



OPEN Ensemble learning for prediction of inorganic scale formation: A case study in Oman

Mohammed Talib Said Al Harrasi, Alireza Kazemi✉ & Reza Yousefzadeh

Inorganic scale formation is one of the major flow assurance issues in geothermal energy, oil, and water production due to its damaging effects on formation rock, wells, and transportation facilities. Owing to the intricate nature of scale formation, developing a closed-form mathematical formulation for its prediction is difficult. Thereby, the ability of six machine learning algorithms and a Power Law Ensemble Model (PLEM) to predict inorganic scale formation in carbonate formations is examined in this study. A new dataset of scale formation from realistic wells in Oman, which included temperature, pressure, artificial lift, ionic composition, pH, total dissolved solids, and scale formation tendency of each well, was collected from two reservoirs (Natih and Shuaiba). The machine learning models are Naive Bayes (NA), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree (DT). The results revealed that the RF, KNN and DT provided the best predictions among the individual experts with a F1-scores of 78.6%, 75.9%, and 71.0%, respectively. By integrating the predictions of the individual experts using the PLEM method, the F1-score increased to 90.3% on the test subset. Moreover, a study was conducted to find out some rules of thumb for each input that cause scale formation, but the results showed that there was no explicit condition, indicating that scale formation is a complicated phenomenon requiring advanced modeling approaches. Finally, a new water analysis report was given to the PLEM (as the best model) to predict scale formation in a well that demonstrated a match between the predicted class and the report outcome.

Keywords Scale, Artificial lift, Machine learning, Data-driven, Geothermal

Scale formation is a major problem in petroleum and geothermal energy production. It occurs when seawater is injected into the formation (for pressure maintenance or geothermal energy production) or when two brines mix and the chemical stability is violated. This can lead to the creation of insoluble solids, which may block pipelines and equipment, reduce production rates, and increase operational costs¹. Scale formation can also have negative environmental impacts, as hazardous chemicals are often used to prevent it. Most of the reservoirs in Oman are composed of limestone, which is prone to scale formation. The usual completion type is an open hole with long horizontal legs. The majority of the wells are equipped with artificial lift systems—including gas lift, sucker rod pumps, and Electrical Submersible Pumps (ESP)—to facilitate the lifting of produced fluid from bottom hole to the surface. Due to the carbonate nature of the reservoirs in Oman and the use of water injection for pressure maintenance, scale formation is one of the primary challenges that appears on casing walls, production tubing, and downhole pumps. Frequent pump failures and the need for well interventions to replace pumps have caused many challenges, such as financial losses and unscheduled well shut-ins. Therefore, it is of great importance to predict scale formation at early stages to enable timely and appropriate preventive actions.

The main reason for scale formation is the incompatibility in ionic compositions, not abnormalities in ionic concentration, pH, pressure, or temperature. Consequently, developing simple rules or closed-form mathematical models to predict scale formation is not easy. The conventional way to determine whether scale formation occurs is through experimental laboratory tests. Nonetheless, The high cost and lengthy procedures of these tests limit their application. Consequently, alternative methods, such as data-driven methods, are often preferred over experimental methods². Machine learning models have been widely used to solve various problems in the oil and gas industry, including production prediction³, well log data generation⁴, anomaly detection in well log data⁵, reservoir property estimation⁶, Pressure Volume Temperature (PVT) property prediction⁷, reservoir characterization and numerical modeling⁸, estimation of minimum miscibility pressure⁹, and fracturing design

Department of Petroleum and Chemical Engineering, College of Engineering, Sultan Qaboos University, Muscat, Oman. ✉email: a.kazemi@squ.edu.om

optimization^{10,11}. In the context of scale formation prediction, Moghadasi et al.¹² developed an intelligent model for the prediction of scale formation as a function of the ionic composition of brines, temperature, and pressure. Hajirezaie et al.¹³ used two models to estimate the reduction in permeability as a function of barium concentration. Their results showed that the models could estimate the permeability reduction due to scale formation with an average relative error of only 1%. Al-Hajri et al.¹⁴ employed several intelligent models, including KNN, gradient boosting, decision tree, and support vector machine to predict calcium carbonate scale formation as a function of pH and ionic composition. Ahmadi and Chen¹⁵ used two models, an Artificial Neural Network (ANN) and a Least Squares – Support Vector Machine coupled with Genetic Algorithm (LSSVM-GA) to estimate formation damage due to deposition of scale during water injection. According to their results, the LSSVM-GA performed better than the ANN. Xu et al.¹⁶ developed five intelligent models, including Multi Linear Regression, KNN, Support Vector Regression, Random Forest (RF), and Gradient Boosting Regression, using 702 experimental samples to predict hydrate formation temperature in the presence of salty water. Yousefzadeh et al.¹⁷ compared the performance of ten machine learning algorithm in predicting scale formation under harsh conditions. However, their study did not include artificial lift type, pressure, and temperature as inputs despite their outstanding influence on scale formation.

To wrap up, although many works have studied scale formation prediction using data-driven models, they have not considered some operational factors, such as artificial lift type, pressure, and temperature at pump depth. Moreover, previous studies have not examined the ability of the PLEM method to predict the inorganic scale formation by integrating the predictions of individual machine learning models. The benefit of using the PLEM method is taking advantage of different types of learning algorithms to strengthen the reliability of predictions and reduce the bias toward a specific method. In this study, six data-driven models, such as Random Forest (RF), Decision Tree (DT), K-nearest Neighbors (KNN), Logistic Regression (LR), Support Vector Machine (SVM), and Naïve Bayes (MB), are adopted as six experts to predict inorganic scale formation as a function of ionic concentrations (Na^+ (ppm), Ca^{2+} (ppm), Mg^{2+} (ppm), SO_4^{2-} (ppm), HCO_3^- (ppm)), artificial lift type (AL Type), Total Dissolved Solids (TDS), temperature at pump depth, pressure at pump depth, and pH, collected from two reservoirs (Natih and Shuaiba) in Oman. Then, the outputs of the individual experts were integrated using the PLEM method coupled with Particle Swarm Optimization (PSO). The contribution of this investigation is demonstrating the ability of individual experts and the potential of the PLEM method to achieve reliable scale formation predictions.

Field description

Shuaiba and Natih oil reservoirs are two of the biggest carbonate reservoirs in the northwest of Oman that originate from the Cretaceous period¹⁸. The Shuaiba formation is 130 m thick, while the Natih formation reaches up to 419 m. Production takes place through a multitude of horizontal wells completed as open hole¹⁹. Three types of artificial lift methods are used to lift produced fluids from downhole to the wellhead, including gas lift, ESP, and sucker rod pumps. Over the past decade, the number of wells with scale buildup has shown a substantial increase in this field, damaging subsurface and surface (e.g. flow lines) equipment. The consequences encompass a variety of damages to the field, such as decreased production due to blocked flow paths, pump failures, and increased acid back flushing and CTU/hoist interventions.

Methods and materials

Figure 1 shows the overall workflow of the study. The first step is gathering the dataset. Then, the dataset is undergone a comprehensive statistical analysis to analyze the correlation between the input parameters, inspect the range of each parameter, assess the distribution of the parameters, and preprocess the data to make it ready for modeling. Next, individual data-driven experts are trained and validated. This step includes tuning the hyperparameters of the experts and analyzing their generalization on unseen data based on several evaluation metrics. Finally, the outputs of the individual experts are integrated using the proposed PLEM model. These steps and the utilized tools and models are described in more details in the following subsections.

Dataset

The utilized dataset consisted of 240 samples gathered from 240 wells from the two Omani reservoirs (Natih and Shuaiba). Each sample included 15 features, such as the well name, well type (producer or injector), artificial lift type (AL Type), target formation and reservoir, collection date, temperature and pressure at pump depth, ionic concentrations (Na^+ (ppm), Ca^{2+} (ppm), Mg^{2+} (ppm), SO_4^{2-} (ppm), HCO_3^- (ppm)), TDS (ppm), and pH. Previous studies did not include AL Type in their inputs and most of them excluded pressure and temperature from the list of inputs. The target of each sample was either 0 (no-scale) or 1 (scale).

The dataset was gathered from multiple sources, including water analysis, X-RD (X-ray diffraction) analysis²⁰, coiled tubing intervention, workover intervention, and downhole pumps' sensor data. By removing incorrect values and missing data, the total number of samples was decreased to 224. Moreover, five out of fifteen features were excluded from the features list since they were not useful for developing the predictive models. The excluded features were well name, well type, collection data, target formation, and target reservoir. Thereby, the inputs included ten features.

Since most of machine learning algorithms expect inputs as digits, we must convert categorical variables into digits. In this regard, AL Type was encoded to digits as follows: ESP (0), rod pump (1), and gas lift (2).

Before moving on to developing the data-driven models, statistical analysis and preprocessing of the variables is of great importance. A typical statistical analysis is studying the correlation between the input variables to discover highly correlated ones. Pearson's correlation and Spearman's correlation²¹ are two of the commonly used methods to compute the correlation between two variables. In this regard, the pairwise correlation between

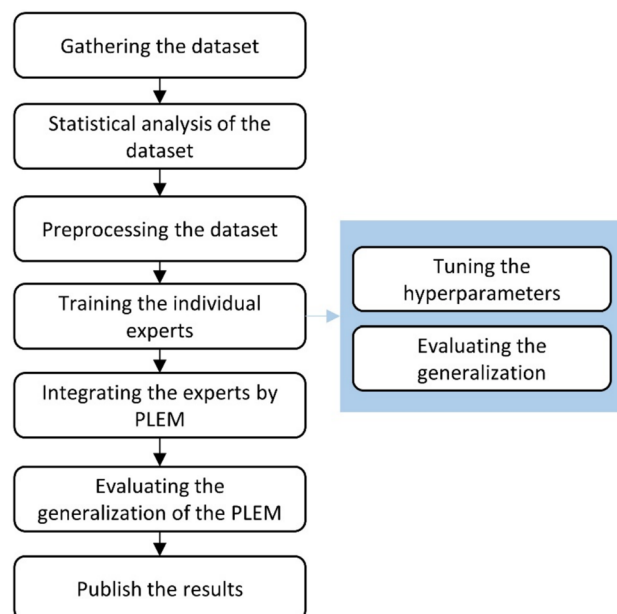


Fig. 1. Flowchart of the study.

