Petroleum Science 22 (2025) 296–323

Contents lists available at ScienceDirect

Petroleum Science

journal homepage: www.keaipublishing.com/en/journals/petroleum-science

Original Paper

Carbon dioxide storage and cumulative oil production predictions in unconventional reservoirs applying optimized machine-learning models

Shadfar Davoodi [a, **], Hung Vo Thanh [b, c, *], David A. Wood [d], Mohammad Mehrad [a], Sergey V. Muravyov [e], Valeriy S. Rukavishnikov [a]

[a] School of Earth Sciences & Engineering, Tomsk Polytechnic University, Lenin Avenue, Tomsk, Russia
[b] Laboratory for Computational Mechanics, Institute for Computational Science and Artificial Intelligence, Van Lang University, Ho Chi Minh City, Viet Nam
[c] Applied Science Research Center, Applied Science Private University, Amman, Jordan
[d] DWA Energy Limited, Lincoln, UK
[e] Division for Automation & Robotics, Tomsk Polytechnic University, Lenin Avenue, Tomsk, Russia

ARTICLE INFO

Article history:
Received 7 April 2024
Received in revised form
9 September 2024
Accepted 20 September 2024
Available online 21 September 2024

Edited by Yan-Hua Sun

Keywords:
Hybrid machine learning
Least-squares support vector machine
Grey wolf optimization
Feature selection
Carbon dioxide storage
Enhanced oil recovery

ABSTRACT

To achieve carbon dioxide (CO_2) storage through enhanced oil recovery, accurate forecasting of CO_2 subsurface storage and cumulative oil production is essential. This study develops hybrid predictive models for the determination of CO_2 storage mass and cumulative oil production in unconventional reservoirs. It does so with two multi-layer perceptron neural networks (MLPNN) and a least-squares support vector machine (LSSVM), hybridized with grey wolf optimization (GWO) and/or particle swarm optimization (PSO). Large, simulated datasets were divided into training (70%) and testing (30%) groups, with normalization applied to both groups. Mahalanobis distance identifies/eliminates outliers in the training subset only. A non-dominated sorting genetic algorithm (NSGA-II) combined with LSSVM selected seven influential features from the nine available input parameters: reservoir depth, porosity, permeability, thickness, bottom-hole pressure, area, CO_2 injection rate, residual oil saturation to gas flooding, and residual oil saturation to water flooding. Predictive models were developed and tested, with performance evaluated with an overfitting index (OFI), scoring analysis, and partial dependence plots (PDP), during training and independent testing to enhance model focus and effectiveness. The LSSVM-GWO model generated the lowest root mean square error (RMSE) values (0.4052 MMT for CO_2 storage and 9.7392 MMbbl for cumulative oil production) in the training group. That trained model also exhibited excellent generalization and minimal overfitting when applied to the testing group (RMSE of 0.6224 MMT for CO_2 storage and 12.5143 MMbbl for cumulative oil production). PDP analysis revealed that the input features “area” and “porosity” had the most influence on the LSSVM-GWO model's prediction performance. This paper presents a new hybrid modeling approach that achieves accurate forecasting of CO_2 subsurface storage and cumulative oil production. It also establishes a new standard for such forecasting, which can lead to the development of more effective and sustainable solutions for oil recovery.
© 2024 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

The current state of the global energy sector is at a pivotal point, characterized by the urgent requirement to balance the escalating demand for energy resources with the imperative for environmental sustainability (Lin and Tan, 2021). Within this particular setting, the utilization of carbon capture and storage for enhanced oil recovery (CCS-EOR) arises as a multifaceted strategy that serves

* Corresponding author. Laboratory for Computational Mechanics, Institute for Computational Science and Artificial Intelligence, Van Lang University, Ho Chi Minh City, Viet Nam.
** Corresponding author. School of Earth Sciences & Engineering, Tomsk Polytechnic University, Lenin Avenue, Tomsk, Russia.
E-mail addresses: davoodis@hw.tpu.ru (S. Davoodi), hung.vothanh@vlu.edu.vn (H.V. Thanh), dw@dwasolutions.com (D.A. Wood).

https://doi.org/10.1016/j.petsci.2024.09.015
1995-8226/© 2024 The Authors. Publishing services by Elsevier B.V. on behalf of KeAi Communications Co. Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Nomenclature

γ	Regularization parameter
$\phi(\cdot)$	Feature map
σ	Radial basis function width
a_i	Lagrangian factor
a20-index	An index that provides insights into the concentration of data points around the $Y = X$ line
ARE	Average relative error
ARDExp	Automatic relevance determination (ARD) exponential kernel function
ARDRQ	ARD rational quadratic kernel function
BHP	Bottom hole pressure
CCS	Carbon capture and storage
CO_2	Carbon dioxide
C	Regularization parameter
CMG-GEM	Computer modeling group's greenhouse emissions model greenhouse emissions model software
Err	Prediction error
EOR	Enhanced oil recovery
Exp	Exponential kernel function
GPR	Gaussian process regression
GLSAU	Goldsmith-Landreth San Andres Unit
GWO	Grey wolf optimization
InjRate	Injection rate
K_r	Relative permeability
K_{rg}	Gas relative permeability
K_{rog}	Oil–gas relative permeability
K_{row}	Water–oil relative permeability
K_{rw}	Water relative permeability
KKT	Karush-Kuhn-Tucker
Lin	Linear kernel function
LHS	Latin hypercube sampling
LSSVM	Least square support vector machine
MD	Mahalanobis distance
ML	Machine learning
MLP	Multi-layer perceptron
MLPNN	Multi-layer perceptron neural network
MMbbl	One million barrels
MMT	One million metric tonnes
NSGA	Non-dominated sorting genetic algorithm
OFI	Over-fitting index
Pb	The best personal position
PDP	Partial dependent plot
Perm	Permeability
PI	Performance index
PI_{ts}	PI for test subset
PI_{tr}	PI for train subset
Poly	Polynomial kernel function
Por	Porosity
PSO	Particle swarm optimization
R	Correlation coefficient
R^2	Coefficient of determination
RE	Relative error
RBF	Radial basis function
RQ	Rational quadratic kernel function
RMSE	Root mean square error
$RMSE_{ts}$	RMSE for test subset
$RMSE_{tr}$	RMSE for train subset
RRMSE	Relative root mean square error
S_g	Gas saturation
S_w	Water saturation
SE	Squared exponential kernel function
SHAP	Shapley additive explanation
Sorg	Residual oil saturation associated with gas flooding
Sorw	Residual oil saturation associated with water flooding
T	Transpose
$TP_{predicted_i}$	Predicted value of target parameter for ith data point
$TP_{simulated_i}$	Simulated value of target parameter for ith data point
trainbr	Bayesian regularization backpropagation training algorithm
trainbfg	BFGS quasi-Newton training algorithm
traincgb	Conjugate gradient with Powell/Beale restarts training algorithm
trainlm	Levenberg-Marquardt training algorithm
trainrp	Resilient backpropagation training algorithm
trainscg	Scaled conjugate gradient training algorithm
V_i	Velocity of ith particle
V_{max}	Maximum velocity
V_{min}	Minimum velocity
w_p	Weight for pth neuron in the MLP hidden layer
WAG	Water alternating gas
X	Feature value
X_{max}	Maximum value of a feature
X_{min}	Minimum value of a feature
X_{norm}	Normalized value of feature in the range of 0–1
y_p	Output of pth neuron in the hidden layer of MLP
Z	The total number of data points
Z_{ts}	Number of data points for test dataset
Z_{tr}	Number of data points for train dataset
b	Constant value
b_p	Bias for pth neuron in the MLP hidden layer
c_1	Cognitive coefficient
c_2	Social coefficient
e	Error
f	Activation function
Gb	The best global position
$k(x_i,x)$	Kernel function
m20	Count of data points with a measured-to-predicted value ratio between 0.8 and 1.2
r_1, r_2	Random numbers within the range of 0 and 1
t	Current iteration in the PSO algorithm
x_i	Input signal in MLPNN, position in PSO algorithm

to augment the process of oil recovery while concurrently preventing substantial quantities of carbon emissions (Zhang and Lau, 2022). Ren and Duncan (2019) conducted the impact of injection techniques and reservoir heterogeneities on the performance of carbon dioxide (CO_2) sequestration in 11 sub-volumes of the San Andres Formation in West Texas. Their study discovered that reservoirs could attain greater CO_2 retention fractions by utilizing a combination of inverted five-spot well designs and large water alternating gas (WAG) ratios. This finding offers valuable insights for future projects involving the storage of CO_2 linked with enhanced oil recovery (EOR) in carbonate sequences. Wang et al. (2020) examined the effects of CO_2 flooding in glutenite reservoirs, with a specific emphasis on the influential elements of well designs and optimum injection procedures. Their findings suggested that the reservoir was efficiently constructed, resulting in excellent recovery efficiency. Furthermore, the displacement of CO_2

remained more evenly distributed, resulting in a wider range of sweeping and improving the entire effectiveness. Al-Mudhafar (2019) developed a proxy model for the prediction of CO_2-EOR in shale oil reservoirs.

1.1. Current state of research in CO_2-EOR and storage

The prediction of CO_2 storage and the total amount of oil produced over time is a significant field of study because of its potential influence on efforts to mitigate climate change and EOR. By integrating these processes, it is possible to decrease the levels of ambient CO_2 and enhance the extraction of oil from current reserves.

Contemporary studies of predictive modeling for CO_2 storage utilize several computational methodologies, such as reservoir simulation, machine learning (ML), and geostatistical methods. Reservoir simulation models, including ECLIPSE, CMG, and TOUGH2, are extensively employed to forecast the capacity of CO_2 storage. These models integrate geological data, fluid properties, and reservoir characteristics to simulate the behavior of CO_2 in underground environments (Ajayi et al., 2019; Liu et al., 2022; Ren et al., 2016). ML methods, such as neural networks, support vector machines, and random forests, have demonstrated potential in improving the accuracy of CO_2 storage predictions. These algorithms are applied to large datasets to identify patterns and enhance predictive abilities (Al-Shargabi et al., 2022; Vo Thanh et al., 2019, 2022). Furthermore, geostatistical techniques such as kriging and stochastic simulation are used to analyze the spatial variability and uncertainty in geological formations. These methods help evaluate reservoir heterogeneity and the risk of potential CO_2 leakage (Dai et al., 2020; Li and Zhang, 2014; Liberty et al., 2022).

CO_2-EOR is a technology that has the dual benefit of increasing oil recovery and simultaneously storing CO_2. The amount of oil produced in CO_2-EOR operations is influenced by reservoir features, injection tactics, and the availability of CO_2. Studies have shown that reservoirs with greater porosity and permeability are better suited for CO_2-EOR. Additionally, the presence of natural fractures can improve the effectiveness of CO_2 injection (Alvarado and Manrique, 2010; Le Van and Chon, 2017).

It is essential to optimize injection tactics, such as continuous CO_2 injection, WAG injection, and hybrid methods, in order to maximize the total amount of oil produced over time (Al-Khdheeawi et al., 2017; Chen et al., 2010; Nait Amar et al., 2021). The accessibility and quality of CO_2 sources have a significant influence on the practicality and cost-efficiency of CO_2-EOR projects. Research indicates that utilizing anthropogenic CO_2 from industrial sources can contribute to sustainability (Chen et al., 2020; Mudhafar et al., 2019; Ruprecht et al., 2014).

Technical, economic, and environmental variables have an impact on the effectiveness of CO_2 storage and EOR projects. The main technical difficulties involve understanding the reservoir, detecting the movement of CO_2, and tracking its migration. Recent advancements in seismic imaging and tracer technology have enhanced the ability to monitor and regulate CO_2 movements within subsurface reservoirs (Ajayi et al., 2019; Lumley, 2010; Susanto et al., 2016). The economic feasibility of CO_2-EOR projects relies on factors such as oil prices, expenses associated with CO_2 capture and transportation, and regulatory incentives. This emphasizes the importance of having a supportive regulatory framework and carbon pricing mechanisms in place to encourage widespread adoption of these projects (Lipponen et al., 2011). Although CO_2-EOR provides environmental advantages by decreasing atmospheric CO_2 levels, there are still issues regarding the possibility of leakage and the long-term stability of CO_2 storage. Current research endeavors to evaluate the enduring soundness of subsurface CO_2 storage locations and establish optimal strategies for minimizing potential hazards (Alves and Lima, 2021; Wilday et al., 2011).

Several research studies have specifically examined the environmental effects of CO_2-EOR, with a special emphasis on the possibility of induced seismic activity and contamination of groundwater (Balch and McPherson, 2016; Han et al., 2010). Studies suggest that careful selection and monitoring of subsurface reservoirs can reduce these hazards, guaranteeing that CO_2-EOR continues to be a secure and feasible choice for both enhanced oil recovery and carbon sequestration (Dai et al., 2014b; Ma et al., 2019).

1.2. Machine learning for CCS-EOR

However, one of the key challenges that has been a persistent issue in the field of CCS-EOR is the significant computational time required to build and evaluate reliable reservoir models (Shahkarami and Mohaghegh, 2020). The intricacy of unconventional residual oil zones poses particular challenges to the accurate simulation of fluid reservoir behavior, which is crucial for the comprehensive understanding of how CCS-EOR works in such reservoirs (Al-Mudhafar et al., 2022). The specific constraints posed by these reservoirs, which are characterized by complex geological formations with low permeability, require innovative solutions (Chen and Reynolds, 2015).

Customized ML techniques offer a fast and effective way to address these challenges (Lee, 2020). ML has the potential to address the challenge of comprehending and managing intricate reservoir dynamics by extracting valuable insights from extensive datasets (Bahrami and James, 2023). ML can be applied to CCS-EOR in various ways, including supervised learning, unsupervised learning, and reinforcement learning, each offer distinct insights (Yao et al., 2023). Unconventional CCS-EOR reservoirs under consideration exhibit substantial geological complexity, as they are characterized by substantial natural fracturing, faulting, and rock heterogeneity. This complexity introduces an additional level of intricacy to the dynamics of fluid flow (Xu et al., 2017).

Fluid-flow simulations are required to optimize CO_2 injection and improve oil recovery, particularly in unconventional reservoirs (Dai et al., 2014a). The intricacy of this complexity arises from the numerous interactions that occur throughout multiple phases, hence presenting a formidable challenge in developing correct predictions (Dang et al., 2015). The precise characterization of unconventional reservoirs is essential for determining their CCS-EOR effectiveness. Accurate comprehension of the spatial arrangements and distributions of geological characteristics within the reservoir is required (Chen and Pawar, 2018). To navigate these complexities, ML needs to be customized to process and interpret the wide range of geological and geophysical data available, enabling a comprehensive understanding of specific unconventional reservoirs (Chen and Pawar, 2019a).

Fluid-flow simulations can be evaluated with ML models of historical data from multiple reservoirs and simulations of them. This approach enables the development of reliable predictions that can be utilized to optimize strategies for CO_2 injection and enhance the recovery of oil resources (Vo Thanh et al., 2020). The integration of geological variables, rock–fluid interactions, and well operations with customized ML can effectively enhance the accuracy of predicting the performance of CO_2-EOR and storage (You et al., 2020), which is the focus of this study. The ML algorithms customized for this purpose are multilayer perceptron (MLP) and least-squares support-vector machines (LSSVM), each coupled with a grey wolf optimizer (GWO) and particle swarm optimization (PSO) to develop hybrid customized LSSVM-GWO, LSSVM-PSO, MLP-GWO, and MLP-PSO models.

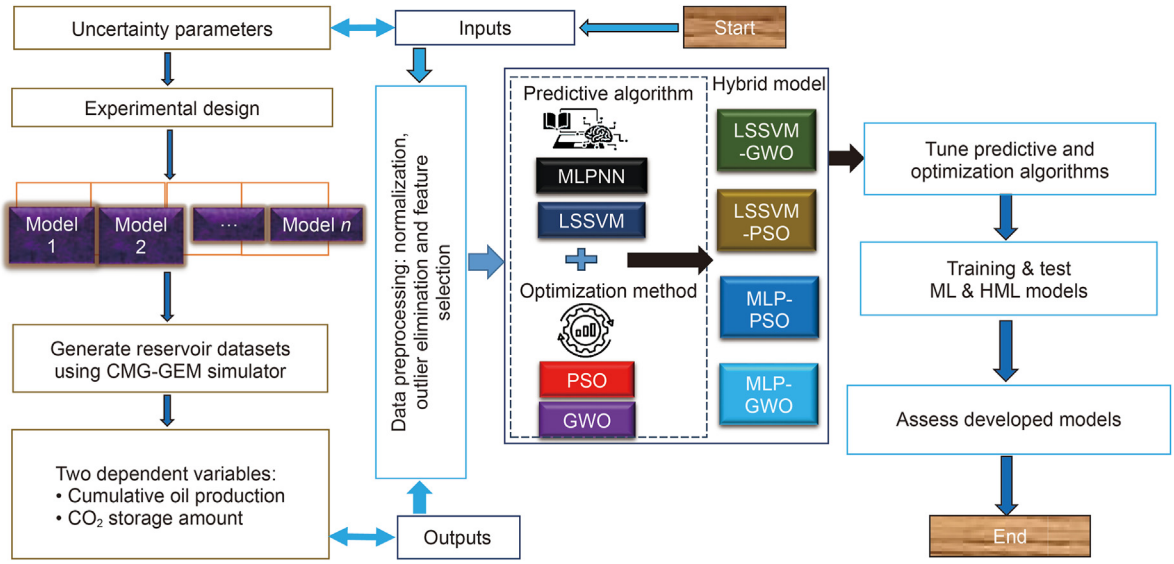

Fig. 1. Workflow summary of applied methods.

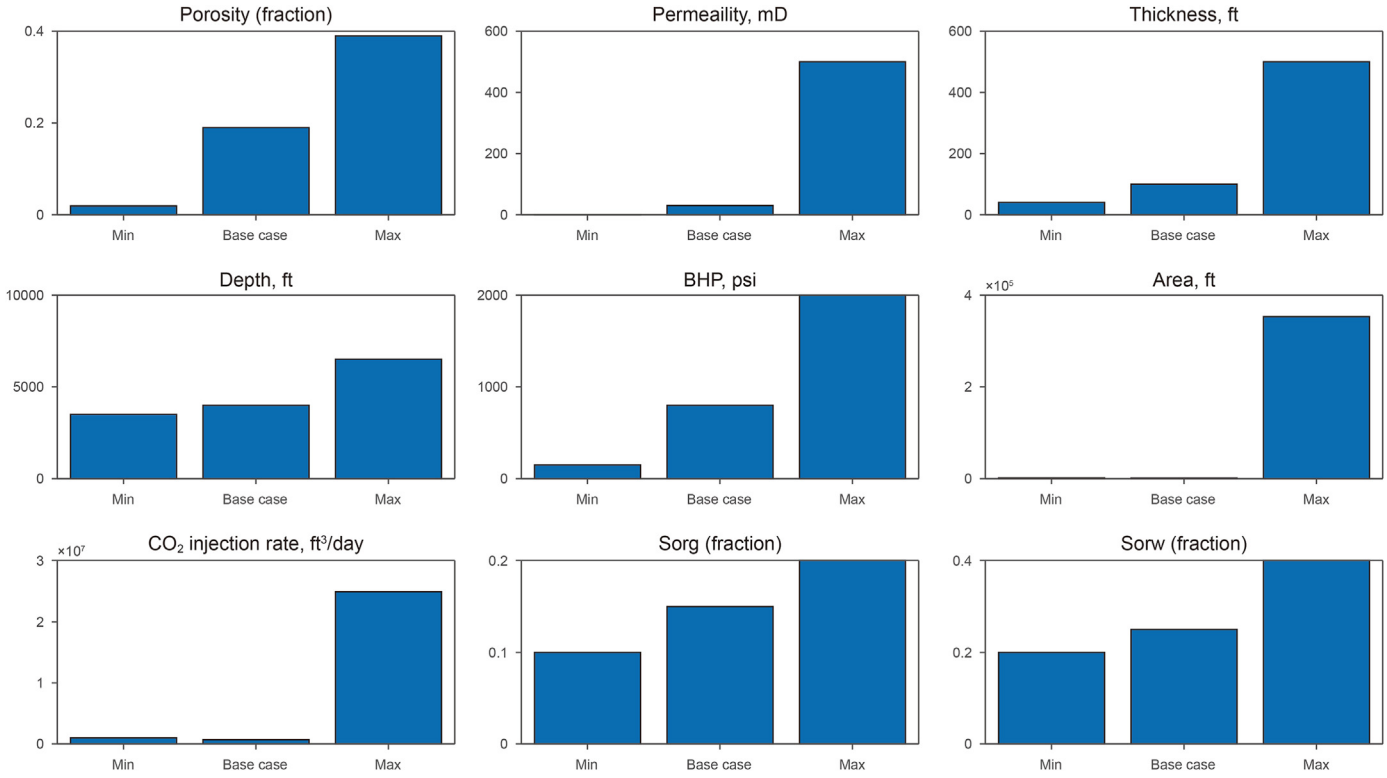

Fig. 2. Nine potential input features considered by the ML models.

