.69
	FS3	0.92	0.84
	FS4	0.93	0.83
ONE CLASS SVM	FS1	0.78	0.57
	FS2	0.72	0.47
	FS3	0.8	0.61
	FS4	0.79	0.6
LOCAL OUTLIER FACTOR	FS1	0.8	0.61
	FS2	0.73	0.24
	FS3	0.61	0.34
	FS4	0.71	0.34
DBSCAN	FS1	0.75	0.95
	FS2	0.8	0.47
	FS3	0.66	0.73
	FS4	0.79	0.73

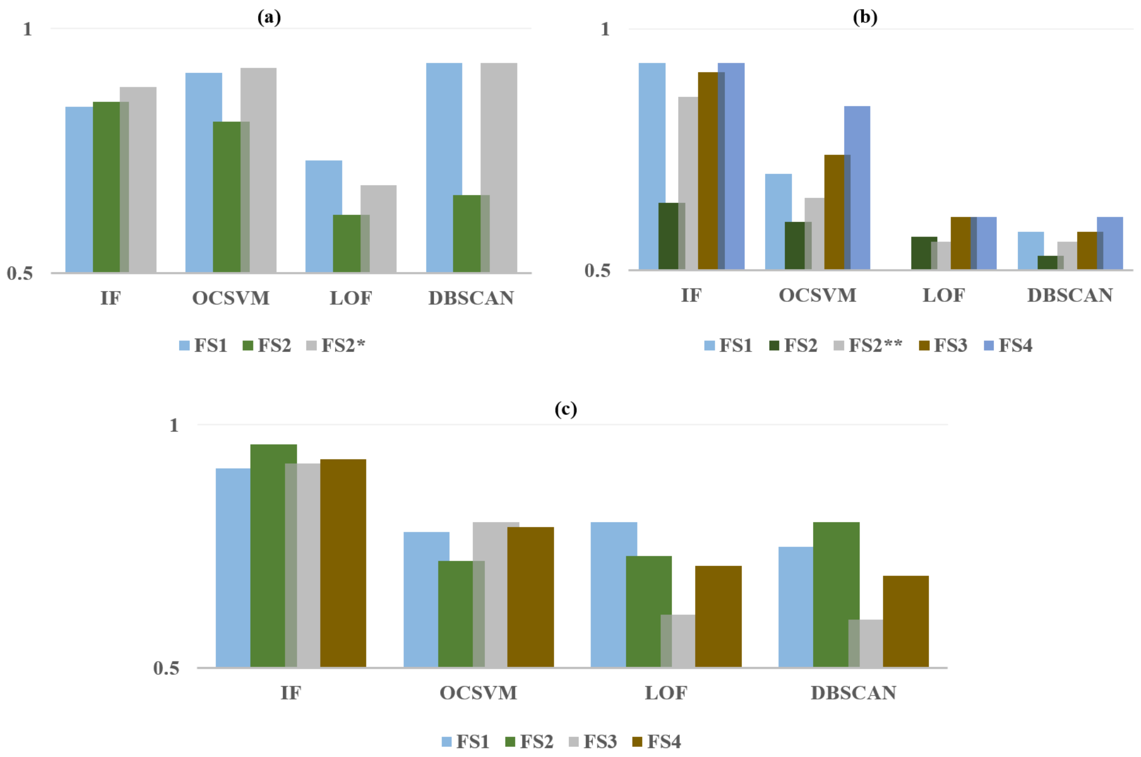

Figure III-4: Bar Plot showing the results (balanced accuracy score) in all cases: (a) Dataset #1, (b) Dataset #2 and (c) Dataset #3

Recommendation for Future Work

This study highlights the effectiveness of outlier detection algorithms, in particular the Isolation Forest in detecting outliers in a dataset. However, for future analysis I would recommend the following:

- Perform similar analysis on different types of subsurface data (not just well logs), e.g. seismic data, drilling data etc.
- Developing a method to tune the hyperparameters to optimize the unsupervised outlier detection algorithms

Conclusions

This chapter provides a comparative study of the performance of four outlier detection models: Isolation Forest (IF), One Class SVM (OCSVM), Local Outlier Factor (LOF) and DBSCAN. Using four different datasets I was able to compare the different models in several real-life scenarios and evaluate their performance. From the results I concluded that the DBSCAN models proves the most effective in detecting noise in log data compared to the other models used in this study. It is also showed that outlier detection algorithms can be used in detecting errors in log reading due to environmental conditions. In this study outlier points due to washouts/bad holes were considered. From the results it is surmised that the Isolation Forest is by far the most robust in detecting this type of outliers in log data and its performance will depend on correctly selecting features which best relate to the cause of the outlier. Isolation forest also proved efficient in detecting outliers in the presence of an infrequently occurring but relevant subgroup in a dataset. Overall, the Isolation Forest is recommended as the preferred algorithm in building robust outlier detection models in detecting outliers in a log dataset, although if the user only requires the removal of noisy data from the log, the DBSCAN proves to be a very powerful algorithm in building such models.

CHAPTER IV

MACHINE LEARNING WORKFLOW FOR PREDICTING NMR T1 DISTRIBUTION RESPONSE OF THE SUBSURFACE

It has been well established that single variable can be predicted using supervised machine learning algorithms, however prediction of multitarget signals present a different challenge. The target sample of signal contains multiple variables which can be dependent on one another. Some researchers [49, 50] have applied this technique with reasonable success. This chapter explores the application of supervised machine learning algorithms in predicting signal data using NMR T1 distribution as a case study. This chapter:

- provides a brief overview of nuclear magnetic resonance
- aims to predict the NMR T1 distribution using readily available well logs
- proposes an error metric to handle multiple variable target with high variability values
- proposes a novel index for measuring the reliability of each sample prediction.

Introduction

Nuclear magnetic resonance (NMR) measurements, including T1 and T2 distributions, are important geological downhole measurements that provide information about pore size, permeability, irreducible water volume, and oil viscosity [26, 27, 28], they are essential in the characterization and development of fields. NMR logging is popular for oil and gas reservoir characterization to assess the movable hydrocarbon volumes and in-situ reservoir permeability [26]. They have also been shown to improve interpretation in tight gas sands and unconventional reservoirs [29, 30] and also in quantifying fracture porosity [31]. However, NMR logging tools are expensive and can prove technically challenging to deploy in oil and gas wells [7]. In this chapter I will propose a machine-learning workflow to synthesize NMR T1 distribution along the length of a well where the deployment of NMR logging tool is not possible or would prove economically infeasible. The synthesis requires machine learning techniques to process easy-to-acquire conventional well logs, such as resistivity, neutron, density, sonic and spectral gamma ray logs, which are readily available in most wells. As NMR logs are not commonly run in most wells but provide important geophysical information about the subsurface geological formations, the ability to successfully predict T1 distribution along the entire length of the well with reasonable degree of accuracy can increase productivity and lower the risks associated with hydrocarbon exploration and production.

Overview of Nuclear Magnetic Resonance Logging

Measurements of NMR-logging tool are the response of the hydrogen nuclei in the formation [49]. NMR logging tool apply an external magnetic field to alter the magnetic moments of the hydrogen nuclei in the geological formation and measure the corresponding relaxation times (T1 and T2) as the nuclei come back to their equilibrium states. Relaxation is a measure of deterioration of NMR signal with time during the conversion of the excited nonequilibrium population to a normal population at thermal equilibrium [32]. T1 relaxation time (longitudinal relaxation time) is the time required for longitudinal magnetization to return to the z-axis at 63% of the original state, whereas T2 relaxation time is the time it takes for transverse magnetization to reach 37% of the initial value and is associated with the loss of spin coherence. T1 relaxation provides information about the inter and intra molecular dynamics. T2 relaxation involves energy transfer between interacting spins via dipole and exchange interactions [33]. NMR relaxations are controlled by three mechanisms [32]: bulk fluid relaxation, surface relaxation and molecular diffusion in magnetic field gradient. T1 and T2 relaxation times can be expressed as:

$$\frac{1}{T_1} = \frac{1}{T_{1b}} + \frac{1}{T_{1s}}$$

Equation IV-1

Where, the subscripts *b* and *s* represent the bulk relaxation and surface relaxation, respectively.

Bulk fluid relaxation is the intrinsic property of a fluid controlled by viscosity and chemical composition [32]. Surface relaxation occurs at the fluid-solid interface and is affected by both fluid and matrix compositions. Conventional log data provide information about the fluid and rock matrix composition in a reservoir formation. The goal of this chapter is to learn the relationship between conventional well logs and NMR T1 relaxation time by processing the available log dataset and quantify the uncertainty in the data-driven predictions/synthesis.

Machine Learning Algorithms

Ordinary Least Square

OLS model is one of the simplest statistical regression models that fits the dataset by reducing the sum of squared errors (SSE) between the predicted and actual data [7]. Each feature/variable is assigned a co-efficient β, whose value is optimized such that when linearly combined the sum of squared error between the actual value and the predicted value is minimized. The OLS model assumes the output is a linear combination of input values x_i and error ε_i [7]. It is expressed mathematically as:

$$\mathbf{y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i} \qquad \textbf{Equation IV-2}$$

Where, i represents a specific depth from the total depth samples available for model training, β_0 is the bias term or intercept and p represents the number of input logs available for the training.

$$\mathbf{SSE = \|y - X\beta\|^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad \textbf{Equation IV-3}$$

ElasticNet

ElasticNet algorithm is a regularized regression model. The OLS method is prone to overfitting and has been modified using regularization terms to form other algorithms. The ElasticNet is one of such algorithms. ElasticNet model is an OLS algorithm constrained by two regularization terms L_1 andL_2, these regularization terms force few of the coefficients β to go to zero and reduce the effect of highly correlated features; thereby reducing the tendency of the model to overfit. The ElasticNet aims to minimize the following loss function:

$$\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \ \boldsymbol{\alpha}_1 \sum_{\mathbf{j}=1}^{\mathbf{m}} \left\|\boldsymbol{\beta}_{\mathbf{j}}\right\| + \ \boldsymbol{\alpha}_2 \sum_{\mathbf{j}=1}^{\mathbf{m}} \left\|\boldsymbol{\beta}_{\mathbf{j}}\right\|^2 \qquad \textbf{Equation IV-4}$$

Where, β is the coefficient/weight is assigned to each feature, α_1 is the L_1-norm penalty parameter and α_2 is the L_2-norm penalty parameter.

Support Vector Machine

Support vector machine is based on the principle of structural risk minimization and has been applied in numerous fields for regression and classification purposes [34]. SVM constructs hyperplane(s) (decision boundaries that classify the data points) that separate data based on their similarity [34]. The objective of the support vector machine algorithm is to find a hyperplane in an *n*-dimensional space (n refers to the number of features) that distinctly classifies the data points and maximizes the margin between different groups. For regression, an error margin value is introduced, and hyperplanes are constructed to best approximate the continuous-valued function.

Artificial Neural Network

ANN is a popular machine learning method suitable for both linear and nonlinear regression problems. A typical Neural Network consists of an input layer, an output layer, and several hidden layers. The performance of the model can be altered by changing the number of hidden layers and/or the number of neurons in each hidden layer by backpropagation of error gradients. Each layer consists of neurons, made of parameters (weights and biases) to perform the matrix computation on outputs computed from the previous layer [7]. The weights and biases are updated until the objective function is optimized [35]. An activation function is used in each layer to incorporate non-linearity to the computations.

Random Forest

Random Forest is an ensemble learning algorithm that can be used for both regression and classification [36]. Random Forest is a collection of decision trees that is trained by the technique of bootstrapping and aggregation, referred to as bagging. Random Forest combines multiple decision trees in parallel into a single predictive model to achieve low bias and low variance. The final prediction in a Random Forest is made by averaging the predictions of all the decision trees in the ensemble. For classification, mode class from the trees is selected as the prediction, while for regression it selects the mean/median prediction of all trees. Random forest does not require feature scaling and requires little effort in tuning the hyperparameters while providing high predictive ability. In comparison to other methods, random forest exhibits lower variance and bias leading to higher generalization. Further, random forest uses bootstrapping that trains each decision tree in the forest on a random sub-sample of the dataset with replacement using a random sub-sample of the features. In this study, the random forest classifier is the most accurate, reliable and efficient method compared to the above-mentioned methods.

Quantile Regression Forest

Quantile regression forest is an extension of random forest that provides non-parametric estimates of the median predicted values as well as prediction quantiles obtained from each tree [37]. QRF provides a conditional distribution of the prediction of each tree. The full conditional distribution of the response variable represents a description of the uncertainty on the response variable given the predictor variable [21] as would been shown later in this work.

Methodology

This section discusses the methodology and workflow to be used in predicting the NMR T1 distribution. Figure IV-1 illustrates the workflow proposed in this chapter.

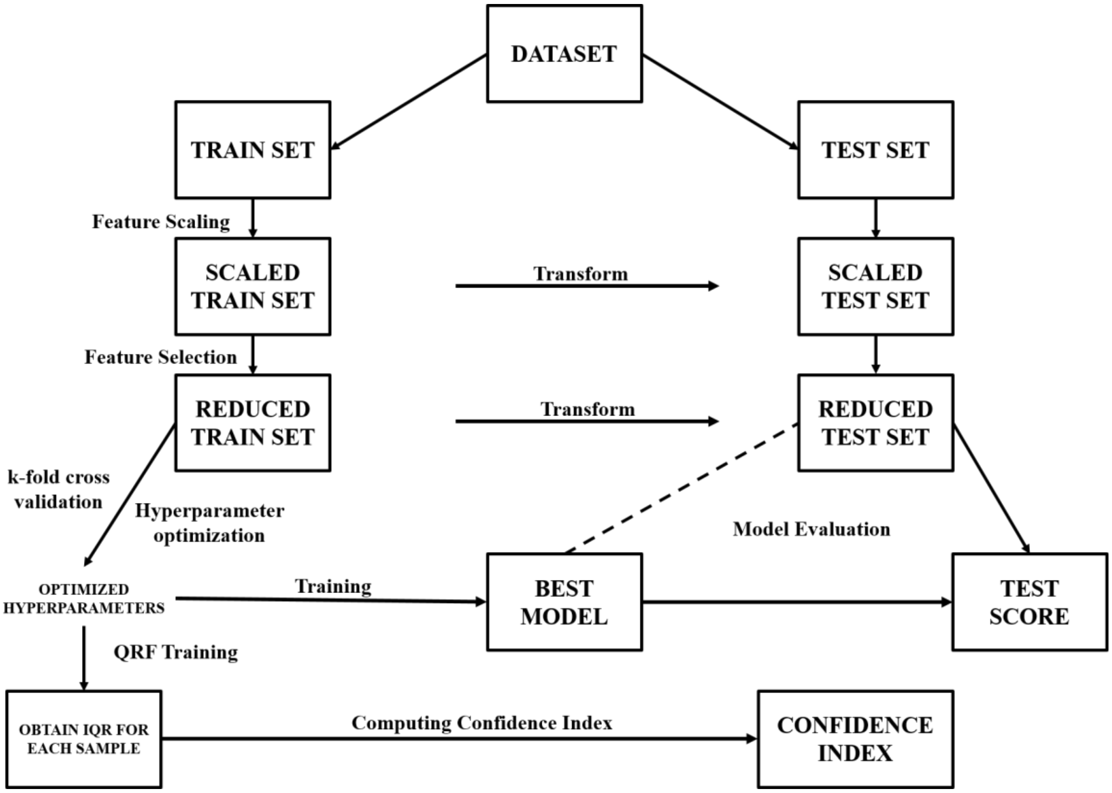

Figure IV-1: Workflow used for NMR T1 Synthesis

Description of Data

The well logs used for this work were obtained from a well drilled into the Arbuckle, Kinderhook shale and the Granite wash. The Arbuckle is a cyclical carbonate that is dominated by intertidal and shallow subtidal facies [38] it is mainly composed of dolomite, mudstone, interbedded shale and chert. The granite wash is a tight sand play composed of coarse detrital material formed from in-situ weathering, which occurred at different geological time. The well under investigation is an injection well that penetrates the Arbuckle deep aquifer and is drilled with a water-based mud. Well logs available for this work are from a 2911-feet long depth interval comprising 5617 data points, where each point corresponds a specific depth. Figure IV-2 depicts the dataset used in this study. The dataset is a split into a feature set and a target set. The feature set consists of 23 conventional logs (e.g. GR, CALI, PE, ACRTs, RHOB, NPHI etc.) shown in Tracks 1 to 5 of log plot in Figure IV-2. The target set consists NMR T1 distributions acquired by Halliburton magnetic resonance imaging (MRI) tool. The target set is shown in Track 6 of log plot shown in Figure IV-2. At each depth, NMR T1 distribution is represented as cumulative responses for 10 bins equally spaced in logarithmic scale between 4 ms to 2048 ms. In this work, the 10 discrete NMR T1 responses at each depth will be simultaneously predicted to synthesize the entire T1 distribution. In summary, the dataset comprises of 5617 samples (individual depth) and each sample can be represented as 23-dimensional feature vector and a corresponding 10-dimensional target vector. In other words, each sample has 23 features/attributes and 10 targets.

