Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Intelligent Hydraulic Flow Unit Mapping: Leveraging Unsupervised and Supervised Learning on Large-Scale Core Data

Mohammed Joobayear Hossain
Shahjalal University of Science And Technology
Minhaz Chowdhury
Shahjalal University of Science And Technology
Amad Hussen
Auburn University
Md Shofiqul Islam
shofiq-pme@sust.edu

Shahjalal University of Science And Technology

Research Article

Keywords: Hydraulic flow unit, Reservoir characterization, Machine learning, Core data, Boosting algorithm

Posted Date: April 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9370518/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Integrating Unsupervised Clustering and Supervised Learning for Scalable Hydraulic Flow Unit Prediction in Reservoir Characterization

Mohammed Joobayear Hossain, Minhaz Chowdhury, Amad Hussen, Md Shofiqul Islam*
Department of Petroleum and Mining Engineering, Shahjalal University of Science and Technology, Sylhet 3114, Bangladesh
*Corresponding author, Email: shofiq-pme@sust.edu

Abstract

Accurate identification of hydraulic flow units (HFUs) is fundamental for reservoir characterization; however, conventional approaches, such as histogram analysis, log-log plots of reservoir quality index (RQI) versus porosity index (ϕz), and Z-score probability tests, often suffer from subjectivity, data overlap, and limited scalability across large datasets. To address these limitations, this study introduces a hybrid machine learning workflow that integrates unsupervised clustering and supervised classification models to automate the identification and prediction of HFUs. In the first phase, unsupervised models including K-means, K-medoids, Fuzzy C-means (FCM), and Gaussian mixture models (GMM) were employed to detect the optimal number of HFUs. The GMM demonstrated superior clustering performance ($R^2 = 0.9278$, RMSE = 0.3365) compared to K-means and K-medoids, whereas FCM underperformed. In the second phase, supervised learning models were applied to predict HFUs using laboratory-derived core features. Among the tested models, k-nearest neighbors (KNN), random forest classifier (RFC), support vector machine (SVM), gradient boosting (GB), extreme gradient boosting (XGBoost), adaptive boosting (AdaBoost), and a stacking hybrid were evaluated. RFC outperformed the others with robust generalization (training accuracy: 0.983; testing accuracy: 0.972), while SVM showed moderate success, and KNN exhibited overfitting. Boosting models, such as XGBoost and GB, achieved high training accuracy but suffered from overfitting. In contrast, AdaBoost demonstrated relatively lower performance but stronger generalization capabilities. The stacking model, though highly accurate in training, also displayed overfitting during testing. Computational efficiency analysis further highlighted the trade-off between training time and predictive performance, with KNN and SVM being the fastest but also the least reliable. At the same time, RFC provided the most balanced accuracy–time outcome. Overall, the proposed workflow establishes an effective and scalable methodology for HFU classification, offering greater consistency, objectivity, and applicability to large reservoir datasets in the Norwegian sector of the North Sea.

Keywords: Hydraulic flow unit, Reservoir characterization, Machine learning, Core data, Boosting algorithm

1. Introduction

Accurate reservoir characterization is critical in petroleum engineering due to the spatial and compositional heterogeneity of reservoirs. This heterogeneity varies across geological scales and is influenced by both depositional environments and post-depositional processes, resulting in variations in reservoir properties from the pore scale to the entire reservoir (Abbaszadeh et al., 1996). Accurately characterizing the reservoir and estimating reservoir properties is therefore challenging. To address this challenge, Hydraulic Flow Unit (HFU) zonation is applied, which categorizes rocks with similar petrophysical and flow characteristics into a single unit, thereby reducing uncertainty in reservoir description (Amaefule et al., 1993).

HFUs are portions of reservoir rock characterized by uniform geological and petrophysical properties that affect fluid flow, such as Porosity, permeability, pore structure, mineral composition, and sedimentary features (Rebelo et al., 2022). Identifying HFUs through the Flow Zone Indicator (FZI), derived from the Reservoir Quality Index (RQI) and Normalized Porosity Index (ϕz) plots, is essential for effective reservoir characterization (Amaefule et al., 1993). The classification of HFUs enhances the understanding of both macroscopic and microscopic heterogeneity and leads to more accurate estimations of reservoir quality. This type of heterogeneity, in turn, informs essential operational and investment decisions such as well placement, production and injection planning, infrastructure design, and enhanced oil recovery while minimizing uncertainty in hydrocarbon reserve assessment (Astsauri et al., 2024).

Several studies were conducted to classify HFU using mathematical, numerical, and physics-based equations for improved representation of reservoir heterogeneity. Amaefule et al. (1993) proposed an innovative method employing the Kozeny-Carman equation. This technique determines the rock quality index (RQI) and generates a log-log plot of the pore-to-matrix ratio (PMR); points from the sample belonging to the same HFU fall on the same straight lines with a consistent slope. However, this manual interpretation is less suitable for accurately identifying HFU due to the overlapping problem that arises when the dataset is large, requiring more time for interpretation.

A unique technique for enhanced depiction of heterogeneity, the HFU classification, established by Hearn et al. (1984), categorizes flow zones based on rock type, correlating geological, static, and petrophysical properties. The fundamental objective of this categorization is to establish a connection between geological features and dynamic petrophysical properties. Abnavi et al. (2018) identify flow units of gas reservoirs in southern Iran using histogram plots and normal probability plots. Results show that the normal probability plot is more accurate than the histogram plot for evaluating reservoir quality. Using the least squares method, Nayak et al. (2021) classify flow zones for calcareous reservoirs in the Mumbai region, demonstrating how regression models facilitate an understanding of flow behavior, even in low-permeability zones. This flow-behavior is because carbonate reservoirs exhibit more intricate pore systems compared to clastic ones. Shenawi et al. (2009) developed a modified HFU classification method for enhancing reservoir characterization of carbonate oil reservoirs in Saudi Arabia. Svirsky et al. (2004) developed an HFU-based classification to differentiate between productive and nonproductive zones, thereby optimizing well placement and planning in the Siberian field. Aguilera et al. (2010) adapted the conventional HFU classification technique to modern unconventional resource development. Flow-unit-based classification distinguishes matrix-dominated zones and fracture-dominated zones for tight gas and shale reservoirs. By analyzing core data, Soto et al. (2010) identify different types of pores and combine this information with the HFU model to understand small-scale pore features alongside larger flow units, which helps predict how fluids will move. This study predominantly employed physical equations and regression models to delineate flow zones and formulate empirical relationships between Porosity and permeability.

Traditionally, HFU class identification above requires more time, is subjective, and is not practical for large datasets. It becomes unreliable when the dataset size is large, which can lead to overlapping points, inconsistent interpretations, and an inability to scale across multiple wells or fields (Adjei et al., 2025). Because of its manual nature, the method lacks automation, repeatability, and efficiency, making it unsuitable for modern reservoir characterization tasks (Man et al, 2021; Trung et al, 2024). In recent times, various innovative data-driven strategies, including machine learning, deep learning, and statistical techniques, have been employed to address classification problems in the oil and gas industry. For the purpose of classifying HFU, several unsupervised, supervised, and deep learning models have been developed, including K-means, GMM, SOM, FCM autoencoder-based K-means, RF, DT, SVR, XGBoost, MLP, LG, RFC, etc.

(Man et al., 2021; Trung et al., 2024; Astsauri et al., 2024; Mohammadinia et al.,2023; Hussain et al.,2025; Adjei et al.,2025)

To enhance HFU classification, Man et al. (2021) developed a sophisticated machine learning algorithm utilizing well log data. Supervised machine learning algorithms, including ANN, SVM, DT, and RF, are deployed to predict FZI values for each flow zone with enhanced accuracy. The results indicate that the RF model surpasses the other models, reduces unnecessary coring expenses, and saves time. Nguyen et al. (2024) utilized a hybrid approach of unsupervised and supervised models. This study initially employed an unsupervised K-means model to categorize HFU based on FZI for a heterogeneous Middle Miocene carbonate reservoir in Vietnam's Song Hong basin. Subsequently, three machine learning algorithms were built utilizing wireline log data to assess permeability with enhanced accuracy. Astsauri et al. (2024) categorize HFU by K-means clustering and automated HFU classification based on well log data for intricate carbonate heterogeneous reservoirs. Multiple supervised models are implemented, including Random Forest, XGBoost, SVM, and ANN. The results indicate that the RF model outperforms the other models. Mohammadinia et al. (2023) devised a K-means unsupervised model for categorizing HFU, using well log data. FZI Predictions are made using the SVM, ANN, RF, LR, and LB models for oil fields in southwestern Iran. The LB model demonstrates superior precision. Hussain et al. (2025) firstly derived FZI based on Porosity and permeability from the Kozeny equation and identified the HFU zone using K-means clustering, enhanced by particle swarm optimization. Results showed K-means clustering improves the precision of HFU classification. For instance, Charkhi et al. (2019) applied an adaptive neuro-fuzzy inference system (ANFIS) to well log data for HFU classification, demonstrating that such models can predict HFUs with a higher level of confidence compared to traditional approaches.

All of the above studies using data-driven models were performed by correlating with well log data for classifying HFU. Only one study, conducted based on core data by Adjei et al. (2025) implemented a Z-score probability test to classify HFU and further compared it with one sophisticated unsupervised model, namely autoencoder-based k-means, for the Subei Basin oilfields in China. These HFU classes further aligned with significant core features and trained a supervised data-driven model to automatically classify HFU. Stephen utilized neural networks, including multilayer perceptron’s (MLPs), logistic regression (LR), and random forest classifiers

(RFCs). The MLP outperformed all other models achieving accuracy 94%, significantly enhancing subsurface characterization.

Our study is proposed:

1. Traditional Z-score probability test and RQI vs ϕz plot will be applied as the conventional approach for HFU classification and subsequently replaced by clustering algorithms for improved automation and accuracy.
2. Four unsupervised clustering algorithms K-means, K-medoids, Gaussian Mixture Model (GMM), and Fuzzy C-means to automatically classify samples into HFUs based on FZI patterns. The clustering performance will be evaluated using statistical metrics such as R^2 and RMSE, and the optimal HFU classes are determined through elbow detection.
3. HFU classes are then aligned with significant core features, and supervised data-driven models, including KNN, RF, SVM, XGBoost, AdaBoost, Gradient Boosting, and Hybrid Stacking, are trained to automatically detect HFUs from core data. This proposed supervised model has not been previously investigated, and this investigation is conducted in the Norwegian sector of the North Sea. This region has not been previously studied for HFU classification.
4. Laboratory-derived core features ranking based on their contribution to the model output, and model ranking analysis based on performance will be conducted.
5. Computational efficiency of all models will be evaluated in terms of training time and scalability to assess their suitability for large-scale applications.

The automatically detecting approach, using AI, addresses the limitations of manual interpretation by reducing time, overcoming the overlapping problem in the histogram plot, and ensuring consistency even in large datasets. This automated workflow enables scaling across multiple wells or fields and accurately identifies HFU, allowing for rapid application to large datasets or future wells with similar features.

2. Research Methodology

2.1 Hydraulic Flow Zone Indicator (FZI)

HFUs are segments of the reservoir that exhibit similar fluid flow behavior based on their flow characteristics, rather than their stratigraphy (Charkhi et al.,2019). Fluid flow in a reservoir

primarily depends on the geometry, shape, and size of the pores. Kozeny, (1927) and Carman, (1937) deduced multiple equations that show how pore geometry, sorting, and grain size are related to the porosity and permeability that control the fluid flow behavior in the reservoir (Rebelo et al., 2022). Equations (1), (2), and (4) derive significant property RQI and normalized Porosity, which further derive the FZI value that correlates with the petrophysical property of reservoir rock. Based on the FZI pattern, reservoirs are classified into multiple flow units, generating an RQI vs. ϕ_z log-log plot using the traditional method. This process generates multiple straight lines, and each straight line corresponds to one HFU, which has a constant FZI (Charkhi et al., 2019). Another process, the log (FZI) distribution, is examined through a histogram, where each peak corresponds to a distinct HFU. Multiple peaks indicate the presence of multiple flow units within the reservoir (Abnavi et al., 2018).

Two important parameters are derived for FZI calculation:

Reservoir Quality Index (RQI):

$$RQI = 0.0314\sqrt{\frac{K}{\phi}} \tag{1}$$

where k is permeability (mD) and ϕ_e is effective porosity (fraction

Normalized Porosity (ϕ_z)

Normalized porosity is obtained from the Kozeny–Carman relation:

$$\phi_z = \frac{\phi_e}{1-\phi_e} \tag{2}$$

From the Kozeny–Carman equation, permeability is expressed as a function of effective porosity and specific surface area per grain volume. This relationship highlights that pore geometry and connectivity strongly control flow capacity.

$$K = \left(\frac{1}{K_T S_{vgr}^2}\right)\left(\frac{\phi_e^3}{(1-\phi_e)^2}\right) \tag{3}$$

The Flow Zone Indicator (FZI) is then computed by linking RQI and normalized porosity

$$FZI = \frac{RQI}{\phi_z} \tag{4}$$

$$K = 1014.24\,(FZI)^2 \frac{\phi_e^3}{(1-\phi_e)^2} \quad (5)$$

The primary goal of this research is to introduce diverse models that investigate various input data configurations. This study utilizes ten main core features: depth, horizontal permeability, Porosity, grain density, pore throat size, specific surface area per pore volume, specific surface area per grain volume, CQI, sorting, and grain size. Firstly, FZI and RQI are calculated from the Kozeny equations (1), (2), and (4), and FZI serves as an input for unsupervised machine learning clustering algorithms, including K-means, K-medoids, Gaussian mixture models, and fuzzy C-means, for deriving a hydraulic flow unit. Derived HFU classes are aligned with the remaining core feature as a target variable for the supervised machine learning algorithm. Multiple supervised ML models (KNN, RF, SVR, XGBoost, AdaBoost, Gradient Boost, and a hybrid stacking model) are implemented and evaluated for classifying the HFU.

2.2 Geological Settings

The Norwegian sector of the North Sea is one of the most mature hydrocarbon provinces globally, with more than 1,240 wells drilled across its rift-related fault blocks, grabens, and structural highs. Its geological setting reflects a complex rift history from the late Paleozoic to Mesozoic, with prolific Upper Jurassic Kimmeridge Clay and Draupne. Formation source rocks charging high-quality sandstone reservoirs of the Brent and Viking groups, while Permian Zechstein evaporites and younger Cretaceous–Cenozoic shales provide regional seals. This stratigraphic and structural framework, combined with abundant well data, makes the region ideal for investigating reservoir heterogeneity (Glennie, 1998; Doré et al., 1999).

2.3 Z-Score Probability Test

To ensure consistent comparison of FZI data, the Z-score method was applied. This approach standardizes values relative to their mean and standard deviation, enabling statistical evaluation of reservoir flow units (Adjei et al., 2025). Since Z-score analysis requires normally distributed data, transformations, such as base-10 logarithm, power transformation, Box-Cox transformation, and Yeo-Johnson transformation, were applied to all values, assessing p-value and best-fit plots.

$$Z = \frac{X-\mu}{\sigma} \quad (6)$$

Where, X = data point

μ = mean, σ = standard deviation

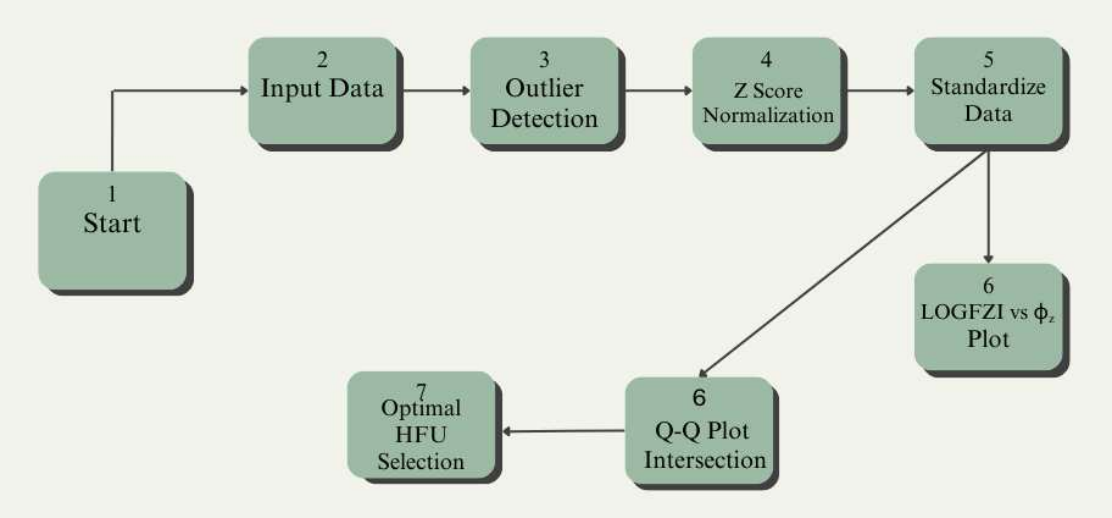


Fig. 1: Traditional HFU Classification Workflow Using Q–Q Plot and logFZI vs ϕ_z Cross-Plot

