**Title: Evaluating Different Clustering Algorithms for Prediction of Rock Types**

**Autho**rs: Munish Kumara,b and Tan Wen Xuanb

**Affiliations:** aERCE Singapore, 48 Tras Street, #03-01, Singapore 078987

bSingapore University of Social Sciences, School of Business,463 Clementi Rd, Singapore 599494

Corresponding author. Email: mkumar@erce.energy

**Abstract:** Rock typing has multitudes of uses, from optimizing drilling spots , to determining perforation zones, evaluating in-place volumes using static and dynamic models, to better understanding complex flow properties that take place in oil and gas reservoirs. Studies on rock typing have mainly utilized porosity and permeability measurements converted to rock typing indices, with newer studies introducing elements of supervised machine learning as well. However, a comparison of different unsupervised machine learning algorithms for rock typing applications has not been widely researched. This paper aims to perform a comparative study of the performance and outputs from 5 unique unsupervised machine learning models, which we will benchmark against a modified iterative, multi-linear, regression (IMLR) rock typing technique. This study will be conducted on a core data comprising 2000 unique data points from the United Kingdom (UK). The 5 unsupervised machine learning model are the KMeans, Self-Organising Map (SOM) + KMeans, Density-Based Spectral Clustering of Application with Noise (DBSCAN), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Gaussian Mixture Models (GMMs). The study’s results showed that KMeans and BIRCH rock typing performance are the best aside from the modified method when applied on the core dataset.

**One-Sentence Summary:** This paper aims to explore the performance of rock typing using different unsupervised machine learning algorithms against a modified IMLR rock typing method.

**Keywords (minimum 6):** Oil and Gas, Rock Typing, Unsupervised Machine Learning, KMeans, Density-Based Spectral Clustering of Application with Noise (DBSCAN), Self-Organising Map (SOM), Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH), Gaussian Mixture Models (GMMs).

# 1.0 Introduction

Rock typing is based on the simple principle that not all reservoir rocks are created equal. Rocks that are similarly deposited, buried under similar conditions, and undergo similar diagenetic processes will share a set of unique characteristics (Archie, 1950) . In work by Rushing, there are identified 3 different classes of rock typing – depositional, petrographic and hydraulic (Rushing, 2008). Of the 3, hydraulic rock typing (RT) seeks to characterize reservoir rocks into similar groups based on petrophysical properties which (a) quantifies the ability of fluids to flow and (b) determines the availability of pore volume for storage (Hollis et. al., 2020). The physical properties that govern fluid storage and flow are fundamentally linked to the dimension (size), geometry (shape, surface roughness, tortuosity) and distribution of the pore and pore-throat network. Such characterisation is routinely performed using routine core analysis porosity and permeability measurements (Corbett, 2004), but it can also be performed with mercury injection capillary pressure (MICP) as well (Tang, 2020). These unique porosity-permeability relationships and flow characteristics are integrated across multiple scales (nm to km), so that large scale geologic features can be adequately represented from small scale petrology and petrophysical properties. Often, RT at the core scale is done manually, with experienced litho-stratigraphers and experts in core analysis involved in the process of subdividing the data into families, based on depositional and geological principles.

In certain instances, the unique relationships from porosity and permeability are not readily discernable. Assuming stratigraphic continuous rock units have similar reservoir properties, and assuming a capillary tube model as defined by Kozeny-Carman, the concept of flow units/ flow zone index (FZI, mm) was developed to honor the geology while deconvolving the relationship between these 2 variables (Amaefule, Mehmet, Djebbar, Kersey, & Dare, October 1993). Of particular importance is the mean hydraulic radius of pore throats, Rmh, which is used to evaluate the reservoir quality index (RQI, mm), and pore volume to grain volume ratio (also known as the normalized porosity index or fz), defined as:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

where k is permeability (mD) and f is porosity (V/V). For rocks with similar RQI, a graph of (log) FZI vs (log) fz will fall on a straight line with a slope of -1. Rock samples with similar but not identical RQI values will cluster around, but not on this line. Rock samples with significantly different RQI values will fall on other parallel lines with even slope but different

intercepts. The intercept of these lines at fz = 1 defines the mean pore throat size of each linear group of sample points. Numerous authors have derived combinations of these 2 properties to generate variations of FZI (e.g. FZI\*, FZIM and MFZI) either from core data or applied data from well-logs to generate Resistivity Zone Index (RZI) (see Appendix 1 for details).

# Using Machine Learning for RT applications

While RT is one way in which reservoir rocks are classified, the advent of faster computing coupled with advances in memory has encouraged geoscientist to attempt new data driven means to attempt rock classification. Some of the newer rock typing work now utilizes machine learning (ML) to complement or even supplant existing methods as the expectation is that such methods would prove more accurate at anomaly detection and correlative evaluation . Most of the work has however been via the utilization of supervised machine learning which requires expert knowledge to firstly prepare the training sets into fundamental rock types that can be later be applied to the testing data sets. What has been less researched is how unsupervised machine learning algorithms would perform in similar rock classification tasks.

