Journal of Petroleum Science and Technology 14(4): 44, 2024, Pages 11-21
DOI:10.22078/jpst.2025.5709.1979

Journal of Petroleum Science and Technology
Research Paper
https://jpst.ripi.ir/

Comprehensive Assessment of Supervised Machine Learning Models for Prediction of Oil Recovery Factor and NPV in Surfactant-Polymer Flooding: Bayesian Optimization and Stacking Ensembles

Kasra Ekhtiyaran Haghighi, Novin Nekuee, Maryam Ghorbani-Bavariani* and Erfan Zarei
Department of Petroleum and Geo-Energy Engineering, Amirkabir University of Technology (AUT), Tehran, Iran

Abstract
Surfactant-polymer (SP) flooding is recognized as an effective chemical enhanced oil recovery (EOR) method, where accurate prediction of oil recovery factor (RF) and net present value (NPV) is vital for field development planning and economic analysis. This study systematically evaluates a range of supervised machine learning algorithms—including CatBoost, artificial neural networks (ANN), XGBoost, LightGBM, and gradient boosting regressor (GBR)—for forecasting RF and NPV based on experimental SP flooding data. Baseline model results were established using default hyperparameters, followed by comprehensive two-stage hyperparameter tuning using grid search and Bayesian optimization with Optuna, along with five-fold cross-validation to ensure robustness. CatBoost and ANN consistently achieved the highest predictive accuracy. In addition, ensemble stacking was then performed by combining top-performing models, further enhancing prediction reliability and generalization. Additional post-processing using quantile adjustment (linear residual correction) addressed residual bias and improved calibration between predicted and observed values. Furthermore, model performance was benchmarked using standard statistical metrics and comparative graphical analysis. Also, the results demonstrate that integrating well-established supervised learning methods with systematic optimization, stacking, and output calibration offers a robust and practical framework for accurate prediction of SP flooding outcomes. Moreover, this approach provides valuable support for data-driven decision-making in EOR project design and evaluation. Furthermore, the proposed framework achieved strong predictive accuracy in the all-stacking ensemble with cross-validation, yielding an R^2 of 0.978 and AAPRE of 2.71 for recovery factor, and an R^2 of 0.944 and AAPRE of 6.18 for net present value. Ultimately, then applying quantile adjustment to the all-stacking ensemble, the performance remained competitive, with an R^2 of 0.964 and AAPRE of 3.61 for recovery factor, and an R^2 of 0.924 and AAPRE of 7.94 for net present value, further demonstrating the robustness of the approach.

Keywords: Surfactant-polymer Flooding; Oil Recovery Factor; Net Present Value; Supervised Machine Learning; Ensemble Stacking; Bayesian Optimization; Quantile Adjustment.

Introduction
Increasing oil recovery from reservoirs remains a strategic and technically challenging task in petroleum engineering, especially as a significant fraction of hydrocarbon reserves persists as residual oil after conventional waterflooding [1]. To address this, advanced Enhanced Oil Recovery (EOR) techniques have been developed, with surfactant–polymer (SP) flooding recognized as a highly promising approach due to its dual capability of lowering interfacial tension and increasing the viscosity of the displacing phase, thereby mobilizing trapped oil and improving recovery efficiency.

However, designing and optimizing SP flooding processes is challenging due to reservoir heterogeneity, fluid-rock interactions, and operational complexities. Although numerical simulators like UTCHEM are widely used for analysis, they are less practical for preliminary studies because of high computational costs, parameter sensitivity, and large data requirements [2]. Consequently, data-driven modeling and the adoption of machine learning (ML) and artificial intelligence (AI) algorithms have gained significant momentum in petroleum engineering research and practice [3,4].

In recent years, numerous studies have explored ML-based prediction of key SP flooding metrics, such as recovery factor (RF) and net present value (NPV). For instance, Karambeigi et al. (2011) pioneered the use of multilayer perceptron (MLP) neural networks trained on UTCHEM simulation data, achieving mean absolute

**Corresponding author:* Maryam Ghorbani-Bavariani, Department of Petroleum and Geo-Energy Engineering, Amirkabir University of Technology (AUT), Tehran, Iran
E-mail addresses: m_ghorbany@aut.ac.ir
Received 2025-06-10, Received in revised form 2025-09-13, Accepted 2025-09-21, Available online 2025-11-04

relative errors of 2.12% for RF and 4.6% for NPV, thus validating ANN models as efficient surrogates for numerical simulation and scenario optimization [3]. Subsequently, Kamari et al. (2016) introduced LSSVM models optimized with Coupled Simulated Annealing (CSA), further reducing absolute prediction errors to 1.9% for RF and 3.1% for NPV and demonstrating via sensitivity analysis that surfactant concentration and slug size were the most influential variables—though increased surfactant concentration could negatively affect NPV due to chemical costs [5].

Advanced architectures such as cascade neural networks and gradient boosting (GBDT), as demonstrated by Larestani et al. (2022), have achieved even lower errors (0.66% for RF and 1.95% for NPV), showcasing the potential of ensemble and deep learning approaches to significantly reduce prediction errors, albeit at the expense of increased data and computational requirements [4]. In another important contribution, Hou et al. (2009) combined genetic algorithms with SVM to predict oil production and water-cut curves with less than 3% error, even for field cases with limited data, affirming the robustness of such hybrid approaches [6].

Methodologically, most of these studies have relied on complex or hybrid models, prioritizing accuracy but raising questions regarding their practical implementation under computational or data constraints. In 2013, Al-Dousari and Garrouch provided a different perspective by defining 18 dimensionless input groups and training a three-hidden-layer ANN to predict oil recovery with less than 3% error, emphasizing the value of rapid reservoir screening and preliminary assessment [1]. Moreover, complementary research by Zerpa et al. in 2005 and Dang et al. in 2018 explored field-scale numerical optimization via surrogate models, balancing accuracy with operational feasibility in the presence of geological uncertainty [2,7].

Nonetheless, critical challenges such as optimal model structure selection, input variable identification, limited sample sizes, overfitting, and sensitivity to outliers continue to complicate the adoption of ML models in practical EOR scenarios. Furthermore, while high-accuracy results are often achieved by leveraging complex architectures or ensemble approaches (e.g., stacking), a systematic evaluation of whether simpler ML models, when enhanced with advanced validation techniques (such as cross-validation and stacking), can approach the accuracy of their complex counterparts remains largely unaddressed.

This study aims to fill this gap by evaluating the performance of ten widely-used simple ML models for predicting RF and NPV, using high-quality numerical laboratory data for SP flooding. Furthermore, by employing advanced validation techniques, including stacking and cross-validation, this study investigates whether the predictive accuracy of simple models can be enhanced without resorting to deep or hybrid architectures. Moreover, the results contribute both to bridging the literature gap and to offering practical guidelines for ML model selection in EOR projects, particularly under resource limitations.

Material and Methods

This research aims to thoroughly investigate the predictive capabilities of a broad spectrum of supervised machine learning (ML) algorithms for forecasting two critical metrics in surfactant-polymer (SP) flooding: oil recovery factor (RF) and net present value (NPV). In addition, the ultimate goal is to determine whether conventional ML models, optimized with modern hyperparameter tuning strategies and ensemble methods, can match or even surpass the predictive performance of deep learning and hybrid models previously reported in literature.

