DOI 10.4430/bgo00522

Boosting NMR logging performance in oilfield applications: a robust ensemble machine learning framework

S.M. Ghiasi[1] and M. Abedi[2]

[1] *Institute of Geophysics, University of Tehran, Tehran, Iran*
[2] *School of Mining Engineering, College of Engineering, University of Tehran, Tehran, Iran*

(Received: 3 August 2025; accepted: 28 November 2025; published online: 13 February 2026)

ABSTRACT The borehole nuclear magnetic resonance technique is a technique widely used in exploration geophysics for subsurface structure imaging. It is primarily utilised in various projects including reservoir characterisation, hydrocarbon exploration, groundwater studies, and fracture characterisation. To interpret porosity, nuclear magnetic resonance (NMR) measurements are typically conducted alongside other logging methods such as caliper, resistivity, and gamma ray. However, due to cost and operational constraints, NMR data may not be acquired in all wells. Consequently, to predict this parameter in wells with missing data, the use of logs from other wells becomes necessary. Machine learning (ML) algorithms, increasingly prevalent, are well-suited for regression problems due to their capacity to model complex and latent relationships within data. Given the inherent difficulty in comprehending the intricate relationships within multi-dimensional spaces involving these measurement parameters, we implemented ML regression algorithms to map predictor parameters to response parameters. Thus, this study evaluates various ML regressors, including their ensemble learning counterparts, to compare their effectiveness in predicting NMR data, both individually and in combined configurations.

Key words: nuclear magnetic resonance (NMR), machine learning, ensemble learning, regression problem.

1. Introduction

Borehole nuclear magnetic resonance (BNMR), a well-known geophysical method, is an essential tool for subsurface fluid characterisation in both the oil and gas industry and groundwater investigations. Rooted in quantum mechanics, BNMR measures the magnetic response of hydrogen nuclei in pore fluids, enabling the direct, non-destructive quantification of porosity, permeability, fluid types, and pore structure. Unlike traditional logging methods such as resistivity, acoustic, and radioactive logging, which rely on indirect inferences and are often limited by lithological complexity, BNMR provides powerful and molecular-level insights into subsurface reservoirs (Beauce *et al.*, 1996; Coates *et al.*, 1999; Toumelin *et al.*, 2004; Müller-Petke *et al.*, 2011; Kirkland and Codd, 2018; Liao *et al.*, 2021; Luo *et al.*, 2022).

BNMR offers several advantages over conventional borehole logging. It is directly sensitive to fluid-filled porosity, unaffected by mineral matrix composition, and capable of estimating permeability and differentiating movable from bound water. While acoustic and density tools lose sensitivity in low-porosity formations and resistivity tools struggle with conductive minerals

© 2026 - The Author(s)

or complex fluids, BNMR maintains high reliability even in challenging reservoir environments (Semenov *et al*., 1988; Kenyon, 1997).

It is worth noting that nuclear magnetic resonance (NMR) logging is not performed in every well due to several factors, primarily related to cost, operational complexity, and reservoir conditions. NMR tools are expensive, both in terms of equipment and operational procedures, which makes them more suitable for complex or unconventional reservoirs where conventional logging methods may not provide sufficient data. Additionally, NMR logging requires significant time and specific operational protocols, which may not be feasible for all wells, especially in large-scale projects with tight time constraints. Furthermore, NMR's limited spatial resolution and sensitivity to specific fluid types or formation conditions (such as high resistivity, temperature, or pressure environments) may reduce its effectiveness in certain wells. In such cases, other methods, like resistivity or acoustic logging, along with predictive models based on available geological data, are used to estimate the properties of the reservoir.

Gamma ray (GR) logging, caliper logging, resistivity logging, and porosity log interpretation are essential geophysical measurements frequently used alongside NMR to enhance the understanding of subsurface formations. GR logging measures the natural radioactivity of the formation, by primarily identifying the presence of shale and distinguishing between different lithologies, which is performed by detecting variations in radioactive isotopes like potassium, thorium, and uranium. This method is crucial for characterising rock types and identifying boundaries between different strata. Caliper logging records borehole diameters and provides valuable information about wellbore stability and formation integrity. It also helps identify potential issues such as borehole enlargement or collapse, which can influence logging tool accuracy and well productivity. Resistivity logging, on the other hand, measures the electrical resistivity of the formation to estimate the reservoir's water saturation. This method distinguishes between water-, oil-, and gas-bearing zones and is often combined with the Archie equation to evaluate porosity and fluid distribution. Finally, porosity log interpretation provides direct or inferred measurements of the rock's void space, typically estimated from density or neutron logs, which help in understanding the porosity and fluid content of the formation. These logs are essential in estimating the amount of storage capacity in the reservoir and evaluating its potential for oil and gas production. When combined with NMR, these methods offer a comprehensive perspective of the formation properties, fluid content, and reservoir behaviour, supporting more accurate exploration and production decisions. Apparently, there may be a relationship between various measured parameters yet it is not straightforward and clear to find as a robust mathematical function. Accordingly, numerical or statistical solutions are needed to approach the correlation of measured parameters.

Predominantly, machine learning (ML) and deep learning (DL) algorithms have transformed regression analysis. They overcome many limitations posed by traditional statistical methods (among which linear regression) which often fail to capture nonlinear relationships, handle high-dimensional data, or model complex interactions.

In recent years, ML regression techniques have gained substantial traction in petrophysics, particularly for predicting BNMR parameters from conventional well logs. These data-driven approaches offer an effective means for modelling the complex, nonlinear relationships that exist between standard logging measurements (e.g. GR, resistivity, density, and neutron logs) and NMR-derived properties such as porosity, permeability, and T_2 relaxation distributions. Elsayed *et al.* (2022) highlighted the growing role of ML methods in NMR data interpretation, emphasising their potential to fill data gaps where direct measurements are unavailable. Rezaee (2022) applied a suite of ML algorithms to synthesise NMR outputs, including MLT_2, free fluid index, and bound volume irreducible, from conventional well logs in western Australia, noting that ensemble methods (e.g. AdaBoost) provided superior performance compared to single models. Similarly, Mustafa *et al.* (2023) demonstrated that artificial neural networks (ANNs), adaptive neuro-fuzzy inference systems, and functional networks could accurately predict NMR porosity in a Middle Eastern carbonate reservoir, achieving correlation coefficients exceeding 0.95. Tamoto (2023) expanded on this by applying supervised boosting algorithms such as CatBoost to estimate NMR porosity from auxiliary logs, achieving high predictive fidelity and robustness. In addition, Zhao *et al.* (2024) utilised ML regression to predict permeability in carbonate reservoirs using NMR log data, further validating the synergy between NMR measurements and ML frameworks for reservoir characterisation. Complementarily, Xu *et al.* (2022) discussed the broader advantages and challenges of ML in petrophysics, emphasising the importance of data quality, feature engineering, and hyperparameter tuning to ensure generalisation and interpretability. Collectively, these studies illustrate that ML-based regression is a mature and promising approach for NMR log prediction, enabling reliable estimation of petrophysical parameters in wells lacking direct NMR measurements, while offering improved accuracy and operational efficiency compared to traditional empirical methods. Decision trees (DTs) partition data into hierarchical subsets using feature thresholds, providing interpretability, and handling nonlinearities without prior assumptions (Breiman *et al*., 1984). Random forests (RFs), an ensemble of decorrelated DTs, enhance predictive accuracy and reduce overfitting via bootstrapping and feature randomisation (Breiman, 2001). Support vector regression (SVR) employs kernel functions to map data into high-dimensional spaces, effectively capturing complex patterns while maximising margin-based generalisation (Cortes and Vapnik, 1995). Gradient boosting machines iteratively optimise weak learners (e.g. DTs) by minimising residual errors, thus achieving state-of-the-art performance in many regression tasks (Friedman, 2001). Polynomial regression extends linear models by incorporating polynomial terms, though it risks overfitting with high degrees. Extremely randomised tree (extra tree) regression introduces additional randomness in DT splits, further reducing variance compared to RFs (Geurts *et al.*, 2006). Kernel ridge regression (KRR) combines L^2 regularisation with kernel tricks to balance bias-variance trade-offs in nonlinear settings (Saunders *et al*., 1998). Multi-layer perceptrons (MLPs), a foundational DL architecture, leverage multiple hidden layers to model intricate nonlinear mappings, excelling in large-scale regression problems (Schmidhuber, 2014). These ML/DL methods surpass traditional approaches by learning feature interactions, handling noisy or missing data, and efficiently scaling with data volume and complexity.

Ensemble learning, which combines predictions from multiple base models, has emerged as a critical strategy to enhance robustness and accuracy. Single models may suffer from high variance (e.g. DTs) or bias (e.g. linear regression), but ensembles mitigate these issues by aggregating diverse learner outputs, thereby, improving generalisation (Dietterich, 2000). Stacking integrates meta-learners to optimally combine base model predictions, often outperforming individual models through layered learning (Wolpert, 1992). Voting ensembles employ majority or averaging mechanisms, while weighted averaging assigns learned or heuristic weights to prioritise more accurate models, thus reducing error propagation. These techniques exploit the "wisdom of crowds" principle, where collective decisions outperform individual ones, particularly in heterogeneous or noisy datasets. By leveraging ensemble strategies, researchers can harness the complementary strengths of diverse algorithms, achieving superior predictive performance and stability in regression tasks compared to conventional single-model approaches.

In this study, we utilised multiple ML algorithms to solve our regression problem and compared them in terms of accuracy, robustness, and generalisation. Then, we applied ensemble learning to the trained models in order to investigate whether it enhances the accuracy and generalisation of the model. As the prediction procedure is commonly used in NMR well-logging explorations, it is crucial to identify and collect the best prediction algorithms and to utilise them in real-world situations. To this regard, Fig. 1 illustrates an overview of the workflow of this study.

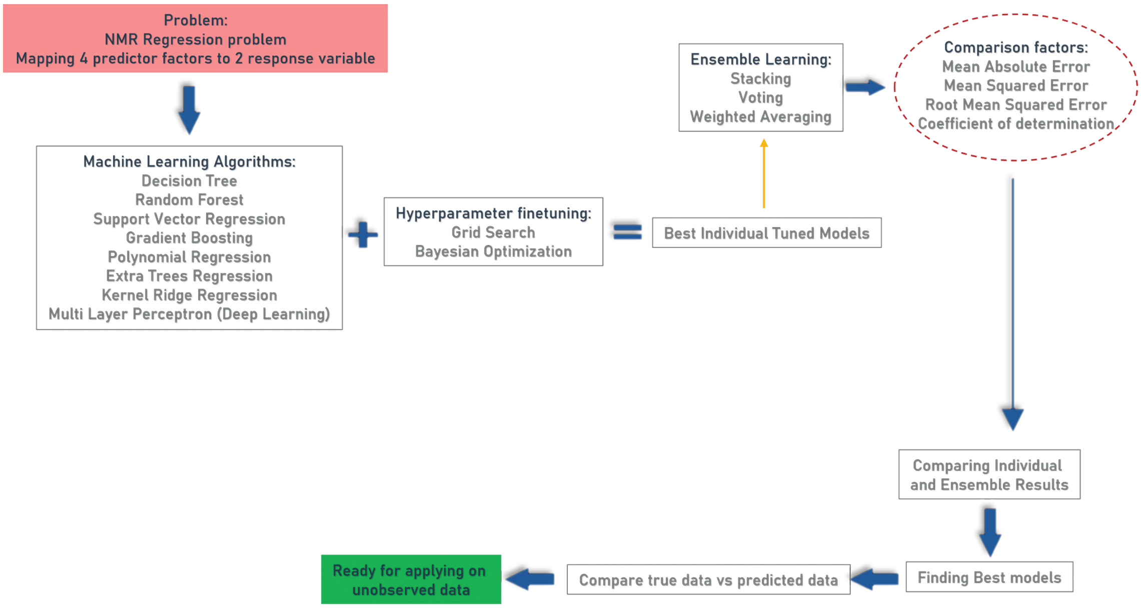
Problem:
NMR Regression problem
Mapping 4 predictor factors to 2 response variable
Machine Learning Algorithms:
Decision Tree
Random Forest
Support Vector Regression
Gradient Boosting
Polynomial Regression
Extra Trees Regression
Kernel Ridge Regression
Multi Layer Perceptron (Deep Learning)
Hyperparameter finetuning:
Grid Search
Bayesian Optimization
Best Individual Tuned Models
Ensemble Learning:
Stacking
Voting
Weighted Averaging
Comparison factors:
Mean Absolute Error
Mean Squared Error
Root Mean Squared Error
Coefficient of determination
Comparing Individual and Ensemble Results
Finding Best models
Compare true data vs predicted data
Ready for applying on unobserved data

Fig. 1 - Overview of this study workflow.

