**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 studied. This paper aims to perform a comparative study of the performance and outputs from 5 different 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 K-Means, Self-Organising Map (SOM) + K-Means, 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 K-Means and BIRCH rock typing performance are the best performing 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, K-Means, 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).

# 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 [1] . In work by Rushing, there are identified 3 different classes of rock typing – depositional, petrographic, and hydraulic [2]. 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 [3]. 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 [4], but it can also be performed with mercury injection capillary pressure (MICP) as well [5]. 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 [6], the concept of flow units/ flow zone index (FZI, mm) was developed to honor the geology while deconvolving the relationship between these 2 variables [6]. 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).

The clustering of FZI that are close in proximity will determine the number of rock types (hydraulic flow units (HFUs)) present in the core dataset. Several approaches are used to cluster core data into HFUs with iterative multi-linear regression (IMLR) and analysis by least square regression being the more accurate methods [7] [8]. The many approaches, however, all aim to do something similar i.e. using FZI to form distinct groups, for determination of the number of rock types or hydraulic flow units there are in the data.

Another approach uses a iterative multi-linear regression (IMLR) technique. The method starts off with a (log) fz vs (log) RQI plot of the data. Initial guess of straight lines with slope = 1 are plotted based on the chart. The datapoints are then allocated to the nearest line. The line intercepts are then recalculated using the least-square regression. The values are then compared the old intercepts to see if the variance is within acceptance range. The process repeats until the variance is acceptable for all lines of the plot.

# 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.

Mohamed et. al [9] 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.

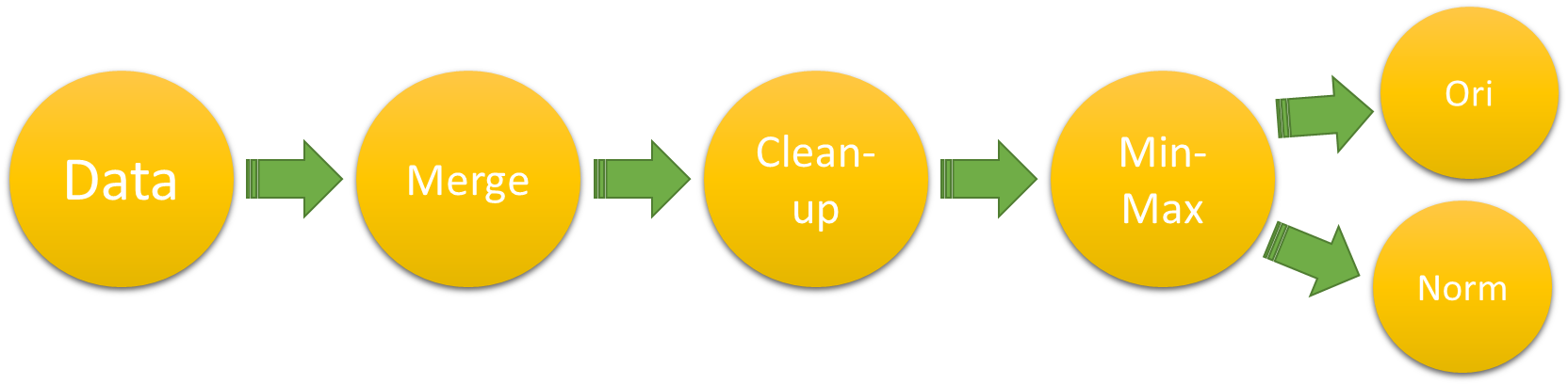
Mohamed et. al. [10] similarly performed rock typing using unsupervised and supervised ML algorithms, with supervised ML classifiers such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest – Decision Tree and one neural network approach. They only used one unsupervised ML method, 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. [11] proposed a method for RT of log data with artificial neural network (ANN) [12]. HFUs were derived using a back propagation technique based on 35 samples from routine core analysis (RCAL). The output is that there were no missed classifications using the proposed methodology on the 35 samples and that it outperforms the multi-linear regression method. In both the work by Mohamed et. al. [10] and Zakyan et. al. [11], the data was mostly log based data and did not use much core data. Additionally, their described methodology was multi step, making it difficult for general application. Mohamed et. al. additionally required that the data be labelled before applying his method.

# Scope & Methods

Given the lack 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, which is 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 utilize open source data from the UK North Sea for this work [13], [14], [15].