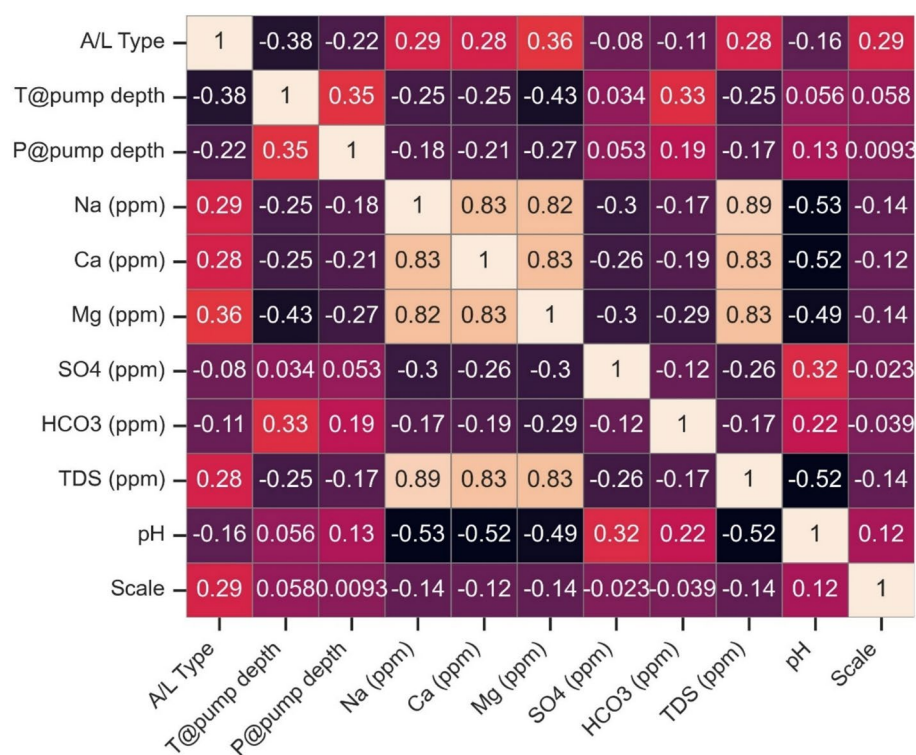


Fig. 2. Correlation matrix between the variables.

variables is checked using Pearson's correlation and one of two variables with a mutual correlation higher than 0.9 is excluded from the inputs. Figure 2 shows the correlation between the variables. As can be seen, there is no correlation higher than 0.9 and all variables are used for model development. To evaluate the significance of the observed correlations and address the null hypothesis, P-values are shown in Fig. 3. By comparing Figs. 2 and 3, it is evident that the P-value is low for high correlation coefficients, whereas it gets larger for weaker correlations. This is because the P-values indicate the statistical significance of the correlations under the null hypothesis of no association. In other words, if P-value < 0.05, it indicates a strong evidence against the null hypothesis that is

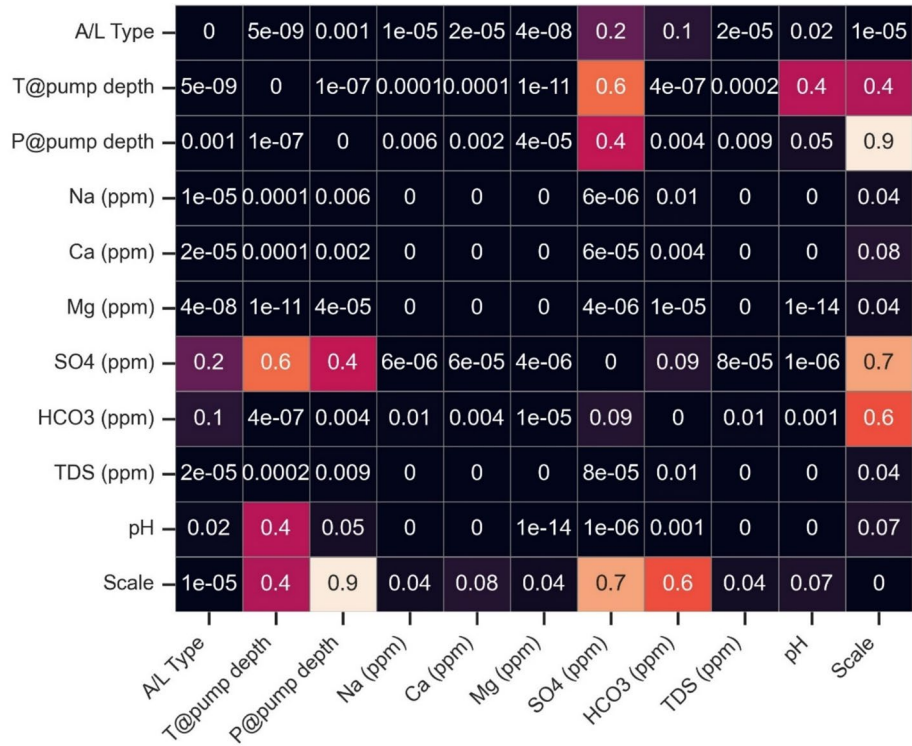


Fig. 3. P-values calculated from t-distribution for correlation coefficients.

	A/L type	Temperature @ pump depth (°C)	Pressure @ pump depth (psi)	Na (ppm)	Ca (ppm)	Mg (ppm)	SO4 (ppm)	HCO3 (ppm)	TDS (ppm)	pH	Scale
Count	224	224	224	224	224	224	224	224	224	224	224
Mean	0.91	80.57	637.58	33,703.99	8469.36	1408.71	230.25	148.56	128,402.30	6.15	0.35
Std	0.98	9.284	309.28	17,260.86	4436.41	662.58	357.11	54.02	60,809.63	0.48	0.47
Min	0	56	80	1836	1300	425.20	0	1	596	4.75	0
25%	0	76	426.75	18,305	4710.25	848.25	104.75	117.97	77,522.50	5.87	0
50%	0	83	614	28,960	7363.65	1261.70	167.50	146.85	113,425	6.06	0
75%	2	87	813	51,431.25	11,855	1837.50	241.25	176.50	190,045	6.42	1
Max	2	113	2363	66,890	23,200	3050	3370	336	263,000	7.96	1
Skewness	0.19	-0.29	1.28	0.27	0.45	0.59	6.46	0.42	0.177	0.57	0.64

Table 1. Statistical analysis of the utilized dataset.

no relation between two variables. In contrast, P-value > 0.05 indicates there is not enough evidence to reject the null hypothesis, which means that the low correlation between two variables is confirmed.

Another statistical analysis performed on the utilized dataset involves analyzing its statistical moments, such as mean, standard deviation, minimum, maximum, etc. to find out the diversity of the data. The results of this analysis are reported in Table 1. As shown in Table 1, the collected data points cover a wide range of operational conditions, making them suitable for developing and testing the data-driven models. Moreover, as shown in Table 1, the range of the variables differs significantly. For example, pH values are below 10, while Na⁺ concentrations are on the order of ten thousands. Thereby, rescaling the variables to similar intervals is paramount. To achieve this, first, the dataset was divided into training and testing subsets using an 80:20 ratio. Then, the samples were standardized to make training more efficient, as shown in Eq. 1.

$$X_s = \frac{X - \overline{X}}{\sigma},$$

(1)

where X_s is the standardized input, $\overline{X}$ denotes the mean, and σ denotes the standard deviation of inputs over the samples in the training subset.

Finally, the distribution of the ten input variables (along with the target formation) and the output is depicted in Fig. 4. Based on this figure:

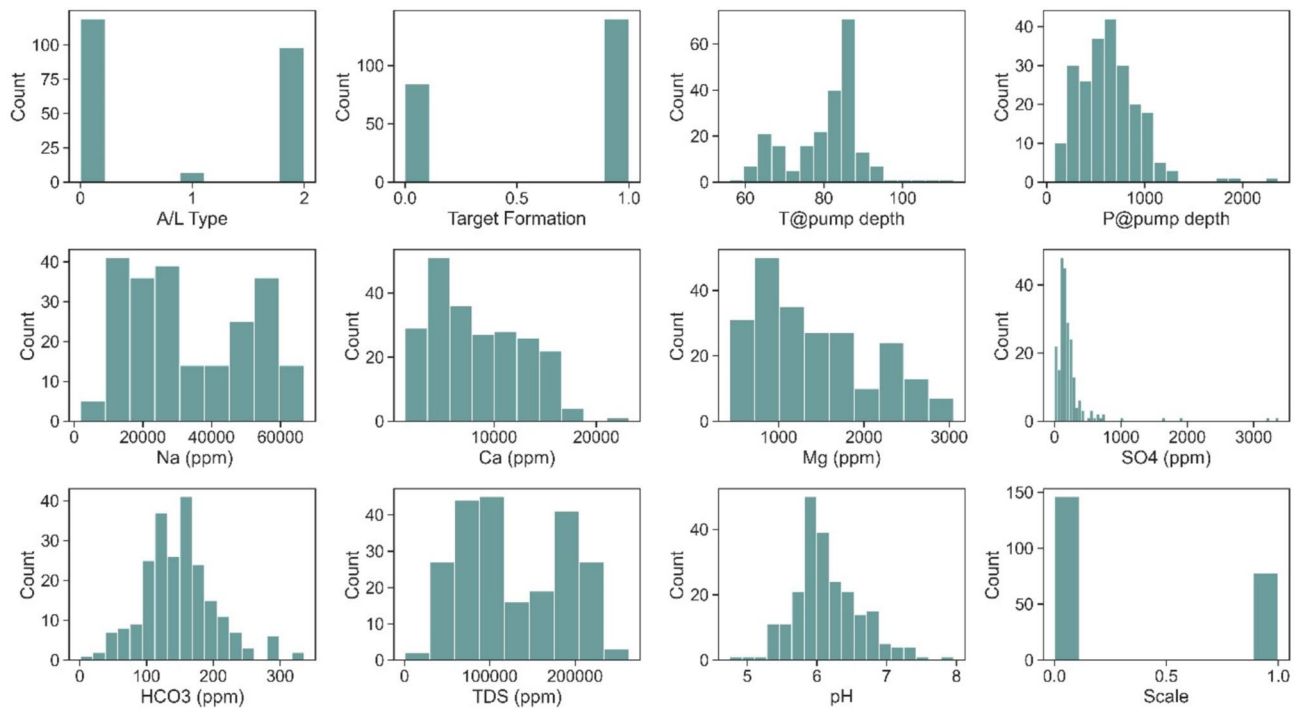


Fig. 4. Distribution of the inputs and the output.