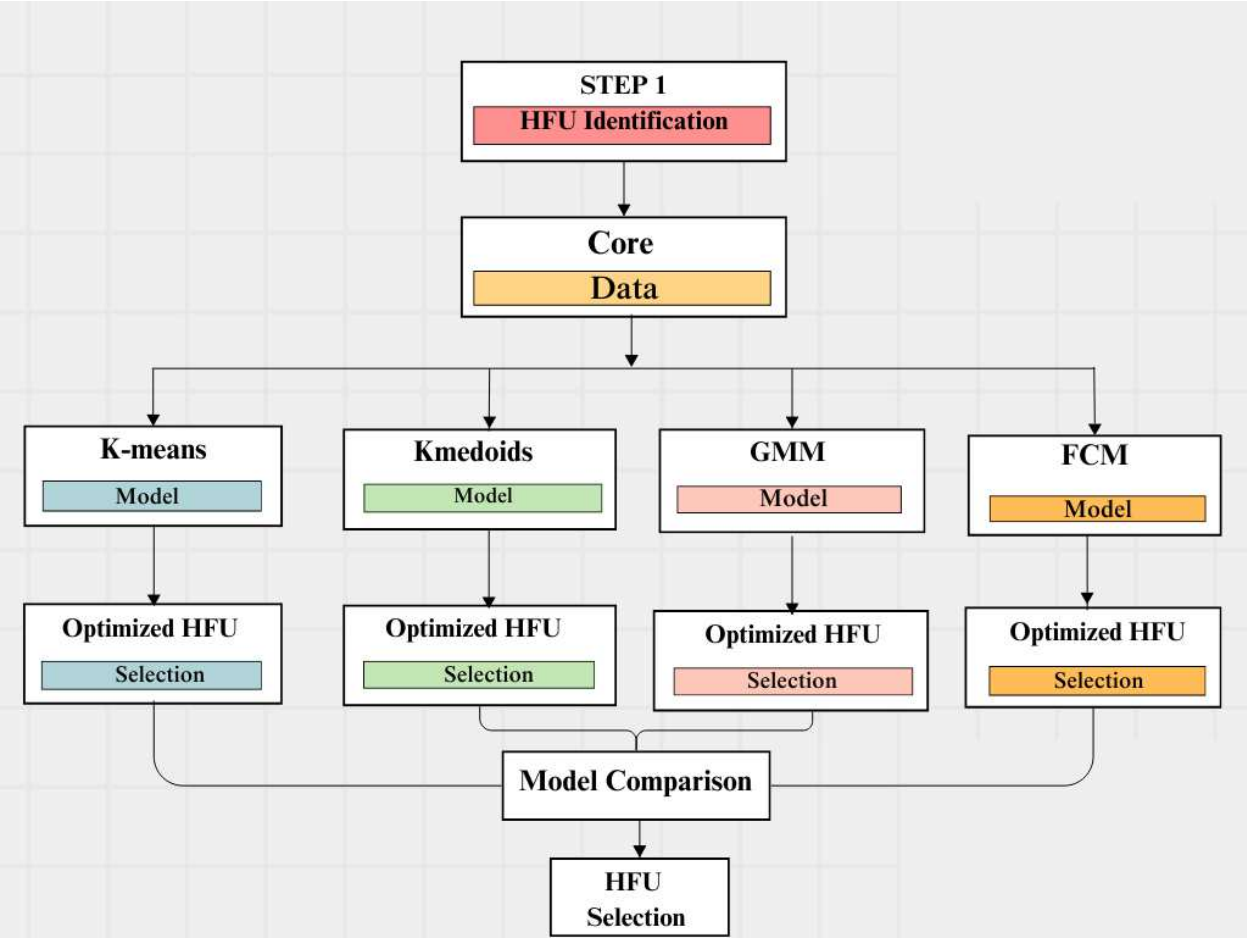


Fig. 2: Hybrid HFU Classification Workflow: a) Unsupervised Clustering Algorithm

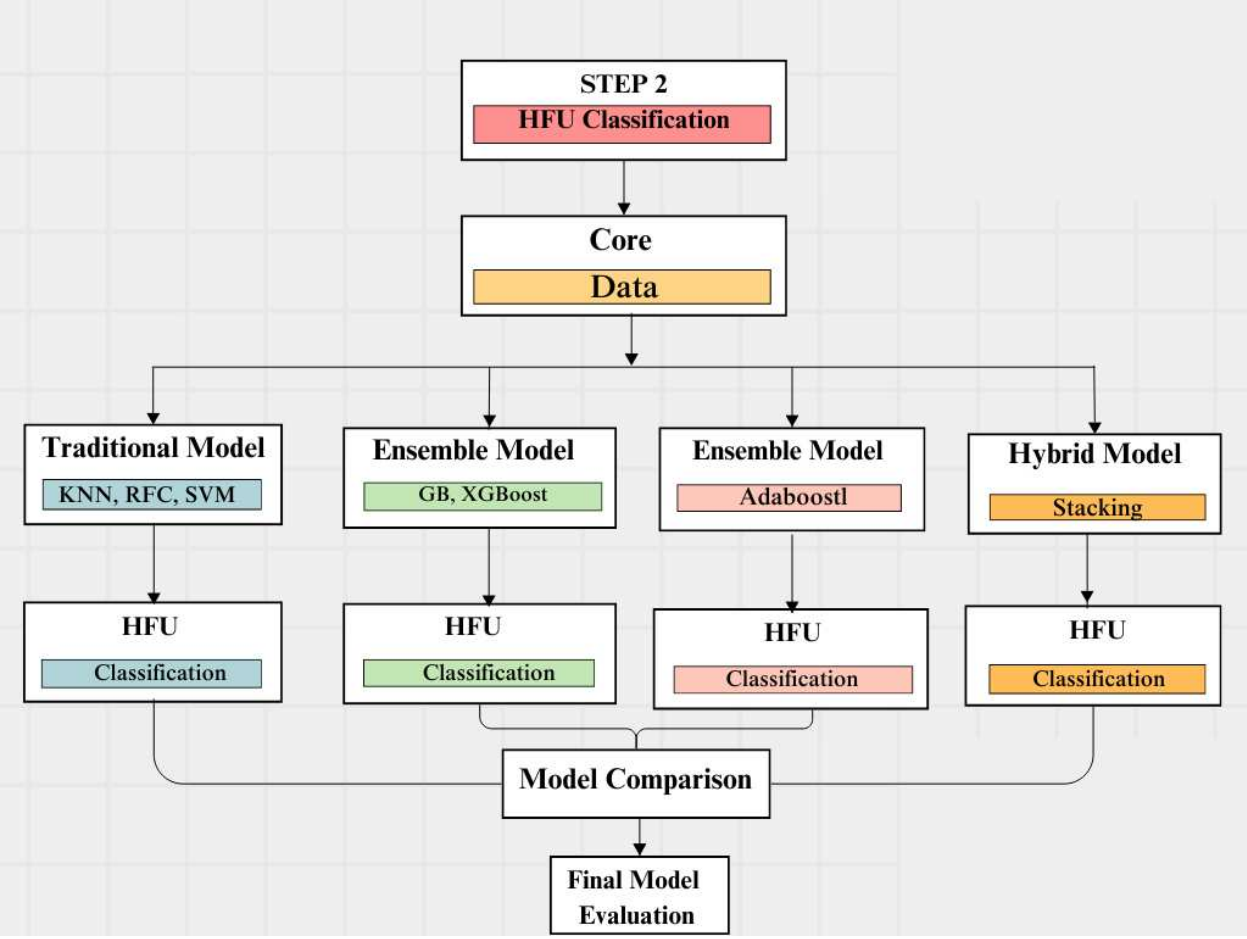


Fig. 2: Hybrid HFU Classification Workflow: b) Supervised Data Driven Machine Learning Approaches

2.4 Model Description

2.4.1 Unsupervised Model

K-means

K-means is a popular unsupervised machine learning algorithm that groups a dataset into K clusters. It is selected for its simplicity and widespread application in clustering tasks. (Jain et al., 2010; Ahmed et al., 2020). The objective of this model is to place the data point into the same cluster, which has similar characteristics (Jain et al., 1999). The algorithm works iteratively; initially, random data points from the dataset are selected as centroids. All data points are assigned to the nearest centroid, and then the centroids are updated as the mean of their assigned points. The algorithm minimizes a performance criterion called P, which is calculated as the sum of squared

error distances between data points and their corresponding cluster center. This process recomputes until the centroids stabilize (Ahmed et al., 2020).

K-medoids Clustering

K-medoids is a clustering algorithm that, similar to K-means, aims to divide a dataset into *K* clusters. However, instead of using the arithmetic mean as the cluster center, K-medoids selects representative objects (medoids) that are actual data points within the dataset (Kaufman et al.,1990). This makes it more robust to outliers and noise, since the medoid is less affected by extreme values compared to the mean (Park et al., 2009).

Formally, let a dataset be defined as

$$X= \{x_1, x_2, \ldots\ldots\ldots\ldots, x_n\}, x_i \in R$$

where *n* is the number of data points. Suppose we want to partition *X* into *K* clusters C= {C1, C2, ……..,C_K} each represented by a medoid $m_j \in C$

The objective function of K-medoids is to minimize the total dissimilarity between points and their nearest medoid:

$$J = \sum_{j=1}^{k} \sum_{xi \in Cj} d\,(x_i, m_j) \quad (7)$$

Where d (x_i, m_j) is a dissimilarity measure, often Euclidean distance for numerical data or other metrics for categorical or mixed data (Kaufman et al.,1990).

Gaussian Mixture Model (GMM)

The Gaussian Mixture Model (GMM) is a probabilistic clustering method that represents data as a mixture of multiple Gaussian distributions. GMM uses soft clustering, where each point has a probability of belonging to every cluster (McLachlan et al., 2000). This makes GMM more flexible in handling clusters of different shapes, sizes, and densities (Reynolds and D.A, 2008). GMM is usually trained using the Expectation–Maximization (EM) algorithm, which iteratively estimates cluster membership probabilities (E-step) and updates the Gaussian parameters—means, covariances, and weights (M-step) (Hoff et al., 2009). The process continues until convergence, typically when the likelihood stabilizes. The model overcomes some of the limitations of K-means,

particularly its restriction to spherical clusters, but it is also more computationally intensive and sensitive to initialization. The number of mixture components must still be specified in advance. However, methods such as the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC) are often used to guide model selection (Reynolds et al., 2009).

Fuzzy C-Means (FCM) Clustering

Fuzzy C-means (FCM) is a clustering technique that extends K-means by using the concept of fuzzy set theory, allowing data points to belong to multiple clusters with varying degrees of membership (Bezdek and J.C., 1981). This soft clustering approach makes FCM especially effective for data with overlapping boundaries or when cluster definitions are uncertain (Krinidis et al., 2010).

The FCM algorithm optimizes an objective function that minimizes the weighted within-cluster distances. Each data point has a degree of membership associated with each cluster, controlled by a fuzzification parameter ($m>1$). A higher value of m produces fuzzier clusters, whereas values close to 1 approach hard clustering. During each iteration, membership degrees are updated based on the distances to cluster centers. Then new centroids are computed as the weighted average of all points, with weights proportional to their membership degrees. This iterative process continues until convergence, typically when changes in the objective function fall below a threshold. One of the key strengths of FCM is its flexibility in handling clusters with uncertainty and overlapping boundaries. It has been widely applied in fields such as remote sensing and pattern recognition (Chen et al., 2004). However, the algorithm is sensitive to noise and outliers, as these can distort the membership values and cause shifts in centroids. To mitigate this limitation, several robust variants of FCM have been developed, incorporating spatial constraints, kernel functions, or local information to improve clustering quality in noisy environments (Krinidis et al., 2010; Chen et al., 2004).

2.4.2 Supervised Model

K- Nearest Neighbor

The K-nearest neighbor is employed for classification and regression tasks that work by calculating the distance of neighbors using the Euclidean method. The optimal K value is predefined, and the model calculates distance when a new data point is seen. When it is a classification problem, the

model identifies the class by a majority vote. For a regression problem, the decision is made by averaging the values of the K nearest neighbors (Halder et al., 2024) Multiple distance functions are used with the particular variant of KNN. The optimal value of 'k' is typically selected using cross-validation to strike a balance between bias and variance. As the KNN model is sensitive to the distance in the input features, the dataset needs to be standardized to reduce bias. (Zhang et al., 2018).

The Euclidean distance between two points

$x= (x_1, x_2..., x_n)$ and $y= (y_1, y_2..., y_n)$

in an n-dimensional feature space is given by: $d(x, y) = \sqrt{\sum_{i=1}^{n}(xi - yi)^2}$ (8

Random Forest Classifier

Random Forest is a supervised learning technique that utilizes the ensemble learning method with several decision trees. Random Forest is a bagging technique wherein all computations are executed concurrently, and there is no interaction among the decision trees throughout their construction. Random Forest can be employed to address both classification and regression applications (Breiman et al., 2001). The term "Random Forest" derives from the bagging concept that employs resampling to create a unique training subset from the sample training data. For the classification task, RF builds several multiple trees, and each tree provides a predicted class. The majority vote among all trees makes the final prediction. It is a robust machine learning technique that mitigates the drawbacks of a decision tree model (Jaiswal et al., 2017).

Support Vector Regression

Support Vector Machine (SVM) is frequently employed for classification tasks. Support Vector Machines (SVM) seek to identify the ideal hyperplane that most effectively distinguishes between the two classes within the input data. A hyperplane is a flat subspace of dimension *p–1* within a *p*-dimensional space, where *p* represents the number of input characteristics. The ideal hyperplane maximizes the margin, defined as the distance between the hyperplane and the nearest data points from each class, referred to as support vectors (Vapnik et al., 1995; Cortes et al., 1995). In SVR, the margin is characterized as the model's error tolerance, sometimes referred to as the ε-insensitive tube. This tube allows for a certain deviation of the data points from the hyperplane without being

classified as errors. The hyperplane represents the optimal fit for the data contained within the ε-insensitive tube (Smola et al., 2004).

SVM can be expressed theoretically as a convex optimization problem. The goal of the task is to identify a function f(x) that is as flat as feasible while maintaining a maximum deviation of ε from the real targets throughout all training data. The flatness of the function indicates diminished sensitivity to minor variations in the input data, hence mitigating the risk of overfitting. In linear functions, flatness is characterized by a minimal value of *w* in the optimal fit function $f(x) = w \cdot x + b$ (Wu et al., 2005).

Gradient Boosting Classifier

Gradient Boosting Classifier is an ensemble technique that is used for classification tasks. It is a boosting approach that combines many weak learners to make a strong prediction model (Friedman, 2001). It works by training models sequentially and learning from the errors made by the previous one; this process is called gradient descent optimization. The gradient boosting model utilizes the gradient descent optimization technique to minimize the model's loss function, as described in (Chen et al., 2016). The loss function could be mean squared error or cross-entropy. The method calculates the gradient of the loss function with respect to the predictions in each iteration. It then trains a new weak model to make this gradient as small as possible (Chen et al., 2016). After that, the new model's predictions are added to the ensemble (the combined predictions of all models), and this process is repeated until a stopping condition is met (Friedman, 2001; Raschka, 2017).

Suppose we have training data (x_i, y_i) for i = 1……, n. The goal is to build a model F(x) that predicts y

$$Fo(x) = \arg min \sum_{i=1}^{n} L(yi\ \gamma) \quad (9)$$

Where L is the loss function squared error $L(y, F(x)) = (y - F(x))^2$

Extreme Gradient Boosting Classifier

XGBoost (Extreme Gradient Boosting) is a scalable tree boosting algorithm that enhances gradient boosting by incorporating regularization, weighted quantile sketching, and parallelization, making it faster and more efficient. This ensemble-based model builds multiple decision trees sequentially,

which are referred to as learners. The model learns from the error that the previous weak model made. This optimization process reduces the risk of overfitting and enhances the model's performance (Chen et al., 2016). XGBoost has multiple hyperparameters that are used for regularization of the model's performance, such as learning rate, maximum depth, gamma function, and n_estimators. The learning rate indicates how each of the weak models contributes to the final prediction, and for enhanced model performance, more trees are required. Increasing the number of trees to achieve better results may also increase the computational cost. The max_depth parameter controls the depth of each tree, which reduces overfitting and is essential for controlling the model's complexity (Chen et al., 2016).

Adaptive Boosting Classifier

AdaBoost (Adaptive Boosting) is a supervised ensemble learning algorithm used primarily for classification tasks. To form a strong classifier, this model combines multiple weak learners and learns from the errors made by the previous one (Freund et al., 1997). In each iteration, a weak learner is trained, and its weighted error is calculated. The learner is then assigned a weight based on its accuracy, with better learners contributing more to the final model. Sample weights are updated so that misclassified samples receive higher emphasis in the next iteration. This process repeats for a predefined number of iterations, and the final prediction is made using a weighted majority vote of all weak learners. By focusing on hard-to-classify examples and combining weak learners optimally, AdaBoost reduces error and improves predictive performance while mitigating overfitting (Schapire et al., 2003).

Stacking

Stacking, also known as layered generalization, is a machine learning technique that enhances predictive accuracy by combining the results of multiple base models. This approach utilizes multiple base models to identify data patterns, thereby mitigating the shortcomings of individual models and leveraging their collective strengths (Sun et al., 2025). Initially, several base models are trained, and their predictions serve as inputs for a meta-model to forecast the target outcome. Stacking is frequently employed in machine learning for classification, regression, and anomaly detection applications (Wolpert, 1992). David H. Wolpert initially presented stacking in his 1992 seminar paper, "Stacked Generalization," and subsequently elaborated on it in his book, "The Elements of Statistical Learning." The referenced literature and further studies underscore the

significance and effectiveness of stacking as a formidable tool in the arsenal of machine learning researchers. Fig. 3 illustrates the essential stacking workflow.

2.5 Cross-validation

Cross-validation is a statistical technique employed in modeling to assess the accuracy of prediction and modeling procedures. Numerous methodologies exist for cross-validation, including K-fold cross-validation, leave-one-out cross-validation, random subsampling, and stratified K-fold cross-validation, among others. In K-fold cross-validation, the dataset is divided into approximately equal-sized folds or subsets. The model is trained K times, employing K-1 folds as training data and the remaining fold for validation. Thus, all data points are ultimately utilized for both training and testing, which is a key advantage of K-Fold Cross-validation. Random subsampling entails dividing the original dataset into two segments: training and testing. Common train/test ratios include 70–30 and 80–20, or may fluctuate depending on the dataset's size. The modeling is conducted with the training dataset, and predictions are generated from the test subset. Random subsampling ensures external prediction and maintains confidence in varied datasets with identical characteristics (Mudhafar and W.J, 2016; Rahimi et al.,2022). This work utilizes 10-fold cross-validation in the training instead of 5-fold cross validation because of better balance between bias and variance of the suggested machine learning algorithms. Ten-fold cross-validation provides a compromise between computational efficiency and thorough model assessment.