The data was available in an initially unstructured format. Input data files had to filtered, collated, and refined before they could be further pre-processed. For instance, there were close to 2000 individual files which needed collation. The format of the data present within each file was not homogeneous (confirmed through random sampling). The file types themselves were not consistent, with 43% in excel format, 39% in TIFF format, 14% in PDF format and 4% in other formats including jpeg and ASCII formats. From the excel data, only 33% was relevant for use. There were also corrupt files present which needed removal. 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. 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 using the k-Nearest Neighbor Imputation method. The method replaces the missing values using the nearest neighbors average values and outliers were removed to prevent data skew. The data is min-max normalized to shrink the feature to the same scale of 0 to 1 and to reduce the space between data points for distance based algorithms to work well [16] before being split into 2 separate csv files, one containing the unnormalized data file (Ori) and the final input data file containing the normalized data (Norm). We schematically illustrate this process in Figure 1.



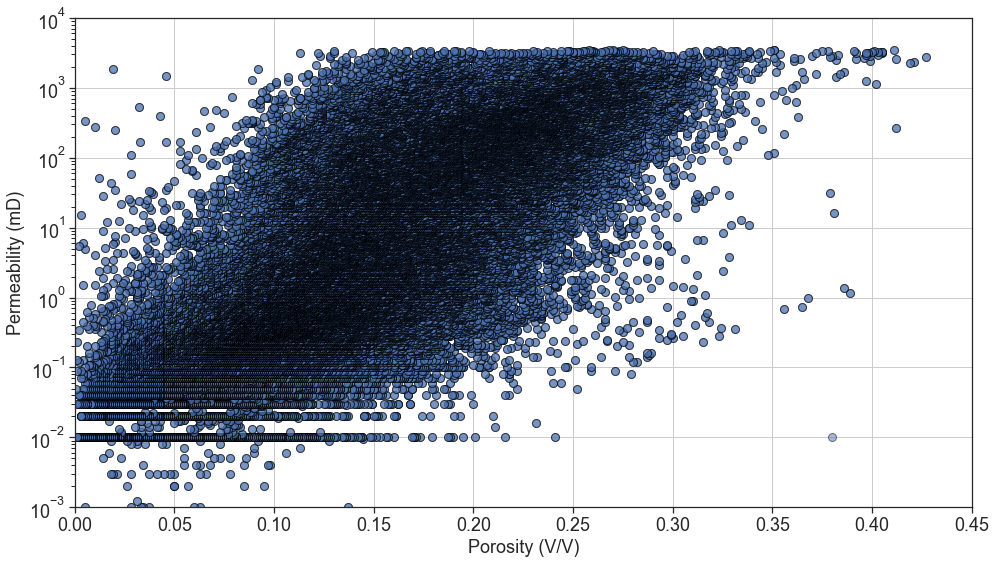
**Figure 1:** Generic data preparation flowchart

## Exploratory Data Analytics and Down sampling

Our final data set was made up of >100,000 unique data points, from sandstone reservoirs. General histograms of the porosity and permeability of this data set is given in Figure 2. This data was collected from a total of estimated 50+ unique wells with depth range of about 400m - 18,000m. 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:** Porosity and Permeability Cross-plot of Whole Dataset

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Property | mean | std | min | 25% | 50% | 75% |
| Porosity (V/V) | 0.15 | 0.067 | 0.001 | 0.102 | 0.148 | 0.198 |
| Permeability (mD) | 176 | 434 | 0.001 | 0.37 | 5.8 | 113 |



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 corelatable 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 did this using a downsampling technique based on a relatively simple K-Means algorithm, where we tested 3 separate sized dataset sizes of 2,000, 40,000 and 100,000+ (full data set) data points. We did not vary the hyperparameters, except for the random state, which we kept fixed in all 3 cases. We also did not optimize the number of clusters and arbitrarily selected 4 clusters. A table of Mean Squared Errors (MSE) for the 4 clusters of each dataset size was our basis for comparison, where, if MSE did not vary by more than XX%, we deemed that the downsampling was lossless. The results of this process is provided in Table 2. Looking at the MSE, given that the dataset sizes are very different the MSE across the 3 dataset is reasonable.