- ESP is the AL Type for 119 samples, sucker rod pump is seen in 98 samples, and gas lift appears in only 9 samples.
- Most of the inputs (Pressure, Ca, Mg, SO_4 , HCO_3 , and pH) have a skewed and nearly Gaussian distribution, which is favorable for classification algorithms that rely on the Gaussian assumption for input variables, while the others have a bi-modal or Bernoulli distribution.
- The non-Gaussian distribution of some variables make it hard for Gaussian-based classification algorithms, such as the LR and NB, to model scale formation.
- The last plot (bottom-right) indicates that almost 2/3 of the samples show no scale (0), while 1/3 of them show scale formation. Thereby, the dataset is imbalanced, which necessitates the use of classification metrics other than accuracy for evaluating the predictive models.

The machine learning models were trained using 80% of the data and tested on the remaining 20%. To tune the hyperparameters of the models, grid search with fivefold cross validation was used.

Naïve bayes

Naïve Bayes (NB) classifier is a probabilistic learning algorithm that leverages statistical principles to learn from data¹⁷. NB can be employed to solve classification problems, such as predicting scale formation. The fundamental rule of this algorithm is Bayes' theorem, as shown in Eq. 2. According to Bayes' theorem, the probability of a sample belonging to class C_m given X is denoted by $p(C_m|X)$, where X is the vector of input features $(x_1, x_2, \dots, x_n)^T$ and C_m is the m -th class. Note that, in Bayes' theorem, the variables are assumed to be conditionally independent. Based on this assumption, Eq. 2 can be rearranged to Eq. 3²².

$$p(C_m|X) = \frac{p(C_m)p(X|C_m)}{p(X)}, \quad (2)$$

$$p(C_m|X) = \frac{1}{p(X)} \cdot p(C_m) \prod_{i=1}^n p(x_i|C_m) \quad (3)$$

Based on Eq. 3, the probability of scale formation ($C_m = 1$) given the input variables is equal to the product of the probability of x_i given $C_m = 1$, multiplied by the prior probability of scale formation (calculated as the ratio of scale samples to all samples), and divided by the probability of observing the given inputs, $p(X)$. Since $p(X)$ is intractable, it is omitted from Eq. 3, which leads to Eq. 4.

$$p(C_m|X) \propto p(C_m) \prod_{i=1}^n p(x_i|C_m) \quad (4)$$

which is known as the *maximum likelihood*. Thereby, by calculating the likelihood for each class, the class with the maximum likelihood is chosen as the most probable class.

Logistic regression

Logistic regression (LR) is a machine learning technique rooted in statistics²³. LR computes the probability that a sample belongs to a specific class. This algorithm resembles a regression algorithm but with a categorical output due to applying the logistic function to the linear combination of the inputs, as shown in Fig. 5. In this figure, b denotes the bias term, and σ is the sigmoid/logistic function.

The logistic function produces outputs between 0 and 1. Moreover, it has a different learning process, which is based on the assumption of conditional independence among the variables and a Gaussian distribution of the inputs. At first, LR was able to solve problems with only two classes, but later on it was extended to multi-class problems²⁴.

Assuming that there are only two classes (0 and 1), the probability that a sample belongs to class 1 is given as follows, which is based on Bayes' theorem:

$$P(Y = 1|X) = \frac{P(Y = 1)P(X|Y = 1)}{P(Y = 1)P(X|Y = 1) + P(Y = 0)P(X|Y = 0)} \quad (5)$$

After dividing the nominator and denominator by the nominator, we will have:

$$P(Y = 1|X) = \frac{1}{1 + \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}} \quad (6)$$

It is equivalent to:

$$P(Y = 1|X) = \frac{1}{1 + \exp\left(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)}\right)} \quad (7)$$

Since the features are assumed to be conditionally independent:

$$P(Y = 1|X) = \frac{1}{1 + \exp\left(\ln \frac{P(Y=0)}{P(Y=1)} + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right)} = \frac{1}{1 + \exp\left(\ln \frac{1-\pi}{\pi} + \sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)}\right)} \quad (8)$$

For the summation term, assuming that $P(X|Y = y)$ is Gaussian:

$$\sum_i \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \sum_i \ln \frac{\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(X_i - \mu_{i0})^2}{2\sigma_i^2}\right)}{\frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(\frac{-(X_i - \mu_{i1})^2}{2\sigma_i^2}\right)} = \sum_i \left(\frac{\mu_{i0} - \mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}\right) \quad (9)$$

Substituting this into Eq. 6 gives:

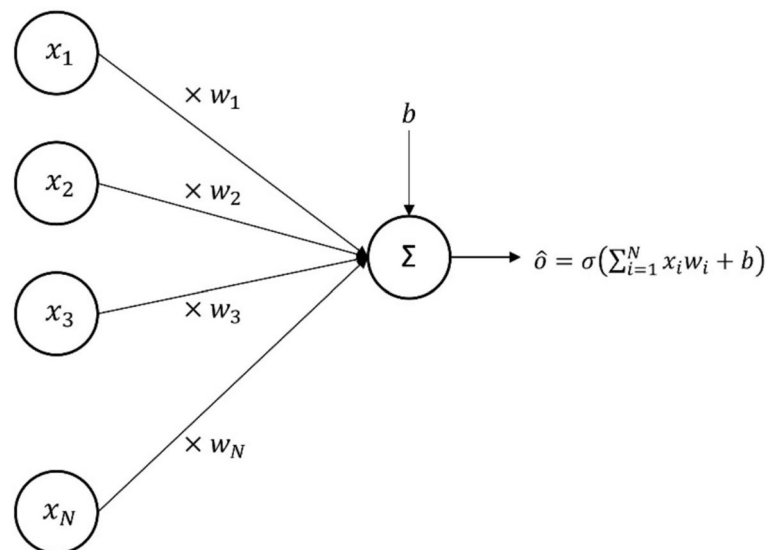


Fig. 5. Schematic diagram of logistic regression.

$$P(Y = 1|X) = \frac{1}{1 + \exp\left(\ln \frac{1-\pi}{\pi} + \sum_i \left(\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2} X_i + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\sigma_i^2}\right)\right)}, \quad (10)$$

which is equivalent to:

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)} \quad (11)$$

Equation 11 is optimized, which is equivalent to maximizing the likelihood of assigning a sample to class 1 using optimization methods such as gradient ascent.

K-nearest neighbors

K-Nearest Neighbors (KNN) is a non-parametric learning algorithm introduced by Evelyn Fix and later upgraded by Thomas Cover^{25,26}. KNN can be used for both classification and regression. The overall principle is that the label of a sample is determined based on the label(s) of its surrounding neighbor(s) in the database. For classification, majority voting is usually used to determine the output, and for regression, arithmetic or distance-based averaging can be used. K determines the number of nearest neighbors to do majority voting or averaging. K = 1 means that the closest neighbor is only used. As the name of the method implies, the nearest neighbors must be found to estimate the output of a sample, which requires calculating the distance between the query sample and all other samples in the training dataset. If the scale of the input features differs substantially, standardization of the samples is vital to have a fair distance calculation^{27,28}. The final class of a query sample (x_q) can be determined by means of Eq. 12.

$$\hat{f}(x_q) \leftarrow \arg \max_{c \in C} \sum_{i=1}^K \delta(c, f(x_i)), \delta(a, b) = 1 \text{ if } a = b, \quad (12)$$

where x_q denotes the query sample, and x_i is the i -th neighbor of the query sample, K is the number of nearest neighbors, $\delta(a, b)$ is the Kronecker delta which is 1 if $a = b$ and 0 otherwise, and f determines the label of x_i among C classes. Figure 6 shows the schematic diagram of KNN with four classes. The circle shows the five ($K = 5$) nearest neighbors of the query sample (x_q).

In distance-weighted version of the KNN, the inverse of the distance between the query sample and adjacent samples is used as the weight of that sample. In this manner, the label of the query sample is determined as follows:

$$\hat{f}(x_q) \leftarrow \arg \max_{c \in C} \sum_{i=1}^K w_i \delta(c, f(x_i)), \delta(a, b) = 1 \text{ if } a = b, \quad (13)$$

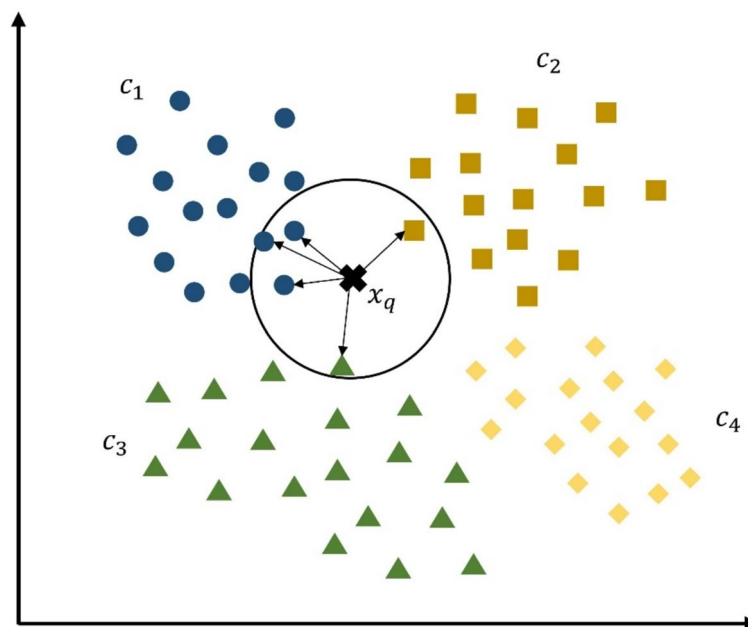


Fig. 6. Schematic diagram of KNN with four classes and K = 5.

where w_i is the weight of sample i . Accordingly, as a sample gets closer to the query sample, it will have a stronger impact on the final label of the query sample.

Decision tree

A decision tree is a supervised machine learning algorithm applicable to both classification and regression tasks. It is a flowchart-like structure that consists of a root node, branches, internal nodes, and leaf nodes, as shown in Fig. 7.

The root node represents the starting point of the tree, and the leaf nodes represent the final predictions. The branches represent the possible outcomes of a decision, and the internal nodes represent the tests that are used to make decisions²⁹. The complicated process of decision making is broken into multiple processes that are simpler than the original problem³⁰. Decision trees are trained using a top-down, greedy approach. This means that the algorithm starts at the root node and recursively splits the dataset into increasingly homogeneous subsets. The splitting is done by finding the attribute that separates the data into two groups in an optimal manner. The attribute that is selected is the one that has the highest information gain or Gini index, which indicates how well the attribute separates the data into two groups. In this study, the Gini index is used to select the nodes. The Gini index is calculated by means of Eq. 14.

$$Gini = 1 - \sum_{i=1}^C (p_i)^2, \quad (14)$$

where C is the number of classes (2 in this study), and p_i is the probability of the i^{th} class. The Gini index is calculated for each split, and the larger and easy to implement splits are preferred to smaller splits with distinctive values.