The MLP (a type of artificial neural network) offers a well-developed ML method widely applied to CO_2 studies. These include CO_2 solubility in brine (Amar et al., 2019), estimating the thermal conductivity of CO_2 for CCS (Nait Amar et al., 2020), and modeling the viscosity of N_2-CO_2 mixtures (Naghizadeh et al., 2022). LSSVM regression models have also been widely applied to CO_2-related studies, including the prediction of CO_2 flooding performance in oil reservoirs (Ahmadi et al., 2018), the prediction of related CO_2 emissions from the oil and gas industry (Zhao et al., 2018), and the prediction of CO_2 trapping capacity in deep saline aquifers (Davoodi et al., 2023).

The GWO draws its inspiration from the social behavior of grey wolves (Duan and Yu, 2023). This optimizer has been combined with MLPs (MLP-GWO) to predict petrophysical changes during CO_2 injection in coal seams (Yan et al., 2020), and to predict oil production performance based on reservoir simulation (Ng and Jahanbani Ghahfarokhi, 2022). It has also been adopted for predicting the oil recovery factor during the WAG EOR process for reservoir simulations (Nait Amar et al., 2021). MLP-GWO is, therefore, a suitable candidate to customize for the prediction of oil

production and CO_2 storage performance in unconventional formations.

Combining MLP with PSO is an alternative way to enhance the selection of its control parameters. This method has been applied to predict CO_2-brine solutions (Amar et al., 2019), CO_2 solubility in water at high-temperature, high-pressure conditions (Hemmati-Sarapardeh et al., 2020), and oil production performance in shale oil reservoirs (Al-mudhafar, 2019). MLP-PSO is also a suitable candidate to customize for CCS-EOR predictions.

The LSSVM-GWO combination has been used to predict fluid-flow behavior in WAG process for stratified reservoir models (Andersen et al., 2022), modeling shale wettability characteristics for CCS purposes (Zhang et al., 2023), and establishing EOR potential in tight reservoirs (Wang et al., 2023). Likewise, LSSVM-PSO has been applied to effectively predict CO_2 solubility in ionic liquids (Dashti et al., 2018), the reservoir deliverability of gas from underground natural gas storage facilities (Thanh et al., 2022), and the properties of CO_2 in relation to CCS conditions (Ahmadi et al., 2016). LSSVM-GWO and LSSVM-PSO are, therefore, both suitable ML configurations for addressing oil production and CCS-related issues.

The noteworthy aspects of this study are.

- Providing a thorough comparative analysis of standalone and hybrid ML models, elucidating their respective advantages and drawbacks in the context of CCS-EOR in unconventional residual oil zones.
- Applying an innovative feature selection algorithm to identify and prioritize key features, thereby enhancing the simplicity and precision of ML configurations.
- Identifying effective ML-optimized combinations that substantially improve CCS-EOR prediction performance.
- Compiling a comprehensive reservoir simulation database consisting of 32,415 data points to provide statistical confidence in the trained and tested ML models developed for CCS-EOR predictions.
- Investigating the hitherto poorly understood aspects of CCS-EOR applied to unconventional residual oil zones.
- Combining the application of multiple prediction-performance algorithms to reveal the overfitting tendencies and feature importance, as well as the prediction accuracy of the customized ML models.

2. Methodology

The process outlined in Fig. 1 encompasses a hybrid ML approach for predicting both the mass of CO_2 storage and the cumulative oil production in residual unconventional formations. The primary input variables significantly affecting the uncertainty of these two target variables are identified through experimental design and reservoir simulations on the input data. Subsequently, the input data undergoes a preprocessing phase, which involves

Table 1
Base case input parameters for the reservoir simulation model.

Parameter	Base case
Porosity	0.2
Permeability, mD	30
Thickness, ft	100
Depth, ft	4000
BHP, psi	800
Area, ft^2	150,000
CO_2 injection rate, ft^3/day	1,000,454
Sorg	0.15
Sorw	0.25

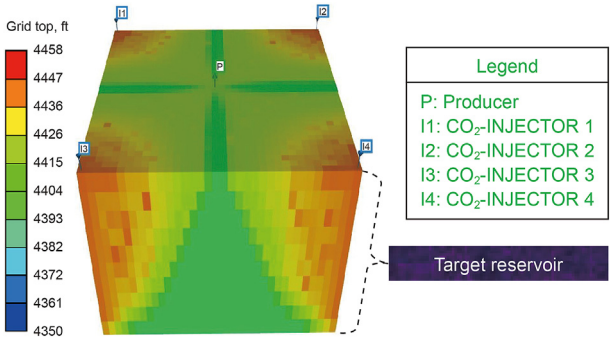

Fig. 3. A schematic representation of the reservoir simulation model employed.

normalizing data points within each variable and identifying and eliminating any outliers or erroneous data. The filtered data is split into a training group and a testing group. The optimal subset of features was selected to achieve a high prediction performance using the non-dominated sorting genetic algorithm (NSGA-II). The control parameters of the ML algorithms, i.e., multi-layer perceptron neural network (MLPNN) and least square support vector machine (LSSVM), are then fine-tuned. Hybrid ML algorithms (HML) combine the ML (MLPNN and LSSVM) algorithms with optimizers, configuring the integration of those two components to suit specific problem conditions. The control parameters of the particle swarm optimization (PSO) and grey wolf optimization (GWO) algorithms were fine-tuned using a trial-and-error method with training data to ensure that their configuration was suitable for the studied dataset.

With optimized parameters in place, the ML and hybrid ML models enter a training phase where they learn the underlying patterns and relationships within the training dataset, enabling them to make predictions on the specified target variables. Following the training process, these models are deployed to generate predictions on the target variables using testing datasets. The performance of each model is rigorously evaluated using multiple prediction performance criteria, which assess the levels of relative error and accuracy achieved by the models on both the training and testing datasets. These evaluation metrics are used to evaluate the degree of model overfitting during training and provide insights into the models' generalizability to new, unseen data. The sequential workflow involves data preprocessing, effective model training and testing, and comprehensive performance evaluation to make predictions regarding CO_2 storage and cumulative oil production in unconventional formations, while ensuring the models are robust and capable of generalization.

2.1. Input variables

A dataset consisting of 32,415 simulated samples was generated using the stochastic Latin hypercube sampling (LHS) technique. LHS produces quasi-random samples from multi-dimensional distributions of variables, particularly those characterized by substantial levels of uncertainty. This approach operates efficiently with large datasets by effectively sampling the full spectrum of variability. It also generates variable distributions suitable for ML evaluations (Stein, 1987). LHS selection of input variables and their respective is guided by sensitivity analysis. The results of previous studies have provided an understanding of many of the relationships between CO_2-enhanced oil recovery (CO_2-EOR) and storage, and these are taken into account. Sensitivity analysis helps identify which input variables have the most significant impact on the target variables and inform the choices made for the LHS technique. By leveraging

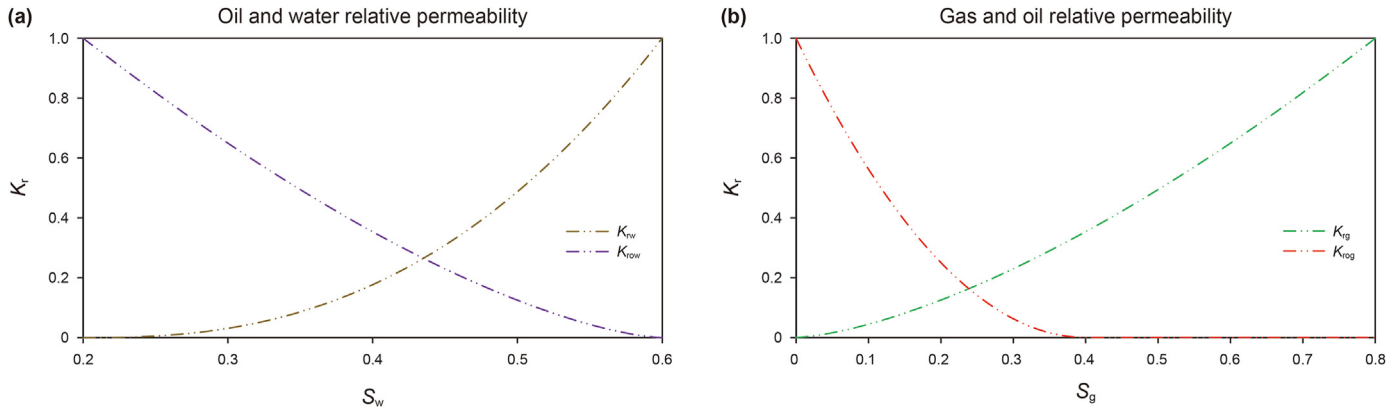

Fig. 4. The relative permeability data adopted for the reservoir simulation process.

the insights from these studies, the input variables are tailored to capture the key factors influencing CO_2-EOR and storage, making the LHS-generated dataset representative of a wide range of real-world scenarios (Abbaszadeh and Shariatipour, 2018; Al Eidan et al., 2015; Gibson-Poole et al., 2006; Lee et al., 2010; Liu and Zhang, 2011; Van Si and Chon, 2018; Vo Thanh et al., 2019).

The nine input features (Fig. 2) selected for consideration by the ML models were chosen based on sensitivity analysis and the results of published studies. These input variables include reservoir depth, porosity, permeability, reservoir size, thickness, bottom hole pressure (BHP), as well as the residual oil saturation associated with gas flooding (Sorg) and water flooding (Sorw). This selection ensures that the ML models have a comprehensive set of inputs that capture the complexities and uncertainties of CO_2-EOR and storage systems, thereby enhancing the reliability of their predictions.

2.2. Reservoir simulation case samples

The LHS sampling of the unconventional reservoir simulations was conducted with the Computer Modeling Group's greenhouse emissions model (CMG-GEM) reservoir simulator, version 2019 (CMG, 2019). CMG-GEM is a compositional reservoir simulator specifically designed for unconventional reservoirs. It employs the equation of state (EOS) and can accommodate three distinct fluid phases. To facilitate these simulations, a database was created, which incorporated the sampled values of the input variables and the computed values of the two target variables—namely, the quantity of CO_2 storage and the cumulative oil production volume—for each simulation run. To be clear, the predicted CO_2 storage volume refers to the quantity of CO_2 that remains stored within the reservoir after injection. This distinction is crucial for understanding the significance of the model's predictions. The CO_2 storage volume indicates the effectiveness of the reservoir in retaining CO_2, which is a critical factor in evaluating the success of subsurface carbon capture and storage (CCS) reservoirs.

2.3. Base case model description

The initial values of the input variables listed in Table 1 were employed to create the 3D reservoir simulation model.

Fig. 3 illustrates the synthetic simulation scenario in its base case, which comprises a total of 12,960 grid cells arranged in a $36 \times 36 \times 10$ grid structure. To provide a logical justification for these fundamental assumptions, they are based on the petrophysical parameters acquired from the Goldsmith-Landreth San Andres Unit (GLSAU) situated in the Permian Basin of Texas, United States. The choice of these assumptions is motivated by the desire to construct a simulation scenario that closely mirrors real-world conditions and geological characteristics found in the GLSAU. This alignment with actual geological data and characteristics enhances the validity and applicability of the findings obtained from this simulation to practical, real-world contexts. By emulating a specific geological setting, the simulation results become more relevant and reliable for drawing insights and making decisions in the context of the GLSAU and similar unconventional reservoirs (Chen and Pawar, 2018). In the simulation model developed, the CO_2 injection method was employed to evaluate both oil production and carbon storage within the GLSAU reservoir. This approach involves the injection of CO_2 into the subsurface reservoir, which serves two primary purposes: enhancing oil recovery (EOR) and storing CO_2 to mitigate greenhouse gas emissions.

The choice of relative permeabilities associated with gas and water saturations has been carefully considered to accurately capture fluid behavior within unconventional reservoirs (Fig. 4). These parameters are critical in modeling the movement and distribution of fluids within the reservoir. The simulation aims to faithfully represent how fluids behave by relying on credible data sources as its foundation (Trentham et al., 2015). By using data from trusted sources, the simulation can provide a more realistic portrayal of fluid dynamics, ensuring that it closely aligns with actual reservoir behavior and enhances the quality and applicability of the results obtained.

The oil phase composition examined in the model is comprised of ten distinct hydrocarbon molecular components: C_1, C_2, C_3, C_4, C_5, C_6, C_7–C_{13}, C_{14}–C_{20}, C_{21}–C_{28}, and C_{29+} (Chen and Pawar, 2018). The inclusion of these ten hydrocarbon components is sufficient to accurately represent the composition of hydrocarbons typically found in reservoirs such as the GLSAU. Each of these components is characterized by specific mole fractions, which are 0.3577, 0.0584, 0.0597, 0.0536, 0.0358, 0.0116, 0.2282, 0.081, 0.0416, and 0.072447, respectively (Chen and Pawar, 2019b). These detailed compositions of the oil phase make it possible to accurately model fluid behavior and phase equilibrium in the reservoir.

2.4. Data preprocessing

The compiled data from the reservoir simulation is divided into two categories: the training subset and the testing subset. The partitioning is done using an appropriate ratio, ensuring a sufficient amount of data is allocated to each subset. This division allows the model to learn patterns and relationships from the training data while providing an independent dataset for evaluating its performance. Normalization is then applied to the data within each category. This procedure standardizes the values of input features,

ensuring they are on a comparable scale. The process continues with an initial evaluation of the training subset. This evaluation aims to identify any outliers present in the dataset. Outliers, which are data points that significantly deviate from the norm, can have a detrimental effect on model performance. Therefore, they are identified and eliminated to prevent their influence on the subsequent modeling process. To achieve rapid and precise predictive models, the most influential input features are selected for ML modeling.

2.4.1. Data separation and normalization

Proper separation of data into training and testing subsets is a crucial step in the construction and evaluation of ML models. The testing set plays a vital role in assessing the model's performance on data records not involved in model training, and helps to identify the degree of overfitting associated with the trained models. Determining the optimal ratio for splitting data between the training and testing subsets is not a one-size-fits-all approach. It depends on various factors, including the dataset's size, the specific problem being addressed, and the analysis objectives. Therefore, in this research, a sensitivity analysis is conducted to identify the suitable ratio. Three different separation ratios for dividing the training/testing (70/30, 80/20, and 90/10) were systematically evaluated to determine the most effective configuration.

To ensure an appropriate assignment of ML hyperparameter values and consider the actual impact of each input feature on the dependent parameter, it is necessary to normalize the input feature values before applying machine learning algorithms. In this study, data normalization is achieved by employing the minimum (X_{min}) and maximum (X_{max}) values of each feature, as outlined in Eq. (1), to obtain the normalized value (X_{norm}) for each feature value (X). This normalization process transforms each data variable distribution to a range of 0–1.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

2.4.2. Outlier detection

Data outliers can exist due to various factors such as measurement errors, inherent variability, or rare and noteworthy observations. Their presence introduces noise into the dataset, which can distort the model's understanding gained from its learning routine and result in inadequate curve fitting. Furthermore, outliers can introduce bias into parameter estimation, compromising the model's accuracy. They undermine the model's robustness, impeding its ability to handle variations and diminishing its reliability. Additionally, outliers can hinder the model's ability to generalize when applied to new, unseen data. By removing outliers, the model can focus on the majority of the training data that represents the underlying patterns and relationships, thereby enhancing its predictive performance. Various techniques can be employed to detect and eliminate outliers, including statistical methods, visualization techniques, or domain knowledge-based approaches. It is important to carefully evaluate and scrutinize the outliers before removal to ensure their legitimacy and avoid discarding valuable information.

The detection of outlier data in the training dataset can be facilitated through the utilization of the Mahalanobis distance technique. The Mahalanobis distance technique was chosen for this study because it considers correlations between variables in multivariate datasets, normalizing the data based on a covariance matrix. It also effectively identifies outliers across multiple

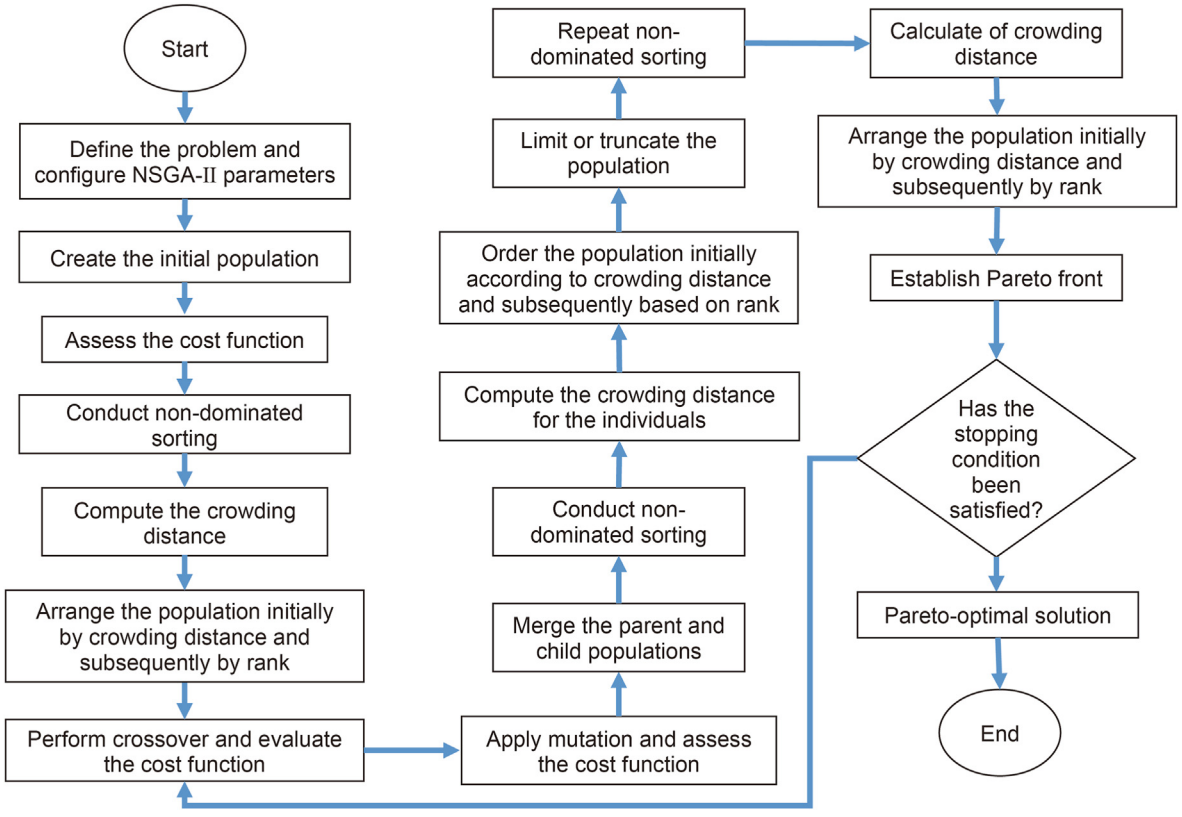

Fig. 5. Flowchart for implementation of non-dominated sorting genetic algorithm (NSGA-II) (Seyed Mostapha Kalami Heris, 2024; Anemangely et al., 2017).