**Table 2:** MSE for the 3 datasets for all clusters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sample Size | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 |
| MSE | MSE | MSE | MSE |
| 100,000 | 3590 | 133000 | 32000 | 1160 |
| 40,000 | 3800 | 134000 | 30800 | 841 |
| Delta |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sample Size | Cluster 1 | Cluster 2 | Cluster 3 | Cluster 4 |
| MSE | MSE | MSE | MSE |
| 40,000 | 3800 | 134000 | 30800 | 841 |
| 2,000 | 7350 | 115000 | 15900 | 1670 |

From our tests, we determined that down sampling to 2000 randomly selected data points is lossless and results in more efficient computation times for the entire process. A table of execution time is given in Table 3. From the comparison in the table, we note a 91.5% improvement in running the downsampled data set, as compared to the full dataset

**Table 3:** Time Taken to run K-Means for the different dataset sizes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Datasets | Time Taken (ms) | | | | Run time Improvement |
| Trial 1 | Trial 2 | Trial 3 | Average |  |
| 2,000 | 108 | 104 | 106 | 106 | 91.5% |
| 40,000 | 547 | 491 | 515 | 518 | 58.2% |
| 100,000+ | 1290 | 1180 | 1260 | 1240 | - |

Given in Figure 4 is an overlay of a random sampling of 1000 data points, with solid filled points represents the outcome of the clustering algorithm where all 3 dataset are identical, while semi-translucent points are data points which have been reassigned to a new cluster value as a result of the downsampling. Overall, we observe <10% of the points are reassigned. We therefore consider the downsampling approach to be robust, as it preserves data integrity while saving on computational resources.

Chart, scatter chart

Description automatically generated

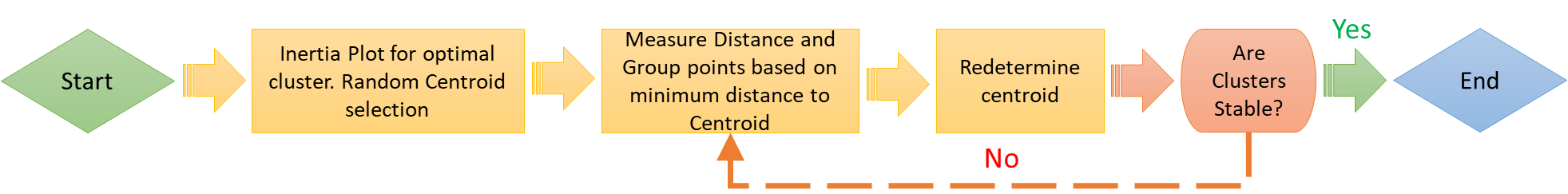
**Figure 4:** Porosity and Permeability Cross-plot of the 3 Datasets Clustering Results

## 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) K-Means (centroid based clustering), (b) Self-Organising Map (SOM) + K-Means (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 due to the popularity of the algorithms as well as them being in different families of clustering algorithms.

### (A) K-Means

K-Means is one of the most widely used clustering methods. It is a centroid-based clustering algorithm based 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 centroid. The algorithm iterates such that the center point will shift to be the average of all the data points within that cluster. The shortcoming of the algorithm is that it must be initialized with an initial number of clusters [17]. We schematically illustrate the process of K-means model training in Figure 5.



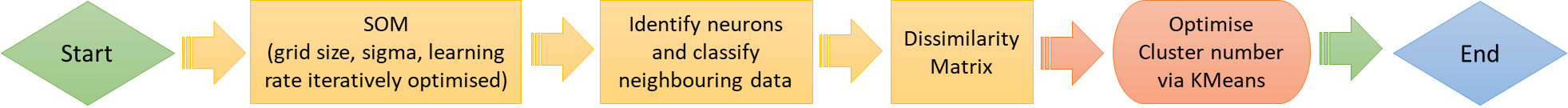
**Figure 5:** Process of K-Means

### (B) Self-Organising Map (SOM) + K-Means

A Self-Organizing Map (SOM) is an ANN trained using unsupervised learning, to ultimately produce a low dimension, discretized representation of the input sample. Unlike K-Means, 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. An 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 K-Means. The distance between neurons and data can be calculated and organized in a (dissimilarity) matrix, with this matrix as an input for K-Means [18].

For this work, we utilized the “MiniSOM” library to construct the SOM. We initialized a large SOM grid of 10X10 with default sigma and learning rates, before applying K-Means to get a smaller number of clusters. The sigma parameter refers to the area of the circle of each node [19]. The optimal cluster was decided with the inertia plot. Shown below in Figure 6 is a simplified schematic of the process.