Random forest

Random forest (RF) is an ensemble learning method composed of multiple decision trees. These trees are trained independently using random subsets from the training dataset with randomly selected features³¹. Random selection of samples and features assures diversity in the tree building process, which reduces the chance of overfitting. The number of selected features is limited by the maximum number of features, which should be tuned by trial and error. The intuition of the random forest tree-building approach is to use several decision trees, instead of only a single tree. The ultimate output is determined through averaging or majority voting, depending on the nature of the problem (regression or classification). In classification tasks, majority voting is used to determine the final label of the sample. An advantage of RF over conventional decision trees is its better stability³². As stated, several trees are constructed to build a random forest. These trees should have a decent performance. Thereby, an out-of-bag (OOB) subset is separated from each random subset of samples to calculate the prediction error as the approximation of the overall error. The schematic diagram of an RF is shown in Fig. 8.

Support vector machine

Support vector machines (SVMs) were introduced by Cortes and Vapnik³³. One of the great advantages of SVMs is their ability to solve non-linear problems via the kernel trick. By means of the kernel trick, the original data is transformed from the original space, where it is not linearly separable, to a higher-dimensional space, where

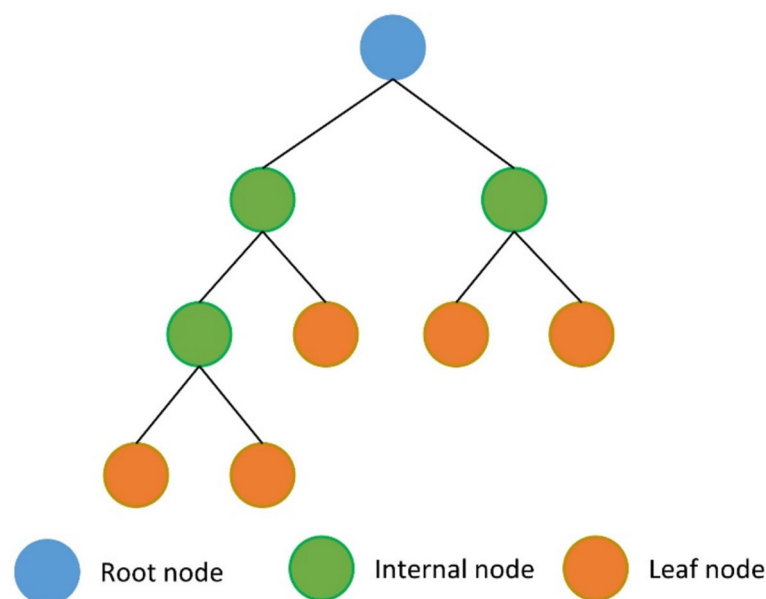


Fig. 7. Schematic representation of a decision tree.

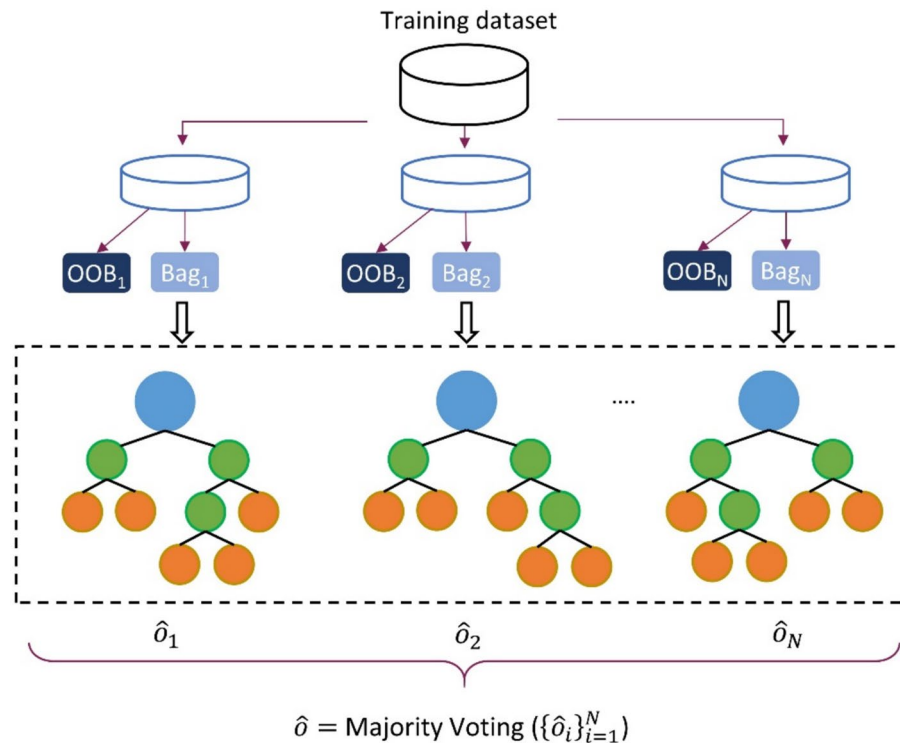


Fig. 8. Schematic diagram of a random forest.

data is linearly separable. The classification principle in SVMs relies on maximizing the margin between the classes. The margin between the classes is defined as the distance between the support vectors from which two hyperplanes are drawn, one for each class. The support vectors are found in a way to maximize the margin between the hyperplanes^{34–36}. The decision boundary that separates the classes is drawn from the middle of the hyperplanes. Assuming that $X = [x_1^T, x_2^T, \dots, x_M^T]$ is the matrix of inputs with M samples, and ϕ is the mapping function to transform the original data to a higher-dimensional space, the classification boundary is given by Eq. 15.

$$W^T \phi(X) + b = 0, \quad (15)$$

where W is the vector of weights and b is the bias term. The margin is calculated as the line that separates the support vectors with the shortest perpendicular distance. This distance is two times the distance between the decision boundary and the hyperplanes. If the distance between the two hyperplanes is ρ and we scale the weights and bias by $\rho/2$, for each sample we will have:

$$y_i [W^T \phi(x_i) + b] \geq 1, \quad i = \{1, 2, \dots, M\} \quad (16)$$

Each support vector (x_s, y_s) lies on a hyperplane, which means that the inequality becomes an equality. Now, we can calculate the distance from the support vectors to the decision boundary (d) as:

$$d = \frac{y_s (W^T x_s + b)}{\|W\|} = \frac{1}{\|W\|}, \quad (17)$$

in which $\|W\|$ is norm 2 of the weight vector. Since the distance between the support vectors and the classification boundary was half the margin between the hyperplanes, the margin will be $\frac{2}{\|W\|}$. Now, the purpose is maximizing $\frac{2}{\|W\|}$, where we can minimize $\frac{1}{2} W^T W$, instead. Thereby, we should solve the following minimization problem:

$$\begin{aligned} \arg \min_{W, b} \quad & \frac{1}{2} W^T W \\ \text{s.t. } & y_i [W^T \phi(x_i) + b] \geq 1, \text{ for } i = 1, \dots, M \end{aligned} \quad (18)$$

Hyperplanes with the maximum margin give the highest generalization on unseen data. To further improve the generalization of the model, slack variables (ξ) are introduced, which convert the hard-margin policy to the soft-margin policy. Slack variables are defined as the distance between misclassified samples from their class-specific

hyperplane, as shown in Fig. 9 for a bi-variate example. In this manner, we ignore the strict rule of correctly classifying all training examples, and let the model make some mistakes in the training phase with the benefit of improved generalization. By considering the slack variables, the problem in Eq. 18 becomes:

$$y_i [W^T \phi(x_i) + b] \geq 1 - \xi_i, \quad (19)$$

where ξ_i is the slack variable for the i^{th} sample. Now, the minimization problem can be re-written as:

$$\begin{aligned} \arg \min_{W, b} & \left(\frac{1}{2} W^T W + c \sum_i \xi_i \right), \\ \text{s.t. } & y_i [W^T \phi(x_i) + b] \geq 1 - \xi_i, \text{ for } i = 1, \dots, M \end{aligned} \quad (20)$$

where c is the regularization coefficient, which specifies the importance of the slack variables. The problem above is a dual optimization problem for which the Lagrange approach is adopted to solve. This approach incorporates the inequality and equality constraints to the dual-optimization problem and transforms it to a single-optimization problem.

$$\begin{aligned} L(W, b, \alpha) &= \frac{1}{2} W^T W - \sum_{i=1}^M \alpha_i [y_i (W^T \phi(x_i) + b) - 1] \\ \text{s.t. } & 0 \leq \alpha_i \leq c, i = 1, \dots, M \end{aligned} \quad (21)$$

where α_i s are known as the Lagrange multipliers. To optimize the above problem (Eq. 21), the derivative of the loss function, L , w.r.t to W and b should be set to zero. The Karush–Kuhn–Tucker (KKT) conditions yield: $W = \sum_{i=1}^M \alpha_i y_i \phi(x_i)$ and $\sum_{i=1}^M \alpha_i y_i = 0$. Plugging these back into Eq. 21, the dual formulation is obtained.