Data Collection and Preprocessing

This study uses a laboratory dataset from Prasamphanich (2009), later curated by Karambeigi et al. (2011) [4,8]. Both studies are established benchmarks for enhanced oil recovery (EOR) machine learning research. Moreover, this dataset consists of 202 independent SP flooding experiments, where each record includes seven critical input variables capturing essential operational and reservoir conditions—such as polymer drive salinity, polymer and surfactant concentrations, slug sizes, and permeability ratios—selected for their documented relevance and predictive influence on chemical EOR outcomes [4]. In addition, the response variables of interest are the oil recovery factor (RF, %) and the net present value (NPV, $MM), both of which represent key technical and economic metrics in SP flooding performance assessment.

For robust and unbiased model evaluation, the full dataset was randomly stratified into a training set (161 samples, approximately 70%) and a test set (41 samples, 30%). Furthermore, prior to model development, all input features were standardized using the Standard Scaler to ensure uniform variable scaling and stable model convergence. In addition, the summary of the dataset's key features, their definitions, and statistical properties is provided in the relevant table within the Data section (see Table 1) and was constructed to closely follow the conventions established in the referenced benchmark studies [8,4].

Supervised Machine Learning Models

A comprehensive panel of ten supervised learning models was selected:

Decision Tree (DT)
Random Forest (RF)
Extra Trees (ET)
AdaBoost
Gradient Boosting Regressor (GBR)
XGBoost
LightGBM
CatBoost
Support Vector Regression (SVR)
Artificial Neural Network (ANN, MLP-based)

The rationale for this diverse selection is rooted in their proven effectiveness for capturing complex non-linearities in EOR modeling [3,9]. Each algorithm brings unique strengths in terms of interpretability, handling of feature interactions, and regularization, offering a holistic landscape of classical ML performance.

CatBoost

CatBoost, short for Categorical Boosting, is a high-performance gradient boosting library developed by Yandex that is particularly adept at modeling tabular data. What sets CatBoost apart is its innovative use of ordered boosting and its native handling of categorical variables.

Table 1 Full description of the gathered dataset.

Parameters	Unit	mean	std	min	25%	50%	75%	max
Surfactant slug size	PV	0.177228	0.072253	0.097	0.097	0.178	0.259	0.259
Surfactant concentration	Vol. fraction	0.017748	0.011192	0.005	0.005	0.0175	0.03	0.03
polymer concentration in surfactant slug	wt %	0.176629	0.066828	0.1	0.1	0.175	0.25	0.25
polymer drive size	PV	0.481748	0.143979	0.324	0.324	0.478	0.648	0.648
polymer concentration in polymer drive	wt %	0.148158	0.044381	0.1	0.1	0.147	0.2	0.2
K_v/K_h ratio	-	0.128515	0.107642	0.01	0.01	0.13	0.25	0.25
Salinity of polymer drive	Meq/ml	0.348599	0.044729	0.3	0.3	0.349	0.4	0.4
Recovery factor (RF)	%	39.66802	9.263624	14.82	33.9225	41.865	46.635	56.99
Net present value (NPV)	$ MM	4.451225	1.533972	1.065	3.37235	4.38985	5.55675	8.1017

Unlike many other boosting algorithms, CatBoost applies an ordered boosting scheme that reduces prediction shift and target leakage, contributing to better generalization, especially on smaller datasets. Additionally, CatBoost's implementation of symmetric (oblivious) trees and its advanced regularization mechanisms make it highly resistant to overfitting. These architectural choices allow the model to efficiently process both numerical and categorical inputs without the need for extensive preprocessing or encoding. In the present study, CatBoost yielded the highest individual predictive accuracy for both RF and NPV, confirming its reputation as one of the most effective machine learning tools for complex regression tasks [10].

LightGBM

LightGBM (Light Gradient Boosting Machine) is a tree-based boosting framework designed for speed and efficiency, developed by Microsoft. Moreover, its standout features include Gradient-based One-Side Sampling (GOSS), which accelerates training by focusing on samples with the largest gradients, and Exclusive Feature Bundling (EFB), which reduces feature dimensionality by bundling mutually exclusive features. Furthermore, unlike many boosting algorithms that grow trees level-wise, LightGBM uses a leaf-wise strategy, selecting the leaf with the largest loss reduction at each split. This approach can significantly improve model accuracy, particularly on large and complex datasets. However, it also increases the risk of overfitting if not properly regularized. In this research, LightGBM demonstrated outstanding computational performance while consistently ranking among the top models in terms of predictive power for both target variables [11].

Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs), inspired by the biological structure of the human brain, are powerful universal function approximators. In this study, an ANN was implemented in the form of a multi-layer perceptron (MLP), which consists of one input layer, one or more hidden layers, and an output layer. Each neuron applies a weighted linear transformation followed by a non-linear activation function, enabling the network to capture intricate relationships between input features and targets. In addition, the learning process is governed by the backpropagation algorithm, which iteratively adjusts the network's weights to minimize the prediction error. Also, architecture and training hyperparameters—including the number of hidden layers, neurons per layer, learning rate, and batch size—were meticulously tuned using Optuna's Bayesian optimization framework. Ultimately, this optimization ensured that the ANN provided highly competitive accuracy without sacrificing generalization.

Gradient Boosting Regressor (GBR)

The Gradient Boosting Regressor (GBR) is an ensemble learning method that sequentially builds decision trees, with each new tree designed to correct the errors of the existing ensemble. By optimizing a chosen loss function (such as least squares for regression), GBR incrementally improves predictive accuracy with each added tree. Moreover, the model's effectiveness hinges on key hyperparameters like learning rate, number of estimators, and subsampling rate. In this study, GBR was carefully optimized and validated through cross-validation, resulting in strong and stable Unlike many other boosting algorithms, CatBoost applies an ordered boosting scheme that reduces prediction shift and target leakage, contributing to better generalization, especially on smaller datasets. Additionally, CatBoost's implementation of symmetric (oblivious) trees and its advanced regularization mechanisms make it highly resistant to overfitting. These architectural choices allow the model to efficiently process both numerical and categorical inputs without the need for extensive preprocessing or encoding. In the present study, CatBoost yielded the highest individual predictive accuracy for both RF and NPV, confirming its reputation as one of the most effective machine learning tools for complex regression tasks [10].

LightGBM

LightGBM (Light Gradient Boosting Machine) is a tree-based boosting framework designed for speed and efficiency, developed by Microsoft. Moreover, its standout features include Gradient-based One-Side Sampling (GOSS), which accelerates training by focusing on samples with the largest gradients, and Exclusive Feature Bundling (EFB), which reduces feature dimensionality by bundling mutually exclusive features. Furthermore, unlike many boosting algorithms that grow trees level-wise, LightGBM uses a leaf-wise strategy, selecting the leaf with the largest loss reduction at each split. This approach can significantly improve model accuracy, particularly on large and complex datasets. However, it also increases the risk of overfitting if not properly regularized. In this research, LightGBM demonstrated outstanding computational performance while consistently ranking among the top models in terms of predictive power for both target variables [11].

Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs), inspired by the biological structure of the human brain, are powerful universal function approximators. In this study, an ANN was implemented in the form of a multi-layer perceptron (MLP), which consists of one input layer, one or more hidden layers, and an output layer. Each neuron applies a weighted linear transformation followed by a non-linear activation function, enabling the network to capture intricate relationships between input features and targets. In addition, the learning process is governed by the backpropagation algorithm, which iteratively adjusts the network's weights to minimize the prediction error. Also, architecture and training hyperparameters—including the number of hidden layers, neurons per layer, learning rate, and batch size—were meticulously tuned using Optuna's Bayesian optimization framework. Ultimately, this optimization ensured that the ANN provided highly competitive accuracy without sacrificing generalization.

Gradient Boosting Regressor (GBR)

The Gradient Boosting Regressor (GBR) is an ensemble learning method that sequentially builds decision trees, with each new tree designed to correct the errors of the existing ensemble. By optimizing a chosen loss function (such as least squares for regression), GBR incrementally improves predictive accuracy with each added tree. Moreover, the model's effectiveness hinges on key hyperparameters like learning rate, number of estimators, and subsampling rate. In this study, GBR was carefully optimized and validated through cross-validation, resulting in strong and stable predictive performance. Its capacity to focus learning on the most challenging data points contributed substantially to its competitive results for both RF and NPV.

XGBoost

XGBoost (Extreme Gradient Boosting) is an advanced gradient boosting algorithm that has gained widespread acclaim for its computational efficiency and predictive power. Moreover, its unique features include built-in L1 and L2 regularization to control overfitting, an optimized approach for handling missing values and sparse data, and the ability to perform parallelized tree construction. Also, XGBoost incorporates native cross-validation tools that streamline model selection and evaluation. These technical enhancements, combined with robust hyperparameter tuning in this study, enabled XGBoost to deliver excellent accuracy and reliability across both target variables, solidifying its position as a leading solution in structured regression tasks [12].

Hyperparameter Optimization: Grid Search and Bayesian Optimization with Optuna

Recognizing that model performance in ML is highly sensitive to hyperparameter selection, this study employed a two-stage optimization strategy:

1. Grid Search: Initially, a wide grid of plausible hyperparameter values was systematically explored for each model to identify promising regions in the parameter space. Grid search guarantees exhaustive coverage but is computationally expensive, especially as the number of parameters increases.
2. Bayesian Optimization with Optuna: Building upon grid search, a more efficient search was performed using the Optuna framework, which leverages Bayesian optimization via Tree-structured Parzen Estimator (TPE) samplers. Moreover, unlike grid search, Bayesian optimization uses information from previous trials to intelligently select subsequent hyperparameter configurations, dramatically improving efficiency and solution quality. The objective function for optimization was set to maximize the R^2 score, evaluated through cross-validation.

Optuna's approach is well-aligned with best practices for hyperparameter tuning in advanced ML pipelines. Its integrated pruning mechanisms also allowed early termination of non-promising trials, further reducing computation time. Furthermore, all hyperparameter tuning was performed exclusively on the training set, using a five-fold cross-validation scheme.

The core principle behind Bayesian optimization in Optuna is the use of an acquisition function to guide the search for optimal hyperparameters. Specifically, Optuna employs the Expected Improvement (EI) criterion, which aims to maximize the expected gain over the current best objective value [13]. Mathematically, EI is defined as:

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} (y^* - y) P(y \mid x) dy \quad (1)$$

Cross-Validation and Model Evaluation

Five-fold cross-validation (KFold=5) was central to both hyperparameter optimization and model evaluation. In each fold, the training set was partitioned into five subsets; four subsets were used for model fitting, and one was reserved for validation. Moreover, this process was repeated such that each subset served as the validation set exactly once. The cross-validation process minimizes the risk of overfitting and provides a reliable estimate of the model's generalization ability across different data splits. The final test set (41 samples), which remained unseen during training and tuning, was used exclusively for final model performance assessment. Evaluation metrics included:

1. R^2 score: proportion of variance explained by the model,
2. Mean Absolute Error (MAE): average absolute difference between predictions and actual values,
3. Root Mean Squared Error (RMSE): sensitive to larger errors, reflecting prediction reliability.

This rigorous scheme ensures comparability with recent advanced EOR modeling studies and provides robust statistical confidence in reported results [13,14].

Ensemble Stacking for Enhanced Predictive Accuracy

To further push the limits of predictive accuracy, an ensemble stacking approach was employed. Stacking involves training several diverse base models (level-1 learners) and using their predictions as features for a meta-model (level-2 learner). According to Pavlyshenko (2018), stacking enables the integration of different model architectures, leveraging their complementary strengths and minimizing individual model weaknesses, which often leads to superior predictive performance compared to any single constituent model [15]. In this study, the best-performing models from the individual optimization phase (Extra Trees, ANN, CatBoost, GBR, XGBoost) were selected as base learners. Out-of-fold predictions from these base models (generated during cross-validation to avoid data leakage) were used to train a simple

meta-learner, typically a linear regressor or another robust ML algorithm. This approach has demonstrated success in various domains, including time-series forecasting and regression, as supported by Pavlyshenko's work and winning Kaggle competition solutions [15].

Quantile Adjustment for Output Calibration

As a final calibration step, quantile adjustment was applied to some models to align the predicted distributions with those observed in the target data, following methods similar to Yin et al. in 2021. Quantile adjustment is especially valuable when systematic bias is detected in certain prediction ranges, as it statistically maps predicted quantiles to empirical quantiles from training data, thereby correcting output skewness and improving reliability.

In this study, quantile adjustment was implemented via a linear residual correction approach. Specifically, model residuals were defined as:

$$r = y - \hat{y} \tag{2}$$

where y is the true target value and ŷ is the model prediction. The residuals were then modeled as a linear function of the model outputs:

$$r = \alpha + \beta\hat{y} \tag{3}$$

where α and β are regression coefficients obtained via least squares on the test set. The model prediction was then adjusted as:

$$\hat{y}_{adj} = (1+\beta)\hat{y} + \alpha \tag{4}$$

This adjustment helps compensate for linear trends in the prediction errors, yielding a more accurate and unbiased forecast of the target variables [9].

Results and Discussion

Model Performance Evaluation

In this section, a comprehensive evaluation of all supervised machine learning models applied in this study is presented. Moreover, the analysis covers three major stages: baseline performance prior to any hyperparameter optimization, the improvements achieved following systematic Bayesian tuning, and a cross-validation assessment to ensure model robustness and generalizability. Furthermore, model performance is quantified using the coefficient of determination (R^2), mean absolute error (MAE), and root mean squared error (RMSE), computed on both training and independent test sets.