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

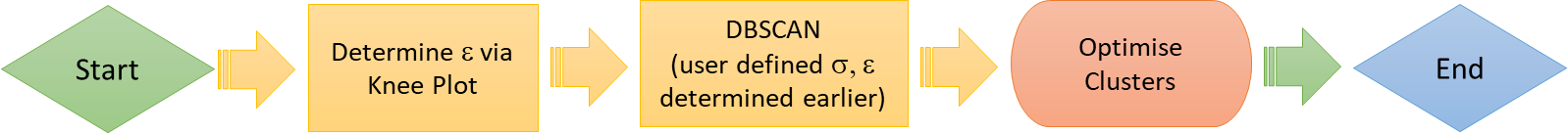
### (C) Density-Based Spatial Clustering of Application with Noise (DBSCAN)

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 (Figure 7). The maximum point of inflexion (or the knee) occurs when the average distance between a point and its s nearest neighbors is the largest. In this work, we set s = 4. maximum inflexion isby; therefore e = is applied for the DBSCAN process, schematically illustrate in Figure 8 [20].

A picture containing chart

Description automatically generated

**Figure 7:** Average distance of point to nearest 4 neighbors plot (ascending) with knee point



**Figure 8:** Process of DBSCAN

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

BIRCH is a hierarchical clustering algorithm that builds an accurate summarized compact version of the original data for clustering. The algorithm builds a CF-tree where datapoints close in proximity forms a subcluster and outlier datapoints are removed. A CF tree is the compact representation of the data set, where each leaf node is a subcluster. The three parameters needed are threshold, branching factor, and n \_cluster which are user-defined. The threshold represents the number of data point each leaf node can have, the branching factor represents how many sub-clusters per leaf node and the n\_clusters represent how many clusters we require. The shortcoming of the algorithm is that it works only for metric attributes [22]. We schematically illustrate the process of K-means model training in Figure 9.

A picture containing funnel chart

Description automatically generated

**Figure 9:** Process of BIRCH

### (E) Gaussian Mixture Model (GMM)

GMMs is a probabilistic based clustering technique that clusters data into ellipsoidal shaped groups based on probabilistic methods, with each cluster distributed in a Gaussian normal distribution [23]. The parameter needed is the number of components. The number of components is determined using Bayesian Information Criterion (BIC). The shortcoming of the algorithm is the assumption of the dataset having a Gaussian distribution apriori, and that the cluster are ellipsoidal in shape. Figure 10 shows the BIC plot where like the Knee plot in Figure 7, the point of maximum inflection is of interest. We illustrate the process of Gaussian Mixture Models training in Figure 11.

Chart, line chart

Description automatically generated

**Figure 10:** BIC Plot for Number of Components Selection

****

**Figure 11:** Process of Gaussian Mixture Models

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

We applied the IMLR technique as the benchmark “conventional” RT method based on a review of literature [7, 24, 25]. Figure 12 demonstrates how IMLR RTs are generated – it utilizes the gradient changes from a probability density plot of log FZI (Figure 12(L)) to determine how best to segment the plot of Log fz vs Log RQI (Figure 12(R)). We note with interest that the example in Figure 12(R) does not show clear did not show a clear separation into distinct clusters if viewed on a (log) fz vs (log) RQI scale. This is something we observed in our data set as well (Figure 3).

To perform our IMLR, we apply a ML method as well. We (i) first evaluate the FZI for the entire data set, (ii) sort in ascending order the data by FZI value, (iii) train, test and predict on the data a linear regression (LR) and record the value, (iv) split the sorted data into 2 subsets, (iv) train, test and predict on each subset of data a linear regression (LR) again, (v) evaluate an average r-squared (r2) for all the subsets and (vi) repeating steps (iv) to (vi) with ever increasing subsets to a maximum value of 10. rate of change of r2 . Given in Figure 13 is the r2 plot, Figure 14 is a schematic of the IMLR method we applied and Figure 15 is the output of the classification process.

Chart, scatter chart

Description automatically generated

RQI

Chart, line chart

Description automatically generated

**Figure 13:** r2 plot for number of cluster selection

Chart, funnel chart

Description automatically generated

**Figure 14:** Process of IMLR



**Figure 15:** Graph depicting rock type class based IMLR chart (b) The rock type classification viewed on a log RQI and log porosity index chart

# 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 Table 4 is the overview of the number of clusters for each method and the metrics used to determine optimal cluster size. Shown in Table 5 is the breakdown of the cluster distribution of the 6 methods. We consistently observe that, across the 5 unsupervised methods, 35% of the data set is labelled as cluster 1, while 2% is labelled as cluster 3. There is no commonality in points labelled as cluster 2, 4 or 5.

Shown in Figure 16 and Figure 17 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 has 6 unique RTs, none of the other ML methods gave more than 4 unique clusters.