2. Methodology

2.1. Nuclear magnetic resonance

NMR techniques in geophysics primarily target the hydrogen nuclei in water molecules. These nuclei possess a property called spin, which gives rise to a magnetic moment. When placed in a static magnetic field (B_0), these magnetic moments precess at the Larmor frequency:

$$f_L = \frac{\omega_L}{2\pi} = \frac{\gamma|B_0|}{2\pi} \qquad (1)$$

where f_L is the Lamor frequency (Hz), i.e. the precession frequency of the hydrogen nuclei, $\gamma = 0.2675 \times 10^9\ S^{-1}\ T^{-1}$ is the gyromagnetic ratio for protons, ω_L is the angular Lamor frequency (rad/s), and B_0 is the static magnetic field strength (Levitt, 2008).

Instruments operate at varying field strengths: surface NMR uses the Earth's magnetic field (25–65 µT), while borehole and laboratory NMR use stronger fields, up to several Tesla resulting in higher Larmor frequencies (Hertrich, 2008).

The net magnetisation at equilibrium is described by Curie's Law:

$$M_0 = \frac{\eta\gamma^2 h^2}{4K_B T} B_0 \qquad (2)$$

where M_0 is the equilibrium magnetisation (A/m), representing the net alignment of proton spins in the applied field, η is the proton density, h is the reduced Planck constant (0.2675×10^{-34} J·s), K_B is the Boltzmann constant (1.381×10^{-23} J/K), and T is the absolute temperature (K). This shows that the magnetisation linearly scales with both the magnetic field strength and proton density, making NMR highly sensitive to water content (Keating and Knight, 2007).

To initiate NMR measurements, a radiofrequency (RF) pulse is applied at the Larmor frequency. After the RF pulse, spins relax to equilibrium, generating signals that decay over time. These decays are governed by the Bloch-Torrey equations:

$$E_{xy}(t) = E_0 \sum_i f_{2i} e^{\frac{-t}{T_{2i}}}, \qquad E_z(t) = E_0 \left(1 - \sum_i f_{1i} e^{\frac{-t}{T_{1i}}}\right) \tag{3}$$

where $E_{xy}(t)$ is the transverse signal decay (observed by NMR measurements), $E_z(t)$ is the longitudinal recovery signal, E_0 is the initial signal amplitude, f_{1i} and f_{2i} are fractional contributions of different relaxation components, T_{1i} is the spin-lattice (longitudinal) relaxation time (ms/s), representing energy dissipation to the surrounding, and T_{2i} is the spin-spin (transverse) relaxation time (ms/s), reflecting dephasing caused by molecular interactions. This multi-exponential decay reflects water molecules residing in different pore environments with varying relaxation times.

In porous media, the relaxation process is significantly influenced by surface relaxation, especially in the fast diffusion regime:

$$\xi = \frac{\rho_{1,2} a}{D} \ll 1 \tag{4}$$

where $\rho_{1,2}$ is the surface relaxivity, a is the mean diffusion length to a surface, and D is the self-diffusion coefficient of water (~2.3×10^{-9} m^2/s at 25 °C) (Brownstein and Tarr, 1979). In this regime, relaxation times can be linked to pore size and surface-to-volume ratios.

Diffusion length, which influences whether spins sample multiple pores during measurement, is estimated as:

$$l = \sqrt{6DT} \,. \tag{5}$$

This characteristic length is crucial for understanding spatial resolution and signal contributions in different pore systems.

Inverting NMR signals to retrieve relaxation-time distributions (T_1 or T_2) is an inherently ill-posed problem. Common inversion approaches include non-negative least squares (NNLS) (Menke, 2018), discrete exponential fitting with a limited number of terms, and stretched-exponential fitting $e^{-(\frac{t}{T})^{\beta}}$ to approximate continuous distributions.

In BNMR and laboratory NMR, the observed transverse relaxation time (T_2^*), obtained from free induction decay (FID), reflects multiple concurrent relaxation mechanisms:

$$\frac{1}{T_2^*} = \frac{1}{T_{2B}} + \frac{1}{T_{2S}} + \frac{1}{T_{2D}} + \frac{1}{T_{2IH}} \tag{6}$$

where T_{2B} denotes bulk-fluid relaxation (typically long and negligible in porous media), T_{2S} represents surface relaxation dominated by pore surface interactions, T_{2D} arises from diffusion in internal field gradients, and T_{2IH} accounts for external field inhomogeneities. To minimise inhomogeneity effects, spin-echo pulse sequences such as Carr-Purcell-Meiboom-Gill (Meiboom and Gill, 1958) are employed, yielding an effective transverse relaxation time (T_2) described by:

$$\frac{1}{T_2} = \frac{1}{T_{2B}} + \frac{1}{T_{2S}} + \frac{1}{T_{2D}}. \tag{7}$$

In porous media, T_{2B} is usually negligible, leaving surface and diffusion relaxation as the dominant contributors. Consequently, measured T_2 distributions provide reliable proxies for pore-size distribution, permeability, and other hydrologic properties (Grunewald and Knight, 2011).

NMR inversion is the computational process of transforming raw NMR relaxation signals (e.g. FID or echo trains) into distributions of relaxation times, primarily T_1 (spin-lattice relaxation time) and T_2 (spin-spin relaxation time). These distributions are critical for interpreting subsurface properties, as they reflect the interaction of fluid molecules with their pore-scale environments. The inversion is mathematically challenging due to the ill-posed nature of the problem as small measurement errors or noise can lead to large uncertainties in the derived distributions. To stabilise solutions, regularisation techniques (e.g. NNLS and Tikhonov regularisation) are applied, enforcing physical constraints like smoothness or non-negativity. Common inversion outputs include:

T_1/T_2 distribution. Histograms or continuous curves showing the relative abundance of fluid populations with distinct relaxation times. For example, short T_2 values (1–10 ms) often correspond to small pores (clay-bound water), while longer T_2 values (> 100 ms) indicate larger pores (free fluid);

Mean of log T_2 (MLT_2) The logarithmic average of T_2 times, which simplifies heterogeneous distributions into a single representative value. It correlates with the average pore size and is widely used in permeability models (e.g. $k \propto MLT_2^2$);

Standard deviation of T_2 (SDT_2). A measure of the spread in relaxation times, reflecting the heterogeneity of the pore sizes. A high standard deviation indicates a broad pore-size distribution (e.g. mixed clay and sand) while a low value suggests uniform pore geometry (e.g. well-sorted sand).

These outputs bridge NMR data to hydrologic and petrophysical properties. For instance, T_2 distributions are used to estimate water content, distinguish bound from mobile fluids, and predict permeability. MLT_2 provides a simplified metric for reservoir quality, while the standard deviation helps identify lithological complexity. In oilfield applications, T_1 is less commonly used than T_2 but can offer complementary insights into fluid viscosity or wettability.

2.2. Individual learning algorithms

2.2.1. Decision trees

A DT for regression is a supervised ML algorithm that predicts continuous values by learning simple decision rules inferred from data features. It recursively splits the dataset into smaller subsets based on conditions that minimise prediction error. At each node, the algorithm chooses a feature and threshold that minimises the mean-square error (*MSE*):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2 \tag{8}$$

where y_i is the true value, and $\hat{y}$ is the predicted mean value in that node.

The data is split to minimise the weighted *MSE* of child nodes:

$$Loss = \frac{n_L}{n} \cdot MSE_L + \frac{n_R}{n} \cdot MSE_R \tag{9}$$

where n_L and n_R are the sample sizes in the child nodes, ensuring larger subsets dominate *Loss*. Splitting continues until the stopping criteria (e.g. max depth or minimum samples) are met. Final predictions are the mean values of target variables in the leaf nodes. DTs are easy to interpret and they handle nonlinear relationships well (Breiman *et al*., 1984; Hastie *et al*., 2009).

2.2.2. Random forests

RF is an ensemble ML algorithm that builds multiple DTs and merges their outputs to improve prediction accuracy and control overfitting. For regression tasks, it predicts the average of outputs from all individual trees. Each tree is trained on a bootstrap sample of the data and, at each split, a random subset of features is considered, enhancing model robustness and reducing variance (Breiman, 2001).

Prediction $\hat{y}$ for a new input *x* in regression is given by:

$$\hat{y} = \frac{1}{T} \sum_{i=1}^{T} f_t(x) \tag{10}$$

where *T* is the number of trees, and $f_t(x)$ is the prediction from the *t*-th tree.

This method benefits from high accuracy, handles nonlinearity well, and is less sensitive to noise or overfitting compared to individual DTs (Hastie *et al*., 2009).

2.2.3. Support vector regression

SVR is a supervised ML algorithm derived from support vector machines, designed for regression tasks. SVR aims to find a function that deviates from the actual target values by no more than a specified margin (ε) while maintaining model simplicity. Instead of minimising the prediction error directly, SVR minimises the model complexity, by reducing the norm of the weight vector, and only penalises errors outside the ε-tube.

The optimisation problem is to minimise:

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \tag{11}$$

subject to:

$$y_i - (w \cdot x_i + b) \leq \varepsilon + \xi_i (w \cdot x_i + b) - y_i \leq \varepsilon + \xi_i^* \xi_i, \xi_i^* \geq 0.$$

Here, *C* controls the trade-off between model complexity and tolerance to deviations (Smola and Schölkopf, 2004). Kernel functions can be used for nonlinear regression.

2.2.4. Gradient boosting

Gradient boosting is a powerful ensemble ML algorithm used for regression and classification tasks. It builds models sequentially, where each new model aims to correct the residuals (errors) of the previous one. In regression, it minimises a loss function, often the *MSE*, using gradient descent.

At each iteration *m* the model adds a weak learner, $h_m(x)$, to current model $F_{m-1}(x)$ to improve predictions:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{12}$$

where γ_m is the step size, optimised to minimise the loss:

$$\gamma_m = arg\ arg\ min \sum_{i=1}^{n} L[y_i, F_{m-1}(x_i) + \gamma h_m(x_i)]\ . \tag{13}$$

The weak learner is trained on the negative gradient of the loss function (i.e. the residuals). This iterative process continues until a stopping criterion, such as a fixed number of iterations or minimal improvement, is met.

2.2.5. Polynomial regression

Polynomial regression is an extension of linear regression used to model nonlinear relationships between independent variable(s) and dependent variable. Unlike linear regression, which assumes a straight-line fit, polynomial regression fits a curve by introducing polynomial terms (e.g. x^2, x^3) into the model.

The general form of a polynomial regression model is:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \epsilon \tag{14}$$

where *y* is the predicted value, *x* is the input variable, β_i represents the model coefficients, *n* is the degree of the polynomial, and ε is the error term.

Model training involves minimising the sum of squared errors between predicted and actual values. Polynomial regression can capture complex trends but may overfit if the degree is too high.

2.2.6. Extra tree regression

Extra tree regression is an ensemble learning algorithm used for regression that builds multiple unpruned DTs from the original dataset. Unlike RF, this algorithm introduces greater randomness by randomly selecting cut-points for each feature and, then, choosing the best split among them. This reduces variance and often improves generalisation.

The model prediction is obtained by averaging the outputs of all trees:

$$\hat{y} = \frac{1}{T} \sum_{i=1}^{T} h_t(x) \tag{15}$$

where $\hat{y}$ is the predicted value, T is the number of trees, and $h_t(x)$ is the prediction from the t-th tree.

Each tree is trained on the full dataset (no bootstrapping) and, at each node, a subset of features is randomly selected. For each of these, a random split is chosen, and the one yielding the best performance (e.g. minimising the *MSE*) is used.

2.2.7. Kernel ridge regression

KRR is a ML algorithm that combines ridge regression with the kernel trick to handle nonlinear relationships in regression tasks. Like ridge regression, it includes an L^2 regularisation term to prevent overfitting. However, instead of working in the original input space, KRR maps the input data into a high-dimensional feature space using a kernel function, enabling it to model complex patterns. The prediction function in KRR is expressed as:

$$\hat{y} = K(K + \lambda I)^{-1} y \tag{16}$$

where K is the kernel matrix with $k_{ij} = k(x_i, x_j)$, λ is the regularisation parameter, I is the identity matrix, and y is the target vector. Common kernels include the radial basis function and polynomial kernels. KRR is particularly useful when the relationship between features and targets is nonlinear.

2.2.8. Multi-layer perceptrons

MLPs are a class of feedforward ANNs commonly used for regression tasks. An MLP consists of an input layer, one or more hidden layers with nonlinear activation functions, and an output layer. Each neuron in a layer is connected to every neuron in the next layer, allowing the model to learn complex, nonlinear relationships between input features and target values.