Baseline Model Results (Pre-Optimization)

Initially, each machine learning model was trained using default hyperparameter values provided by standard Python libraries. The primary objective at this stage was to establish a baseline for subsequent comparisons. Performance metrics for both oil recovery factor (RF) and net present value (NPV) were calculated on the training and test sets (see Table 2). Fig.s 1 and 2 show that ensemble methods (e.g., CatBoost, ANN, LightGBM) achieved strong predictive accuracy even without hyperparameter tuning. In contrast, simpler models such as Decision Tree and SVR had higher errors and overfitting. This initial assessment highlights the inherent advantages of ensemble approaches in handling complex, nonlinear relationships present in the SP flooding dataset.

Optimized Model Results (Post-Bayesian Tuning)

Following the baseline assessment, all models underwent rigorous hyperparameter optimization using Optuna's Bayesian framework with five-fold cross-validation. The optimal values for the key hyperparameters of each model are provided (see Table 3). The optimized models were then evaluated on the independent test set, and their predictive performance metrics—including R^2, MAE, RMSE, AAPRE, and APRE for both RF and NPV—are presented in Table 4.

Moreover, this tuning process led to noticeable improvements in predictive accuracy across nearly all algorithms, as evidenced by higher R^2 values and lower error metrics on the test set.

Cross-Validation Analysis

To ensure the robustness and reliability of the optimized model predictions, a five-fold cross-validation analysis was conducted on the training data after hyperparameter tuning. The cross-validation results for R^2, MAE, RMSE, AAPRE, and APRE across all folds for each optimized model are summarized in Table 4. Moreover, for the top-performing models (e.g., CatBoost, ANN, XGBoost), the observed low variance and consistently high R^2 values confirm strong model stability and generalization capacity. Furthermore, these findings reinforce the credibility of the optimized models for practical application in SP flooding performance prediction.

Comparative Analysis of Machine Learning Algorithms

In this section, the final predictive performance of all machine learning models is comprehensively compared. Performance metrics including R^2, MAE, RMSE, AAPRE, and APRE—after Bayesian hyperparameter optimization and cross-validation—form the basis for model ranking and analysis.

As shown in Table 4, CatBoost and the artificial neural network (ANN) achieved the highest R^2 values and the lowest error metrics, emerging as the most accurate models for predicting both the oil recovery factor (RF) and net present value (NPV). In contrast, models such as SVR and Decision Tree exhibited relatively weaker performance and greater sensitivity to data and parameter variations.

Additionally, the minimal differences between training and test metrics for the top-performing models indicate strong generalizability and robustness.

Ensemble Stacking Performance

This section evaluates the effectiveness of ensemble stacking for enhancing predictive accuracy. To this end, the four top-performing models identified in the previous analysis (CatBoost, ANN, LGBM and GBR) were combined in pairwise stacking ensembles, with a suitable meta-learner (such as linear regression) to aggregate their predictions. After implementing the stacking models, an additional five-fold cross-validation was performed on each stacking ensemble to further assess the impact of stacking alone versus stacking combined with cross-validation.

Table 2 Baseline results of machine learning models (pre-optimization): R^2, MAE, RMSE, AAPRE, and APRE for RF and NPV (training and test sets).

Metric	RandomForest	ExtraTrees	DT	XGB	CatBoost	SVR	ANN	LGBM	GBR	AdaBoost
Train										
R2	0.976	1.000	0.9123	1.000	1.000	0.980	0.963	0.974	0.991	0.888
MSE	0.547	0.000	0.9599	0.000	0.000	0.709	1.268	0.406	0.094	2.917
MAE	0.491	0.000	0.9196	0.004	0.013	0.570	0.711	0.468	0.231	1.206
APRE	-0.99	-0.00	-2.41	-0.00	-0.00	-0.76	-0.34	-0.67	-0.27	-3.13
AAPRE	3.99	0.00	7.91	0.03	0.07	3.66	4.34	4.13	2.20	9.42
Test										
R2	0.864	0.894	0.6845	0.861	0.898	0.881	0.918	0.910	0.908	0.814
MSE	3.858	2.404	11.6790	3.955	3.079	4.617	3.012	2.048	2.325	5.758
MAE	1.332	1.071	2.3536	1.365	1.114	1.410	1.079	0.989	0.969	1.620
APRE	-5.65	-5.35	-5.87	-5.08	-5.43	-4.50	0.38	-3.36	-2.20	-7.02
AAPRE	12.17	10.46	18.28	11.21	10.20	10.52	6.85	8.73	8.33	14.58
RF										
R2	0.916	0.949	0.723	0.914	0.932	0.896	0.933	0.956	0.950	0.873
MSE	7.226	4.391	23.728	7.410	5.802	8.884	5.772	3.741	4.302	10.877
MAE	2.097	1.644	3.983	2.190	1.765	2.349	1.804	1.528	1.503	2.555
APRE	-2.17	-1.69	-2.16	-2.57	-2.81	-2.07	0.39	-1.23	-0.71	-2.58
AAPRE	6.10	4.70	11.09	6.30	5.66	6.95	5.02	4.45	4.24	7.50
NPV										
R2	0.812	0.840	0.482	0.808	0.863	0.866	0.903	0.863	0.866	0.754
MSE	0.489	0.417	1.348	0.500	0.356	0.349	0.252	0.356	0.349	0.640
MAE	0.567	0.498	0.861	0.540	0.462	0.471	0.355	0.450	0.435	0.685
APRE	-9.13	-9.01	-9.57	-7.60	-8.06	-6.93	0.36	-5.49	-3.69	-11.46
AAPRE	18.24	16.23	25.47	16.12	14.73	14.08	8.67	13.01	12.41	21.66