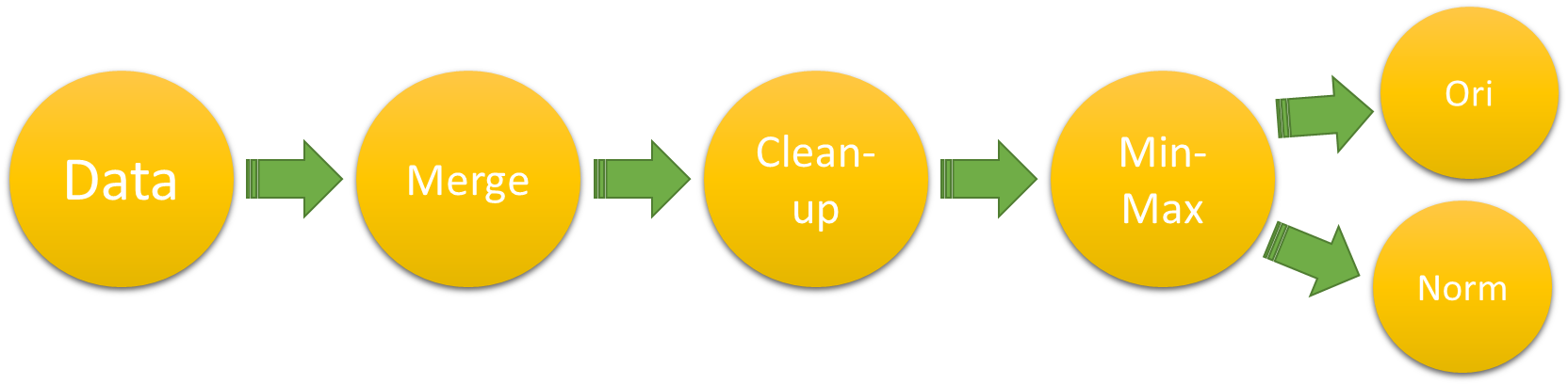
Mohammadian et al (Mohammadian, Kheirollahi, Ostadhassan, & Sabet, 2022) approached the task of rock typing using both supervised and unsupervised algorithms. He started with a data set of 128 carbonate core samples, and firstly applied supervised machine learning (Extreme Gradient Boosting or XGB) to generate an output in the form of a FZIM\*. The second step of this process involved using the FZIM\* output as input into a K-Means algorithm, ultimately resulting in 4 unique RT. A few things to note about this work is that his initial data set was small, he utilized other input variables besides porosity and permeability, like connate water saturation and data from MICP, and he required multiple ML methods to perform the classification. Mohamed et. al. (Mohamed, Mohamed, Mazher, & Pieprzica , 2019) similarly performed rock typing using unsupervised and supervised ML algorithms, with supervised ML classifiers such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest – Decision Tree and one neural network approach. They only used one unsupervised ML method, K-means. Input variables were in the form of log data from 8 wells, but with facies already manually classified prior. They found that there were inherent differences between what was manually classified and what was classified computationally and stated that the unsupervised learning algorithm was the worst performing. Zakyan et. al. (Zakyan, Permadi , & Pratama, 2022) proposed a method which RT log data with artificial neural network (ANN) as an enhanced clay typing methodology. In both the work by (Mohamed, Mohamed, Mazher, & Pieprzica , 2019) and (Zakyan, Permadi , & Pratama, 2022), they data was purely lab based and did not consider core data.

# Scope & Methods

Given the dearth of information regarding unsupervised ML methods for core classification, we were interested in comparing the performance of 5 different unsupervised ML algorithms and benchmark it against the more “conventional” RT method based on the iterative, multi-linear, regression (IMLR) technique. We wanted to use a data source that was heterogeneous and would encompass a wide range of values, while also having sufficient density of data to warrant a ML technique. We decided that we would limit ourselves to one geographic locale (UK North Sea (Data Underground, n.d.), (GeoProvider, n.d.), (UK National Data Repository, n.d.)) and generate a “pseudo-data set” comprising of samples across the multiple sampled wells.

The data was available in an initially unstructured format. Input data files had to filtered, collated and refined before it can be further pre-processed. For instance, there were close to 2000 individual files which needed downloading and compilation. The format of the data present within each file was confirmed through random sampling and was not homogeneous. The file types themselves were not consistent, with ~30% of the files in ASCII format, 25% in comma separated variable format and the remaining in various excel vintages. There were also corrupt files present which needed to sieve out. We observed that some data files had as many as 250 columns, of which a majority were NULL or empty. We found that the most complete of the columns was only the porosity and permeability data, the very minimum of the data types needed for RT. Units were also inconsistent and had to be fixed across the different data files. Doing all this manually would have been extremely time consuming and tedious, with an estimate of >3000 man-hours spent sorting and standardizing the data. Repetitive tasks like this would also likely result in human errors, so we opted for a computational solution which automatically merged the individual data files into a single “mega-merge” (MM) file.

Following the generation of this MM file, we performed a “clean-up” by removing data which was of the NULL data type, alphanumeric or non-physical data (negative numbers or zero division errors). What was remaining was computationally corrected for, using statistical methods where empty data types were filled in with (mean/P10/P50?) values and outliers were removed to prevent data skew. The data is min-max normalized (Kumar, Swaminathan, Rusli, & Thomas-Hy, 2022) before being split into 2 separate csv files, one containing the unnormalized data file and the final input data file containing the normalized data. We schematically illustrate this process in Figure 1.



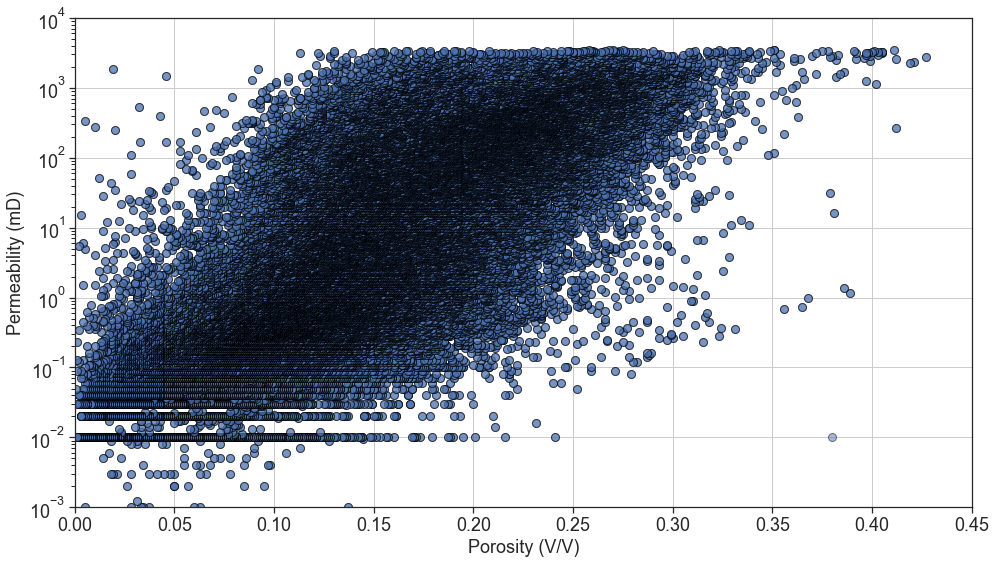
**Figure 1:** Generic data preparation flowchart