2. 6 Performance metrics:

Unsupervised clustering and Data Driven models are evaluated using 5 statistical evaluation metrics, Coefficient of determinant, Root Mean Squared Error, Precision, Recall and F1-score. Table 1 shows mathematical description of this five metrics.

Table 1: Mathematical descriptions of performance metrics

Performance Metrics	Mathematical Illustration	Nomenclatures
Coefficient of determinant (R^2)	$R^2 = 1 - \frac{\sum_{i=1}^{n}(X_i - Y_i)^2}{\sum_{i=1}^{n}(X_i - \overline{Y_i})^2}$	Y_i = predicted value of target parameters X_i = corresponding true value $\overline{Y_i}$ = mean value of the qi TP = True positive FP = False Positive
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{(X_i - Y_i)^2}{n}}$	
Precision	$\frac{TP}{TP + FP}$	

Recall	$\frac{TP}{TP + FN}$	FN = False Negative
F1-score	$2 \times \frac{Precision \times Recall}{Precision + Recall}$	

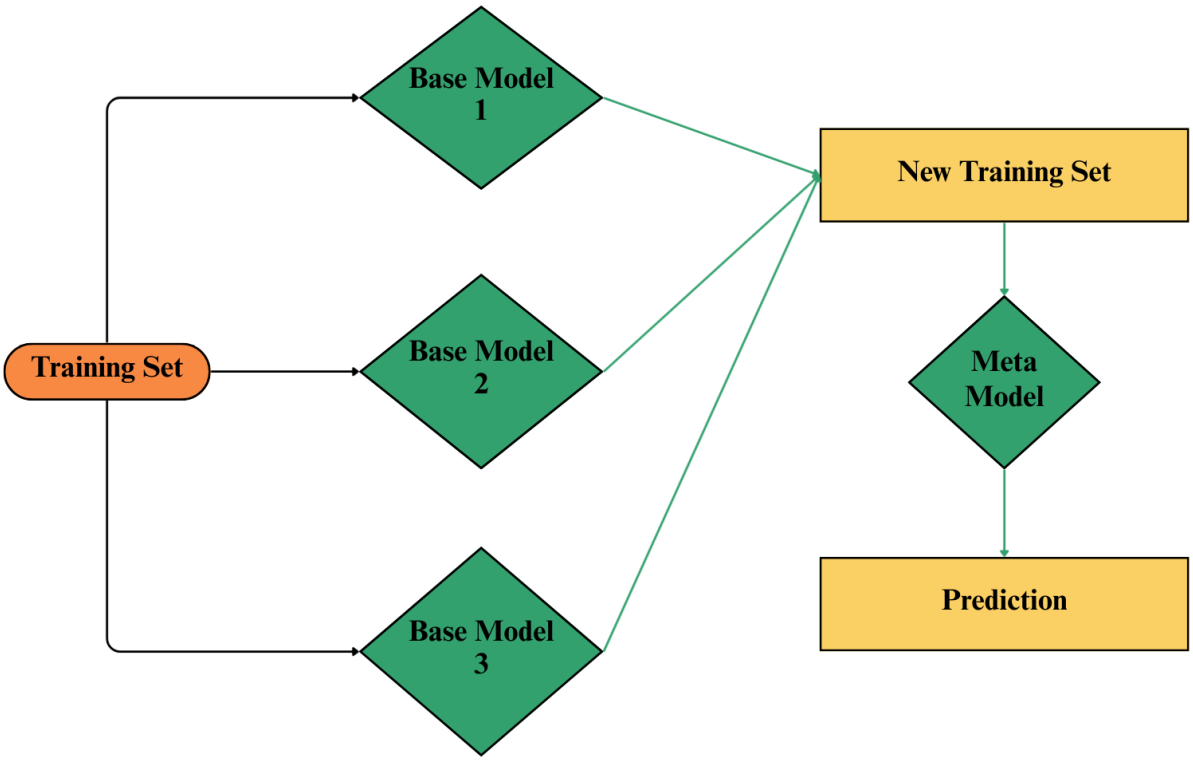


Fig. 3: Work Flow of Stacking Model

3. Data Preparation and Preprocessing

3.1 Data Preparation

The dataset used in this study is obtained from the RealPore database (GeoProvider AS), which contains laboratory-derived core measurements from approximately 1,240 wells across the Norwegian continental shelf. For this research, we specifically analyze data from well 31/5-B-4H, located in Block 31/5 of the North Sea. According to records from the Norwegian Petroleum Directorate (NPD), this block belongs to the Troll Field area. Hence, the data utilized in this study originates from the Troll Field, within the Norwegian sector of the North Sea. In the studied field,

after collecting the required data, the information related to the flow behavior of the reservoir was collected. The data were prepared in four stages: data acquisition, data conditioning and preprocessing, parameter derivation and environment setup, and petrophysical model development. HFU zones are determined by categorizing depth intervals based on pore throat attributes, which are the dominant factors in a rock's permeability (Astsauri et al., 2024). In addition to measured RCAL data, several derived features were computed to better characterize pore structure, including pore throat size, specific surface area per pore volume, and specific surface area per grain volume. These derived features are based on the Kozeny–Carman framework, which relates permeability to porosity and surface area. Kozeny (1927) and Carman (1937) extended Poiseuille's and Darcy's laws to establish one of the most fundamental permeability expressions. Kozeny derived one of the most fundamental and popular expressions of permeability as a function of Porosity and specific surface area.

$$q = \frac{n\pi r^4}{8\mu}\frac{\Delta p}{L} \tag{10}$$

$$q = \frac{KA}{\mu}\frac{\Delta p}{L} \tag{11}$$

Where Δp is pressure drop, μ is viscosity, and L is length of the core sample, A is cross-sectional area, K is the permeability.

Combining Equation 10 and 11 for K gives,

$$K = \frac{n\pi r^4}{8A} \tag{12}$$

From the definition of porosity,

$$\phi = \frac{PV}{BV} = \frac{n\pi r^4}{A} \tag{13}$$

Solving Equation 12 and 13,

$$K = \frac{\phi r^2}{8} \tag{14}$$

The surface area per unit grain volume (Sgr) is related to $S_{VP,}$ r, and ϕ_e as follows:

$$S_{vgr} = \frac{n2\pi rL}{AL(1-\phi)} = \frac{n\pi r^2}{A}\frac{2}{r}\frac{1}{1-\phi} \tag{15}$$

$$S_{vp} = \frac{A_s}{V_P} = \frac{n2\pi rL}{n\pi r^2 L} = \frac{2}{r} \quad (16)$$

Combining Equation 13, 15 and 16,

$$S_{vgr} = S_{vp}\frac{\phi}{1-\phi} \quad (17)$$

Where ϕ is porosity, PV is pore volume, BV is bulk volume, r is pore throat radius, Sgr is surface area per unit grain volume, S_{VP} is surface area per unit pore volume

3.2 Data Preprocessing

Data preprocessing is a crucial step in training and testing machine learning models. The dataset obtained from the Norwegian section of the North Sea consists of 452 core data samples. In the preprocessing step, data cleaning is conducted, as the dataset has a limited number of null values, and duplicate rows are removed from the dataset. The quantitative features of core data include horizontal permeability, Porosity, grain density, pore throat size, specific surface area per pore volume, and specific surface area per grain volume. The qualitative attributes consist of CQI, grain size, and sorting. These categorical variables were transformed into numerical form using ordinal encoding for grain size and label encoding for CQI and sorting. This transformation is necessary for training and testing machine learning models.

Table 2: Statistical description of core feature

	Depth (m)	Horizontal Permeability (mD)	Grain Density (g/cm3)	Pore throat size (cm)	Surface Area (per PoreVol) (cm^{-1})	Surface Area per Grain Vol (cm^{-1})	Porosity (ϕ)	FZI
Count	452.000	452.000	452.000	452.00	452.000	452.000	45.21	452.00
Mean	1572.92	1677.900	2.670	115.035	0.0603	0.017	30.49	3.992
Std	41.914	2603.840	0.023	111.381	0.1056	0.048	10.95	6.223
Min	1498.000	0.040	2.560	0.942	0.0033	0.006	0.51	0.058
25%	1537.930	57.725	2.660	27.480	0.0089	0.002	30.47	1.488
50%	1574.370	486.000	2.670	75.325	0.0192	0.005	34.45	2.492
75%	1608.560	2342.500	2.680	191.911	0.0516	0.013	37.31	4.566
Max	1641.70	14800.000	2.841	834.152	1.000	0.488	42.42	89.202

Skewness	-0.103	2.410	1.443	1.320	3.752	7.873	-1.758	7.977
Kurtosis	-1.183	6.628	8.751	3.181	20.280	70.892	1.788	90.523

The correlation of each feature is evaluated, and data transformation is also conducted, using the Yeo-Johnson method, only for porosity, as it increases the correlation by 0.05 to 0.21 with the targeted HFU. After transformation, the porosity feature is eliminated, leaving the transformed Porosity. In a classification problem, class balancing is a significant task for increasing the efficiency of a machine learning model. For this, data augmentation was also performed using random oversampling and a Gaussian noise-based method. FZI values are utilized for HFU classification in an unsupervised clustering model; however, further utilization of FZI values for supervised HFU classification can create data leakage and mislead the model's performance. That's why FZI values are also eliminated when performing the data-driven model.

3.3 Data Analysis

The input parameters of this study are Depth, Horizontal permeability, Grain density, pore throat size, specific area per grain volume, specific area per pore volume, grain size, CQI, sorting, and the target variable is HFU. The correlation heatmap in Fig. 4 demonstrates the correlation of significant reservoir properties. A strong positive correlation of 0.65 is observed between pore throat size and permeability, as well as between hydraulic flow units (HFU) and permeability (0.66), confirming that pore structure significantly controls fluid flow capacity. Similarly, grain size shows a moderate positive relationship with HFU (0.57), while surface area per grain volume is strongly linked with CQI (0.56). In contrast, CQI showed strong negative correlations with pore throat size (–0.77) and porosity-transformed (–0.75), suggesting that higher CQI values are associated with reduced pore connectivity. These relationships indicate that permeability, Porosity, and flow indices reflect complementary aspects of reservoir quality.

The histogram and KDE plots in Fig. 5 provide insights into the statistical distribution of each variable. Depth shows an approximately uniform spread across the sampled interval. In contrast, permeability, pore throat size, and surface area per grain volume, as well as surface area per pore volume, are highly right-skewed, reflecting strong heterogeneity and lower values with few extreme outliers. Grain density is approximately normally distributed, with a mean of 2.65 g/cm^3, consistent with quartz-rich lithologies. However, HFU, CQI, sorting, and grain size exhibited

discrete multimodal distributions, indicating their categorical nature, which is linked to depositional or facies-related variability. The occurrence of skewed and non-normal distributions emphasizes the need for data transformation and the application of advanced, non-linear modeling approaches to accurately capture reservoir heterogeneity.

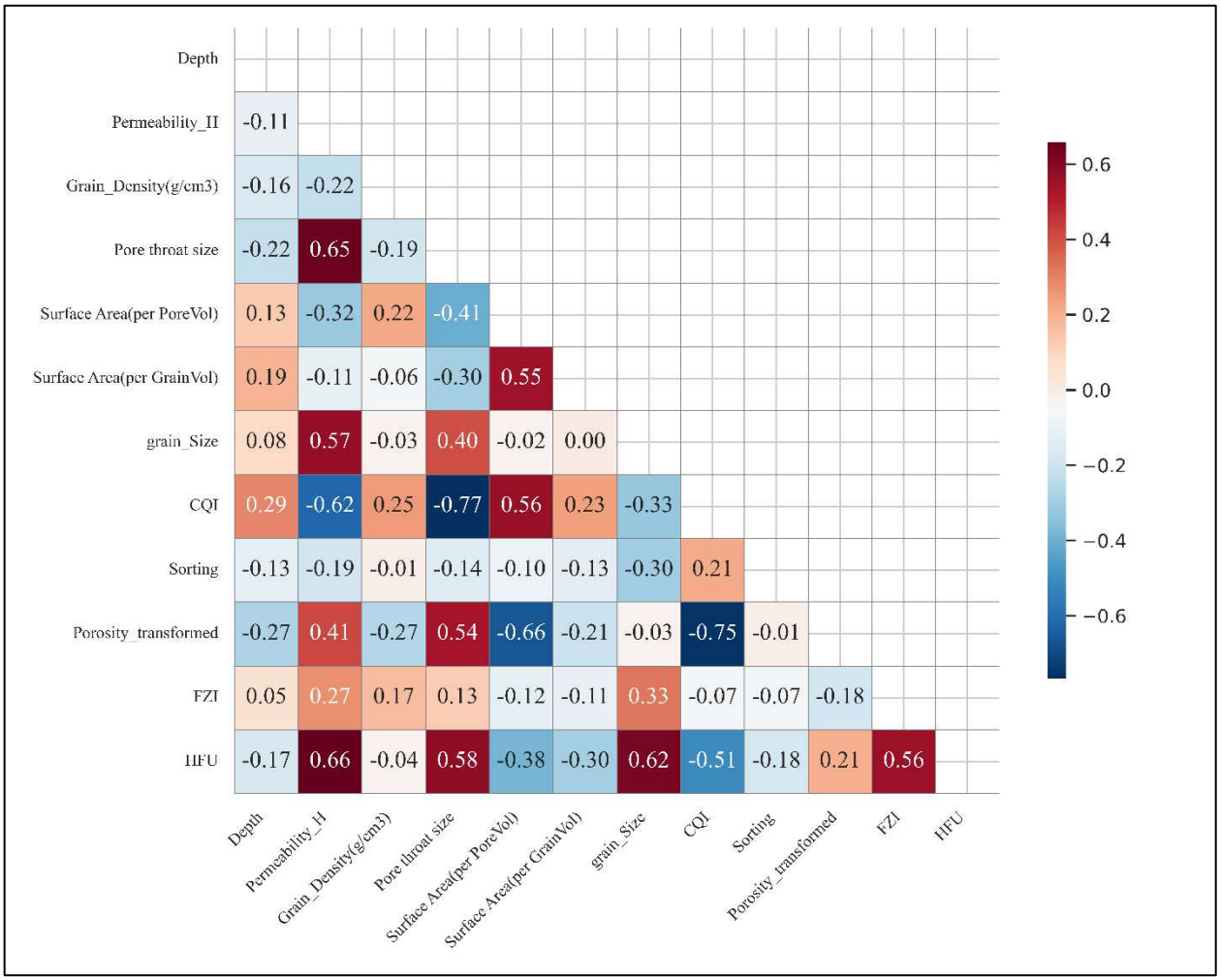


Fig.4: Pearson Correlation Matrix of Features

3.4 Data Transformation

The FZI and permeability demonstrate a non-normal distribution, as illustrated in Fig. 5. Therefore, merely identifying the HFU zone may yield incorrect results due to its very non-normal distribution (Astsauri et al., 2024). A technique is employed to convert the FZI and permeability values. Four transformation methods were employed: logarithmic transformation, square root transformation, Box-Cox transformation, and Yeo-Johnson transformation, which are visualized

in Figs. 6 and 7. These four transformations were evaluated based on the p-value of each approach and the Q-Q plot. The Q-Q plot indicated that the Yeo-Johnson transformation for FZI, with a corresponding p-value of 1.0200e-14, and the Box-Cox transformation for permeability, with a corresponding p-value of 3.2759e-03, are the best approaches. In contrast, logarithmic transformation, square root transformation, and Yeo-Johnson transformation showed p-values of 2.2121e-22, 1.8567e-26, and 5.0196e-16, respectively, for permeability. The transformations for FZI, which include the logarithmic transformation, square root transformation, and Box-Cox transformation, yielded p-values of 9.1007e-14, 8.5663e-31, and 9.7410e-06, respectively. These transformations aim to approximate a normal distribution, as illustrated in Figs. 6 and 7.

Table 2 illustrates an initial statistical description of the data. This provides key insights into the dataset: the mean represents the central tendency or average value, while the standard deviation (std) indicates how much the data varies around the mean. Skewness indicates the symmetry of the distribution, where positive values denote a longer right tail and negative values denote a longer left tail. Kurtosis signifies the heaviness of the tails; high kurtosis suggests the presence of extreme values or outliers, whereas low kurtosis indicates a flatter distribution with fewer extremes.