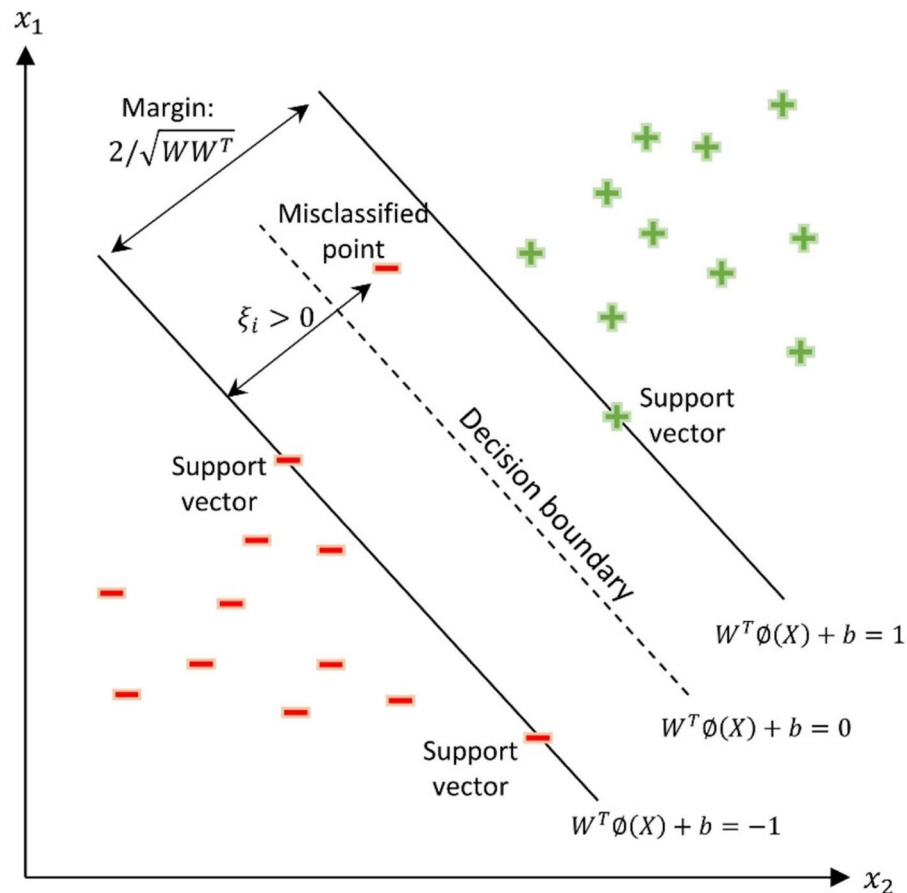


Fig. 9. Schematic of the SVM.

$$\begin{aligned} \arg \max_{\alpha} & -\frac{1}{2} \sum_{i,j=1}^M \alpha_i \alpha_j y_i y_j \varnothing(x_i) \varnothing(x_j) + \sum_{i=1}^M \alpha_i, \\ \text{s.t.} & \sum_{i=1}^M \alpha_i y_i = 0, 0 \leq \alpha_i \leq c, i = 1, \dots, M \end{aligned} \quad (22)$$

To solve Eq. 22, Quadratic Programming is used, which gives the Lagrange multipliers. Only the support vectors have non-zero Lagrange multipliers. Since the bias term (b) stays out of the dual formulation, we have to obtain it by implementing the initial constraints. According to Mercer's theorem³⁷, we can avoid computing $\varnothing(x_i) \varnothing(x_j)$ since it is computationally demanding. Instead, we can use the kernel trick^{38,39}, in which $\varnothing(x_i) \varnothing(x_j)$ is replaced with a kernel function $K(x_i, x_j)$. A kernel function is, in fact, the mathematical result of $\varnothing(x_i) \varnothing(x_j)$ in a symbolic form, which should be found by trial and error. During the inference time, the model's output given a new sample z can be obtained as follows:

$$y = \text{sign} \left(\sum_{i=1}^M \alpha_i y_i K(x_i, z) + b \right) \quad (23)$$

Power law ensemble model

Ensemble learning is a learning paradigm that solves a problem by integrating the outputs of individual predictive models. This approach allows us to benefit from the strengths of different kinds of learning algorithms, rather than relying on a single method. We can merge the outputs of the ensemble models using linear or non-linear combinations. The former is known as the static strategy, and the latter is known as the dynamic strategy. The static strategy is easier to implement but tends to produce sub-optimal results. The dynamic strategy requires optimizing the coefficients of the non-linear combination and yields decent results. One of the dynamic strategies is the PLEM method, in which a power law model is used to combine the outputs of the individual experts in the ensemble, as shown in Eq. 24.

$$y = \sum_{i=1}^S \alpha_i y_i^{\beta_i}, \quad (24)$$

where S is the size of the ensemble (6 in this study), α_i is the coefficient of the i^{th} experts, and β_i is the exponent of the i^{th} expert. The coefficients and exponents need to be optimized by an optimizer to achieve the best results from PLEM. We have adopted the Particle Swarm Optimization algorithm (PSO) to perform this optimization.

The choice of PLEM with PSO was motivated by both its flexibility and capacity to capture non-linear relationships among the outputs of individual experts. Unlike conventional ensemble learning strategies such as stacking or weighted voting, which typically rely on linear or fixed-weight aggregations, PLEM offers non-linear integration that can better reflect varying confidence levels or predictive performance of individual models across different prediction intervals. Furthermore, using PSO for optimizing the coefficients of PLEM enables efficient global search effectively in complex, non-convex solution spaces—a great advantage when ensemble outputs are noisy or moderately correlated.

Evaluation metrics

As a classification task, the ability of the individual experts and the PLEM method to predict scale formation is evaluated using both quantitative and visual metrics. These include the Receiver Operating Characteristic (ROC) curve, confusion matrix, and classification metrics, such as accuracy, precision, recall, and F1-score. Before moving on to the definition of these metrics, some common terms should be defined.

True positive (TP): TP is attributed to a sample that its predicted and actual label is positive.

False positive (FP): FP refers to a sample that it is predicted as positive, but its actual label is negative.

True negative (TN): TN refers to a sample that its predicted and actual label is negative.

False negative (FN): FN is attributed to a sample that it is predicted as negative, but its actual label is positive.

Now, the classification metrics are defined as follows.

Accuracy is defined as the portion of all correctly classified samples, as given in Eq. 25. Precision is defined as the intersection of positive predictions with the actual predictions, indicating that how accurate the positive predictions are, which is defined in Eq. 26. Recall determines how accurate the model acts on positive samples. In other words, it measures the portion of the actual positive samples correctly recognized as positives (Eq. 27). Finally, to merge the precision and recall to have a unified metric, F1-score is used, which is defined in Eq. 28.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (25)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (26)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (27)$$

$$F1 - score = 2 \times \frac{Precision \times recall}{precision + recall} \quad (28)$$

Confusion matrix

For binary classification, the confusion matrix is a 2×2 matrix, in which the rows represent the actual classes and the columns represent the predicted classes. The diagonal cells contain the true positive and true negative counts, while the off-diagonal cells show the false positives and false negatives. Figure 10 shows a schematic of a confusion matrix.

ROC curve

The ROC curve is a visual performance metric for a binary classifier across different decision thresholds for differentiating between positive (1) and negative (0) classes. The ROC curve is generated by plotting the true positive rate (TPR) against the false positive rate (FPR) at multiple thresholds, where TPR is $TP/(TP + FN)$, while FPR is $FP/(FP + TN)$.

As the ROC curve approaches to the top-left corner, both specificity and recall increase. To have a single quantitative measure of the model's performance based on the ROC curve, the Area Under the Curve (AUC) is utilized. A higher AUC is an indication of better recall and specificity at various thresholds.

Results and discussion

To achieve optimal predictive performance from machine learning models, their hyperparameters must be tuned. A common method for hyperparameter tuning is grid search, where a pre-specified set of hyperparameter combinations is examined based on performance on a validation subset. The set of hyperparameters leading to the highest validation score (e.g., accuracy) is chosen as the best set.

Nonetheless, when the training dataset is small, separating a subset of samples as a validation subset makes the training set even smaller. Thereby, K-fold cross validation is used to tackle this challenge. In this strategy, the training set is divided into K non-overlapping folds; training is done K times with fixed hyperparameters, and each time one of the folds is used as the validation subset and the remaining folds for training. Each time the validation score is calculated and, in the end, the ultimate score is calculated as the average of the validation scores on K folds, known as the cross-validation score. The set of hyperparameters with the highest cross-validation score is chosen as the optimal hyperparameters to train the final model.

In this study, grid search with five-fold ($K = 5$) cross validation was used to tune the hyperparameters. The range of hyperparameters evaluated for each machine learning model, along with the optimal values and cross-validation scores, are reported in Table 2.

The optimal hyperparameters were used to train the individual experts. Then, their performance on the test subset was evaluated and reported in Table 3.

Table 3 shows that RF, KNN, and DT have the best performance across all evaluation metrics. NB shows the worst performance since it assumes that the input variables are conditionally independent and follow a Gaussian distribution. However, the reality is something else. In practice, scale formation occurs due to a combination of various factors, including ion concentration, pH, pressure, temperature, TDS, and artificial lift type. Thereby, assuming that the input variables are conditionally independent is not a good assumption, as they are correlated. Moreover, Fig. 4 showed that not all the variables follow a Gaussian distribution. For the same reason, LR also showed indecent results. The best-performing method, which achieved high values for all evaluation metrics, is RF. This is because RF uses random subsets of features to construct its estimators, resulting in reduced bias. This means we do not rely on only a single set of features; instead, various subsets of features are used to build the estimators. Each estimator produces its own class prediction (0 or 1), and the final result is computed based on

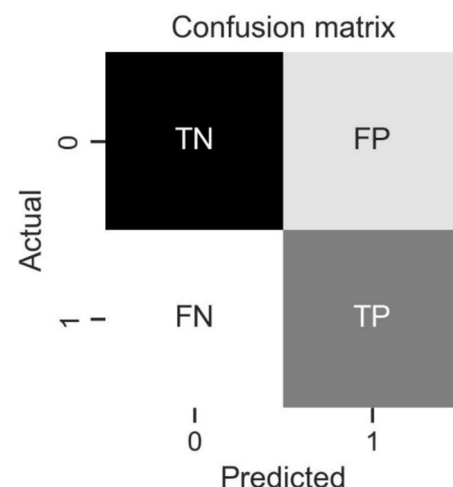


Fig. 10. Schematics of a confusion matrix.