## Exploratory Data Analytics and Downsampling

Our final data set was made up of >100,000 unique data points. General histograms of the porosity and permeability of this data set is given in Figure 2. A cross-plot of the data is shown in Figure 3, and a table of descriptive statistics is given in Table 1.

|  |  |
| --- | --- |
|  |  |

**Figure 2:** Histograms depicting (Left) Porosity and (Right) Permeability of the core database



**Figure 3:** Histograms depicting (Left) Porosity and (Right) Permeability of the MM core database

**Table 1:** Descriptive Statistics for the MM Core Database

|  |  |  |
| --- | --- | --- |
| Statistics | Porosity (V/V) | Permeability (mD) |
| mean | 0.149581 | 176.162276 |
| std | 0.066884 | 434.395962 |
| min | 0.000500 | 0.001000 |
| 25% | 0.102000 | 0.370000 |
| 50% | 0.148000 | 5.800000 |
| 75% | 0.198000 | 113.000000 |
| max | 0.427000 | 3430.000000 |

While the data set above satisfies the requirement for sample size and heterogeneity, we are faced with a challenge in that the data itself is too dense to properly perform core classification using a clustering process based purely on porosity and permeability. As stated earlier, RT split by porosity-permeability would be manually done and assisted with core description and facies observation on physical core samples. However, for data sets like this, this would not be possible. For optimal clustering using computational methods, the individual clusters would have to be distinct and separate enough so that distance-based methods (which most clustering algorithms are based on) would work reasonably well. Samples that were repeats or which had values close to one another (highly correlatable to one another) do not necessarily add value to partitioning out of the individual clusters. Another issue with very large data sets like this is the rapidity of the analysis; on data sets like this, running multiple clustering algorithms would be time consuming and computationally expensive.

We therefore attempted to down sample the data, being careful to optimize run time, while ensuring that no critical data was sacrificed in this process. We downsampled via a relatively simple K-Means algorithm, testing sizes at and 100,000+ (full data set) data pointsWe did not vary the hyperparameters, except for the r, which was for all 3 iterationsWe used 4 clusters as an initial first pass for this testFrom our tests, we determined that downsampling to 2000 randomly selected data points is lossless and results in more efficient computation times for the entire process. Shown in Figure 4 is the same 1000 data points selected from all 3 tests (Charts were then plotted to visualize the cluster output for the 3 datasets using the 1000 random samples with same random state.) which shows this approach of down sampling preserves data integrity by displaying similar outcomes but significantly saves on computational resources.

Chart, line chart

Description automatically generated

Fig 6: K-Means Inertia Plot for the 2k, 40k and whole dataset normalized data

|  |  |  |
| --- | --- | --- |
|  |  |  |
| (a) | (b) | (c) |

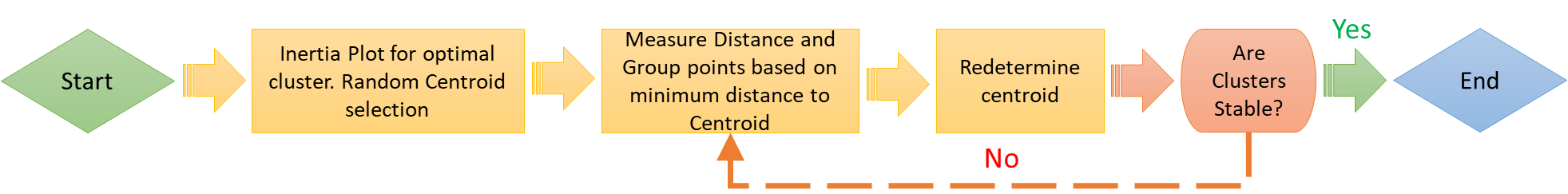
**Figure 4:** 3 different cross-plots of cluttering for (a) 2000, (b) 40000 and (c) 100000+ data points

## ML Algorithm Choices

We select the following unsupervised ML algorithms for our work which are a combination of traditional unsupervised ML models and artificial neural networks (ANN). The ML models selected are heterogeneous, with (a) KMeans (centroid based clustering), (b) Self-Organising Map (SOM) + KMeans (ANN based clustering), (c) Density-Based Spectral Clustering of Application with Noise (DBSCAN) (density based clustering), (d) Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) (hierarchical clustering) and (e) Gaussian Mixture Models (GMMs) (probabilistic based clustering). These algorithms were chosen because XXXXXX

### (A) K-Means

is algorithmbased on principles of Euclidean distance, where observations are split into k clusters, and where each observation is than attributed to the nearest mean or cluster centroidThe algorithm iterates such thatwithin.The shortcoming of the algorithm is that it must be initialized with an initial number of clusters (Sharma, 2022). We schematically illustrate the process of K-means Figure 5



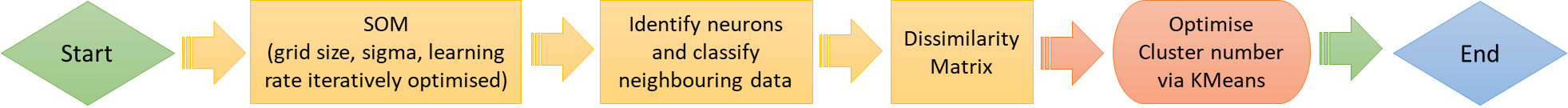
**Figure 5:** Process of K-Means

### (B) Self-Organising Map (SOM) + KMeans

ANNtrained using unsupervised learning, to ultimately produce a low dimension, discretized representation of the input sample. Unlike KMeans, where the centroids (nodes) are free moving and have no relationship to one another, a SOM “feature map” tries to topologically preserve the properties of the input space by "pulling” neighboring nodes along with it, making it ideal for applications in clustering. A optimal SOM is one where the feature space is distorted as minimally as possible. SOMs’ learn by competitively adjusting weights to neutrons. The final output of a SOM map is one where all neuron positions are known and where the calculated relative Euclidean distance is retained between points, with points close to one another being mapped to similar units.