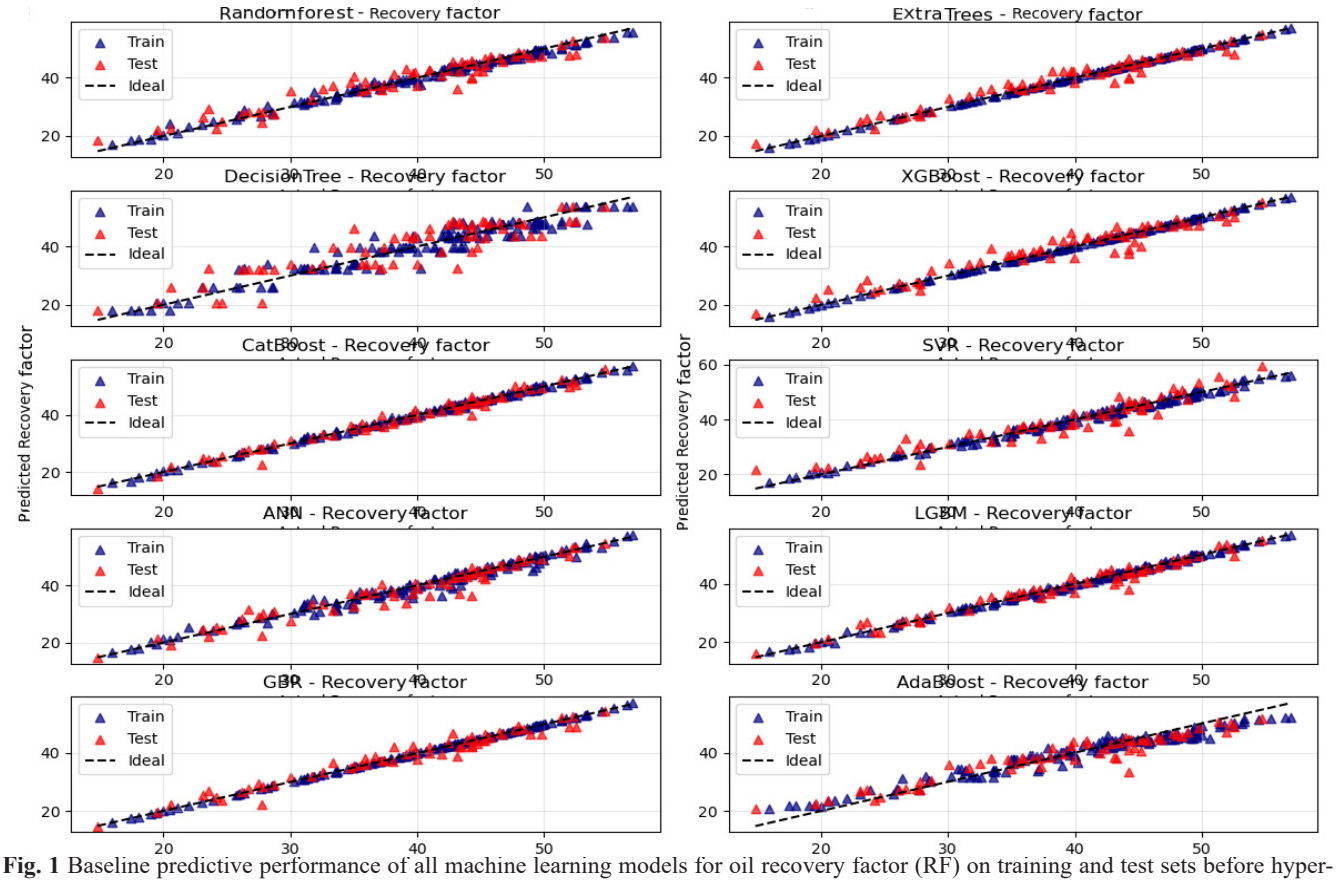

Fig. 1 Baseline predictive performance of all machine learning models for oil recovery factor (RF) on training and test sets before hyper-parameter optimization.

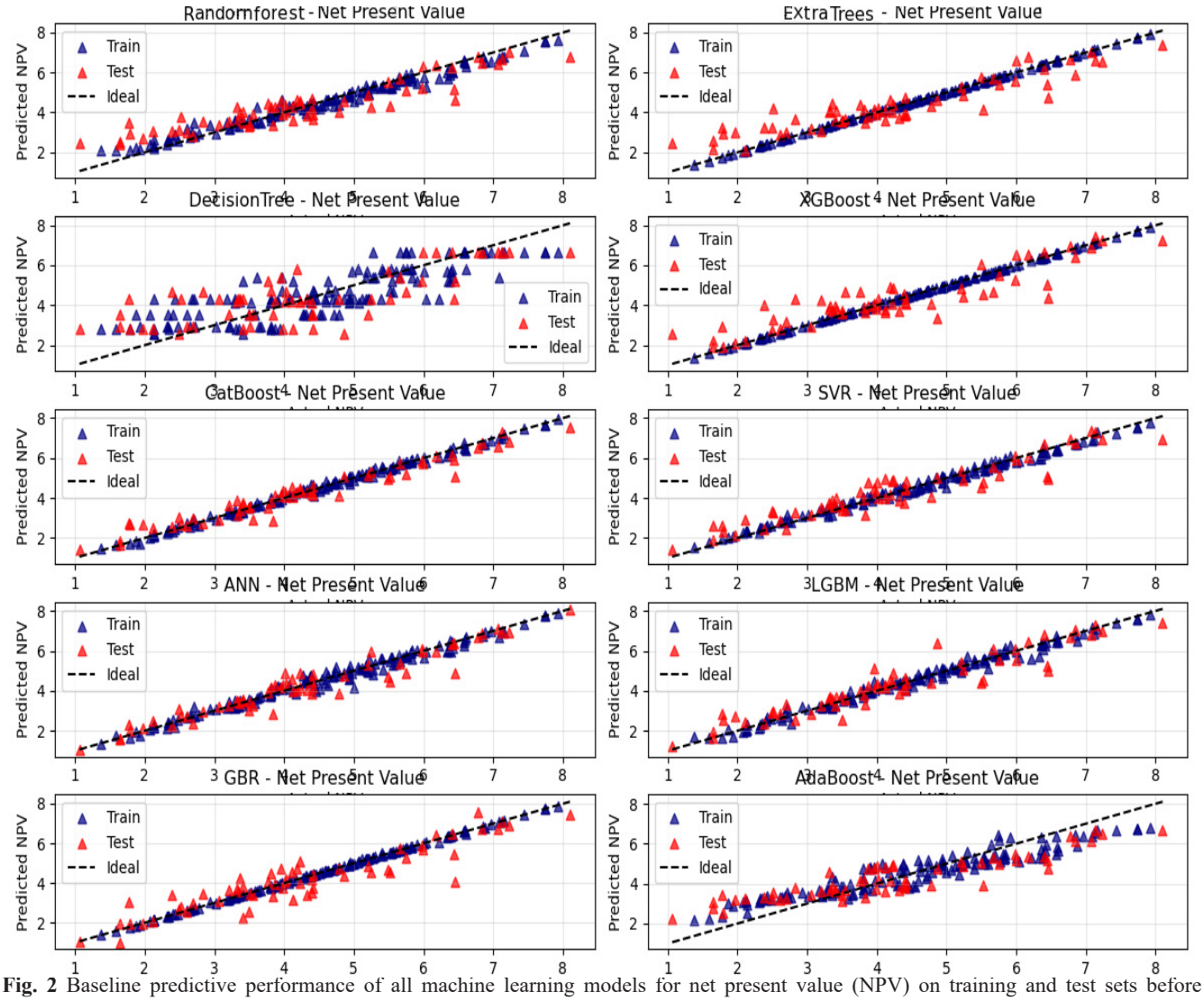

Fig. 2 Baseline predictive performance of all machine learning models for net present value (NPV) on training and test sets before hyperparameter optimization.

Table 3 Summary of hyperparameters demonstrating best performance across ML models.

Model	Best Hyper Parameters
RandomForest	n_estimators: 198, max_depth: 8
ExtraTrees	n_estimators: 172, max_depth: 11
DecisionTree	max_depth: 4, min_samples_split: 8, criterion: friedman_mse
XGBoost	n_estimators: 90, max_depth: 3, learning_rate: 0.1591, reg_alpha: 1.1848, reg_lambda: 0.0929
CatBoost	iterations: 122, depth: 3, learning_rate: 0.1912, loss_function: RMSE
SVR	C: 2.2363, epsilon: 0.1111, kernel: rbf
ANN	hidden_layer_sizes: [100], activation: relu, max_iter: 271
LGBM	n_estimators: 164, max_depth: 9, learning_rate: 0.1675
GBR	n_estimators: 187, max_depth: 3, learning_rate: 0.2749
AdaBoost	n_estimators: 105, learning_rate: 0.887

Table 4 Optimized results of machine learning models (Post-optimization): R^2, MAE, RMSE, AAPRE, and APRE for RF and NPV (training and test sets).