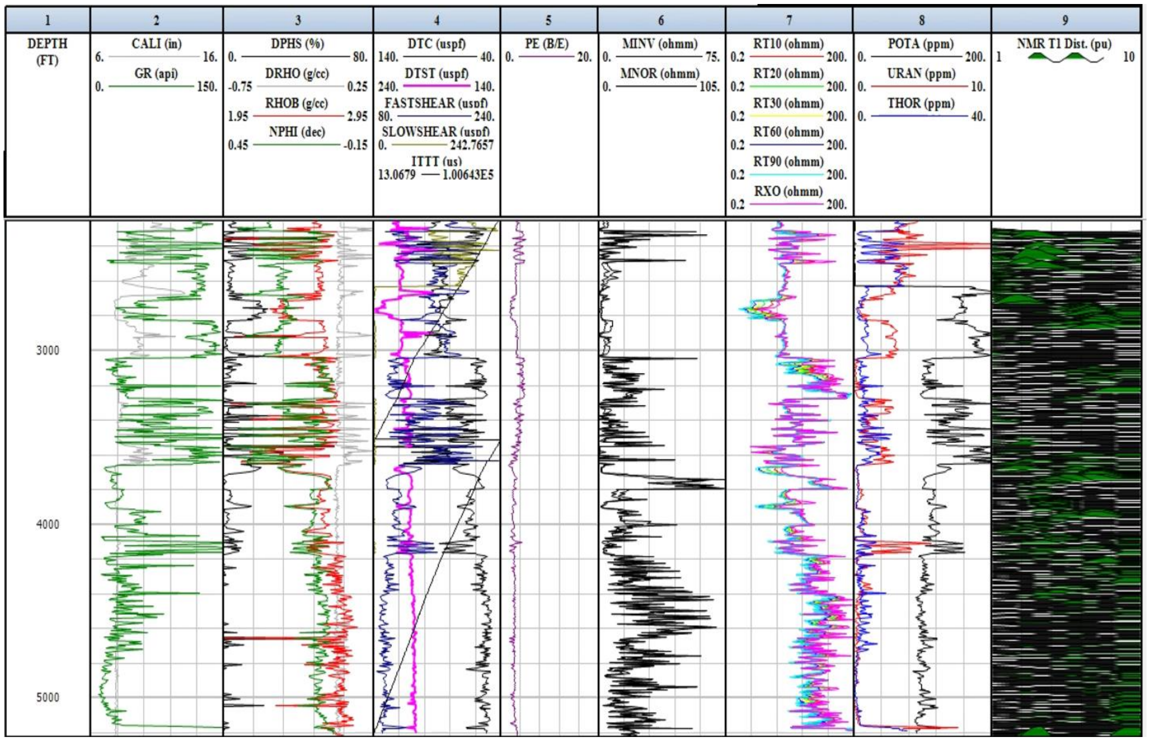

Figure IV-2: Well log representation of dataset: Depth (TRACK 1), Caliper and Gamma ray (TRACK 2), Density and Neutron (TRACK 3), Sonic measurements (TRACK 4), Photoelectric Log (TRACK 5), Micro-resistivity measurements (TRACK 6), Resistivity measurements (TRACK 7), Spectral Gamma Ray measurements (TRACK 8), NMR measurements (TRACK 9)

Data Pre-Processing

Data pre-processing is an essential step in developing a standard machine learning workflow. Data pre-processing broadly refers to all the steps taken in converting raw data (unprocessed data) to a dataset ready to be fit by a machine-learning algorithm. Data pre-processing tasks includes feature scaling, dimensionality reduction, feature selection, and feature transformation, to name a few. In this chapter, as shown in Figure IV-1, the pre-processing steps would involve data splitting, feature scaling and feature selection. Figure IV-3 describes in detail the data preprocessing pipeline adopted in our work. To ensure data quality, the caliper log is checked to ensure there is no significant borehole problems with hole diameter greater than 8.2” removed. (Bit size: 7.875”). Depth matching and depth shifting was performed to ensure that the feature vector and target vector correspond to the same sample/depth. 70% of the dataset (including the features and targets) is used as the training dataset and the remaining 30% is used to validate the results of model. Training dataset is processed by the considered model to learn the relationship between the feature and target vector. The validation dataset is used to evaluate the predicted target vector for a sample against the true target vector. In the subsequent sections, the feature selection workflow implemented in our study will be elaborated.

Figure IV-3 highlights the preprocessing workflow used in most machine learning projects. Data cleaning essentially prepares the dataset in a format that can be processed by the algorithm (i.e. numerical values, no missing values etc.). Data partitioning is performed to negate the effect of overfitting. Overfitting is characterized by a model fitting too closely to a specific dataset (training dataset) thereby reducing its accuracy when deployed on a new dataset (generalization). Partitioning the datasets provides a “test”/validation dataset that would be used to evaluate the applicability of the model when deployed. The “test” dataset is not used during any part of model training so as closely mirror a deployment scenario. Data partitioning involves randomly selecting a subset of the dataset for training the model and another for evaluating the model, this prevents a situation where the model “memorizes” the dataset and has no general applicability. The other data-preprocessing steps are feature scaling and feature selection. Feature scaling is discussed in **Section 3.4.1.2**

Note: Although Feature scaling is not required for Random Forest. Several models which require feature scaling such as OLS, ElasticNet, SVM and Neural Network are used in this project.

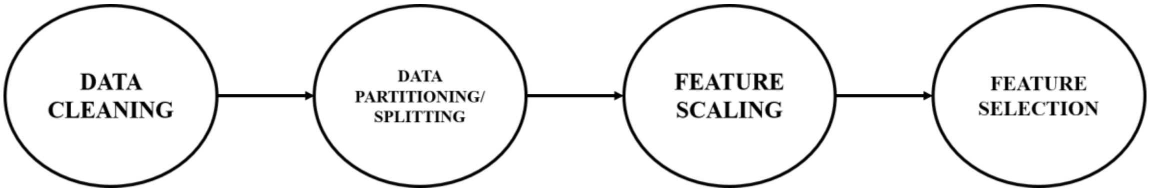

Figure IV-3: Pre-processing workflow for this project

Feature Selection

Dimensionality reduction reduces undesired characteristics in high dimensional data, namely, noise (variance), redundancy (highly correlated variables), and data inadequacy (features ≫ samples). Dimensionality reduction leads to some loss of information but have potential benefits such as: reducing storage requirements, aiding data visualization and comprehension, reduce model fitting [39]. High dimensional data leads to increased training time and increased risk of overfitting [40]. Dimensionality reduction methods can be broadly categorized into feature selection and feature extraction methods. Feature selection methods select the most relevant features from the original set of features based on an objective function. (Correlation criterion, information theoretic criterion, accuracy score etc.)

Features obtained using feature selection retain their original characteristics and meaning as in the original feature set, whereas those obtained using feature extraction are transformations of the original features that are different from the original feature set [41]. Feature selection methods are categorized as either filter or wrapper methods [42]. Filter methods ranks the features based on some mechanism (e.g. variance) and a threshold is set such that feature which do not meet the thresholds are removed. Examples of filter methods are dependence measure, mutual information, Markov blanket, and fast correlation-based filter [42]. Wrapper methods involves using different subsets of features and evaluating each subset through the results obtained by the model. These methods are essentially tied or “wrapped” to the model used in fitting the data [42]. An example of this is the Forward selection which start with an empty feature space; following that features are added one at a time for each step. For each step, the method selects the feature that most improves the model accuracy, features are added until there is negligible increase in model accuracy when adding new features. Two popular feature selection methods were combined in this work, namely mutual information and f-test.

Mutual Information

Mutual Information is a non-linear measure of the linear or nonlinear correlation between variables. Mutual information between two random variables measures the dependence between them, it is also referred to as the Information Theoretic Ranking Criteria (ITRC). It can be expressed mathematically as [42]:

$$\mathbf{MI(X;y) := \sum_X \sum_y p(x_j\,, y_i)\, log\frac{p(x_j,y_i)}{p(x_j)p(y_i)}} \qquad \textbf{Equation IV-5}$$

Where, target y_i is one element in the multitarget variable y, feature x_i is one element in the feature vector X, $p(x_j, y_i)$ is the joint probability density function of feature x_j and target y_i, and $p(x_j)$ and $p(y_i)$ are the marginal density functions. If X and y are independent, i.e. no information about y can be obtained from X, the mutual information is 0. For our study, *X* represents each of the 23 logs (features) while *y* is the 10-dimensional target vector.

F-Test

The F-test is a statistical tool used to compare the similarity of two models. F-Test performs a hypothesis test by creating two linear regression models X and Y. X is a model built using randomly selected constants as a feature vector and Y is the model built using a constant and a feature. X and Y are then used to predict the target. The sum of squared error (SSE) between the two models is recorded. If X and Y have similar results, the null hypothesis is accepted and it is assumed there is no relationship between that feature and the target. A key difference between the F-test and mutual information is the F-test estimate the degree of linear dependency between two random variables (*X* and *y*), while the mutual information methods can capture any kind of statistical dependency but require more samples for accurate estimation. In this study thresholds of 0.15 and 250 were set for the mutual information and f scores, respectively, for feature selection. Features that do not meet this threshold are removed to create a lower dimensional space. The selected features are further filtered by considering the correlation between the selected features, if two features have a correlation coefficient greater than 0.9, the feature with the lower mutual information score is removed. The mutual information score and F-score for each feature in the initial feature set is shown on a bar plot in Figure IV-4. The final features used for model training and predictions are: RHOB (Density log), DTC (Compressional wave velocity), DTST (Stoneley wave delay time), NPHI (Neutron log), RT10 (10inch resistivity reading), RT30 (30inch resistivity reading) and URAN (Spectral Gamma ray – Uranium). These selected features were used as features to predict the NMR T1 responses for the 10 bins equally spaced in logarithmic scale between 4 ms to 2048 ms.

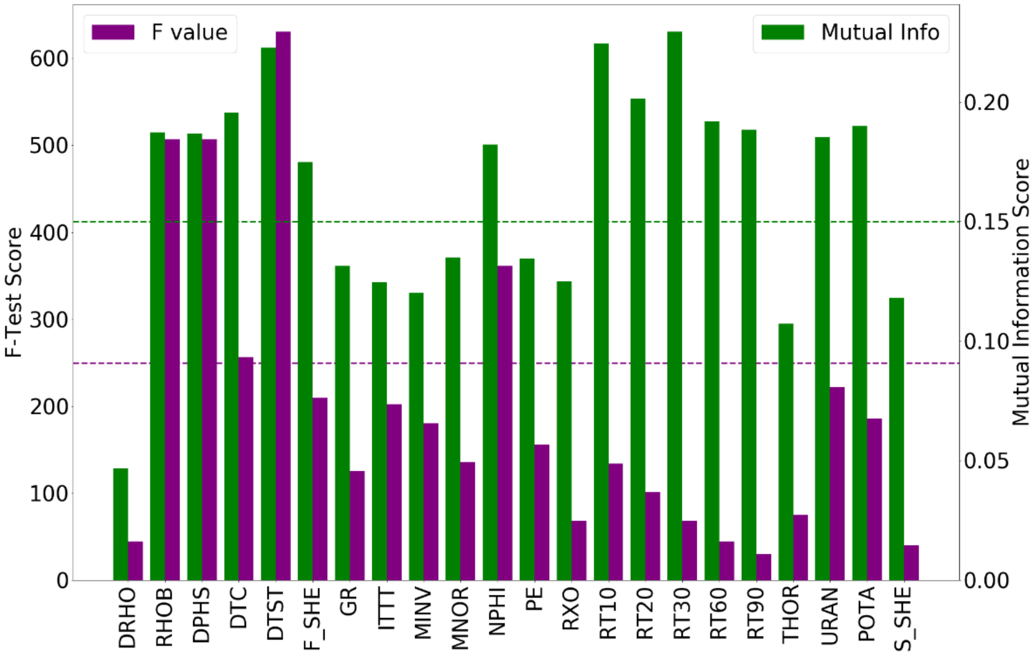

Figure IV-4: Bar plot showing the Mutual Information and F1 score for each feature used in Project

Evaluation Metric

For the hydrocarbon-bearing reservoir under investigation in this study, NMR T1 responses across the 10 bins lie between 0 - 21. The majority of bin values are either zero or near zero as shown in Figure IV-5. As a result of this, commonly regression evaluation metrics, such as root mean square (RMSE) and mean absolute error (MAE), would prove inadequate in evaluating the model as their results will be biased towards bins with smaller values. To illustrate this, suppose a model predicts a bin value as 10 when the actual value is 9.6 and the same model predicts a bin value as 0.004 when the actual value is 0.2; when using the RMS or MAE, the latter seems to be the better prediction with an error of 0.196 compared to the former which has an error of 0.4. This deceptive result would negative affect model evaluation and ranking. In handling this problem, a metric that provides a relative measure of error rather than an absolute one is employed. The mean absolute percentage error (MAPE) provides a relative measure of errors as the difference between the predicted and actual is scaled using the actual value. It is mathematically expressed as:

$$\mathbf{MAPE} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \left|\frac{\mathbf{y_i - \hat{y}_i}}{\mathbf{y_i}}\right| \qquad \textbf{Equation IV-6}$$

The limitation of MAPE for a sparse multitarget data is that the MAPE values will tend to infinity as most of the values in denominator are near zero. When the actual values are zeros or very close to zeros the MAPE is ineffective [43]. Tabataba et al. [44] suggests adding a small value to the denominator for normalization, for which the term corrected MAPE (cMAPE) is coined. In view of this, a slight modification is made to the formula by adding a 1 to the normalization parameters resulting in the following equation:

$$\mathbf{MMAPE} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \left|\frac{\mathbf{y_i - \hat{y}_i}}{\mathbf{y_i + 1}}\right| \qquad \textbf{Equation IV-7}$$

MMAPE (Modified Mean Absolute Percentage Error) overcomes the problem of MAPE tending to infinity. One drawback to this method however is that for bins with values close to zero it does not spot large relative error. For example, if a bin has a value of 0.01 and the model predicts 0.1, the MMAPE outputs an error 0.09 (9%). In practice this method performs well for model evaluation and should be applied in similar cases.