Output $\hat{y}$ of an MLP is computed as:

$$\hat{y} = f\left\{W^{(L)} \cdot f\left[W^{L-1} \cdot \ldots f\left(W^{(1)} x + b^{(1)}\right) \ldots + b^{(L-1)}\right]\right\} \tag{17}$$

where x is the input, $W^{(i)}$ and b^i are the weights and biases of layer i, and f is an activation function like the rectified linear unit or the hyperbolic tangent. For regression, the output layer typically uses a linear activation function. The model is trained by minimising the *MSE* loss using backpropagation and gradient descent (Goodfellow *et al.*, 2016).

2.3. Ensemble learning

Ensemble learning is a ML technique that combines multiple models (often called weak learners) that achieve better predictive performance than any single model alone. Common ensemble methods include bagging, boosting, and stacking. These techniques help reduce variance, bias, or improve predictions by leveraging the strengths of individual models (Zhou, 2025). For example, RF uses bagging with DTs, while AdaBoost uses boosting to focus on hard-to-predict instances (Dietterich, 2000).

2.3.1. Stacking

Stacking is an ensemble learning technique in ML that combines multiple base models (also known as level-0 models) to improve predictive performance. It involves training a meta-model (level-1 model) on the predictions made by the base models, which are used as input features. This method helps to reduce variance and bias, enhancing the model's overall accuracy.

In regression problems, stacking follows these steps:

1. base model training: multiple base learners (e.g. linear regression, DTs) are trained on the original dataset;
2. meta-model training: the predictions of the base models are used as input features to train a meta-model, which is often a simple linear regression.

The formula for the stacking model prediction can be written as:

$$\hat{y}_{stack} = \sum_{i=1}^{M} w_i \hat{y}_i \tag{18}$$

where $\hat{y}_i$ is the prediction of the i-th base model, w_i represents the weights learned by the meta-model, and M is the number of base models.

2.3.2. Voting

In a voting ensemble learning algorithm for regression, multiple models (usually base learners) are trained independently, and their predictions are aggregated to make a final prediction. In the case of regression, the most common method of aggregation is averaging. The final output is calculated as the weighted or unweighted average of the predictions from all the models in the ensemble.

Mathematically, for a set of base learners $f_1, f_2, ..., f_n$ the final prediction $\hat{y}_{ensemble}$ is given by:

$$\hat{y}_{ensemble} = \frac{1}{n} \sum_{i=1}^{n} f_i(x). \tag{19}$$

If weighted averaging is used, the formula becomes:

$$\hat{y}_{ensemble} = \sum_{i=1}^{n} w_i f_i(x) \tag{20}$$

where w_i are the weights assigned to each model based on its performance (e.g. accuracy or error rates). Voting ensemble methods help reduce overfitting and improve the robustness of the predictions.

2.3.3. Weighted averaging

Weighted averaging ensemble learning is an ensemble technique that combines the predictions of multiple regression models by assigning each model a weight based on its performance. The final prediction is calculated by taking a weighted average of the individual model predictions, where more accurate models receive higher weights.

Mathematically, prediction $y_{ensemble}$ is given by:

$$y_{ensemble} = \sum_{i=1}^{n} w_i \cdot y_i \tag{21}$$

where $y_{ensemble}$ is the final predicted value, n is the number of models, w_i is the weight assigned to model i, and y_i is the prediction of model i.

Weights are typically determined based on the model's performance (e.g. inverse of the error). The approach improves accuracy by leveraging diverse model strengths.

2.3.4. Comparative analysis of ensemble techniques

The three ensemble methods, i.e. stacking, voting, and weighted averaging, differ in their approaches for combining base models, their adaptability, and their computational complexity. Stacking distinguishes itself through its hierarchical learning framework, where a meta-model is trained to optimise the integration of base model predictions. This two-stage process allows stacking to capture complex, nonlinear relationships among models, making it particularly effective for heterogeneous datasets or tasks requiring high precision. However, this flexibility comes at the cost of increased computational overhead and the need for careful design to avoid overfitting the meta-model.

In contrast, voting ensembles employ a static aggregation rule, typically an unweighted or weighted average, without any secondary training phase. While this simplicity ensures computational efficiency and ease of implementation, voting lacks the adaptability to refine its combination strategy based on data patterns. Its strength lies in reducing variance and improving robustness through the "wisdom of the crowd" effect, but it may underperform when base models exhibit significant variability in accuracy or when their errors are correlated.

Weighted averaging strikes a balance between these extremes. By assigning predefined weights to base models (often proportional to their validation performance, it introduces a performance-aware aggregation mechanism without the complexity of meta-learning. This approach is advantageous when model accuracies are stable and well-understood, as it prioritises stronger predictors while maintaining computational simplicity. Nevertheless, like voting, weighted averaging cannot adapt dynamically to new data or model interactions, limiting its utility in scenarios where the optimal model combination shifts across different regions of the input space.

The choice among these methods ultimately depends on the problem constraints and priorities. Stacking is ideal for maximising predictive accuracy in complex, high-dimensional regression tasks, provided sufficient computational resources are available. Voting offers a lightweight solution for scenarios where interpretability and speed are critical, while weighted averaging provides middle ground, leveraging performance-based weighting without incurring the training burden of stacking. Practitioners must carefully weigh these trade-offs, considering factors such as dataset heterogeneity, computational budget, and the desired balance between adaptability and simplicity.

3. Results and discussion

This study applies the aforementioned ML algorithms to data from the Keathley Canyon region in the Gulf of Mexico. The goal is to learn the mapping from GR, caliper, and resistivity logs, and interpreted porosity data (input variables) to the SDT_2 and the MLT_2 (output variables). The dataset consists of 2,745 samples, divided into 80% for training and 20% for testing. A key challenge in the early stages of model development is tuning the hyperparameters of each algorithm to achieve optimal performance. To address this, we employed the grid search and Bayesian optimisation methods for hyperparameter tuning. After obtaining the best configurations, the regression models were trained and evaluated using performance metrics such as *MSE*, root of *MSE* (*RMSE*), mean absolute error (*MAE*), and the coefficient of determination (R^2). To assess model robustness, we also used cross validation (CV) scores and learning curve analysis. Furthermore, feature importance and feature permutation techniques were applied to evaluate the contribution of each input variable. Finally, after analysing the performance of the individual algorithms, we implemented ensemble learning methods, such as stacking, voting, and weighted averaging, to investigate their potential for improving prediction accuracy in the BNMR method.

3.1. Hyperparameter sensitivity

Hyperparameter sensitivity testing is crucial for evaluating how variations in hyperparameters affect the performance of ML models, especially in regression algorithms. The approach typically involves the alteration of hyperparameters to observe their impact on model accuracy, stability, and generalisation. This testing ensures that the model performs optimally under a range of settings and avoids overfitting or underfitting. It is particularly necessary in projects to prevent over-optimisation and ensure robustness across different datasets or conditions (Bergstra and Bengio, 2012).

Grid search is an exhaustive search method that tests all possible combinations of hyperparameter values within a predefined grid. It is simple but computationally expensive. The formula for grid search can be written as *Best parameters = arg arg* $L(f_\theta)$ where θ represents the hyperparameters, Θ is the hyperparameter space, and $L(f_\theta)$ is the loss function. Bayesian optimisation employs probabilistic models to identify optimal hyperparameters. It iteratively explores the parameter space, concentrating on regions that are more likely to enhance model performance. The formula involves maximising an acquisition function, α (θ), such as expected improvement which is θ_{next} = *arg arg* α (θ) where α (θ) reflects the expected improvement based on prior evaluations (Snoek *et al*., 2012).

Fig. 2 presents the learning curves of the individual ML algorithms, where the *x*-axis and *y*-axis represent the hyperparameter combinations and the corresponding R^2 scores, respectively. Investigating the hyperparameter sensitivity of the algorithms enables us to assess which models require more extensive tuning (i.e. beyond their default settings) to achieve optimal performance. Additionally, this analysis enables us to search for better hyperparameter combinations so as to develop more accurate and efficient predictor models.

As illustrated in Fig. 2a, the grid search method for the DT algorithm shows a relatively narrow range of R^2 score variation (exception made for a few combinations) which implies moderate sensitivity to hyperparameter tuning. This observation also holds for Bayesian optimisation; however, the grid search approach achieved higher R^2 values overall. Fig. 2b represents the RF model. As expected, the learning curves obtained from both grid search and Bayesian optimisation show moderate sensitivity to hyperparameter tuning. This sensitivity is slightly lower than that observed in the DT model, particularly for grid search. According to Fig. 2c, the SVR model demonstrates very high sensitivity to hyperparameter variation, as SVR hyperparameters directly influence model complexity and generalisation ability. As shown in Fig. 2d, the gradient boosting algorithm also demonstrates relatively high sensitivity, with Bayesian optimisation producing more stable R^2 scores, many of which exceeding 0.9. In contrast, Fig. 2e illustrates the low sensitivity of the polynomial regression model, where both grid search and Bayesian

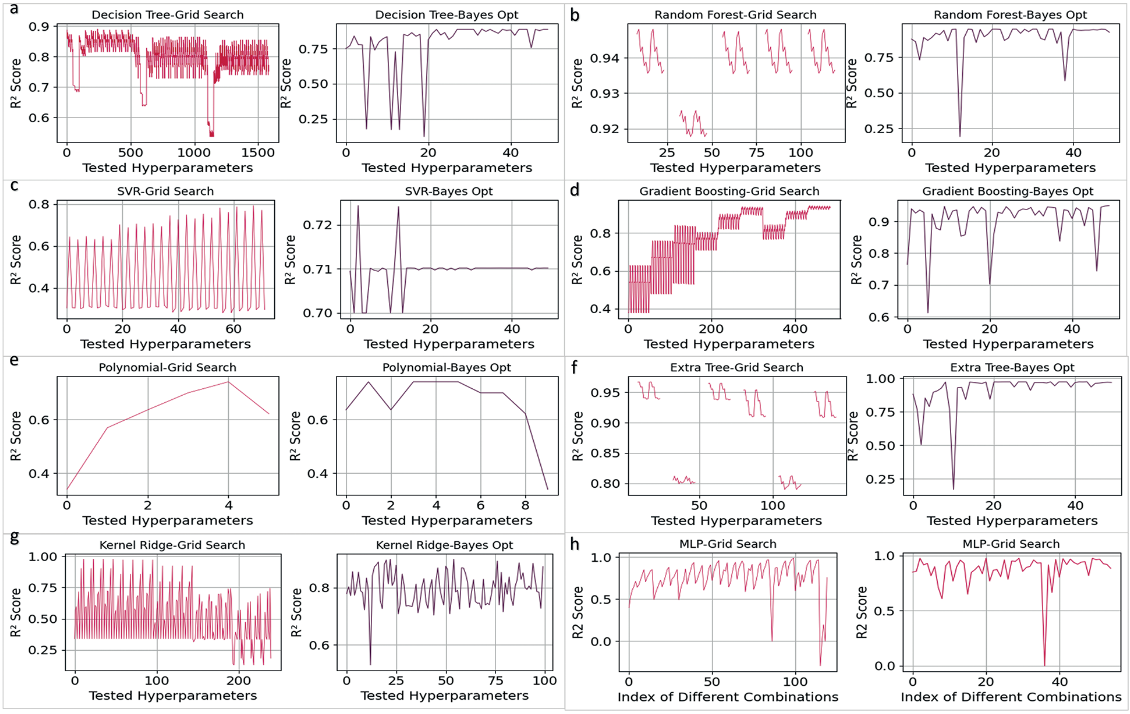

Fig. 2 - Hyperparameter tuning learning curve of employed ML algorithm for both grid search and Bayesian optimisation approaches: a) DT), b) RF, c) SVR, d) gradient boosting, e) polynomial regression, f) extra trees regression, g) KRR, and h) MLP.

optimisation yield similar R^2 ranges and overall behaviour. As expected, Fig. 2f, representing the extra tree regression model, shows a sensitivity pattern similar to that of the RF. The KRR model (Fig. 2g) indicates relatively high sensitivity to hyperparameter tuning in both grid search and Bayesian optimisation, though grid search achieved higher R^2 scores. Based on the wide range of R^2 score variation in the learning curve of the MLP model (especially with grid search), there is a clear indication of high sensitivity to hyperparameter tuning. Therefore, a careful specification of the number of hidden layers, neurons per layer, learning rate, and regularisation parameters was required to avoid underfitting or overfitting.