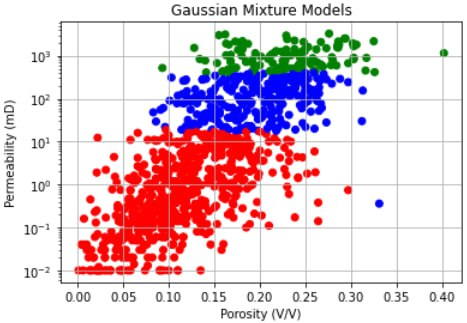
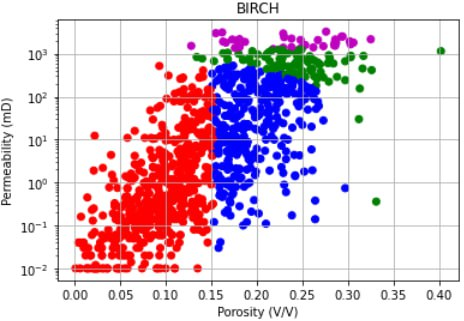
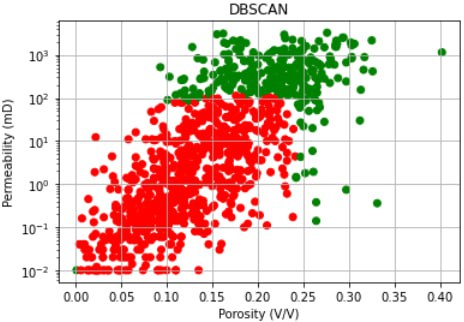
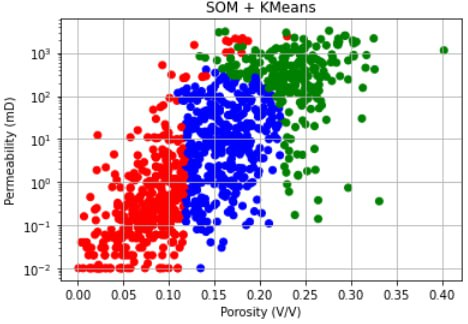
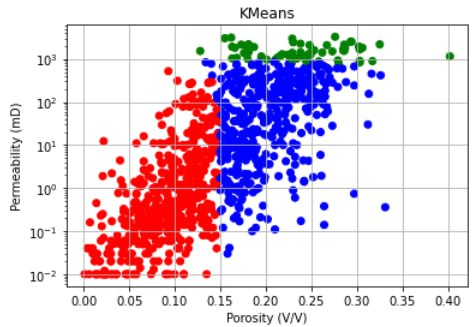
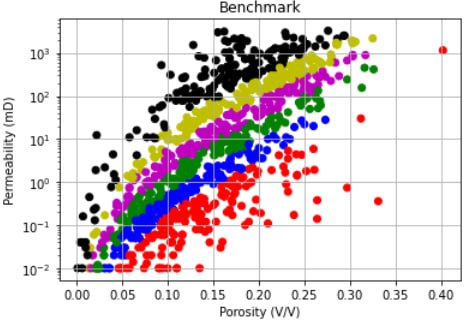
**Table 4:** Number of clusters and metrics for cluster sections per method