However, SOMs can sometime give large numbers of clusters, necessitating a simplification of sorts by combining these clusters, done with KMeans. The distance between neurons and data can be calculated and organized in a (dissimilarity) matrix, with this matrix as an input for KMeans (Ralhan, 2018) (Brentan, Meirelles, Luvizotto Jr., & Izquierdo, 2018).

For this work, wutilized the“”to construct the SOM ,Shown bin Figure 6 simplifiedschematic



**Figure 6:** Process of SOM + K-Means

### (C) Density-Based Spatial Clustering of Application with Noise (DBSCAN)

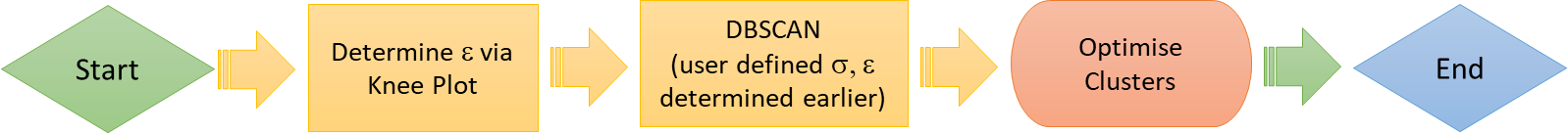
DBSCAN is a type of density-based clustering which groups data points together based on a certain defined maximum distance between two points, known as e. Those that are sufficiently close together belong to a single cluster and those that don’t are treated as belonging to be separate cluster. The two parameters governing the separation of clusters are epsilon (e), defined as the maximum distance between points allowed to be considered as part of a cluster, and minimum samples (s), defined as the minimum points that are required for a formation of a cluster. Both e and s are user defined, but while the choice of s is somewhat arbitrary, e is constrained using a knee plot. We describe the knee plot in XXXX, and schematically illustrate the DBSCAN process in Figure 7 (Kumar, 2021).

The nearest neighbor of 4 for all data points was used to find the average distance between each point from its 4 nearest neighbors. The average distance of the points was plotted in ascending order and the point of the greatest bend is used as the epsilon value (Kumar, 2021). The knee is the point of the greatest bend to be used as the epsilon value for DBSCAN. Fig 12 shows the knee point of the dataset to and Fig 13 shows the process flow of DBSCAN.

A picture containing chart

Description automatically generated

Fig 12: Elbow Plot with Knee Point Identified by Kneed for Epsilon Value



**Figure 7:** Process of DBSCAN

### (D) Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)

BIRCH is a hierarchical clustering algorithm that works on clustering data that is a compact summary of itself, rather than on the actual larger dataset, always staying true to the distribution of the original data. BIRCH firstly summaries the information via Clustering Feature (CF) entries, where each entry contains information related to (i) the number of data points in the cluster (N), (ii) the linear sum of the N data points and (iii) the sum squared of the N data points. Subclusters are formed by the addition of multiple CFs.

A CF tree is thus the ultimate compact representation of the data set, where each leaf node is a subcluster. The three parameters needed are threshold, branching factor, and n \_cluster. The threshold represents the number of data points leaf node sub-clusters can have in a CF tree, the branching factor represents how many sub-clusters can a leaf node have and the n\_clusters represent how many clusters we require (Verma, 2021).

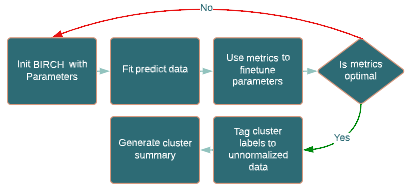


Fig 14: Process of BIRCH Method

### (E) Gaussian Mixture Models (GMMs)

GMMs assume that the data consists of several Gaussian distributions and for each data, the points are assigned to a cluster based on probabilistic methods. For example, a data point that lies between two clusters has a different probability of being in either one using the GMMs. It assigns the data to the cluster it most likely belongs to. GMMs work like K-Means but the clusters can be non-circular, unlike K-Means. The initialization is selected as K-Means and the number of components is determined using Bayesian Information Criterion (BIC) (Singh, 2022). BIC is a model selection criterion where lower BIC means the model has lower penalty terms and thus the model is better (Datalab, 2019). The penalty term relates to number of parameters (Datalab, 2019). Fig 15 shows the optimal number components using the BIC plot and Fig 16 shows the process flow for GMMs method.

