Advances in
Geo-Energy Research

Vol. 5, No. 3, p. 297-317, 2021

Original article

# Enhanced oil recovery by nanoparticles flooding: From numerical modeling improvement to machine learning prediction

Budoor Alwated[1], Mohamed F. El-Amin[1,2]*

[1]College of Engineering, Effat University, Jeddah 21478, Saudi Arabia
[2]Mathematics Department, Faculty of Science, Aswan University, Aswan 81528, Egypt

**Keywords:**
Enhanced oil recovery
nanoparticles
machine learning
random forest
artificial neural networks
decision tree
gradient boosting regression

**Cited as:**
Alwated, B., El-Amin, M. F. Enhanced oil recovery by nanoparticles flooding: From numerical modeling improvement to machine learning prediction. Advances in Geo-Energy Research, 2021, 5(3): 297-317, doi: 10.46690/ager.2021.03.06

**Abstract:**
Nowadays, enhanced oil recovery using nanoparticles is considered an innovative approach to increase oil production. This paper focuses on predicting nanoparticles transport in porous media using machine learning techniques including random forest, gradient boosting regression, decision tree, and artificial neural networks. Due to the lack of data on nanoparticles transport in porous media, this work generates artificial datasets using a numerical model that are validated against experimental data from the literature. Six experiments with different nanoparticles types with various physical features are selected to validate the numerical model. Therefore, the researchers produce six datasets from the experiments and create an additional dataset by combining all other datasets. Also, data preprocessing, correlation, and features importance methods are investigated using the Scikit-learn library. Moreover, hyperparameters tuning are optimized using the GridSearchCV algorithm. The performance of predictive models is evaluated using the mean absolute error, the R-squared correlation, the mean squared error, and the root mean squared error. The results show that the decision tree model has the best performance and highest accuracy in one of the datasets. On the other hand, the random forest model has the lowest root mean squared error and highest R-squared values in the rest of the datasets, including the combined dataset.

## 1. Introduction

In the past few decades, there was a growth in the worldwide energy demand, including oil, natural gas, and coal. There is a continuous escalation in the world energy demand and constant progress in technology development for exploring new reservoirs or improving the techniques used in enhanced oil recovery (EOR) (Kong and Ohadi, 2010). Nanoparticles are used in EOR due to the difficulty of finding a new source of hydrocarbon as most of the oil fields have 60 to 70% of hydrocarbon is not extracted (Li, 2016). Silica is a form of nanoparticle that is environmentally friendly, has a natural structure similar to sandstone oil reservoirs, and can increase oil production by increasing the recovery factor. The nature of nanoparticles allows the increase in the surface area and can affect the molecules reaction. Although the small size of nanoparticles facilitates its transfer in porous media, some nanoparticles can be attached to rocks by surface filtration, straining, and physicochemical filtration, which can severely lower the porosity and permeability of the porous medium. Hence, various factors can affect the nanoparticles transportability in the pore throats, such as nanofluid concentration, injection rate, slug size, and particle size. Traditionally, numerical simulations predict the hydrocarbon movement in porous media, which has many uncertainties and complex numerical techniques. Nowadays, machine learning is intensively used in many disciplines, including petroleum engineering.

This paper uses numerical modeling of nanoparticles in porous media based on filtration theory to generate datasets for machine learning techniques to forecast the nanoparticles transport. Due to the lack of published datasets about nanoparticles transport in porous media as most datasets from

Yandy Scientific Press

*Corresponding author.
*E-mail address*: bualwated@effat.edu.sa (B. Alwated); momousa@effatuniversity.edu.sa (M. F. El-Amin).

Received April 4, 2021; revised April 28, 2021; accepted May 1, 2021; available online May 10, 2021.

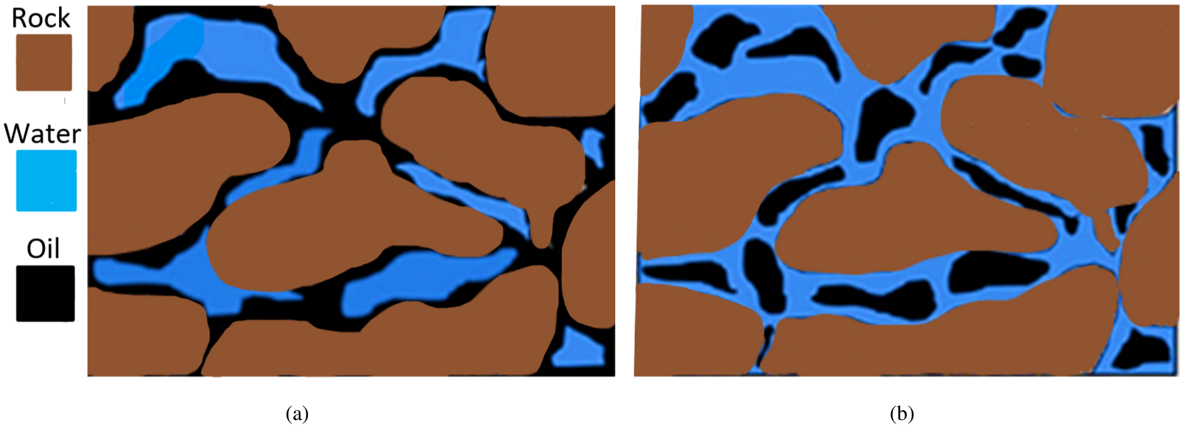


**Fig. 1.** (a) Oil-wet and (b) Water-wet rocks.

petroleum engineering companies are confidential. This study generates an artificial dataset based on mathematical continuum models that are validated against experimental results from the literature. The artificial dataset is used as input to train machine learning models for nanoparticles concentration predictions. The machine learning techniques that are used in the prediction include decision tree (DT), random forest (RF), gradient boosting regression (GBR), and artificial neural networks (ANN).

The paper structure is as follows. Section 2 presents background information about EOR using nanoparticles. Section 3 discusses the research methodology. The mathematical models of nanoparticles transport in porous media are presented in section 4, and section 5 covers the machine learning modeling. Section 6 discusses the evaluation metrics, and section 7 demonstrates our results and discussions. Finally, section 8 presents the conclusion.

## 2. Background

### 2.1 EOR

EOR is a term referred to as the technique of raising the hydrocarbon amount produced from a well Alvarado and Manrique (2010a). EOR allows altering the hydrocarbons' actual properties, making it different from the secondary recovery methods where water flooding and gas injection are used for pushing the oil through the well (Van Poollen, 1980). EOR has three different techniques: thermal recovery, chemical injection, and gas injection (Alvarado and Manrique, 2010b). The selection of the EOR method depends on information obtained from the reservoir evaluation phase, including reservoir characterization, screening, scoping, and reservoir modeling and simulation. The most traditional EOR types are thermal, chemical, and gas injection recovery. The thermal recovery method focuses on injecting hot steam into an injection well. The injected hot steam would reduce the oil viscosity to improve flow in the reservoir (Alvarado and Manrique, 2010b). The chemical injection method allows freeing trapped oil within the reservoir. In the chemical injection method, chemical substances such as polymers that are long-chained molecules are injected into the subsurface reservoir to increase waterflooding efficiency and boost the effectiveness of surfactants (Manning, 1983). Finally, the gas injection method focuses on injecting natural gas, nitrogen, or carbon dioxide into the reservoir. Injecting gas can mix with or dissolve within the oil, reducing the oil viscosity and increasing the flow, which would enhance the extraction (Manning, 1983).

### 2.2 Nanoparticles for EOR

Nanotechnology offers an innovative approach to govern petroleum recovery processes. Nanoparticles help improve the geo-mechanism of reservoirs due to modification in reservoir properties such as reactivity of chemicals, active surfaces, and a higher specific area (Kazemzadeh, 2019). Nanoparticles can boost hydrocarbon recovery by modifying various rock and fluid properties. Rock properties include conductivity modification, rock, and oil interaction, and wettability alteration. Fluid properties include altering the fluid viscosity, reducing the interfacial tension (IFT), and stabilizing the emulsion, leading to the additional recovery of more than 20% compared to conventional chemical surfactant-polymer flooding. The high temperature inside the reservoir decreases the efficiency of surfactant polymer flooding. However, the nanofluid mixture has a stable behavior at increased temperatures, making it an effective solution for EOR techniques for high temperatures (Lashari and Ganat, 2020). Nanoparticles can alter the wettability of the rock from oil-wet to water-wet. In oil-wet rock, oil tends to stick on the walls of the porous media, and the waterflooding technique is not productive in these rocks because oil droplets cannot move easily between the pores of the matrix. However, the injection of nanopacticles can change the wettability of rock to water-wet. In water-wet rock, water tends to imbibe to the rock surface, and by water flooding, oil moves toward the production well; thus, oil recovery increases. Fig. 1. illustrates oil-wet and water-wet rocks.

### 2.3 Nanoparticles transport in porous media

The advancement in nanotechnology allows nanoparticles to be injected into subsurface environments to transport them into hydrocarbon reservoirs. Nanoparticles used in EOR process designed to travel far in the subsurface reservoirs. Therefore, understanding nanoparticles' properties are essential to identify their mobility in the subsurface. Many experimental studies investigated nanoparticles transport in porous media and the retention mechanisms. Based on the application in industries, many nanomaterials are tested for their mobility in porous media. The porous media used for the laboratory experiments of nanoparticles transport are columns packed with sand grains or glass-beads. The main finding of such experiments showed that nanomaterials retention in those columns relies on the material properties in terms of size, shape, and surface. Nanoparticles of similar types have effluent histories affected by the flow velocity, the gain surface area, and chemical additives in the solution (Zhang, 2012). Ju and Fan (2009) conducted a theoretical and experimental study. They stated that the concentration of silica nanoparticles injected in core flooding for EOR was 2.0% to 3.0%. When nanoparticles get adsorb onto a rock surface, they can change their wettability and enhance recovery. Maghzi et al. (2014) discussed the dispersion of silica nanoparticles in Polyacrylamide. The experiment conducted investigated the rheological properties of Polyacrylamide and silica nanoparticles. They found out an improvement in the fluid viscosity and the polymers' pseudoplastic behavior. A 0.1 wt% addition of silica nanoparticles, leads to 10% additional recovery. Wasan and Nikolov (2003) investigated the spreading behavior of nanofluids mixed with surfactant on a solid surface. Ju and Fan (2009) used experimental and numerical approaches to observe the wettability modification that was caused by lipophobic and hydrophilic polysilicon nanoparticles. They found out that as the nanoparticles absorbed on rock grain, the wettability changes. Ogolo et al. (2012) conducted experiments that revealed that Aluminium oxide and Silicon oxide are effective EOR agents. When combined with distilled water and brine as dispersion agents, aluminum oxide nanoparticles increases the oil recovery. Silicon oxide alters rock wettability and the interfacial tension between oil and water, whereas aluminum oxide decreases oil viscosity. Youssif et al. (2018) studied the consequence of the injection of silica nanoparticles of oil recovery. They used in their experiment silica nanofluid of different concentrations ranging between 0.01 wt% and 0.5 wt%. They found out that as the nanoparticles' concentration increases until it reaches optimum, the recovery factor rises. Khalilinezhad et al. (2016) used the multiphase simulator University of Texas Chemical Compositional Simulator to study the effect of nanoparticles on the flow behavior of injected flood in porous media. They also used a polymer shear thinning model to validate the adsorption of nanoparticles on sandstone surface area and rheological results. They discovered that incorporating nanoparticles into polymers decreases sandstone retention and adsorption, while rheological activity is shear-based (Khalilinezhad et al., 2017). Jeong and Kim (2009) studied the transport of copper oxide (CuO) nanoparticles in two-dimensional porous media. They examined the aggregation of copper oxide nanoparticles in pores. They discovered that nanoparticles' accumulation and deposition are affected by the nanoparticle's flow velocity and surfactant content. Moreover, they found out that the flow velocity affects the density such that the flow velocity declines as the number of aggregates enlarge. Shaniv et al. (2021) examined the polystyrene nanoparticles transport in fully water-saturated soil. They highlighted that particle size and surface texture can affect polystyrene mobility in the soil. Abdelfatah et al. (2017) used the combination of Darcy's equation and convection-diffusion equation to develop a mathematical model to investigate nanoparticles transport, interaction, and their behavior with the fluid. They stated that various mechanisms play a significant role in nanoparticles transport, such as permeability, injection rate, concentration, and size. El-Amin et al. (2013, 2015) presented a numerical simulation and developed a mathematical model for nanoparticles water suspensions in two-phase flow in porous media considering capillary forces and Brownian diffusion. Through their studies, they monitored the effect of injecting nanoparticles on the properties of solid and fluid. El-Amin et al. (2012a, 2012b) developed mathematical models for nanoparticles transport in porous media considering the capillary forces, Brownian diffusion, and buoyancy.