3.2. Model performance and accuracy

In ML regression, performance metrics help assess model accuracy. *MAE* measures the average magnitude of errors between predicted and actual values, offering a straightforward assessment of model accuracy (Hyndman and Koehler, 2006). *MSE* squares the errors, emphasising larger discrepancies, and is sensitive to outliers (James *et al.*, 2013). *RMSE* is the square root of *MSE*, providing error magnitude in the same units as the data, which is useful for interpretability (Chai and Draxler, 2014). R^2 evaluates how well the model explains the variance in the target variable, with a higher R^2 indicating a better fit (Theil, 1961). These metrics are essential for understanding the model's strengths and limitations, guiding model refinement and ensuring effective deployment in real-world applications.

Accordingly, we collected the most efficient and accurate hyperparameter configurations and trained each of the eight models using their best corresponding hyperparameters on 80% of the sample dataset. Afterwards, we applied the aforementioned parameters to evaluate the performance of the trained models on the test dataset. Five metrics, as presented in Fig. 3, and more specifically *MSE* (Fig. 3a), *RMSE* (Fig. 3b), *MAE* (Fig. 3c), R^2 (Fig. 3d), and accuracy (Fig. 3e), were used to compare the performance of the models. At a glance, it can be observed that the SVR and polynomial models exhibit poorer performance compared to the other models, indicated by relatively high values of *MSE* (greater than 0.2), *RMSE* (greater than 0.4), and *MAE* (greater than 0.3). Consequently, these models also show lower R^2 scores and reduced accuracy in mapping predicted test output values to the actual values. In contrast, the remaining trained models demonstrate strong performance in predicting the test dataset values. As shown, the *MSE* values for the DT, RF, extra trees, KRR, and MLP models are all below 0.05. Accordingly, their *RMSE* and *MAE* values are also within acceptable low ranges. The results show that we achieved accuracies exceeding 80% for the DT, extra trees, KRR, and MLP algorithms in solving the regression problem. It is worth noting that our hyperparameter tuning process aimed not

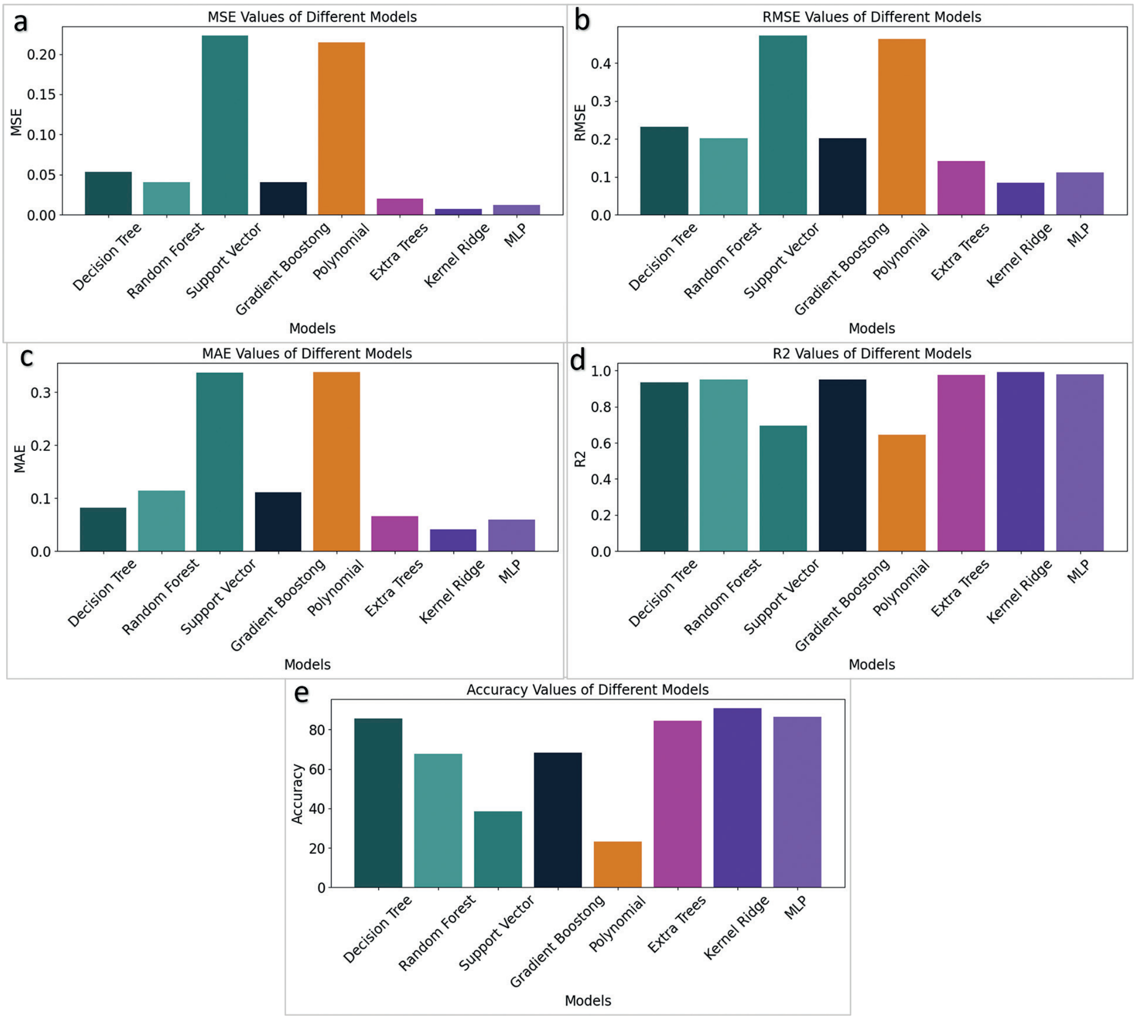

Fig. 3 - Performance evaluation of individual models using various metrics: a) *MSE*, b) *RMSE*, c) *MAE*, d) R^2, and e) accuracy.

only to achieve higher performance but also to prevent overfitting, thereby preserving the generalisation capability of the models.

Since all the models are trained on the same dataset, the R^2 score is a suitable metric to evaluate which models better fit the data (though not necessarily indicating a good model in absolute terms). In addition, we calculated the baseline performance of our target data using two approaches: first, the R^2 score of a baseline model that predicts the mean of the target values and, second, the R^2 score using a median predictor, which is more robust to outliers. The results indicate R^2 scores greater than 0.7 for almost all ML techniques. These results show that the R^2 scores of our individual models represent a significantly higher performance compared to simply predicting the mean or median of the target data. Furthermore, the standard deviation of the target data (training dataset) is 0.96 for MLT_2 and 0.94 for SDT_2. Considering the *RMSE* values of our models, which are substantially lower than the standard deviation of the target variables, it can be concluded that our models capture meaningful patterns in the data, even the weaker models, such as the SVR and the polynomial regressor models.

3.3. Robustness and stability

Robustness and stability testing are essential in evaluating the generalisability and reliability of ML regression models. CV, particularly k-fold CV, divides a dataset into several subsets to iteratively train and validate the model. This approach helps evaluate model performance across different data splits and reduces the risk of overfitting (Hastie *et al.*, 2009). It provides a more accurate estimate of model performance compared to a single train-test split. Learning curves, which plot model performance (e.g. R^2) against the number of training samples, are used to determine underfitting or overfitting and evaluate whether additional data could improve model performance (Géron, 2019). A stable learning curve suggests model robustness to data variability. These approaches are essential in real-world projects to ensure model performance is not overly sensitive to noise or specific data subsets. Without them, models may yield misleading results when deployed, especially on unseen data. Thus, robustness and stability testing support model selection, hyperparameter tuning, and data sufficiency assessment, ensuring dependable deployment in practical applications.

Fig. 4a illustrates the mean values of the CV scores (R^2) for the different models. According to the results, all models exhibit high R^2 scores across the various CV folds, with the exception of the support vector and polynomial models, which fall below 80%. This indicates relatively robust performance across different splits of the training data. In addition, we evaluated the standard deviation of the R^2 scores for each model through the CV approach (Fig. 4b). As theoretically expected, the DT and MLP algorithms are more prone to overfitting due to their nature and the limited size of the training dataset. However, the standard deviation values for these models are relatively low and acceptable, more specifically, 0.03 or less for the DT and 0.04 for the MLP. The other algorithms demonstrate greater stability across varying subsets of the training data, which suggests a better generalisation capability. To investigate the robustness and stability of our models, we also analysed the learning curves in relation to the size of the training dataset. This analysis shows how the algorithm performance depends on the training data size. As shown in Fig. 5, nearly all models exhibit similar learning curve behaviour and values, converging towards the training data. The variation range in the learning curve R^2 scores for all models is approximately between 0.7 and 0.9, indicating that the models do not necessarily overfit the training data.

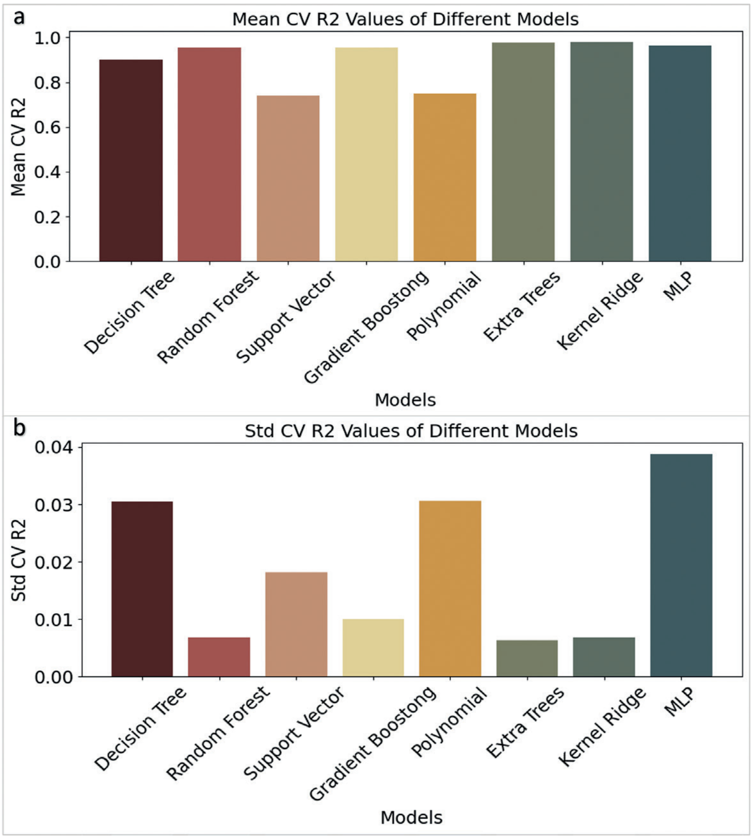

Fig. 4 - CV scores of models R^2 scores using two statistical approaches: a) mean of CV R^2 scores, and b) standard deviation of CV R^2 scores.

3.4. Feature importance and interpretability

Feature importance and permutation importance are key tools for interpreting ML regression models. Feature importance provides a measure of the contribution each input variable makes to the model's prediction. In tree-based models, it is often computed based on the reduction in error that each feature provides during training (Breiman 2001). However, these built-in methods can be biased toward features with more levels. Permutation importance, on the other hand, is model-agnostic and assesses the importance of each feature by measuring the increase in prediction error after randomly shuffling the feature's values (Fisher *et al.*, 2019). This disrupts the relationship between the feature and target variable, showing how much the model relies on such feature. The use of these approaches in ML projects helps in model interpretation, variable selection, and identification of spurious relationships. It enhances trust in the model, ensures better generalisability, and can lead to simplified models with comparable performance.

As shown in Fig. 6, nearly all the models recognise that interpreted porosity is the most important and decisive feature among the predictor parameters. This consistency suggests that porosity is likely a key factor in the underlying process that the regression problem is aiming to capture. In other words, regardless of the model architecture or the method used to estimate feature importance (whether based on coefficients, impurity reduction, or permutation importance), porosity consistently provides the most predictive power. The convergence on the same feature indicates that the predictive patterns in the data are stable. Such general agreement further enhances the overall reliability and trustworthiness of the model predictions.