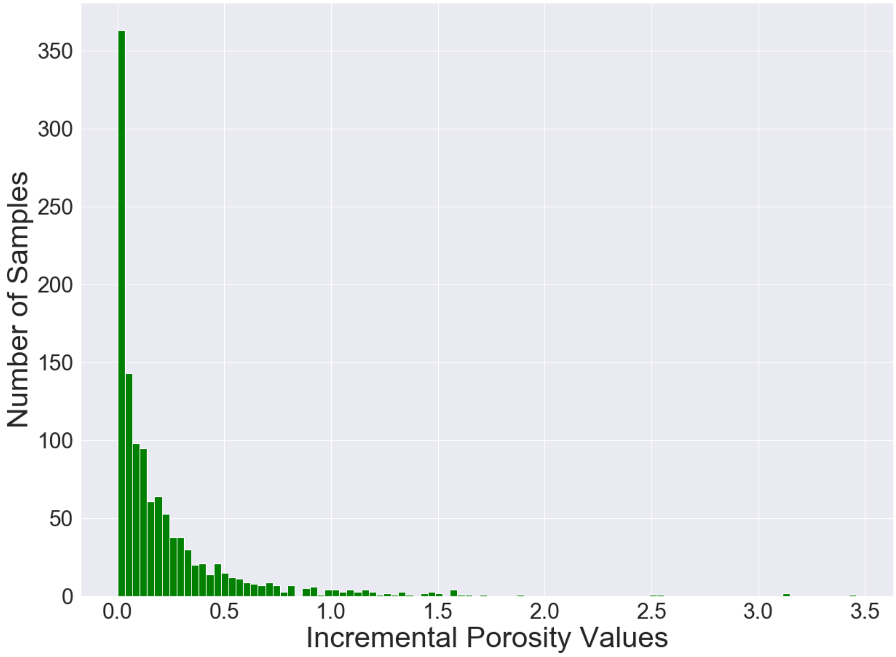

Figure IV-5: Distribution of incremental porosity values in Bin 2 (8ms), the distribution is skewed highly to the right, with many values zero or near zero, this poses significant problems in selecting an error metric for model evaluation

Figure IV-5 shows the distribution of incremental porosity for the Bin 1 of the measured T1 time. It highlights the problem stated in the section above, the values range from 0 to 7 with most of the values near 0 or 0. Using RMSE or MAE will create a model bias towards 0 (tends to underestimate) and using the basic MAPE is unadvisable as the error will tend to infinity and would create a something close to random model. For this reason, the model is evaluated using the MMAPE (Modified Mean Absolute Percentage Error) expressed in Equation 4.7.

"Confidence Index" Computation

Quantile Regression Forest (QRF) is used to infer the full conditional distribution (spread) of the response variable (target) for high-dimensional predictor variables (feature) [37]. A random regression forest comprises of multiple estimators (trees), each estimator/tree makes a prediction and the final prediction provided by the random forest is the average (mean/median) of the predictions of all trees. The quantile regression forest provides the quantile values of the predictions of all the trees, not just average as is provided by the basic Random Forest algorithm. This distribution provided by the quantile regression forest can be used to measure the uncertainty of each prediction. When each estimator provides varied predictions, the final prediction from the regression forest has high uncertainty, while in cases where each estimator provides similar prediction, the final prediction is considered to have less uncertainty. QRF is used to build prediction intervals that determine the certainty of the prediction for a specific sample [37]. For example, a 95% prediction interval for the value I is given by:

$$\mathbf{I_{(x)} = [Q_{0.025}(x), Q_{0.975}(x)]} \qquad \textbf{Equation IV-8}$$

where, $Q_{0.025}(x)$ represents the 2.5th percentile of the prediction distribution of x and $Q_{0.975}(x)$ represents the 97.5th percentile of the prediction distribution of x. The following mathematical formulation can determine the confidence index (CI) for NMR T1 synthesis at each depth/sample:

$$\mathbf{CI(y) = 1 - scale\left(\frac{Q_{0.75}(y) - Q_{0.25}(y)}{\hat{y}}\right)} \qquad \textbf{Equation IV-9}$$

$$\mathbf{scale(x) = \frac{x_i - x_{min}}{x_{max} - x_{min}}} \qquad \textbf{Equation IV-10}$$

where, $Q_{0.75}(y)$ is the 3rd quartile of the prediction distribution of y, $Q_{0.25}(y)$ is the 1st quartile of the prediction distribution of y and $\hat{y}$ is $Q_{0.5}(y)$. The results are scaled so that the index has a value between 0 -1 with samples having values closer to 1 indicative of low uncertainty. The confidence index allows the user to evaluate the performance of a random forest model by analyzing the variability in the predictions from each tree in the random forest, such that a low variability in predictions is an indicator of accurate predictions and can be used to evaluate a models performance when deployed on a new dataset. The confidence index proposed in this chapter is novel; however, assessing prediction reliability using Quantile regression forest have been proposed by various authors [45, 46, 47]. The accurate predictions are associated with the narrow prediction intervals while the poorer predictions are associated with the wider prediction intervals. It should be noted that a wide prediction interval does not always equal a poor prediction, but a narrow prediction interval is almost always associated with accurate prediction. Hence, the confidence index tends to be pessimistic as a measure of accuracy.

Discussion of Results

In predicting the NMR T1 distributions, five algorithms ordinary least squares (OLS), ElasticNet, support vector machines (SVM), neural networks and random forest (RF). The algorithms process the train dataset which consists of 2825 datapoints and the trained model is used to predict the NMR T1 distributions of the validation dataset which consists of 1212 datapoints with each of the 10 bins predicted independent of the other. The average error for each bin is averaged to obtain the error for each sample, and the error from all samples are averaged to obtain the error from each model. The results are presented in Table 4.1. The best performing models were observed to be the random forest and artificial neural network model with an average MMAPE of 0.14 and 0.21 respectively. The sample error distribution of the random forest model on the validation dataset is presented as a histogram in Figure IV-6(a). The majority of the sample errors in the validation dataset (approximately 83% of the testing dataset) lie between 0 – 0.2 in terms of MMAPE. In Figure IV-7(a) 16 randomly selected samples predicted using the random forest models with MMAPE errors between 0 and 0.2. Figure IV-7(b) shows another 16 randomly selected samples with MMAPE errors greater than 0.3. These samples (MMAPE > 0.3) represents 3% of the test samples.

.

Table 4: NMR T1 Prediction Results from the different algorithms

Model	Result		
	cMAPE	R2 Score	RMSE
OLS	0.32	0.85	0.85
ElasticNet	0.33	0.39	0.88
SVM	0.22	0.68	0.63
Random Forest	0.14	0.84	0.4
Neural Network	0.19	0.71	0.56

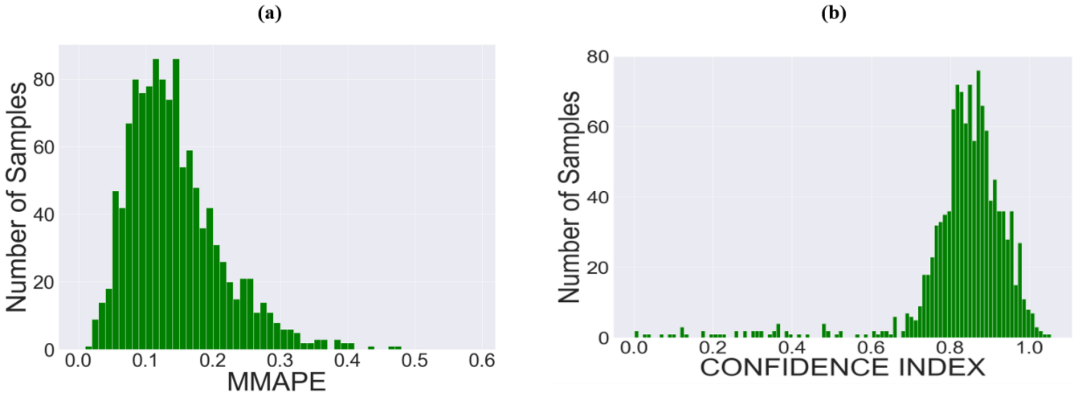

Figure IV-6: (a) Distribution of average error in terms of Modified MAPE and (b) distribution of confidence index for each depth in the test dataset

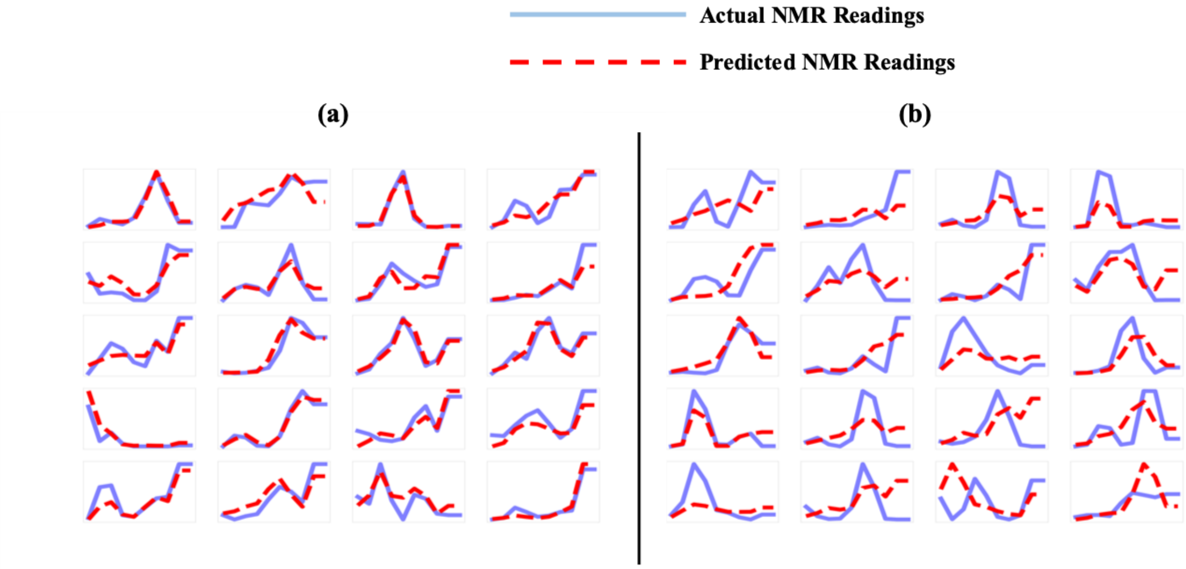

Figure IV-7: Comparison of measured NMR T1 distributions (continuous lines) against predicted NMR T1 distributions (dashed lines) for samples/depths with (a) MMAPE less than 0.2 and (b) MMAPE greater than 0.3

Confidence Index

The Quantile regression forest is fit with the train dataset and the quantile values from each prediction of the validation dataset is recorded. Predictions from the Quantile regression forests using the 1st and 3rd quartile is displayed in Figure IV-8. In Figure IV-8, the more accurate the random forest sample predictions correspond to smaller interquartile range (i.e. the thickness of the shaded interval) justifying the use of the proposed confidence index as an indicator of prediction certainty. The quartile values are used in computing a confidence index using Equations 4.9 & 4.10. The confidence index from every sample is obtained and plotted as a histogram displayed in Figure IV-6(b). Majority of confidence-index values lie between 0.8 – 1, which represent 78% of all test samples. Figure IV-9(a) displays 16 randomly selected test samples with confidence index between 0.8 and 1, while Figure IV-9(b) shows the samples with the lower confidence index (<0.7) which represents 4% of the test samples. In Figure IV-9(a), the predicted and measured NMR T1 distributions exhibit good match. In Figure IV-9(b), which shows the test samples with relatively lower confidence index, exhibits poor match between the predicted and measured NMR T1 distribution. Figure IV-9(a) and Figure IV-9(b) highlights the effectiveness of the confidence index in identifying accurate predictions without the use of a test data for reference. Hence, the confidence index can be reliably used when processing unseen, new datasets.