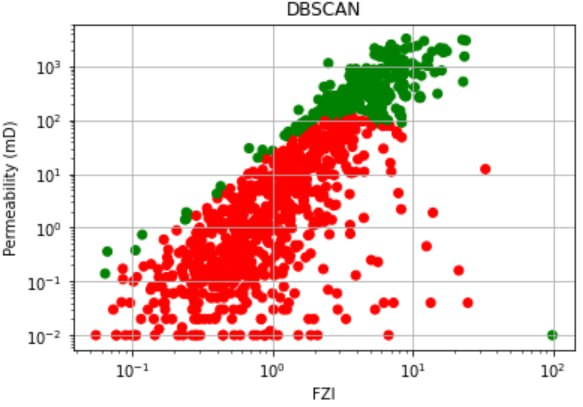
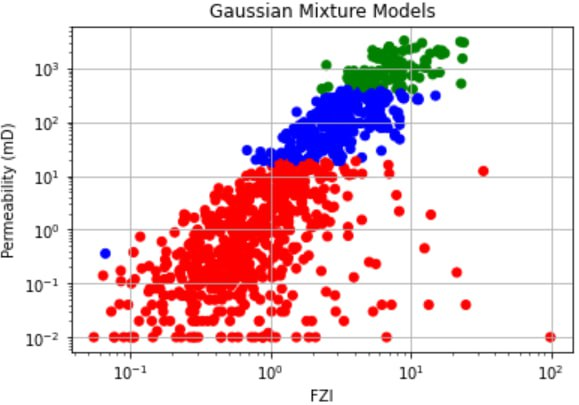
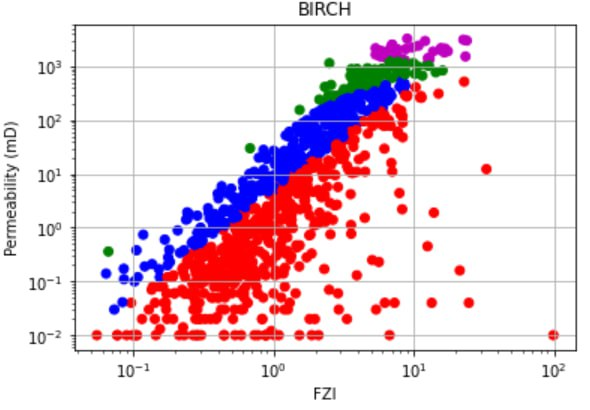
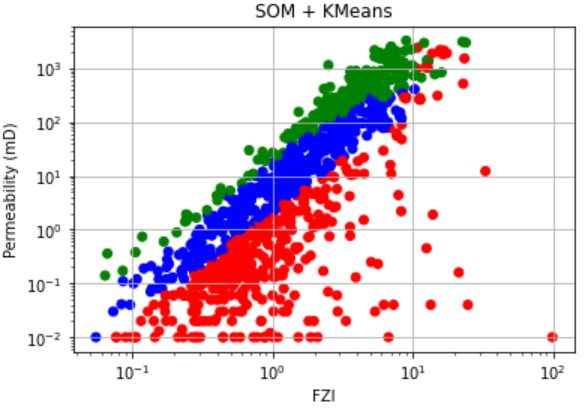
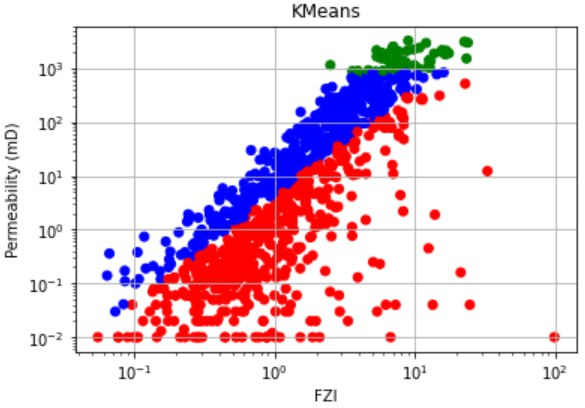
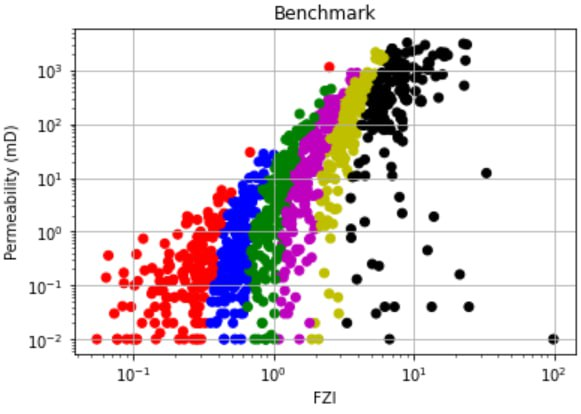
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | IMLR | K-Means | SOM +  K-Means | DBSCAN | BIRCH | GMM |
| Number of Clusters | 6 | 3 | 3 | 2 | 4 | 3 |
| Cluster Determined by | r2 | Inertia Plot | Inertia Plot | Silhouette Score/  Calinski Harabasz Score/  Davis Bouldin Score | Silhouette Score/  Calinski Harabasz Score/  Davis Bouldin Score | BIC |

**Table 5:** Cluster distribution breakdown per method

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Cluster 1 | | Cluster 2 | | Cluster 3 | | Cluster 4 | | Cluster 5 | | Cluster 6 | |
| Method | Num | % | Num | % | Num | % | Num | % | Num | % | Num | % |
| IMLR | 335 | 17 | 333 | 17 | 333 | 17 | 333 | 17 | 333 | 17 | 333 | 17 |
| K-Means | 1010 | 51 | 881 | 44 | 109 | 5 | - | - | - | - | - | - |
| SOM + K-Means | 737 | 37 | 792 | 40 | 471 | 24 | - | - | - | - | - | - |
| DBSCAN | 1439 | 72 | 561 | 28 | - | - | - | - | - | - | - | - |
| BIRCH | 1052 | 53 | 696 | 35 | 184 | 9 | 68 | 3 | - | - | - | - |
| GMM | 1203 | 60 | 575 | 29 | 222 | 11 | - | - | - | - | - | - |