Model	Hyperparameter	Optimal value	Range	Cross-validation score
LR	C	2	1–100	76.6%
	Fit intercept	True	True, False	
	Solver	Saga	lbfgs, Sag, Newton	
	Penalty norm	L2	L1, L2	
	Max iterations	200	100, 200, 300, 400	
RF	Criterion	Entropy	Entropy, Gini	88.3%
	Max_depth	5	2–10	
	n_estimators	100	50–500	
	Min_samples_split	3	2–20	
	Min_samples_leaf	1	1–5	
	Max_features	Auto	Sqrt, log2, Auto	
	oob_score	True	True, False	
DT	Criterion	Entropy	Entropy, Gini	81.4%
	Min_samples_split	100	2–150	
	Min_samples_leaf	1	1–5	
	Max_depth	6	2–10	
SVM	Kernel	Rbf	Linear, Poly, Rbf	79.4%
	Probability	True	True, False	
	C	1	0.001, 0.01, 0.1, 1, 10	
KNN	n_neighbors	5	1–11	84.8%
	Metric	Minkowski	Minkowski, Euclidean, l1, l2	
	P	2	1, 2	
NB	Fit_prior	True	True, False	74.2%
	Class_prior	None	None, True	

Table 2. Optimal hyperparameters for each model, along with their investigated range and cross-validation scores.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
KNN	84.4	73.3	78.6	75.9
SVM	77.8	64.3	64.3	64.3
NB	71.1	52.9	64.3	58.1
RF	86.7	78.6	78.6	78.6
LR	75.6	61.5	57.1	59.3
DT	80.0	64.7	78.6	71.0

Table 3. Performance of the machine learning models to predict scale formation on the test subset.

the majority voting paradigm. This approach leads to the creation of diverse rules for predicting scale formation and reduces overfitting. Additionally, tree-based algorithms are effective at extracting complicated rules from tabular data and are not constrained by the assumptions of conditional independence or Gaussian distribution of variables, which also explains the good performance of DT. The reason for the outperformance of RF over DT lies in RF's reduced overfitting due to its use of an ensemble of trees, rather than relying on a single tree. The second best model is KNN. The reason for its good performance is that it is an instance-based algorithm and does not estimate the output using a universal function and does not impose assumptions like conditional independence or Gaussian distribution on the input variables. Since scale formation as a function of the input variables is a complex phenomenon, it is difficult to model it with a universal function. It is worth pointing out that the best three methods have similar recall values, but different precisions, which indicates that their differences lie in how accurate is their positive predictions are.

To assess the performance of the individual models for each class, confusion matrices were constructed and are shown in Fig. 11. The numbers on the main diagonal of the confusion matrix show the true negatives and true positives. Specifically, the top-left number shows the true negatives, and the bottom-right number shows the true positives. In line with Table 3, KNN, DT, and RF have the highest number of true positives, while LR has the fewest. Accordingly, KNN, DT, and RF have a better accuracy at predicting the positive samples than other individual models, meaning that most positive samples were correctly recognized by them. The reason for the better performance of RF compared to KNN and DT is its higher number of true negatives, indicating that RF was more successful in correctly identifying negative (no-scale) samples; however, DT misclassified three of those negative samples as positives, and KNN misclassified one as positive (scale). As shown in Fig. 11, other

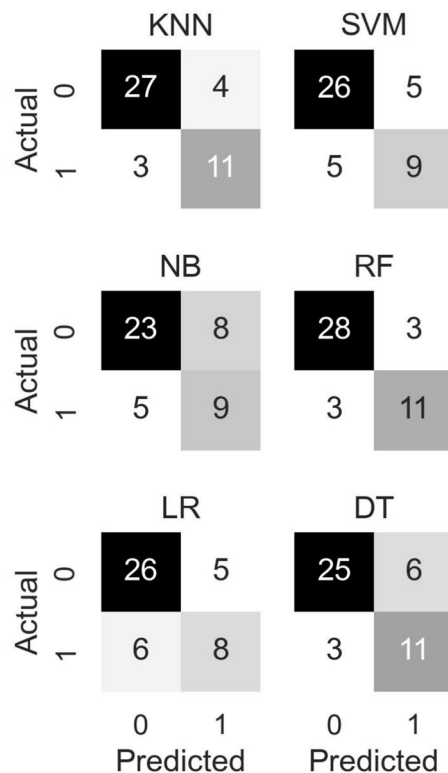


Fig. 11. Confusion matrix for each machine learning model on the test subset.

models (SVM, NB, and LR) have fewer true positives and true negatives, and more false positives and more false negatives than the other three models. In this regards, NB had the highest number of false positives (8), and LR had the highest number of false negatives (6). This indicates that NB tends to predict no-scale conditions as scale with a high probability, which can lead to unnecessary treatments and financial loss. On the other hand, all the three models (SVM, NB, and LR) have a high probability of missing actual positives (scale conditions), potentially resulting in damage to facilities due to undetected scale formation.

To make the comparison more convenient, the ROC curve for each machine learning model, along with their calculated AUC, is illustrated in Fig. 12. As the ROC curve approaches to the top-left corner, it indicates high recall (low false positives), which means that most of the positive predictions by the model were truly positives. According to Fig. 12, RF has the highest AUC, followed by KNN and DT. The lowest AUC is achieved by LR. Therefore, the results provided by the ROC curves are in agreement with the results in Table 3 and Fig. 11. Consequently, positive samples are mostly captured by RF, KNN, and DT models, and their positive predictions were actually positives, making them more reliable tools than other individual models for predicting inorganic scale formation.

To improve the predictions, the outputs of the individual machine learning models were integrated using the PLEM method. To do so, the coefficients of the PLEM model were optimized using the PSO algorithm on the training subset, and the predictions were made on the testing subset. Figure 13 shows the confusion matrix and ROC curve for the PLEM method on the test subset. As can be seen, the PLEM method has successfully identified all 14 positive samples in the test subset, and its only failure was in three negative samples incorrectly labeled as positives. Thus, three out of seventeen positive predictions by the PLEM method were incorrect, reducing its precision. Moreover, the ROC curve with an AUC of 0.95 shows a stronger inclination toward the top-left corner than the ROC curves of the individual models, indicating the superior performance of the PLEM method over its constituent models.

Figure 14 provides a comprehensive comparison between the metrics of the PLEM and the individual models. The PLEM model outperformed its constituent parts in all evaluation metrics. A recall of 1.0 for the PLEM indicates that all positive samples were recognized by the model, but it has incorrectly classified three negative samples as positives, resulting in a precision of 0.824. Consequently, no-scale conditions are reliably predicted by the PLEM model (no false negatives). In other words, when the PLEM predicts that no scale will form, it is 100% accurate, but when it predicts scale forming, it is 82.4% accurate. Nevertheless, its predictions (scale or no-scale) are more reliable than those of the individual models.

In the following, each input variable is plotted against scale formation to identify any possible explicit rule for scale formation. Figure 15 illustrates the input–output analysis. According to this figure, when analyzing the relationship between each variable and scale formation, no clear distinction is observed. In other words, scale formation occurs within similar intervals of each variable. Therefore, scale formation does not occur because a specific variable is higher or lower than a threshold but due to a combination of conditions and incompatibility

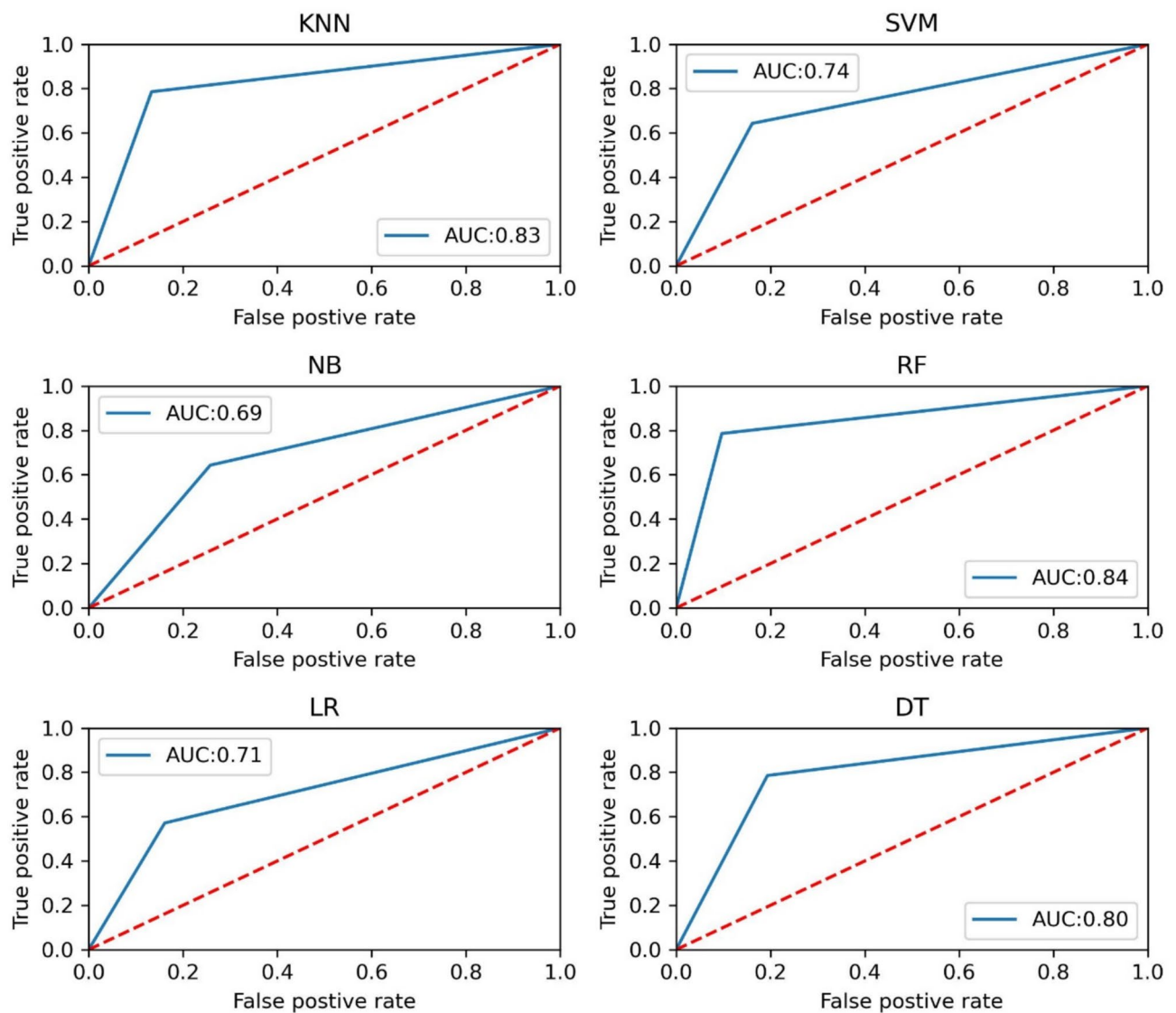


Fig. 12. ROC curves for the individual machine learning models.