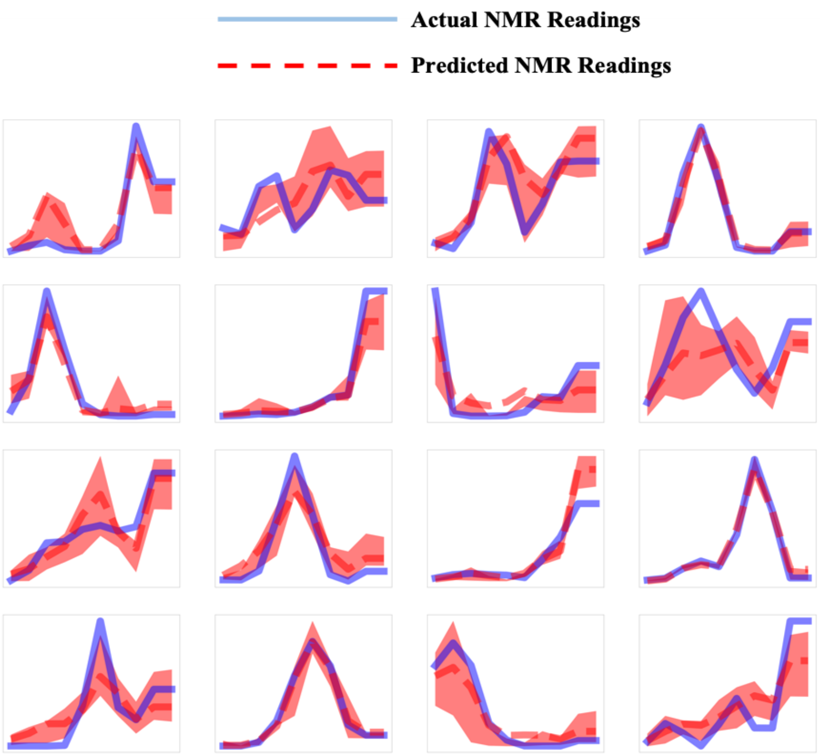

Figure IV-8: Porosity values (Vertical axis) and Bins 4ms – 2048ms (Horizontal axis). Measured NMR T1 distribution (continuous line), predicted NMR T1 distribution (dashed line) and prediction intervals (shaded region). Narrow prediction intervals indicate low uncertainty in the prediction. It should be noted that a wide prediction interval does not always mean a poor prediction, but a narrow prediction interval is almost always associated with accurate prediction

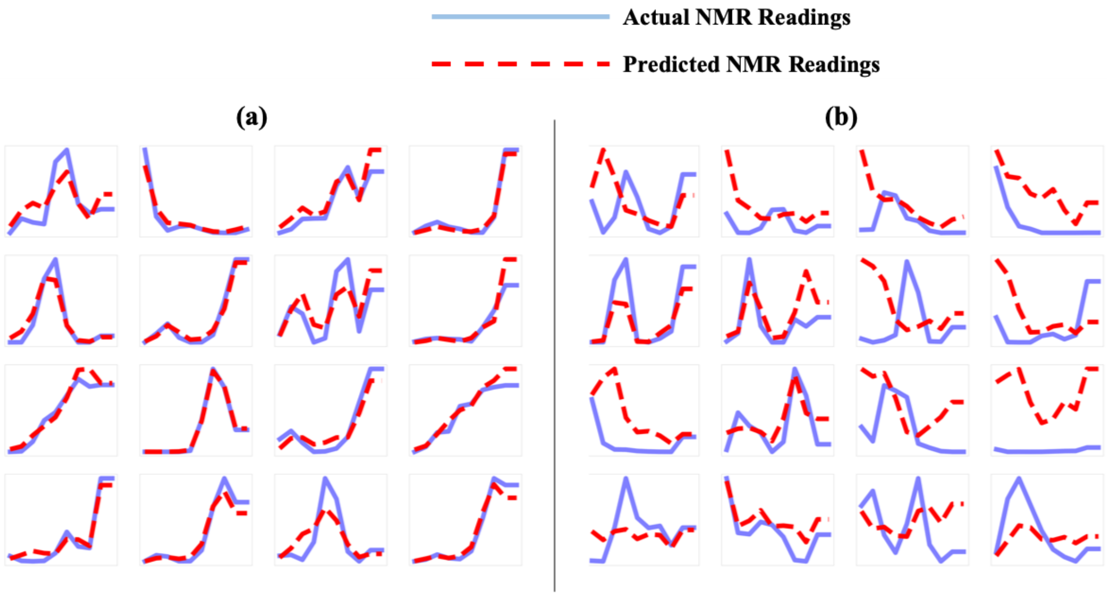

Figure IV-9: Comparison of measured NMR T1 distributions (continuous lines) against predicted NMR T1 distributions (dashed lines) for samples/depths with confidence index (a) greater than 0.7 and (b) less than 0.7

Recommendation for Future Work

This work successfully predicts the NMR T1 distribution of a well using readily available logs, however the model training and testing was done on a small dataset and the samples were performed in one well. For future work I will recommend:

- That a larger dataset from multiple wells be used for training and testing model, testing the model on entire wells is highly recommended
- Perform sensitivity analysis on how the error between the predicted and actual values affects the estimated parameters from the NMR distribution (water saturation, pore size distribution, fluid viscosity)
- Introduction of noise to the dataset to evaluate how robust the generated model is to noise and other adverse environmental conditions which are known to affect NMR measurements

Conclusions

In this chapter, three novel accomplishments are presented. A robust machine-learning workflow is applied to synthesize multitarget NMR T1 distribution along a continuous depth interval. A robust metric is proposed to evaluate the error in synthesis of NMR T1 distribution. A metric referred to as the “confidence index” is applied to quantify the uncertainty and the accuracy of the synthesis of NMR T1 distribution when the model is deployed on unseen, new data. Several data driven models were used to synthesize the NMR T1 distribution of a formation using 23 input logs. Of the 23, 7 logs (density, compressional wave velocity, Stoneley wave travel time, neutron, resistivity at 10 inches, resistivity at 30 inches and Uranium log) were selected as they were observed to provide the most information about the T1 distribution based on the feature selection methods, namely F–Test and Mutual Information. Of the models trained on the dataset, the Random Forest is the most accurate when synthesizing the NMR T1 distribution with an average MMAPE of 0.14, such that 83% of the testing samples have MMAPE values between 0 to 0.2, indicating the robustness of multitarget NMR T1 synthesis. The Quantile regression forest is then used to compute a confidence index which serves as an indicator of accuracy and certainty of multitarget synthesis. The confidence index can serve as an effective measure of accuracy during model deployment on new, unseen data for which NMR T1 distribution is not measured.

CHAPTER V

PRACTICAL APPLICATION OF MACHINE LEARNING: CASE STUDIES IDENTIFYING POROUS LAYERS IN SYCAMORE AND KEY COMPLETION PARAMETERS IN UNCONVENTIONALS

This chapter will discuss 2 case studies requiring supervised learning. They cover different areas in petroleum engineering: petrophysics/geology and well completions. The aim of this chapter is to show the workflows that can be created and the interpretation capabilities of machine learning techniques. Each case study will contain a brief introduction of the problem case and an explanation of the methods used, plots and diagrams will be used to analyze the result from each case and, finally, a technical analysis will be provided.

This chapter would:

- Use feature extraction techniques for dimensionality reduction and defining cluster labels based on the new dimensional space
- Provide insightful geological information based on the assigned cluster labels
- Use regression models in predicting key metrics in evaluating performance of unconventional wells
- Identify the key parameters that affect the performance of unconventional wells.

Case 1: Chemofacies Classification of Sycamore Core using X-ray Fluorescence Data in identifying Landing Zones

The Sycamore Formation is a regionally extensive, low permeability oil reservoir in the Admore basin of Southern Oklahoma [48]. It has two distinct members: 1) the lower, non-reservoir member which consists of glauconitic shale, and minor argillaceous-siliceous limestone and the upper member which comprises the Sycamore reservoir, consisting of thin-medium bedded, peloidal silty turbidites or liquefied sediment-gravity flows [48].

An approximately 300ft of core was obtained from the Sycamore Formation. On visual inspection 5 lithofacies were identified based on their grain size as: 1) argillaceous mudstone, 2) massive siltstone 3) bioturbated siltstone 4) planar laminated siltstone, and 5) poorly sorted fine-grained sandstone. The lithofacies classification diagram is shown in Figure V-1.

The poorly laminated sandstone and planar laminated siltstone existed in one feet respectively of the entire core. For ease of investigation the poorly laminated sandstone and planar laminated siltstone was lumped into the massive siltstone category, leading to the formation of 3 distinct categories: 1.) argillaceous mudstone, 2) massive siltstone, and 3) bioturbated siltstone.

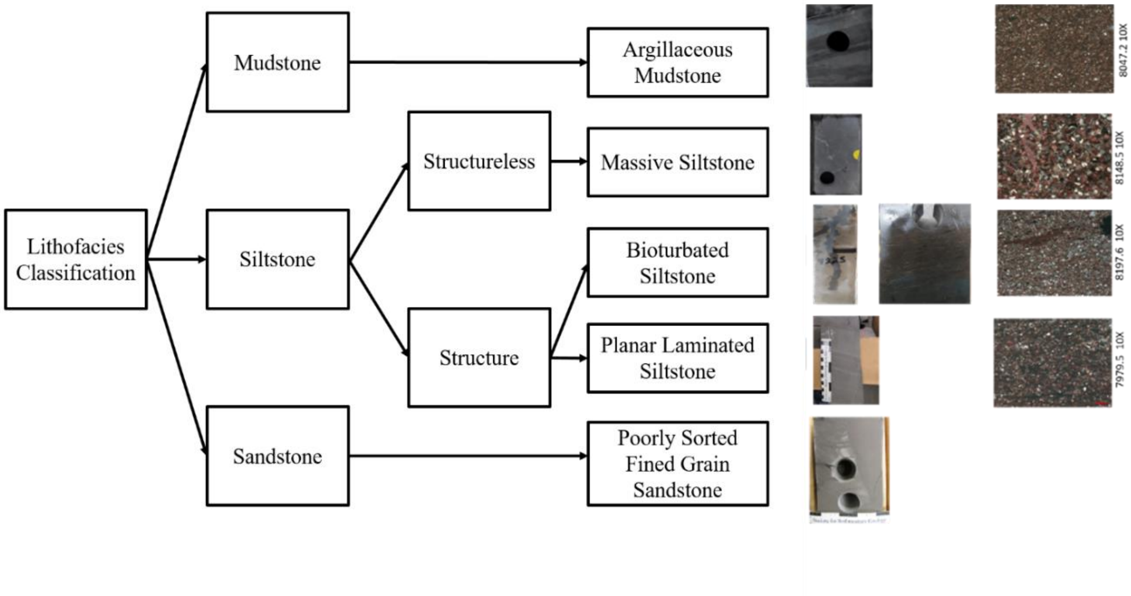

Figure V-1: Lithofacies Classification based on Visual Inspection of Core samples with thin-section images

An **X-ray fluorescence spectrometer** which is an X-ray instrument used for non-destructive chemical analyses of rocks, mineral and fluids. It does this by radiating an X-ray know as an incident beam and some of this energy is absorbed by the atoms in this material and excites it which causes emission of fluorescent X-rays with discrete energies characteristic of the elements present in the sample. An X-ray fluorescence spectrometer was used in determining the elemental composition of the core on a foot by foot basis. Some elemental composition from the XRF process is shown from Figure V-3. Using the SEM images from selected thin section samples from the siltstone, certain distinctions are observed with some looking to be more than the other. The SEM images are shown in

Figure V-2. This is further observed when the porosity is measured on randomly selected samples in the core. Porosity is seen differ in the siltstone, with some of the siltstone seeming to have high porosities and others low porosity.

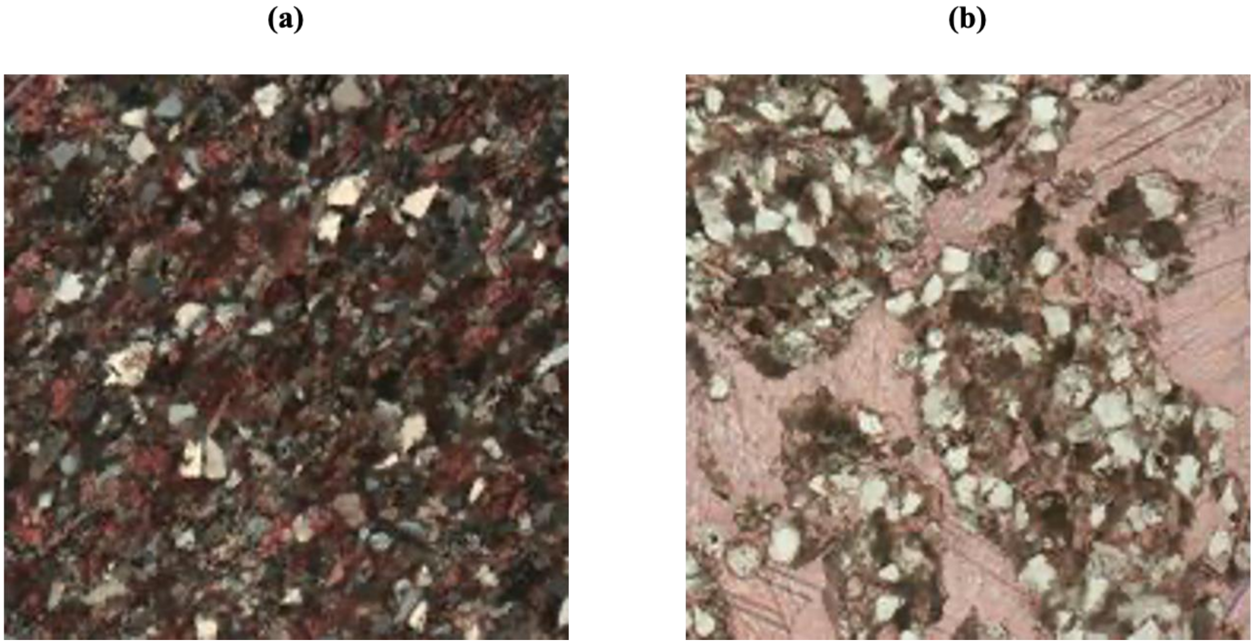

Figure V-2: Thin Sections from similar regions identified as Siltstone, from analyzing the image a) more organic content, porous and less cemented as compared to b)

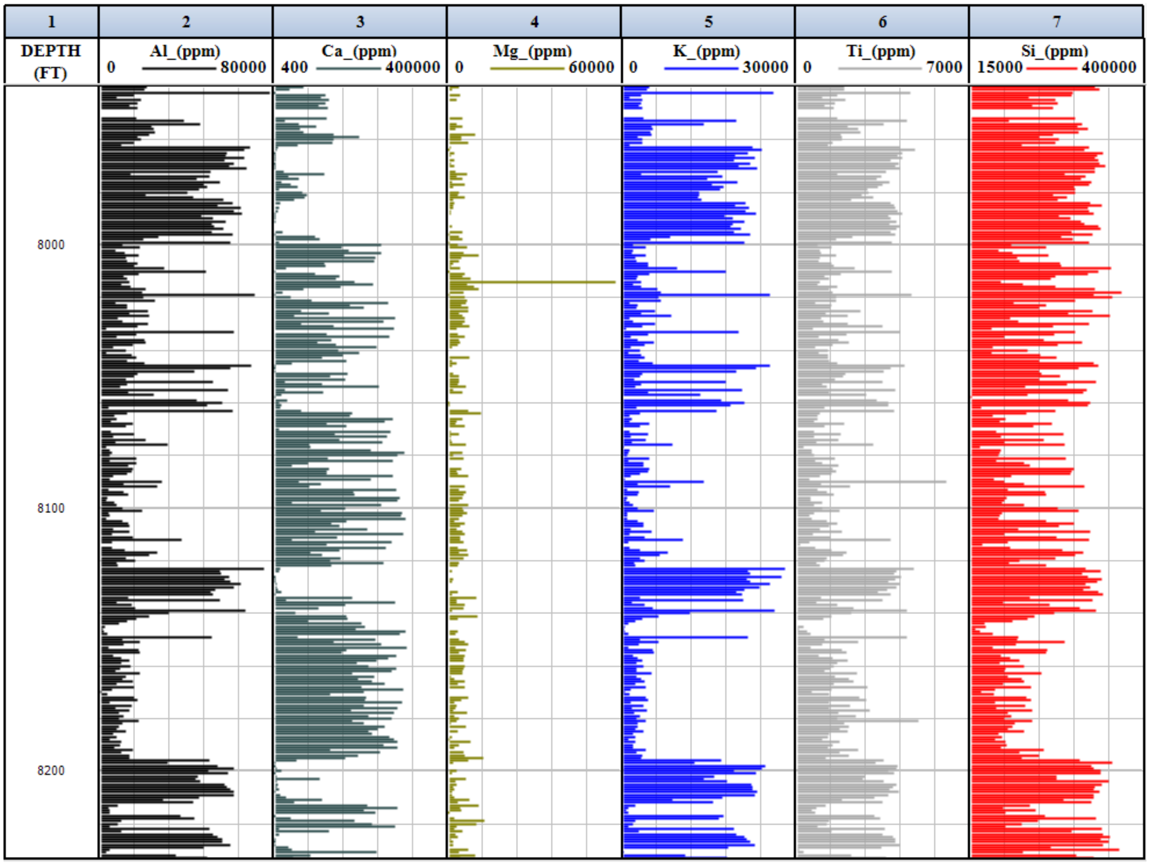

Figure V-3: Elemental Composition from XRF (X-ray Fluorescence) spectroscopy for select elements: Aluminum (Track 2), Calcium (Track 3), Magnesium (Track 4), Potassium (Track 5), Titanium (Track 6) and Silicon (Track 7)

In this case study, I aim to identify this porous siltstone by analyzing the elemental composition obtained from the XRF analysis using machine learning. This is done using unsupervised learning techniques: **PCA** (principal component analysis) and **K-Means** for feature extraction/reduction and clustering, respectively.