## 2.4 Machine learning in petroleum industry and EOR

Artificial intelligence (AI) is a discipline that employs complex algorithms and networking tools to solve multidimensional problems by imitating human brainpower allowing machines to perform computational tasks (Yousef et al., 2020). Machine learning is an area of AI that deals with algorithm design and development to enable computers to use empirical data and learn behaviors or patterns (Zhang et al., 2020; Pirizadeh et al., 2021). The machine learning models find relations between inputs and outputs. Machine learning can solve regression, clustering, filtering, classification, and forecasting problems. Examples of machine learning techniques are ANNs, RF, GBR, and DT.

Traditional mathematical models of reservoir fields are used to simulate oil recovery processing. However, those models are considered complex and have high computation time (Daribayev et al., 2020). Thus, this leads to a longer time to predict oil recovery. Parallel algorithms can be effective in solving such problems considering the heterogeneity of computing systems. Machine learning methods may also solve these problems. The model is trained using historical oil field data and synthetic data from surrogate models based on injection and production wells. Irfan and Shafie (2021) used deep learning and artificial neural networks to solve and simulate fluid flow problems. You et al. (2020) proposed a method for optimizing oil recovery, $CO_2$ storage volume, and reducing greenhouse gas emissions by integrating artificial neural networks and multi-objective optimizers. Van den Doel et al. (2020) presented a method to monitor the subsurface temperature remotely using low-frequency radar pulses. They used feed-forward neural networks to extract the modulation and measure the down hole temperatures. Yousef et al. (2020) introduced a top-down model for a carbonate reservoir in the Middle East that uses neural networks to predict reservoir output three months ahead. Esfe et al. (2018) proposed an artificial neural network model with two hidden

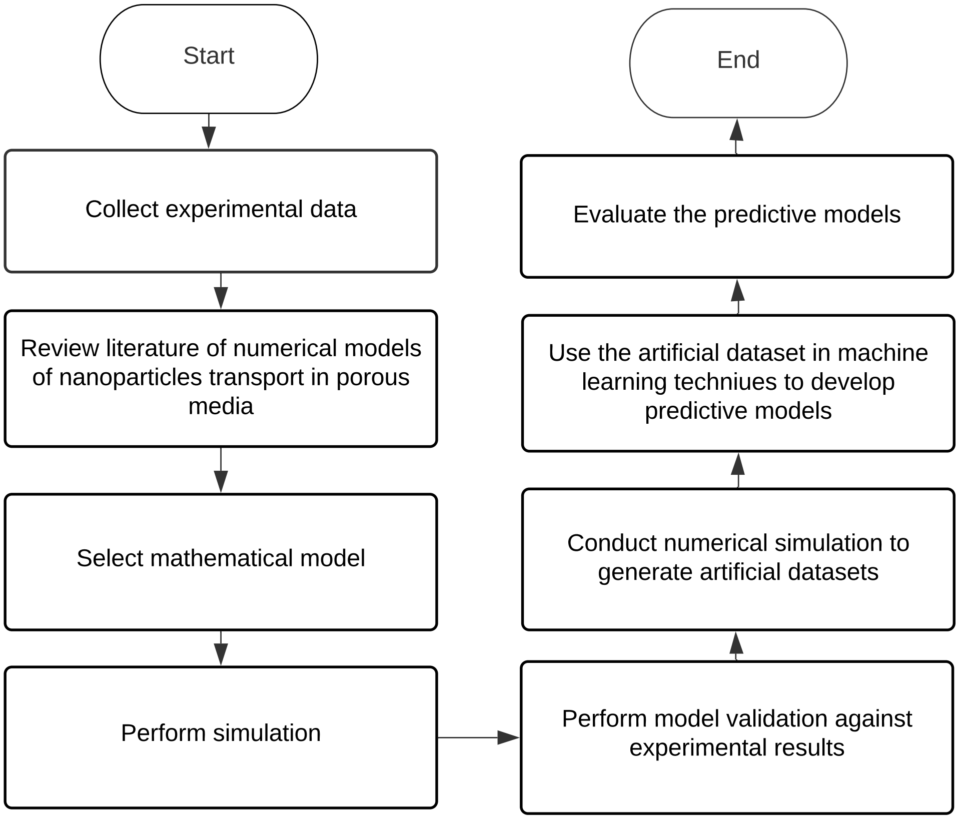

Start
Collect experimental data
Review literature of numerical models of nanoparticles transport in porous media
Select mathematical model
Perform simulation
Perform model validation against experimental results
Conduct numerical simulation to generate artificial datasets
Use the artificial dataset in machine learning techniues to develop predictive models
Evaluate the predictive models
End

**Fig. 2.** Research methodology flowchart.

layers for CuO/EG nanofluid dynamic viscosity prediction over a temperature range of 27.5-50 °C. They evaluated the developed model using mean relative error and the $R^2$ value which was 0.0175 for mean relative error 0.999 and 0.0175 for $R^2$. Changdar et al. (2020) used deep learning to develop a modern approach of data driven viscosity prediction model for water-based nanofluids. The feature selected were nanoparticles density, size, volume fraction, temperature, and viscosity of the base fluid. They found out that their proposed model outperforms other traditional computer-aided models, theoretical and empirical correlations by 99% accuracy.

Subasi et al. (2020) developed a machine learning model based on stochastic gradient boosting regression for predicting reservoir permeability based on well log information. Their studies test several machine learning techniques such as random forest, artificial neural networks, K-nearest neighbors (KNN), support vector machine (SVM), and stochastic gradient boosting. They found out that stochastic gradient boosting achieved the highest performance in several evaluation metrics tests, such as accuracy and root mean squared error compared to other tested models. Moreover, El-Amin and Subasi (2019) presented a new power-law scaling velocity related to dimensionless time, and it is a function of characteristic injection velocities. Their work used machine learning techniques such as KNN, SVM, RF, and ANN to forecast the dimensionless oil recovery time based on the oil and rock primary physical data. Zhou et al. (2021) predicted the nanoparticles transport behavior in porous media using the data-driven approach. Their work filled all the missing data in their dataset using random forest combining one-hot encoding. They used the CatBoost technique combined with the synthetic minority oversampling technique to perform the regression for predicting the nanoparticles retention. Their proposed method showed good performance in predicting retention.

## 3. Research methodology

The steps implemented in conducting this research are as follows: first, the published experimental studies of the transport of nanoparticles in porous media are collected. Second, the literature of published numerical models of nanoparticles transport in porous media is reviewed. Third, a mathematical model is selected, and the finite difference method is used to solve it numerically using MATLAB. In the fourth step, A model validation against the experimental data is performed. Once the model is validated, an artificial dataset is generated to be used in machine learning. Finally, the predictive models are evaluated using the performance evaluation metrics such as root mean squared errors. Fig. 2. presents the flow chart that summarizes the method adopted in conducting this research.

The first step in the machine learning process is to acquire a dataset. The second step is the preprocessing phase. This phase focuses on removing noise, segmentation, scaling, and removing whitespace from the dataset. The third phase is about feature selection and extraction. This phase is about reducing the dimensionality and selecting effective features. The fourth phase is selecting proper learning techniques. Phase five is

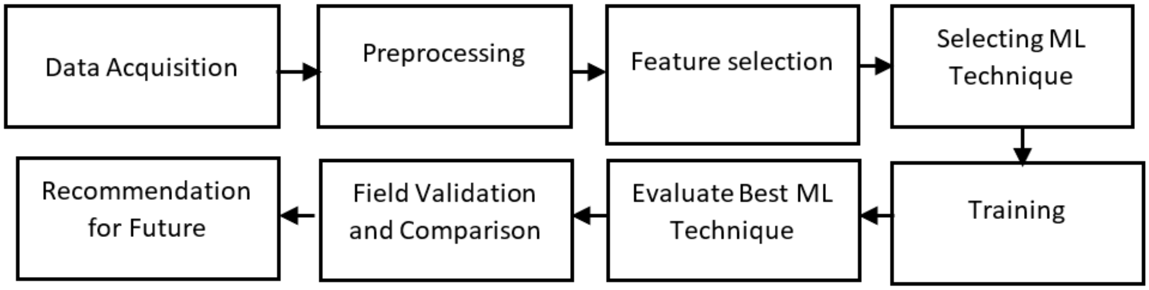

Data Acquisition
Preprocessing
Feature selection
Selecting ML Technique
Training
Evaluate Best ML Technique
Field Validation and Comparison
Recommendation for Future

**Fig. 3.** Flowchart of machine learning phases.

about learning or training using the training dataset. The sixth phase is the evaluation. This phase is concerned with testing the performance of the model on the testing set. Fig. 3 presents the flowchart of machine learning phases.

## 4. Mathematical model of nanoparticles transport

The colloid filtration theory is commonly used to simulate nanoparticles or colloidal particle transport and attachment in water-saturated porous media. The nanoparticles transport model selection is developed based on colloid filtration theory.

### 4.1 Filtration mechanisms

There are three types of filtration mechanism of solid particles during transport in porous media (fine migration in porous media): surface filtration, straining, and physicochemical filtration (McDowell-Boyer et al., 1986). Surface cake filtration occurs when the size of the nanoparticles is greater than the porous media grain. Accordingly, nanoparticles will be unable to penetrate the pores and make a filter cake at the surface of the grains, which can sharply reduce the medium's permeability. Straining filtration occurs when nanoparticles are stuck in some nanopores in the media grain by straining at tiny pore throats in the medium, thus reducing permeability. However, the decline in permeability caused by straining is limited compared to the reduction caused by filter cake. In physical and chemical filtration, when the size of the nanoparticles is smaller than media grains, the nanoparticles will not block any pore throat; however, the nanoparticles are retained because of the physical and chemical interactions between the particle and the medium (Fan, 2018). Most nanoparticles can move through the pore throats in sedimentary rocks without straining due to their small diameters compared to typical pore throats. Thus, physicochemical filtration can occur. This physicochemical filtration can lead to a strong van der walls attraction between nanoparticles and between nanoparticles and the rock surface. Therefore, there are two alternative nanoparticles transport models. One model depends on colloid filtration theory, which is used in this paper. The other model depends on chemical adsorption caused by a change in chemical potential between the fluid and solid phases.

Cushing and Lawler (1998) based the mathematical equation of the transport of nanoparticles in porous media on an advection-dispersion equation with a filtration term inserted to it for the mass balance of particles during transport:

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - v_p\frac{\partial c}{\partial x} \tag{1}$$

$$\frac{\partial c}{\partial t} + \frac{\rho_b}{\phi}\frac{\partial s}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - v_p\frac{\partial c}{\partial x} \tag{2}$$

The second term in Eq. (2) is the filtration term represents the nanoparticles retention:

$$\frac{\rho_b}{\phi}\frac{\partial s}{\partial t} = k_{dep}c \tag{3}$$

where $c$ is the concentration of nanoparticles in the carrier fluid, $s$ is the nanoparticles deposition concentration [mass of nanoparticles/mass of porous medium], $D$ is the dispersion coefficient, $\rho_b$ is the bulk density of the porous medium, $v_p$ is the interstitial velocity, $\phi$ is the medium porosity, and $k_{dep}$ is the particle deposition rate coefficient.