Metric	RF	ET	DT	XGB	CatBoost	SVR	ANN	LGBM	GBR
Recovery Factor									
Tr_MSE	0.959519	4.96E-05	6.996863	0.000112	0.231499	0.826303	2.104708	0.258302	0.00659
Tr_R2	0.988639	0.999999	0.917157	0.999999	0.997259	0.990217	0.97508	0.996942	0.99992
Tr_MAE	0.764282	0.002409	2.045536	0.006557	0.377354	0.852089	1.100355	0.397195	0.04967
Tr_AAPRE	2.122234	0.007401	5.373528	0.018483	1.027051	2.290806	3.013209	1.08943	0.12653
Tr_APRE	-0.42826	-0.00021	-0.49494	-0.00125	-0.03611	-0.27676	-0.26158	-0.04119	-0.0020
Ts_MSE	6.661154	4.720786	19.67938	7.409871	3.172464	8.674852	7.013838	2.798198	3.45313
Ts_R2	0.922292	0.944928	0.770423	0.913557	0.96299	0.8988	0.918177	0.967357	0.95971
Ts_MAE	2.007117	1.672919	3.829754	2.189554	1.31587	2.392541	1.970918	1.264908	1.40730
Ts_AAPRE	5.929168	4.772363	11.03958	6.30349	3.507464	7.043103	5.432111	3.534021	3.87307
Ts_APRE	-2.43524	-1.76594	-3.63289	-2.56592	-0.28622	-2.59688	0.634676	-0.61903	-0.7951
NPV									
Tr_MSE	0.072231	7.15E-06	0.719667	4.13E-06	0.012025	0.028452	0.056607	0.034788	0.0009
Tr_R2	0.96718	0.999997	0.672999	0.999998	0.994536	0.987072	0.974279	0.984193	0.99959
Tr_MAE	0.20942	0.001058	0.700713	0.001371	0.084327	0.154218	0.175957	0.138841	0.0175
Tr_AAPRE	5.70884	0.028515	19.19618	0.035172	2.082433	3.935984	4.085204	3.529672	0.42809
Tr_APRE	-1.64121	-0.00257	-5.40278	-0.00404	-0.09155	-1.25026	-0.31652	-0.245	-0.0042
Ts_MSE	0.475061	0.450577	1.186562	0.500044	0.141198	0.305917	0.217731	0.286349	0.39946
Ts_R2	0.817306	0.826721	0.543683	0.807698	0.945699	0.882353	0.916267	0.889879	0.84637
Ts_MAE	0.554555	0.524728	0.869882	0.539819	0.270294	0.436644	0.332154	0.399516	0.46997
Ts_AAPRE	17.7504	16.85321	27.84419	16.12119	7.415529	12.52612	8.672863	10.94444	12.4287
Ts_APRE	-9.17704	-9.07961	-13.4048	-7.59703	-0.85964	-6.18286	-1.45035	-4.27704	-0.3523
Cross-Validation (Recovery Factor)									
MSE	6.704745	5.12618	19.97534	6.650344	2.228277	6.863062	5.779726	2.903111	2.81547
R2	0.921481	0.939967	0.766069	0.922118	0.973905	0.919627	0.932314	0.966002	0.96702
MAE	2.091215	1.764576	3.668545	1.97311	1.106162	2.024862	1.78269	1.321998	1.28181
AAPRE	5.943228	4.846041	10.02741	5.532362	2.997472	5.700665	5.028768	3.608108	3.49888
APRE	-0.90065	-0.77661	-0.53505	-1.28978	-0.16251	-0.90046	-0.54414	-0.12412	-0.0783
Cross-Validation (NPV)									
MSE	0.419037	0.281055	0.783921	0.386552	0.123259	0.210944	0.169281	0.247078	0.27286
R2	0.821033	0.879964	0.665194	0.834907	0.947357	0.909908	0.927702	0.894475	0.88346
MAE	0.513971	0.411125	0.711339	0.475012	0.253766	0.366315	0.313813	0.361381	0.37921
AAPRE	14.30808	11.48898	19.13909	13.14833	6.288757	9.581899	7.78606	8.935511	9.9784
APRE	-5.06033	-4.01989	-4.74494	-5.03397	-0.69949	-3.32829	-1.46785	-1.78545	-2.8440

In addition, an "all-stacking" structure was also constructed in which all the selected models were combined simultaneously in a single stacking ensemble, in order to evaluate the synergistic effects of utilizing all base learners together. The performance metrics of these different stacking strategies, including the all-stacking approach, are reported in Table 5, and a comparative visualization of their results is provided in Fig. 3.

As shown in the table and corresponding figure (i.e. Fig. 3), stacking generally improved model accuracy compared to individual base learners, and in most cases, stacking combined with cross-validation outperformed stacking alone. Also, the all-stacking ensemble demonstrated favorable performance across most metrics.

These findings demonstrate that ensemble stacking, especially when coupled with cross-validation, and even when utilizing all candidate models, can substantially enhance the prediction quality of SP flooding performance by leveraging the complementary strengths of different algorithms.

Quantile Adjustment/Linear Residual Correction Results

This section examines the effectiveness of Quantile Adjustment (or Linear Residual Correction) in further improving model accuracy. Moreover, it should be noted that this calibration technique was applied exclusively to the outputs of the Stacking and All-Stacking ensembles, as these approaches had already demonstrated the highest predictive performance.

Furthermore, the primary objective of this step was to correct systematic prediction bias and better align the predicted distribution with the actual data, thereby reducing residual error and enhancing the reliability of forecasts.

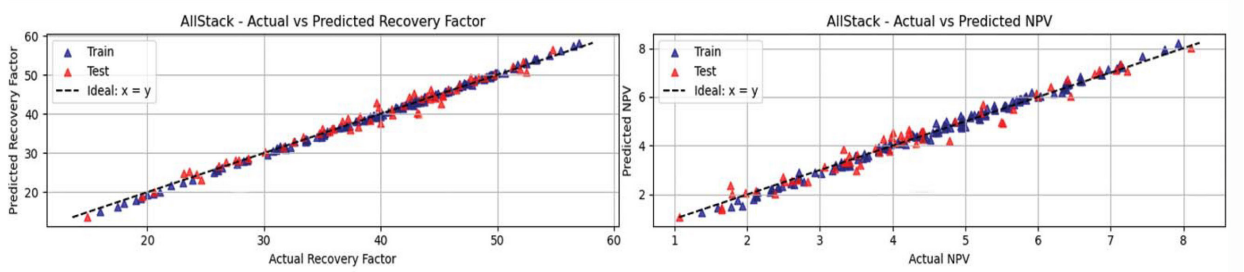

Fig. 3 Predictive results of the all-stacking ensemble for both net present value (NPV) and oil recovery factor (RF), showing the performance of the combined CatBoost, ANN, LGBM, and GBR models. Both metrics are displayed to highlight the accuracyn improvements achieved by ensemble stacking in SP flooding performance prediction.