Table 2
Advantages and disadvantages of applied ML algorithms.

Algorithm	Advantages	Disadvantages
MLPNN	• Capable of handling multiple input and output variables (Schmidhuber, 2015) • Highly flexible and capable of modeling complex non-linear relationships (Gurney, 2018) • Good at capturing intricate patterns in data through deep architectures (Goodfellow et al., 2016) • Beneficial for tasks where feature extraction is challenging (He et al., 2016) • Can approximate any continuous function given sufficient neurons and data (Schmidhuber, 2015)	• Prone to overfitting, especially with insufficient training data (Bishop and Nasrabadi, 2006) • Requires large amounts of data for training (Haykin, 1998) • Requires extensive hyperparameter tuning (e.g., number of layers, neurons per layer, learning rate) (LeCun et al., 2015) • Training can be computationally intensive and time-consuming (Hinton and Salakhutdinov, 2006) • Sensitive to initial weights and can get stuck in local minima (Suykens and Vandewalle, 1999)
LSSVM	• Efficient in terms of training time and computational resources (Smola and Schölkopf, 2004) • Capable of producing sparse models which are easier to interpret (Cortes and Vapnik, 1995) • Provides good generalization performance with less risk of overfitting (Vapnik, 1995) • Less sensitive to the curse of dimensionality due to support vectors (Schölkopf and Smola, 2002)	• Requires selection of a suitable kernel function, which can be non-trivial (Anemangely et al., 2019; Shawe-Taylor and Cristianini, 2004) • Performance is highly dependent on the choice of kernel and regularization parameters (Mehrad et al., 2020)

dimensions, and is computationally efficient for large datasets (Bishop and Nasrabadi, 2006), making it a practical solution for outlier detection. ML models are initially configured using all available input features to forecast CO_2 storage and cumulative oil production. A prediction error (Err) value for each data point is computed with Eq. (2) comparing the simulated ($TP_{simulated_i}$) and predicted ($TP_{predicted_i}$) values of the target parameters. The Mahalanobis distance for each data point is then calculated with Eq. (3), and a comparison is made with the threshold value established for the Mahalanobis distance (MD) in accordance with the problem conditions. If the Mahalanobis distance value for any given data point surpasses the predetermined threshold, then that data point is recognized as a potential outlier. Such data points are subsequently eliminated from the training dataset unless there is a good reason to justify their retention.

$$Err_i = TP_{simulated_i} - TP_{predicted_i} \quad (2)$$

$$MD_i = \sqrt{Err_i \times (Cov(Err))^{-1} \times Err_i^T} \quad (3)$$

The Gaussian process regression (GPR) algorithm is used to conduct the outlier detection procedure with the datasets compiled based on all the available input features. GPR is configured for the Mahalanobis distance technique to leverage its ability to model complex relationships in the data. The Mahalanobis distance is calculated based on the predictions and covariance relationships provided by the GPR model. The MATLAB code for the GPR-Mahalanobis distance technique developed is provided in Appendix A. GPR requires the selection of an appropriate kernel function as its key control parameter tuning.

2.4.3. Feature selection

ML input feature selection can be achieved by various methods, including the application of filters, wrappers, and combinations of those techniques. Filter methods rely on statistical indices such as correlation coefficients. Wrapper methods are slower because they require model evaluations of various feature combinations but tend to be more rigorous (Osman et al., 2018). Embedded techniques combine filters with wrapper methods. A hybrid algorithm, combining MLP with the second version of the non-dominated sorting genetic algorithm (NSGA-II), was configured in this study as a wrapper procedure. The NSGA-II functions to minimize the

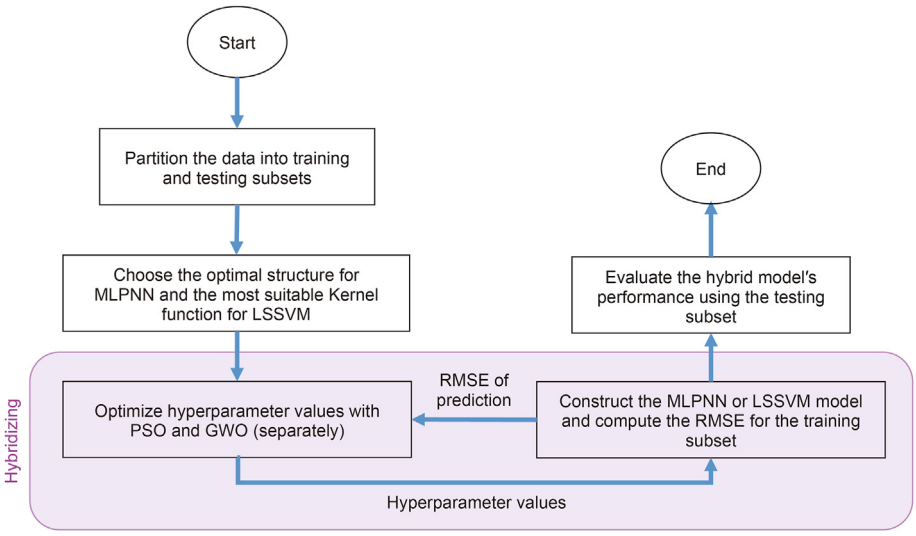

Fig. 6. The hybridization process of combining optimization and predictive algorithms.

number of selected features based on the root mean square errors (RMSE) generated.

The NSGA optimization method was introduced by Srinivas and Deb (1994) to solve multi-objective optimization problems. The key distinction between NSGA and the genetic algorithm lies in the selection operator (Subramanian et al., 2009). One of the primary challenges in solving such problems is the non-sortability of the feasible solutions set in the multidimensional space. Consequently, the algorithm relies on the domination concept, where a solution dominates others when there is no superior option. Additionally, a solution must clearly dominate other alternatives in at least one aspect (Coello et al., 2007; Souza et al., 2010; Subramanian et al., 2009). NSGA employs the fitness sharing method to handle solutions that provide similar performance levels, ensuring proper distribution and uniformity of solutions in the search space. Considering the sensitivity of NSGA's performance and quality to fitness-sharing parameters and other factors, Deb et al. (2002) introduced the second version of the algorithm, NSGA-II. This version utilizes crowding distance as an alternative method for fitness sharing, reducing the complexity of problem-solving (Deb et al., 2002; Du, 2012; Garrett, 2008; Knowles et al., 2007). Furthermore, NSGA-II employs a binary and elitism selection operator, recording and archiving non-dominated solutions from previous stages, thereby enhancing its performance compared to older versions (Deb et al., 2000).

Fig. 5 displays an implementation flowchart for NSGA-II. The algorithm selects some solutions from each generation based on the binary tournament selection method, comparing two randomly selected solutions from the population. Lower ranking and longer crowding distance indicate a better solution. The selected set of members from each generation undergoes iterative adjustments of the binary selection operator, participating in crossover and mutation. The resulting new population is again sorted based on ranking and crowding distance. A new population, equivalent in size to the previous one, is chosen from the top of the list, and the remaining members are discarded. This process repeats until the termination criterion is met. Non-dominated solutions derived from multi-objective optimization problems are selected from the Pareto front they establish. Among these solutions, none is inherently preferable, and the optimum decision depends on a problem's specified priorities.

In this study, each optimized feature selection on the Pareto front established by NSGA-II is evaluated further by the LSSVM algorithm configured to predict the target parameter. The resulting RMSE value of the LSSVM predictions is then processed by the NSGA-II algorithm as the cost associated with the selection of those features. To ensure a stable outcome in feature selection, the data employed for this purpose is partitioned into two randomly selected subsets, namely training (80% of all data points) and testing (20% of all data points). The evaluation of the target predictions by NSGA-II applies weighting coefficients, to yield a weighted RMSE. That weighting assigns 0.4 of the RMSE for the training subset predictions and 0.6 of the RMSE for the testing subset predictions (Eq. (4)). The MATLAB code for the LSSVM-NSGA-II feature selection technique developed is provided in Appendix B.

$$\text{Weighted RMSE} = 0.4 \times \text{RMSE}_{\text{tr}} + 0.6 \times \text{RMSE}_{\text{ts}} \tag{4}$$

2.5. Machine learning algorithms

Since the MLPNN and LSSVM algorithms used in this section are well-known and extensively documented, detailed explanations of these algorithms are omitted here. Instead, Table 2 presents the advantages and disadvantages of each algorithm, according to the cited published sources. Further description of these algorithms is provided in the supplementary file.

2.6. Hybridizing predictive and optimization algorithms

Achieving a model characterized by high precision and generalizability necessitates adjusting algorithm hyperparameters (Anemangely et al., 2018, 2019). In this study, the PSO and GWO metaheuristic algorithms are utilized to optimize weights and biases in the MLPNN and kernel-function hyperparameters in the LSSVM. They do so by adjusting decision variable (hyperparameter) values to minimize the target prediction error values.

In the MLPNN algorithm, determining the number of decision variables (the sum of weights and biases) necessitates initially establishing the appropriate network structure, which is performed via trial-and-error testing of diverse potential network structures. The determination of the number of decision variables by the optimization algorithm results in the introduction of optimal values to MLPNN in each iteration, leading to the creation of the MLPNN model. Subsequently, the model undergoes evaluation with the training data. The values of weights and biases are enhanced in each subsequent iteration of the optimization algorithm based on the feedback derived from the prediction error values generated in that iteration. This iterative process continues until the specified stopping conditions of the optimization algorithm are met, and the optimal values of weights and biases are then reported. Finally, an evaluation is conducted by applying the trained MLPNN model with these hyperparameter values to the testing subset. The entire process is illustrated as a flowchart in Fig. 6.

The functionality of the LSSVM algorithm is significantly influenced by the type of kernel function employed and the value of its regularization parameter. The kernel selection process involves a trial-and-error method. The values of the kernel's hyperparameters determined by grid search are extracted and treated as one of the population members in the optimization algorithm. Across multiple iterations, the optimization algorithm refines these hyperparameter values, aiming to align the LSSVM algorithm output with the measured values of the target parameter (Fig. 6).

In determining the optimal structure (MLPNN) and suitable kernel function (LSSVM), 10-fold cross-validation is utilized to mitigate the initial randomization of MLPNN and LSSVM hyperparameters. In this context, the training data is divided into ten equal parts, and the MLPNN and LSSVM algorithms are independently executed ten times. During each execution, one category serves as the testing subset, while the remaining nine categories function as the training subset, ensuring that each data record is utilized as part of the testing subset at least once. Ultimately, the mean value derived from these ten executions is regarded as the expected error for that specific MLPNN structure or the LSSVM kernel function.

2.7. Evaluation criteria

A wide range of prediction-performance evaluation criteria are evaluated to assess the performance of models during both the training and testing phases, aiming to identify the most suitable model for the task at hand. These evaluation criteria can be categorized into positive criteria and negative criteria. Positive criteria, such as the correlation coefficient and determination coefficient, indicate better model performance when their values are higher. Conversely, negative criteria, primarily error-based metrics, indicate superior model performance when their values are lower.

One commonly used error measure involves calculating the

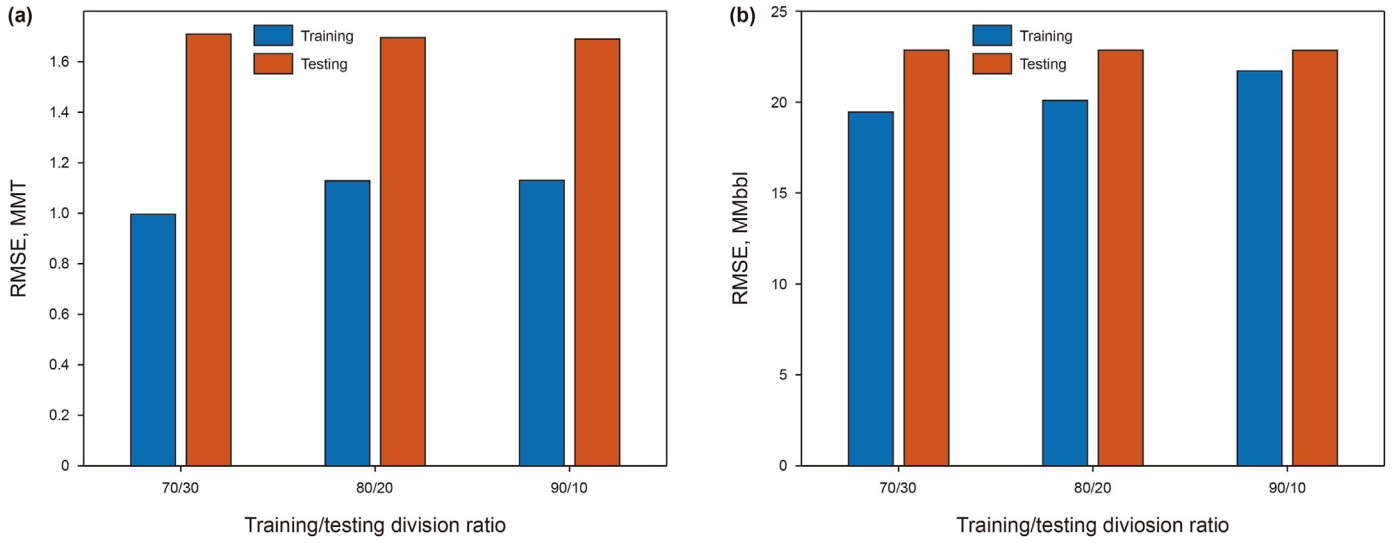

Fig. 7. Examining differences in RMSE across LSSVM models created using different ratios for the division of training and testing data in the prediction of CO_2 storage (**a**) and cumulative oil production (**b**).

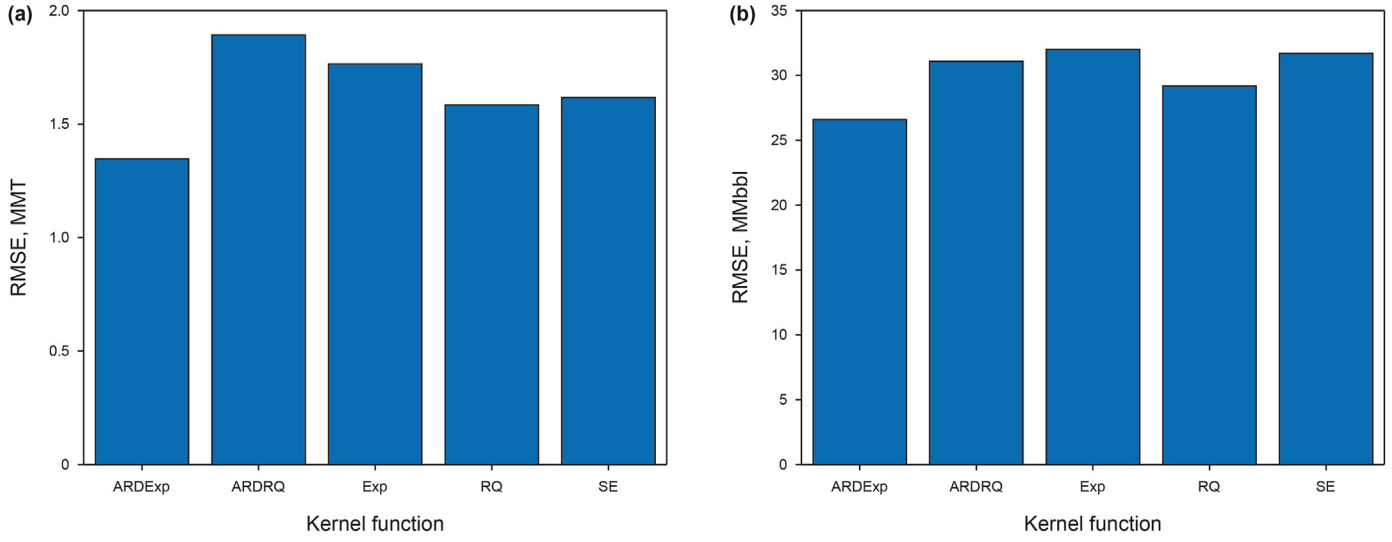

Fig. 8. Analyzing the disparities in RMSE values generated by models using various kernel functions of the GPR algorithm for the modeling of both CO_2 storage (**a**) and cumulative oil production (**b**).

difference between simulated values and predicted values of the target parameter, CO_2 storage and cumulative oil production, using Eq. (2). This serves as the basis for determining the relative error (RE, Eq. (5)) and the average relative error (ARE, Eq. (6)). Additionally, the root mean squared error (RMSE, Eq. (7)) is a widely accepted error measure that provides a comprehensive understanding of model performance due to its unit consistency. (Gandomi et al., 2011) introduced a performance index (PI, Eq. (8)) for model evaluation, which combines the correlation coefficient (R) and the relative root mean square error (RRMSE). A lower PI value indicates superior model performance. The RRMSE in the PI is calculated using Eq. (9) based on RMSE and the average of the simulated values of the target parameter ($\overline{TP_{simulated_i}}$).