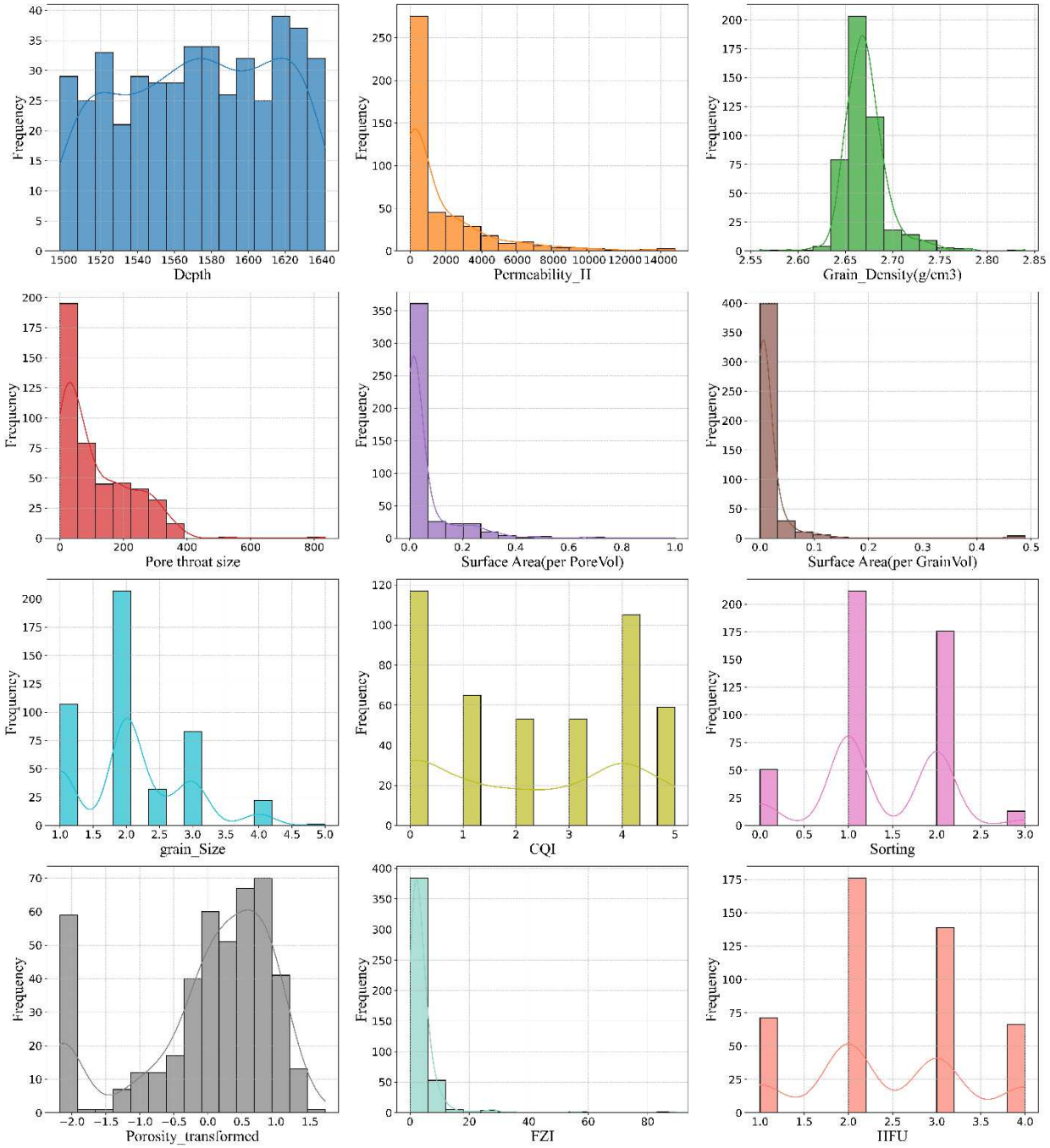


Fig. 5: Frequency Distributions of Input Features

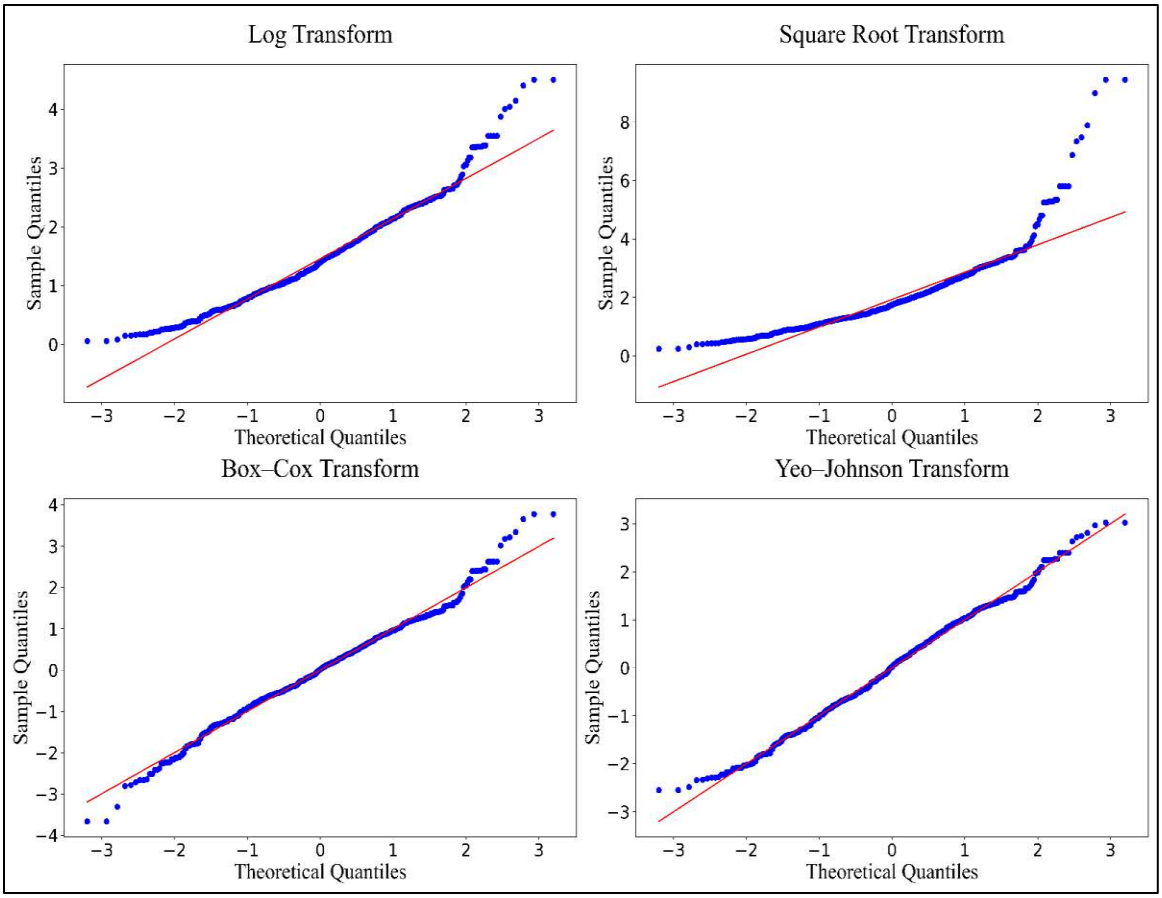


Fig. 6: Assessment of FZI Distribution Normality Across Different Transformation Techniques

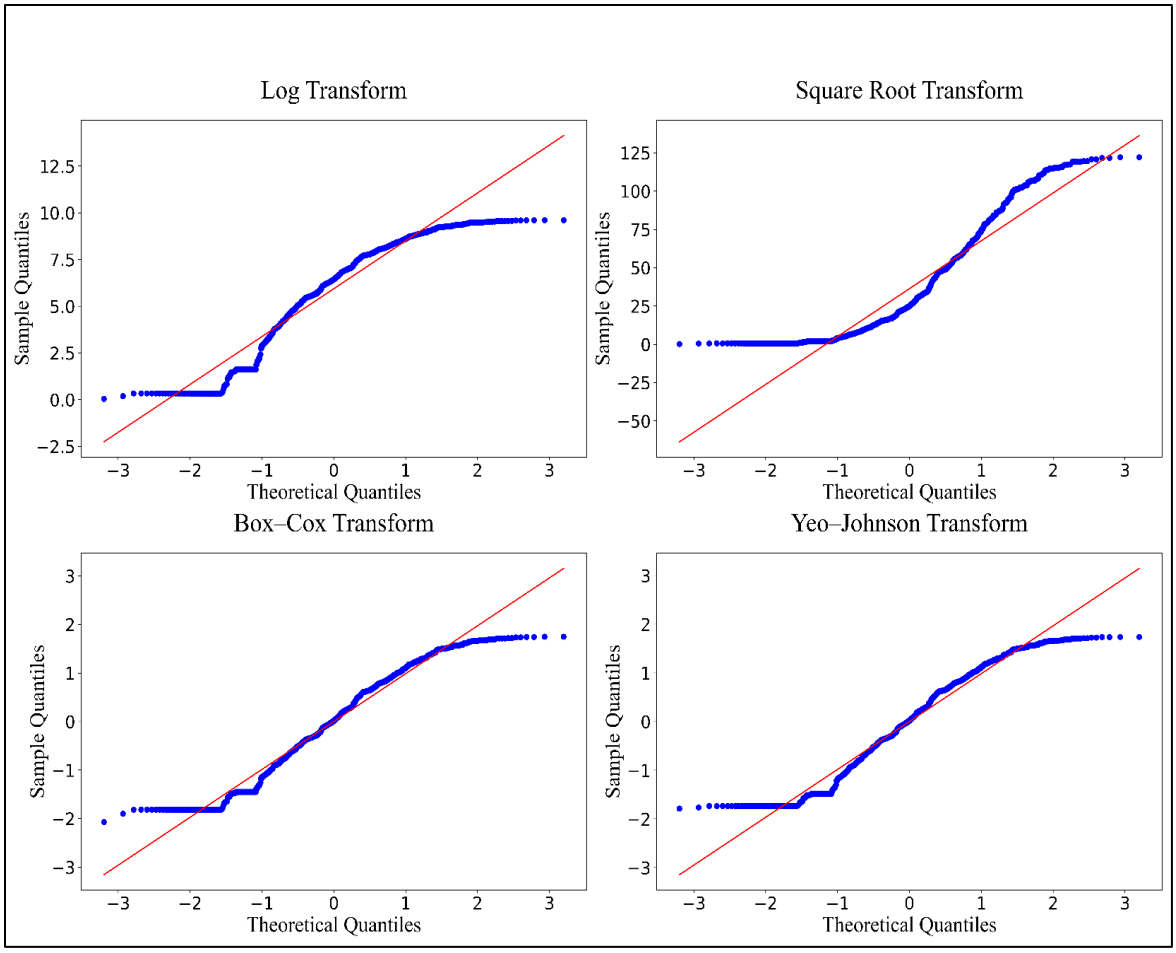


Fig. 7: Assessment of Horizontal Permeability Distribution Normality Across Different Transformation Techniques

4. Result and Discussion

4.1 Result

4.1.1 Traditional HFU classification

Determining the number of HFUs from the histogram is challenging, as the individual HFU distributions overlap. Accurate estimation requires isolating and identifying these individual distributions. Therefore, although the histogram illustrates the variability of HFU distribution within the formation, it serves as a qualitative tool and does not provide an exact number of HFUs (Abnavi et al., 2018). Statistical interpretation of log–log plots (RQI vs. ϕz) and the Z-score probability test was carried out, with data points of similar characteristics manually grouped into flow units. In Fig. 8 and Fig 9 shows a z-score probability test and log-log interpretation for

identifying the HFU class. The pore-to-matrix ratios (PMRs) from the sample belonging to the same hydraulic flow unit (HFU) fall on the same straight lines with a consistent slope in Fig. 9. The line corresponding to PMR=1 intersects each of the straight lines, and the corresponding RQI value indicates the flow zone indicator for each respective flow zone. High-capacity flow zone shows average FZI value 14.61 in log-log plot while low-capacity flow zone showed average FZI 0.59. This figure highlights, the regression lines show very high R2 score from 0.95 to 0.99, the clusters of HFU 4 to 6 clearly overlap, making it difficult to draw distinct boundaries. Small differences in slope and intercept shift the regression lines noticeably, leading to potential misclassification, while close average FZI values fails to provide reliable separation and reduces statistical robustness.

The Z-score probability test in Fig. 8 revealed difficulty in distinguishing HFU class, as the slopes of their straight-line trends are quite similar. This small difference may lead to ambiguity during visual interpretation, making it challenging to separate the two zones. However, this approach requires more time, is subjective, and is impractical for large datasets. It becomes unreliable when the dataset size is large, which can lead to overlapping points and inconsistent interpretations.

To overcome these limitations, this study further investigates unsupervised clustering algorithms, in the first phase, to automatically classify samples into HFUs based on FZI patterns. Second, a supervised classification phase trains machine learning models (RF, SVM, KNN, XGBoost, Gradient Boosting, Adaboost, and Stacking) using selected core features to predict HFUs in new data, which is the main objective of this study.

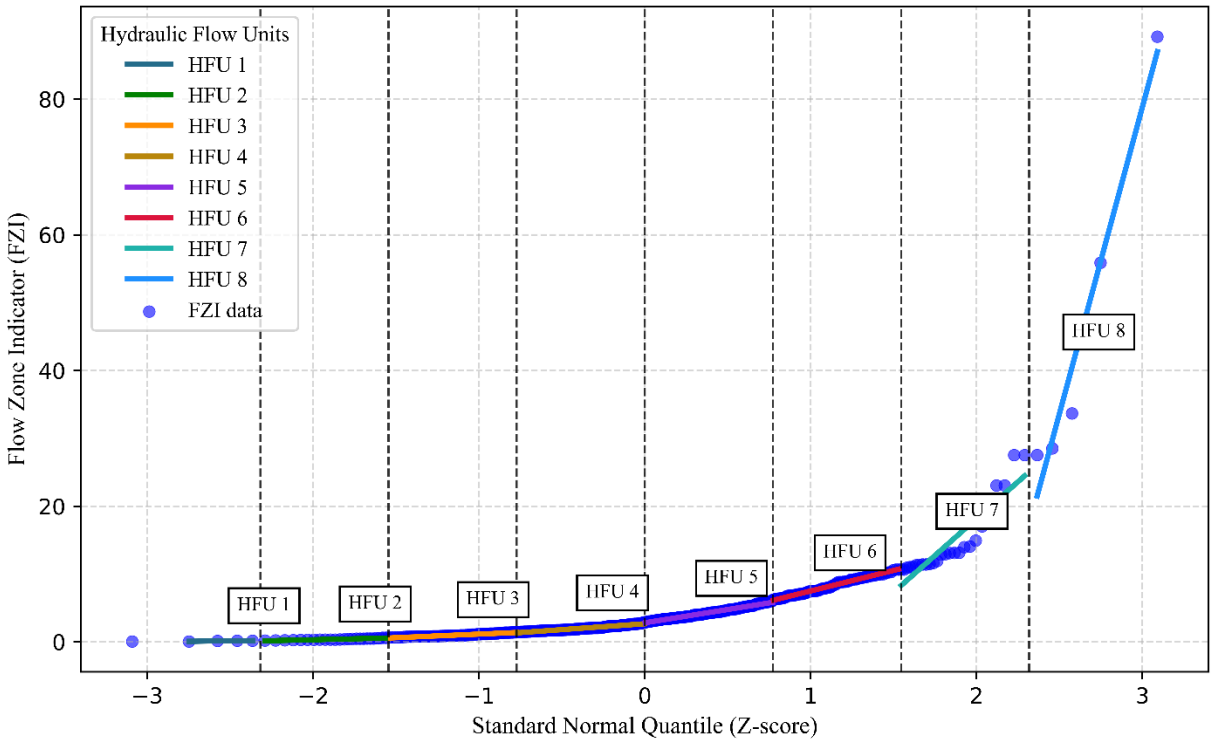


Fig. 8: Traditional HFU classification using Z score probability test

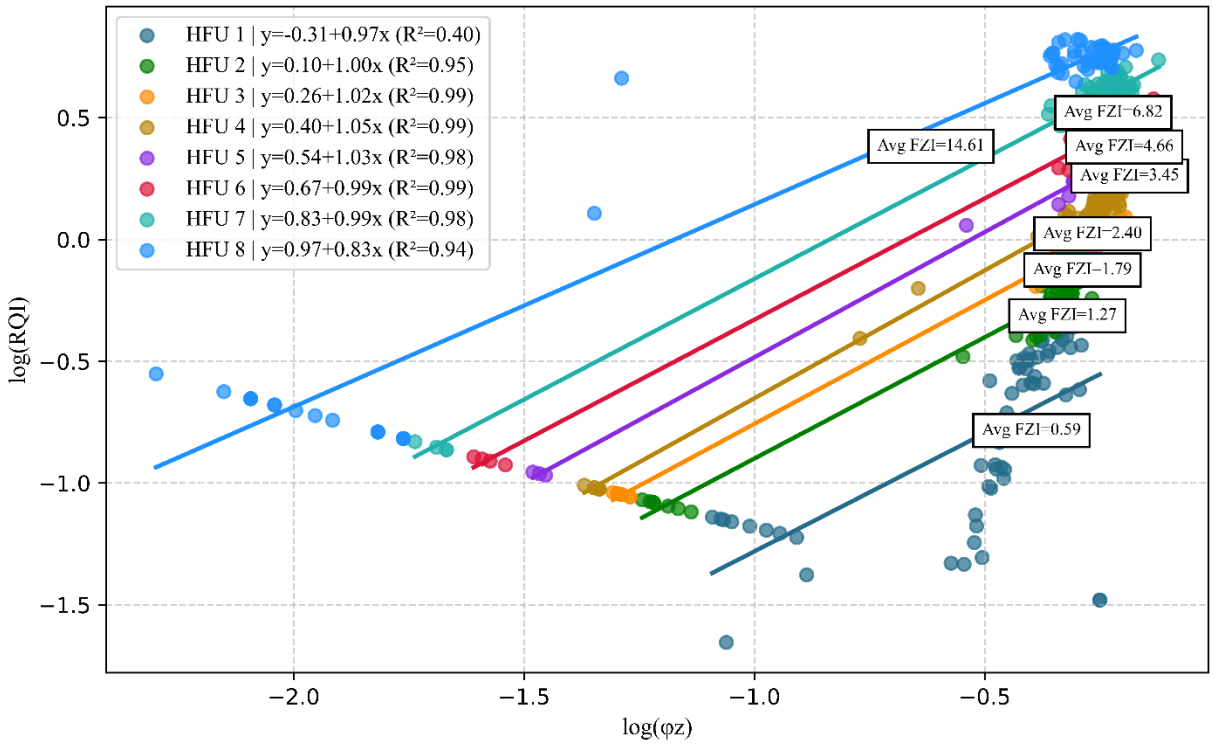


Fig. 9: Traditional HFU classification using log-log plot (RQI vs ϕz)

4.1.2 Unsupervised model

Identifying optimal HFU class:

This study employed a prominent unsupervised clustering algorithm to determine the optimal class of HFU. To detect HFU class automatically, this study investigates four popular clustering algorithms, K-means, K medoids, Fuzzy C-means, and one probabilistic model, GMM. The Optimal Number of HFU class are identified by using the Elbow method. Each of the algorithms was justified by two statistical metrics: R2 and RMSE score. The purpose of this optimization is to identify the most effective training method for clustering against the FZI and to define the optimal group of HFU.