Table 5 Ensemble Stacking Performance results: R^2, MAE, RMSE, AAPRE, and APRE for RF and NPV (training and test sets).

Metric	CatBoost+ANN	CatBoost+LGBM	CatBoost+GBR	ANN+LGBM	ANN+GBR	LGBM+GBR
Stacking (Recovery Factor)						
Train_MSE	2.698602	2.358054	2.150505	0.993406	0.646507	0.693687
Train_R2	0.968049	0.972081	0.974538	0.988238	0.992345	0.991787
Train_MAE	1.327935	1.227706	1.168376	0.824018	0.65536	0.654523
Train_AAPRE	3.982039	3.673231	3.607506	2.187002	1.783502	1.821771
Train_APRE	1.469305	1.173055	1.392511	-0.1118	0.23244	0.238957
Test_MSE	3.528875	3.587497	3.992103	3.180473	3.514282	3.712499
Test_R2	**0.958833**	**0.958149**	**0.953429**	**0.962897**	**0.959003**	**0.95669**
Test_MAE	1.409456	1.466952	1.493462	1.357024	1.354077	1.442621
Test_AAPRE	3.648426	3.960061	4.084902	3.609701	3.755093	4.023784
Test_APRE	0.354747	-0.28076	0.202905	-0.27081	0.366005	-0.29702
(Stacking NPV)						
Train_MSE	0.100969	0.210884	0.246088	0.077601	0.072955	0.107899
Train_R2	0.954122	0.904179	0.888183	0.96474	0.966851	0.950973
Train_MAE	0.260958	0.378848	0.407302	0.231588	0.225243	0.275994
Train_AAPRE	7.333909	10.53043	11.51121	5.657281	5.620078	6.721875
Train_APRE	2.860209	4.465512	5.253292	0.414441	0.637057	-0.03661
Test_MSE	0.217069	0.23219	0.23404	0.207289	0.210588	0.326632
Test_R2	**0.916522**	**0.910707**	**0.909995**	**0.920283**	**0.919014**	**0.874387**
Test_MAE	0.322208	0.342727	0.337772	0.325549	0.307667	0.442754
Test_AAPRE	8.306425	8.759523	8.733197	8.638763	8.324238	11.61109
Test_APRE	1.63265	-0.60502	-0.38521	1.502644	2.257352	-2.60952
Stacking-CV (Recovery Factor)						
MSE	2.770597	2.655071	2.34387	2.6374	2.17992	2.59067
R2	**0.967554**	**0.968906**	**0.972551**	**0.969113**	**0.974471**	**0.969661**
MAE	1.246256	1.26573	1.224095	1.213963	1.152231	1.247128
AAPRE%	3.346057	3.432858	3.365754	3.229834	3.089444	3.388114
APRE%	0.260527	0.278702	0.351378	0.092874	0.13958	0.175558
Stacking-CV (NPV)						
MSE	0.134411	0.152444	0.144135	0.184973	0.164713	0.305703
R2	**0.942594**	**0.934893**	**0.938441**	**0.921**	**0.929652**	**0.869437**
MAE	0.26624	0.282053	0.279742	0.309045	0.29583	0.394908
AAPRE %	6.534909	7.02267	7.083888	7.257725	7.213983	9.638
APRE %	-0.41025	-0.43053	-0.68643	-0.67652	-0.72305	-1.71956
All-Stack						
Metric		**Stacking-CV (Recovery Factor)**		**Stacking-CV (NPV)**		
MSE		1.8427		0.1200		
R2		**0.9784**		**0.9445**		
MAE		1.0076		0.2595		
AAPRE%		2.71		6.18		
APRE%		0.05		-0.72		

In addition, the results after applying Quantile Adjustment for both Stacking and All-Stacking ensembles are summarized in Table 6. As demonstrated, the application of Quantile Adjustment led to a noticeable improvement in accuracy metrics (notably higher R^2 and lower AAPRE and APRE) for both ensemble approaches. These findings highlight that post-processing correction techniques such as Quantile Adjustment can effectively reduce residual error and substantially boost the trustworthiness of SP flooding performance predictions.

Summary of Key Findings

The comprehensive assessment performed in this study highlights the prominent role of supervised machine learning algorithms—particularly ensemble methods and stacking architectures—in predicting oil recovery factor (RF) and net present value (NPV) in surfactant-polymer flooding. While ensemble models such as CatBoost, ANN, and LightGBM showed strong baseline accuracy, their performance was further enhanced through Bayesian hyperparameter optimization and systematic cross-validation, yielding R^2 values above 0.96 for the leading models. Comparative results indicated that stacking the four top models with cross-validation achieved the highest predictive accuracy for both target variables.

Table 6 Quantile Adjustment / Linear Residual Correction Results on Stacking and all-stack without Cross-Validation.

Metric	CatBoost+ANN	CatBoost+GBR	ANN+GBR
Stacking-Quantile (Recovery Factor)			
MSE_Before	3.528875	3.992103	3.514282
R2_Before	**0.958833**	**0.953429**	**0.959003**
MAE_Before	1.409456	1.493462	1.354077
AAPRE_Before%	3.648426	4.084902	3.755093
APRE_Before%	0.354747	0.202905	0.366005
MSE_After	3.193147	3.591879	3.214084
R2_After	**0.962749**	**0.958098**	**0.962505**
MAE_After	1.364355	1.404825	1.324036
AAPRE_After%	3.799427	3.932009	3.698947
APRE_After%	-0.36852	-0.3205	-0.28069
Stacking-Quantile (NPV)			
MSE_Before	0.217069	0.234022	0.210585
R2_Before	**0.916522**	**0.910002**	**0.919015**
MAE_Before	0.322208	0.337752	0.307672
AAPRE_Before%	8.306425	8.732469	8.324728
APRE_Before%	1.63265	-0.38465	2.258463
MSE_After	0.203807	0.228071	0.196663
R2_After	**0.921622**	**0.912291**	**0.924369**
MAE_After	0.310595	0.33196	0.301341
AAPRE_After%	8.145633	8.975882	7.785155
APRE_After%	-1.5082	-2.01375	-1.21918
All-Stack			
Metric	**Stacking-Quantile (Recovery Factor)**		**Stacking-Quantile (NPV)**
MSE_Before	3.417935		0.211179
R2_Before	**0.960127**		**0.918787**
MAE_Before	1.354232		0.317784
AAPRE_Before%	3.665531		8.245644
APRE_Before%	0.379784		1.726153
MSE_After	3.053109		0.197816
R2_After	**0.964383**		**0.923926**
MAE_After	1.295451		0.304885
AAPRE_After%	3.61039		7.945188
APRE_After%	-0.28579		-1.44354

Specifically, the all-stacking ensemble of the four best models with cross-validation attained an R^2 of 0.978 and AAPRE of 2.71 for RF prediction, and an R^2 of 0.944 and AAPRE of 6.18 for NPV. Additionally, to further investigate calibration effects, quantile adjustment was specifically applied to the outputs of stacking with three selected models (without cross-validation), resulting in an R^2 of 0.964 and AAPRE of 3.61 for RF, and an R^2 of 0.924 and AAPRE of 7.94 for NPV. This enabled a direct comparison between the all-model stacking approach with cross-validation and the quantile-adjusted three-model stacking. Moreover, these analyses demonstrated that both strategies led to meaningful improvements in predictive reliability, with the four-model stacking and cross-validation delivering the most robust performance overall. Collectively, these findings underscore the value of architectural choice and output calibration for accurate EOR modeling and provide a practical framework for data-driven decision-making in chemical flooding optimization.