One widely used positive metric is the coefficient of determination (R^2), which is calculated using Eq. (10). Another positive measure is the a20-index, which provides insights into the concentration of data points around the Y(predicted) = X(measured) line. This metric involves counting data points (m20) with a measured-to-predicted value ratio between 0.8 and 1.2. This count is then divided by the total number of data points (Z) used in both the training and testing stages, as described in Eq. (11).

$$RE = \frac{Err_i}{TP_{simulated_i}} \times 100 \tag{5}$$

$$ARE = \frac{\sum_{i=1}^{Z} RE}{Z} \tag{6}$$

$$RMSE = \sqrt{\frac{1}{Z}\sum_{i=1}^{Z} Err_i^2} \tag{7}$$

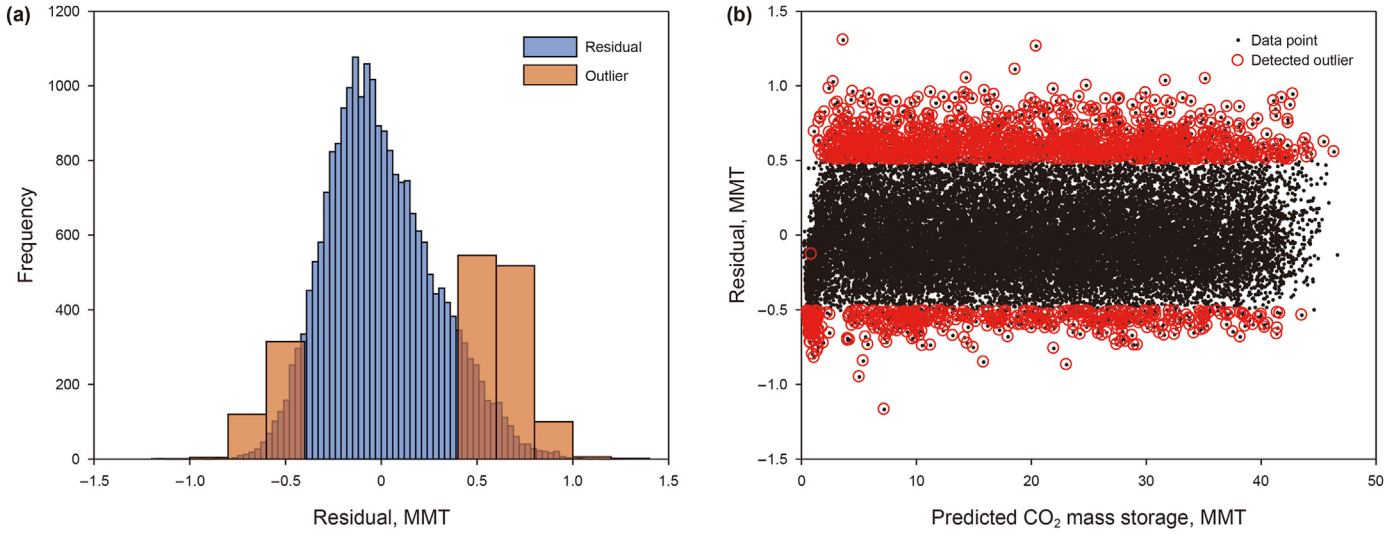

Fig. 9. Visualization of identified outliers within the training subset for CO_2 mass storage: (**a**) Frequency distribution; (**b**) Error residuals.

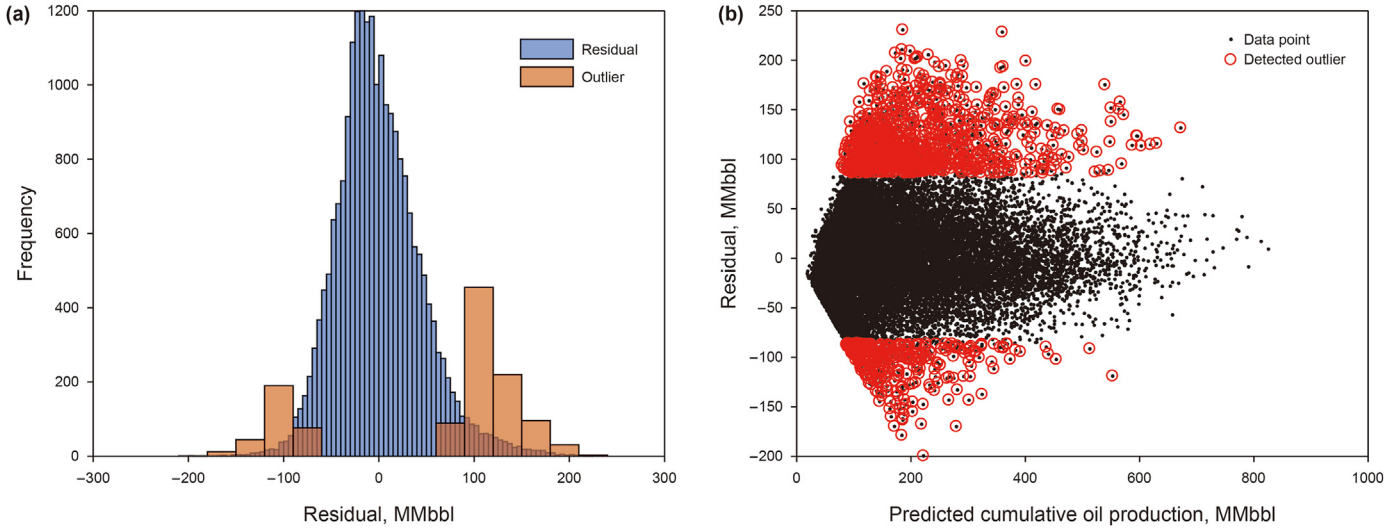

Fig. 10. Visualization of identified outliers within the training data set for cumulative oil production volume: (**a**) Frequency distribution; (**b**) Error residuals.

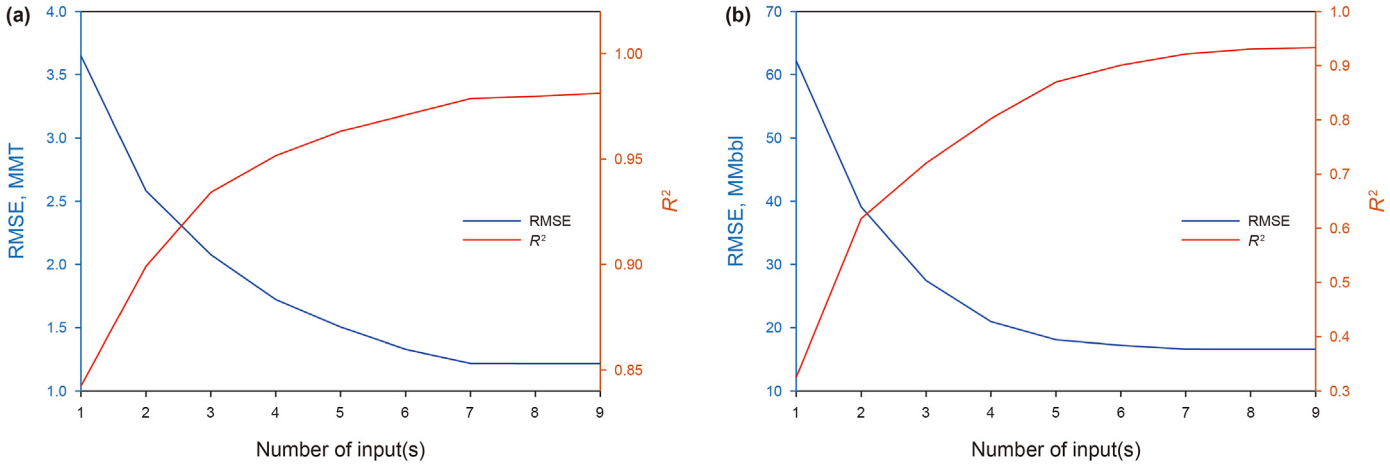

Fig. 11. Variations in RMSE and R^2 are used to select the optimal input feature combinations for modeling: (**a**) CO_2 storage mass; (**b**) Cumulative oil production.

Table 3
Optimal feature sets are determined based on various feature combinations used for CO_2 storage modeling, showing prediction performance in terms of RMSE and R^2 values.

Number of inputs	Selected feature(s)	RMSE	R^2
1	Area	3.6495	0.8423
2	Area, Por	2.5819	0.8991
3	Area, Por, Perm	2.0773	0.9342
4	Area, Por, Perm, BHP	1.7221	0.9516
5	Area, Por, Perm, BHP, InjRate	1.5043	0.9632
6	Area, Por, Perm, BHP, InjRate, Thickness	1.3285	0.9708
7	Area, Por, Perm, BHP, InjRate, Thickness, Sorg	1.2186	0.9786
8	Area, Por, Perm, BHP, InjRate, Thickness, Sorg, Depth	1.2181	0.9796
9	Area, Por, Perm, BHP, InjRate, Thickness, Sorg, Depth, Sorw	1.2179	0.9811

Table 4
The optimal feature sets are determined based on various feature combinations used for cumulative oil production modeling, showing prediction performance in terms of RMSE and R^2 values.

Number of inputs	Selected feature(s)	RMSE	R^2
1	Por	44.1696	0.3254
2	Por, Perm	35.0749	0.6179
3	Por, Perm, Area	27.4408	0.7203
4	Por, Perm, Area, InjRate	20.9376	0.8022
5	Por, Perm, Area, InjRate, BHP	18.0583	0.8698
6	Por, Perm, Area, InjRate, BHP, Thickness	17.1967	0.9005
7	Por, Perm, Area, InjRate, BHP, Thickness, Sorg	16.5843	0.9213
8	Por, Perm, Area, InjRate, BHP, Thickness, Sorg, Sorw	16.5741	0.9305
9	Por, Perm, Area, InjRate, BHP, Thickness, Sorg, Sorw, Depth	16.5686	0.9332

$$\mathrm{PI} = \frac{\mathrm{RRMSE}}{1+R} \tag{8}$$

$$\mathrm{RRMSE} = \frac{\mathrm{RMSE}}{|\overline{\mathrm{TP}_{\mathrm{simulated}}}|} \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{Z} \mathrm{Err}_i^2}{\sum_{i=1}^{Z} \left(\mathrm{TP}_{\mathrm{predicted}_i} - \frac{\sum_{i=1}^{Z} \mathrm{TP}_{\mathrm{simulated}_i}}{Z} \right)^2} \tag{10}$$

$$\mathrm{a20\text{-}index} = \frac{\mathrm{m20}}{Z} \tag{11}$$

3. Results

3.1. Preprocessing of data

Considering the primary objective of achieving optimal accuracy in predicting the target parameter within unseen data through the development of predictive models, it becomes imperative to assess the performance of these models on data not utilized during the training phase. To facilitate this evaluation, it is common practice to reserve a portion of the collected data for testing, as not all data is employed in model training. Consequently, following model development, performance evaluation is conducted using this reserved test data. In this study, data separation into training and

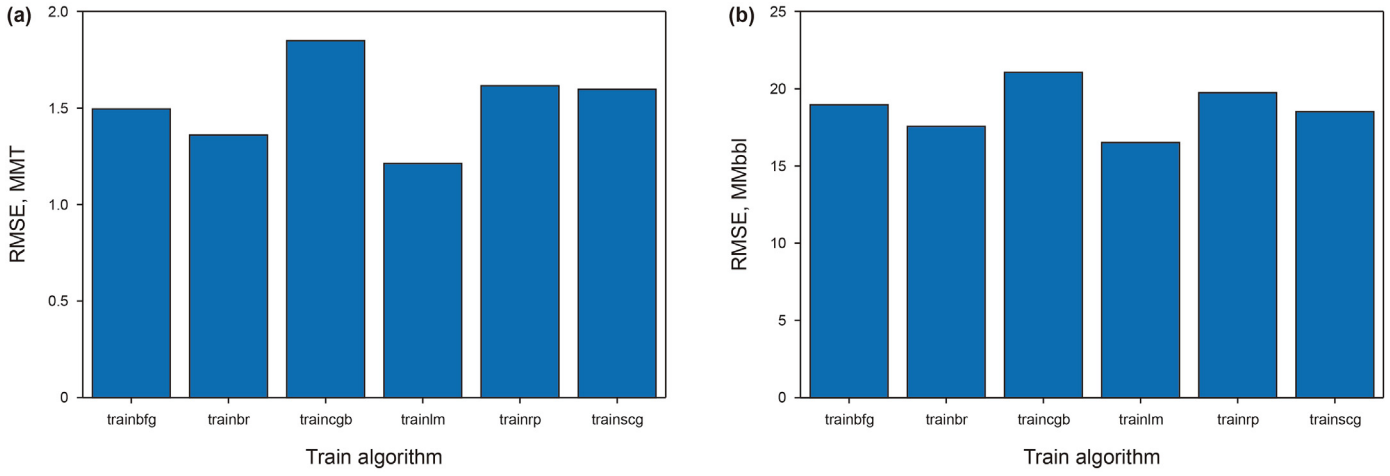

Fig. 12. Comparisons of RMSE values among various training algorithms implemented with MLPNN models for CO_2 storage (**a**) and cumulative oil production (**b**) predictions.

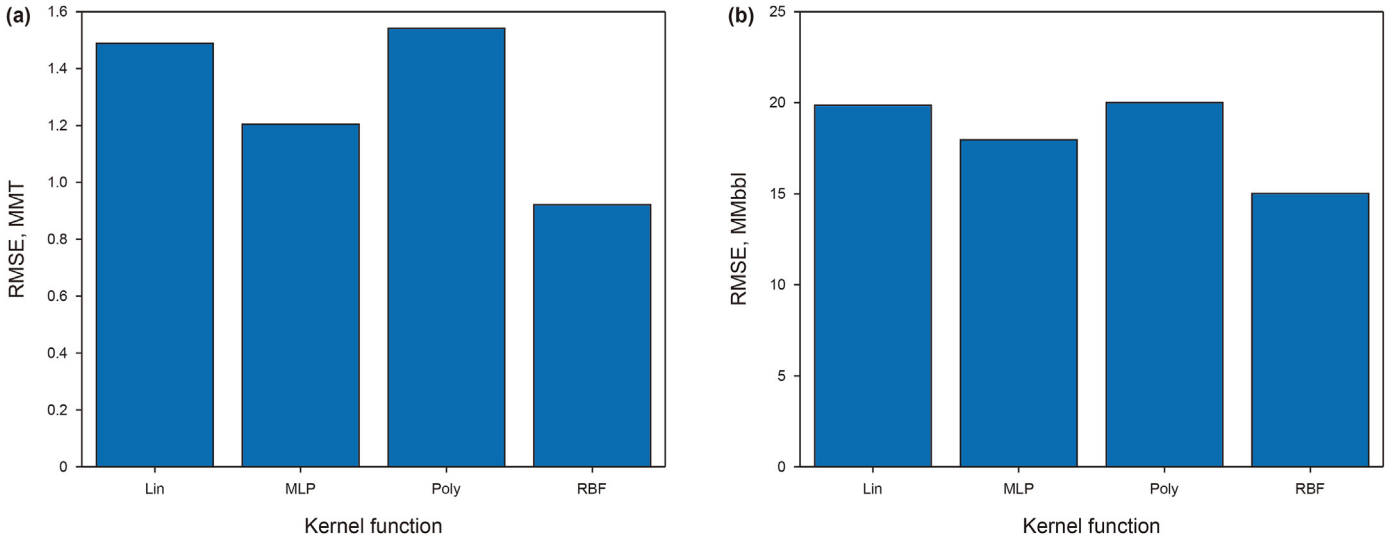

Fig. 13. Comparison of RMSE values for LSSVM models generated with various kernel functions in predicting CO_2 storage (**a**) and cumulative oil production (**b**).

Table 5
Controllable parameter values in optimization algorithms hybridized with predictor algorithms.

Parameter	MLPNN		LSSVM	
	PSO	GWO	PSO	GWO
Maximum iteration	200	200	200	200
Population size	60	50	50	40
w (inertia weight)	0.97	–	0.98	–
c_1	2.05	–	2.05	–
c_2	2.05	–	2.05	–

Table 6
The optimum values of hyperparameters for simple and hybrid LSSVM models for the prediction of CO_2 storage mass and cumulative oil production.

Target parameter	Optimization algorithm	σ	C
CO_2 storage	Grid search	101.5119	724216.0455
	PSO	156.8406	715295.4188
	GWO	161.5274	720518.6049
Cumulative oil production	Grid search	31.4095	175.7268
	PSO	48.0749	163.8495
	GWO	41.2685	169.8374

testing subsets precedes the modeling process. A critical consideration in this procedure is determining the appropriate ratio for data division into training and testing sets, as it significantly impacts the accuracy and generalizability of the models. Fundamentally, the training data category should encompass sufficient qualitative and quantitative diversity, enabling the model to generalize effectively when applied to unseen data. To achieve an optimal data separation, three scenarios employed in previous studies, specifically 70/30, 80/20, and 90/10, were taken into account for this study's data division. Following data separation, LSSVM models were developed using the training subset and subsequently evaluated on the testing subset.

Fig. 7 illustrates the comparison of RMSE values among the developed LSSVM models at different separation ratios. The figure indicates that as the volume of training data increases, the error during the training phase also rises, while the error during the testing phase exhibits minimal variation. This observation could be attributed to the limited divergence in data values with the expansion of the learning dataset, resulting in similar error values on the testing data. Based on this outcome, a data separation ratio of 70% (22,690 data points) for training and 30% (9725 data points) for testing is employed.

The training subset is evaluated to identify and potentially remove outlying data records. Since, in practical scenarios, test data is supplied post-model development, the analysis of outliers is exclusively performed on the training data to yield results consistent with actual outcomes. The Mahalanobis method is applied to each of the compiled simulation datasets: one with CO_2 mass storage as the target, the other with cumulative oil production as the target. Recognizing that the outcome of this assessment hinges significantly on the kernel function employed in the GPR algorithm, diverse models of this algorithm featuring distinct kernel functions were generated, and their RMSE values were assessed. Fig. 8 displays the GPR model results identifying the ARDExp (ARD exponential) kernel function as the kernel that yields the minimum error for both CO_2 storage and cumulative oil production prediction models. The ARDExp kernel outperforms the exponential (Exp), squared exponential (SE), rational quadratic (RQ), and ARD rational quadratic (ARDRQ)) kernels.

In Figs. 9 and 10, the outliers identified are depicted according to the outcomes of the GPR model developed with the selected kernel function applying the Mahalanobis method to the training data. The identified outliers are distinguished by their higher residual error values. Consequently, a total of 1613 (~7.1% of the training subset) and 1219 (~5.4% of the training subset) data points recognized as outliers were excluded from the training data for the CO_2 dataset and cumulative oil production, respectively. The determination of the number of outliers is based on adopting a threshold set at three times the average Mahalanobis distance across all data points. In this analysis, it is important to highlight the utilization of the 10-fold cross-validation method, a measure implemented to mitigate the impact of random data selection for training and testing on outlier detection outcomes. The consistent results across the ten modeling iterations demonstrate the stability of the outlier detection process. This uniformity can be attributed to the substantial number of data records involved in the training steps.