The XRF data consist of elemental composition of magnesium, aluminum, silicon, phosphorous, sulphur, potassium, calcium, titanium, vanadium, manganese, iron, thorium, uranium, strontium, zirconium and molybdenum. The PCA technique is used to reduce the dimensional space of the feature set before the feature set is fed into the K-Means algorithm as the K-Means algorithm is a distance-based model, which is adversely affected by a high dimensional feature space. K-means is clustering technique that can identify samples exhibiting similar properties in terms of the feature set, these techniques are further explained in the section below.

Machine Learning Algorithms

Principal Component Analysis

PCA is a technique for reducing the dimensionality of high dimensional datasets while minimizing information loss, it does this by creating new uncorrelated variables that are linear functions of the original feature set and successively maximize variance between the new variables. [51]. It creates these variables using eigenvalue/eigenvectors or SVD (single value decomposition) [51] of the centered data matrix (feature set subtracted by the mean of the individual features). As earlier explained the PCA seeks to create a new feature set of uncorrelated variables which are linear functions of the original feature while maximizing the sum of the variance from each created feature by this definition PCA is feature extraction technique as it creates a new feature set which is linear function of the original dataset i.e. the new feature set is “extracted” from the original dataset. PCA can be used for feature reduction by measuring the variance of each new feature in the new feature set and selecting the features that account for a certain percentage of the total variance (sum of the variance of each feature in the new feature set). By convention this percentage is set between 70 – 90% depending on the user.

K-Means

K-means is a clustering technique used in defining samples in a feature set with "similar" feature properties into clusters. K-means is a centroid based clustering technique and it form clusters in an n-dimensional space by initializing k (user set) random points in an n-dimensional space and each sample is in the feature set is assigned to the closest point based on the distance metrics of choice by the user (usually Euclidean). K clusters are now formed, the point is then moved to the center (centroid) of the new formed cluster based on the mean distance of the samples in the kth cluster (hence the name K-means), the samples are then re-assigned based on their distance proximity to this new point. The process is repeated n times by which a stable solution would have been found. K-means is a simple but effective method for clustering and is one of the most used clustering algorithms. The downside to the K-means algorithm is that the results become questionable as the dimensional space increases as the concept of the distance between samples becomes difficult to quantify using the usually distance metrics because of this K-means is commonly used in associated with some sort of feature reduction technique in high dimensional spaces. It is also adversely affected by outliers; the present of outliers distort the position of the centroid leading to unstable and unrepresentative clusters.

Selecting the number of clusters to be used is a crucial step in the K-Means clustering process, 3 methods were used in the study: 1) elbow plot, 2) silhouette plot and 3) domain knowledge.

- **Elbow Plot**: an elbow plot is used to visualize the reduction in variance in each cluster as the number of clusters *k* increases, the guiding logic here is that at optimum number of clusters the reduction in variation within the cluster becomes less significant. The plot of the variance measure within the cluster vs *k* value is plotted it forms an elbow shape as is seen in Figure V-4. The k value where the slope becomes noticeably less steep as it moves to the k + 1 value is considered a strong candidate for the value of k. Figure V-4 illustrates a typical elbow plot used for selecting the value of k.

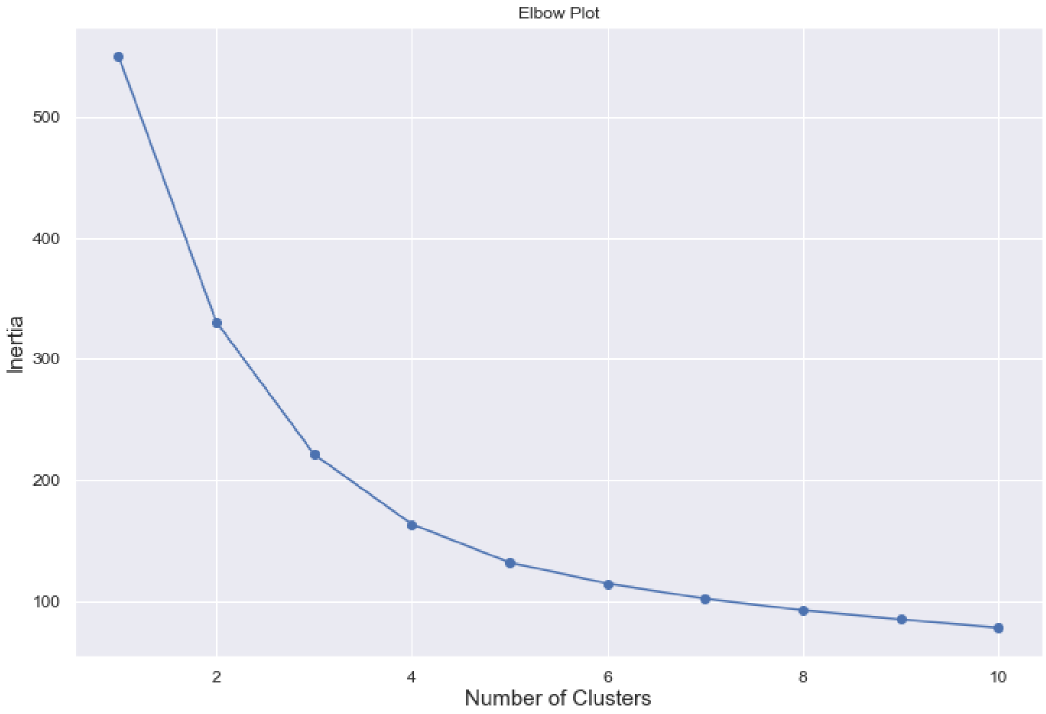

Figure V-4: Typical elbow plot used to determine the optimum number of clusters, in this case anywhere between 3 -5 would be a good choice of k

- **Silhouette Analysis**: The silhouette analysis is a tool used to measure how tightly grouped the samples in each in cluster are, and is an effective tool in identifying an optimum value of k. The silhouette coefficient is calculated using the following equation [22]:

$$\boldsymbol{s^{(i)}} = \frac{\boldsymbol{b^{(i)}-a^{(i)}}}{\textbf{max}\,\{\boldsymbol{b^{(i)},a^{(i)}}\}} \qquad \textbf{Equation V-1}$$

where $b^{(i)}$ is the cluster separation from the next closest sample, it is the average distance between the sample *xi* and all samples in the nearest cluster $a^{(i)}$ is the average distance between a sample xi and all other samples in its cluster $s^{(i)}$ is the silhouette coefficient and is bounded in the range of -1 to 1. An ideal silhouette coefficient for a sample will be as close to 1 as possible and will occur when *b(i)* >> *a(i)* and *a(i)* is a small as possible i.e. distinct and tight clusters. The silhouette score for each cluster will be the average silhouette coefficient for samples in that cluster. Not only is it necessary to search for a high silhouette score, the distribution of the silhouette coefficient should be somewhat uniform for a cluster to be classified as good. Figure V-5 shows a typical silhouette plot used for determining optimum value of *k*, each cluster label is represented by a thickness and length, the thickness and length of each bar represents the number of samples in each label and the length represents the average silhouette score. Ideally each bar should meet a set threshold for silhouette score (0.5 – 0.9) and should be spread evenly (this would depend on cluster expected by the user).

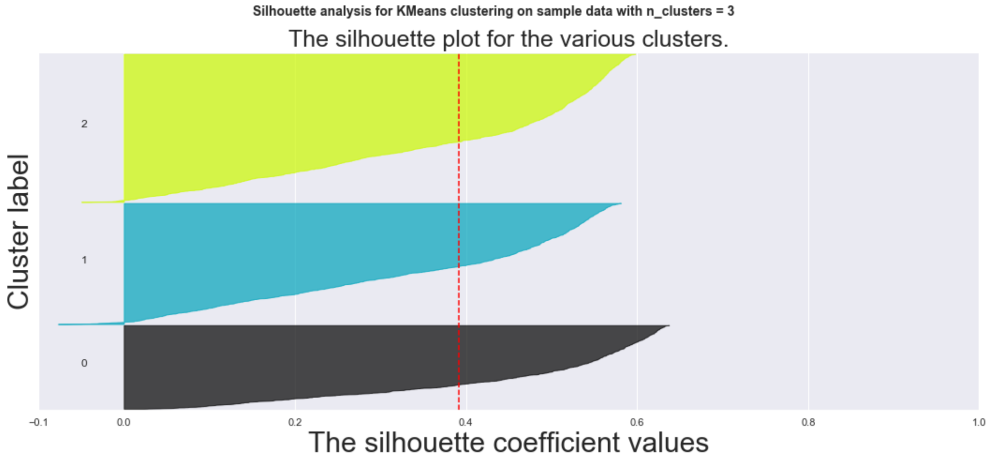

Figure V-5: A Silhouette plot for K-Means clustering on sample data with 3 clusters, it is ideal that each "finger" are of similar width and length and cross a user defined threshold for similarity

- **Domain Knowledge**: Knowledge of the problem can help decide the optimum number of clusters, as clustering is an unsupervised machine learning algorithm the algorithm has no priori knowledge of the dataset the optimum number of clusters selected by the above methods may bear no resemblance to practical scenario. Hence, I generally advice that two values of k should be selected using the elbow plot and silhouette score (which in most cases tend to point to the same answer) and the deciding factor should be based on the users understanding of the data.

The preprocessing workflow is like what has been used in previous sections, the features were scaled using a **MinMax** scaler, feature selection was done using PCA. Since K-means is adversely affected by outliers, the Isolation forest model was used to identify outliers to be excluded from the dataset based on the workflow discussed in chapter 3. The initial dataset had 291 samples after outlier detection and removal 269 samples remained i.e. 12 samples were identified as outliers.

Discussion of Results

The feature set (elemental composition of the core samples across the 300ft core) is fed to the PCA algorithm, 7 principal component (PC) features are observed to account for approximately 92% of the variance in the extracted feature set. The 7 PC features (reduced feature set) are then used create the clustering model. Using the silhouette score/plot combined with the elbow plot as seen in Figures V-6 and V-7 below it can be observed that the optimum value of k is either 3 or 4. From the elbow plot 3 – 5 are reasonable choices for k, looking at the silhouette plot for k = 5, the width of label 2 looks very thin, this disqualifies 5 as a choice for k. Since the aim of this clustering model is delineate between the porous/non-porous siltstone combined with fact that major elemental differences between the bioturbated and argillaceous mudstone is not expected. 3 was decided as the optimum choice of k.

The composition of certain proxy elements is plotted for each cluster label using a box plot (Figure V-8) to interpret the results of the clusters. Looking at the boxplot from this proxy elements label 1 likely represents the mudstone with its high aluminum and titanium content which are proxies for clay and organic content and low calcium content and silicon/aluminum ratio.

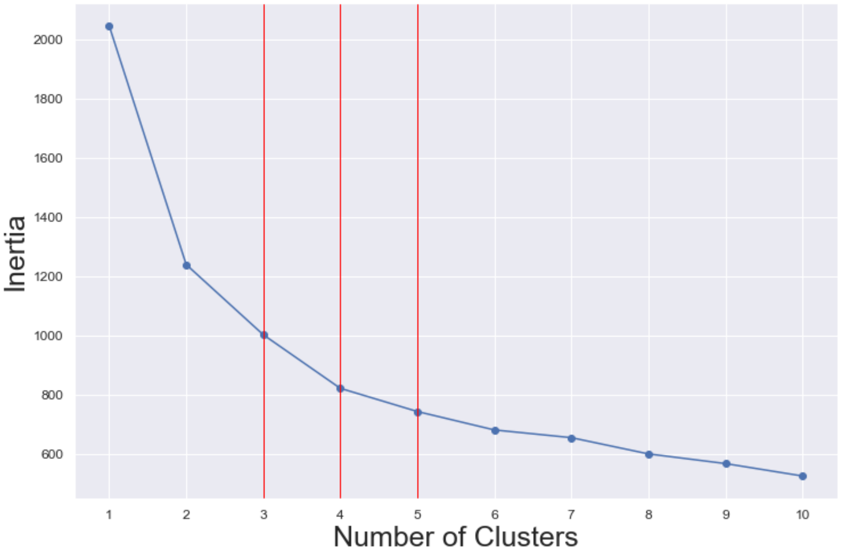

Figure V-6: Elbow plot to select optimum k for clustering

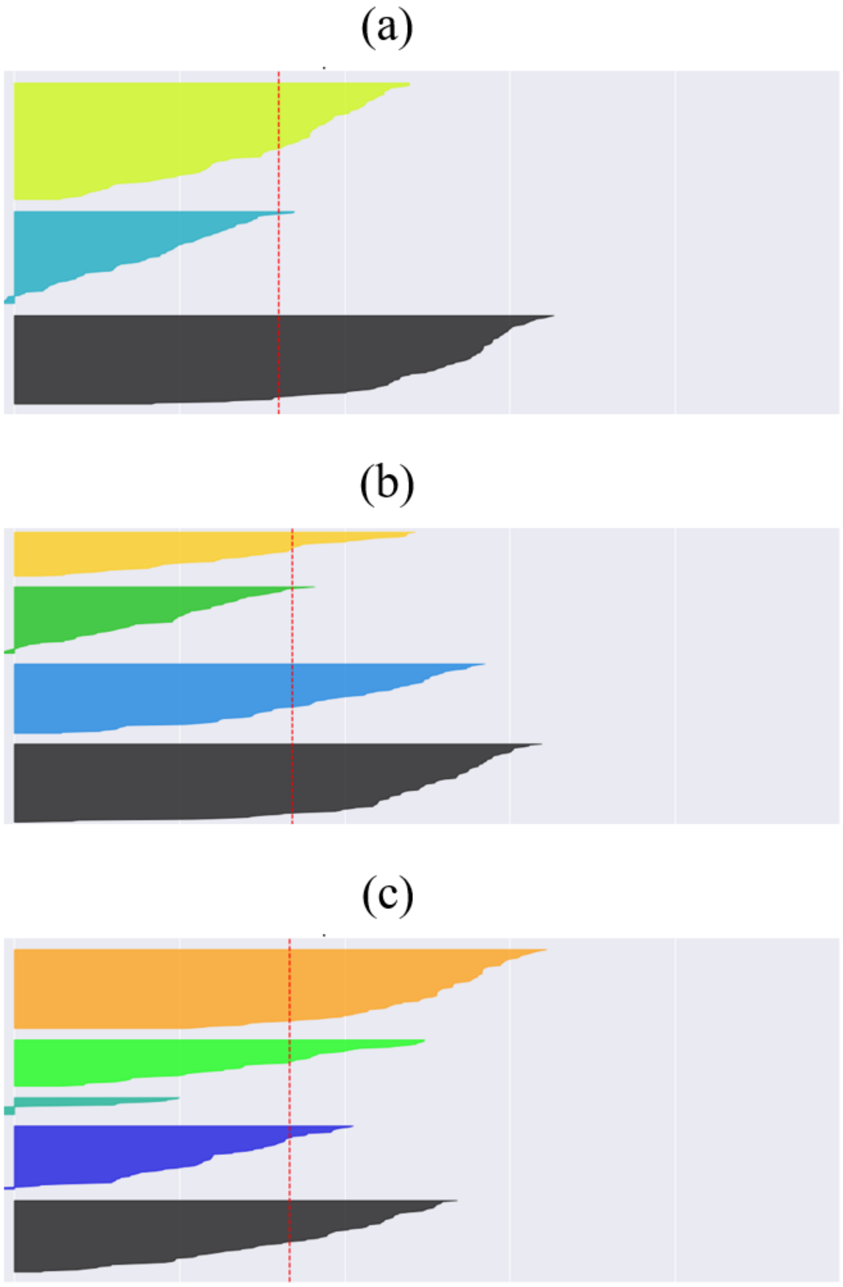

Figure V-7: Silhouette Plot used for analysis a) n_clusters = 3, b) n_clusters = 4, and c) n_clusters = 5