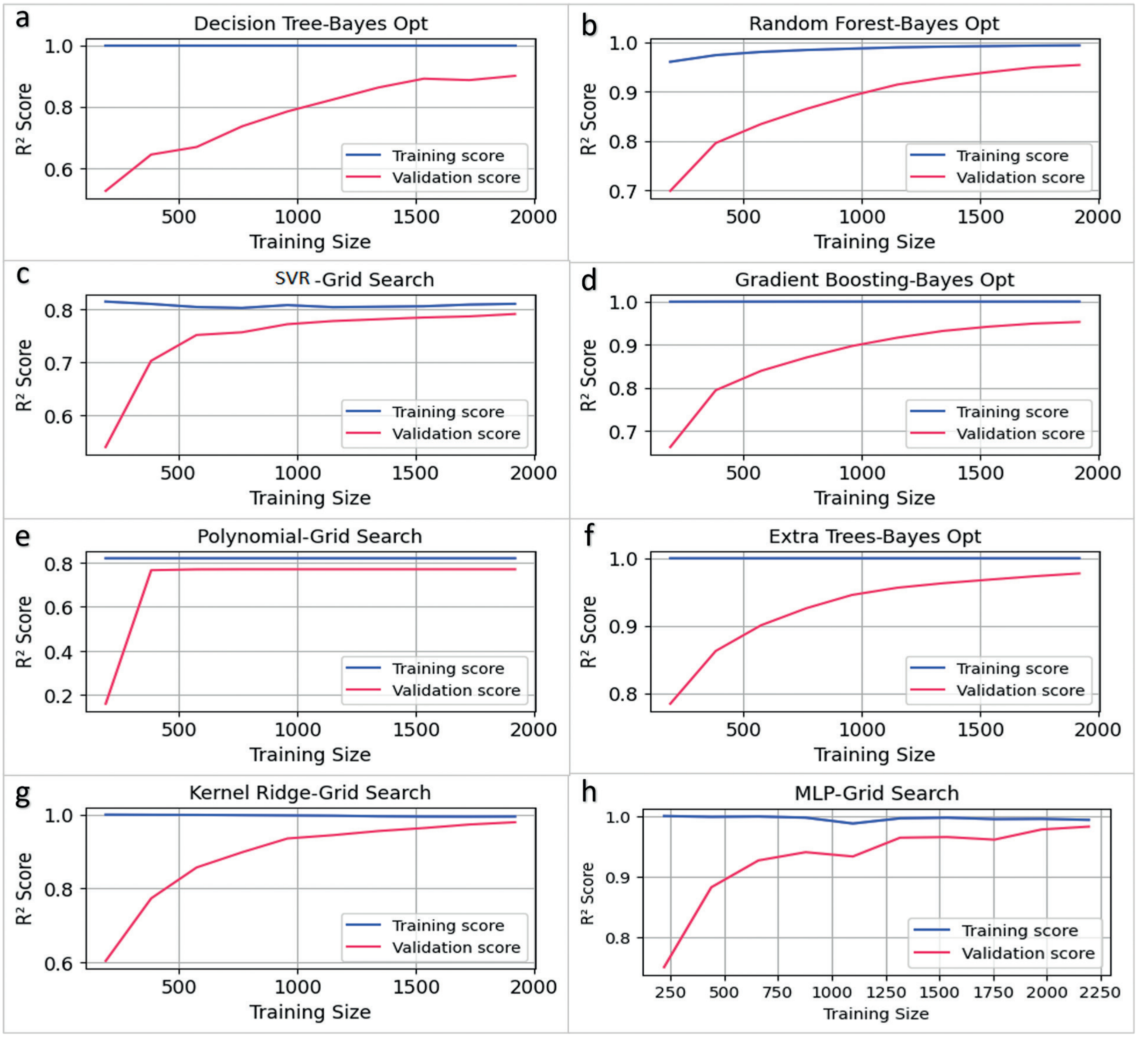

Fig. 5 - Learning curve of different models, R^2 score versus different training sizes for: a) DT, b) RF, c) SVR, d) gradient boosting, e) polynomial regression, f) extra trees regression, g) KRR, and h) MLP.

3.5. Ensemble performance

Ensemble learning is a ML paradigm where multiple models, often called base learners or weak learners, are combined to produce a stronger predictive model. The central idea is that while individual models may be prone to errors, combining them can lead to improved accuracy, robustness, and generalisation (Zhou, 2025). Stacking (stacked generalisation) involves training multiple base models and, then, using a meta-model to learn how to best combine their predictions. The meta-model is trained on the outputs (predictions) of the base models, with the aim of correcting their individual weaknesses. Voting is typically used in classification tasks, where each base model votes for a class, and the final prediction is made by majority (hard voting) or by averaging predicted probabilities (soft voting). It is simple yet effective when models are diverse. Weighted averaging is commonly used in regression, where predictions are combined using a weighted mean. Models with better performance get higher weights, emphasising more reliable predictors. Ensemble learning is crucial because it reduces variance, bias, and avoids overfitting,

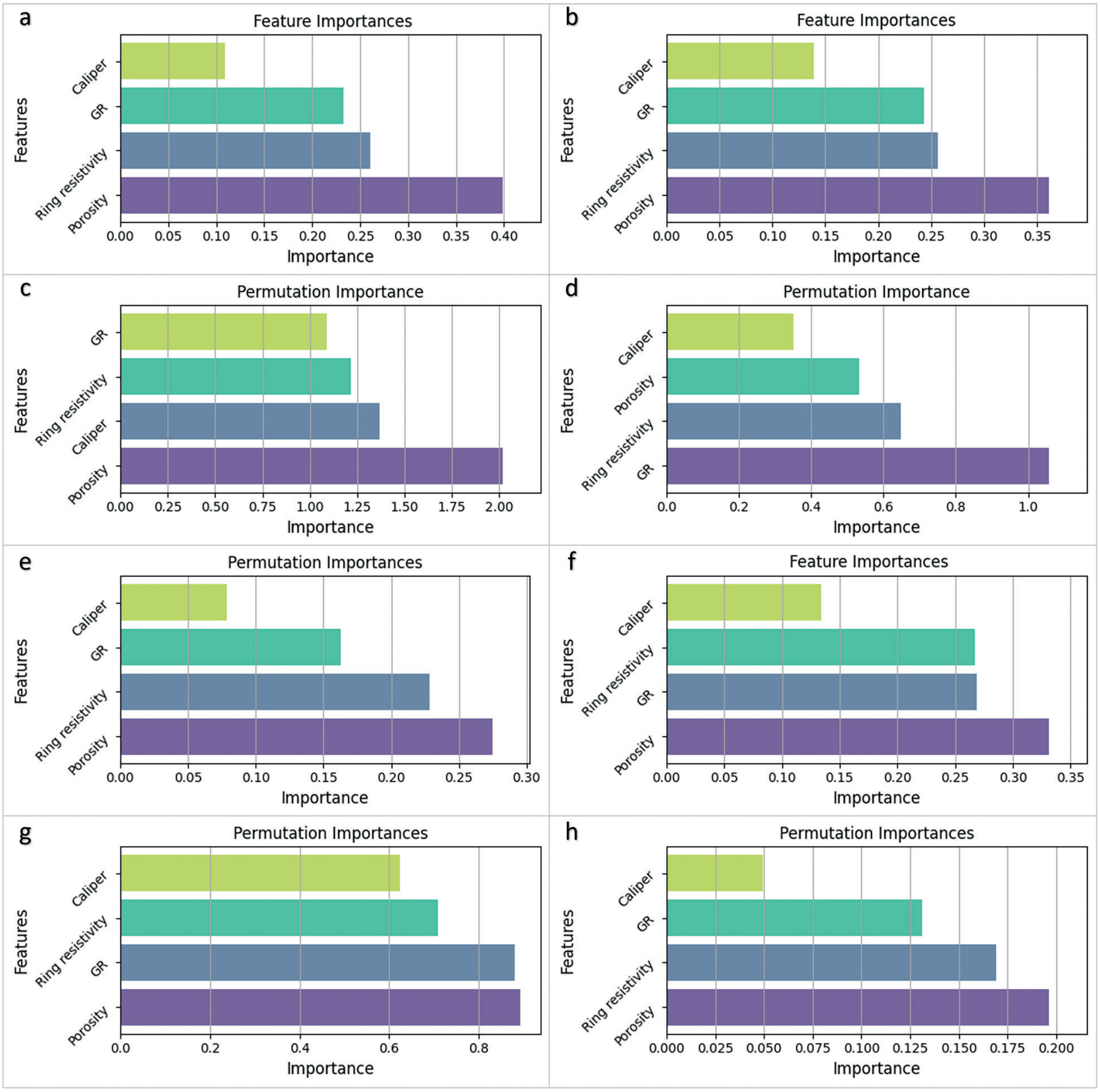

Fig. 6 - Feature and permutation importance of employed models: a) DT, b) RF, c) SVR, d) gradient boosting, e) polynomial regression, f) extra trees regression, g) KRR, and h) MLP.

often outperforming single models (Dietterich, 2000). It is especially effective when individual models are diverse and complementary.

Accordingly, we employed three ensemble methods, stacking, voting, and weighted averaging, to investigate whether combining our base models (individual ML models) could improve performance on our NMR data. To recall the workflow, we used grid search and Bayesian optimisation to achieve the best combination of hyperparameters for our models. We, then, trained each model individually using the fine-tuned hyperparameters. These trained models were subsequently used as base models in our ensemble learning methods. As shown in Fig. 7, we compared the ensemble models using the five metric parameters applied to the individual models. According to the results, the stacking ensemble method outperforms the

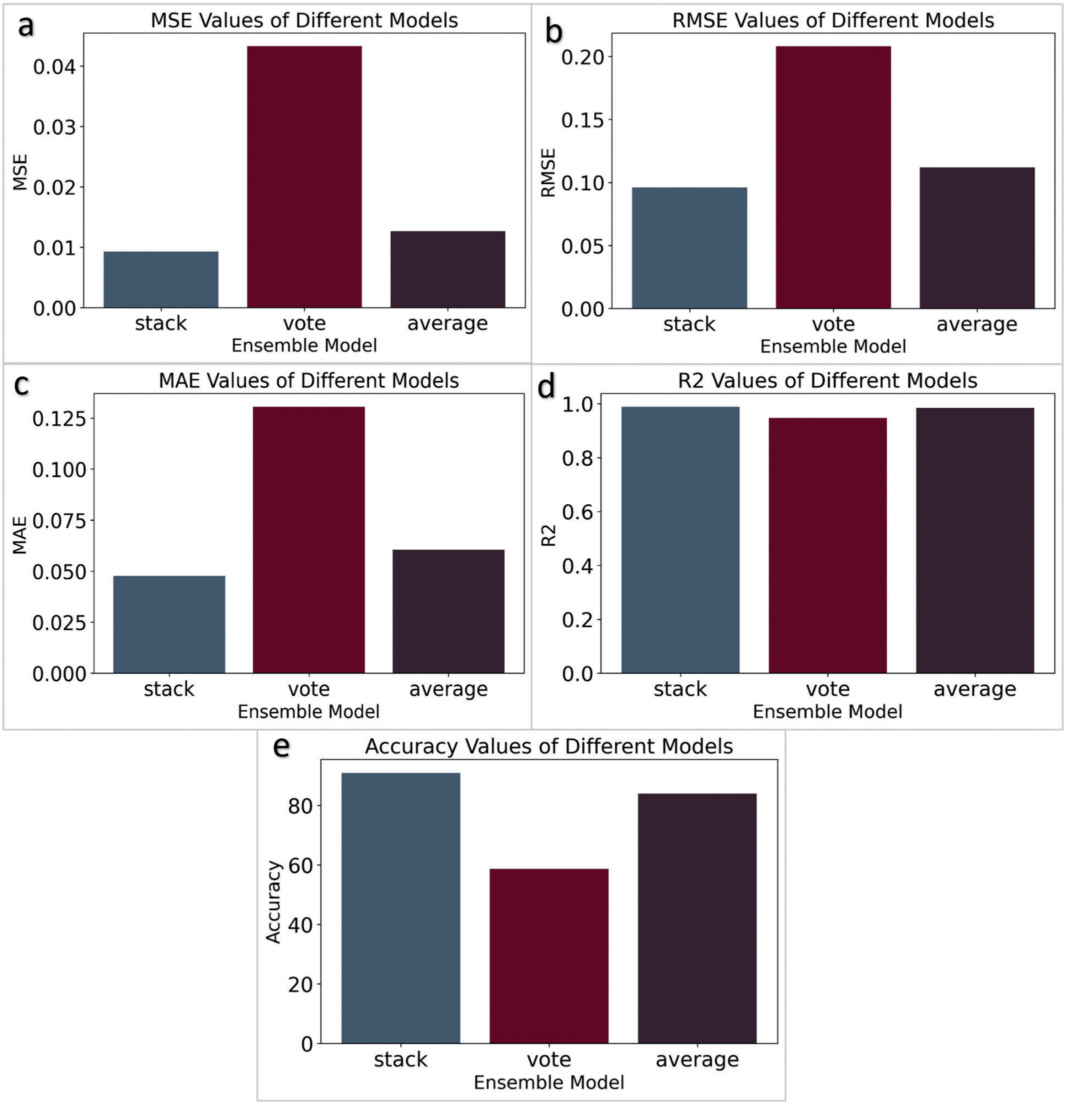

Fig. 7 - Performance evaluation of ensemble models using various metrics: a) *MSE*, b) *RMSE*, c) *MAE*, d) R^2, and e) accuracy.