The colloid filtration model highlights that if the dispersion concentration $c$ is greater than zero, the retention concentration of nanoparticles $s$ will continue to increase without an upper bound. Thus, with the continuous injection of dispersion, the effluent concentration of nanoparticles will never reach the injection concentration after a complete breakthrough. Accordingly, the colloid filtration model indicates an irreversible deposition of suspended nanoparticles with a capacity limited to the level when there is no room for filtration to occur. As the deposited nanoparticles increase, the porous medium's permeability gradually reduces (El-Amin et al., 2015). This research ignores the permeability reduction caused by the nanoparticles deposition.

The chemical potential gradient drives the adsorption of nanoparticles with diameters less than 10 nanometers. Park et al. (2009) carried out a series of experiments that revealed that chemical adsorption of nanoparticles could occur on surfaces that obey the Langmuir isotherm at equilibrium, implying that the adsorption mechanism modeled as a balance of adsorption and desorption. Thus, Eq. (2) changes to:

$$\frac{\partial s}{\partial t} = k_a\left(1 - \frac{s}{s_{\max}}\right)c - k_d\frac{s}{s_{\max}} \tag{4}$$

where $k_a$ is the adsorption rate coefficient, and $k_d$ is the desorption rate coefficient. $s_{\max}$ is the adsorption capacity on the substrate surface, In a Langmuir type of adsorption, all adsorptions are reversible. Opposite to the colloid filtration theory, no prediction happens for the permanent attachment of nanoparticles after post flush. Benamar et al. (2007) used the colloid filtration model with the filtration term Eq. (2) to fit the effluent concentration histories of nanoparticles transport through water-saturated columns and found similar results with some experimental findings but not with others. Moreover, Cushing and Lawler (1998) added the maximum retention capacity to colloid filtration model with site blocking:

$$\frac{\rho_b}{\phi}\frac{\partial s}{\partial t} = k_{dep}\ c\ (1-\frac{s}{s_{\max}}) \tag{5}$$

where $s_{\max}$ is the maximum retention capacity.

Liu et al. (2009) presented experimental work with results showing that when they injected a post flush containing no particles into the column, some attached particle was released. The tails of the particle breakthrough curves are longer than those of the tracer curve. Bradford et al. (2002) introduced a detachment term to the colloid filtration model to explain particle detachment from the solid phase:

$$\frac{\rho_b}{\phi}\frac{\partial s}{\partial t} = k_{dep}c - \frac{\rho_b}{\phi}k_{det}s \tag{6}$$

Wang et al. (2008) modified the colloid filtration term by adding both the site-blocking with maximum retention (adsorption) capacity and the detachment term for reversible adsorption:

$$\frac{\rho_b}{\phi}\frac{\partial s}{\partial t} = k_{dep}\ c\left(1-\frac{s}{s_{\max}}\right) - \frac{\rho_b}{\phi}k_{det}s \tag{7}$$

As long as the dispersion concentration is above zero, this filtration model assumes that the nanoparticles surface concentrations $s$ will rise with no upper limit. The particle deposition rate coefficient, $k_{dep}$ is a function of porosity, single-collector contact performance, grain size, and flow rate, which affect the plateau value of the effluent history and the nanoparticles deposition (Zhang, 2012.) Furthermore, the model predicts continuous growth in the surface concentration during the injection of nanoparticles if there is no intrinsic capacity.

### 4.2 The traditional mathematical model

This paper uses the modified colloid filtration theory with two sites model (Zhang, 2012). In the two sites model, the adsorbed nanoparticles on one group of sites can be removed, while the particles on the other group of sites are permanently retained on the solid surface. Each of the two groups of sites has its adsorption capacity. The shape of the surface where the nanoparticle adsorption is one factor that affects nanoparticles removal (Zhang, 2012). The colloid filtration model presented as follow:

$$\frac{\partial c}{\partial t} + \frac{\rho_b}{\phi}\ \frac{\partial s}{\partial t} = D\frac{\partial^2 c}{\partial x^2} - v_p\ \frac{\partial c}{\partial x} \tag{8}$$

$$\frac{\rho_b}{\phi}\frac{\partial s}{\partial t} = \frac{\rho_b}{\phi}\frac{\partial s_1}{\partial t} + \frac{\rho_b}{\phi}\frac{\partial s_2}{\partial t} \tag{9}$$

$$\frac{\rho_b}{\phi}\frac{\partial s_1}{\partial t} = k_{irr}\left(1-\frac{s_1}{s_{1\max}}\right)c \tag{10}$$

$$\frac{\rho_b}{\phi}\frac{\partial s_2}{\partial t} = k_{ra}\left(1-\frac{s_2}{s_{2\max}}\right)c\ -\frac{\rho_b}{\phi}\ k_{rd}s_2 \tag{11}$$

where $s_1$ is the reversible adsorbed nanoparticles concentration on a solid surface, $s_2$ is the irreversibly adsorbed nanoparticles concentration on a solid surface, $s_{1\max}$ is the capacity for irreversible adsorption, $s_{2\max}$ is the capacity for reversible adsorption, $k_{rd}$ is the coefficient of the desorption rate, $k_{ra}$ is the coefficient of the reversible adsorption rate, and $k_{irr}$ is the coefficient of the irreversible adsorption rate.

**The initial conditions**

$$c\,(x,0)\,,\quad s\,(x,0)\,,\qquad 0\ \leq x \leq t \tag{12}$$

**The boundary conditions are:**

$$c(0,t) = \begin{Bmatrix} c_0, & 0 \leq t \leq t_s \\ 0, & t \geqslant 0 \end{Bmatrix} \tag{13}$$

$$\left.\frac{\partial c}{\partial x}\right|_{x=l} = 0,\ t \geq 0$$

where $t_s$ is the nanoparticles dispersion slug size, $l$ is the column length.

## 5. Machine learning modeling

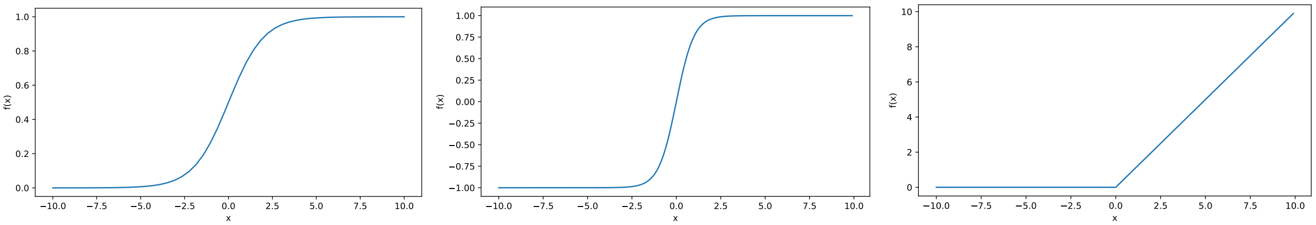
There are various techniques for machine learning, such as random forest, decision tree, artificial neural network, and gradient boosting.

### 5.1 Random forest

Breiman (2001) proposed the random forest algorithm as a method for regression and classification. Random forest is an ensemble learning method in which a large group of decision trees works together as an ensemble to solve a problem. Each decision tree generates a class prediction in the random, and the class with the highest votes used for perdition. The random forest method is derived based on the idea of bagging. In classification problems, the algorithm uses the majority of votes of the class label across trees in the ensemble for the prediction. While in regression problems, random forest calculates the prediction average between trees; it merges several randomized decision trees and aggregates their predictions by averaging El-Amin and Subasi (2020a). Random forest increases the forecasting accuracy and reducing variance by averaging several noisy and unbiased trees. A random forest with a total number of trees has the following variance:

$$\rho\sigma^2 + \frac{1-\rho}{M}\sigma^2 \tag{14}$$

where $\sigma^2$ is the variance of an individual tree, $\rho$ is the correlation between the trees, and $M$ is the total number of trees in the ensemble.



(a) sigmoid (b) tanh (c) ReLu

**Fig. 4.** Artificial neural networks activation functions.

## 5.2 Artificial neural networks

Artificial neural networks are an effective method for identifying and classifying complex patterns. ANN is a conceptual model inspired by biological neurons and simulates the human brain based on predicting patterns (Álvarez del Castillo et al., 2012). An ANN model is composed of neurons, which are the basic processing units. The neural network models have three components: the learning algorithm, the network architecture, and the transfer function (Lippmann, 1987). ANN is a set of nodes known as neurons, weighted connections between these neurons that can be adjusted during the network's learning process, and an activation function that determines each node's output value based on its input values.

In neural network model training, the main parameters used to optimize the learning process are the learning rate, momentum, and minimal error. The learning rate value range between 0 and 1, and it specifies how fast the learning process is performed. The momentum is used to smooth out the optimization process by using a fraction of the last weight change and adding it to the new weight change. The minimal error is a stop criterion for the learning process. Computing a new weight for a connection can be calculated as follows:

$$W = lm\varepsilon W_p \tag{15}$$

where $W$ is the new weight change, $l$ is the learning rate, $m$ is the momentum, $\varepsilon$ is the minimal error, and $W_p$ is the weight change of the previous cycle. Furthermore, artificial neurons calculate the weighted sum of inputs wi and add a bias term $w_0$

$$y = \sum_{i=1}^{n} w_i x_i + w_0 \tag{16}$$

where $w_i$ is the weight, $x_i$ is the input features, and $w_0$ is the bias. If $y >$ threshold, $y$ will be activated, and if $y <$ threshold, $y$ will not be activated. The three common activation functions are:

**Sigmoid activation function (sigmoid)**

The values below 0 drop off in the sigmoid function, and the values above 0 escalated to 1.

$$f(x) = \frac{1}{1+e^{-x}} \tag{17}$$

**Hyperbolic tangent activation function (tanh)**

The hyperbolic tangent function is an activation function that is a smoother and zero-centered function whose range lies between -1 to 1. Thus the output of the tanh function is as follows:

$$f(x) = \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right) \tag{18}$$

**Rectified linear unit activation function (ReLu)**

ReLu is an activation function was proposed by Nair and Hinton (2010) that works by thresholding values at 0 with the function:

$$c(0,t) = \max(0,x) = \left\{ \begin{array}{ll} x_i & \text{if } \ \mathrm{x_i} \geq 0 \\ 0 & \text{if } \ \mathrm{x_i} < 0 \end{array} \right\} \tag{19}$$

ReLu function gives an output $x$ if $x$ is positive and 0 otherwise. It is linear in the positive dimension when $x \geq 0$, but zero in the negative dimension when $x < 0$. Fig. 4 presents the three activation functions.

The models typically have three layers: the input layers, the hidden layers, and the output layers. The input layers connect to the hidden layers, which process the data using weighted connections (Hansen and Salamon, 1990). Hence the input layer assigns weight to the input data and calculates the prediction at the output nodes. Each neuron in the hidden layer communicates with all neurons in the output layer (Sahli, 2020). The tuning of weights between layers affects the network's efficiency (Mohaghegh and Ameri, 1995).

Moreover, the network learns to send training examples to the network one by one (El-Amin and Subasi, 2020b). As a result, ANN's predictive ability grows. The activation mechanism for the output layer is 'pure linear'. Furthermore, in a multi-layer context, neural networks are viewed as a combination of regression and multivariate techniques.

## 5.3 Gradient boosting regression

Gradient boosting regression is a machine learning method for building predictive models. It creates an ensemble of shallow trees in sequence, with each tree learning and improving on the previous one. The advantage of using an ensemble tree is that the averaging can minimize the variance. The GBR model minimizes the loss function by growing trees sequentially and updating the weight of the training data distribution. It decreases model bias and variance through forwarding stage-wise modeling and averaging (Zhang and Haghani, 2015).