Table 3: Statistical Performance of Permeability Prediction Methods Based on HFU Clustering

HFU	R^2 GMM	RMSE GMM	R^2 K-medoids	RMSE K-medoids	R^2 K-means	RMSE K-means	R2 FCM	RMSE FCM
2	0.802	0.5564	0.8018	0.5567	0.8021	0.5562	0.5438	0.8446
3	0.8865	0.4213	0.8747	0.4427	0.8819	0.4296	0.7468	0.6292
4	0.9276	0.3365	0.9244	0.3437	0.9274	0.3368	0.7836	0.5817
5	0.9512	0.2762	0.9437	0.2968	0.9498	0.2803	0.8688	0.4530
6	0.9588	0.2539	0.9571	0.2591	0.9681	0.2235	0.8949	0.4054
7	0.9659	0.2308	0.9653	0.2330	0.9751	0.1974	0.9037	0.3881
8	0.9807	0.1736	0.9680	0.2236	0.9813	0.1711	0.9403	0.3055
9	0.9830	0.1630	0.9830	0.1630	0.9834	0.1610	0.9448	0.2938
10	0.9844	0.1562	0.9849	0.1536	0.9852	0.1519	0.9563	0.2614

In the first step, reservoir was divided into groups of 2–10 HFU based on FZI pattern, and clusters were ordered based on their mean FZI to define HFUs from lowest to highest flow capacity. For each HFU, an average FZI value was assigned and subsequently used to estimate permeability as a function of porosity using equation (5). The predicted permeability values were then compared against core-measured permeability after applying a logarithmic transformation (Nguyen et al.,

2024). Table 3 presents the correlation coefficient (R^2) and root mean square error (RMSE) of actual permeability versus predicted permeability, calculated for each method within the clustering range of 2 to 10 HFU. These optimization techniques demonstrated that as the number of HFUs increases from 2 to 10, the correlation coefficient between actual permeability and predicted permeability for each method also increases, from 0.802 to 0.985. In contrast, the root mean square error decreases from 0.5564 to 0.1562.

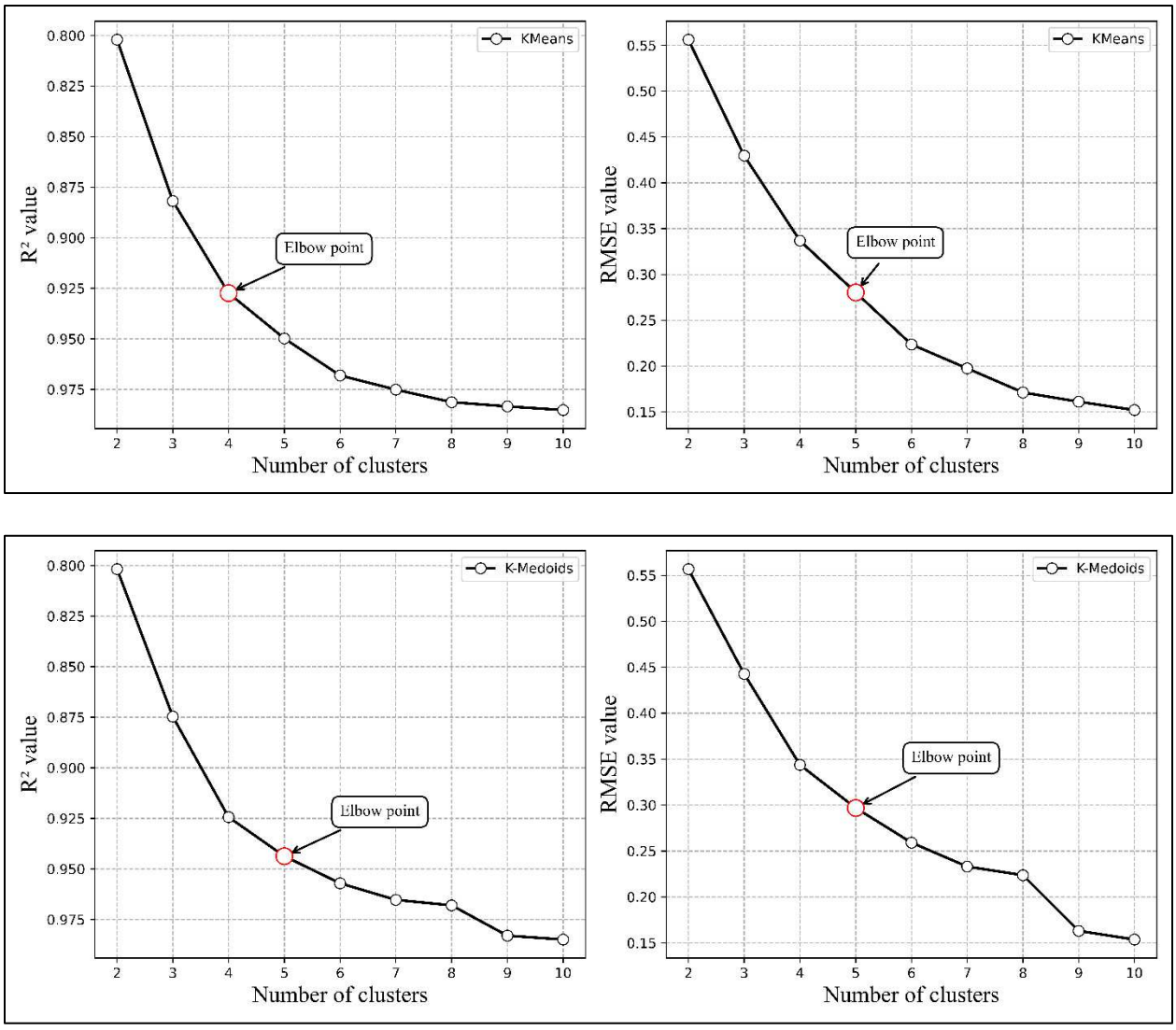


Fig. 10: Elbow Detection of Optimal HFU Clusters Based on R^2 and RMSE (a. K means, b. K medoids)

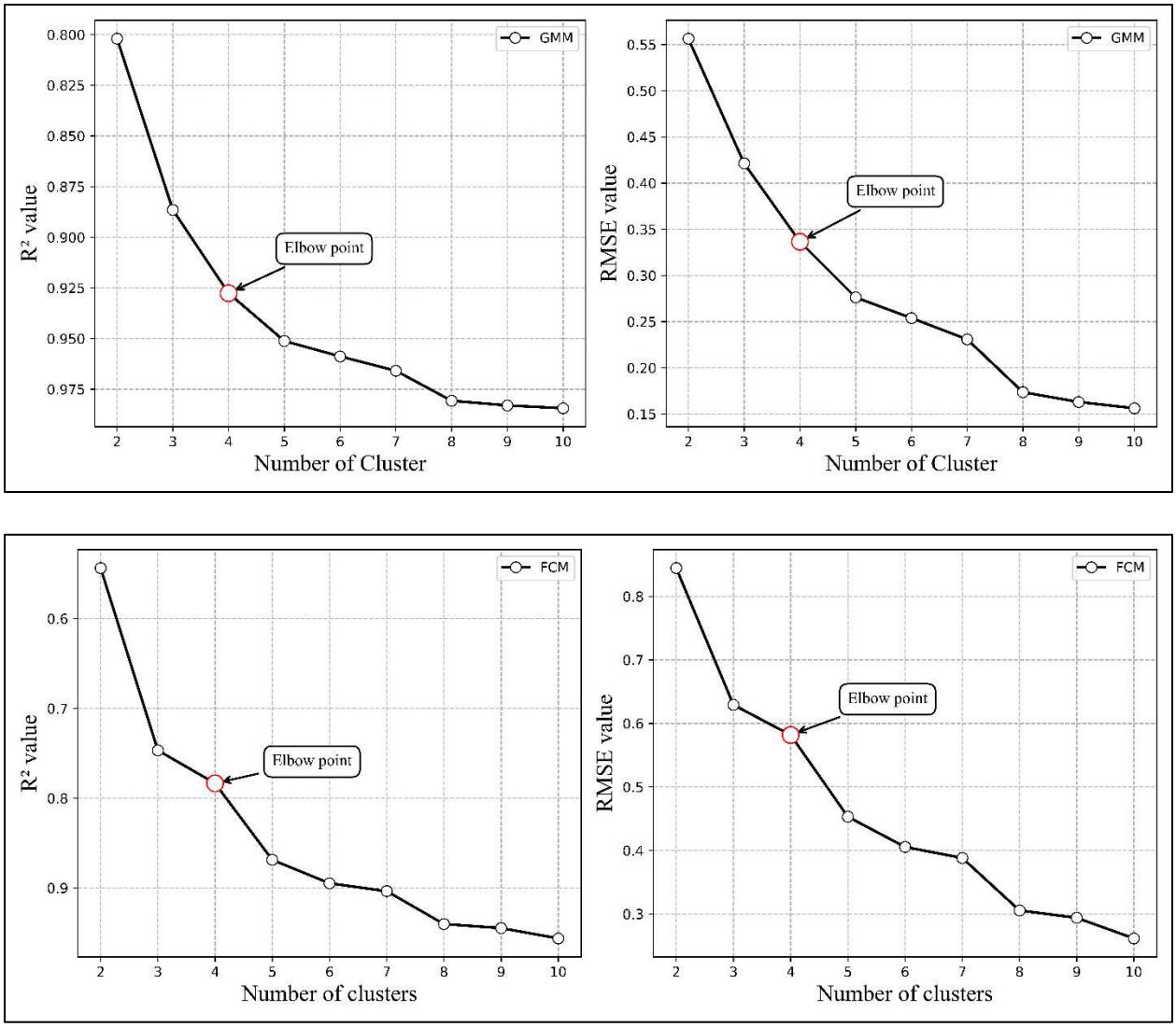
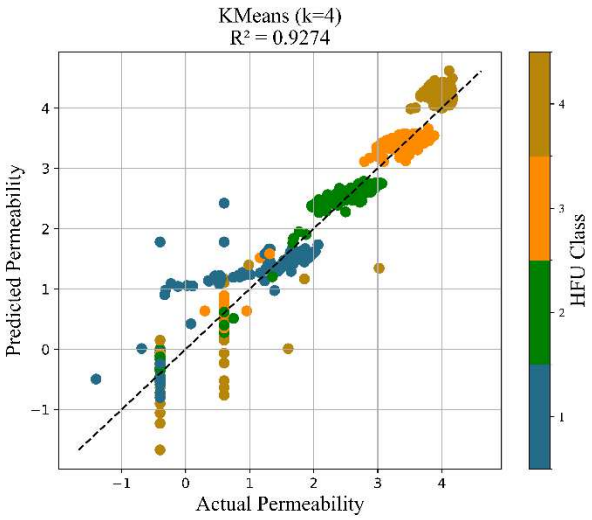
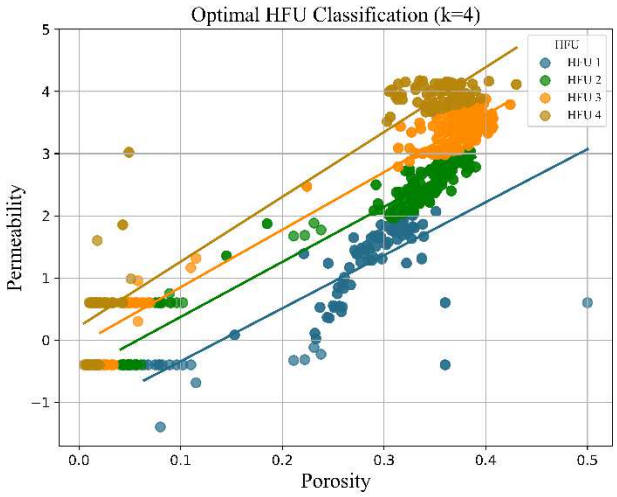
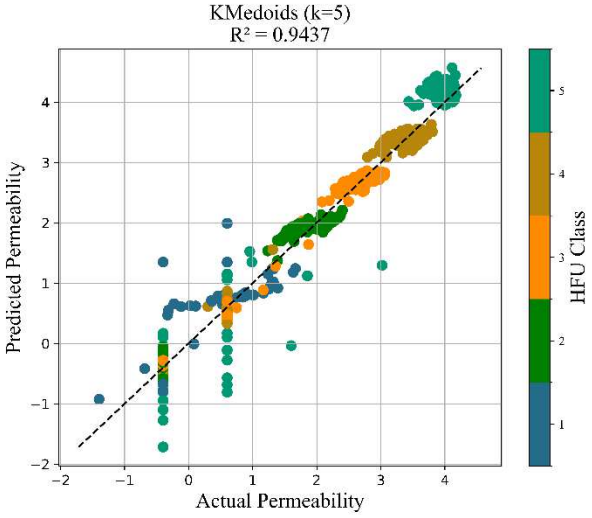


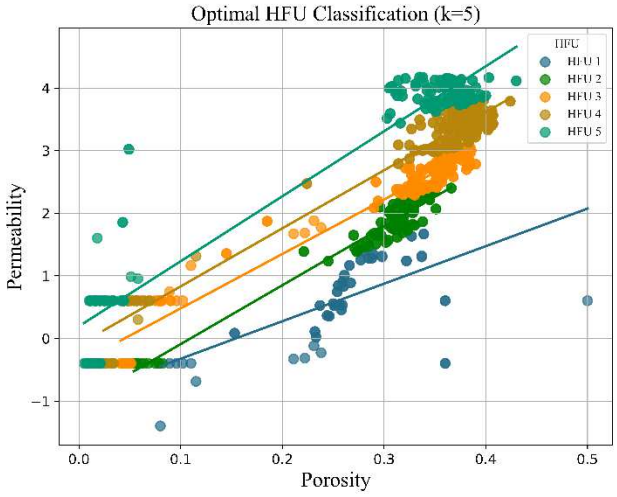
Fig. 10: Elbow Detection of Optimal HFU Clusters Based on R^2 and RMSE (c. GMM, d. FCM)

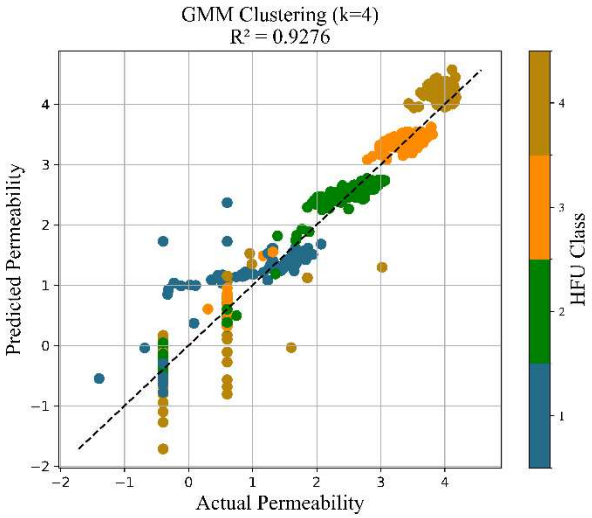
The GMM model outperforms the other three models, achieving an R^2 accuracy of 0.9278 and the lowest RMSE of 0.3365. The K-means and K-medoids models exhibit comparable performance to the GMM model, with an R^2 and RMSE 0.9274 and 0.3368 for K-means, 0.9244 and 0.3437 for K-medoids. Conversely, the Fuzzy C approach exhibits suboptimal performance, as indicated by its relatively low accuracy in the R^2 and RMSE metrics, which means this model is not suitable for identifying the optimal number of HFU classes. The comparative assessment of performance metrics reveals the following order of effectiveness: GMM > K-means > K-medoids > Fuzzy C-means. Graphical visualization in Fig.10, and 11 demonstrate Elbow detection and optimal HFU classification, comparing actual versus predicted permeability.