others, demonstrating lower *MSE*, *RMSE*, and *MAE* values, as well as higher R^2 scores and accuracy (> 90%). The weighted averaging algorithm also performs similarly to stacking, while the voting method significantly underperforms the others, with its accuracy approaching only 60%. The inferior performance of the voting ensemble observed in this study can be explained by the correlated prediction errors among base models and the intrinsic limitations of the voting mechanism in regression tasks. Unlike stacking or weighted averaging, which adaptively combine models based on their relative performance or learned meta-relationships, the voting ensemble relies on a simple averaging scheme that assumes base learners contribute equally and independently. In practice, since all base models were trained on the same dataset and feature space, their residual errors were likely correlated, leading to redundancy rather than complementary learning. Consequently, the unweighted aggregation in the voting method was

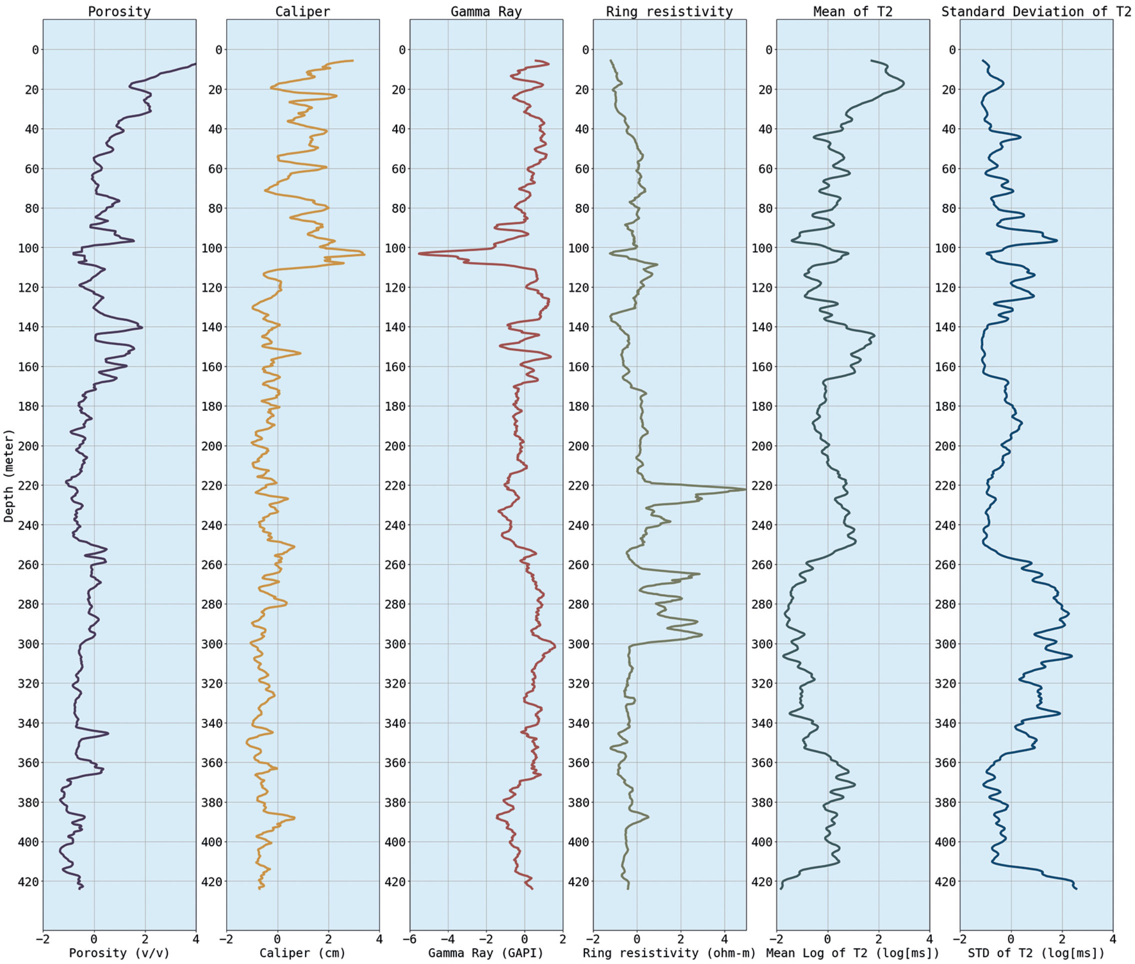

Fig. 8 - Sample dataset NMR logs. From left to right: interpreted porosity, caliper, GR, resistivity, *MLT*$_2$, and *SDT*$_2$.

unable to effectively mitigate bias or variance, resulting in lower predictive accuracy compared to the more flexible stacking and weighted averaging approaches.

3.6. Practical deployment considerations

The developed ML models can be integrated into existing well-logging workflows to predict NMR-derived parameters in wells lacking direct measurements. Using standard inputs (such as GR, caliper, resistivity, and interpreted porosity), the models provide rapid predictions with minimal operational overhead, supporting timely decision-making during drilling and reservoir evaluation. Ensemble methods such as stacking and weighted averaging, while more computationally intensive than single models, remain feasible for field deployment due to the low input dimensionality and small number of base learners. Offline hyperparameter tuning ensures that pre-trained models can be applied directly to incoming well-log data. Lightweight implementations or model compression can further optimise computational efficiency without sacrificing accuracy. This approach enables reliable NMR predictions, reduces reliance on costly measurements, and enhances operational decision-making in real-world exploration and production workflows.

4. Nuclear magnetic resonance log prediction

We trained eight individual ML algorithms to predict the mean and standard deviation of the NMR T_2 distributions by learning the hidden relationships between the four principal well-log parameters (caliper, GR, resistivity, and interpreted porosity). As demonstrated in Fig. 8, these input logs, which represent the core measurements acquired during the NMR project, were standardised and preprocessed prior to model training to ensure consistency and reliability. Standardisation was applied to minimise the effects of differing measurement units and data ranges, thereby enhancing model performance, particularly for algorithms sensitive to feature scaling. The raw logs were further smoothed using a moving-window approach (~3 m) to suppress high-frequency noise that cannot be effectively captured in regression analysis. In addition, missing or invalid entries (i.e. not a number, NaN) were removed to prevent bias in model fitting. Among the predictor variables, interpreted porosity emerged as the most influential parameter in establishing the mapping between input features and NMR targets.

In Fig. 9, all the individual and ensemble learning algorithms are compared based on five

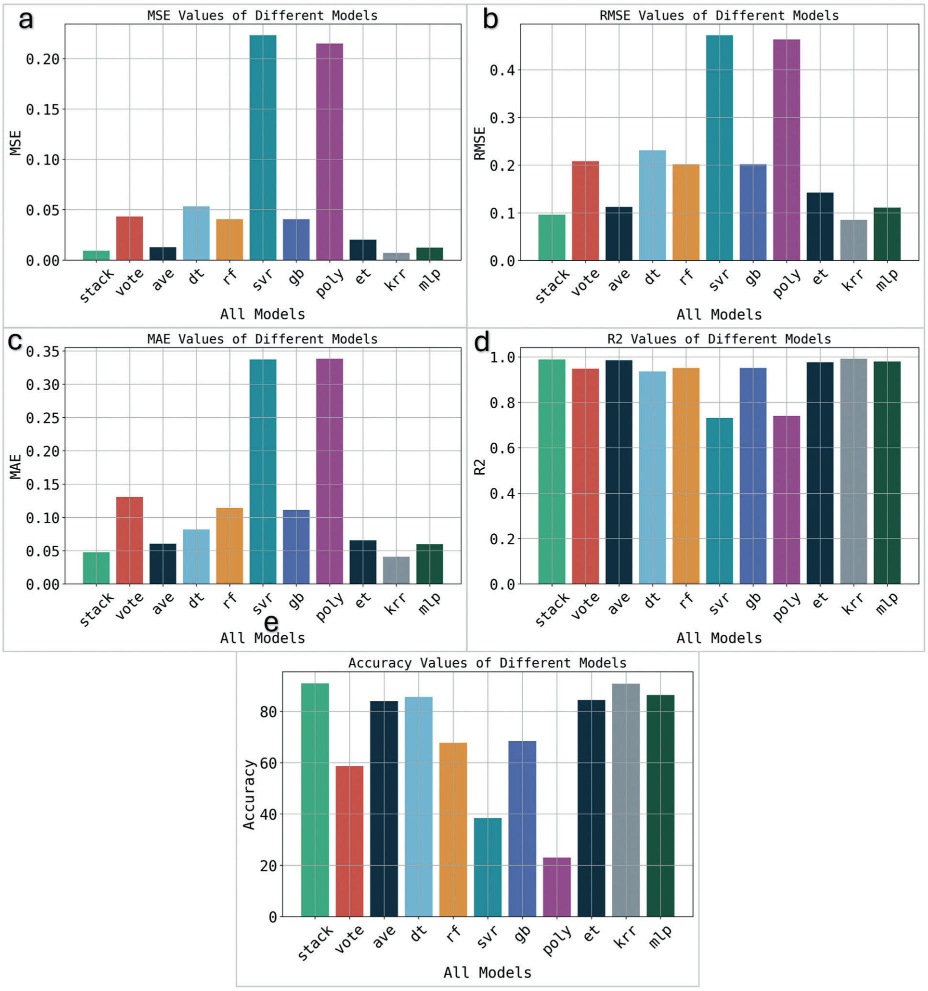

Fig. 9 - Performance evaluation of all employed models using various metrics: a) *MSE*, b) *RMSE*, c) *MAE*, d) R^2, and e) accuracy.

investigated metrics to assess their performance. As shown, with the exception of voting, the ensemble learning algorithms demonstrated robust and high predictive performance compared to the individual models. Among the base models, the KRR algorithm achieved the best performance and accuracy, while the polynomial regression algorithm exhibited the weakest performance. In the ensemble learning models, stacking achieved the highest precision and efficiency, while voting performed the worst. Overall, stacking slightly improved the performance of the learning procedure compared to the individual models, particularly in terms of reducing loss and increasing accuracy.

Figs. 10, 11, and 12 separately illustrate the true and predicted logs for all ML algorithms employed for the MLT_2 and the SDT_2. As shown, the models that demonstrated better performance and accuracy in the metric assessments are well-fitted to their true logs. Although overfitting is possible, the robustness and stability results show that our models perform consistently across different data folds. This is especially true for models typically prone to overfitting which still achieve low loss. It is important to emphasise that our goal was to determine whether, even with a relatively small training dataset and data that retain a reasonable level of noise, the regression models could achieve acceptable fitting, thus avoiding both underfitting and overfitting. In real-world scenarios, it is often preferable to predict NMR logs for missing wells using data from nearby wells, leveraging a maximised ML model performance rather than relying on models trained with data from other NMR projects. Therefore, it can be concluded that ensemble

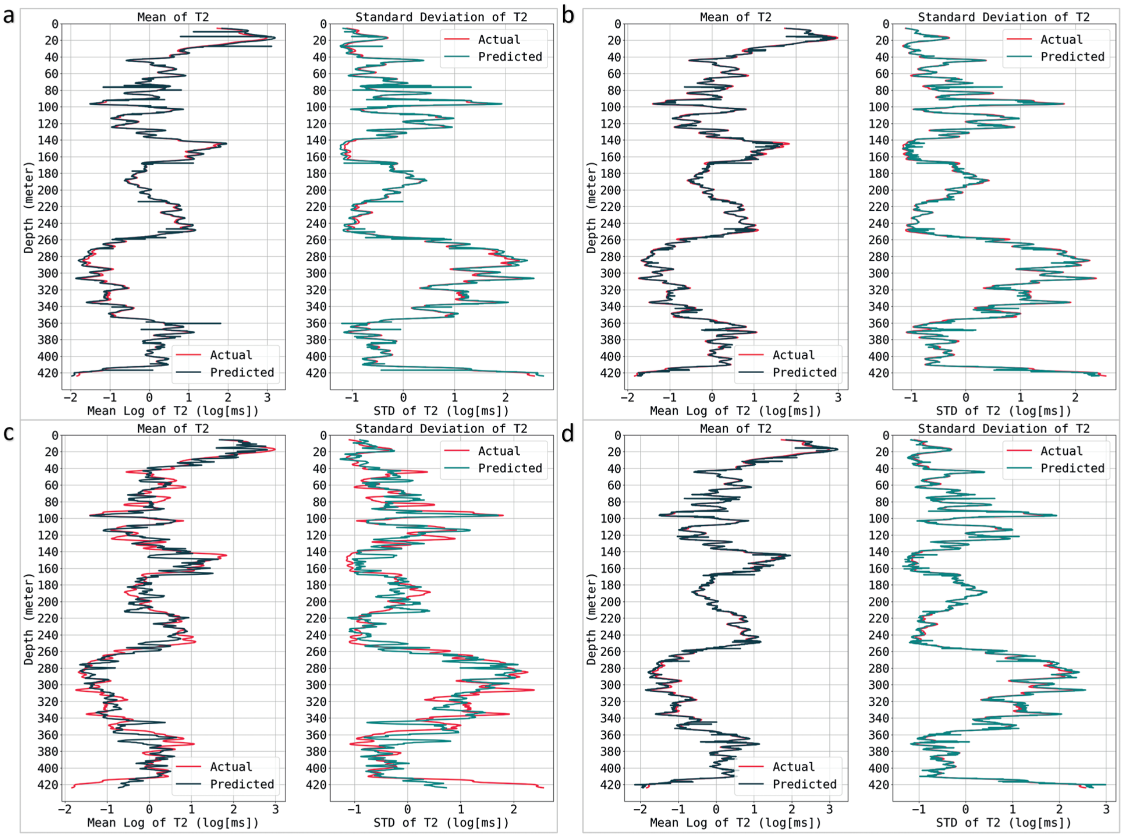

Fig. 10 - Comparative plots of true and predicted NMR log (MLT_2 and SDT_2, separately) for: a) DT, b) RF, c) SVR, and d) gradient boosting.