**Table 1.** Performance evaluation metrics.

| | |
|---|---|
| Mean absolute error (MAE) | MAE= $\frac{1}{n}\sum_{i=1}^{n}\lvert\text{actual}_i-\text{predicted}_i\rvert$ |
| Mean squared error (MSE) | MSE= $\frac{1}{n}\sum_{i=1}^{n}(\text{actual}_i-\text{predicted}_i)^2$ |
| Root mean squared error (RMSE) | RMSE= $\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\text{actual}_i-\text{predicted}_i)^2}$ |
| R squared ( $R^2$) | $R^2=1-\frac{MSE_{\text{model}}}{MSE_{\text{base}}}$ |

## 5.4 Decision tree

A decision tree is a statistical model that measures a target value using a collection of binary rules. Each tree is a simple model with branches, nodes, and leaves. In machine learning, DT models solve problems in classification and regression. Moreover, the decision tree can implement a sequential decision process. As the function gets evaluated from the root node, one of the two nodes (branches) is chosen. Each node in the tree is essentially a decision rule. This process gets repeated until the final leaf, which is usually the target, is reached.

## 6. Performance evaluation metrics

The evaluation metrics used to evaluate the machine learning models are mean-absolute error (MAE), R squared ($R^2$), mean squared error (MSE), and root mean squared error (RMSE). Table 1 presents the various evaluation metrics. The measurements are described by determining the numeric predictions for each of the $n$ test cases and the actual (observed) and expected (estimated) values for test case $i$. The following are some of the most common metrics for assessing performance:

## 6.1 Mean Absolute Error

The mean absolute error is the absolute difference between actual and predicted values. It indicates the magnitude of error in the prediction. However, it does not explain the error direction (e.g., over or under predicting).

## 6.2 Mean Squared Error

The mean squared error is similar to the mean absolute error (MAE) in that it indicates the magnitude of the error. MSE determines the average squared distance between the actual and the expected values.

## 6.3 Root Mean Squared Error

The square root of the mean squared error is the root mean squared error. RMSE highlights the average deviation of predictions from actual values. In RMSE, the error is unbiased and follows a normal distribution. For error presentation, RMSE converts the units back to the output variables' original units.

## 6.4 $R^2$ Correlation

The $R^2$ metric indicates how well a series of forecasted values match the actual values. In statistical literature, R squared metric refers to the coefficient of determination. The R squared value ranges between 0 for no-fit and 1 for perfect fit; for example, with a value close to zero and less than 0.5, the forecasts have an imprecise match to the actual values.

In the above table, $n$ represents the number of samples in the dataset, $\text{actual}_i$ is the actual value for the $i^{\text{th}}$ sample, and $\text{predicted}_i$ is the predicted value for the $i^{\text{th}}$ sample.

## 7. Results and discussion

## 7.1 Traditional modeling results and dataset generation

### 7.1.1 Collected experiments and measurements

This section discusses various laboratory experiments on nanoparticles transport in water-saturated porous columns found in the literature (Murphy, 2012). The nanoparticles selected are made from different nanomaterials, such as silica and iron oxide. The used porous media in those experiments are columns filled with sand grains and saturated with water or glass beads. This paper presents six different experimental works of nanoparticles transport in porous media (Murphy, 2012). Table 2 demonstrates a summary of measured experimental parameters used in Murphy's experiments.

The selected experimental data from Murphy (2012) are experiments 73, 75, 76, 91, 92, and 93 referred to in this paper as experiments (i), (ii), (iii), (iv), (v), and (vi), respectively. Experiments (i), (ii), and (iii) are conducted with diluted nanoparticles with 5.0 wt%, 3.5 wt%, and 1.5 wt%, respectively. Murphy injected Nexsil DP dispersion fluid at 1 mL/min in 100% Boise sandstone sand pack. In experiments (iv) and (vi) with the 2.9 PV and 3.1 PV, he injected Coating I coated iron oxide nanoparticles with a concentration of 0.1 wt% into a 100% Boise sandstone sand pack at 1 mL/min and 10 mL/min, respectively. Experiment (iv) targeted the iron oxide nanoparticles retention at low flow rates and low concentrations. In contrast, experiment (v) examined the retention of iron oxide nanoparticles at high flow rates and low concentrations. In the experiment (vi), Murphy used the coated iron oxide nanoparticles with coating II with 3.8 PV of 0.1 wt.% and injected them into a 100% Boise sandstone sand pack. Murphy injected the diluted nanoparticles at a 10 mL/min rate, then reduced the rate to 1 mL/min at 2.5 PVI. Murphy also investigated the effect of lowering flow rates on the effluent history in the experiment (vi). Fig. 5 presents a regeneration of the six experiments' breakthrough curves (Murphy, 2012).

Based on the published experiments of nanoparticles transport in porous media, the retention of nanoparticles in columns depends on the nanomaterial surface properties, shape, and size. The effluent histories of the same type of nanoparticles are affected by the collector surface, flow velocity, and chemical components in the solution.

**Table 2.** Experimental conditions of Nyacol DP nanoparticles (experiments: i, ii, iii) and Iron Oxide (IO) nanoparticles (experiments: iv, v, vi) (Murphy, 2012; Zhang, 2012).

| Exp | 73 | 75 | 76 | 91 | 92 | 93 |
|---|---|---|---|---|---|---|
| Particle diameter [nm] | 27 | 27 | 27 | 150 | 150 | 147 |
| $v_p$ [cc] | 14.9 | 14.6 | 14.6 | 14.5 | 14.8 | 15.5 |
| Porosity [%] | 51.4 | 50.3 | 50.4 | 46.4 | 47.3 | 50 |
| Sand type | Boise | Boise | Boise | Boise | Boise | Boise |
| surface area $S_A$ [m$^2$] | 47.9 | 447.8 | 48 | 49.9 | 49 | 46.5 |
| Flow rate $q$ [cc/min] | 1 | 1 | 0.88 | 1 | 8.33 | 9.3 then 1.07 |
| interstitial velocity $v$ [ft/day] | 98.12 | 100.1 | 88.12 | 108.7 | 888 | 937 then 108 |
| Slug size $PVI$ [PVs] | 3 | 3 | 2.64 | 2.9 | 3.108 | 3.8 |
| Injection concentration $C_I$ [wt%] | 5 | 2.84 | 1.5 | 0.1 | 0.1 | 0.1 |
| Nanoparticle | Nexsil DP | Nexsil DP | Nexsil DP | IO (Coating 1) | IO (Coating 1) | IO (Coating 2) |
| grain size $D_p$ [$\mu$m] | 177-210 | 177-210 | 177-210 | 150-180 | 150-181 | 150-177 |
| $t_{\text{arrival}}$ | 1.06 | 1.04 | 1.09 | 2.6 | 1.86 | 1.85 |
| Intrinsic adsorption capacity $s_{\max}$, [g/g] | 3.31E-02 | 3.31E-02 | 3.31E-02 | 2.87E-01 | 2.87E-01 | 2.87E-01 |
| Adsorption or attachment coefficient $k_a$, [1/s] | 8.00E-04 | 8.00E-04 | 8.00E-04 | 2.90E-03 | 2.90E-03 | 1.00E-01 |
| Desorption or detachment coefficient $k_d$, [1/s] | 3.00E-03 | 3.00E-03 | 3.00E-03 | 2.00E-03 | 2.00E-03 | 1.50E-01 |
| dispersion coefficient $D$, [m$^2$/s] | 1.88E-06 | 1.94E- 06 | 1.95E-06 | 3.16E-06 | 1.95E-05 | 2.2E-05 |
| particle density [kg/m$^3$] | 1670 | 1670 | 1670 | 1670 | 1670 | 1670 |
| matrix density [kg/m$^3$] | 2600 | 2600 | 2600 | 2600 | 2600 | 2600 |
| column length [ft] | 1 | 1 | 1 | 1 | 1 | 1 |

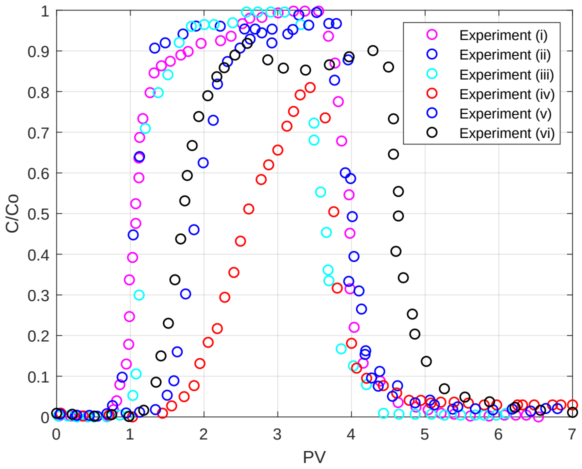


**Fig. 5.** Regeneration of the breakthrough curves of the six experiments taking from (Murphy 2012).

### 7.1.2 Simulation and Model Validation with Experimental Data

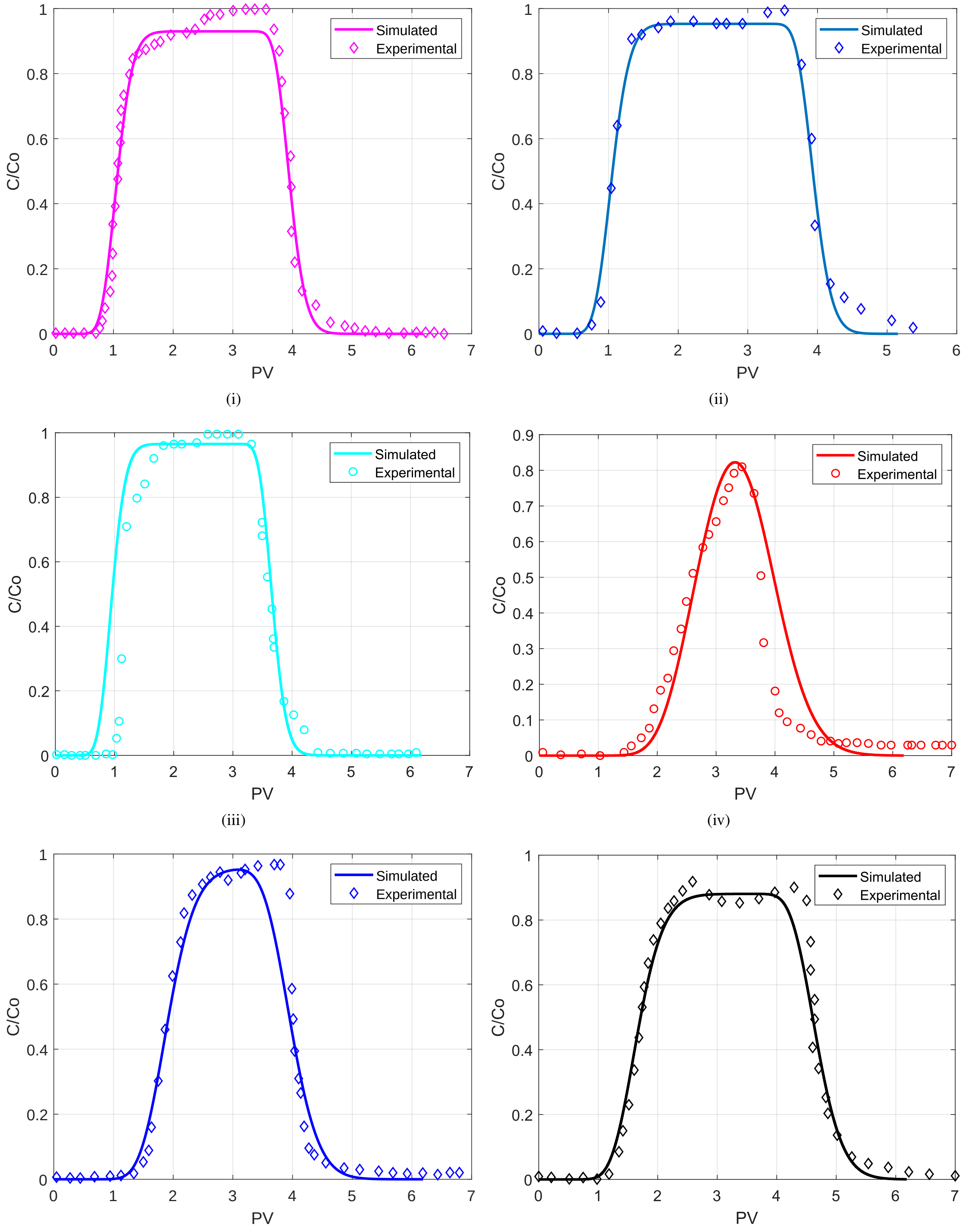
This section uses the proposed modified colloid filtration model with two sites to simulate colloid nanoparticles transport in porous media. The governing Eqs. (8)-(11), as well as the initial and boundary conditions (12) and (13), are solved numerically using the finite difference method and implemented in MATLAB environment. Table 3 presents the simulation parameters. The model variables used the given experimental conditions to validate it. This paper compares the simulated outcomes against experimental results and found out that the simulated data showed a reasonable agreement with experimental data, as shown in Fig. 6.