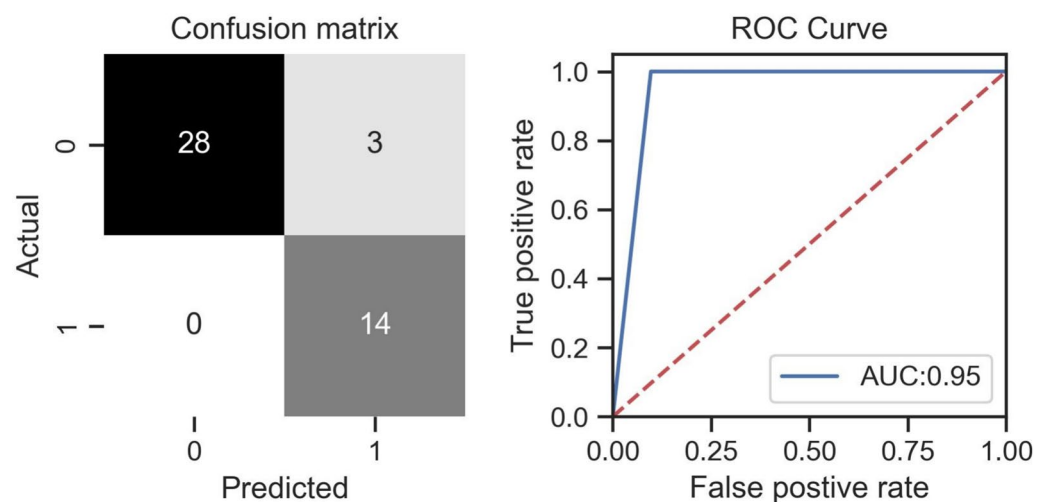


Fig. 13. Confusion matrix and ROC curve for the PLEM method on the test subset.

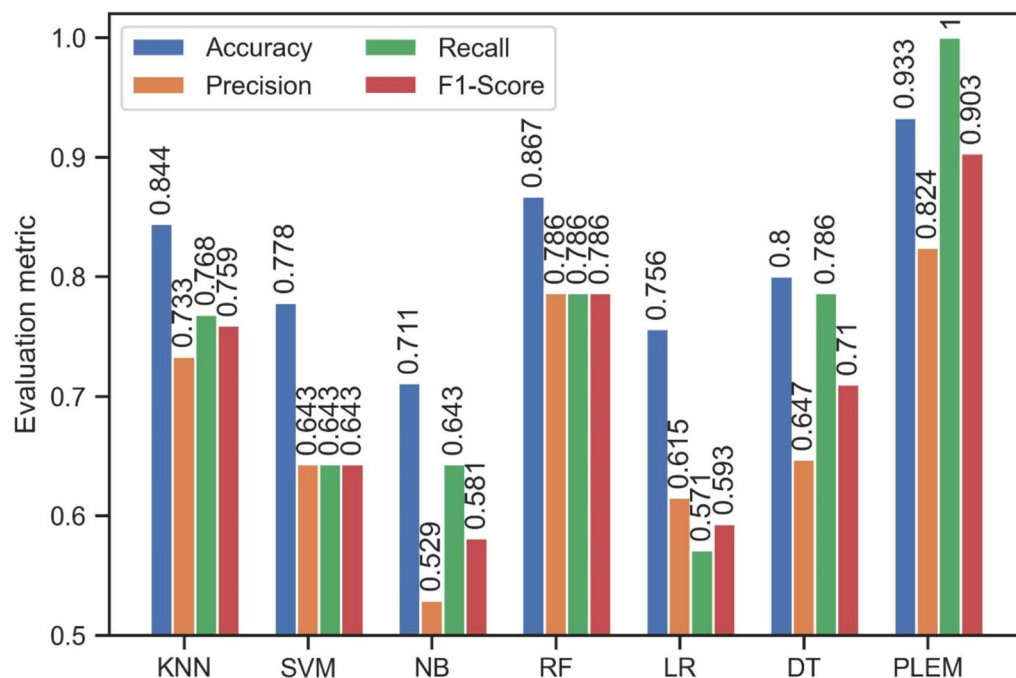


Fig. 14. Comparison between the PLEM method and its constituent models.

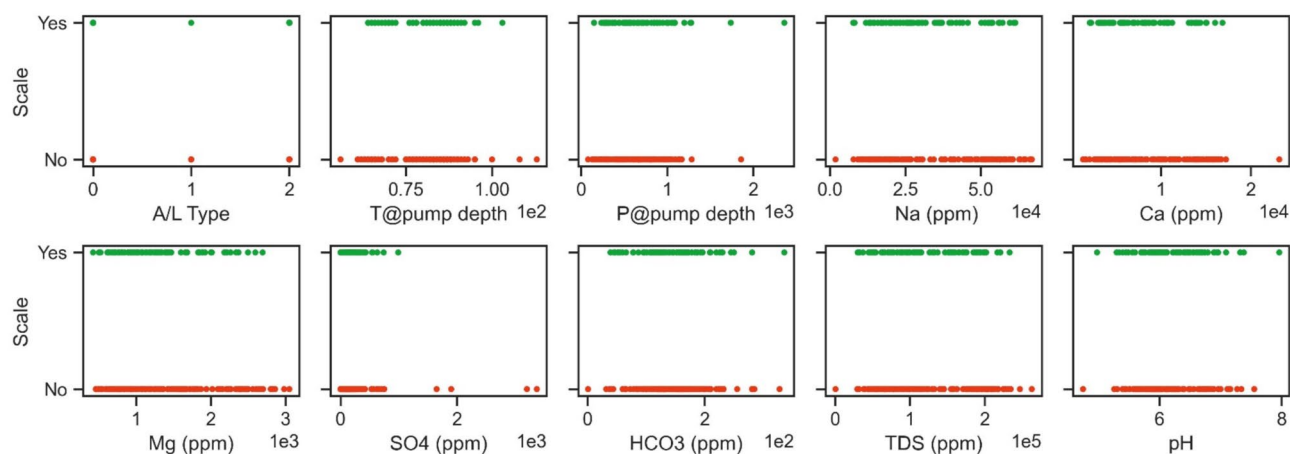


Fig. 15. Input–output analysis for extracting scale-forming rules.

among ions, pH, pressure, and temperature. As a consequence, we cannot develop specific rules based solely on thresholds for influential factors; instead, data-driven models are needed to uncover hidden relationships between the inputs and the output (scale formation). Extracting these rules is not easy without the help of machine learning models.

To investigate the accuracy of the PLEM as the best-performing model on a real case, a water analysis report from a random wellhead sample was taken from a well in the Shuaiba field. Figure 16 provides the results of the water analysis on the sample. The target well was completed with a sucker rod pump operating under a pressure of ~290 psi at the pump depth. By processing the data presented in Fig. 16, the model predicted that the well had a high probability of forming inorganic scale.

The well was gone under investigation and the following observations were recorded:

- Frequent pump failures were reported in the well, leading to three pump replacements.
- The pumps suffered from a short run-life, especially the last two pumps had a life span of 150 and 116 days.
- After a thorough root-cause analysis, it was revealed that the well suffered from scale formation, which was in agreement with the model's prediction.

Page

:

Work Order

:

Client

:

Project

:

Full Water Analyses

Analytical Results

Matrix: WATER

Sub-Matrix: WATER

Client Sample ID

Client sampling date / time

18-May-2022 00:00

Compound	CAS#	LOR	Unit	
INORGANIC NONMETALLIC PARAMETERS				
Chloride	16887-00-6	1	mg/L	12300
Silica	7631-86-9	0.1	mg/L	9.56
Sulfide as S2-	18496-25-8	0.005	mg/L	<0.005
Unionized Hydrogen Slfide	----	0.1	mg/L	<0.1
MAJOR CATIONS & ANIONS				
^ Bicarbonate as HCO3-	----	1	mg/L	103
MAJOR CATIONS & ANIONS - TOTAL				
Calcium	7440-70-2	1.00	mg/L	1060
Mgnesium	7439-95-4	1.00	mg/L	328
Ptassium	7440-09-7	1.00	mg/L	129
Sodium	7440-23-5	1.00	mg/L	8260
^ Sulfate as SO4 2-	14808-79-8	1.00	mg/L	3440
METALS				
^ Ferric Iron	----	0.05	mg/L	11.8
Ferrous Iron	----	0.05	mg/L	<0.05
METALS - TOTAL				
Barium	7440-39-3	0.100	mg/L	<0.100
Iron	7439-89-6	0.050	mg/L	11.8
Manganese	7440-09-7	0.010	mg/L	0.459
Strontium	7440-23-5	0.010	mg/L	20.4
NUTRIENTS				
^ Nitrate as NO3	14797-55-8	0.040	mg/L	2.21
^ Nitrate as NO2	14797-65-0	0.020	mg/L	<0.02
PHYSICAL PARAMETERTS				
Carbon Dioxide as CO2 (Dissolved)	----	1	mg/L	73
pH Value	----	0.01	pH Unit	7.72
Specific Gravity	----	-	-	1.01968
Temperature	----	0.5	°C	22.3
Total Dissolved Solids @180°C	----	5	mg/L	25800
^ Total Suspended Solids	----	5	mg/L	32

Fig. 16. Full water analysis report for a new sample.

Table 4 provides a description of the last three pumps used in the candidate well, their run life, reported failures, root causes, and observations extracted from well intervention history report. As can be seen, all three pumps suffered from scale formation, and the issue worsened over time, leading to shorter run-life after each pump replacement. The developed tool is being planned for deployment across more wells to prepare it for future use.

Reservoir	Run-life (days)	Reported failure	Root cause	Observations
Shuaiba	633	Downhole issue	Inorganic scale	Deposited solids and scale in sucker rod pumps
Shuaiba	150	Downhole issue	Inorganic scale	Inorganic scale in sucker rod pumps
Shuaiba	116	Downhole issue	Inorganic scale	A sucker rod pump fully covered with inorganic scale

Table 4. Description of the last three pumps used in the candidate well.

Real-life applications

The match between the what was reported in the well intervention history and the PLEM method's prediction indicates the reliability of the proposed methodology for predicting inorganic scale formation. The findings of this study can be used to develop data-driven models for real-life applications, such as:

- Early detection of scale formation and take preventive actions, such as timely replacement of downhole pumps before severe scale buildup.
- Optimization of operational conditions, such as water composition, pressure, or pH, to prevent scale build up. The model can serve as a surrogate to predict and prevent scale formation by adjusting controllable operational parameters.
- By timely predicting the scale formation the company can implement chemical injection in high-potential wells for scale formation to reduce costs related to scale remediation and pump failures.
- Petroleum, geothermal energy, and water treatment projects can take benefit from the developed data-driven model to reduce the costs and increase operational safety.