Following the removal of outliers, the NSGA-II algorithm was

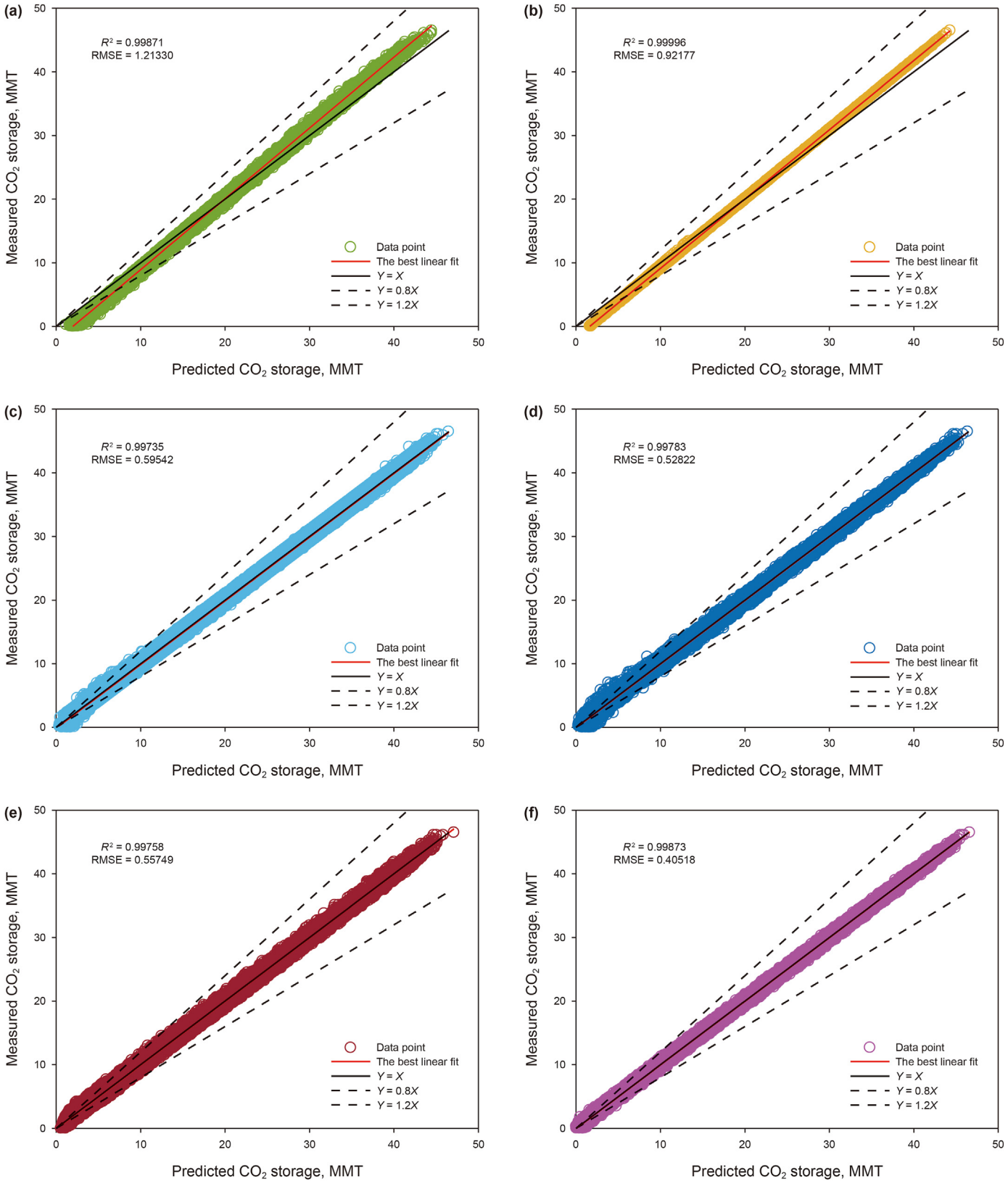

Fig. 14. Cross plots illustrating the comparison between simulated and predicted CO_2 storage mass values utilizing MLPNN (**a**), LSSVM (**b**), MLP-PSO (**c**), MLP-GWO (**d**), LSSVM-PSO (**e**), and LSSVM-GWO (**f**) models on the training subset.

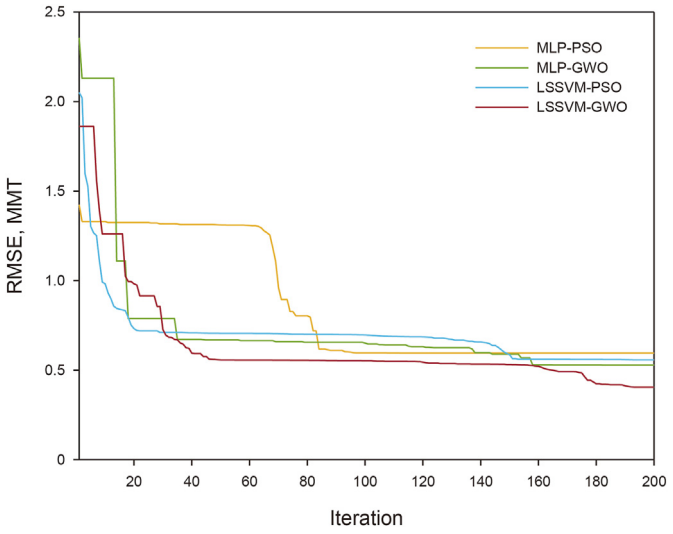

Fig. 15. Variations in RMSE across successive iterations of hybrid model optimizers in the prediction of CO_2 storage mass for the training subset.

applied to select features with a more pronounced impact on the target parameters. For this purpose, an LSSVM model with an RBF kernel was used to evaluate the selected features combined with the NSGA-II algorithm. Fig. 11 displays the variations in RMSE and R^2 values generated by different feature combinations with the LSSVM-NSGA-II predictive models for CO_2 storage and cumulative oil production. Up to combinations involving seven features, an increase in the number of features corresponds to an improvement in prediction performance. However, prediction performance improvements are minimal for combinations involving more than seven features. Hence, datasets involving just seven input features (excluding features "depth" and "Sorw") were used for detailed ML modeling.

Tables 3 and 4 summarize the prediction performance of different feature combinations applied to the two datasets. Irrespective of the prediction target, the prediction performance achieved by the seven-feature combination (injection rate (InjRate), permeability (Perm), Thickness, porosity (Por), BHP, Area, and Sorg) is only marginally improved by adding features depth and Sorw for CO_2 storage mass and cumulative oil production (see Table 4).

3.2. Tuning the controllable parameters of ML algorithms

Diverse MLPNN models (for both CO_2 storage mass and cumulative oil production) were initially generated, varying in the number of hidden layers between 1 and 3, and varying the number of neurons in each MLPNN layer from 2 to 9 in each hidden layer. The analysis identified the optimum MLPNN structure for the studied dataset. It consists of three hidden layers with 7, 5, and 5 neurons in the first, second, and third layers, respectively. This MLPNN structure yielded the lowest RMSE values for both training and testing subsets. Hence, this specific structure was selected for detailed analysis.

The assessment of various MLPNN training algorithms revealed that the Levenberg-Marquardt (trainlm) algorithm generated the least errors (Fig. 12).

With respect to MLPNN layer activation functions, three options (hyperbolic tangent, linear, and logistic sigmoid) were evaluated for each hidden layer. The results identified that the best prediction performance was achieved by applying the sigmoid activation function to the first and second hidden layers, and the linear activation function to the last hidden layer.

Four LSSVM kernel-function options were evaluated (radial basis function (RBF), linear (Lin), polynomial (Poly), and multi-layer perceptron (MLP)) for both CO_2 storage mass and cumulative oil production datasets. Fig. 13 reveals that the LSSVM models with the RBF kernel generate the lowest RMSE values for both datasets, so that kernel was selected for both models.

A grid-search algorithm was used to determine the optimal values of LSSVM hyperparameters: (1) RBFwidth (σ), which controls the influence of each support vector on the decision boundary; and, (2) the regularization parameter (C), which controls the trade-off between achieving a small training error and having a simpler model.

The efficacy of hybrid algorithms, alongside predictive algorithms, is contingent upon the performance of the GWO and PSO optimization algorithms. Therefore, the configuration of control parameters in optimization algorithms needs to be established in accordance with the problem conditions. To achieve this, a trial-and-error method was employed, resulting in the determination of suitable values for the control parameters in the PSO and GWO algorithms, as outlined in Table 5.

In the analyses performed for simple and hybrid LSSVM, the best hyperparameter values obtained for the kernel functions are presented in Table 6 for the models that predict CO_2 storage mass and cumulative oil production.

The development of detailed predictive models for CO_2 storage mass and cumulative oil production involved the application of both simple and hybrid ML algorithms, configured with optimized hyperparameters, to the selected features of the pre-processed datasets.

3.3. Developing predictive models for CO_2 storage mass

Fig. 14 displays cross plots of simulated and predicted CO_2 storage mass values for each developed model. The best-fit line through the data points of the simple ML models deviates slightly

Table 7
Comparison of prediction performance for ML and hybrid ML models applied to the CO_2 storage mass training subset.

Type	Model	ARE	RMSE, MMT	RRMSE	R^2	PI	a20-index
Simple	MLP	0.6973	1.2133	0.0618	0.9987	0.0309	0.8667
	LSSVM	0.5922	0.9218	0.0470	0.9999	0.0235	0.8914
Hybrid	MLP-PSO	0.4180	0.5954	0.0303	0.9974	0.0152	0.9382
	MLP-GWO	0.3662	0.5282	0.0269	0.9978	0.0134	0.9458
	LSSVM-PSO	0.2461	0.5575	0.0284	0.9976	0.0142	0.9528
	LSSVM-GWO	0.1365	0.4052	0.0206	0.9987	0.0103	0.9657

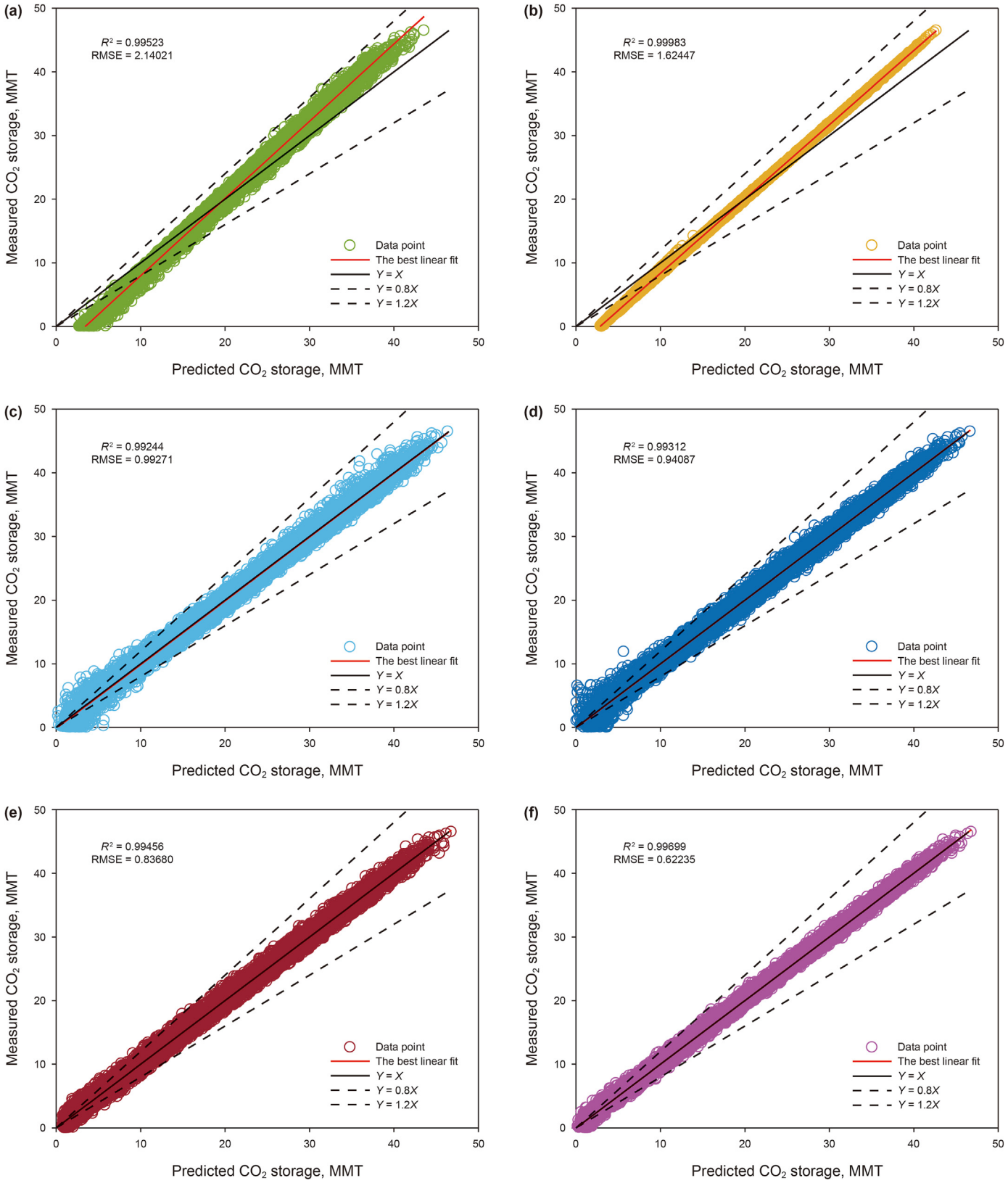

Fig. 16. Cross plots illustrating the comparison between simulated and predicted CO_2 storage mass values utilizing MLPNN (**a**), LSSVM (**b**), MLP-PSO (**c**), MLP-GWO (**d**), LSSVM-PSO (**e**), and LSSVM-GWO (**f**) models on testing data.

Table 8
Comparison of prediction performance for ML and hybrid ML models applied to the CO_2 storage mass testing subset.

Type	Model	ARE	RMSE, MMT	RRMSE	R^2	PI	a20-index
Simple	MLP	1.9875	2.1402	0.1087	0.9952	0.0544	0.7982
	LSSVM	1.5273	1.6245	0.0825	0.9998	0.0412	0.8268
Hybrid	MLP-PSO	1.4344	0.9927	0.0504	0.9924	0.0252	0.9023
	MLP-GWO	1.2797	0.9408	0.0478	0.9931	0.0239	0.9050
	LSSVM-PSO	0.4950	0.8368	0.0425	0.9946	0.0213	0.9239
	LSSVM-GWO	0.5333	0.6224	0.0316	0.9970	0.0158	0.9420

from the Y(predicted) = X(simulated) line, both at high and low values of CO_2 storage mass. This indicates that, in low values of CO_2 storage mass, these models tend to overestimate values, while for high CO_2 storage mass cases, they tend to underestimate values. In contrast, the best-fit lines for the hybrid models coincide precisely with the $Y = X$ line.

Fig. 15 displays the trend of RMSE reduction per optimizer iteration for the CO_2 storage mass prediction for the hybrid models applied to the training subset. The GWO algorithm converges to lower RMSE values than the PSO, as recorded in Fig. 15.

Table 7 displays the prediction performance metrics for simple and hybrid ML models applied to the training subset for CO_2 storage mass. The hybrid models generate better prediction performance than the simple ML models, with the LSSVM-GWO generating the lowest prediction errors.

Fig. 16 displays cross plots of simulated versus predicted values of CO_2 storage mass using both simple and hybrid models for the testing subset. The best-fit line for the simple ML models deviates from the $Y = X$ line, indicating an overestimation at low values of CO_2 storage mass and an underestimation at high values of CO_2 storage mass. Given the occurrence of this phenomenon during the training phase, its manifestation in unseen data is expected. Hence, caution is advised when employing simple models on unseen data. In contrast, the best-fit line in the hybrid models consistently aligns with the $Y = X$ line, signifying the reliable overall performance of the trained models applied to unseen data.

Table 8 displays the prediction performance metrics for simple and hybrid ML models applied to the testing subset for CO_2 storage mass. The hybrid models generate better prediction performance than the simple ML models, with the LSSVM-GWO generating the lowest prediction errors.

3.4. Developing the predictive models for cumulative oil production

Fig. 17 displays cross plots of simulated versus predicted values for cumulative oil production values for each developed model. The best-fit line through the data points of the simple ML models deviates slightly from the Y(predicted) = X(simulated) line, both at high and low values of cumulative oil production. This indicates that the cumulative oil production is underestimated at high values and overestimated at low values. In contrast, the best-fit lines for the hybrid models plot more closely to the $Y = X$ line, especially for the LSSVM-GWO model.

Fig. 18 displays the trend of RMSE reduction per optimizer iteration for the cumulative oil production prediction for the hybrid models applied to the training subset. The GWO algorithm converges to lower RMSE values than the PSO, as recorded in Fig. 18.

Table 9 displays the prediction performance metrics for simple and hybrid ML models applied to the training subset for cumulative oil production. The hybrid models generate better prediction performance than the simple ML models, with the LSSVM-GWO generating the lowest prediction errors.

Fig. 19 displays cross-plots of simulated versus predicted cumulative oil production values using both simple and hybrid models for the testing subset. In this visual representation, the best-fit line diverges from the $Y = X$ line, indicating that the developed models tend to overestimate cumulative oil production at lower values and underestimate it at higher values. The occurrence of this phenomenon in the testing stage aligns with similar outcomes with the training subset. The degree of deviation is less pronounced for the hybrid models than the simple ML models, with the LSSVM-GWO model generating the least deviation.

Table 10 displays the prediction performance metrics for simple and hybrid ML models applied to the testing subset for cumulative oil production. The hybrid models generate better prediction performance than the simple ML models, with the LSSVM-GWO generating the lowest prediction errors. This observation suggests that the trained LSSVM-GWO model possesses a higher generalization ability compared to the other developed models, highlighting its effectiveness in making accurate predictions beyond the training subset.

4. Discussion

Further insight into the prediction performance of the developed models with the two datasets can be gained by considering an over-fitting index, integrated performance scoring, and partial dependence analyses of the input features.

4.1. Over-fitting analysis of developed models

An over-fitting index (OFI) (Gandomi and Roke, 2015) is applied, as defined in Eq. (12), that relies on the PI criterion during both the training (PI_{tr}) and testing (PI_{ts}) stages, adjusted for the relative quantity of data records in the training (Z_{tr}), testing (Z_{ts}), and overall (Z) sets. Given that a lower PI value signifies a model with superior prediction performance, the OFI relationship presented serves to standardize the PI value for the model across both training and testing phases. Therefore, a low OFI value indicates that a prediction model exhibits reduced overfitting.

$$\mathrm{OFI} = \left(\frac{Z_{tr} - Z_{ts}}{Z}\right)\mathrm{PI}_{tr} + 2\left(\frac{Z_{ts}}{Z}\right)\mathrm{PI}_{ts} \tag{12}$$

Table 11 displays OFI values corresponding to the performance of the developed models for predicting CO_2 storage and cumulative oil production. The simple models exhibit higher OFI values compared to hybrid models. Specifically, the MLPNN model generates the highest OFI value, signifying a notable level of overfitting, while the LSSVM-GWO hybrid model generates the lowest OFI value, suggesting a reduced tendency for overfitting. Based on the outcomes of this analysis, the models can be ranked in terms of overfitting severity as follows: MLPNN (highest) > LSSVM > MLP-PSO > MLP-GWO > LSSVM-PSO > LSSVM-GWO (lowest). This implies that LSSVM models are likely to provide more resilient prediction performance across various datasets or scenarios, making them more generalizable than the MLPNN models.