Conclusions

This study presents a comprehensive evaluation of supervised machine learning models for predicting oil recovery factor (RF) and net present value (NPV) in surfactant-polymer (SP) flooding, utilizing a diverse set of algorithms, systematic hyperparameter optimization, and advanced ensemble techniques. Consequently, the results demonstrate that ensemble approaches, especially stacking strategies combining CatBoost, ANN, LGBM, and GBR, substantially outperform simpler algorithms, delivering high predictive accuracy for both technical and economic performance indicators. Furthermore, the integration of Bayesian hyperparameter optimization and cross-validation further enhanced model reliability, reducing overfitting and improving generalization to unseen data.

Moreover, the targeted application of quantile adjustment to the outputs of selected stacking models yielded further improvements by correcting systematic prediction bias, thus refining the alignment of predicted and actual values. Ultimately, the comparative analyses reveal that ensemble stacking with cross-validation achieves the highest overall accuracy, while quantile adjustment offers additional calibration benefits in certain scenarios.

This research shows that machine learning pipelines using interpretable algorithms and rigorous evaluation can accurately screen EOR performance. In addition, such pipelines often eliminate the need for complex deep learning models in real-world applications. Moreover, the proposed workflow thus offers a robust, data-driven framework for supporting decision-making and optimization in chemical flooding projects, and sets the stage for further exploration of hybrid, uncertainty-aware, or physics-informed modeling approaches in the future.

This work also positions itself relative to prior CEOR modeling studies such as Kamari et. al.'s study in 2016 and Larestani et. al.'s study in 2022, which mainly relied on cascade neural networks, hybrid frameworks, or surrogate simulation strategies to achieve predictive accuracy. While such approaches have proven effective, they often entail higher computational costs and reduced interpretability. In contrast, the present study demonstrates that systematic benchmarking of diverse machine learning algorithms—combined with Bayesian hyperparameter optimization, cross-validation, ensemble stacking, and calibration—can deliver equally strong or even superior performance. This methodological positioning highlights the novelty of the proposed workflow and its practical value for EOR engineers, showing that robust predictions can be achieved without exclusive reliance on complex deep or hybrid ANN architectures.

Nomenclatures

AAPRE: Absolute Average Percentage Relative Error
ANN: Artificial neural network
APRE: Average Percentage Relative Error
CSA: Coupled Simulated Annealing
EFB: Exclusive Feature Bundling
EI: Expected Improvement
GBR: Gradient boosting regressor
GOSS: Gradient-based One-Side Sampling
MAE: Mean Absolute Error
MLP: Multilayer perceptron
ML: Machine learning
NPV: Net present value
RF: Recovery factor
RMSE: Root Mean Squared Error
SP: Surfactant-polymer
SVR: Support vector regression
TPE: Tree-structured Parzen Estimator
XGBoost: Extreme Gradient Boosting

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.petrol.2011.07.012

References

1. Al-Dousari, M. M., & Garrouch, A. A. (2013). An artificial neural network model for predicting the recovery performance of surfactant polymer floods. Journal of Petroleum Science and Engineering, 109, 51–62. https://doi.org/10.1016/j.petrol.2013.08.012.
2. Zerpa, L. E., Queipo, N. V., Pintos, S., & Salager, J.-L. (2005). An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. Journal of Petroleum Science and Engineering, 47(1-2), 197–208. https://doi.org/10.1016/j.petrol.2005.03.002.
3. Larestani, A., Mousavi, S. P., Hadavimoghaddam, F., Ostadhassan, M., & Hemmati-Sarapardeh, A. (2022). Predicting the surfactant-polymer flooding performance in chemical enhanced oil recovery: Cascade neural network and gradient boosting decision tree. Alexandria Engineering Journal. Advance online publication. https://doi.org/10.1016/j.aej.2022.01.023.
4. Karambeigi, M. S., Zabihi, R., & Hekmat, Z. (2011). Neuro-simulation modeling of chemical flooding. Journal of Petroleum Science and Engineering, 78(2), 208–219. https://doi.org/10.1016/j.petrol.2011.07.012.
5. Kamari, A., Gharagheizi, F., Shokrollahi, A., Arabloo, M., & Mohammadi, A. H. (2016). Integrating a robust model for predicting surfactant–polymer flooding

performance. Journal of Petroleum Science and Engineering, 137, 87–96. https://doi.org/10.1016/j.petrol.2015.10.034.
6. Hou, J., Li, Z., Cao, X., & Song, X. (2009). Integrating genetic algorithm and support vector machine for polymer flooding production performance prediction. Journal of Petroleum Science and Engineering, 68(1–2), 29–39. https://doi.org/10.1016/j.petrol.2009.05.017.
7. Dang, C., Nghiem, L., Nguyen, N., Yang, C., Chen, Z., & Bae, W. (2018). Modeling and optimization of alkaline–surfactant–polymer flooding and hybrid enhanced oil recovery processes. Journal of Petroleum Science and Engineering, 169, 578–601. https://doi.org/10.1016/j.petrol.2018.06.017.
8. Prasanphanich, J. (2009). Gas reserves estimation by Monte Carlo simulation and chemical flooding optimization using experimental design and response surface methodology (Master's thesis). University of Texas at Austin.
9. Yin, Z., Nan, Z., Cao, Z., & Zhang, G. (2021). Evaluating the applicability of a quantile–quantile adjustment approach for downscaling monthly GCM projections to site scale over the Qinghai-Tibet Plateau. Atmosphere, 12(11), 1170. https://doi.org/10.3390/atmos12111170.
10. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. Advances in Neural Information Processing Systems, 31, 6638–6648.
11. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3146–3154.
12. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). https://doi.org/10.1145/2939672.2939785.
13. Pravin, P. S., Tan, J. Z. M., Yap, K. S., & Wu, Z. (2022). Hyperparameter optimization strategies for machine learning-based stochastic energy efficient scheduling in cyber-physical production systems. Digital Chemical Engineering, 4, 100047. https://doi.org/10.1016/j.dche.2022.100047.
14. Kakimoto, Y., Omae, Y., Toyotani, J., & Takahashi, H. (2022). Fast screening framework for infection control scenario identification. Mathematical Biosciences and Engineering, 19(12), 12316–12333. https://doi.org/10.3934/mbe.2022574.
15. Pavlyshenko, B. (2018, August). Using stacking approaches for machine learning models. In 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP) (pp. 255–258). IEEE. https://doi.org/10.1109/DSMP.2018.8478510.