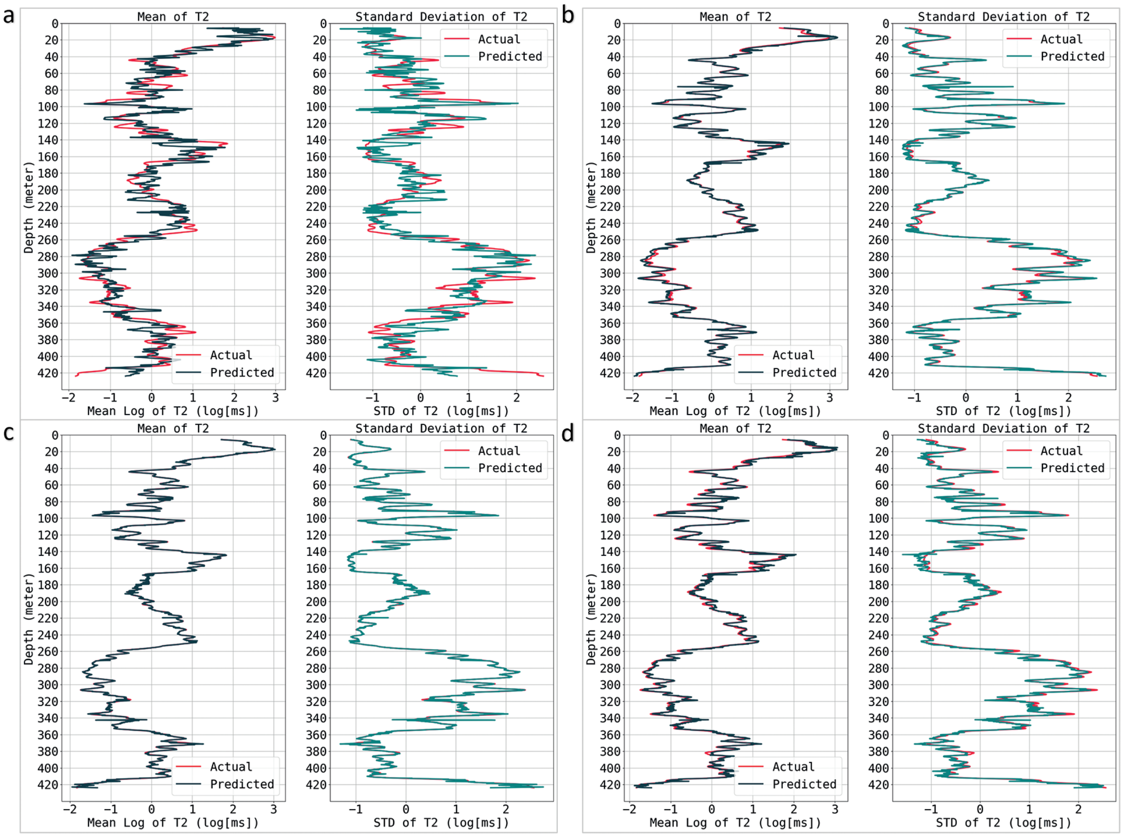

Fig. 11 - Comparative plots of true and predicted NMR log (*MLT_2* and *SDT_2*, separately) for: a) polynomial regression, b) extra trees regression, c) KRR, and d) MLP.

learning significantly helps improve training performance and accuracy while mitigating the problem of overfitting. As shown in Figs. 11c and 12a, KRR is the best-performing model among the individual learners, while stacking exhibits the highest performance among the ensemble learners. Although stacking improved overall performance, it showed lower overfitting compared to KRR, which is a crucial and promising outcome.

To determine the most appropriate model for predicting NMR logs in unmeasured boreholes, we evaluated the performance of all trained algorithms on unobserved data. The stacking ensemble method was ultimately selected for application in the Walker Ridge region of the Gulf of Mexico, owing to its superior accuracy, robustness, and generalisation across all evaluation metrics. Although KRR also yielded satisfactory results as an individual learner, the stacking approach demonstrated more stable and reliable predictions when extended to new wells. The predicted T_2 logs for the Walker Ridge dataset are shown in Fig. 13.

5. Conclusions

This study demonstrates the application of ML algorithms to predict NMR-derived parameters (i.e. the *SDT_2* and *MLT_2*) using well-log data (GR, caliper, resistivity, and interpreted porosity)

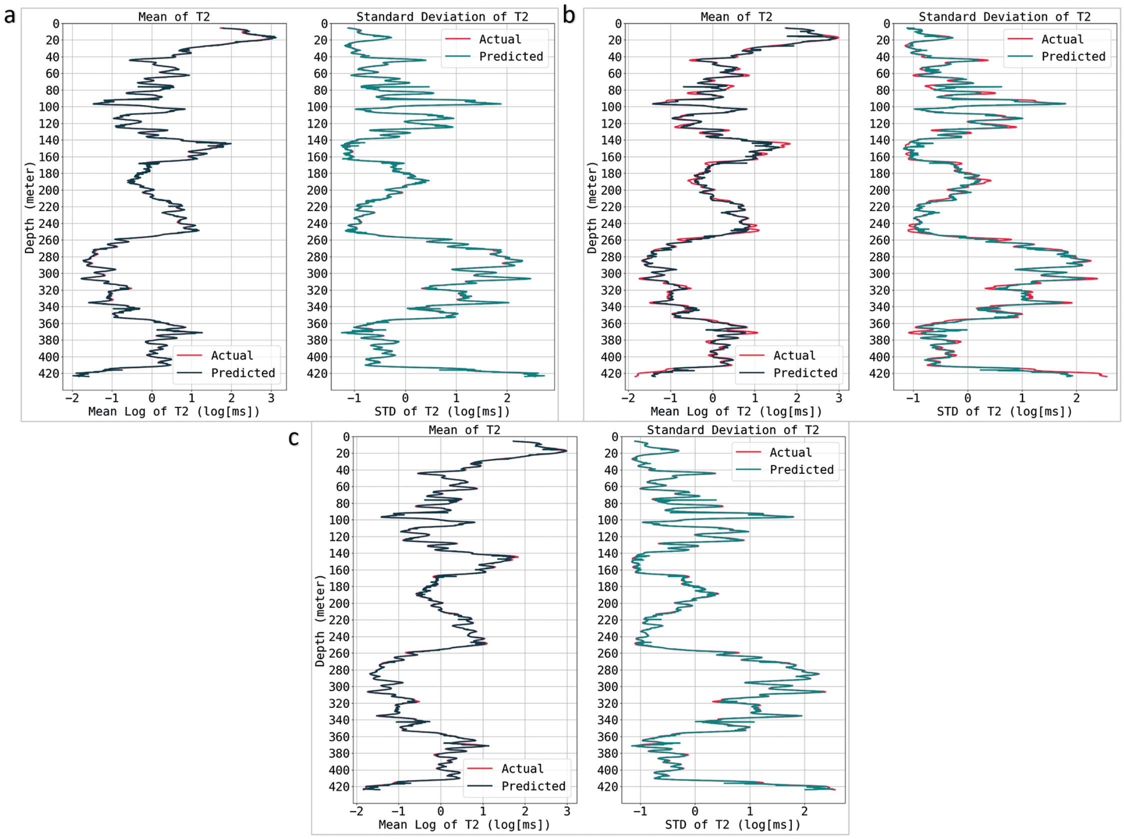

Fig. 12 - Comparative plots of true and predicted NMR log (MLT_2 and SDT_2, separately) for: a) stacking, b) voting, and c) averaging.

from the Keathley Canyon region in the Gulf of Mexico. The integration of hyperparameter tuning, model evaluation, and ensemble learning revealed critical insights into the performance, robustness, and interpretability of regression models for petrophysical applications.

Hyperparameter optimisation using grid search and Bayesian optimisation showed a pivotal effect in balancing model accuracy and generalisation. Algorithms such as gradient boosting, KRR, and MLP exhibited pronounced sensitivity to parameter configurations, necessitating meticulous tuning to avoid overfitting. Among individual models, DTs, RFs, extra trees, KRR, and MLP delivered superior predictive accuracy, achieving *MSE* values below 0.05 and R^2 scores exceeding 0.8. In contrast, SVR and polynomial regression underperformed due to their inherent instability with hyperparameter variations and limited adaptability to the dataset complexity.

The robustness of the models was rigorously validated through CV and learning curve analyses. Most algorithms demonstrated stable performance across diverse data splits, with R^2 scores consistently above 0.7. DTs and MLP, though slightly prone to overfitting, maintained acceptable stability (with standard deviation ≤ 0.04). Notably, the *RMSE* values for all models were substantially lower than the standard deviations of the target variables (SDT_2 and MLT_2), confirming their ability to capture meaningful petrophysical relationships despite noise in the dataset. Interpreted porosity emerged as the most influential predictor across all models, with feature importance and permutation analyses consistently highlighting its dominant role

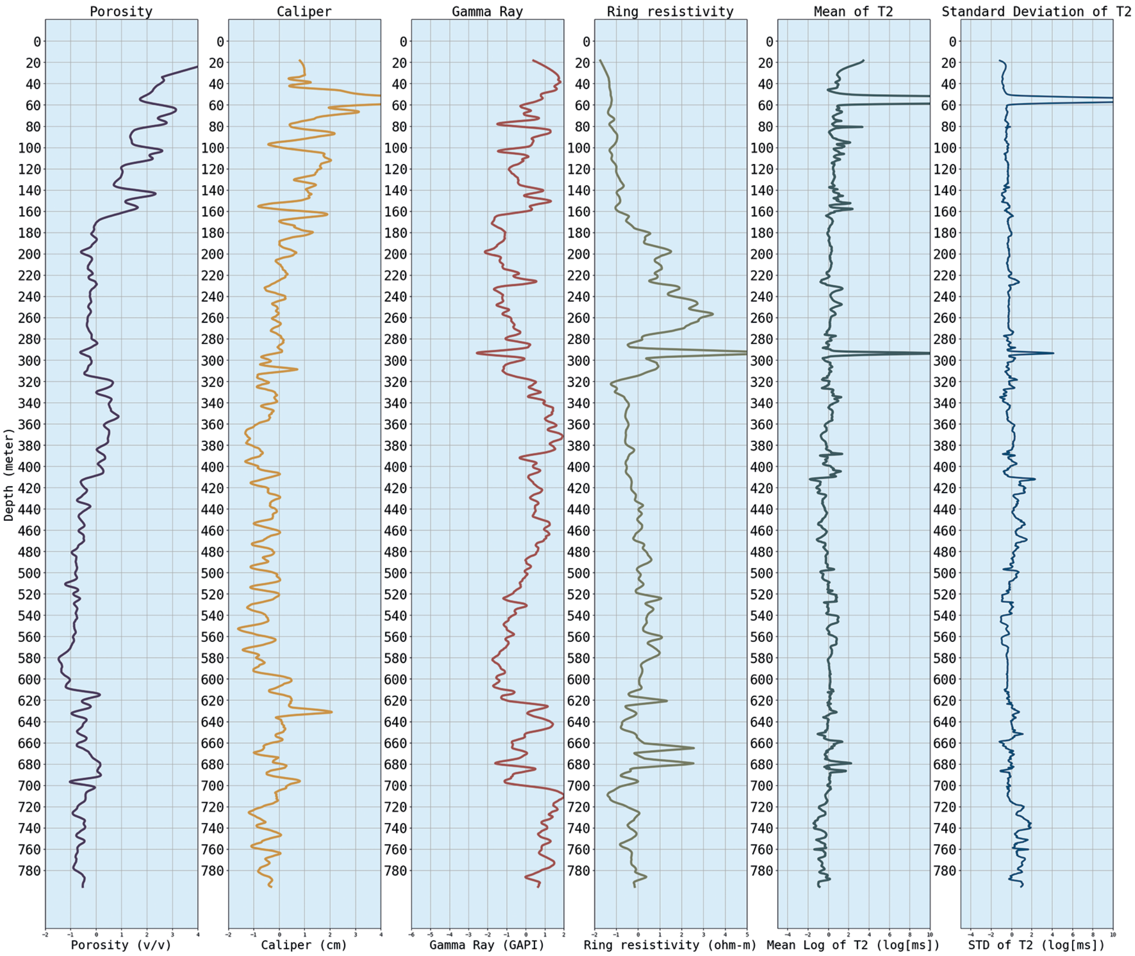

Fig. 13 - Walker Ridge dataset NMR logs. From left to right: interpreted porosity, caliper, GR, resistivity, predicted MLT_2, and predicted SDT_2.

in governing NMR log responses. This finding aligns with established petrophysical principles, reinforcing the reliability and interpretability of the model.