Chart, line chart

Description automatically generated

Fig 15: BIC Plot for Optimal Number of Components

Diagram

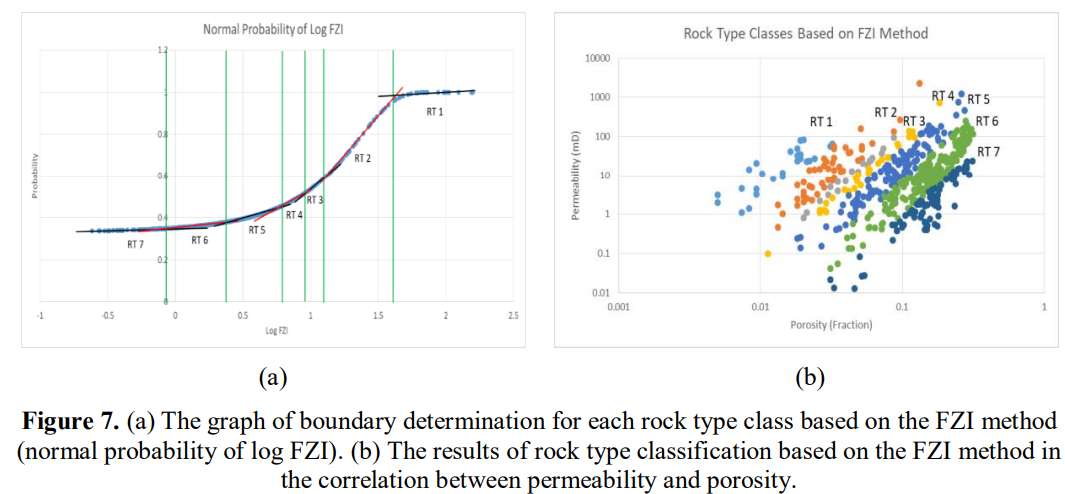
Description automatically generated

Fig 16: Process of Gaussian Mixture Model Method

## Conventional Rock Typing using Iterative Multi-Linear Regression (IMLR)

We applied an IMLR technique as the benchmark “conventional” RT method based on a review of literature which described it as a fairly accurate and robust method (Khalid, 2020; Al-Ajmi, 2000; Desouky, 2003). We, however, modified the technique slightly to work on our data set, which in and of itself did not show a clear separation into district clusters if viewed on a (log) fz vs (log) RQI scale .

Fig 8: Log Porosity Index vs Log RQI Scatterplot of the Normalized Training Dataset



Our modification involved (i) first evaluating the FZI for the entire data set, (ii) sorting in ascending order the data by FZI value, (iii) training, testing and predicting on the data a linear regression (LR) and recording the value, (iv) splitting the sorted data into 2 subsets, (iv) training, testing and predicting on each subset of data a linear regression (LR), (v) evaluating an average r-squared (r2) for all the subsets and (vi) repeating steps (iv) to (vi) with every increasing subsets to a maximum value of 10. We than plot a graph of r2 vs number of clusters, and select the optimal cluster based on the highest r2 value. We than apply the optimized cluster number back to the entire data set and label the input data accordingly. Shown in Fig XX is a schematic of the modified IMLR and shown in Fig XX is the output of the classification process.

Diagram

Description automatically generated

Fig 9: Process of the Modified Rock Typing Method

# Results and Discussion

## Comparing the Number of Hydraulic Rock Types generated.

We first compare the results of the number of unique RT generated by the ML models versus the IMLR model. Shown in Fig XX is the distribution plot of the various outputs. Right away, we note that the outputs from the various algorithms plots look dissimilar. We noted that while the conventional RT method have 6 unique RT, none of the other ML methods gave more than 4 unique clusters.

The results displayed below for the scatterplots show that the 5 cluster methods cluster grouping on the scatterplot do not resemble the modified method. As for the cluster average table comparisons, the cluster averages of each method do not resemble one another.