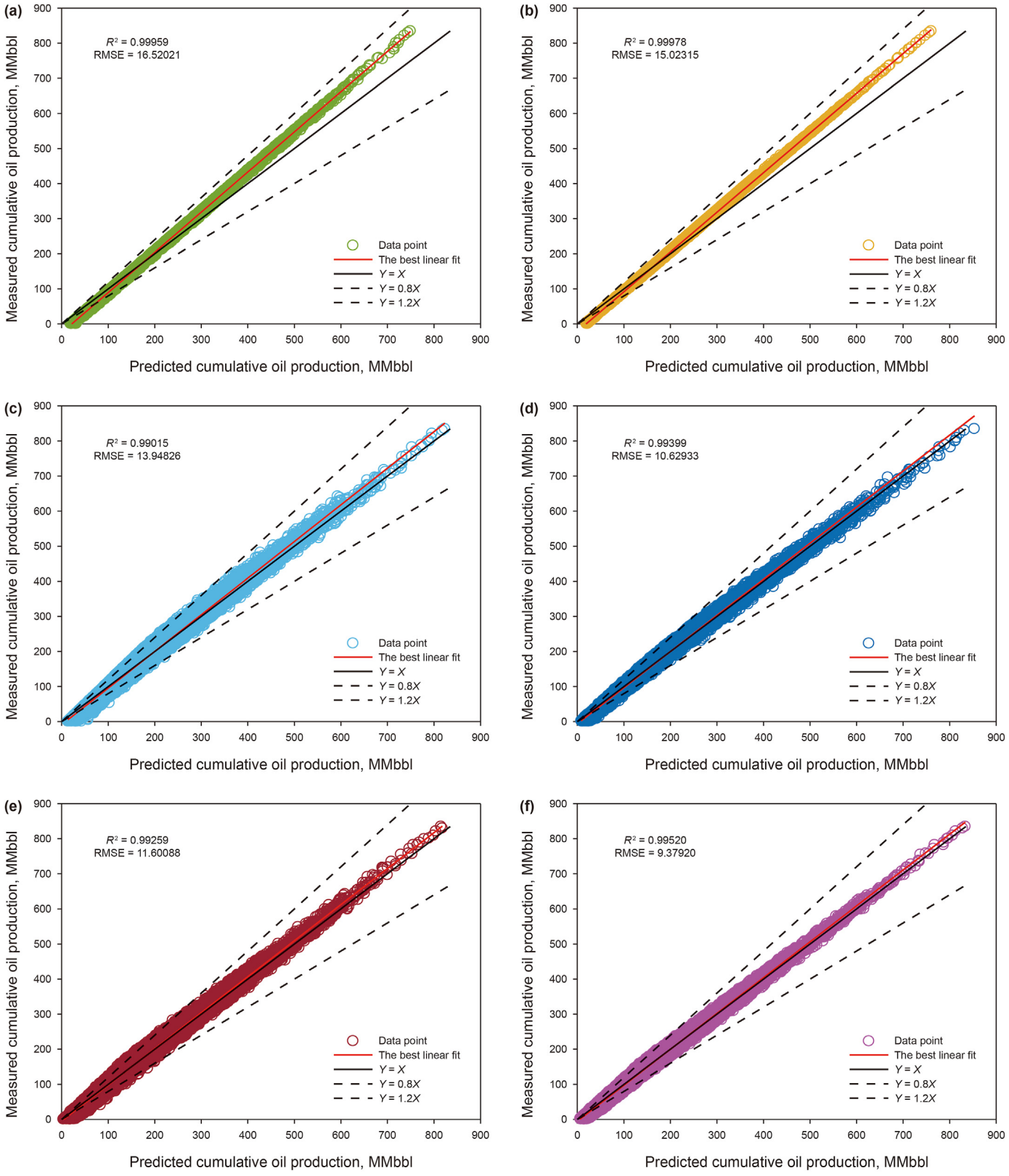

Fig. 17. Cross plots illustrating the comparison between simulated and predicted cumulative oil production values utilizing MLPNN (**a**), LSSVM (b), MLP-PSO (**c**), MLP-GWO (**d**), LSSVM-PSO (**e**), and LSSVM-GWO (**f**) models on training subset.

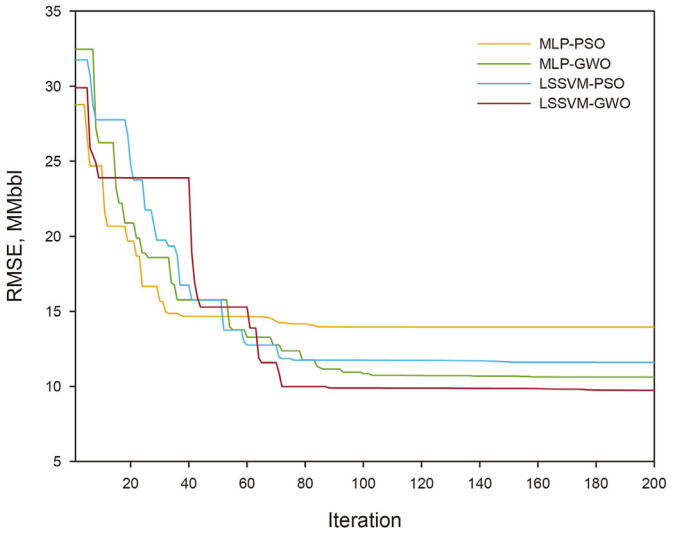

Fig. 18. Variations in RMSE across successive iterations of hybrid model optimizers in the prediction of cumulative oil production for the training subset.

4.2. Prediction "score" analysis for the developed models

ML prediction models sometimes exhibit strong prediction performance according to one criterion while demonstrating a poorer performance according to another, potentially causing confusion. To alleviate this issue, a scoring method combining the results of several performance metrics is applied. For a particular performance metric, the model achieving the most favorable performance relative to other models in that particular criterion is awarded the highest score. Conversely, the model with the poorest performance is assigned the lowest score. The value of the highest score is determined by the total number of models developed, resulting in scores ranging from 1 to 6 in this study. Such "scores" are separately aggregated for the training and testing stages for each performance metric. Additionally, an overall score, which is the sum of the training and testing scores, is also presented.

Table 12 displays the scores assigned to each of the developed models for predicting CO_2 storage mass. The models can be ranked in terms of prediction performance as: LSSVM-GWO (best performance) > LSSVM-PSO > MLP-GWO > MLP-PSO > LSSVM > MLPNN (worst performance). That ranking is clearly visualized in a radar diagram (Fig. 20).

Table 13 displays the scores assigned to each of the developed models for predicting cumulative oil production. The models can be ranked in terms of prediction performance as follows: LSSVM-GWO (best performance) > LSSVM-PSO > MLP-GWO > MLP-PSO > LSSVM > MLPNN (worst performance). That ranking is easily visualized in a radar diagram (Fig. 21).

4.3. Feature impact analysis for the LSSVM-GWO model

Analyzing the impact of each input feature on the outcome of the model generating the highest prediction performance (LSSVM-GWO) can reveal which parameters exert stronger and weaker influences on the model's solution. The Shapley additive explanation (SHAP) method was employed to ascertain the significance of each input feature on the LSSVM-GWO predictive model's outputs. SHAP values are assigned based on the impact of each data record of each input feature on a model's overall prediction performance. SHAP values are presented in terms of the positive and negative aspects of a feature on the predictions derived from partial-dependence plots. The individual SHAP values (shown by the color scale: red high SAP value, blue low SHAP value) for each data record provide detailed insight into how a feature influences the model's prediction output (horizontal scale; negative numbers mean a negative impact on the model output; positive numbers mean a positive impact on model output). The SHAP summary plot displays mean absolute SHAP values and indicates the overall relative importance of each input feature.

Fig. 22 displays the SHAP data-record detail and summary plots for the LSSVM-GWO model applied to predict the CO_2 storage mass dataset. The "Area" input feature is revealed as the most influential; higher "Area" SHAP values (red in Fig. 22(a)) positively contribute to CO_2 storage mass prediction, whereas lower "Area" values (blue in Fig. 22(a)) have a negative impact. The second-most influential feature is porosity ("Por"); Lower Por values (blue in Fig. 22(a); displaying an elongated tail) negatively influence predictions, whereas higher Por SHAP values (red in Fig. 22(a)) exhibit a positive influence. The permeability feature ("Perm") shows a similar shape to Por in its SHAP individual distribution but with less extreme values. On the other hand, the CO_2 injection rate ("InjRate") shows a more symmetrical distribution of SHAP values, which are more evenly spaced on either side of the zero-model influence point; higher InjRate values contribute more positively to model predictions. For BHP and Sorg, high SHAP values contribute negatively to predictions, whereas the thickness parameter has minimal influence on predictions. The summary SHAP diagram (absolute mean values; Fig. 22(b)) ranks the features as Area (most influential) > Por > BHP > Perm > Sorg > InjRate > Thickness (least influential).

Fig. 23 shows the detailed feature-influence plot and the summary plot for partial dependency SHAP analysis of the LSSVM-GWO model applied to the cumulative oil production dataset. The SHAP observations from the SHAP detailed feature-influence plot (Fig. 23(a)) reveal distinctive patterns for each feature. The Por and Prem features exhibit a consistent trend, with low values contributing negatively and high values positively to predictions. The Area predictor shows a linear relationship, where higher values positively impact predictions, and lower values have a negative effect on the prediction model's output. The BHP feature displays a strong

Table 9
Comparison of prediction performance for ML and hybrid ML models applied to the cumulative oil production training subset.

Type	Model	ARE	RMSE, MMbbl	RRMSE	R^2	PI	a20-index
Simple	MLP	1.0143	16.5202	0.0985	0.9996	0.0493	0.7808
	LSSVM	0.8568	15.0232	0.0896	0.9998	0.0448	0.8042
Hybrid	MLP-PSO	0.7835	13.9483	0.0832	0.9902	0.0417	0.8261
	MLP-GWO	0.7040	10.6293	0.0634	0.9940	0.0318	0.8492
	LSSVM-PSO	0.6010	11.6009	0.0692	0.9926	0.0347	0.8625
	LSSVM-GWO	0.5252	9.73920	0.0559	0.9952	0.0280	0.8710

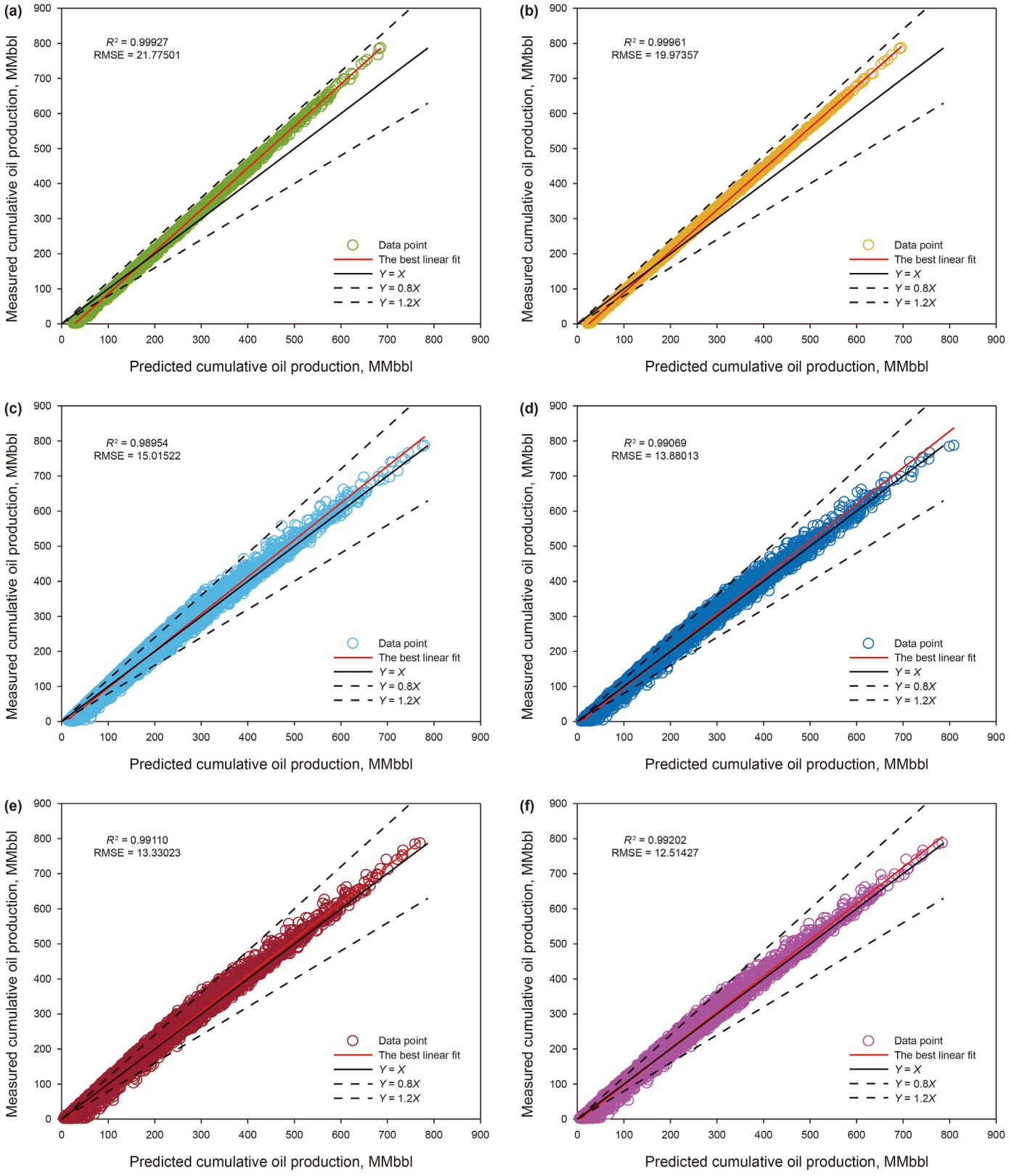

Fig. 19. Cross plots illustrating the comparison between simulated and predicted cumulative oil production values utilizing MLPNN (**a**), LSSVM (**b**), MLP-PSO (**c**), MLP-GWO (**d**), LSSVM-PSO (**e**), and LSSVM-GWO (**f**) models on testing data.

Table 10
Comparison of prediction performance for ML and hybrid ML models applied to the cumulative oil production testing subset.

Type	Model	ARE	RMSE, MMbbl	RRMSE	R^2	PI	a20-index
Simple	MLP	14.7337	21.7750	0.1258	0.9993	0.0629	0.7482
	LSSVM	11.6632	19.9736	0.1154	0.9996	0.0577	0.7758
Hybrid	MLP-PSO	11.2356	15.0152	0.0867	0.9895	0.0435	0.8333
	MLP-GWO	4.9752	13.8801	0.0802	0.9907	0.0402	0.8349
	LSSVM-PSO	6.1576	13.3302	0.0770	0.9911	0.0386	0.8697
	LSSVM-GWO	5.4651	12.5143	0.0723	0.9920	0.0362	0.8640

Table 11
Calculated OFI values for developed predictive models applied to the CO_2 storage mass and cumulative oil production datasets.

Target parameter	MLP	LSSVM	MLP-PSO	MLP-GWO	LSSVM-PSO	LSSVM-GWO
CO_2 storage	0.0457	0.0347	0.0215	0.0201	0.0187	0.0138
Cumulative oil production	0.0578	0.0528	0.0428	0.0370	0.0371	0.0331

Table 12
Model scores of developed models for individual criteria and cumulative scores in training, testing, and overall phases for CO_2 storage mass prediction.

Model	Subset	ARE	RMSE	RRMSE	R^2	PI	a20-index	Score	Total score
MLPNN	Train	1	1	1	5	1	1	10	19
	Test	1	1	1	4	1	1	9	
LSSVM	Train	2	2	2	6	2	2	16	32
	Test	2	2	2	6	2	2	16	
MLP-PSO	Train	3	3	3	2	3	3	17	33
	Test	3	3	3	1	3	3	16	
MLP-GWO	Train	4	5	5	4	5	4	27	49
	Test	4	4	4	2	4	4	22	
LSSVM-PSO	Train	5	4	4	3	4	5	25	54
	Test	6	5	5	3	5	5	29	
LSSVM-GWO*	Train	6	6	6	5	6	6	35	69
	Test	5	6	6	5	6	6	34	

Note: * Represents the best-performing model.

Fig. 20. Radar chart contrasting prediction performance scores for CO_2 storage mass prediction between training and testing subsets and the overall score achieved by ML and hybrid models.

positive correlation between higher values and favorable predictions. Most high SHAP values for the InjRate feature contribute positively to the predictions generated by the models, but some high values have negative impacts on model performance. The Thickness feature values also result in mixed impact, with some low and high values influencing predictions positively, and others negatively. The Sorg feature has impacts on model output close to zero, indicating a limited impact on cumulative oil production predictions. These findings provide nuanced insights into the complex relationships between each input feature and model predictions, contributing to a more comprehensive understanding of the model's behavior. According to the SHAP summary plot (Fig. 23(b)) analysis for the best cumulative oil production predictive model, it is established that the Por feature exerts the most significant influence on the model output, while the Sorg feature has the least impact on the model's predictions. The SHAP model ranks the features as Por (most influential) > Area > Perm > BHP > Thickness > InjRate > Sorg (least influential). Note that the ranking order of input variables is quite different from that associated with the CO_2 storage mass model (Fig. 23).

Table 13
Model scores of developed models for individual criteria and cumulative scores in training, testing, and overall phases for cumulative oil production prediction.

Model	Subset	ARE	RMSE	RRMSE	R^2	PI	a20-index	Score	Total score
MLPNN	Train	1	1	1	5	1	1	10	20
	Test	1	1	1	5	1	1	10	
LSSVM	Train	2	2	2	6	2	2	16	32
	Test	2	2	2	6	2	2	16	
MLP-PSO	Train	3	3	3	1	3	3	16	32
	Test	3	3	3	1	3	3	16	
MLP-GWO	Train	4	5	5	3	5	4	26	50
	Test	6	4	4	2	4	4	24	
LSSVM-PSO	Train	5	4	4	2	4	5	24	52
	Test	4	5	5	3	5	6	28	
LSSVM-GWO*	Train	6	6	6	4	6	6	34	66
	Test	5	6	6	4	6	5	32	

Note: * Represents the best-performing model.

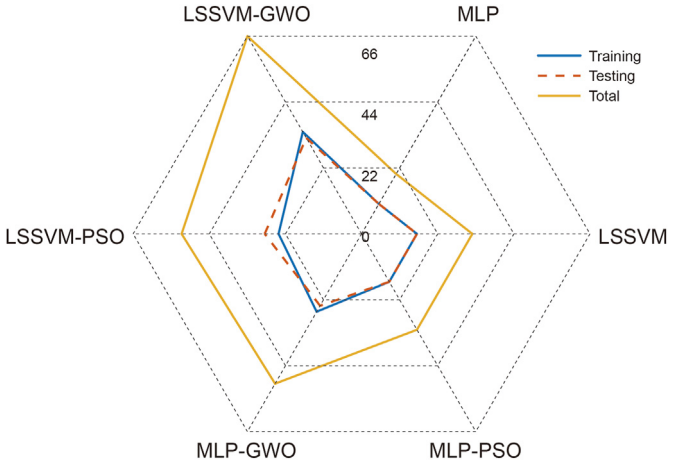

Fig. 21. Radar chart contrasting prediction performance scores for CO_2 storage prediction between training and testing subsets and the overall score achieved by ML and hybrid models.

5. Conclusions

The prediction of CO_2 storage mass and cumulative oil production for CCS-EOR simulations in unconventional reservoirs was performed using two machine learning algorithms (MLPNN and LSSVM) and four hybrid ML models (MLP-PSO, MLP-GWO, LSSVM-PSO, and LSSVM-GWO). A large dataset (>32,000 records) was split into training (~70%) and testing (~30%) groups, with normalization applied to both based on the training data. Outliers were removed using Mahalanobis distance applied to the training set. NSGA-II-LSSVM was used to select the most influential features from nine input parameters: depth, porosity (Por), permeability (Perm), thickness, bottom-hole pressure (BHP), area, CO_2 injection rate (InjRate), residual oil saturation to gas flooding (Sorg), and residual oil saturation to water flooding (Sorw). Predictive models were developed and tested, and performance was evaluated using statistical metrics, together with an overfitting index, scoring, partial dependence, and SHAP analysis. The study's results led to the following conclusions.