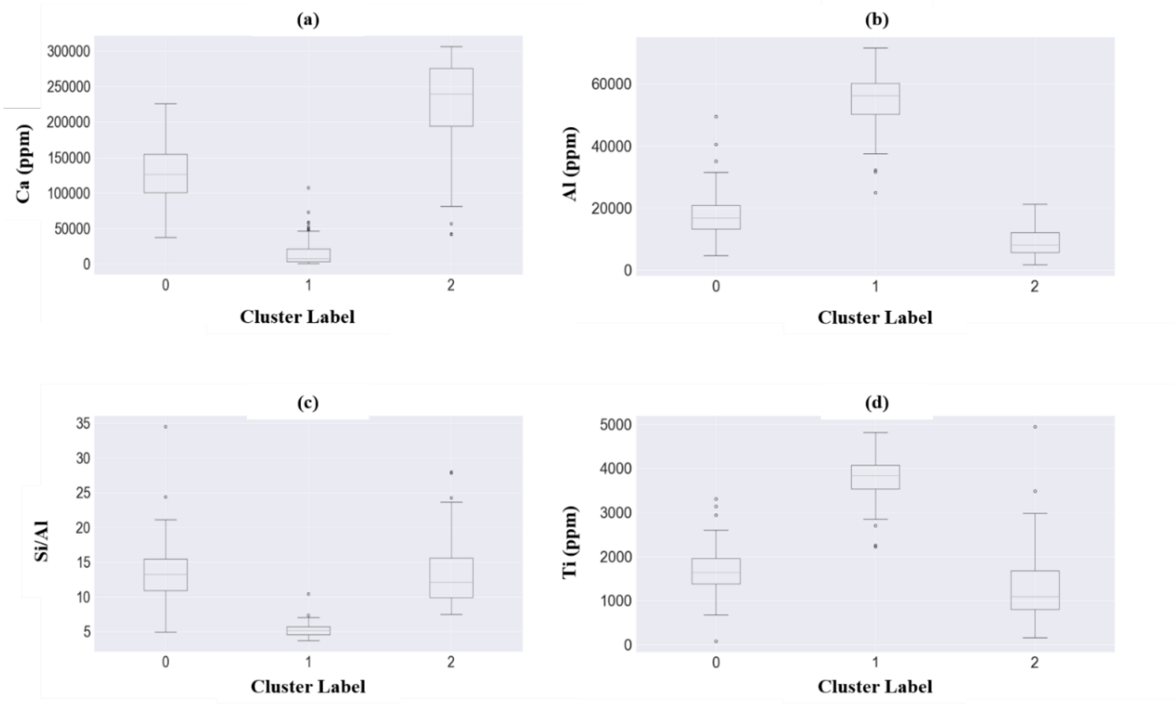

Figure V-8: Boxplot showing distribution of certain proxy elements in the cluster labels: a) Calcium, b) Aluminum, c) Si-Al ratio and d) Titanium

Label 0 and 2 seems to be the siltstone with their relatively high Si/Al ratio and lower aluminum content. One key difference however is their carbonate content with Label 2 having significantly higher values for calcium than label 0, also label 0 seems to have on average slightly higher values of aluminum and titanium indicating a higher organic content than label 2. Collating this evidence, it is hypothesized that label 0 represents the more porous siltstone while label 2 represents the less porous siltstone. The porosity difference I hypothesize is due to less cementation in label 0 than label 2 using the lower amount of carbonate present as evidence and also the increased amount of organic material in label 0 (higher amounts of aluminum and titanium which are proxies for organic content). This is likely due to the depositional process that led to the formation of the sycamore and meramac formation, Figure V-9 shows the comparison of the cluster defined labels and the labels observed by visual inspection. Two things can be quickly observed. One is that there is good match with what has been written so far also the porous siltstone seem to occur less as depth increases. Steady increase of organic matter which promotes carbonate dissolution during the geological depositional process is the likely cause of this phenomenon.

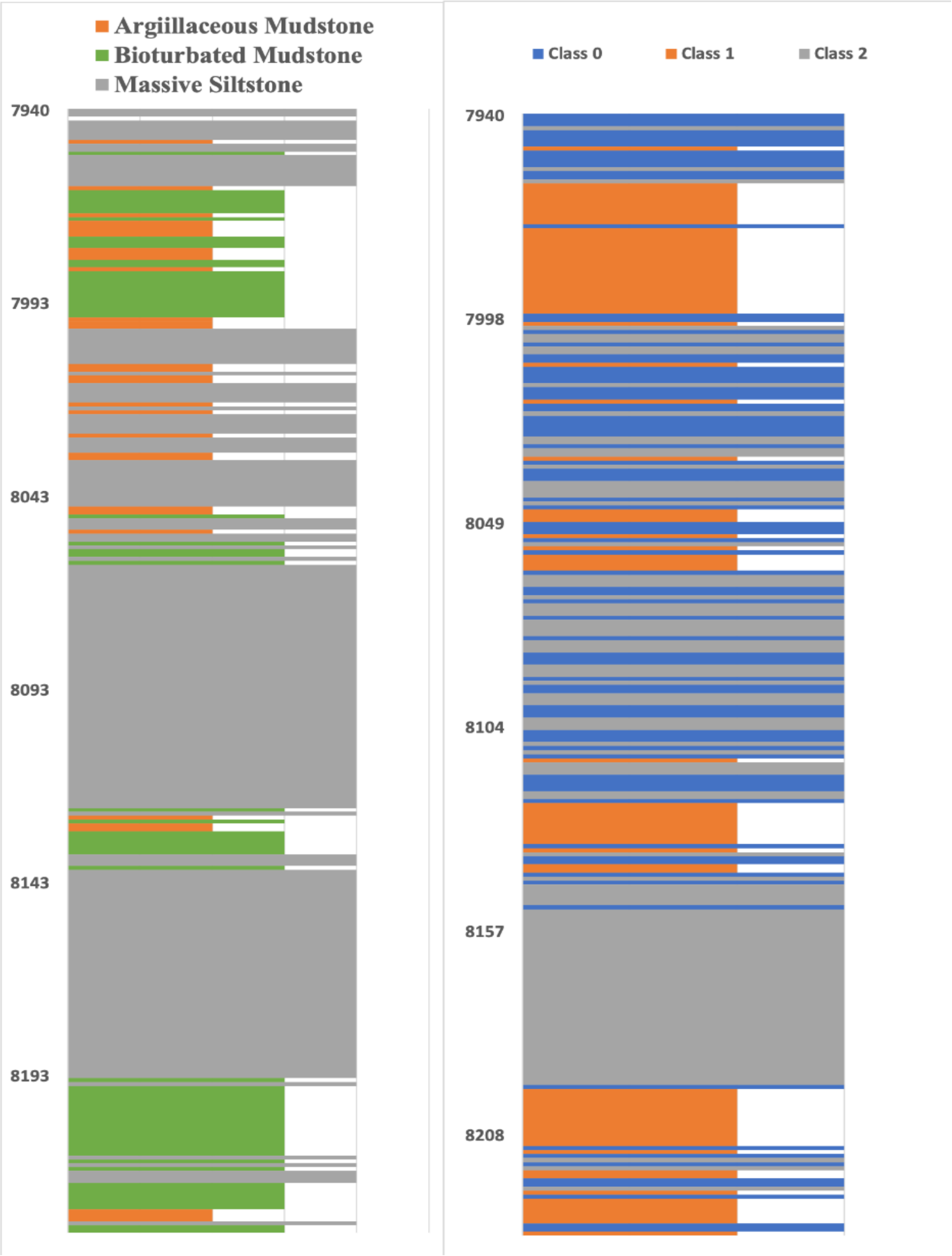

Figure V-9: Comparing the cluster labels from K-means and Lithofacies obtained from visual inspection

To further validate this study, thin section images of the depths in lower end of core representing points that have been identified as non-porous siltstone were compared with images at shallower depth in points identified as porous siltstone are compared. The images identified as porous limestone seem to show more organic content and less carbonate content as compared to the images obtained from region identified as less porous as seen in Figure V-10.

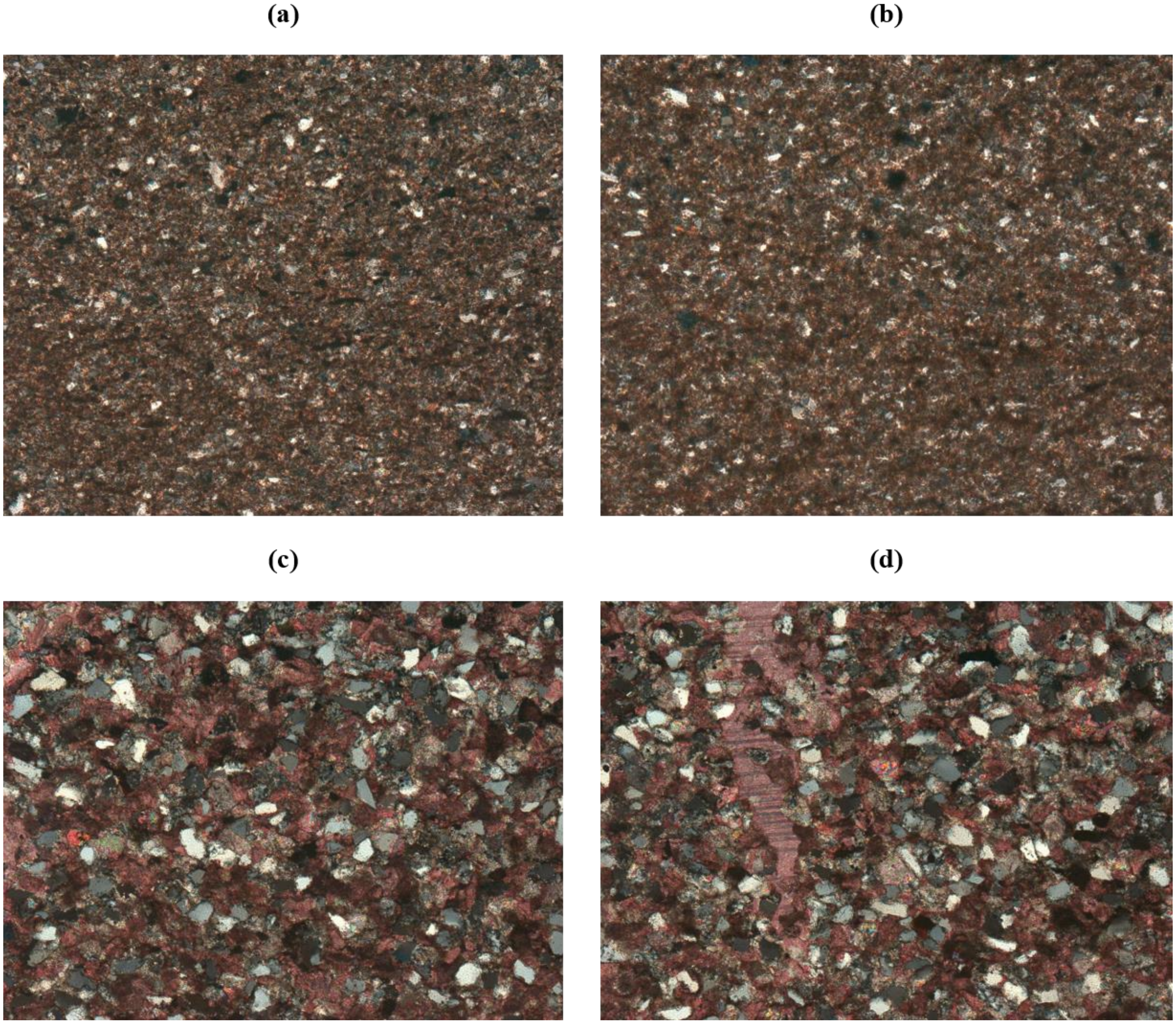

Figure V-10: Comparing thin section images from the upper core section containing what was classified as porous siltstone (a & b) with the images from the lower core section containing what was classified as cemented siltstone (c & d) both siltstone were considered to be the same when investigated visually (images are taken at the same resolution)

Case 2: Identifying Key Parameters in Completion Design

Oil and gas from unconventional sources in North America such as shale and tight sands has increased significantly in recent times. Completion has always been a key aspect in the petroleum production process and how a well is completed can significantly affect the production of the well. In unconventional reservoirs, how the well is completed plays a crucial role in determining if a well would ultimately be economic or not. The completion process for unconventional wells are typically more complex as more operations are performed from the onset such as hydraulic fracturing, also the wells are completed horizontally to contact as much reservoir area as possible. Identifying the key factors that affect this operation can aid engineers in completing wells more effectively and aid the decision-making process.