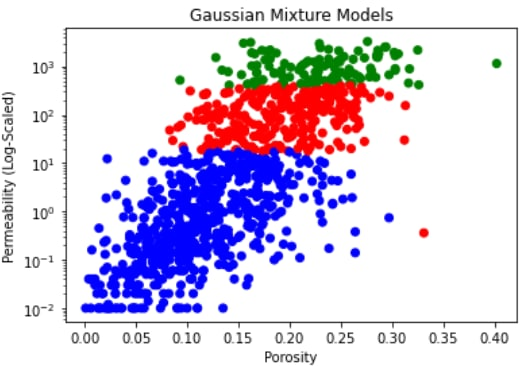
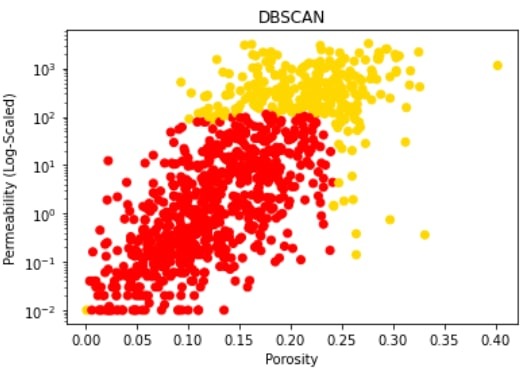
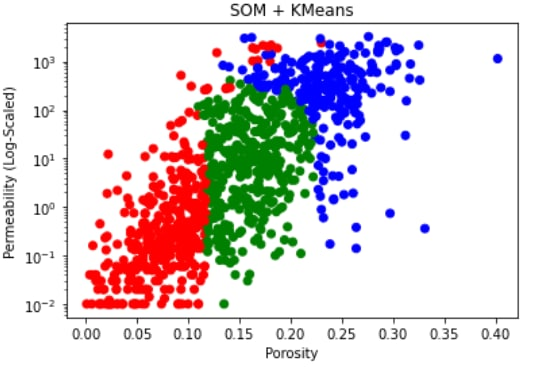
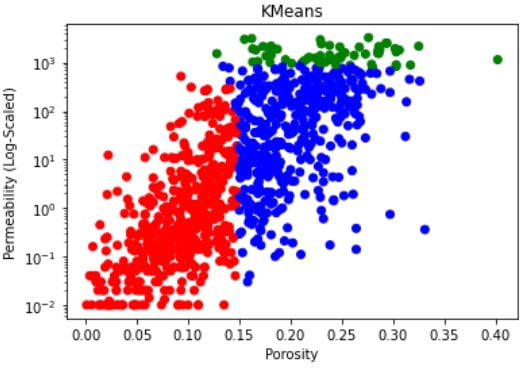
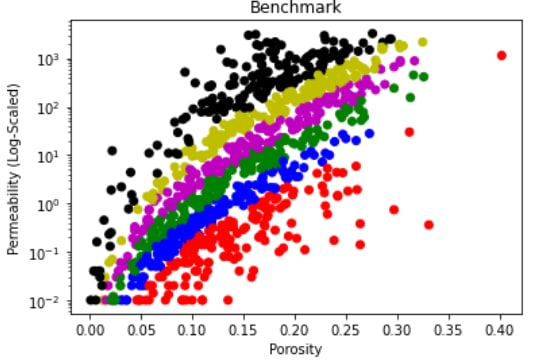
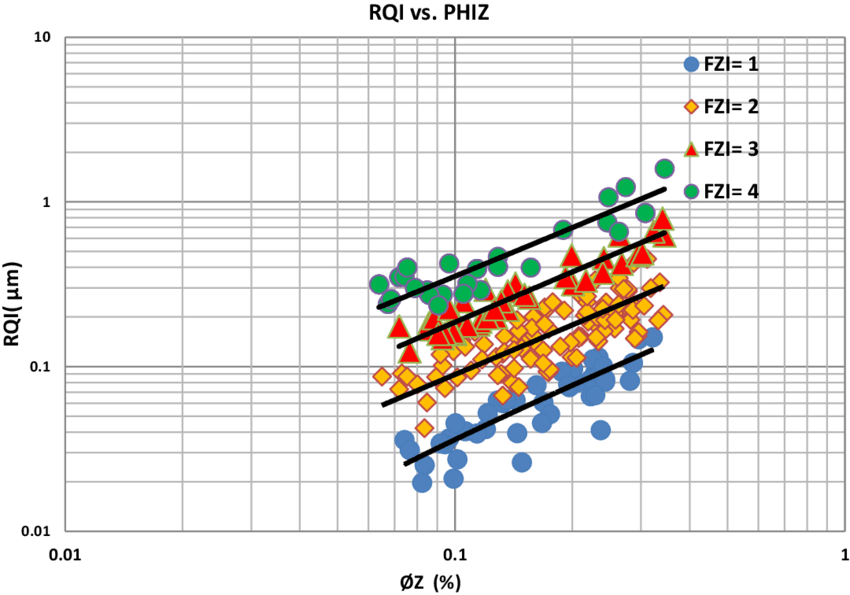
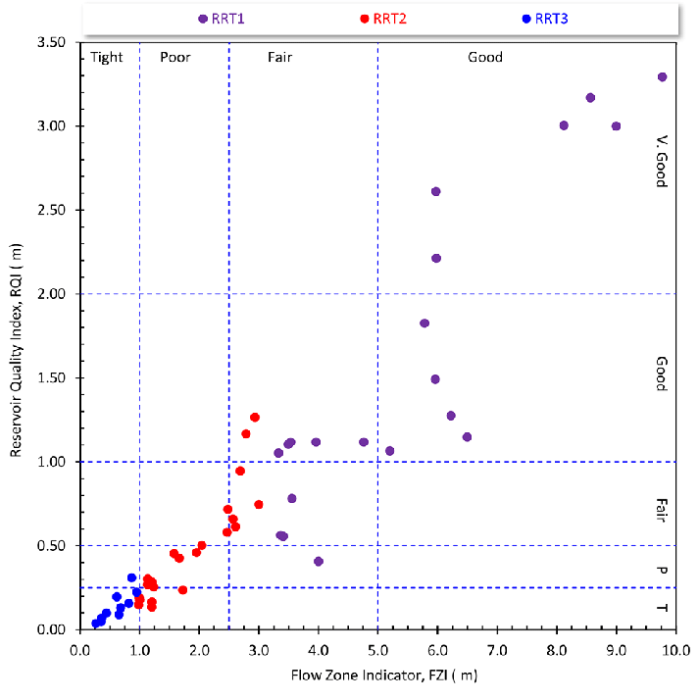
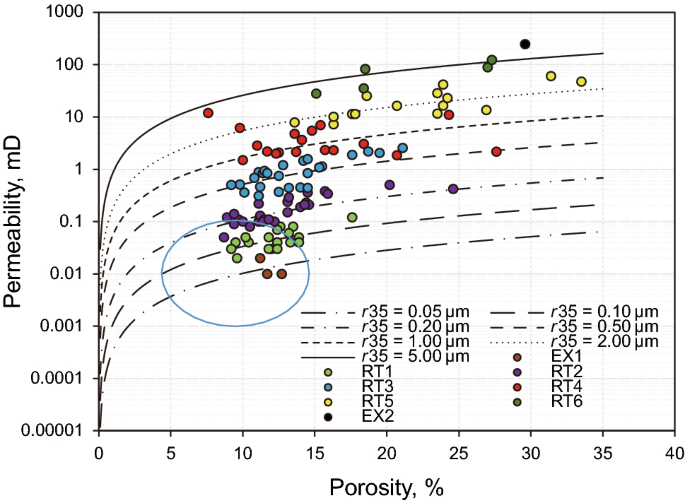


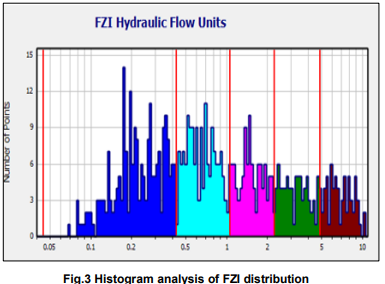
Fig 17: Porosity vs Log Scaled Permeability Scatterplot Cluster Distributions

Just curious but have you tried plotting your results in a different way e.g. normalized porosity vs RQI, FZI vs RQI, FZI vs Kcor etc. Some examples below