**Figure 16:** Porosity vs Log Scaled Permeability Scatterplot Cluster Distributions

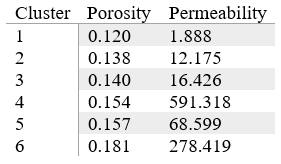


**Figure 17:** Log FZI vs Log Scaled Permeability Scatterplot Cluster Distributions

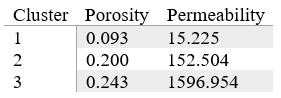
Based on Table 1 BIRCH and K-Means ae the most similar with the average porosity and permeability of cluster 1s in both algorithms to be very similar in terms of value. Likewise for K-Means’ cluster 3 to BRICH’s cluster 4. The results displayed below for the scatterplots show that the 5 cluster methods 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.

Table 7: Cluster Porosity and Permeability Averages

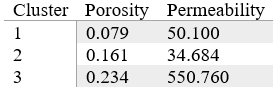
Benchmark



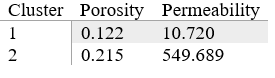
K-Means



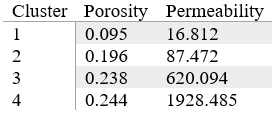
SOM + K-Means



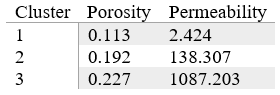
DBSCAN



BIRCH



Gaussian Mixture Models



## Comparing Residuals between Model Outcomes

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

The evaluation uses the 6 rock typing methods to perform permeability prediction and subsequently calculating the residuals for comparing performance. The benchmark method uses the porosity and FZI to predict the RQI and permeability through fitting linear regression lines. The five other methods involve fitting linear regression lines in each cluster, predicting permeability, and then calculating the residuals.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Porosity** | **IMLR** | | **K-Means** | | **SOM+K-Means** | | **DBSCAN** | | **BIRCH** | | **GMM** | | **Exp Data**  **(Permeability)** |
| **A** | **R** | **A** | **R** | **A** | **R** | **A** | **R** | **A** | **R** | **A** | **R** | **-** |
| 0.05 |  |  |  |  |  |  |  |  |  |  |  |  | 0.00000583 |
| 0.10 |  |  |  |  |  |  |  |  |  |  |  |  | 0.0000175 |
| 0.15 |  |  |  |  |  |  |  |  |  |  |  |  | 0.000466 |
| 0.20 |  |  |  |  |  |  |  |  |  |  |  |  | 0.0000875 |
| 0.25 |  |  |  |  |  |  |  |  |  |  |  |  | 0.000172 |
| 0.30 |  |  |  |  |  |  |  |  |  |  |  |  | 0.000627 |

Diagram

Description automatically generated[[

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 K-Means algorithm using the sci-kit learn K-Means. 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 |
| K-Means | 0.529 | 2556.740 | 0.642 |
| SOM+K-Means | 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, K-Means have the highest score, for the Calinski-Harabaz Index, K-Means and BIRCH have high scores and for the Davis-Bouldin Index, K-Means and BIRCH the lowest relative scores. From an overall viewpoint, both K-Means 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 K-Means, SOM+K-Means, 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. K-Means 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, K-Means 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, K-Means and BIRCH are the best-performing algorithms based on 3 metrics.

Overall, when comparing the 5 unsupervised machine learning models to the benchmark model, K-Means 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