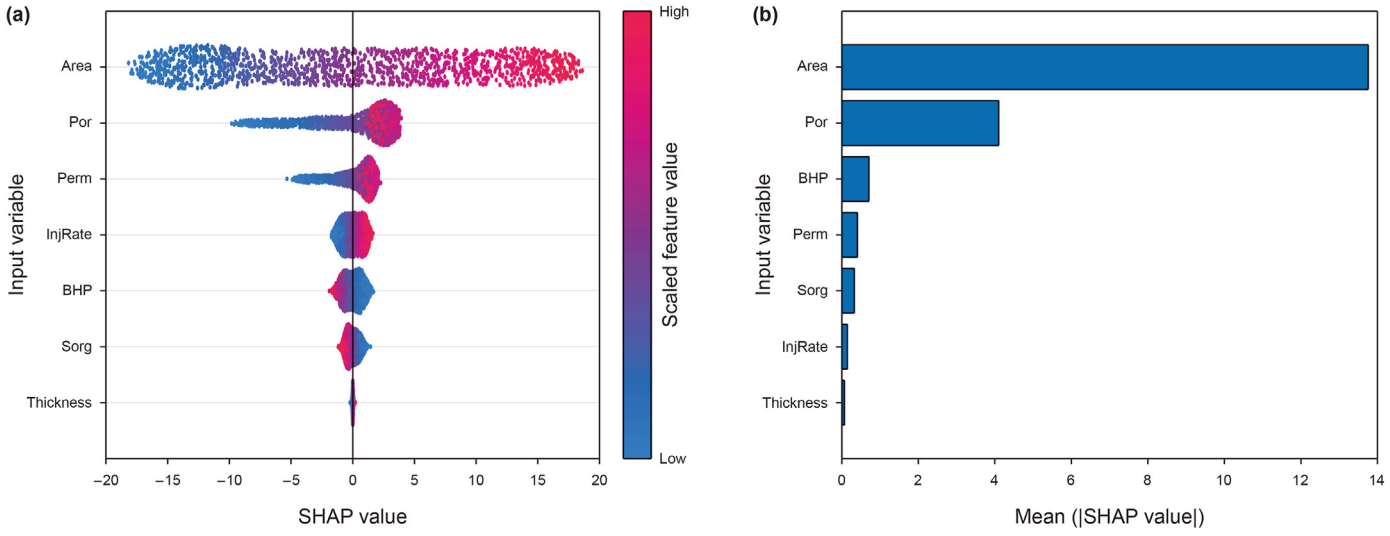

Fig. 22. Visualizing the influence of each input feature on CO_2 storage mass predictions with SHAP values in the LSSVM-GWO model: SHAP detailed feature impact plot (**a**) and SHAP summary plot (**b**) of feature importance.

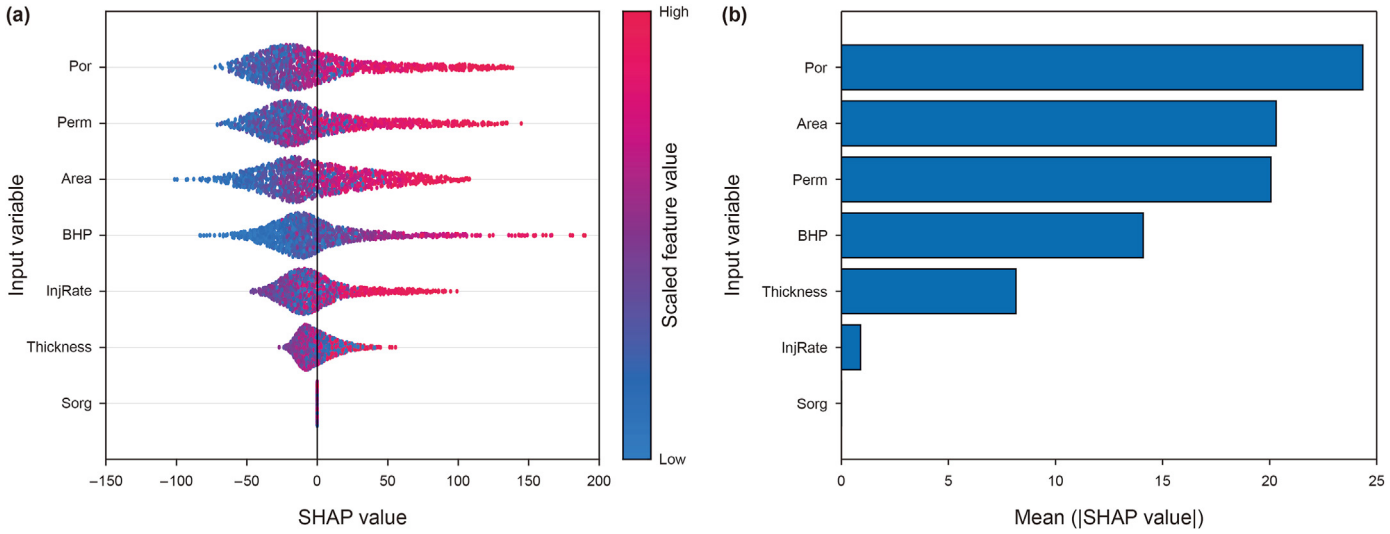

Fig. 23. Visualizing the influence of each feature on cumulative oil production predictions with SHAP values in the LSSVM-GWO model: SHAP detailed feature impact plot (**a**) and SHAP summary plot (**b**) of feature importance.

- Assessments with various training-to-testing data ratios (70:30, 80:20, and 90:10) showed that the 70:30 ratio yielded the most generalizable outcomes.
- Outlier identification revealed mean outlier counts of 1613 for the CO_2 storage mass and 1219 for the cumulative oil production datasets.
- Integrating NSGA-II with LSSVM for feature selection demonstrated that using seven features (BHP, Thickness, Perm, Por, Sorg, InjRate, and Area) was sufficient to generate low prediction errors, leading to the omission of depth and Sorw from the models.
- Trial-and-error selection of the suitable kernel function for LSSVM revealed that the RBF kernel function achieved the lowest prediction errors for CO_2 storage mass and cumulative oil production.
- Sensitivity analysis of the MLPNN architecture, training algorithm, and activation functions revealed that using three hidden layers with 7, 5, and 5 neurons, the Levenberg-Marquardt training algorithm and Gaussian activation function yielded the best prediction performance.
- LSSVM-GWO achieved the lowest RMSE (0.4052 MMT for CO_2 storage, 9.7392 MMbbl for oil production) and highest R^2 (0.9987 for CO_2 storage, 0.9952 for oil production) among all models in the training phase.
- LSSVM-GWO also demonstrated low RMSE with the testing data (0.6224 MMT for CO_2 storage, 12.5143 MMbbl for oil production), indicating high generalizability.
- Metaheuristic optimization algorithms outperform Levenberg–Marquardt and grid search in achieving global optimality for the MLPNN and LSSVM models. The grey wolf optimization (GWO) algorithm performs better than the particle swarm optimization (PSO) algorithm.
- A unified scoring method ranked the models for CO_2 storage mass and cumulative oil production as follows: LSSVM-GWO (best) > LSSVM-PSO > MLP-GWO > MLP-PSO > LSSVM > MLPNN (worst).
- Over-fitting index (OFI) analysis ranked the models by overfitting as follows: LSSVM-GWO (minimal overfitting) < LSSVM-PSO < MLP-GWO < MLP-PSO < LSSVM < MLPNN (maximal overfitting). This confirms that LSSVM-GWO is the most generalizable model.
- SHAP analysis revealed the "Area" and "Por" input features as most influential for CO_2 storage mass and oil production predictions, respectively. On the other hand the "Thickness" and "Sorg" were identified as the least impactful for those two models, respectively.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

CRediT authorship contribution statement

Shadfar Davoodi: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Hung Vo Thanh:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **David A. Wood:** Writing – review & editing, Writing – original draft, Validation, Formal analysis, Conceptualization. **Mohammad Mehrad:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Sergey V. Muravyov:** Writing – review & editing, Writing – original draft, Visualization, Software, Conceptualization. **Valeriy S. Rukavishnikov:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Code for calculating the Mahalanobis distance using the GPR model.

MATLAB code of the outlier detection technique applied

```
Mahalanobis distance outlier detection developed MATLAB code.
% Load data file
FullFileName = fullfile(FilePath, FileName)
[Data, Label, ~] = xlsread(FullFileName)
X = Data(:,1:end-1)
Y = Data(:,end)
% Train Data
nFeature = size(X,2)
nSample = numel(Y)
IDXsh = randperm(nSample)
trRatio = 0.7;
NotrSample = round(trRatio × nSample);
trIDX = IDXsh(1:NotrSample);
tsIDX = IDXsh(NotrSample+1:end);
Xtr = X(trIDX,);
Ytr = Y(trIDX,);
Xts = X(tsIDX,);
Yts = Y(tsIDX,);
% Train GPR model (model = rpg)
Mdl = fitrgp(Xtr, Ytr,"FitMethod","exact","Standardize",true,"KernelFunction","ardexponential");
% Predict response using GPR model
y_pred = predict(Mdl, Xtr);
% Calculate residuals
residuals = Ytr - y_pred;
% Calculate Mahalanobis distance
M = cov(residuals);
invM = inv(M);
D = zeros(size(residuals, 1), 1);
for i = 1:size(residuals, 1)
  D(i) = sqrt(residuals(i,) * invM × residuals(i,)');
end
% Set threshold for outlier detection
threshold = 3 × std(D)
% Identify outliers
outliers = find(D > threshold);
```

Appendix B. Code for feature selection using LSSVM-NSGA-II

MATLAB code of the feature selection technique applied

```
LSSVM-NSGA-II feature selection technique developed MATLAB code.
clc;
clear;
close all;
%% Problem Definition
data = LoadData;
CostFunction = @(s) FeatureSelectionCost(s,data); % Cost Function
nVar = data.nx;   % Number of Decision Variables
VarSize = [1 nVar];   % Size of Decision Variables Matrix
% Number of Objective Functions
nObj = numel(CostFunction(randi([0 1],VarSize)));
% The two goals for this study are: decreasing the number of features for the prediction and minimizing the error of prediction for that number of selected features in the
  first goal.
%% NSGA-II Parameters
MaxIt = 30;   % Maximum Number of Iterations
nPop = 50;   % Population Size
pCrossover = 0.7;   % Crossover Percentage
nCrossover = 2 × round(pCrossover × nPop/2); % Number of Parnets (Offsprings)
pMutation = 0.2;   % Mutation Percentage
```

(continued on next page)

(*continued*)

MATLAB code of the feature selection technique applied

```
nMutation = round(pMutation × nPop); % Number of Mutants
mu = 0.1;   % Mutation Rate
%% Initialization
empty_individual.Position = [];
empty_individual.Cost = [];
empty_individual.Out = [];
empty_individual.Rank = [];
empty_individual.DominationSet = [];
empty_individual.DominatedCount = [];
empty_individual.CrowdingDistance = [];
pop = repmat(empty_individual, nPop,1);
for i = 1:nPop
  if i~ = 1
    pop(i).Position = randi([0 1],VarSize);
  else
    pop(i).Position = ones(VarSize);
  end
  [pop(i).Cost, pop(i).Out] = CostFunction(pop(i).Position);
end
% Non-Dominated Sorting
[pop, F] = NonDominatedSorting(pop);
% Calculate Crowding Distance
pop = CalcCrowdingDistance(pop,F);
% Sort Population
[pop, F] = SortPopulation(pop);
%% NSGA-II Main Loop
for it = 1:MaxIt
  % Crossover
  popc = repmat(empty_individual, nCrossover/2,2);
  for k = 1:nCrossover/2
    i1 = randi([1 nPop]);
    p1 = pop(i1);
    i2 = randi([1 nPop]);
    p2 = pop(i2);
    [popc(k,1).Position, popc(k,2).Position] = Crossover(p1.Position,p2.Position);
    [popc(k,1).Cost, popc(k,1).Out] = CostFunction(popc(k,1).Position);
    [popc(k,2).Cost, popc(k,2).Out] = CostFunction(popc(k,2).Position);
  end
  popc = popc;
  % Mutation
  popm = repmat(empty_individual, nMutation,1);
  for k = 1:nMutation
    i = randi([1 nPop]);
    p = pop(i);
    popm(k).Position = Mutate(p.Position,mu);
    [popm(k).Cost, popm(k).Out] = CostFunction(popm(k).Position);
  end
  % Merge
  pop = [pop
    popc
    popm]; %#ok
  % Non-Dominated Sorting
  [pop, F] = NonDominatedSorting(pop);
  % Calculate Crowding Distance
  pop = CalcCrowdingDistance(pop,F);
  % Sort Population
  [pop, F] = SortPopulation(pop); %#ok
  % Truncate
  pop = pop(1:nPop);
  % Non-Dominated Sorting
  [pop, F] = NonDominatedSorting(pop);
  % Calculate Crowding Distance
  pop = CalcCrowdingDistance(pop,F);
  % Sort Population
  [pop, F] = SortPopulation(pop);
  % Store F1
  F1 = pop(F{1});
  F1=GetUniqueMembers(F1);
  % Show Iteration Information
  disp(['Iteration ' num2str(it) ': Number of F1 Members = ' num2str(numel(F1))]);
  % Plot F1 Costs
  figure(1);
  PlotCosts(F1);
  pause(0.1);
end
```

(*continued*)

MATLAB code of the feature selection technique applied

```
function [z, out] = FeatureSelectionCost(s,data)
  % Read Data Elements
  x = data.x;
  t = data.t;
  % Selected Features
  S = find(s~ = 0);
  % Number of Selected Features (First goal of optimization that should be minimized)
  nf = numel(S);
  % Ratio of Selected Features
  rf = nf/numel(s);
  % Selecting Features
  xs = x(S,);
  % Weights of Train and Test Errors
  wTrain = 0.4;
  wTest = 1-wTrain;
  % Number of Runs
  nRun = 5;
  EE = zeros(1,nRun);
  for r = 1:nRun
    % Create and Train LSSVM
    results = CreateAndTrainLSSVM(xs,t);
    % Calculate Overall Error
    EE(r) = wTrain × results.TrainData.E + wTest × results.TestData.E;
  end
  EE = sort(EE,'ascend');
  EE = EE(1:3);
  E = mean(EE);   % The second goal of the optimization (error of the prediction) that should be minimized.
  if isinf(E)
E = 30;
  end
  % Calculate Final Cost
  z = [nf
E];   %First element is for number of selected feature (first goal) and the second element is the error of the prediction (second goal) with that number of element or selected
    feature.
  % Set Outputs
  out.S=S;
  out.nf = nf;
  out.rf = rf;
  out.E = E;
  out.z = z;
end
```

References

Abbaszadeh, M., Shariatipour, S., 2018. Investigating the impact of reservoir properties and injection parameters on carbon dioxide dissolution in saline aquifers. Fluids 3, 76. https://doi.org/10.3390/fluids3040076.

Ahmadi, M.A., Kashiwao, T., Rozyn, J., Bahadori, A., 2016. Accurate prediction of properties of carbon dioxide for carbon capture and sequestration operations. Petrol. Sci. Technol. 34, 97–103. https://doi.org/10.1080/10916466.2015.1107847.

Ahmadi, M.A., Zendehboudi, S., James, L.A., 2018. Developing a robust proxy model of CO_2 injection: Coupling Box–Behnken design and a connectionist method. Fuel 215, 904–914. https://doi.org/10.1016/j.fuel.2017.11.030.

Ajayi, T., Gomes, J.S., Bera, A., 2019. A review of CO_2 storage in geological formations emphasizing modeling, monitoring and capacity estimation approaches. Petrol. Sci. 16, 1028–1063. https://doi.org/10.1007/s12182-019-0340-8.

Al-Khdheeawi, E.A., Vialle, S., Barifcani, A., Sarmadivaleh, M., Iglauer, S., 2017. Influence of injection well configuration and rock wettability on CO_2 plume behaviour and CO_2 trapping capacity in heterogeneous reservoirs. J. Nat. Gas Sci. Eng. 43, 190–206. https://doi.org/10.1016/j.jngse.2017.03.016.

Al-mudhafar, W.J., 2019. Polynomial and nonparametric regressions for efficient predictive proxy metamodeling: Application through the CO_2-EOR in shale oil reservoirs. J. Nat. Gas Sci. Eng. 72, 103038. https://doi.org/10.1016/j.jngse.2019.103038.

Al-Shargabi, M., Davoodi, S., Wood, D.A., Rukavishnikov, V.S., Minaev, K.M., 2022. Carbon dioxide applications for enhanced oil recovery assisted by nanoparticles: Recent developments. ACS Omega 7, 9984–9994. https://doi.org/10.1021/acsomega.1c07123.

Al-Mudhafar, W.J., Rao, D.N., Srinivasan, S., Thanh, H.V., Lawe, E.M. Al, 2022. Rapid evaluation and optimization of carbon dioxide-enhanced oil recoveryusing reduced-physics proxy models. Energy Sci. Eng. 10, 4112–4135. https://doi.org/10.1002/ese3.1276.

Al Eidan, A.A., Bachu, S., Melzer, L.S., Lars, E.I., Ackiewicz, M., 2015. Technical challenges in the conversion of CO_2-EOR projects to CO_2 storage projects. In: SPE Asia Pacific Enhanc. Oil Recover Conf. https://doi.org/10.2118/174575-MS.

Alvarado, V., Manrique, E., 2010. Enhanced oil recovery: an update review. Energies 3, 1529–1575. https://doi.org/10.3390/EN3091529.

Alves, D.T.S., Lima, G.B.A., 2021. Establishing an onshore pipeline incident database to support operational risk management in Brazil - Part 2: Bowtie proposition and statistics of failure. Process Saf. Environ. Protect. 155, 80–97. https://doi.org/10.1016/j.psep.2021.09.003.

Amar, N., Hemmati-sarapardeh, A., Varamesh, A., 2019. Predicting solubility of CO_2 in brine by advanced machine learning systems: Application to carbon capture and sequestration. J. CO_2 Util. 33, 83–95. https://doi.org/10.1016/j.jcou.2019.05.009.

Andersen, P.Ø., Nygård, J.I., Kengessova, A., 2022. Prediction of oil recovery factor in stratified reservoirs after immiscible water-alternating gas injection based on PSO-, GSA-, GWO-, and GA-LSSVM. Energies 15, 656. https://doi.org/10.3390/en15020656.

Anemangely, M., Ramezanzadeh, A., Amiri, H., Hoseinpour, S.A., 2019. Machine learning technique for the prediction of shear wave velocity using petrophysical logs. J. Pet. Sci. Eng. 174, 306–327. https://doi.org/10.1016/j.petrol.2018.11.032.

Anemangely, M., Ramezanzadeh, A., Tokhmechi, B., 2017. Shear wave travel time estimation from petrophysical logs using ANFIS-PSO algorithm: A case study from Ab-Teymour Oilfield. J. Nat. Gas Sci. Eng. 38, 373–387.

Anemangely, M., Ramezanzadeh, A., Tokhmechi, B., Molaghab, A., Mohammadian, A., 2018. Drilling rate prediction from petrophysical logs and mud logging data using an optimized multilayer perceptron neural network. J. Geophys. Eng. 15, 1146–1159. https://doi.org/10.1088/1742-2140/aaac5d.

Bahrami, P., James, L.A., 2023. Screening of waterflooding using smart proxy model coupled with deep convolutional neural network. J. Pet. Sci. Eng. 221. https://doi.org/10.1016/j.petrol.2022.111300.

Balch, R., McPherson, B., 2016. Integrating enhanced oil recovery and carbon capture and storage projects: A case study at Farnsworth field, Texas. In: SPE Western Regional Meeting. https://doi.org/10.2118/180408-MS.