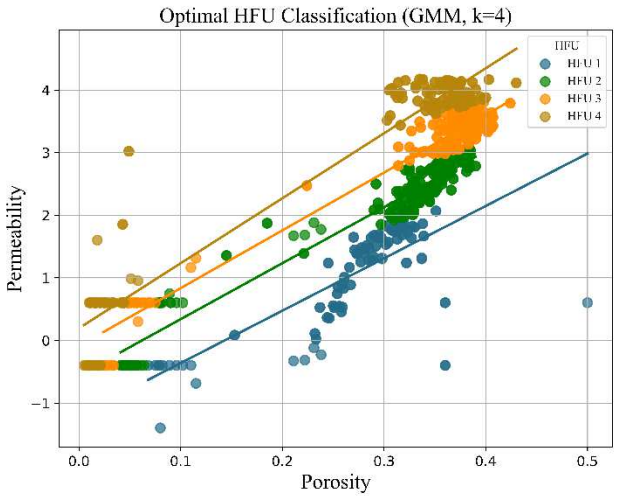

KMeans (k=4)
R² = 0.9274
Predicted Permeability
Actual Permeability
HFU Class


Optimal HFU Classification (k=4)
HFU
HFU 1
HFU 2
HFU 3
HFU 4
Permeability
Porosity


KMedoids (k=5)
R² = 0.9437
Predicted Permeability
Actual Permeability
HFU Class


Optimal HFU Classification (k=5)
HFU
HFU 1
HFU 2
HFU 3
HFU 4
HFU 5
Permeability
Porosity


GMM Clustering (k=4)
R² = 0.9276
Predicted Permeability
Actual Permeability
HFU Class


Optimal HFU Classification (GMM, k=4)
HFU
HFU 1
HFU 2
HFU 3
HFU 4
Permeability
Porosity

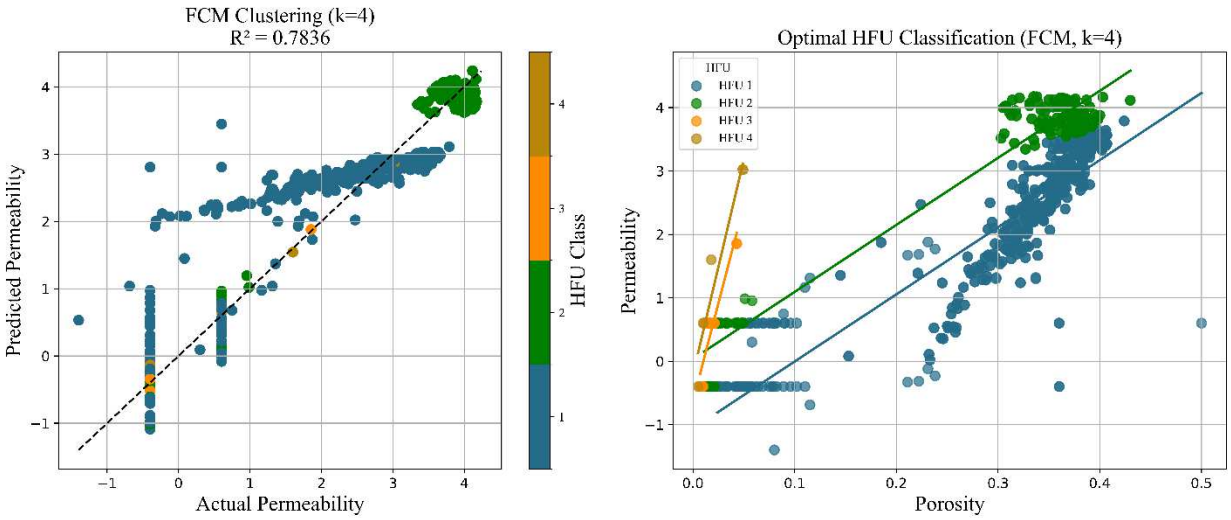


Fig. 11: Permeability prediction and hydraulic flow unit (HFU) classification results: (a) Predicted versus actual permeability, and (b) porosity–permeability cross-plot with HFU classes. (K means, K medoids, GMM, FCM)

4.1.3 Supervised Machine Learning Algorithm:

The main objective of this study is to predict the HFU class using a prominent supervised machine learning algorithm that leverages laboratory-derived core features. A total of seven supervised models are implemented for this classification problem. Supervised classification models, including KNN, RFC, SVM, three boosting models (XGBoost, AdaBoost, and Gradient Boosting), and one hybrid machine learning model, are employed. To increase efficiency and generalize the model output, the model hyperparameter configuration was also changed using an exhaustive grid search approach, as shown in Tables 7-12. Precision, recall, and F1 score for both the training and testing sets were also evaluated to investigate superiority in Table 5 and Table 6.

4.1.3.1 Evaluation of KNN, RF, and SVM models

According to Table 4, the K-Nearest Neighbors (KNN) model achieved an overall accuracy of 1.0 on the training set; however, its performance dropped significantly on the testing set, resulting in an accuracy of 0.878. This result suggests that the KNN model is likely overfitting the training data, as it accurately memorizes the training class but fails to generalize equally well to unseen data. Model hyperparameters are tuned to achieve optimal performance with the generalizing

model, as shown in Table 7. The KNN model provides the best model configuration as knn__n_neighbors: 5, knn__weights: distance, and knn__metric: Euclidean.

Table 4: Evaluation of Training and Testing Accuracy Across Applied Models

Model	Accuracy (test)	Accuracy (Train)
KNN	0.878	1.000
RF	0.972	0.983
SVM	0.959	0.971
Gradient boosting	0.934	1.000
XGBoost	0.951	1.000
Adaboost	0.904	0.916
Stacking	0.951	1.000

Further investigation of model generalization precision, recall, and F1-score is also evaluated for both the training and testing sets. This evaluation showed that the model performs best in the training set, achieving 100% accuracy. However, in the generalization test, the precision, recall, and F1-score significantly dropped for HFU classes 2 and 3, while showing generalized results for HFU classes 1 and 4. As the data set is augmented and there is no possibility of class imbalance, it has been proven that the KNN model misclassifies the HFU classes 2 and 3 for new, unseen data. In contrast, the RFC model achieved an overall accuracy of 0.983 for the training set and 0.972 for the testing set, demonstrating that the RFC is more generalized than the KNN model and exhibits no significant overfitting. This evaluation is justified by examining the test accuracy of each class, which shows that the average precision, recall, and F1-score for HFU classes are 0.972, 0.973, and 0.9725, respectively.

On the other hand, the SVM model exhibits superior performance, achieving an accuracy of 0.971 for the training set and 0.959 for the testing set. The SVM model demonstrated moderate accuracy, albeit with slight overfitting, as evidenced by the drop in accuracy from training to testing. Precision, recall, and F-1 score also proved the superiority of this model by providing high accuracy for HFU classes 1, 3, and 4, but slightly lower in HFU class 1.

According to this comparative analysis of the three models, the RFC model outperforms the other two models, exhibiting reduced overfitting and greater generalizability to new, unseen data. Where SVM shows the second-highest accuracy with moderate performance, this investigation proved that the KNN model is not suitable for classifying the HFU class.

Table 5: Assessment of Model Performance Using Precision, Recall, and F1-Score (Train set)

Model	Score	HFU Level			
		1	2	3	4
KNN	Precision	1.000	1.000	1.000	1.000
	Recall	1.000	1.000	1.000	1.000
	F1- Score	1.000	1.000	1.000	1.000
RF	Precision	0.982	0.983	0.978	0.990
	Recall	1.000	0.967	0.967	1.000
	F1- Score	0.991	0.975	0.972	0.995
SVM	Precision	0.914	0.984	1.000	0.991
	Recall	0.983	0.921	0.990	1.000
	F1- Score	0.947	0.952	0.995	0.995
Gradient Boosting	Precision	1.000	1.000	1.000	1.000
	Recall	1.000	1.000	1.000	1.000
	F1- Score	1.000	1.000	1.000	1.000
XGBoost	Precision	1.000	1.000	1.000	1.000
	Recall	1.000	1.000	1.000	1.000
	F1- Score	1.000	1.000	1.000	1.000
ADABoost	Precision	0.975	0.860	0.946	0.980
	Recall	0.701	0.971	0.954	0.924
	F1- Score	0.816	0.912	0.950	0.951
Stacking	Precision	1.000	1.000	1.000	1.000
	Recall	1.000	1.000	1.000	1.000
	F1- Score	1.000	1.000	1.000	1.000

4.1.3.2 Evaluation of Boosting Model

Three boosting algorithms were employed and their performance was evaluated using, precision, recall, and F1-score, as presented in Tables 4, 5, and 6 for both the training and testing datasets. As shown in Table 4, Gradient Boosting and XGBoost yielded training accuracies of 1.0 and testing accuracies of 0.934 and 0.9513, respectively. At the same time, the AdaBoost model exhibits inferior performance with an accuracy of 0.916 for the training set and 0.904 for the testing set. The results of the three models demonstrate that the gradient boosting and XGBoost models exhibit significant overfitting with the lowest generalization. In contrast, the AdaBoost model shows less overfitting compared to the other two. The overall accuracy of these three models demonstrated that XGBoost outperformed the other models, with lower generalization error.

Table 6: Assessment of Model Performance Using Precision, Recall, and F1-Score (Test set)

Model	Score	HFU Level			
		1	2	3	4
KNN	Precision	0.909	0.892	0.750	0.966
	Recall	1.000	0.714	0.857	0.966
	F1- Score	0.952	0.793	0.800	0.966
RF	Precision	0.972	0.980	0.957	0.979
	Recall	0.972	0.944	0.978	1.000
	F1- Score	0.972	0.962	0.967	0.989
SVM	Precision	0.966	0.916	1.000	0.967
	Recall	0.966	0.942	0.928	1.000
	F1- Score	0.966	0.929	0.962	0.983
Gradient Boosting	Precision	0.967	0.894	0.923	0.964
	Recall	1.000	0.971	0.833	0.900
	F1- Score	0.983	0.931	0.888	0.931
XGBoost	Precision	1.000	0.912	0.972	0.936
	Recall	1.000	0.981	0.833	0.977
	F1- Score	1.000	0.945	0.897	0.956
ADABoost	Precision	1.000	0.897	0.962	0.928

	Recall	0.785	0.972	0.928	1.000
	F1- Score	0.880	0.933	0.945	0.962
Stacking	Precision	1.000	0.912	0.972	0.936
	Recall	1.000	0.981	0.833	0.977
	F1- Score	1.000	0.945	0.897	0.956

Further investigation of model performance on the training set revealed that Gradient Boosting and XGBoost achieved 100% precision, recall, and F1 score for classifying HFU, whereas the AdaBoost model showed slightly lower precision, recall and F1 score, achieving 0.940, 0.8875, and 0.907, respectively. For the testing set, XGBoost achieved 100% precision accuracy for HFU class 1 and showed superior accuracy for other classes as well. The analysis of evaluation metrics concluded that XGBoost outperforms the other two boosting algorithms, achieving the highest accuracy, while the AdaBoost model exhibits poorer performance but with higher generalization capability.

4.1.3.3 Evaluation of Hybrid Stacking Model

The stacking model was built using RFC and XGBoost. These two algorithms are the best-performing algorithms in this study. The overall classification accuracy obtained by the stacking model is 1 for training and 0.951 for testing. The model accuracy dropped nearly 5% on new unseen data, which suggests significant overfitting. The precision, recall, and F1 score of this model are 1 for all classes in the training set, while the score is significantly lower for HFU class 3 in the testing set. This indicates that the stacking model exhibits significant overfitting, which compromises its generalization capacity.

Table 7: Hyperparameter Settings of the KNN

Parameter	Minimum	Maximum	Best
knn__n_neighbors	3	9	5
knn__weights	uniform	distance	distance
knn__metric	Euclidean	Euclidean	Euclidean

Table 8: Hyperparameter Settings of the RF

Parameter	Minimum	Maximum	Best
n_estimators	10	30	30
max_depth	10	30	4
min_samples_split	5	10	5
min_samples_leaf	1	2	2

Table 9: Hyperparameter Settings of the SVM Models

Parameter	Minimum	Maximum	Best
C	1	100	100
gamma	scale	0.001	scale
kernel	rbf	linear	linear

Table 10: Hyperparameter Settings of the XGBoost

Parameter	Minimum	Maximum	Best
max_depth	3	5	4
learning_rate	0.1	0.3	0.3
n_estimators	50	100	50

Table 11: Hyperparameter Settings of GB Models

Parameter	Minimum	Maximum	Best
n_estimators	10	100	100
max_depth	3	5	None
min_samples_split	2		2
max_leaf_nodes	none	7	7

Table12: Hyperparameter Settings of the Adaboost

Parameter	Minimum	Maximum	Best
n_estimators	50	100	100
learning_rate	0.01	1.0	1
algorithm	SAMME		SAMME

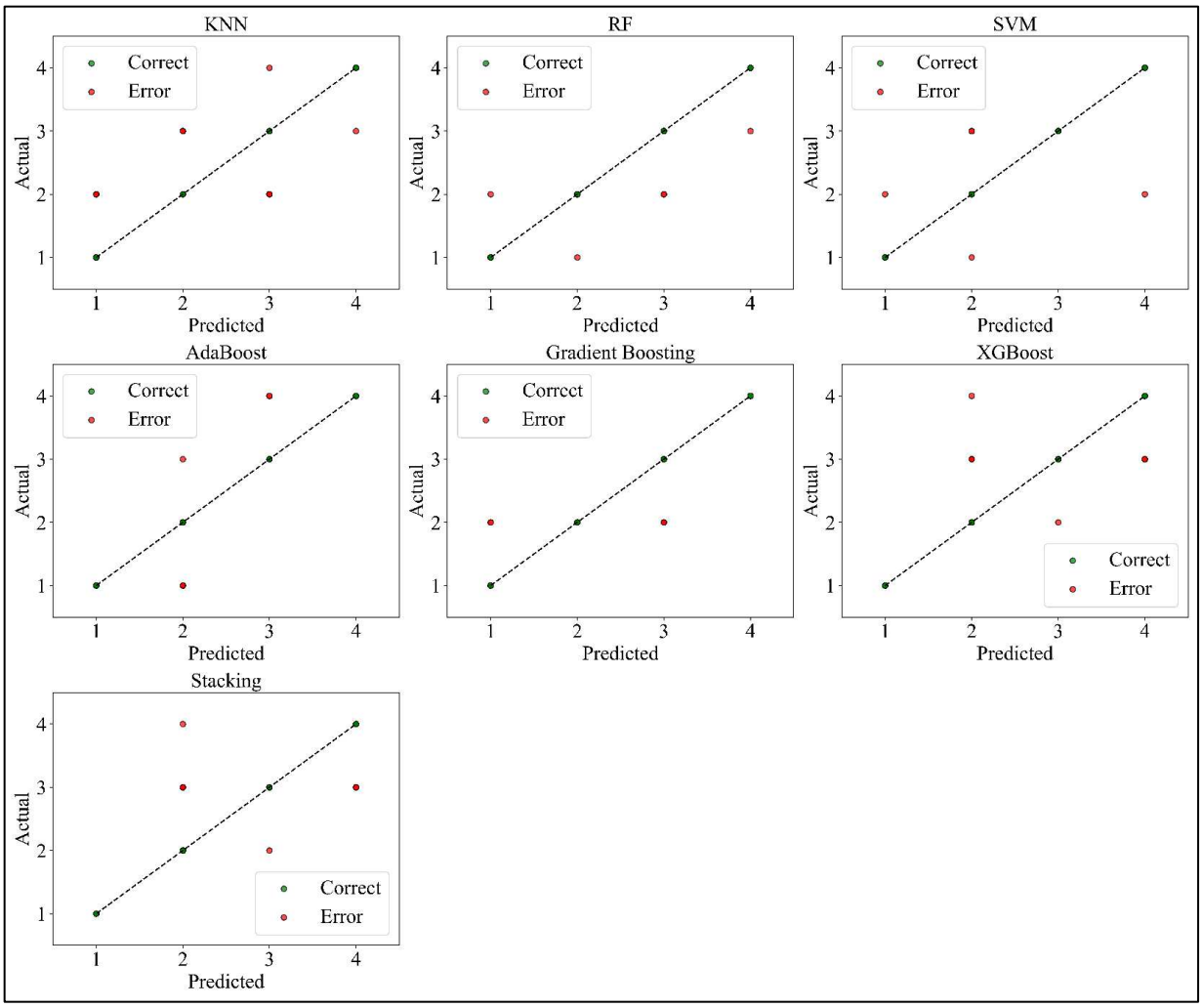


Fig. 12: Predicted vs Actual HFU Class Cross-Plots for testing set a) KNN, b) RF, c) SVM, d) Adaboost, e) Gradient Boost, f) XGBoost, g) stacking

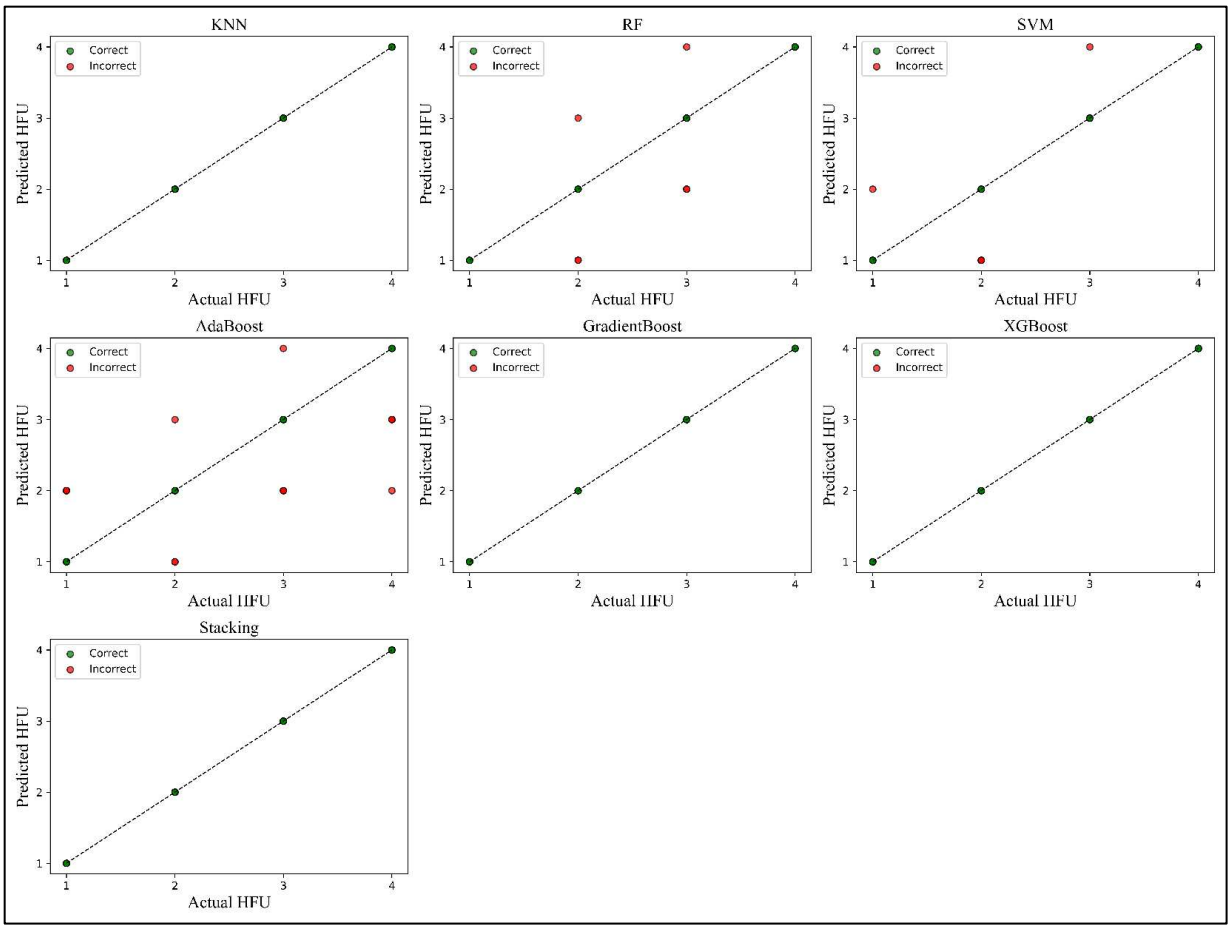


Fig. 13: Predicted vs Actual HFU Class Cross-Plots for training set (KNN, RF, SVM, Adaboost, Gradient Boost, XGBoost and stacking)