## 7.2 Machine learning results

### 7.2.1 Artificial datasets

After simulating the validated model with experimental data from the literature, A finite difference method is used to generate an artificial dataset of nanoparticles transport in porous media for machine learning algorithms. The finite difference method implemented in MATLAB is utilized to produce six different synthetic datasets from each of the six simulated experiments. Moreover, another diversified dataset is built by combining the six datasets. Datasets of the experiments (i), (ii), (vi), and the combined dataset are employed in machine learning.

The artificial dataset is composed of 16 features or independent variables. The features are time in seconds ($t$), pore volume ($pv$), space ($x$), the irreversible adsorption rate coefficient ($k_{irr}$), the reversible adsorption rate coefficient ($k_{ra}$), the desorption rate coefficient ($k_{rd}$), the capacity for irreversible adsorption ($s_{1\max}$), the capacity for reversible adsorption($s_{2\max}$), porosity ($\phi$), particle density ($\rho$), dispersion coefficient ($D$), velocity ($v_p$), flow rate ($q_1$), surface area ($A$), the reversible adsorbed nanoparticles concentration on a solid surface ($s_1$), and the irreversible adsorbed nanoparticles concentration on a solid surface ($s_2$). On the other hand, the target variable or the dependent variable selected for



(i)

(ii)

(iii)

(iv)

(v)

(vi)

**Fig. 6.** Experimental (points) and simulated (curves) of nanoparticles concentration profile at different times effluent histories of experiments (i), (ii), (iii), (iv), (v) and (vi).

**Table 3.** Simulation parameters in the two-site model.

| Exp | 73 | 75 | 76 | 91 | 92 | 93 |
|---|---|---|---|---|---|---|
| $v_p$ [cc] | 14.8E-06 | 14.8E-06 | 14.8E-06 | 14.8E-06 | 14.8E-06 | 14.8E-06 |
| Porosity $\varphi$ [%] | 51.4 | 50.3 | 51.4 | 51.4 | 50.3 | 51.4 |
| total surface area $S_A$ [m$^2$] | 4.49E-05 | 4.49E-05 | 48 | 49.9 | 4.49E-5 | 4.49E-5 |
| $q$ [cc/min] | 1 | 1 | 0.88 | 1 | 8.33 | 9.3 then 1.07 |
| interstitial velocity $v$ [t/day] | 98.12 | 100.1 | 88.12 | 108.7 | 888 | 937 then 108 |
| Slug size $PVI$ [PVs] | 2.64 | 2.64 | 2.27 | 1.21 | 1.88 | 2.74 |
| Injection concentration $C_I$ [wt%] | 5 | 2.84 | 1.5 | 0.1 | 0.1 | 0.1 |
| particle density, [kg/m$^3$] | 1.52 E-03 | 1.52 E-03 | 1.52 E-03 | 2.87E-01 | 1.52 E-03 | 1.52E3 |
| dispersion coefficient, [m$^2$/s] | 2.1E-06 | 1.39E-06 | 0.8E-06 | 3.16E-06 | 0.7E-6 | 2.39E-6 |
| $s_{1\max}$ [g/g] | 12.4% | 12.4% | 12.4% | 5.75E-04 | 3.40E-04 | 12.4% |
| $s_{2\max}$ [g/g] | 1.5% | 1.5% | 1.5% | 2.87E-01 | 7.60E-06 | 1.5% |
| $K_{\text{irr}}$ [1/s] | 8.0E-03 | 8.0E-03 | 8.0E-03 | 7.00E-03 | 8.00E-03 | 8.00E-03 |
| $K_{\text{ra}}$ [1/s] | 1.0E-05 | 1.0E-05 | 1.0E-05 | 1.20E-04 | 1.0E-05 | 1.0E-05 |
| $K_{\text{rd}}$ [1/s] | 1.4E-02 | 1.4E-02 | 1.4E-02 | 5.00E-04 | 1.4E-02 | 1.4E-02 |

**Table 4.** The statistical information of the dataset features of the experiment (i).

| | **count** | **mean** | **std** | **min** | **25%** | **50%** | **75%** | **max** |
|---|---|---|---|---|---|---|---|---|
| $t$ | | 300 | 173.23441 | 0 | 150 | 300 | 450 | 600 |
| $x$ | | 0.15 | 0.089443 | 0 | 0.07 | 0.15 | 0.23 | 0.3 |
| $k_irr$ | | 8.00E-03 | 5.20E-18 | 8.00E-03 | 8.00E-03 | 8.00E-03 | 8.00E-03 | 8.00E-03 |
| $k_{ra}$ | | 0.00001 | 0 | 0.00001 | 0.00001 | 0.00001 | 0.00001 | 0.00001 |
| $k_{rd}$ | | 1.40E-02 | 1.21E-17 | 1.40E-02 | 1.40E-02 | 1.40E-02 | 1.40E-02 | 1.40E-02 |
| $s_{1\max}$ | | 1.24E-01 | 8.33E-17 | 1.24E-01 | 1.24E-01 | 1.24E-01 | 1.24E-01 | 1.24E-01 |
| $s_{2\max}$ | | 1.50E-02 | 1.04E-17 | 1.50E-02 | 1.50E-02 | 1.50E-02 | 1.50E-02 | 1.50E-02 |
| $\phi$ | 186031 | 5.14E-01 | 1.11E-16 | 5.14E-01 | 5.14E-01 | 5.14E-01 | 5.14E-01 | 5.14E-01 |
| $\rho$ | | 1520 | 0 | 1520 | 1520 | 1520 | 1520 | 1520 |
| $D$ | | 1.88E-06 | 2.12E-22 | 1.88E-06 | 1.88E-06 | 1.88E-06 | 1.88E-06 | 1.88E-06 |
| $v_p$ | | 3.09E-03 | 2.17E-18 | 3.09E-03 | 3.09E-03 | 3.09E-03 | 3.09E-03 | 3.09E-03 |
| $q_1$ | | 1.39E-07 | 0.00E+00 | 1.39E-07 | 1.39E-07 | 1.39E-07 | 1.39E-07 | 1.39E-07 |
| $A$ | | 4.49E-05 | 3.39E-20 | 4.49E-05 | 4.49E-05 | 4.49E-05 | 4.49E-05 | 4.49E-05 |
| $s_1$ | | 0.00005 | 0.000031 | 0 | 0.000019 | 0.000062 | 0.00008 | 0.00008 |
| $s_2$ | | 1.10E-08 | 8.97E-09 | 0.00E+00 | 1.73E-09 | 9.74E-09 | 2.07E-08 | 2.37E-08 |
| $c$ | | 4.73E-01 | 4.76E-01 | 0.00E+00 | 3.76E-10 | 2.47E-01 | 1.00E+00 | 1.00E+00 |

prediction is the nanoparticles concentration ($c$). Tables 4 and 5 present the statistical information of the dataset features. It includes all instances, the mean, the standard deviation, the minimum value, the quarter, the half, the three quarters, and the maximum value of each feature of datasets of the experiment (i) and the combined datasets, respectively.

### 7.2.2 Data pre-processing

Data processing includes data cleaning, applying normalization techniques, and removing outliers. In the generated artificial dataset, they have no empty cells to drop. The 16 features are utilized as input for all the techniques, and the target feature selected is the nanoparticles concentration. Moreover, the Standard Scaler function of the Scikit-learn library is used to scale and standardize the values of the independent variables. This process would keep the independent variables within a similar range. The important features for each model are identified to be used for predicting the target, which is very important in assigning weights.

### 7.2.3 Data correlation

In the preprocessing phase of the dataset, it is ensured that no empty cells. Moreover, the correlations between each feature in the dataset are tested. Checking the correlations

**Table 5.** The statistical information of the combined dataset features from all six datasets.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| $t$ | | 292.856123 | 170.427085 | 0.000000 | 145.500000 | 291.500000 | 437.500000 | 600.000000 |
| $pv$ | | 3.019029 | 1.756919 | 0.000000 | 1.499947 | 3.005049 | 4.510151 | 6.185350 |
| $x$ | | 0.150000 | 0.089443 | 0.000000 | 0.070000 | 0.150000 | 0.230000 | 0.300000 |
| $k_{irr}$ | | 0.007829 | 0.000377 | 0.007000 | 0.008000 | 0.008000 | 0.008000 | 0.008000 |
| $k_{ra}$ | | 0.000029 | 0.000041 | 0.000010 | 0.000010 | 0.000010 | 0.000010 | 0.000120 |
| $k_{rd}$ | | 0.011686 | 0.005088 | 0.000500 | 0.014000 | 0.014000 | 0.014000 | 0.014000 |
| $S_{1\max}$ | | 0.081644 | 0.058641 | 0.000340 | 0.000575 | 0.124000 | 0.124000 | 0.124000 |
| $s_{2\max}$ | | 0.059057 | 0.103828 | 0.000008 | 0.015000 | 0.01500 | 0.015000 | 0.287000 |
| $\phi$ | 217186 | 0.510543 | 0.005107 | 0.503000 | 0.503000 | 0.514000 | 0.514000 | 0.514000 |
| $\rho$ | | 1259.483969 | 572.749952 | 0.287000 | 1520.000 | 1520.0 | 1520.0 | 1520.0 |
| $D$ | | 1.241718e-06 | 7.838905e-07 | 5.000000e-09 | 7.000000e-07 | 1.110000e-06 | 1.880000e-06 | 2.390000e-06 |
| $\mathbf{v}_p$ | | 3.092675e-03 | 1.734727e-18 | 3.092675e-03 | 3.092675e-03 | 3.092675e-03 | 3.092675e-03 | 3.092675e-03 |
| $q_1$ | | 1.388610e-07 | 0.000000e+00 | 1.388610e-07 | 1.388610e-07 | 1.388610e-07 | 1.388610e-07 | 1.388610e-07 |
| $A$ | | 4.490000e-05 | 1.355256e-20 | 4.490000e-05 | 4.490000e-05 | 4.490000e-05 | 4.490000e-05 | 4.490000e-05 |
| $s_1$ | | 0.000115 | 0.000185 | 0.000000 | 0.000016 | 0.000049 | 0.000079 | 0.000575 |
| $s_2$ | | 2.652439e-04 | 7.314784e-04 | 0.000000 | 1.680945e-09 | 1.187305e-08 | 2.204130e-08 | 9.946130e-01 |
| $c$ | | 4.020986e-01 | 4.537503e-01 | 0.000000 | 3.543135e-08 | 5.370073e-02 | 2.630185e-03 | 1.000000 |

is vital for exploring the artificial dataset and identifying the features that affect the target variable the most for the prediction. Fig. 7 illustrates the visualization of the correlation matrices through plotting a heatmap in Python developed from the combined dataset. Fig. 7 shows that kirr, krd, and rho are highly correlated, however, $c$ is correlated with $s_1$, $s_2$, $D$, $\rho$, $s_{1\max}$, $s_{2\max}$, $k_{irr}$, $k_{ra}$, $k_{rd}$, $pv$, and $t$.