Conclusions

A dataset of 224 field samples of scale formation gathered from Natih and Shuaiba reservoirs in Oman was used to develop data-driven models for predicting scale formation. Six machine learning models with different learning algorithms were developed, and their predictions were integrated using the PLEM method. Results showed that three of the machine learning models (RF, KNN, and DT) could predict scale formation with an F1-score higher than 70%. However, by integrating the outputs of the individual models using the PLEM method the F1-score improved to 90.3%. This indicates the strong ability of the PLEM method to improve the predictions significantly. The reason for the inferior performance of LR and NB is that they assume conditional independence among input variables, whereas factors influencing scale formation are correlated. Moreover, the SVM attempts to find a universal classifier, which is a challenging task in problems where classes change with small perturbations in inputs.

The finding of this study show that ensemble learning with a power-law integration approach is a reliable technique to leverage strengths of individual data-driven models with different learning strategies to improve the prediction of scale formation. The result of input–output analysis showed that scale formation does not result from the high or low concentration of a single ion but is a complex phenomenon that requires advanced models to uncover the relationships between inputs and outputs. Moreover, tree-based models and the KNN model performed better than other individual experts, showcasing their suitability for modeling highly sensitive relationships between input and output variables.

In practice, the PLEM method can be used to predict scale formation before the situation becomes severe, allowing to take preventive actions, such as timely pump replacement and avoiding unexpected production blockage. Furthermore, the predictive tool can help identify the root cause of scale formation by altering input parameters and inspecting whether scale forms.

The limitation of this work is that it does not use time series data, such as fluid rates, pressure over time, temperature over time, ion concentrations over time, pH over time, and TDS over time, to forecast scale formation ahead of time. It also cannot identify the risk of scale formation at early stages before scale formation starts. Moreover, this study only considered inorganic scale. It is recommended to extend the dataset and models to include all types of solids, such as organic and inorganic scale, wax, asphaltene, sand, and metals. The data-driven predictive models used in this study can only determine whether inorganic scale will form in oil and gas wells; they cannot predict the trend, specific scale formation amount, or type of scale formation. Additionally, the models do not account for the influence of the presence of gas and oil phases on the trend of scale formation. It is recommended to gather a more comprehensive dataset of scale formation, including the amount, type, and root cause of scale, gas analysis results, and oil composition, to enable the predictive model to overcome the mentioned limitations. For example, if the model can identify the root cause and type of scale, it can suggest tailored solutions. In addition, gas analysis results can enrich the inputs of the model with values such as CO₂ concentration, H₂S concentration, and dissolved oxygen, as these parameters play an important role in scale formation.

Data availability

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

Received: 25 March 2025; Accepted: 30 May 2025

Published online: 01 July 2025

References

- Green, D. W. & Willhite, G. P. *Enhanced oil recovery*. (Society of Petroleum Engineers, 1998).
- Johnston, C. & Sutherland, L. The influence of turbulence (or hydrodynamic effects) on strontium sulphate scale formation and inhibitor performance. in *SPE International Oilfield Scale Conference and Exhibition* (2014). <https://doi.org/10.2118/SPE-16976-MS>.
- Xiong, H., Kim, C. & Fu, J. A data-driven approach to forecasting production with applications to multiple shale plays. In *Paper presented at the SPE Improved Oil Recovery Conference, Virtual* (2020). <https://doi.org/10.2118/200365-MS>.
- Al-Fakih, A., Koeshidayatullah, A., Mukerji, T., Al-Azani, S. & Kaka, S. I. Well log data generation and imputation using sequence based generative adversarial networks. *Sci. Rep.* **15**, 11000 (2025).
- Al-Fakih, A., Koeshidayatullah, A., Mukerji, T. & Kaka, S. Enhanced anomaly detection in well log data through the application of ensemble GANs. *ArXiv abs/2411.1*, (2024).
- Al-Fakih, A., Kaka, S. I. & Koeshidayatullah, A. I. Reservoir property prediction in the north sea using machine learning. *IEEE Access* **11**, 140148–140160 (2023).
- Patidar, A. K., Singh, S., Anand, S. & Kumar, P. Enhancing PVT property predictions for black oil reservoirs through the application of supervised machine learning techniques. *Geoenergy Sci. Eng.* **243**, 213307 (2024).
- Koray, A.-M., Bui, D., Kubi, E. A., Ampomah, W. & Amosu, A. Machine learning based reservoir characterization and numerical modeling from integrated well log and core data. *Geoenergy Sci. Eng.* **243**, 213296 (2024).
- Wu, C. et al. Determination of gas-oil minimum miscibility pressure for impure CO₂ through optimized machine learning models. *Geoenergy Sci. Eng.* **242**, 213216 (2024).
- Wang, S. & Chen, S. Insights to fracture stimulation design in unconventional reservoirs based on machine learning modeling. *J. Pet. Sci. Eng.* **174**, 682–695 (2019).
- Zhang, C. et al. Formational stages of natural fractures revealed by U-Pb dating and C-O-Sr-Nd isotopes of dolomites in the Ediacaran Dengying Formation, Sichuan Basin, southwest China. *Geol. Soc. Am. Bull.* **136**(11–12), 4671–4688. <https://doi.org/10.1130/B37360.1> (2024).
- Moghadasi, J. et al. scale formation in iranian oil reservoir and production equipment during water injection. *Int. Symp. Oilfield Scale SPE-80406-MS* <https://doi.org/10.2118/80406-MS> (2003).
- Hajirezaie, S., Wu, X. & Peters, C. A. Scale formation in porous media and its impact on reservoir performance during water flooding. *J. Nat. Gas Sci. Eng.* **39**, 188–202 (2017).
- Al-Hajri, N. M., Al-Ghamdi, A. & Mahmoud, M. Scale-prediction/inhibition design using machine-learning techniques and probabilistic approach. *SPE Prod. Oper.* **35**, 0987–1009 (2020).
- Ahmadi, M. & Chen, Z. Machine learning-based models for predicting permeability impairment due to scale deposition. *J. Pet. Explor. Prod. Technol.* **10**, 2873–2884 (2020).
- Xu, H., Jiao, Z., Zhang, Z., Huffman, M. & Wang, Q. Prediction of methane hydrate formation conditions in salt water using machine learning algorithms. *Comput. Chem. Eng.* **151**, 107358 (2021).
- Yousefzadeh, R., Bemani, A., Kazemi, A. & Ahmadi, M. An insight into the prediction of scale precipitation in harsh conditions using different machine learning algorithms. *SPE Prod. Oper.* <https://doi.org/10.2118/212846-PA> (2022).
- Zhang, C. et al. Mechanism for the formation of natural fractures and their effects on shale oil accumulation in Junggar Basin, NW China. *Int. J. Coal Geol.* **254**, 103973. <https://doi.org/10.1016/j.coal.2022.103973> (2022).
- Yu, H., Zhao, Z., Taleghani, A. D., Lian, Z. & Zhang, Q. Modeling thermal-induced wellhead growth through the lifecycle of a well. *Geoenergy Sci. Eng.* **241**, 213098. <https://doi.org/10.1016/j.geoen.2024.213098> (2024).
- Zhang, C. et al. Major factors controlling fracture development in the Middle Permian Lucaogou Formation tight oil reservoir, Junggar Basin, NW China. *J. Asian Earth Sci.* **146**, 279–295. <https://doi.org/10.1016/j.jseas.2017.04.032> (2017).
- Wang, J., Chen, Y. & Zou, Q. Inferring gene regulatory network from single-cell transcriptomes with graph autoencoder model. *PLOS Genet.* **19**(9), e1010942. <https://doi.org/10.1371/journal.pgen.1010942> (2023).
- Murphy, K. P. Naive bayes classifiers. *Univ. Br. Columbia* **18**, 1–8 (2006).
- Theil, H. A multinomial extension of the linear logit model. *Int. Econ. Rev. (Philadelphia)* **10**, 251–259 (1969).
- Böhning, D. Multinomial logistic regression algorithm. *Ann. Inst. Stat. Math.* **44**, 197–200 (1992).
- Fix, E. & Hodges, J. L. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *Int. Stat. Rev. Int. Stat.* **57**, 238–247 (1989).
- Altman, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **46**, 175–185 (1992).
- Piryonesi, S. M. & El-Diraby, T. E. Role of data analytics in infrastructure asset management: Overcoming data size and quality problems. *J. Transp. Eng.* <https://doi.org/10.1061/JPEODX.0000175> (2020).
- Hastie, T., Tibshirani, R. & Friedman, J. H. *The elements of statistical learning: data mining, inference, and prediction* (Springer New York, 2009). <https://doi.org/10.1007/978-0-387-84858-7>.
- Patel, H. H. & Prajapati, P. Study and analysis of decision tree based classification algorithms. *Int. J. Comput. Sci. Eng.* **6**, 74–78 (2018).
- Pal, M. & Mather, P. M. Decision tree based classification of remotely sensed data. In *22nd Asian Conference on Remote Sensing* (2001).
- Livingston, F. Implementation of Breiman's random forest machine learning algorithm. *ECE591Q Mach. Learn J. Pap.* **1**(1), 13 (2005).
- Liaw, A. & Wiener, M. Classification and regression by randomforest. *R news* **2**, 18–22 (2002).
- Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995).
- Morariu, D., Vintan, L. & Tres, V. Feature selection methods for an improved SVM classifier. In *Proc. of the 3rd International Conference of Intelligent Systems (ICIS'06)* (2006). doi:<https://doi.org/10.5281/zenodo.1332490>.
- Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach. Learn.* **46**, 389–422 (2002).
- Prajapati, G. L. & Patle, A. On performing classification using SVM with radial basis and polynomial kernel functions. In *3rd International Conference on Emerging Trends in Engineering and Technology. IEEE* 512–515 (2010). <https://doi.org/10.1109/ICETE.2010.134>.
- Takhanov, R. On the speed of uniform convergence in Mercer's theorem. *J. Math. Anal. Appl.* **518**, 126718 (2023).
- Burges, C. J., Smola, A. J. & Scholkopf, B. Advances in kernel methods. In *Support Vector Learning* (The MIT Press, 1999).
- Lin, S. W., Lee, Z. J., Chen, S. C. & Tseng, T. Y. Parameter determination of support vector machine and feature selection using simulated annealing approach. *Appl. Soft Comput.* **8**, 1505–1512 (2008).

Author contributions

M.H: Resources, Data Curation, Investigation, Writing – reviewing, Formal analysis A.K: Conceptualization, Supervision, Writing—Original Draft, Validation, Formal analysis R.Y: Writing—Original Draft, Coding, Investigation, Visualization, Formal analysis.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to A.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025