Table 13: Computational cost across all model

Model	Computational efficiency (seconds)
KNN	2.66
RF	7.27
SVM	3.97
XGBoost	34.92
Gradient Boosting	14.67
Adaboost	7.94
Stacking	24.99

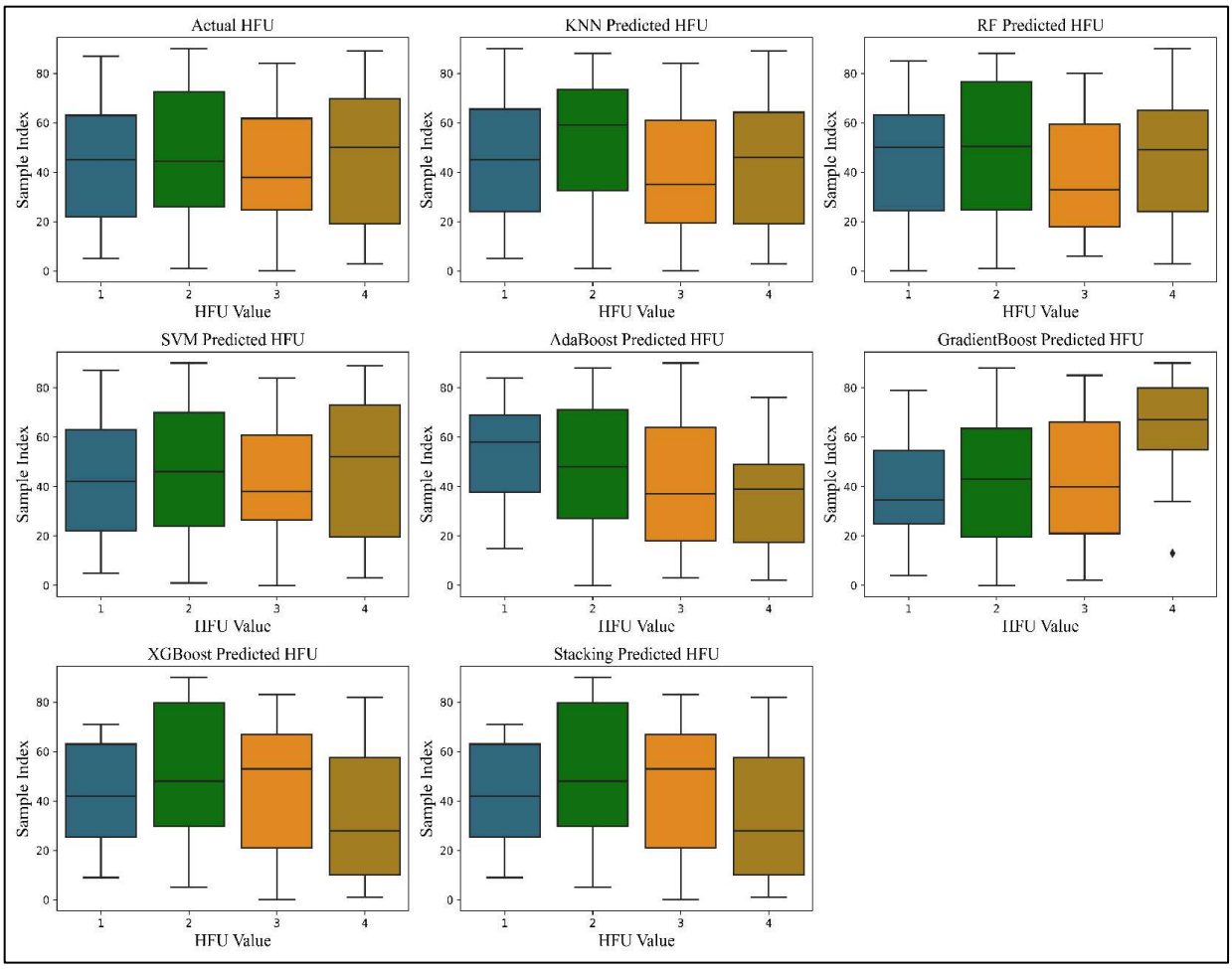


Fig. 14: Comparative box plot of predicted versus actual HFU classifications on the test set

4.1.4 Computational efficiency

The computational cost of each model is also evaluated by measuring training time during tuning. According to Table 13, the KNN and SVM models showed the lowest computational cost, while the XGBoost and stacking models showed the highest iterative computation time due to their inherent complexity. For XGBoost, the sequential boosting process involves building multiple trees iteratively, where each tree corrects the errors of the previous one, combined with gradient-based optimization using second-order derivatives (Chen et al., 2016), which substantially increases computation. In contrast, stacking training XGBoost and RFC as base learners, followed by the construction of a meta-learner LG on their predictions, and incorporating 10-fold cross-validation to ensure robustness, all of which affect training time. RF and AdaBoost showed moderate

computation costs of 7.27 and 7.94 seconds, respectively, with relatively lower iterative complexity compared to boosting algorithms. Gradient Boosting, however, required 14.67 seconds due to its stepwise tree-building process, where each tree is trained to correct the residuals of the previous one, making it more computationally demanding than RF and AdaBoost.

Although KNN (2.33 s) and SVM (2.33 s) required less computation time than RF (7.27 s), the Random Forest achieved higher predictive accuracy, highlighting a trade-off between computational cost and model performance. In other words, while KNN and SVM were faster, RF's slightly higher computation time was justified by its superior accuracy, making it more efficient overall in terms of accuracy-time balance.

4.2 Discussion

The results of this study highlight the efficiency of combining unsupervised and supervised machine learning approaches in optimizing and predicting Hydraulic Flow Units (HFU) in the Norwegian sector of the North Sea. The unsupervised clustering analysis demonstrated that probabilistic models, particularly Gaussian Mixture Models (GMM), are highly effective in identifying the optimal number of HFU classes. The systematic increase in correlation coefficient (R^2) from 0.802 to 0.985 and the concurrent decrease in RMSE from 0.5564 to 0.1562 occurred as the number of HFU classes increased from 2 to 10. Fig. 9–11 illustrates the elbow detection using R^2 and RMSE scores, along with the flow zone distribution for each class. Among the clustering algorithms, GMM consistently provides superior performance than K-means, K-medoids, and Fuzzy C-means model, suggesting that probabilistic distribution of FZI values yields the most accurate HFU classification. The inferior performance of Fuzzy C-means indicates its limited suitability for identifying optimal HFU class, likely due to overlapping cluster boundaries and lower sensitivity to variations in FZI.

The supervised data-driven machine learning approaches are implemented for HFU classification. Suggesting HFU classes from the GMM model are aligned with significant core features, and supervised machine learning algorithms are performed to automatically detect the HFU class. The evaluation metrics of the supervised model are presented in Table 4, indicating that the RFC model outperforms all other models, achieving overall accuracies of 0.983 and 0.972 for the training and

testing sets, respectively. The generalization ability and robustness of the models were evaluated by changing their configurations through an exhaustive grid search, as summarized in Tables 7–12, combined with 10-fold cross-validation. In Fig. 12 and 13 visualized actual vs predicted cross plot both training and testing set for all models that showed how each class performed.

In Fig. 15, the confusion matrix illustrates the distribution of true positives, true negatives, false positives, and false negatives, demonstrating that the KNN model exhibits the lowest accuracy with significant overfitting compared to all other models across both the training and testing sets. Support Vector Machine (SVM) models also showed robust performance, though with slightly lower generalization than RFC, indicating that kernel-based methods remain competitive but sensitive to hyperparameter tuning.

The gradient boosting model achieves an accuracy of 0.93 on the testing set, while the training accuracy is 1. This indicates that the model performs poorly on new, unseen data and misclassifies the HFU class. XGBoost and AdaBoost demonstrated moderate performance, with training accuracies of 0.916 and 0.916 for the training set, respectively. The test accuracies were 0.956 and 0.902, respectively. AdaBoost demonstrated lower training accuracy but better generalization. The hybrid stacking model, which combined RFC and XGBoost, achieved near-perfect training accuracy; however, the model suffered a notable drop (~5%) on unseen data, particularly for the HFU class 3. This overfitting challenge of hybrid machine learning modeling in reservoir HFU classification.

Computational efficiency analysis, as shown in Table 13, indicates that KNN and SVM are computationally inexpensive but less robust in terms of predictive accuracy. On the other hand, ensemble and hybrid models, particularly boosting algorithms and Stacking, required significantly more iterative time but achieved higher accuracy, with significant overfitting. Random Forests, offering a balance between computation and accuracy, emerge as a novel model for reservoir flow zone classification in the context of reservoir characterization.

The findings of this study emphasize the importance of selecting algorithms based on their predictive accuracy, robustness, generalization capability, and computational time. In practice, RFC and XGBoost emerge as the most reliable classifiers, while GMM provides the most robust unsupervised grouping, enabling informed decisions for reservoir modeling and flow simulation in the Norwegian North Sea sector.

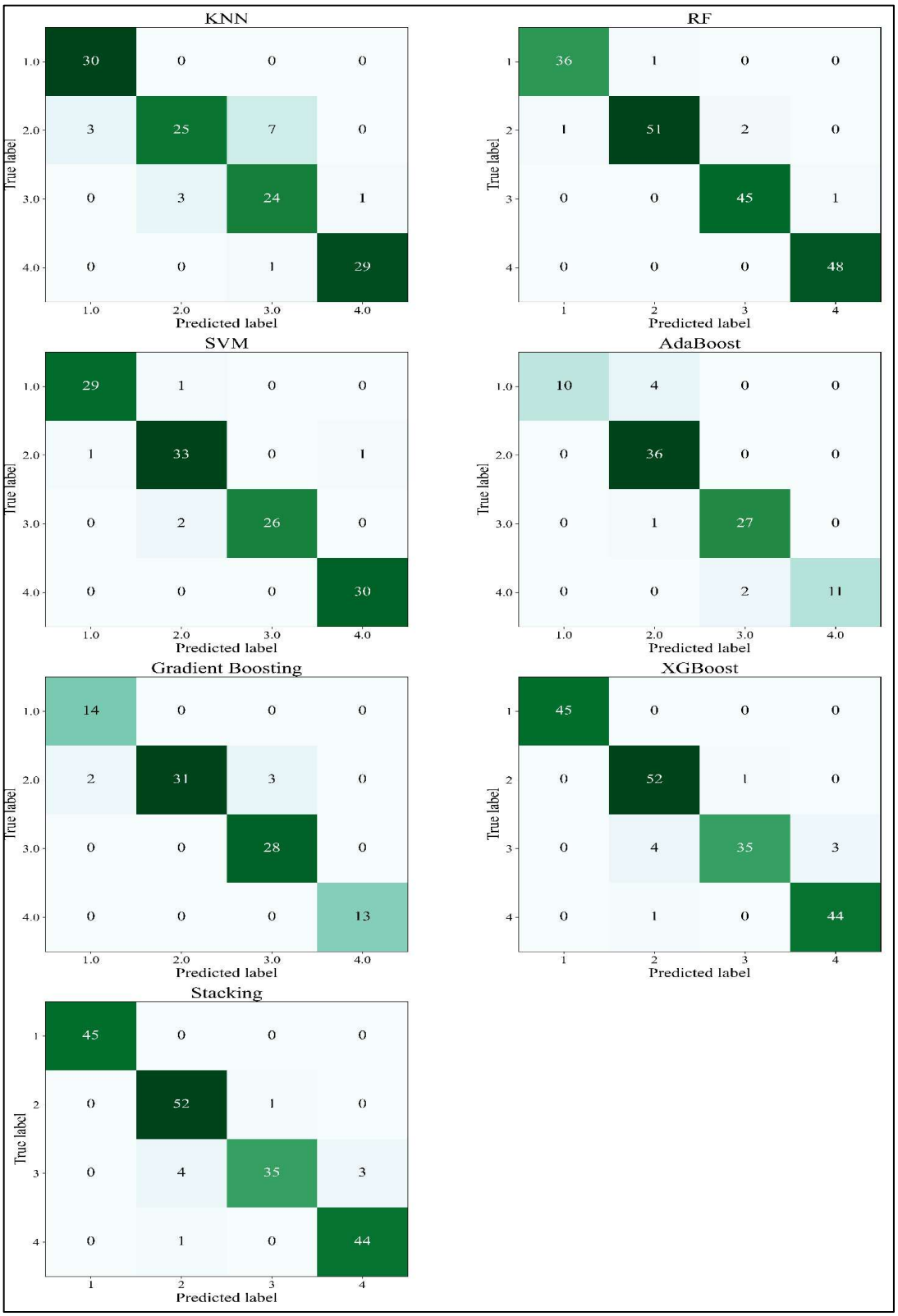


Fig. 15: Confusion matrix (RFC, KNN, SVM, XGBoost, Adaboost, Gradient boost, and Stacking) in HFU classification.

4.3 Model Ranking

The comparative ranking of machine learning models reveals notable differences in their performance for HFU classification. The ranking results indicate that Random Forest is the best-performing model, achieving the top position across all four metrics and the highest overall rank score of 28. XGBoost and Stacking followed closely, both with a rank score of 22, showing strong and consistent performance across metrics. Gradient Boosting ranked next with a score of 18, driven by its high recall rank of 2, which highlights its strength in detecting minority classes. SVM achieved a moderate rank score of 13, reflecting balanced performance but not outperforming ensemble models. AdaBoost scored 10, performing well in precision but weaker in recall. Finally, KNN ranked lowest with a score of 4, consistently placing last in all metrics, which shows its sensitivity to noise and reduced generalization. Overall, ensemble methods, especially Random Forest, XGBoost, and Stacking, consistently outperformed traditional classifiers in HFU classification.

Table 14: Model Ranking by manual approach:

Model	Accuracy	Precision	Recall	F1 Score	Accuracy Rank	Precision Rank	Recall Rank	F1 Score Rank	Rank Score
KNN	0.8681	0.8712	0.8797	0.870	7	7	7	7	4
RF	0.9780	0.9772	0.9807	0.978	1	1	1	1	28
SVM	0.9450	0.9513	0.9452	0.947	4	5	5	5	13
AdaBoost	0.9230	0.9522	0.8811	0.908	6	4	6	6	10
GB	0.9450	0.9445	0.9652	0.951	4	6	2	2	18
XGBoost	0.9560	0.9540	0.9492	0.949	2	2	3	3	22
Stacking	0.9560	0.9540	0.9492	0.949	2	2	3	3	22

4.4 Feature importance analysis

According to Table 14, it is demonstrated that the RFC model outperforms all other applied algorithms, including KNN, SVM, XGBoost, AdaBoost, Gradient Boost, and stacking, in classifying HFUs. To further examine how each input parameter contributes to model performance, a feature importance analysis was conducted. The ranking of features was determined by eliminating each feature and evaluating its impact on model accuracy. Fig. 16

illustrates the importance of shape features, revealing how each parameter contributes to predicting each of the HFU classes. Both Fig. 16 and Fig. 17 reveal that Horizontal Permeability has the highest influence on HFU classification, followed by Surface Area (per PoreVol) and Porosity_transformed. At the same time, parameters such as depth, sorting, and grain density demonstrate minimal impact. To enhance interpretability, SHAP (Shapley Additive exPlanations) values were employed with the best-performing RFC model. SHAP analysis provides both global and local explanations of feature contributions, capturing complex non-linear interactions.