### 7.2.4 Features' importance

Feature importance referred to the techniques that give a ranking to input features based on how effective they are at predicting a target variable. In predictive modeling, feature importance scores play an essential role in providing insight into the data, insight into the model, and the basis for dimensionality reduction and feature selection, increasing the efficiency and effectiveness of a predictive model. Using important features and having high scores while deleting insignificant features can help in simplifying the developed model, improve its performance, and speed up the modeling process. Moreover, selecting the key features would help avoid overfitting by reducing the number of features employed for training. The importance of features for each machine learning model is investigated to train the model. The Scikit-learn library and feature importance function are applied for calculating the score of each feature. Fig. 8 illustrates the features importance score of DT, RF, and GBR models of the combined dataset. It can be seen that the time $t$ and $x$, $D$, $pv$, $s_1$, and $s_2$ were important features used for predicting the nanoparticles concentration in both DT and RF compared to other features. The injection history is mainly measured by the injection rate given by pore volume ($pv$), which is considered an important feature, especially for the GBR machine learning model.

The generated artificial datasets from the numerical simulation results are employed in the machine learning algorithms. Four different machine learning algorithms are used to design the predictive models, including DT, ANN, GBR, and RF. The datasets are divided into two subsets in a ratio of 80: 20. 80% of the datasets are the training sets, the sets of samples used to train the models, and 20% of the data is for testing. The test sets were a collection of samples used solely to evaluate output in unobserved data. This study utilizes Jupyter Notebook with Python programming language for implementation. Jupyter Notebook is an open-source web application to write live code to build statistical and machine learning models and perform numerical simulations (Mendez et al., 2019). The train test split function from the Scikit-learn library is used to split the dataset into training and testing. It generates four variables: $x$ train, $y$ train, $x$ test, and $y$ test. The model is trained with $x$ train and $y$ train, while the $x$ test is used to evaluate the model on the external testing set. Comparing the predicted value to the actual value can identify the error and the model's accuracy.

## 7.3 Machine learning results and discussion

This subsection presents the results of predicting nanoparticles concentration in datasets from the experiment (i), (ii), (vi), and the combined dataset of the six experiments using the four machine learning algorithms: DT, GBR, ANN, and RF. The performance of the models are evaluated using mean square error, mean absolute error, $R^2$ correlation, and root mean squared error. It is found that DR model has the lowest root mean square error and the highest $R^2$ in the dataset of the experiment (i). For datasets of experiments (ii) and (vi), RF had the lowest root mean squared error value and the highest

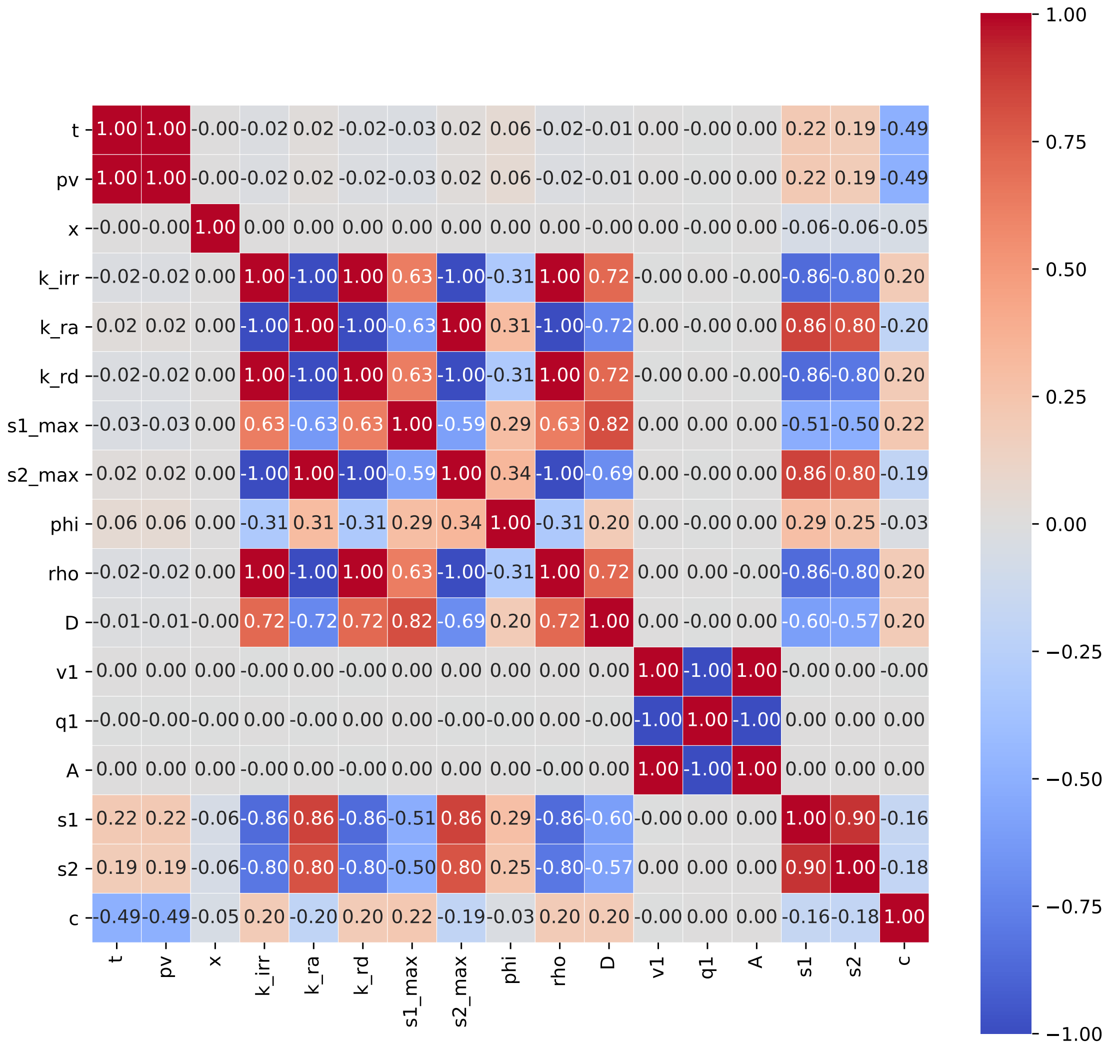
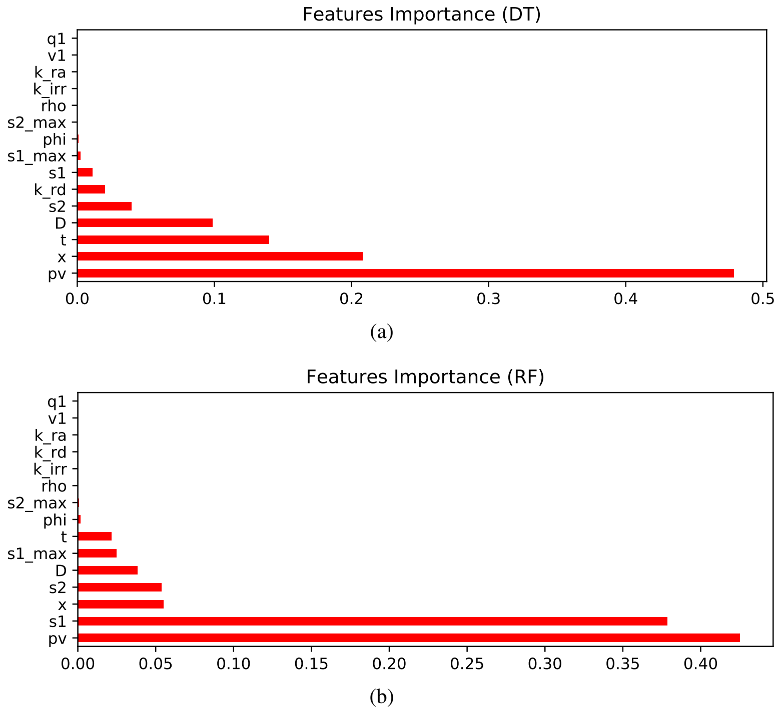
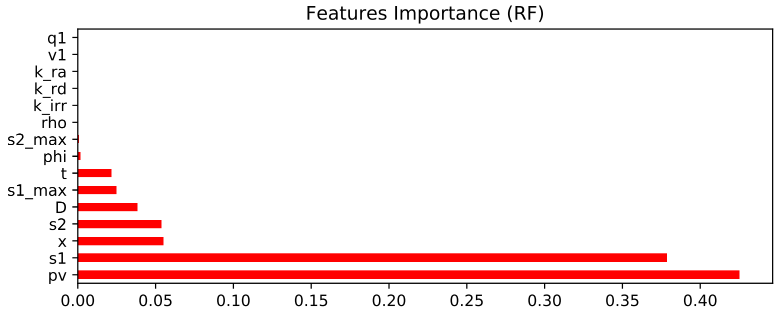


**Fig. 7.** Correlation Values between the dataset variables.



(a)

(b)



(c)

**Fig. 8.** Features' importance of DT and RF, and GBR models of the combined dataset.

$R^2$ value. Moreover, in the combined dataset, it is found that the RF model can reliably predict the nanoparticles' concentration with high accuracy and low error. RF has showed a good performance by achieving the highest $R^2$ value of (0.999716) and the lowest RMSE of (0.007654). Moreover, the DT model has high $R^2$ squared value of (0.999507) and low RMSR of (0.010076) compared to other models. Furthermore, the predicted outcomes are compared with the actual for each model by plotting the scatter plots, as demonstrated in Figs. 9-12 for the dataset of the experiments (i), (ii), (vi), and the combined dataset respectively. The scatterplots between the actual and the predicted nanoparticles concentration of RF and DT models show a good linear correlation. The overall results show that RF and DT models can accurately deal with nanoparticles concentration prediction. Tables 6-9 present the evaluation metric of all models of the four datasets.

### 7.3.1 Tuning the hyperparameters

Hyperparameters are the parameters that can be adjusted and fine-tuned to improve the machine learning model's performance. Tuning the hyperparameters can improve the model performance and reduce overfitting.

**1) Random forest algorithm**

Based on the RF features' importance in Fig. 8, the significant features for building the model are used. The key features are $t$, $x$, $D$, $pv$, $s_1$, $s_2$, $D$, and $\rho$, to predict the target $c$. In the RF technique, two hyperparameters are applied, including the max features and the number of estimators. Max features (number of features) that are used to construct the predictive model. $N$ estimator (number of trees) is used to build the prediction model. In addition, the GridSearchCV package of the Scikit-learn library is employed to find the optimal hyperparameter values. The GridSearchCV method tunes hyperparameter by performing an exhaustive search of optimal parameters in a grid-wise manner. GridSearchCV function can perform all possible pairwise computations of the two hyperparameters. The number of computations is the product of the parameter values. Running the GridSearchCV for RF hyperparameters tuning, It is found out that the best parameters that would give the highest accuracy of a score of 1.00 are using max features of 6 and n estimators of 80. Figs. 13 and 14 present the 2D contour plot and the 3D surface plot of hyperparameter tuning of RF model with accuracy scores, respectively.

**Table 6.** Model performance evaluation of dataset of the experiment (i).

| Metric | DT | RF_sc | GBR_sc | ANN (tanh) |
|---|---|---|---|---|
| RMSE | 0.001163 | 0.001783 | 0.017212 | 0.014800 |
| MSE | 0.000001 | 0.000003 | 0.000296 | 0.000219 |
| MAE | 0.000398 | 0.000198 | 0.006612 | 0.005813 |
| $R^2$ | 0.999994 | 0.999986 | 0.998692 | 0.999033 |

*sc: scaled dataset with standard scaler function.

**Table 7.** Model performance evaluation for the dataset of the experiment (ii).

| Metric | DT | RF_sc | GBR_sc | ANN (sigmoid) |
|---|---|---|---|---|
| RMSE | 0.004825 | 0.004743 | 0.020561 | 0.022170 |
| MSE | 0.000023 | 0.000022 | 0.000423 | 0.000492 |
| MAE | 0.000519 | 0.000260 | 0.008206 | 0.006243 |
| $R^2$ | 0.999894 | 0.999897 | 0.998073 | 0.997759 |

*sc: scaled dataset with standard scaler function.