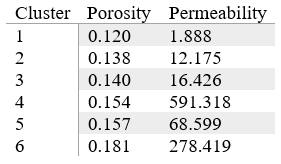
Chart, scatter chart

Description automatically generated

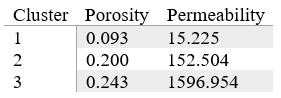
 Chart, line chart

Description automatically generated

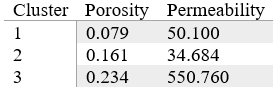
Benchmark



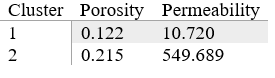
KMeans



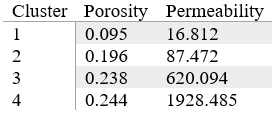
SOM + KMeans



DBSCAN



BIRCH



Gaussian Mixture Models

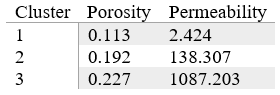


Table 1: Cluster Porosity and Permeability Averages

## Comparing Residuals between Model Outcomes

For our second method, we extract 50 samples from our original 2000 data points and determine how well we are able to predict permeability using regressors that define each cluster type.

For this approach of evaluation, we did permeability prediction, then calculate the residuals to get the residual models for each of the 6 methods. Permeability prediction is one of the applications that are related to rock typing and has been used in the field for a long time. For this evaluation, the purpose was to compare the residual models for all 6 methods that are obtained by permeability prediction which can show us the sample data deviates from each of the models. This can give us some insight into the performance of each model for rock typing.

For the benchmark method, using the sample sets’ porosity, the log porosity index values were calculated and used to predict the log RQI by using the linear regression prediction depending on the FZI of the sample which affects which linear regression model to be used for prediction as our benchmark method when training the model, splits the dataset into equal parts depending on the iteration number. As the FZI is already sorted in the data preparation stage, the linear regression fitted HFU is based on a specific range of non-overlapping FZI values. The sample data points FZI is compared against the ranges for the 6 FZI ranges that were determined when training the model and the corresponding linear regression would be used to predict log RQI of the sample data point. The predicted RQI can then be worked out from the predicted RQI and the predicted permeability can then be calculated by performing some calculations. The residuals are then obtained by having the actual sample permeabilities minus the predicted permeabilities. A scatterplot with predicted permeability on the x-axis and residual on the y-axis with a dashed horizontal line at point 0 is plotted. This process is repeated for all 6 methods. Some of the methods have a simple approach to calculating the predicted permeability by simply using the predict function like the KMeans algorithm using the sci-kit learn KMeans. However, most of the methods do not have such prebuilt functions. What was done is very similar to the process of getting the predicted permeability and residuals for the benchmark method. In short, each cluster is fitted with a linear regression that will be used for permeability prediction for the samples. The sample’s predicted cluster is first obtained for the sample set data points. The samples’ predicted permeability is then obtained by predicting using the different cluster’s linear regression prediction method. The residuals can then be derived, and the residual model plotted. The residual models for all 6 models are then plotted and will be shown below.

[still using lr rather than power functions, difficulty in plotting them]

[pending new charts]

## Comparing Metrics between the various Unsupervised Machine Learning Algorithms

Our final method of comparison uses ML metrics such as the Silhouette Coefficient, Calinski-Harabaz Index, and the Davis-Bouldin Index to compare between the different ML models. These metrics evaluate the uniqueness of clusters.

For the Silhouette Coefficient and the Calinski-Harabaz index, the higher values are better. For the David-Bouldin Index, the lower values are better. The required inputs needed are the dataset and labels, of which the unsupervised algorithms used have implementations retrieve the labels. Below is an overview of all 3 metrics on the 5 unsupervised machine learning algorithms with values rounded to 3 decimal points.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Silhouette Coefficient | Calinski-Harabaz Index | Davis-Bouldin Index |
| KMeans | 0.529 | 2556.740 | 0.642 |
| SOM+KMeans | 0.412 | 1505.616 | 0.859 |
| DBSCAN | 0.404 | 1180.614 | 1.041 |
| BIRCH | 0.474 | 2170.326 | 0.839 |
| GMMs | 0.263 | 1090.622 | 1.032 |

Table 2: Summary of the 5 Clustering Models’ Metrices

Based on Table 2, looking on an individual level for the silhouette coefficient, KMeans have the highest score, for the Calinski-Harabaz Index, KMeans and BIRCH have high scores and for the Davis-Bouldin Index, KMeans and BIRCH the lowest relative scores. From an overall viewpoint, both KMeans and BIRCH have good scorings based on the 3 metrics and have been shown to outperform the other algorithms.

# Limitations of Study and Conclusion

In this study, we performed rock typing using our implementation of the traditional rock typing method as well as 5 other unsupervised machine learning algorithms, namely KMeans, SOM+KMeans, DBSCAN, BIRCH, and GMM. The scaled-down dataset used for training consists of 2000 samples of the UK core data with porosity and permeability data which is the petrophysical properties that are typically used for rock typing in the field. The own implementation of the traditional method is derived by closely studying several studies. It acts as a benchmark model for us to compare the other models. We then compared all 6 trained models with optimized parameters using 3 different approaches.

When comparing the cluster distributions in the porosity vs log-scaled permeability scatterplots and the cluster averages of the 6 models, we observe that none of the 5 models' cluster distributions and cluster averages resemble the benchmark method results. KMeans and BIRCH cluster distribution and cluster averages are close to one another.

The 2nd approach to the evaluation was to perform permeability prediction using a sample set of 50 samples and plot residual models for comparison. The model that has less spread from the residual = 0 point would mean that the model fits the data best. Based on the comparison of the residual models, KMeans and BIRCH models have a similar residual spread to the benchmark model with the other models showing poor performance when used for permeability prediction.

The 3rd approach compares the Silhouette Coefficient, Calinski-Harabaz Index, and the Davis-Bouldin Index metrics for the 5 unsupervised machine learning algorithms. Overall, KMeans and BIRCH are the best-performing algorithms based on 3 metrics.