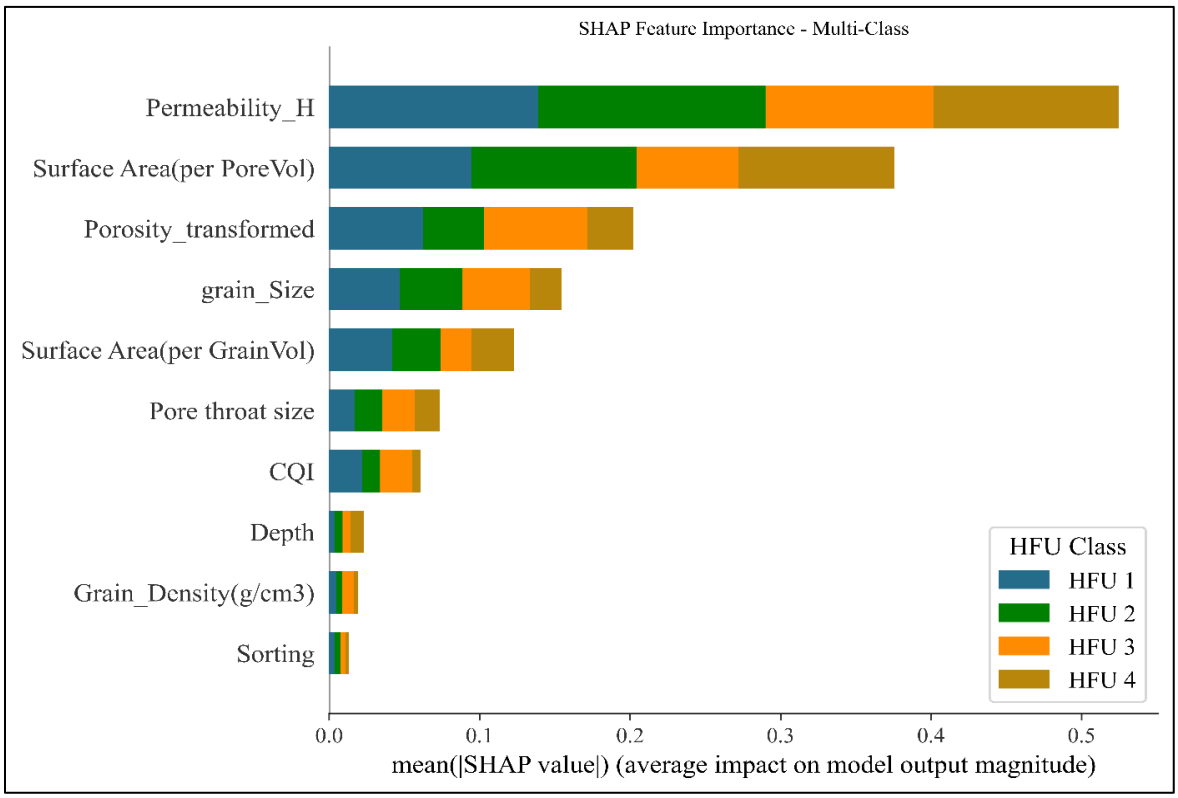


Fig. 16: Global feature importance ranking based on mean SHAP values

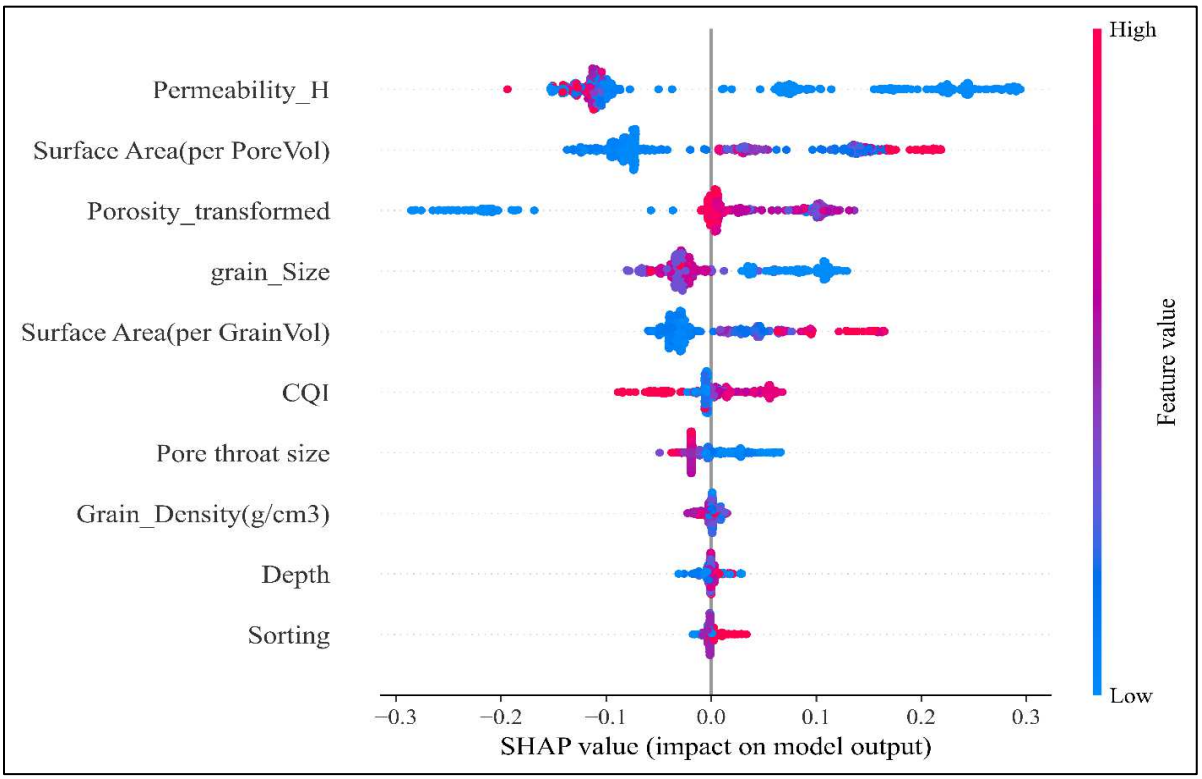


Fig. 17: SHAP bees warm plot on HFU classification

5. Conclusion

This study highlights the limitations of traditional qualitative and semi-quantitative techniques in identifying hydraulic flow units (HFUs). It demonstrates the advantages of adopting machine learning approaches for accurate and scalable classification. Histogram analysis, z-score probability test and statistical log-log plots provide valuable initial insights but remain inherently subjective, especially when applied to large and complex datasets. By integrating unsupervised clustering algorithms and supervised machine learning models, this research delivers a robust workflow that enhances precision, reproducibility, and efficiency in HFU identification.

Unsupervised clustering served as a practical first step to determine the optimal number of HFUs. Among the tested models, Gaussian Mixture Models (GMMs) outperformed the others, achieving the highest predictive R2 accuracy 0.9276 and the lowest RMSE 0.3365. This finding confirms the potential of probabilistic models in handling overlapping data distributions, where conventional clustering methods fall short.

The second phase applied supervised learning to predict HFU classes in new datasets, revealing that Random Forest consistently outperformed other classifiers with an excellent balance between training and testing accuracy 0.983 and 0.972. Although boosting models like XGBoost and Gradient Boosting demonstrated high accuracy in training, they were prone to overfitting, limiting their reliability for generalization. Conversely, AdaBoost, despite lower performance, showed stronger robustness. The stacking model, though theoretically advantageous, also exhibited overfitting tendencies, highlighting the importance of model selection based on both predictive strength and generalization capacity.

A critical trade-off between computational efficiency and predictive accuracy was also observed. While models such as KNN and SVM required minimal computational time 2.66 and 3.97, their limited predictive generalization undermines their practical utility. In contrast, Random Forest, with moderate training time, delivered superior accuracy and remains the most sensible solution.

This study establishes that combining unsupervised clustering with Random Forest-based supervised classification provides a scalable and reliable approach for HFU prediction. The developed workflow offers substantial improvements over manual interpretation, ensuring consistency across large datasets and facilitating broader application in reservoir characterization and field-scale development planning. Future research should extend this framework to diverse geological settings and incorporate real-time well log and remote sensing data to enhance predictive performance further.

References

Abbaszadeh, M., Fujii, H., & Fujimoto, F. (1996). Permeability prediction by hydraulic flow units: Theory and applications. *SPE Formation Evaluation, 11*(4), 263–271. https://doi.org/10.2118/30158-PA

Abnavi, A. D., Karimian Torghabeh, A., & Qajar, J. (2021). Hydraulic flow units and ANFIS methods to predict permeability in heterogeneous carbonate reservoir: Middle East gas reservoir. *Arabian Journal of Geosciences, 14*, Article 754. https://doi.org/10.1007/s12517-021-07084-5

Adjei, S., Takyi-Amponsem, D. Y., Quaye, J. A., Sokama-Neuyam, Y. A., Oleka, J. O., Asigbi, G. K., & Duodu, T. K. (2025). Automated hydraulic flow unit determination using FZI-Z-score probability, K-means

clustering with autoencoders, and machine learning classification: A case study of the Subei Basin, China. *Journal of Applied Geology*. https://doi.org/10.1016/j.jappgeo.2025.105778

Aguilera, R. (2010, June 27–30). Flow units: From conventional to tight gas to shale gas reservoirs. In *SPE Trinidad and Tobago Energy Resources Conference, Port of Spain, Trinidad*. https://doi.org/10.2118/132845-MS

Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics, 9*(8), 1295. https://doi.org/10.3390/electronics9081295

Al-Mudhafar, W. J. (2016, May 5–6). *Incorporation of bootstrapping and cross-validation for efficient multivariate facies and petrophysical modeling*. In **SPE Low Perm Symposium** (pp.15). Society of Petroleum Engineers. https://doi.org/10.2118/180277-MS

Amaefule, J. O., Altunbay, M., Tiab, D., Kersey, D. G., & Keelan, D. K. (1993, October 3–6). Enhanced reservoir description: Using core and log data to identify hydraulic (flow) units and predict permeability in uncored intervals/wells. In *Proceedings of the SPE Annual Technical Conference and Exhibition* (SPE Paper 26436-MS). https://doi.org/10.2118/26436-MS

Astsauri, T., Habiburrahman, M., Ibrahim, A. F., & Wang, Y. (2024). Utilizing machine learning for flow zone indicators prediction and hydraulic flow unit classification. *Scientific Reports, 14*(1), 11442. https://doi.org/10.1038/s41598-024-54893-1

Babadagli, T., & Al-Salmi, S. (2002, October). Improvement of permeability prediction for carbonate reservoirs using well log data. In *SPE Asia Pacific Oil and Gas Conference and Exhibition*, Melbourne, Australia (Paper SPE-77889). https://doi.org/10.2118/77889-MS

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms* (Advanced Applications in Pattern Recognition). Springer. https://doi.org/10.1007/978-1-4757-0450-1

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Carman, P. C. (1937). *Fluid flow through granular beds*. *Transactions of the Institution of Chemical Engineers, 15*, 150–166.

Chen, S., & Zhang, D. (2004). Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 34*(4), 1907–1916. https://doi.org/10.1109/TSMCB.2004.831819

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)* (pp. 785–794). https://doi.org/10.1145/2939672.2939785

Charkhi, A. H., Javaherian, A., & Nabi-Bidhendi, M. (2019). An adaptive neuro-fuzzy inference system for prediction of hydraulic flow units in uncored wells: A carbonate reservoir. *Exploration Geophysics, 50*(2). https://doi.org/10.1071/EG18045

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. https://doi.org/10.1007/BF00994018

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological), 39*(1), 1–38. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x

Doré, A. G., Lundin, E. R., Jensen, L. N., Birkeland, Ø., Eliassen, P. E., & Fichler, C. (1999). Principal tectonic events in the evolution of the northwest European Atlantic margin. *Petroleum Geoscience, 5*(4), 303–324. https://doi.org/10.1144/petgeo.5.4.303

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Glennie, K. W. (Ed.). (1998). *Petroleum geology of the North Sea: Basic concepts and recent advances* (4th ed.). Blackwell Science. https://doi.org/10.1002/9781444313413

Hearn, C. L., Ebanks, W. J., Jr., Tye, R. S., & Ranganathan, V. (1984). Geological factors influencing reservoir performance of the Hartzog Draw Field, Wyoming. *Journal of Petroleum Technology, 36*(8), 1335–1344. https://doi.org/10.2118/12016-PA

Hoff, P. D. (2009). *A first course in Bayesian statistical methods*. Springer. https://doi.org/10.1007/978-0-387-92407-6

Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., & Khraisat, A. (2024). *Enhancing K-nearest neighbor algorithm: A comprehensive review and performance analysis of modifications*. *Journal of Big Data*, *11*, Article 113. https://doi.org/10.1186/s40537-024-00973-y

Hussain, W., Ali, M., Kasala, E. E., Ali, S., Al-Khulaidi, G., Sadiq, I., Nyakilla, E. E., Hussain, S., & Abelly, E. N. (2025). Machine learning-driven classification of hydraulic flow units for enhanced reservoir characterization. *Physics of Fluids, 37*(3), 037194. https://doi.org/10.1063/5.0259689

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 31*(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). *Data clustering: A review*. ACM Computing Surveys, 31(3), 264–323. https://doi.org/10.1145/331499.331504

Jaiswal, J., & Samikannu, R. (2017, February). Application of random forest algorithm on feature subset selection and classification and regression. In *Proceedings of the 2017 World Congress on Computing and Communication Technologies (WCCCT)*. https://doi.org/10.1109/WCCCT.2016.25

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis*. Wiley. https://doi.org/10.1002/9780470316801

Kozeny, J. (1927). Ueber kapillare Leitung des Wassers im Boden. *Sitzungsberichte der Akademie der Wissenschaften in Wien, 136*, 271.

Krinidis, S., & Chatzis, V. (2010). A robust fuzzy local information C-means clustering algorithm. *IEEE Transactions on Image Processing, 19*(5), 1328–1337. https://doi.org/10.1109/TIP.2009.2037575

Lake, L. W., Johns, R. T., Rossen, W. R., & Pope, G. A. (Eds.). (2006). *Petroleum engineering handbook*. Society of Petroleum Engineers. https://doi.org/10.2118/10299-PEH

Man, H. Q., Hien, D. H., Thong, K. D., Dung, B. V., Hoa, N. M., Hoa, T. K., Kieu, N. V., & Ngoc, P. Q. (2021). Hydraulic flow unit classification and prediction using machine learning techniques: A case study from the Nam Con Son Basin, offshore Vietnam. *Energies, 14*(22), 7714. https://doi.org/10.3390/en14227714

McLachlan, G. J., & Peel, D. (2000). *Finite mixture models*. Wiley. https://doi.org/10.1002/0471721182

Mohaghegh, S., Balan, B., & Ameri, S. (1997). Permeability determination from well log data. *SPE Formation Evaluation, 12*(3), 170–174. https://doi.org/10.2118/30978-PA

Mohammadinia, M., Khazaei, M., Nikoo, M. R., & Ghahri, M. R. (2023). Machine learning-based identification of hydraulic flow units and permeability prediction in carbonate reservoirs. *Journal of Petroleum Science and Engineering, 229*, 111714. https://doi.org/10.1016/j.petrol.2023.111714

Nayak, N. P., et al. (2021). Application of hydraulic flow unit for prediction of flow zone in carbonate reservoir. *Arabian Journal of Geosciences, 14*(8), 1–8. https://doi.org/10.1007/s12517-021-07052-z

Norwegian Petroleum Directorate. (2023). *Resource report*. https://www.npd.no/en/facts/publications/reports/resource-report/

Park, H., & Jun, C. (2009). A simple and fast algorithm for K-medoids clustering. *Expert Systems with Applications, 36*(2), 3336–3341. https://doi.org/10.1016/j.eswa.2008.01.039

Raschka, S., & Mirjalili, V. (2019). *Python machine learning* (3rd ed.). Packt Publishing. https://doi.org/10.1007/978-1-78995-829-4

Rebelo, T. B., Batezelli, A., Mattos, N. H. S., & Leite, E. P. (2022). Flow units in complex carbonate reservoirs: A study case of the Brazilian pre-salt. *Marine and Petroleum Geology, 140*, 105639. https://doi.org/10.1016/j.marpetgeo.2022.105639

Reynolds, D. (2009). Gaussian mixture models. In S. Z. Li & A. Jain (Eds.), *Encyclopedia of biometrics* (pp. 659–663). Springer. https://doi.org/10.1007/978-0-387-73003-5_150

Rahimi, M., & Riahi, M. A. (2022). Reservoir facies classification based on random forest and geostatistics methods in an offshore oilfield. *Journal of Applied Geophysics, 201,* 104640. https://doi.org/10.1016/j.jappgeo.2022.104640

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. In F. R. Hampel, R. D. Martin, & S. R. S. Varadhan (Eds.), *Nonlinear estimation and classification* (pp. 149–171). Springer. https://doi.org/10.1007/978-1-4615-0357-0_9

Shenawi, S., Al-Mohammadi, H., & Faqehy, M. (2009, October 19–21). Development of generalized porosity-permeability transforms by hydraulic units for carbonate oil reservoirs in Saudi Arabia. In

SPE/EAGE Reservoir Characterization & Simulation Conference, Abu Dhabi, UAE. https://doi.org/10.2118/125380-MS

Siddique, A. B., Munshi, T. A., Rakin, N. I., Hashan, M., Chnapa, S. S., & Jahan, L. N. (2025). Application of supervised machine learning and unsupervised data compression models for pore pressure prediction employing drilling, petrophysical, and well log data. *Scientific Reports, 15*(1), 24706. https://doi.org/10.1038/s41598-025-89199-3

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

Soto Becerra, R., Arteaga, D., Martin Hidalgo, C., & Rodriguez, F. (2010, December). *Pore-type determination from core data using a new polar-transformation function from hydraulic flow units* (SPE Paper 136805-MS). In *Proceedings of the SPE Latin American and Caribbean Petroleum Engineering Conference* (pp. 1–6). Society of Petroleum Engineers. https://doi.org/10.2118/136805-MS

Svirsky, D., Ryazanov, A., Pankov, M., Corbett, P. W. M., & Posysoev, A. (2004, March 29–30). *Hydraulic flow units resolve reservoir description challenges in a Siberian oil field* (SPE Paper 87056-MS). In *Proceedings of the SPE Asia Pacific Conference on Integrated Modelling for Asset Management, Kuala Lumpur, Malaysia*. Society of Petroleum Engineers. https://doi.org/10.2118/87056-MS

Sun, Y., Wang, Y., & Zhang, X. (2025). A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *Journal of Machine Learning Research*, *26*(1), 1–45. https://doi.org/10.5555/jmlr.2025.26.1

Reynolds, D. A. (2008). *Gaussian mixture models*. In *Encyclopedia of Biometrics* (pp. 659–663). Springer. https://doi.org/10.1007/978-0-387-73003-5_196

Trung, D. N., Quang, M. H., Khac, H. T., Thien, H. P., Minh, H. C., & Viet, H. N. (2024). Applying machine learning for hydraulic flow unit classification and permeability prediction: Case study from carbonate reservoir in the southern part, Song Hong Basin. *Vietnam Journal of Marine Science and Technology, 24*(3), 219–234. https://doi.org/10.15625/1859-3097/18654

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer. https://doi.org/10.1007/978-1-4757-2440-0

Wu, G., Chang, E. Y., & Zhang, Z. (2005, December). An analysis of support vector machines for classification. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)* (pp. 647–650). https://doi.org/10.1109/ICDM.2005.96

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2018). Efficient kNN classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems, 29*(5), 1774–1785. https://doi.org/10.1109/TNNLS.2017.2673241