**Table 8.** Model performance evaluation of dataset of the experiment (vi).

| Metric | DT | RF | GBR_sc | ANN (sigmoid) |
|---|---|---|---|---|
| RMSE | 0.000829 | 0.000561 | 0.017218 | 0.018258 |
| MSE | 0.000001 | 0.000000 | 0.000296 | 0.000333 |
| MAE | 0.000423 | 0.000202 | 0.009545 | 0.009762 |
| $R^2$ | 0.999997 | 0.999998 | 0.998571 | 0.998393 |

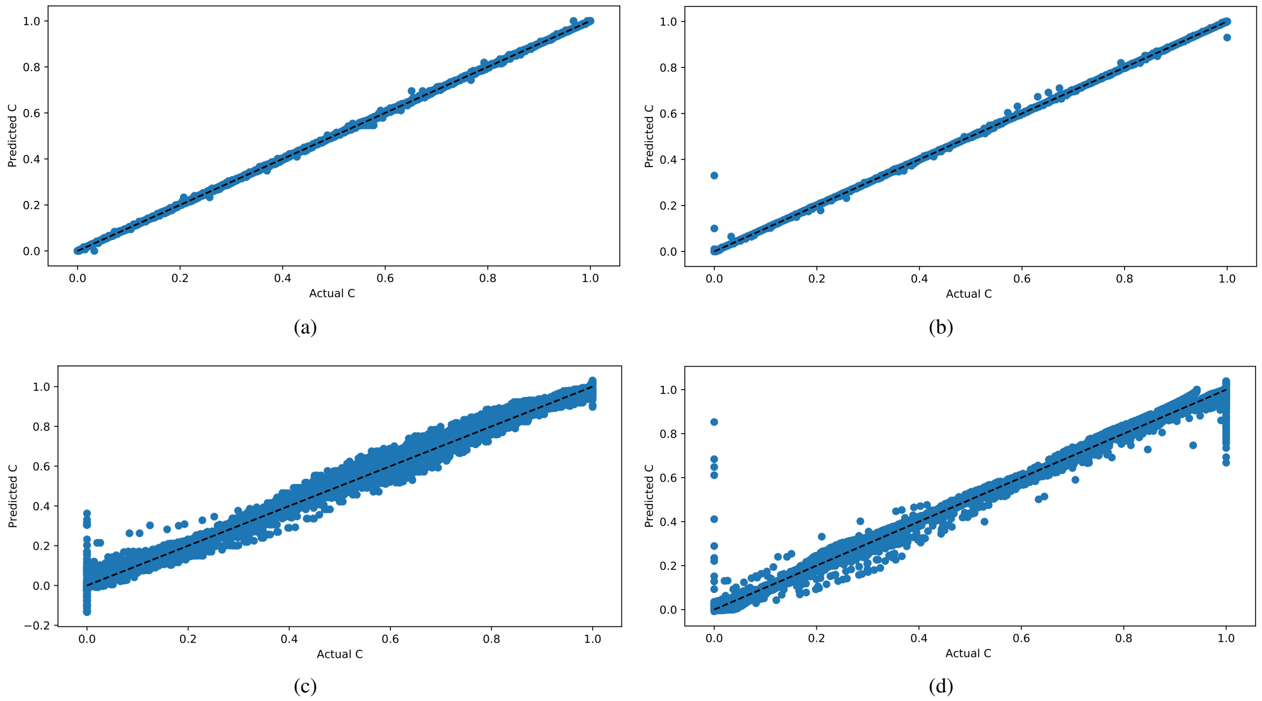
*sc: scaled dataset with standard scaler function.

**Table 9.** Model performance evaluation of the combined dataset.

| Metric | DT_sc | RF | GBR_sc | ANN (sigmoid) |
|---|---|---|---|---|
| RMSE | 0.010076 | 0.007654 | 0.089218 | 0.028778 |
| MSE | 0.000102 | 0.000059 | 0.007960 | 0.000828 |
| MAE | 0.001864 | 0.000862 | 0.050141 | 0.013217 |
| $R^2$ | 0.999507 | 0.999716 | 0.961272 | 0.995971 |

*sc: scaled dataset with standard scaler function.

**2) Gradient boosting regression**



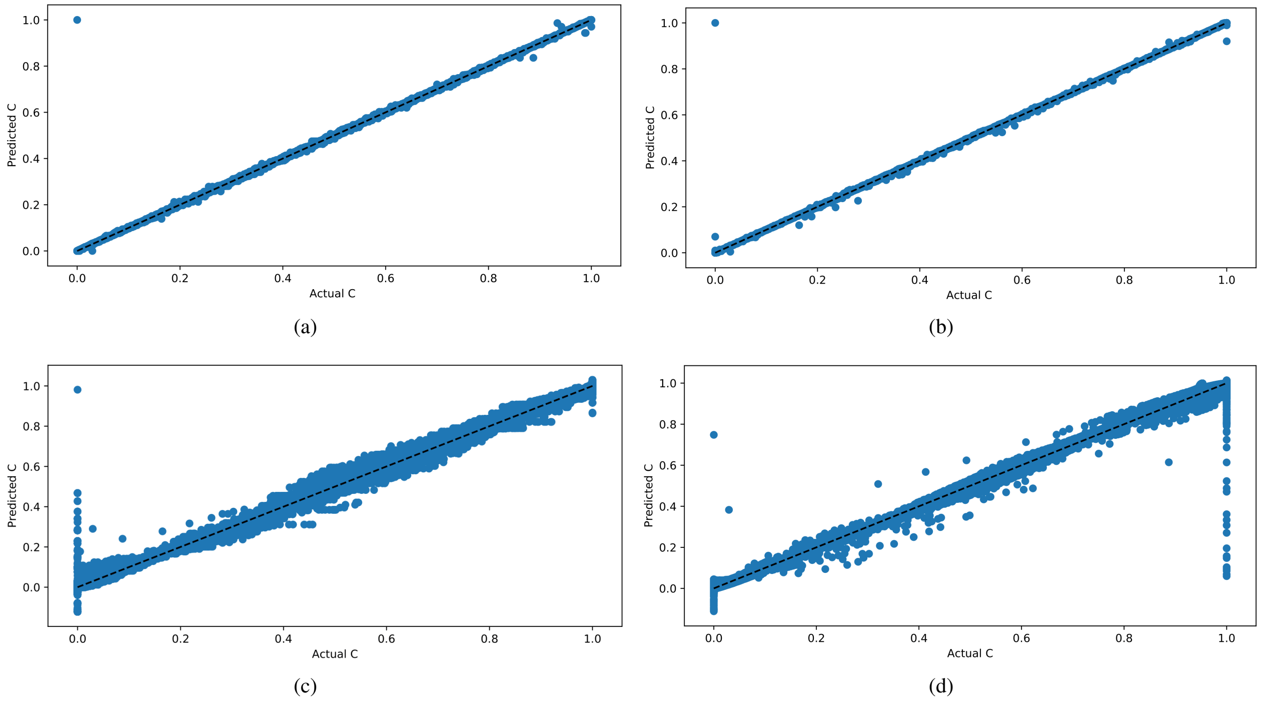
(a)

(b)

(c)

(d)

**Fig. 9.** Scatter plots of actual and predicted nanoparticles concentrations using different machine learning techniques DT, RF, GBR, and ANN of the experiment (i) dataset.



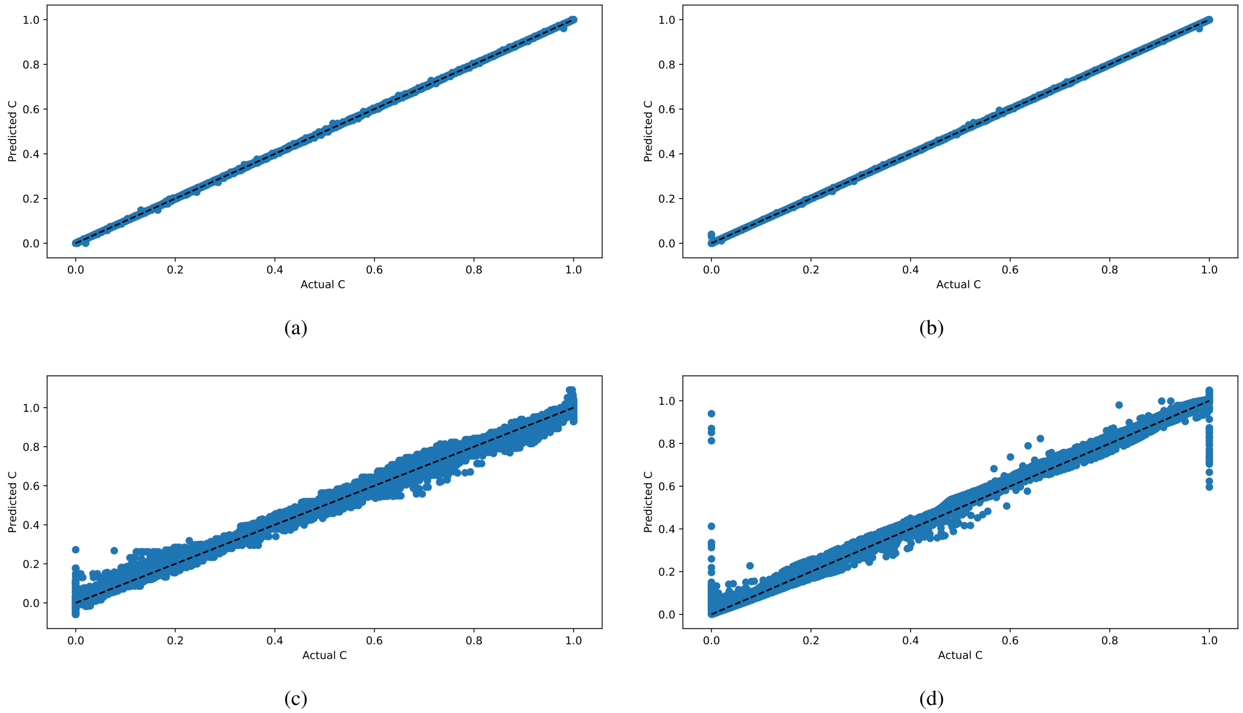
(a)

(b)

(c)

(d)

**Fig. 10.** Scatter plots of actual and predicted nanoparticles concentrations using different machine learning techniques DT, RF, GBR, and ANN of the experiment (ii) dataset.



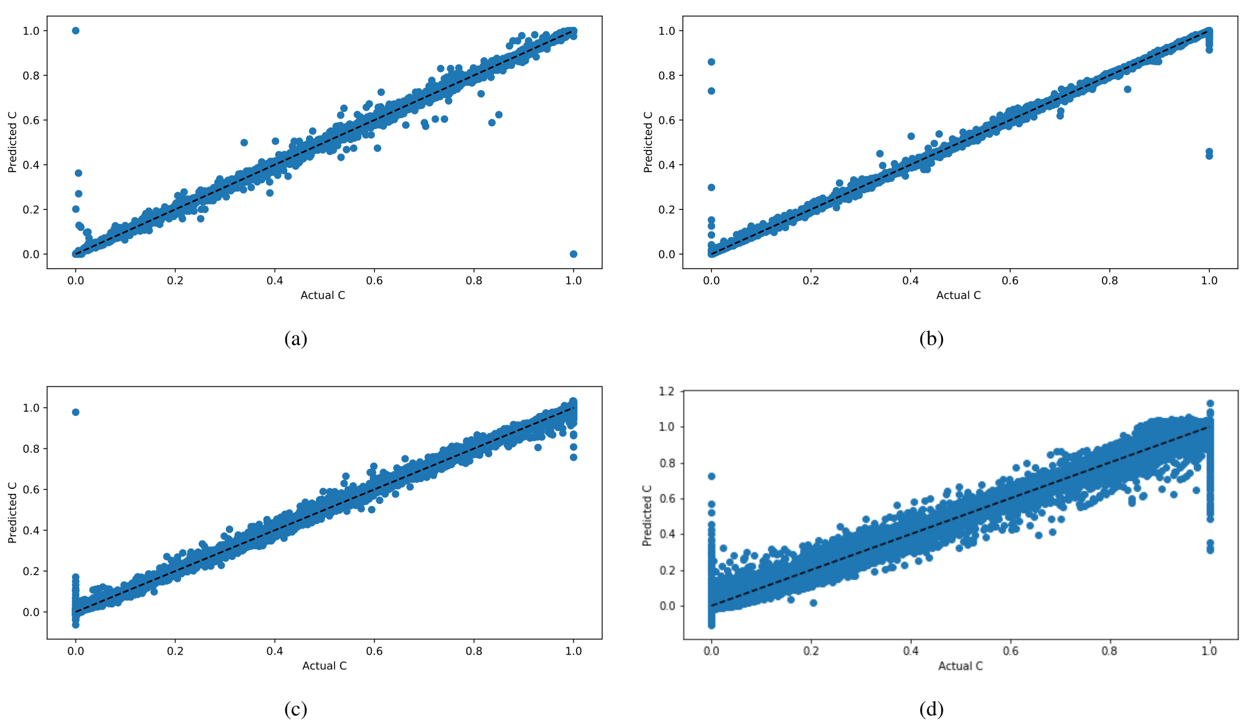
(a)

(b)

(c)

(d)

**Fig. 11.** Scatter plotx of actual and predicted nanoparticles concentrations using different machine learning techniques DT, RF, GBR, and ANN of the dataset of the experiment (vi).