Bishop, C.M., Nasrabadi, N.M., 2006. Pattern Recognition and Machine Learning. Springer.

Chen, B., Pawar, R., 2018. Capacity assessment of CO_2 storage and enhanced oil recovery in residual oil zones. In: SPE Annu. Tech. Conf. Exhib.. https://doi.org/10.2118/191604-MS.
Chen, B., Pawar, R.J., 2019a. Characterization of CO_2 storage and enhanced oil recovery in residual oil zones. Energy 183, 291–304. https://doi.org/10.1016/j.energy.2019.06.142.
Chen, B., Pawar, R.J., 2019b. Capacity assessment and co-optimization of CO_2 storage and enhanced oil recovery in residual oil zones. J. Pet. Sci. Eng. 182, 106342. https://doi.org/10.1016/j.petrol.2019.106342.
Chen, B., Reynolds, A.C., 2015. Ensemble-based optimization of the WAG injection process. In: SPE Reservoir Simulation Symposium. https://doi.org/10.1080/10916466.2014.956897.
Chen, J., Gao, M., Cheng, S., Hou, W., Song, M., Liu, X., Liu, Y., Shan, Y., 2020. County-level CO_2 emissions and sequestration in China during 1997–2017. Sci. Data 7. https://doi.org/10.1038/s41597-020-00736-3.
Chen, S., Li, H., Yang, D., Tontiwachwuthikul, P., 2010. Optimal parametric design for water-alternating-gas (WAG) process in a CO_2-miscible flooding reservoir. J. Can. Pet. Technol. 49, 75–82. https://doi.org/10.2118/141650-PA.
CMG, 2019. Manual of Computer Modelling Group's Software. Calgary, Canada.
Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A., 2007. Evolutionary Algorithms for Solving Multi-Objective Problems. Springer. https://doi.org/10.1007/978-0-387-36797-2.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20, 273–297. https://doi.org/10.1007/bf00994018.
Dai, Z., Middleton, R., Viswanathan, H., Fessenden-Rahn, J., Bauman, J., Pawar, R., Lee, S.-Y., McPherson, B., 2014a. An integrated framework for optimizing CO_2 sequestration and enhanced oil recovery. Environ. Sci. Technol. Lett. 1, 49–54. https://doi.org/10.1021/ez4001033.
Dai, Z., Stauffer, P.H., Carey, J.W., Middleton, R.S., Lu, Z., Jacobs, J.F., Hnottavange-Telleen, K., Spangler, L.H., 2014b. Pre-site characterization risk analysis for commercial-scale carbon sequestration. Environ. Sci. Technol. 48, 3908–3915. https://doi.org/10.1021/es405468p.
Dai, Z., Xu, L., Xiao, T., McPherson, B., Zhang, X., Zheng, L., Dong, S., Yang, Z., Soltanian, M.R., Yang, C., Ampomah, W., Jia, W., Yin, S., Xu, T., Bacon, D., Viswanathan, H., 2020. Reactive chemical transport simulations of geologic carbon sequestration: Methods and applications. Earth Sci. Rev. 208, 103265. https://doi.org/10.1016/j.earscirev.2020.103265.
Dang, C.T.Q., Chen, Z., Nguyen, N.T.B., Phung, T.H., 2015. An integrated geology and reservoir engineering approach for modelling of a giant fractured basement reservoir. Int. J. Oil Gas Coal Technol. 10, 39–59. https://doi.org/10.1504/IJOGCT.2015.070043.
Dashti, A., Riasat Harami, H., Rezakazemi, M., Shirazian, S., 2018. Estimating CH_4 and CO_2 solubilities in ionic liquids using computational intelligence approaches. J. Mol. Liq. 271, 661–669. https://doi.org/10.1016/j.molliq.2018.08.150.
Davoodi, S., Vo Thanh, H., Wood, D.A., Mehrad, M., Rukavishnikov, V.S., 2023. Combined machine-learning and optimization models for predicting carbon dioxide trapping indexes in deep geological formations. Appl. Soft Comput. 143, 110408. https://doi.org/10.1016/J.ASOC.2023.110408.
Deb, K., Agrawal, S., Pratab, S., Meyarivan, T., 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. Parallel Probl. Solving from Nat. VI Conf. from Nature 849–858. Paris, 18–20 September.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197. https://doi.org/10.1109/4235.996017.
Du, Z., 2012. Intelligence Computation and Evolutionary Computation: Results of 2012 International Conference of Intelligence Computation and Evolutionary Computation ICEC 2012 Held July 7, 2012. Springer Science & Business Media, Wuhan, China. https://doi.org/10.1007/978-3-642-31656-2.
Duan, Y., Yu, X., 2023. A collaboration-based hybrid GWO-SCA optimizer for engineering optimization problems. Expert Syst. Appl. 213, 119017. https://doi.org/10.1016/j.eswa.2022.119017.
Gandomi, A.H., Alavi, A.H., Mousavi, M., Tabatabaei, S.M., 2011. A hybrid computational approach to derive new ground-motion prediction equations. Eng. Appl. Artif. Intell. 24, 717–732. https://doi.org/10.1016/j.engappai.2011.01.005.
Gandomi, A.H., Roke, D.A., 2015. Assessment of artificial neural network and genetic programming as predictive tools. Adv. Eng. Software 88, 63–72. https://doi.org/10.1016/j.advengsoft.2015.05.007.
Garrett, A., 2008. Neural Enhancement for Multiobjective Optimization. Ph.D. Dissertation. Auburn University.
Gibson-Poole, C.M., Svendsen, L., Underschul, J.T., Ennis-King, J., Ruth, P.J. van, Nelson, E.J., Daniel, R.F., Cinar, Y., 2006. Gippsland basin geosequestration: A potential solution for the latrobe valley brown coal CO_2 emissions. APPEA J 46, 413–434. https://doi.org/10.1071/AJ05024.
Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Cambridge, MA, USA.
Gurney, K., 2018. An Introduction to Neural Networks. CRC Press, London, UK.
Han, W.S., McPherson, B.J., Lichtner, P.C., Wang, F.P., 2010. Evaluation of trapping mechanisms in geologic CO_2 sequestration: Case study of SACROC northern platform, A 35-year CO_2 injection site. Am. J. Sci. 310, 282–324. https://doi.org/10.2475/04.2010.03.
Haykin, S., 1998. Neural Networks: A Comprehensive Foundation. Prentice Hall PTR.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition.
Hemmati-Sarapardeh, A., Amar, M.N., Soltanian, M.R., Dai, Z., Zhang, X., 2020. Modeling CO_2 solubility in water at high pressure and temperature conditions. Energy Fuels. 34, 4761–4776. https://doi.org/10.1021/acs.energyfuels.0c00114.
Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507. https://doi.org/10.1126/science.1127647.
Knowles, J., Corne, D., Deb, K., 2007. Multiobjective Problem Solving from Nature: From Concepts to Applications. Springer Science & Business Media. https://doi.org/10.1007/978-3-540-72964-8.
Le Van, S., Chon, B.H., 2017. Evaluating the critical performances of a CO_2–Enhanced oil recovery process using artificial neural network models. J. Pet. Sci. Eng. 157, 207–222. https://doi.org/10.1016/j.petrol.2017.07.034.
LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539.
Lee, J.H., Park, Y.C., Sung, W.M., Lee, Y.S., 2010. A simulation of a trap mechanism for the sequestration of CO_2 into Gorae V Aquifer, Korea. Energy Sources, Part A Recover. Util. Environ. Eff. 32, 796–808. https://doi.org/10.1080/15567030903436822.
Lee, K.J., 2020. Data-driven models to predict hydrocarbon production from unconventional reservoirs by thermal recovery. J. Energy Resour. Technol. Trans. 142, 1–17. https://doi.org/10.1115/1.4047309.
Li, S., Zhang, Y., 2014. Model complexity in carbon sequestration: A design of experiment and response surface uncertainty analysis. Int. J. Greenh. Gas Control 22, 123–138. https://doi.org/10.1016/j.ijggc.2013.12.007.
Liberty, L.M., Yelton, J., Skurtveit, E., Braathen, A., Midtkandal, I., Evans, J.P., 2022. Regolith and host rock influences on CO_2 leakage: Active source seismic profiling across the Little Grand Wash fault, Utah. Int. J. Greenh. Gas Control 119, 103742. https://doi.org/10.1016/j.ijggc.2022.103742.
Lin, B., Tan, Z., 2021. How much impact will low oil price and carbon trading mechanism have on the value of carbon capture utilization and storage (CCUS) project? Analysis based on real option method. J. Clean. Prod. 298, 126768. https://doi.org/10.1016/j.jclepro.2021.126768.
Lipponen, J., Burnard, K., Beck, B., Gale, J., Pegler, B., 2011. The IEA CCS technology roadmap: One year on. Energy Proc. 4, 5752–5761. https://doi.org/10.1016/j.egypro.2011.02.571.
Liu, B., Zhang, Y., 2011. CO_2 modeling in a deep saline aquifer: A predictive uncertainty analysis using design of experiment. Environ. Sci. Technol. 45, 3504–3510. https://doi.org/10.1021/es103187b.
Liu, T., Wu, P., Chen, Z., Li, Y., 2022. Review on carbon dioxide replacement of natural gas hydrate: Research progress and perspectives. Energy Fuels. 36, 7321–7336. https://doi.org/10.1021/acs.energyfuels.2c01292.
Lumley, D., 2010. 4D seismic monitoring of CO_2 sequestration. Lead. Edge 29, 150–155. https://doi.org/10.1190/1.3304817.
Ma, W., Jafarpour, B., Qin, J., 2019. Dynamic characterization of geologic CO_2 storage aquifers from monitoring data with ensemble Kalman filter. Int. J. Greenh. Gas Control 81, 199–215. https://doi.org/10.1016/j.ijggc.2018.10.009.
Mehrad, M., Bajolvand, M., Ramezanzadeh, A., Neycharan, J.G., 2020. Developing a new rigorous drilling rate prediction model using a machine learning technique. J. Pet. Sci. Eng. 192, 107338. https://doi.org/10.1016/J.PETROL.2020.107338.
Mudhafar, W.J. Al, Rao, D.N., Srinivasan, S., 2019. Geological and production uncertainty assessments of the cyclic CO_2-assisted gravity drainage EOR process: A case study from South Rumaila oil field. J. Pet. Explor. Prod. Technol. 9, 1457–1474. https://doi.org/10.1007/s13202-018-0542-4.
Naghizadeh, A., Larestani, A., Nait Amar, M., Hemmati-Sarapardeh, A., 2022. Predicting viscosity of CO_2–N_2 gaseous mixtures using advanced intelligent schemes. J. Pet. Sci. Eng. 208, 109359. https://doi.org/10.1016/j.petrol.2021.109359.
Nait Amar, M., Jahanbani Ghahfarokhi, A., Ng, C.S.W., Zeraibi, N., 2021. Optimization of WAG in real geological field using rigorous soft computing techniques and nature-inspired algorithms. J. Pet. Sci. Eng. 206, 109038. https://doi.org/10.1016/j.petrol.2021.109038.
Nait Amar, M., Jahanbani Ghahfarokhi, A., Zeraibi, N., 2020. Predicting thermal conductivity of carbon dioxide using group of data-driven models. J. Taiwan Inst. Chem. Eng. 113, 165–177. https://doi.org/10.1016/J.JTICE.2020.08.001.
Ng, C.S.W., Jahanbani Ghahfarokhi, A., 2022. Adaptive proxy-based robust production optimization with multilayer perceptron. Appl. Comput. Geosci. 16, 100103. https://doi.org/10.1016/j.acags.2022.100103.
Osman, H., Ghafari, M., Nierstrasz, O., 2018. The impact of feature selection on predicting the number of bugs. arXiv Prepr. arXiv1807.04486. https://doi.org/10.48550/arXiv.1807.04486.
Ren, B., Duncan, I.J., 2019. Reservoir simulation of carbon storage associated with CO_2 EOR in residual oil zones, San Andres formation of West Texas, Permian Basin, USA. Energy 167, 391–401. https://doi.org/10.1016/j.energy.2018.11.007.
Ren, B., Ren, S., Zhang, L., Chen, G., Zhang, H., 2016. Monitoring on CO_2 migration in a tight oil reservoir during CCS-EOR in Jilin Oilfield, China. Energy 98, 108–121. https://doi.org/10.1016/j.energy.2016.01.028.
Ruprecht, C., Pini, R., Falta, R., Benson, S., Murdoch, L., 2014. Hysteretic trapping and relative permeability of CO_2 in sandstone at reservoir conditions. Int. J. Greenh. Gas Control 27, 15–27. https://doi.org/10.1016/j.ijggc.2014.05.003.
Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural Network. 61, 85–117. https://doi.org/10.1016/j.neunet.2014.09.003.
Schölkopf, B., Smola, A.J., 2002. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and beyond. MIT Press. https://doi.org/10.7551/mitpress/4175.001.0001.
Seyed Mostapha Kalami Heris, 2024. Non-dominated Sorting Genetic Algorithm II (NSGA-II). Retrieved September 21, 2024. MATLAB Central File Exchange.

https://www.mathworks.com/matlabcentral/fileexchange/52869-non-dominated-sorting-genetic-algorithm-ii-nsga-ii.
Shahkarami, A., Mohaghegh, S., 2020. Applications of smart proxies for subsurface modeling. Petrol. Explor. Dev. 47, 400–412. https://doi.org/10.1016/S1876-3804(20)60057-X.
Shawe-Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Analysis. Cambridge University Press. https://doi.org/10.1017/CBO9780511809682.
Smola, A.J., Schölkopf, B., 2004. A tutorial on support vector regression. Stat. Comput. 14, 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.
Souza, R.G.L.D., Sekaran, K.C., Kandasamy, A., 2010. Improved NSGA-II based on a novel ranking scheme. https://doi.org/10.48550/arXiv.1002.4005.
Srinivas, N., Deb, K., 1994. Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol. Comput. 2, 221–248. https://doi.org/10.1162/evco.1994.2.3.221.
Stein, M., 1987. Large sample properties of simulations using Latin hypercube sampling. Technometrics 29, 143–151. https://doi.org/10.1080/00401706.1987.10488205.
Subramanian, R., Subramanian, K., Subramanian, B., 2009. Application of a fast and elitist multi-objective genetic algorithm to Reactive Power Dispatch. Serbian J. Electr. Eng. 6, 119–133. https://doi.org/10.2298/SJEE0901119S.
Susanto, V., Sasaki, K., Sugai, Y., Kawasaki, W., 2016. Field test study on leakage monitoring at a geological CO_2 storage site using hydrogen as a tracer. Int. J. Greenh. Gas Control 50, 37–48. https://doi.org/10.1016/j.ijggc.2016.04.001.
Suykens, J.A.K., Vandewalle, J., 1999. Least squares support vector machine classifiers. Neural Process. Lett. 9, 293–300. https://doi.org/10.1023/A:1018628609742.
Thanh, H.V., Zamanyad, A., Safaei-Farouji, M., Ashraf, U., Hemeng, Z., 2022. Application of hybrid artificial intelligent models to predict deliverability of underground natural gas storage sites. Renew. Energy. https://doi.org/10.1016/J.RENENE.2022.09.132.
Trentham, R.C., Melzer, L.S., Melzer, L.S., Koperna, G., 2015. Case studies of the ROZ CO_2 flood and the combined ROZ/MPZ CO_2 flood at the Goldsmith Landreth Unit, Ector County, Texas. Using “next generation” CO_2 EOR technologies to optimize the residual oil zone CO_2 flood. https://doi.org/10.2172/1224947.
Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer. https://doi.org/10.1007/978-1-4757-3264-1.
Van Si, L., Chon, B.H., 2018. Effective prediction and management of a CO_2 flooding process for enhancing oil recovery using artificial neural networks. J. Energy Resour. Technol. Trans. 140, 1–14. https://doi.org/10.1115/1.4038054.
Vo Thanh, H., Sugai, Y., Nguele, R., Sasaki, K., 2019. Robust optimization of CO_2 sequestration through a water alternating gas process under geological uncertainties. Appl. Energy 103208. https://doi.org/10.1016/j.jngse.2020.103208.
Vo Thanh, H., Sugai, Y., Sasaki, K., 2020. Application of artificial neural network for predicting the performance of CO_2 enhanced oil recovery and storage in residual oil zones. Sci. Rep. 10, 18204. https://doi.org/10.1038/s41598-020-73931-2.
Vo Thanh, H., Yasin, Q., Al-mudhafar, W.J., Lee, K., 2022. Knowledge-based machine learning techniques for accurate prediction of CO_2 storage performance in underground saline aquifers. Appl. Energy 314. https://doi.org/10.1016/j.apenergy.2022.118985.
Wang, J., Zhang, Y., Xie, J., 2020. Influencing factors and application prospects of CO_2 flooding in heterogeneous glutenite reservoirs. Sci. Rep. 10 (1), 1839. https://doi.org/10.1038/s41598-020-58792-z.
Wang, M., Hui, G., Pang, Y., Wang, S., Chen, S., 2023. Optimization of machine learning approaches for shale gas production forecast. Geoenergy Sci. Eng. 226, 211719. https://doi.org/10.1016/j.geoen.2023.211719.
Wilday, J., Wardman, M., Johnson, M., Haines, M., 2011. Hazards from carbon dioxide capture, transport and storage. Process Saf. Environ. Protect. 89, 482–491. https://doi.org/10.1016/j.psep.2011.09.002.
Xu, R., Zeng, K., Zhang, C., Jiang, P., 2017. Assessing the feasibility and CO_2 storage capacity of CO_2 enhanced shale gas recovery using triple-porosity reservoir model. Appl. Therm. Eng. 115, 1306–1314. https://doi.org/10.1016/j.applthermaleng.2017.01.062.
Yan, H., Zhang, J., Rahman, S.S., Zhou, N., Suo, Y., 2020. Predicting permeability changes with injecting CO_2 in coal seams during CO_2 geological sequestration: A comparative study among six SVM-based hybrid models. Sci. Total Environ. 705. https://doi.org/10.1016/j.scitotenv.2019.135941.
Yao, P., Yu, Z., Zhang, Y., Xu, T., 2023. Application of machine learning in carbon capture and storage: An in-depth insight from the perspective of geoscience. Fuel 333, 126296. https://doi.org/10.1016/J.FUEL.2022.126296.
You, J., Ampomah, W., Sun, Q., 2020. Development and application of a machine learning based multi-objective optimization workflow for CO_2-EOR projects. Fuel 264, 116758. https://doi.org/10.1016/j.fuel.2019.116758.
Zhang, H., Thanh, H.V., Rahimi, M., Al-Mudhafar, W.J., Tangparitkul, S., Zhang, T., Dai, Z., Ashraf, U., 2023. Improving predictions of shale wettability using advanced machine learning techniques and nature-inspired methods: Implications for carbon capture utilization and storage. Sci. Total Environ. 877, 162944. https://doi.org/10.1016/j.scitotenv.2023.162944.
Zhang, K., Lau, H.C., 2022. Regional opportunities for CO_2 capture and storage in Southeast Asia. Int. J. Greenh. Gas Control 116, 103628. https://doi.org/10.1016/j.ijggc.2022.103628.
Zhao, H., Huang, G., Yan, N., 2018. Forecasting energy-related CO_2 emissions employing a novel SSA-LSSVM model: Considering structural factors in China. Energies 11. https://doi.org/10.3390/en11040781.