|  |  |
| --- | --- |
| [1] | G. Archie, “Introduction to Petrophysics of Reservoir Rocks,” *AAPG Bulletin,* vol. 34, pp. 943-961, 1950. |
| [2] | J. A. Rushing, “Rock Typing — Keys to Understanding Productivity in Tight Gas Sands,” *Society of Petroleum Engineers (SPE) Paper Number 114164,* pp. 1-31, 2008. |
| [3] | C. Hollis, V. Vahrenkamp, C. Tull, A. MookerJee, C. Taberner and Y. Huang, “Pore system characterisation in heterogeneous carbonates: An alternative approach to widely-used rock-typing methodologies,” *Marine and Petroleum Geology,* vol. 27, no. 4, pp. 772-793, 2010. |
| [4] | P. W. M. Corbett and D. K. Potter, “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,* vol. 5, no. 9, pp. 1-12, 2004. |
| [5] | X. Tang, W. Y. Zhou, W. L. Yang, C. Zhang and C. M. Zhang, “An improved method in petrophysical rock typing based on mercury-injection capillary pressure data.,” *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects,* pp. 1-16, 2020. |
| [6] | J. O. Amaefule, A. Mehmet, T. Djebbar, D. G. Kersey and K. K. Dare, “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,* pp. 1-12, October 1993. |
| [7] | M. Khalid, S. E. D. Desouky, M. Rashed, T. Shazly and K. Sediek, “Application of hydraulic flow units’ approach for improving reservoir characterization and predicting permeability.,” *Journal of Petroleum Exploration and Production Technology,* vol. 10, no. 2, pp. 467-479, 2020. |
| [8] | H. Abdulelah, S. Mahmood and G. Hamada, “Hydraulic flow units for reservoir characterization: A successful application on arab-d carbonate,” in *Materials Science and Engineering 380 (2018) 012020*, 2018. |
| [9] | E. Mohammadian, M. Kheirollahi, M. Ostadhassan and M. Sabet, “A case study of petrophysical rock typing and permeability prediction using machine learning in a heterogenous carbonate reservoir in Iran,” *Nature Portfolio,* vol. 12, no. 4505, pp. 1-16, 2022. |
| [10] | I. M. Mohamed, S. Mohamed, I. Mazher and C. Pieprzica , “Formation Lithology Classification: Insights into Machine Learning Methods,” *SPE Annual Technical Conference and Exhibition,* pp. 1-12, 2019. |
| [11] | H. Zakyan, A. K. Permadi and E. A. Pratama, “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,* vol. 15, no. 10, pp. 1-12, 2022. |
| [12] | I. Juhasz, “Normalised Qv - The Key To Shaly Sand Evaluation Using The Waxman-Smits Equation In The Absence Of Core Data,” *OnePetro,* 1981. |
| [13] | GeoProvider AS, [Online]. Available: https://www.geoprovider.no/. |
| [14] | UK National Data Repository, [Online]. Available: https://ndr.nstauthority.co.uk/. |
| [15] | software underground, [Online]. Available: https://dataunderground.org/. |
| [16] | M. Kumar, K. Swaminathan, A. Rusli and A. Thomas-Hy, “Applying Data Analytics & Machine Learning Methods for Recovery Factor Prediction and Uncertainty Modelling,” *Society of Petroleum Engineers,* pp. 1-18, 2022. |
| [17] | P. Franti and S. Sieranoja, “How much can k-means be improved by using better initialization and repeats?,” *Pattern Recognition,* vol. 93, pp. 95-112, 2019. |
| [18] | B. Brentan, G. Meirelles, E. Luvizotto Jr. and J. Izquierdo, “Hybrid SOMþk-Means clustering to improve planning, operation and management in water distribution systems,” *Environmental Modelling & Software,* pp. 1-12, 2018. |
| [19] | G. Vettigli, “MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map,” 2018. [Online]. Available: https://github.com/JustGlowing/minisom/. |
| [20] | A. Ram, S. Jalal, A. S. Jalal and M. Kumar, “A Density based Algorithm for Discovering Density Varied Clusters in Large Spatial Databases,” *International Journal of Computer Applications,* 2010. |
| [21] | A. Starczewski, P. Goetzen and J. E. Meng, “A New Method for Automatic Determining of the DBSCAN Parameters,” *Journal of Artificial Intelligence and Soft Computing Research,* pp. 209-221, 23 May 2020. |
| [22] | T. Zhang, R. Ramakrishnan and M. Livny, “BIRCH: A New Data Clustering Algorithm and Its Applications,” *Data Mining and Knowledge Discovery,* no. 1, pp. 141-182, 1997. |
| [23] | E. Patel and D. S. Kushwaha, “Clustering Cloud Workloads: K-Means vs Gaussian Mixture Model,” *Procedia Computer Science,* vol. 171, pp. 158-167, 2020. |
| [24] | F. A. Al-Ajmi and S. A. Holditch, “Permeability estimation using hydraulic flow units in a central Arabia reservoir,” *SPE Annual Technical Conference and Exhibition. OnePetro.,* pp. 1-12, 2000. |
| [25] | S. Desouky, “A New Method for Normalization of Capillary Pressure Curves,” *Oil & Gas Science and Technology – Rev. IFP,* vol. 58, no. 5, pp. 1-6, 2003. |



**Appendix 1 – Rock Typing Indices Equations**

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