Overall, when comparing the 5 unsupervised machine learning models to the benchmark model, KMeans and BIRCH show to be able to cluster the rock types the best and the performance is comparable to the performance of the benchmark model. Although the clustering algorithm models cluster distribution is not like the benchmark models’, it is still able to be used as a method of rock typing. Using unsupervised machine learning algorithms would bring about time savings and efficiency while not compromising the quality and accuracy of the clustering of rock types. However, this is not to say to solely replace the expertise of experts, data is only useful depending on who is using it. Experts will still need to take over after training the models to make sense of the results. This would bring us to progress in the oil and gas rock typing area.

The study only used the limited dataset of UK core data. There is room for improvement by including more algorithms, using more datasets to account for variability, and benchmarking on multiple traditional methods of rock typing.

# Bibliography

Amaefule, J. O., Mehmet, A., Djebbar, T., Kersey, D. G., & Dare, K. K. (October 1993). Enhanced Reservoir Description: Using Core and Log Data to Identify Hydraulic (Flow) Units and Predict Permeability in Uncored Intervals/Wells. *SPE Annual Technical Conference and Exhibition*, 1-12.

Archie, G. (1950). Introduction to Petrophysics of Reservoir Rocks. *AAPG Bulletin, 34*, 943-961.

Brentan, B., Meirelles, G., Luvizotto Jr., E., & Izquierdo, J. (2018). Hybrid SOMþk-Means clustering to improve planning, operation and management in water distribution systems. *Environmental Modelling & Software*, 1-12.

Corbett, P. W. (2004). Petrotyping: A basemap and atlas for navigating through permeability and porosity data for reservoir comparison and permeability prediction. *Paper SCA2004-30 presented at the International Symposium of the Society of Core Analysts, 5*(9), 1-12.

Kumar, M., Swaminathan, K., Rusli, A., & Thomas-Hy, A. (2022). Applying Data Analytics & Machine Learning Methods for Recovery Factor Prediction and Uncertainty Modelling. *Society of Petroleum Engineers*, 1-18.

Mohamed, I. M., Mohamed, S., Mazher, I., & Pieprzica , C. (2019). Formation Lithology Classification: Insights into Machine Learning Methods. *SPE Annual Technical Conference and Exhibition*, 1-12.

Mohammadian, E., Kheirollahi, M., Ostadhassan, M., & Sabet, M. (2022). A case study of petrophysical rock typing and permeability prediction using machine learning in a heterogenous carbonate reservoir in Iran. *Nature Portfolio, 12*(4505), 1-16.

Rushing, J. A. (2008). Rock Typing — Keys to Understanding Productivity in Tight Gas Sands. *Society of Petroleum Engineers (SPE) Paper Number 114164*, 1-31.

Tang, X. Z. (2020). An improved method in petrophysical rock typing based on mercury-injection capillary pressure data. . *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 1-16.

Zakyan, H., Permadi , A. K., & Pratama, E. A. (2022). An Improved Method of Clay-Induced Rock Typing Derived from Log Data in Modelling Low Salinity Water Injection: A Case Study on an Oil Field in Indonesia. *Energies, 15*(10), 1-12.

**References**

Amaefule, J. O., Altunbay, M., Tiab, D., Kersey, D. G., & Keelan, D. K. (1993). *Enhanced Reservoir Hydraulic (Flow) Units and Predict Permeability in Uncored Intervals/Wells* [Paper Presentation]. Paper presented at the SPE Annual Technical Conference and Exhibition, Houston, Texas, October 1993. https://doi.org/10.2523/26436-ms

Datalab, A. (2019, January 16). *What is Bayesian Information Criterion (BIC)?* Medium. Retrieved from https://medium.com/@analyttica/what-is-bayesian-information-criterion-bic-b3396a894be6

Izadi, M. & Ghalambor, A. (2013). A new approach in permeability and hydraulic-flow-unit determination. *SPE Reserv. Eval. Eng.* **16**(03), 257–264.

Kolodzie, S. (1980). *Analysis of pore throat size and use of the Waxman–Smits equation to determine OOIP in Spindle Field, Colorado* [Paper Presentation]. Paper presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, September 1980. https://doi.org/10.2118/9382-MS

Mirzaei-Paiaman, A., Ostadhassan, M., Rezaee, R., Saboorian-Jooybari, H. & Chen, Z. (2018). A new approach in petrophysical rock typing. *Journal of Petroleum Science and Engineering,* **166**, 445–464. https://doi.org/10.1016/j.petrol.2018.03.075

Nooruddin, H. A. & Hossain, M. E. (2011). Modified Kozeny–Carmen correlation for enhanced hydraulic flow unit characterization. *Journal of Petroleum Science and Engineering,* **80**(1), 107–115. https://doi.org/10.1016/j.petrol.2011.11.003

**Appendix 1 – Rock Typing Indices Equations**

|  |  |  |  |
| --- | --- | --- | --- |
| Authors | Indices | Equations | Equation number |
| (Amaefule et al, 1993) |  | Where:  k is permeability (mD)  f is porosity(V/V)  fz is normalized porosity(V/V)  RQI is Rock Quality Index (mm)  FZI is Flow Zone Index (mm) |  |
| (Mirzaei-Paiaman et al, 2018) | FZI\* | Where:  k -> permeability  -> porosity |  |
| (Nooruddin and Hossain, 2011) | FZIM | Where:  m -> cementation factor |  |
| (Izadi and Ghalambor, 2013) | MFZI | Where:  -> irreducible water saturation  -> Rock Quality Index |  |
| (Shahat et. al.,2021) | RFZI | Where:  -> resistivity index  -> true formation resistivity  -> formation resistivity when water saturation is 100%  -> formation water resistivity  -> surface grain volume  k -> permeability  -> shape factor |  |
| (Kolodzie, 1980) | Winland | Where:  -> uncorrected air permeability  -> porosity  r35 -> pore throat radius with 35% mercury saturation |  |