Machine learning models can be used for prediction, this point has already been established in previous chapters, and it can also be used to identify the features that mostly affect the target (the variable to be predicted). In this case study, a parameter grid is fed to a simulation software and the corresponding estimated ultimate recovery (EUR) which is the approximate amount of oil/gas that can potentially be recovered or has already been produced from a well or reserve and IP90 which is the average daily production after the first 90 days of production, is recorded. The IP90 has gathered attention in recent times as a key metric in evaluating unconventional wells, since majority of the production from unconventional wells comes in the early days due to the steep decline observed in unconventional wells. Wells with high IP90s are economically viable.

The parameters are lateral spacing, area (areal spacing), total vertical depth, lateral length, stages, perforation cluster, sand intensity, fluid intensity, pay thickness, fracture ½ length and fracture conductivity. In this case the values used in the parameter grid is the feature set while IP90 and EUR are the targets. The aim of this work is to the identify the key parameters of the ones listed that affect the IP90 and EUR of a gas well and compare the parameters that affect each target, this would be done by splitting the dataset into a train and test dataset creating a model using several algorithms and the train dataset. The models that performs best on the test dataset is selected and the key features are identified. In this study I would be using three algorithms for model training: 1) OLS 2) ElasticNet and 3) Random Forest. These models have already been explained in previous sections but the mechanism for identifying the driving features are explained below.

Feature Importance Computation from Select Algorithms

Linear Models

Linear models such OLS (ordinary least squares), LASSO (least absolute shrinkage and selector operator) and ElasticNet computes a coefficient β for each feature variable such linear combination of the coefficients and the variables results in the predicted the value. The coefficient β is optimized to minimize an objective function, for the OLS that function the sum of square error between the predicted value and the actual value (SSE) as expressed in equation 4.2 and for the ElasticNet the function is the SSE plus the L1 and L2 norms (distance between coefficients) for regularization, this is expressed in equation 4.3 and 4.4.

This coefficient provides useful information about the final result, as the larger the absolute value of the coefficient assigned to a variable the larger the effect that variable has on the predicted value, it also provides direction using sign with a positive sign inferring a direct relationship and negative sign an inverse relationship. It is important to note that this is only effective when the feature set has been scaled such that each feature has the same range.

Random Forest

The random forest algorithm consists of n number of trees called estimators with each tree consisting of randomly selected features and samples. The use of randomly selected features and samples is done to make each tree uncorrelated and independent of each other, thereby reducing the risk of overfitting. A good way to think about this is imagine a police officer question 2 suspects in different rooms, this way the information provided by one suspect is independent of the other and the officer can get an unbiased information from the suspects. The random forest determines feature importance by one of two methods: 1) mean decrease in impurity, 2) Permutation importance.

- **Mean Decrease in Impurity**: For each decision tree a split is performed to minimize the impurity in each node i.e. the measure of how often a randomly chosen target sample would be labelled incorrectly if they were randomly labelled based on their distribution. The feature that best achieves this in a particular node would be used for splitting a tree. The mean decrease in impurity measures the sum of the decrease in impurity for all features divided by the number of time it was used for splitting proportional to the number of samples it splits (remember each tree does not have the same number of samples). The feature with the highest value is considered to be the most importance in predicting the target sample.

- **Permutation Importance/Mean Decrease Accuracy:** In this method values from each feature are randomly shuffled *n* times such that the information from the feature is distorted and the mean decrease in accuracy for all n times is recorded. The feature with largest decrease in accuracy is considered to be the most important as the information distortion from that feature most affects the model results.

In this study the **mean decrease in impurity** was used in a calculating the feature importance from the Random Forest.

Discussion of Results

Before the models are trained, the correlation between the features and the target (EUR and IP90) is checked using spearman and pearson correlation coefficients. In both cases the **area**, **lateral length**, **stages** and **perforation cluster** have the highest correlation to the target. The barplot of the feature correlation with the target is shown in Figure V-11.

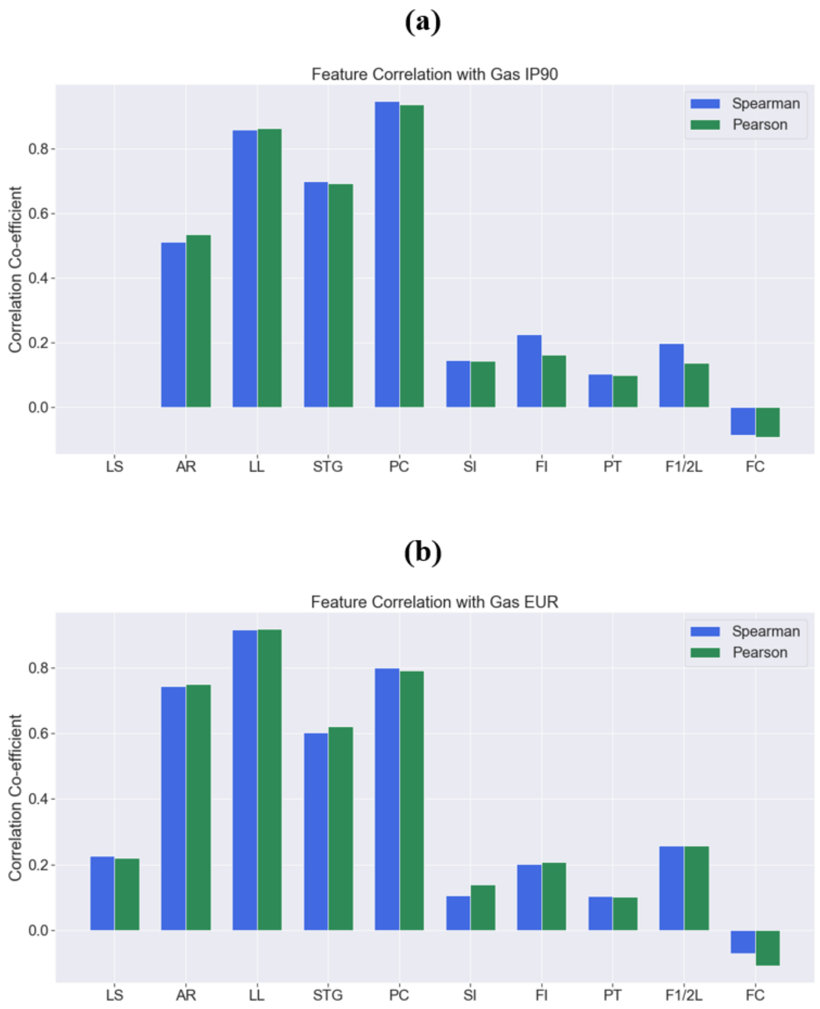

Figure V-11: Feature correlation with the target: a) Feature correlation with IP90 b) Feature correlation with EUR

The algorithms are trained on 1349 randomly selected samples and created models are tested on the remaining 579 samples. The Random Forest performs best in predicting both

EUR and IP90 with an r2 score of 0.99 for both case and an RMSE (root mean square error) of 0.077 and 37.8 respectively on the random selected test samples. The results are shown in Table 5.1, the comparison of the predicted and actual target values is shown in Figure V-12.

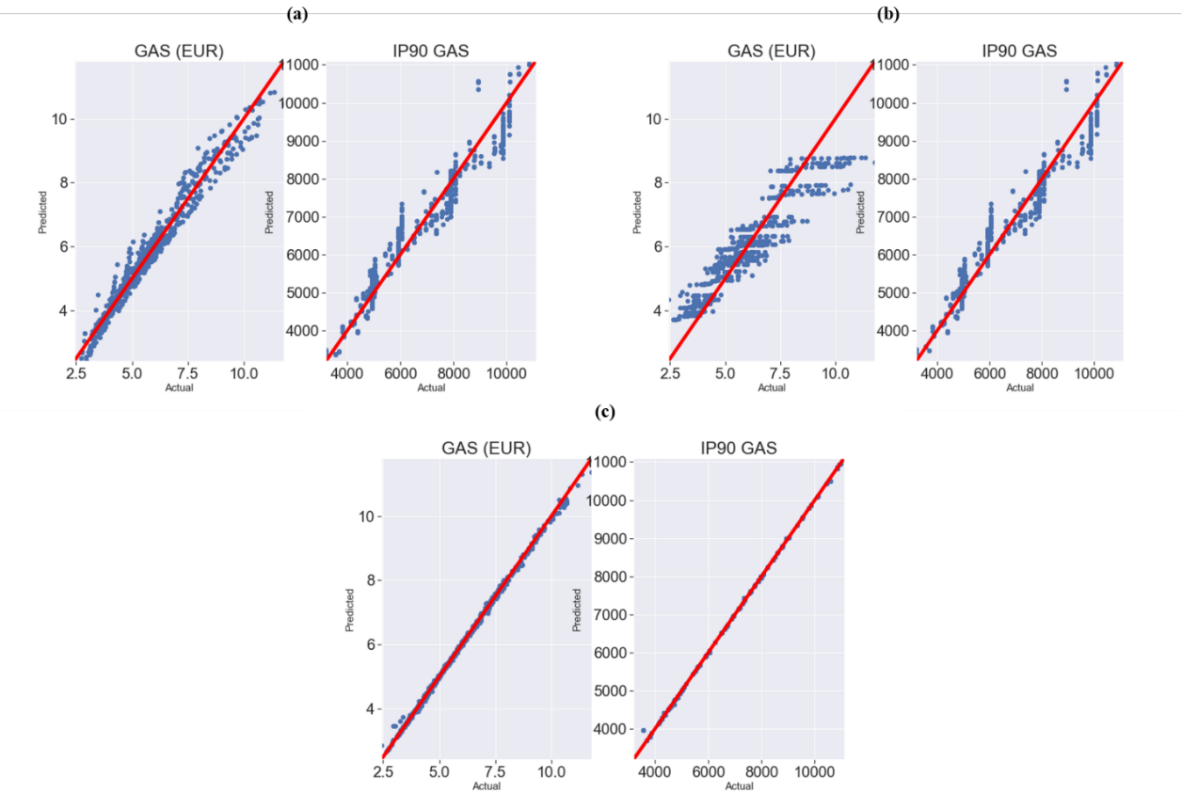

Figure V-12: Comparing the predicted and actual values of EUR and IP90 from the 3 models used: a) OLS b) ElasticNet c) Random Forest

Table 5: Result table comparing the model results in case study 2

MODEL	TARGET	METRICS	
		R2 SCORE	RMSE
OLS	EUR (Bcf)	0.96	0.37
	IP90 (Mcf/D)	0.94	532
ELASTICNET	EUR (Bcf)	0.7	0.767
	IP90 (Mcf/D)	0.94	532
RANDOM FOREST	EUR (Bcf)	0.998	0.08
	IP90 (Mcf/D)	0.999	33.8

Of the linear models, the Ordinary least square model performed the best with an r2 score of 0.96 and 0.94 respectively. As the random forest model performs the best of the 3, its feature importance ranking will take precedence and will be the primary consideration, coefficients from the OLS would then be used to verify the results of the Random Forest. Ideally the random forest's mean decrease in impurity and the OLS coefficients should be pointing towards the same variables/features, also the OLS coefficient provides information on the directional relationship between the feature and target through the signs (i.e. does an increase in the value of this feature lead to an increase in the value of the target). Figures V-13 and V-14 show the normalized mean decreased in impurity for the random forest and coefficients for each variable from the OLS model respectively. The plots highlight that the lateral length is the key factor in terms of determining the EUR which makes sense as the longer the lateral the larger the area contacted by the reservoir, however this should be taken with a grain of salt as we know that this relationship is not perfectly linear as the longer the lateral the larger the frictional effect on the fluid which can ultimately impair the fluid flow. However, generally it can be said that the longer the lateral length the larger the ultimate recovery from that well, interestingly though when you look at IP90 the effect of lateral length is largely diminished and the perforation cluster seems to be the key factor, with IP90 having a direct relationship with perforation cluster as observed from the positive coefficient from the OLS model.

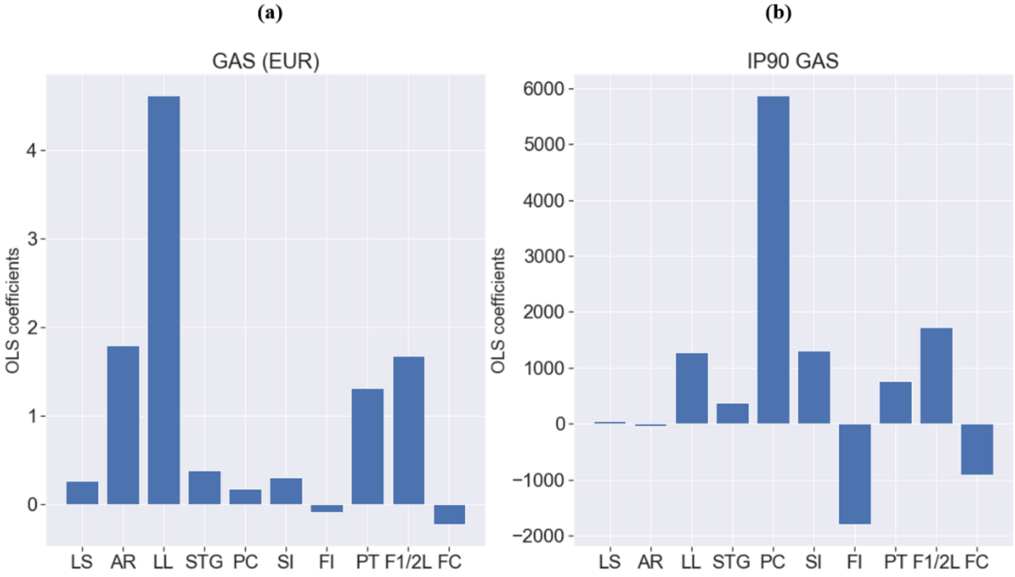

Figure V-13: Barplot of the coefficient values from each features with respect to the target: a) Target: EUR, b) Target: IP90

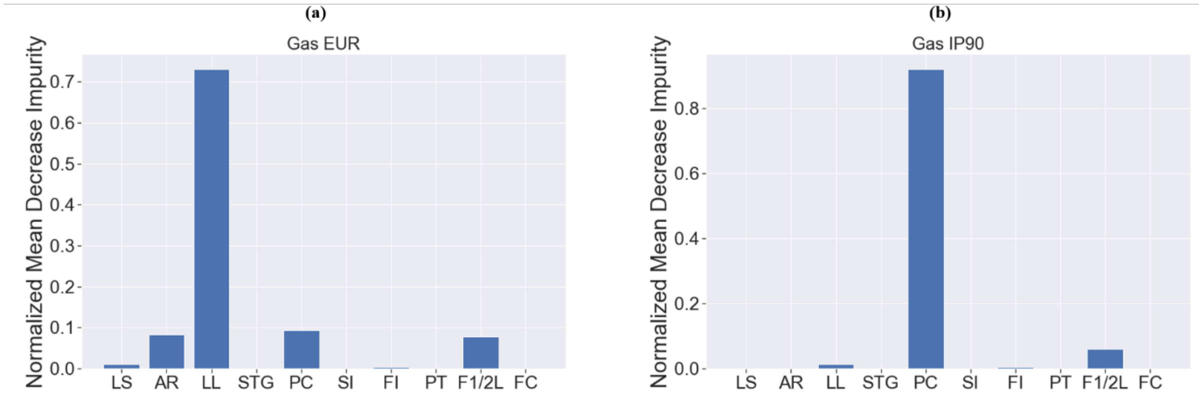

Figure V-14: Barplot of the normalized mean decrease impurity for each feature with respect to the target: a) Target: EUR, b) Target: IP90

Knowing that the most productive time of an unconventional well is in its early life this poses an important question to the operator, should completion funds be diverted unto making more perforations per unit length and recouping funds are quick as possible or drilling longer laterals and getting more production over a longer period. This decision would involve the business team (time value of money) and technical department (would the perforation close over time leading to underperformance of the well?).