Ensemble learning methods further enhanced predictive performance, with stacking and weighted averaging outperforming individual models. Stacking achieved the highest accuracy (> 90%) by strategically combining base learners (e.g. KRR) through a meta-model, thereby mitigating overfitting and improving generalisation. Weighted averaging also performed robustly, while voting ensembles lagged significantly, emphasising the limitations of simplistic aggregation in regression tasks. The success of stacking highlights the value of leveraging complementary model strengths to refine predictions, particularly in geoscientific applications where data variability and noise are inherent challenges.

From a practical perspective, this work validates the feasibility of training reliable ML models on limited, region-specific datasets to predict NMR logs for wells lacking direct measurements. By prioritising local data over generalised models, the approach reduces overfitting risks and enhances operational relevance in hydrocarbon exploration. The study also highlights the value of interpretability tools, like feature importance analysis, to build trust with decision-makers and align with their expertise.

Although the developed models demonstrated strong predictive performance, their training was based on a relatively limited dataset comprising 2,745 samples from the Keathley Canyon region in the Gulf of Mexico. This geographic and geological homogeneity may restrict the generalisability of the results to other basins with different lithological, petrophysical, or depositional characteristics. To enhance the robustness and broader applicability of such models, future research should incorporate larger, multi-basin datasets that capture a wider range of geological variability. Moreover, employing transfer learning, where a model trained on one region is fine-tuned using data from another, could facilitate knowledge transfer across basins and reduce the need for extensive retraining. Complementary strategies such as cross-basin validation and domain adaptation should also be explored to systematically evaluate and improve model generalisation in diverse subsurface environments. Integrating advanced ensemble architectures, such as hybrid DL models, could further optimise accuracy. Additionally, embedding domain-specific constraints (e.g. petrophysical equations) into ML workflows may enhance interpretability and physical consistency. This research advances the integration of ML in petrophysics, offering a scalable, data-driven framework for NMR log prediction. The findings hold significant implications for reservoir characterisation, enabling efficient and reliable estimation of critical parameters in hydrocarbon exploration while balancing computational precision with ease of deployment.

Author contributions. S.M. Ghiasi was responsible for authoring the work, implementing the algorithm, and creating the visual representations. Meanwhile, M. Abedi contributed by supplying the data, proposing the algorithms, and refining the final manuscript.

Acknowledgments. The authors express their sincere gratitude to the School of Mining Engineering and the Institute of Geophysics at the University of Tehran for their invaluable support. Data are available upon reasonable request to the corresponding author via email address: maysamabedi@ut.ac.ir.

REFERENCES

Beauce A., Bernard J., Legchenko A. and Valla P.; 1996: *Une nouvelle méthode géophysique pour les études hydrogéologiques: l'application de la résonance magnétique nucléaire*. Hydrogéol. (Orléans), 1, 71-76.

Bergstra J. and Bengio Y.; 2012: *Random search for hyper-parameter optimization*. The J. Mach. Learn. Res., 13, 281-305.

Breiman L.; 2001: *Random forests*. Mach. Learn., 45, 5-32.

Breiman L., Friedman J.H., Olshen R.A. and Stone C.J.; 1984: *Classification and regression trees, 1st ed*. Chapman and Hall/CRC, 358 pp., doi: 10.1201/9781315139470.

Brownstein K.R. and Tarr C.E.; 1979: *Importance of classical diffusion in NMR studies of water in biological cells*. Phys. Rev. A, 19, 2446, doi: 10.1103/PhysRevA.19.2446.

Chai T. and Draxler R.R.; 2014: *Root mean square error (RMSE) or mean absolute error (MAE)*. Geosci. Model Dev. Discuss., 7, 1525-1534.

Coates G.R., Xiao L.Z. and Prammer M.G.; 1999: *NMR logging: principles and applications*. Halliburton Energy Services, Gulf Professional Publishing, Houston, TX, USA, 250 pp.

Cortes C. and Vapnik V.; 1995: *Support-vector networks*. Mach. Learn., 20, 273-297.

Dietterich T.G.; 2000: *Ensemble methods in machine learning*. In: Springer, Berlin-Heidelberg, Germany, Vol. 1857, pp. 1-15, Multiple Classifier Systems, Lecture Notes in Computer Science, doi: 10.1007/3-540-45014-9_1.

Elsayed M., Isah A., Hiba M., Hassan A., Al-Garadi K., Mahmoud M., El-Husseiny A. and Radwan A.E.; 2022: *A review on the applications of nuclear magnetic resonance (NMR) in the oil and gas industry: laboratory and field-scale measurements.* J. Pet. Explor. Prod. Technol., 12, 2747-2784.

Fisher A., Rudin C. and Dominici F.; 2019: *All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously*. J. Mach. Learn. Res., 20, 1-81.

Friedman J.H.; 2001: *Greedy function approximation: a gradient boosting machine*. *Ann. Stat.*, 29, 1189-1232.

Géron A.; 2019: *Hands-on machine learning with scikit-learn, keras, and tensorflow: concepts, tools, and techniques to build intelligent systems, 2nd ed.* O'Reilly Media Inc., Sebastopol, CA, USA, 493 pp.

Geurts P., Ernst D. and Wehenkel L.; 2006: *Extremely randomized trees*. Mach. Learn., 63, 3-42.

Goodfellow I., Bengio Y. and Courville A.; 2016: *Deep learning*. MIT Press, Cambridge, MA, USA, 792 pp.

Grunewald E. and Knight R.; 2011: *A laboratory study of NMR relaxation times in unconsolidated heterogeneous sediments*. Geophys., 76, G73-G83.

Hastie T., Tibshirani R. and Friedman J.; 2009: *The elements of statistical learning: data mining, inference, and prediction, 2nd ed.* Springer Series in Statistics, Berlin, Germany, 698 pp.

Hertrich M.; 2008: *Imaging techniques based on nuclear magnetic resonance*. Surv. Geophys., 29, 301-327.

Hyndman R.J. and Koehler A.B.; 2006: *Another look at measures of forecast accuracy*. Int. J. Forecast., 22, 679-688.

James G., Witten D., Hastie T. and Tibshirani R.; 2013: *An introduction to statistical learning, with application in R*. Springer Texts in Statistic, Springer, New York, NY, USA, 441 pp., doi: 10.1007/978-1-4614-7138-7.

Keating K. and Knight R.; 2007: *A laboratory study to determine the effect of iron oxides on proton NMR measurements*. Geophys., 72, E27-E32.

Kenyon W.E.; 1997: *Petrophysical principles of applications of NMR logging*. The log analyst, 38, 21-43.

Kirkland C.M. and Codd S.L.; 2018: *Low-field borehole NMR applications in the near-surface environment*. Vadose Zone J., 17, 1-11.

Levitt M.H.; 2008: *Spin dynamics: basics of nuclear magnetic resonance, 2nd ed.* John Wiley & Sons Ltd, Chichester, West Sussex, England, 752 pp.

Liao G., Luo S. and Xiao L.; 2021: *Borehole nuclear magnetic resonance study at the China University of petroleum*. J. Magn. Reson., 324, 106914, doi: 10.1016/j.jmr.2021.106914.

Luo S., Guo J. and Xiao L.; 2022: *Prospects of borehole NMR instruments and applications*. Magn. Reson. Lett., 2, 224-232.

Meiboom S. and Gill D.; 1958: *Modified spin-echo method for measuring nuclear relaxation times*. Rev. Sci. Instrum., 29, 688-691.

Menke W.; 2018: *Geophysical data analysis: discrete inverse theory, 4th ed*. Academic Press, Cambridge, MA, USA, 342 pp., doi: 10.1016/B978-0-12-813555-6.00001-0.

Müller-Petke M., Hiller T., Herrmann R. and Yaramanci U.; 2011: *Reliability and limitations of surface NMR assessed by comparison to borehole NMR*. Near Surf. Geophys., 9, 123-134.

Mustafa A., Tariq Z., Mahmoud M. and Abdulraheem A.; 2023: *Machine learning accelerated approach to infer nuclear magnetic resonance porosity for a Middle Eastern carbonate reservoir*. Sci. Rep., 13, 3956.

Rezaee R.; 2022: *Synthesizing nuclear magnetic resonance (NMR) outputs for clastic rocks using machine learning methods, examples from north west shelf and Perth basin, western Australia.* Energ., 15, 518, doi: 10.3390/en15020518.

Saunders C., Gammerman A. and Vovk V.; 1998: *Ridge regression learning algorithm in dual variables.* In: Shavlik J.W. (ed), Proc. 15th Int. Conf. Mach. Learn (ICML '98), Morgan Kaufmann Publ. Inc., San Francisco, CA, USA, pp. 515-521.

Schmidhuber J.; 2014: *Deep learning in neural networks*: an overview. Neural networks, 61, 85-117, doi: 10.1016/j.neunet.2014.09.003.

Semenov A.G., Burshtein A.I., Pusep A.Y. and Schirov M.D.; 1988: *A device for measurement of underground mineral parameters*. USSR patent: 1079063, (in Russian).

Smola A.J. and Schölkopf B.; 2004: *A tutorial on support vector regression*. Stat. Comput., 14, 199-222.

Snoek J., Larochelle H. and Adams R.P.; 2012: *Practical bayesian optimization of machine learning algorithms.* Adv. neural inf. process. syst., 25, 2960-2968.

Tamoto H.; 2023: *Prediction of nuclear magnetic resonance porosity well-logs using supervised machine learning models with auxiliary logs.* J. Pet. Sci. Eng., 224, 112395.

Theil H.; 1961: *Economic forecasts and policy.* North-Holland Publishing Company, Amsterdam, The Netherlands, 567 pp.

Toumelin E., Torres-Verdin C., Sun B. and Dunn K.J.; 2004: *A numerical assessment of modern borehole NMR interpretation techniques.* In: Proc. SPE Annual Technical Conference and Exhibition, Houston, TX, USA, SPE-90539-MS, doi: 10.2118/90539-MS.

Wolpert D.H.; 1992: *Stacked generalization.* Neural networks, 5, 241-259, doi: 10.1016/S0893-6080(05)80023-1.

Xu C., Fu L., Lin T., Li W. and Ma S.; 2022: *Machine learning in petrophysics: advantages and limitations.* Artif. Intell. Geosci., 3, 157-161, doi: 10.1016/j.aiig.2022.11.004.

Zhao J., Wang Q., Rong W., Zeng J., Ren Y. and Chen H.; 2024: *Permeability prediction of carbonate reservoirs based on nuclear magnetic resonance (NMR) logging and machine learning.* Energies, 17, 1458.

Zhou Z.H.; 2025: *Ensemble methods: foundations and algorithms*. Taylor & Francis Ltd, Chapman & Hall/CRC Machine Learning & Pattern Recognition, Milton Park, UK, 348 pp., doi: 10.1201/b12207.

Corresponding author: Maysam Abedi
School of Mining Engineering, University of Tehran
Amirabad, Tehran 98, Iran
Phone: +98 216 1114563; e-mail: maysamabedi@ut.ac.ir