These points are highlighted to show that as powerful as machine learning is as a tool for analysis, domain knowledge and expertise in the field is still required to make the best of the information provided to best maximize it potential.

Conclusions

In this chapter, 2 case studies were explored to highlight practical aspect of machine learning applications in petroleum engineering and how data driven methods can provide quick insights to the technical and business development teams when machine learning is combined with expertise. Potential landing zones in the sycamore were identified by locating reservoir structures in the sycamore siltstone using the elemental composition of the core also information on the depositional environment of the formation was hypothesized based on model evidence. The second case looked at involved identifying the key completion parameters that affect the production potential of an unconventional well using IP90 and EUR as a metric. It was found that the lateral length of the drilled well is the most important factor in term of estimated ultimate recovery and perforation cluster (perforation per unit length) as the key factor with regards to IP90. The question as to what metric should be considered when planning an unconventional well was left open to the operator's preference but fair warning was highlighted about the frictional energy losses that can occur in longer laterals and maximizing early time productivity in unconventional wells.

CHAPTER VI

CONCLUSIONS

This study provides practical applications of data driven methods for exploration and production, exploring the general machine learning workflow and shedding some light on the current state of the art machine learning practices in the oil and gas industry in the first two chapters. Chapter 3 and 4 presents the results of my independent study on applications dealing with machine learning assisted outlier detection in subsurface signals and signal prediction using machine learning algorithms with NMR distributions used as a case study. Chapter 5 explores two case studies with the application of machine learning in geology/petrophysics and well completions and seeks to explore the practical aspects of machine learning and the need for subject matter expertise. These studies brought about the following conclusions:

- Machine learning methods can be used for outlier detection in well log and by implication subsurface measurement data, such as seismic data.
- Machine learning assisted outlier detection models trump the current convention for outlier detection by considering feature interaction in multivariate data, which is common for most datasets from the industry, as well as being able to handle contextual outliers.
- The Isolation Forest is a powerful tool for outlier detection in most cases and when information about the dataset is limited it is the best option of the algorithms tested in this work with Balance accuracy score of over 80% in all cases

explored.

- Signals and multi-target dataset can be successfully predicted using machine learning models.
- The random forest outperforms all other models explored for predicting the NMR distribution with an MMAPE of 0.15.
- Using the quantile regression forest, we can compute an effective method for validating the model without the use of a validation dataset.
- The confidence index can be used as a potential tool for stochastic analysis when the model is being deployed in a new well.
- Clustering methods can be used to identify difficult to spot pattern in complex subsurface data as is seen in case 1 in chapter 5, where we identified the porous beds in the core sample using elemental composition and provided a plausible explanation for this occurrence.
- Key Factors and drivers affecting complex flow problems can be identified using data driven techniques which can in turn spur healthy technical and business debate as can be seen in chapter 5 case 2 where lateral length was identified as the major factor in increasing EUR and more perforation clusters as the main factor in increasing IP90.

REFERENCES

[1] J. Pollock, Z. Stoecker-Sylvia, V. Veedu, N. Panchal, and H. Elshahawi, “Machine Learning for Improved Directional Drilling,” presented at the Offshore Technology Conference, Houston, TX, 2018, doi: https://doi.org/10.4043/28633-MS.

[2] J. Zhao, S. Yuelin, Z. Zhengxin, and S. Jihnston, “Machine Learning-Based Trigger Detection of Drilling Events Based on Drilling Data,” presented at the SPE Eastern Regional Meeting, Lexington, Kentucky, Oct. 2017, doi: https://doi.org/10.2118/187512-MS.

[3] R. Zhong, R. Johnson, and Z. Chen, “Using Machine Learning Methods to Identify Coals from Drilling and Logging-While-Drilling LWD Data,” presented at the SPE/AAPG/SEG Asia Pacific Unconventional Resources, Brisbane, Australia, Nov. 2019, doi: https://doi.org/10.15530/AP-URTEC-2019-198288.

[4] S. Bhowmik, G. Noiray, and H. Naik, “Riser Design Automation with Machine Learning,” presented at the Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE, Nov. 2019, doi: https://doi.org/10.2118/197219-MS.

[5] Y. Wu, S. Misra, C. Sondergeld, M. Curtis, and J. Jernigen, “Machine learning for locating organic matter and pores in scanning electron microscopy images of organic-rich shales,” *FUEL*, vol. 253, pp. 662–676, Oct. 2019.

[6] J. He, S. Misra, and H. Li, “Comparative Study of Shallow Learning Models for Generating Compressional and Shear Traveltime Logs,” *Petrophysics*, vol. 59, no. 06, pp. 826–840, Dec. 2018.

[7] H. Li and S. Misra, “Predicition of subsurface NMR T2 distribution from formation-mineral composition using variational autoencoder,” presented at the SEG International Exposition and Annual Meeting, Houston, Texas, 2017.

[8] V. Gaganis and N. Varotsis, “Machine Learning methods to Speed up Compositional Reservoir Simulation,” presented at the SPE Europe/EAGE Annual Conference, Copenhagen, Denmark, Jun. 2012, doi: https://doi.org/10.2118/154505-MS.

[9] C. Onwuchekwa, “Application of Machine Learning Ideas to Reservoir Fluid Properties Estimation,” presented at the SPE Nigeria Annual International Conference and Exhibition, Lagos, Nigeria, Aug. 2018, doi: https://doi.org/10.2118/193461-MS.

[10] S. T. Dang, H. Han, C. Sondergeld, and C. Rai, "Application of Machine Learning to NMR measurements in Determining Fluid Saturation," presented at the SPWLA 61st Annual Logging Symposium, Jun. 2020.

[11] Q. Cao, R. Banerjee, S. Gupta, J. Li, W. Zhou, and B. Jeyachandra, "Data Driven Production Forecasting Using Machine Learning," presented at the SPE Argentina Exploration and Production of Unconventional Resources Symposium, Buenos Aires, Argentina, Jun. 2016.

[12] T. Ounsakul, T. Siriattanachatchawan, W. Pattarachupong, and P. Ekkawong, "Artificial Lift Selection using Machine Learning," presented at the International Petroleum Technology Conference, Beijing, China, Mar. 2019, doi: https://doi.org/10.2523/IPTC-19423-MS.

[13] K. Patel and R. Patwardhan, "Machine Learning in Oil & Gas Industry: A Novel Application of Clustering for Oilfield Advanced Process Control," presented at the SPE Middle East Oil and Gas Show and Conference, Manama, Bahrain, Mar. 2019, doi: https://doi.org/10.2118/194827-MS.

[14] S. Omrani, I. Dobrovolschi, K. Loh, and P. C. Belfroid, "Slugging Monitoring and Classification with Machine Learning," presented at the BHR 19th International Conference on Multiphase Production Technology, Cannes, France, Jun. 2019.

[15] S. Ando, "Clustering needles in a haystack: an information theoretic analysis of minority and outlier detection.," 2007, pp. 13–22.

[16] H. Zengyou, X. Xiaofei, and D. Shengchun, "Discovering cluster-based local outliers," *Pattern Recognition Letter*, vol. 24, pp. 9–10, 2003.

[17] N. Chaudhary and J. Lee, "Detecting and removing outliers in production data to enhance production forecasting," presented at the SPE/IAEE Hydrocarbon Economics and Evaluation Symposium, Houston, TX, 2016.

[18] M. Luis, S.-P. Nayat, and M. Jose, "Anomaly Detection based on sensor data in petroleum industry applications," *Sensors*, vol. 15, no. 2, pp. 2774–2797, 2015.

[19] E. Lewinson, "Outlier detection with isolation forest," *Towards Data Science*, Jul. 2018.

[20] T. Liu, K. Ming, and Z.-H. Zhou, "Isolation Forest," presented at the IEEE International Conference, Pisa, Italy, 2008.

[21] Ridvan Akkurt, Tim T. Conroy, David Psaila, Andrea Paxton, Jacob Low, and Paul Spaans, "Accelerating and Enhancing Petrophysical Analysis with Machine

Learning: A Case Study of an Automated System for Well Log Outlier Detection and Reconstruction," presented at the SPWLA 59th Annual Logging Symposium, London, UK, 2018.

[22] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn and TensorFlow,* 2nd Edition. Packt, 2017.

[23] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial database with noise," presented at the International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, 1996.

[24] P. Weing, "Local outlier factor for anomaly detection," *Towards Data Science*, 2018.

[25] F. Pedregosa and et al, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] C. Minh, V. Jain, D. Maggs, D. Murray, and Y. Xiao, "Estimation of Sw from NMR T2 Logging," presented at the SPWLA 59th Annual Logging Symposium, London, UK, Jun. 2018.

[27] W. E. Kenyon, "Petrophysical Principles of Applications of NMR Logging," *The Log Analyst*, vol. 38, no. 02, Mar. 1997.

[28] Li, H., Misra, S., & He, J. (2019). Neural network modeling of in situ fluid-filled pore size distributions in subsurface shale reservoirs under data constraints. *Neural Computing and Applications*, 1-13.

[29] M. Mullen, J. Bray, and R. Bonnie, "Fluid Typing with T1 NMR: Incorporating T1 and T2 Measurements for Improved Interpretation in Tight Gas Sands and Unconventional Reservoirs," presented at the SPWLA 46th Annual Logging Symposium, New Orleans, Louisiana, 2005.

[30] X. Liang, M. Zhi-qiang, W. Zhao-nian, and J. Yan, "Application of NMR logs in tight gas reservoirs for formation evaluation: A case study of Sichuan basin in China," *Journal of Petroleum Science and Engineering*, vol. 81, pp. 182–195, Jan. 2012, doi: https://doi.org/10.1016/j.petrol.2011.12.025.

[31] V. Dragan, G. Derrick, and D. Mick, "Determination of Natural Fracture Porosity using NMR," presented at the SPE/AAPG/SEG Unconventional Resources Technology, San Antonio, Texas, Aug. 2016, doi: https://doi.org/10.15530/URTEC-2016-2447768.

[32] George R. Coates, Lizhi Xiao, and Manfred G. Prammer, *NMR Logging Principles and Applications*. Haliburton Energy Services Publication, 1999.

[33] R. L. Kleinberg, C. Stanley, W. E. Kenyon, R. Akkkurt, and Farooqui, "Nuclear Magnetic Resonance of Rocks: T1 vs T2," presented at the Annual Technical Conference and Exhibition, Houston, Texas, 1993.

[34] C. Lei *et al.*, "A comparison of random forest and support vector machine approaches to predict coal spontaneous combustion in gob" *FUEL*, vol. 239, pp. 297–311, Mar. 2019.

[35] M. Buscema, "Back Propagation Neural Networks," *Substance Use & Misuse*, vol. 33, no. 2, Feb. 1998, doi: https://doi.org/10.3109/10826089809115863.

[36] L. Breiman, "Random Forests." University of California, Berkeley, CA, Jan-2001.

[37] Nicolai Meinshausen, "Quantile Regression Forests," *Journal of Machine Learning Research*, vol. 7, pp. 983–999, 2006.

[38] R. Fritz, P. Medlock, M. Kuykendall, and J. Wilson, "The Geology of the Arbuckle Group in the Midcontinent: Sequence Stratigraphy, Reservoir Development, and the Potential for Hydrocarbon Exploration," presented at the AAPG Annual Convention and Exhibition, Pittsburgh, Pennsylvania, 2013.

[39] G. Isabelle and A. Elisseeff, "An Introduction to Variable and Feature Selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, Mar. 2003

[40] Sudarshan Asaithambi, "Why, How and When to apply Feature Selection," *Towards Data Science*, 31-Jan-2018.

[41] Misra, S., Chakravarty, A., Bhoumick, P., & Rai, C. S. (2019). Unsupervised clustering methods for noninvasive characterization of fracture-induced geomechanical alterations. Machine Learning for Subsurface Characterization, 39

[42] B. Ghojogh *et al.*, "Feature Selection and Feature Extraction in Pattern Analysis: A Literature Review," *arXiv*, 2019.

[43] A. Botchkarev, "A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 14, pp. 45–79, 2019, doi: https://doi.org/10.28945/4184.

[44] F. TabaTaba *et al.*, "A framework for evaluating epidemic forecasts," *BMC Infectious Diseases*, vol. 17, no. 345, 2017, doi: https://doi.org/10.1186/s12879-017-2365-1.

[45] E.A. Freeman and G.G. Moisen, “An application of quantile random forests for predictive mapping of forest attributes,” in Pushing boundaries: new directions in inventory techniques and applications: Forest Inventory Analysis (FIA) symposium 2015, Portland, Oregon.

[46] K. Bogner, F. Pappenberger, and M. Zappa, “Machine Learning Techniques for Predicting the Energy Consumption/Production and its uncertainties Driven by Meteorological Observations and forecasts,” *Sustainability*, vol. 11, no. 3328, doi: https://doi:10.3390/su11123328.

[47] Y. Fang, P. Xu, J. Yang, and Y. Qin, “A quantile regression forest based method to predict drug response and assess prediction reliability,” *PLoS One*, vol. 13, no. 10, Oct. 2018, doi: https://dx.doi.org/10.1371%2Fjournal.pone.0205155.

[48] J. Granath, “Structural Evolution of the Ardmore Basin, Oklahoma: Progressive Deformation in the Foreland of the Ouachita Collision,” *Advancing Earth and Space Science*, vol. 8, no. 5, 1989.

[49] H. Li and S. Misra, “NMT T2 distributions in a shale petroleum sysytem using variational autoencoder-based neural networks,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2395–2397, 2017

[50] S. Misra and J. He, “Stacked Neural Network Architecture to Model the Multi-frequency Conductivity/Permitivitiy responses of Subsurface Shale formations,” in *Machine Learning for Subsurface Characterization*, 1st ed., 2019.

[51] I. Jolliffe and J. Cadima, “Principal Component Analysis: A review and recent developments,” *Philosophical Transactions of the Royal Society A*, vol. 374, no. 2065, 2016.

[52] S. Misra and Y. Wu, “Machine Learning Assisted Segmentation of Scanning Electron Microscopy images of organic-rich shales with Feature Extraction and Feature Ranking,” in *Machine Learning for Subsurface Characterization*, 1st ed., 2019.

[53] S. Misra, O. Osogba, and M. Powers, “Unsupervised Outlier Detection Techniques for Well Logs and Geophysical Data,” in *Machine Learning for Subsurface Characterization*, 1st ed., 2019.