(a)

(b)

(c)

(d)

**Fig. 12.** Scatter plots of actual and predicted nanoparticles concentrations using different machine learning techniques DT, RF, GBR, and ANN of the combined dataset of all the six experiments.

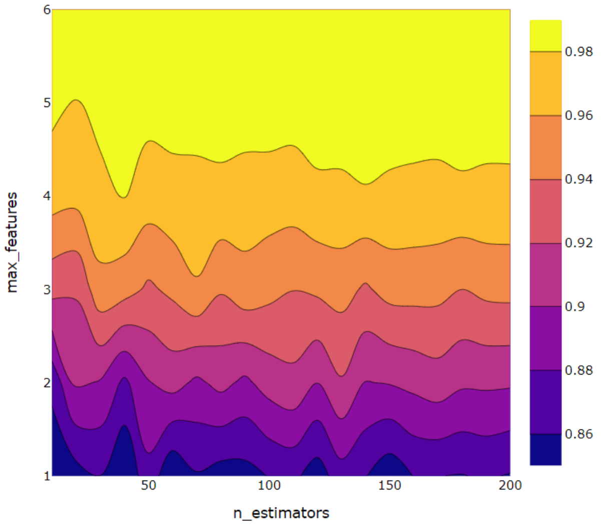


**Fig. 13.** 2D contour plot of hyperparameters tuning of RF model.

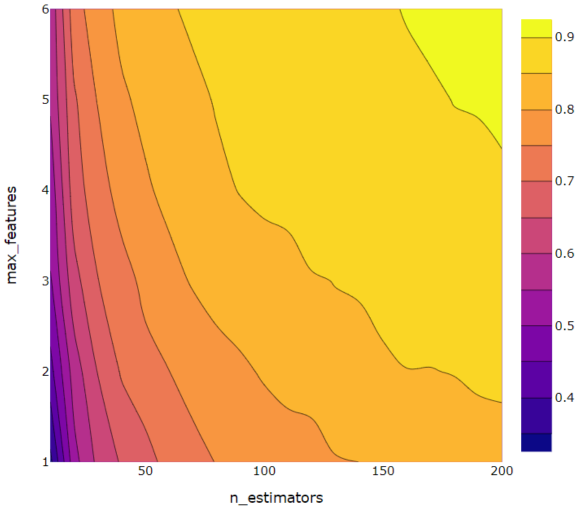


**Fig. 15.** 2D contour plot of hyperparameters tuning of GBR model.

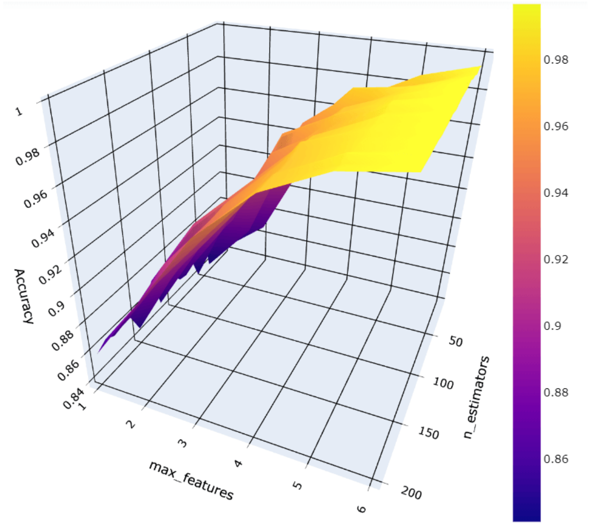


**Fig. 14.** 3D surface plot of hyperparameters tuning of RF model with accuracy scores.

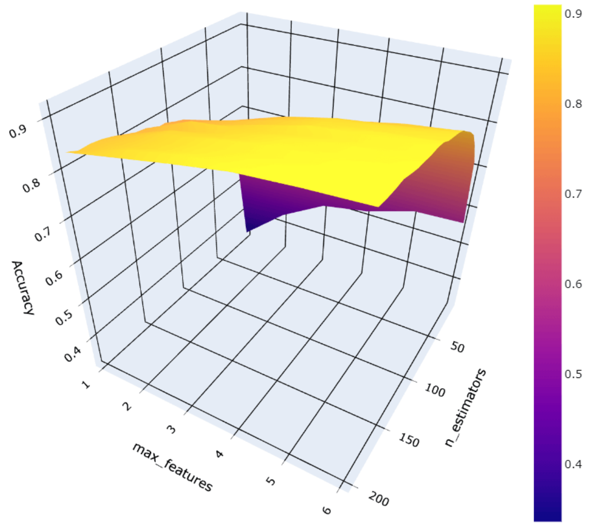


**Fig. 16.** 3D surface plot of hyperparameters tuning of GBR model with accuracy scores.

The features that are used in training the GBR model are $t$, $x$, $D$, $pv$, $s_1$, $s_2$, $s_{1\max}$, $k_{irr}$, and $s_{2\max}$, which are the key features presented in Fig. 8. the GridSearchCV function is utilized for GBR hyperparameters tuning; it is found that the best parameters that give the accuracy score of 0.9 are max features of 6 and $n$ estimators of 200. Figs. 15 and 16 present the 2D contour plot and the 3D surface plot of hyperparameters tuning of the GBR model with accuracy scores, respectively. Comparing between GBR models with and without scaling the dataset. It is observed that without scaling, the combined dataset has a RMSE of (0.139996) and $R^2$ of (0.904826). However, when the combined dataset is scaled using the standard scaler function, better performance for the predictive model is achived with a RMSE of (0.089218) and $R^2$ of (0.961272).

### 3) Decision Tree algorithm

The features that is selected to train the DT model based on the important features presented in Fig. 8 are $t$, $x$, $D$, $pv$, $s_1$, $s_2$, $k_{rd}$, and $s_{1\max}$. The features are standardized using the standard scaler function and found out that there was an improvement in the model performance. Using the GridSearchCV method to tune the hyperparameter, it is remarked that the best parameters are a max depth of 21 and max features of 10, leading to an accuracy score of 1. The 2D contour plot and the 3D surface plot of hyperparameters tuning of DT model with the accuracy scores are shown in Figs. 17 and 18, respectively.

### 4) ANN optimization

The ANN model is built using the TensorFlow library with one input layer, three hidden layers, and one output layer. Standard Scaler function is used to scale the dataset. Different three hidden layers with various number of neurons are utilized in the ANN model. One model has three hidden layers with six neurons each and with ReLu activation function. Another ANN model is built using three hidden layers with 15 neurons each and with ReLu activation function in the first two hidden

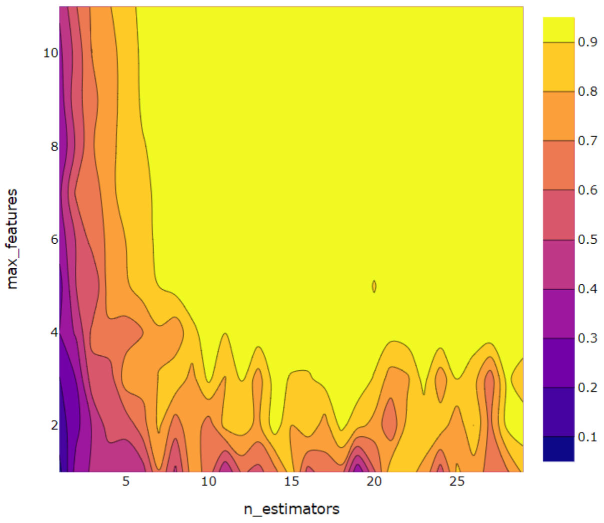


**Fig. 17.** 2D contour plot of hyperparameters tuning of DT model.

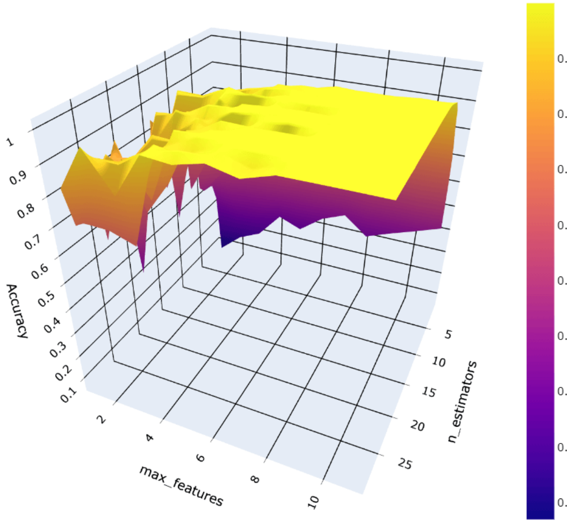


**Fig. 18.** 3D surface plot of hyperparameters tuning of DT model with accuracy scores.

layers and a sigmoid activation function in the last hidden layer. A third neural networks model is built with three hidden layers of six neurons each and ReLu activation function in the first two hidden layers, and the third hidden layer with tanh activation function. The model is compiled with adam optimizer. Fig. 19 illustrates the scatter plot between the actual and predicted data of the ANN models of different datasets and with different activation functions: ReLu, sigmoid, and tanh. While tables 10-13 present the performance evaluation of the ANN models with different activation functions of diverse datasets of experiments (i), (ii), (vi), and the combined dataset. It is noticed that the third hidden layer with a sigmoid activation function has better results in most datasets.

Our objective is to reduce the error concerning the weights to develop an accurate and not-overfit model. Moreover, other parameters that can be tuned to enhance the model performance further are:

**Table 10.** Evaluation metric of ANN models with different activation function of the dataset of the experiment (i).

| Metric | ANN (tanh) | ANN (sigmoid) | ANN (ReLu) |
|---|---|---|---|
| RMSE | 0.014800 | 0.017889 | 0.016492 |
| MSE | 0.000219 | 0.000320 | 0.000272 |
| MAE | 0.005813 | 0.007737 | 0.007472 |
| $R^2$ | 0.999033 | 0.998588 | 0.998799 |

**Table 11.** Evaluation metric of ANN models with different activation function of the dataset of the experiment (ii).

| Metric | ANN (tanh) | ANN (sigmoid) | ANN (ReLu) |
|---|---|---|---|
| RMSE | 0.034156 | 0.022170 | 0.023138 |
| MSE | 0.001167 | 0.000492 | 0.000535 |
| MAE | 0.013908 | 0.006243 | 0.013118 |
| $R^2$ | 0.994682 | 0.997759 | 0.997560 |

**Table 12.** Evaluation metric of ANN models with different activation function of the dataset of the experiment (vi).

| Metric | ANN (tanh) | ANN (sigmoid) | ANN (ReLu) |
|---|---|---|---|
| RMSE | 0.029627 | 0.018258 | 0.036722 |
| MSE | 0.000878 | 0.000333 | 0.001348 |
| MAE | 0.017099 | 0.009762 | 0.020380 |
| $R^2$ | 0.995769 | 0.998393 | 0.993500 |

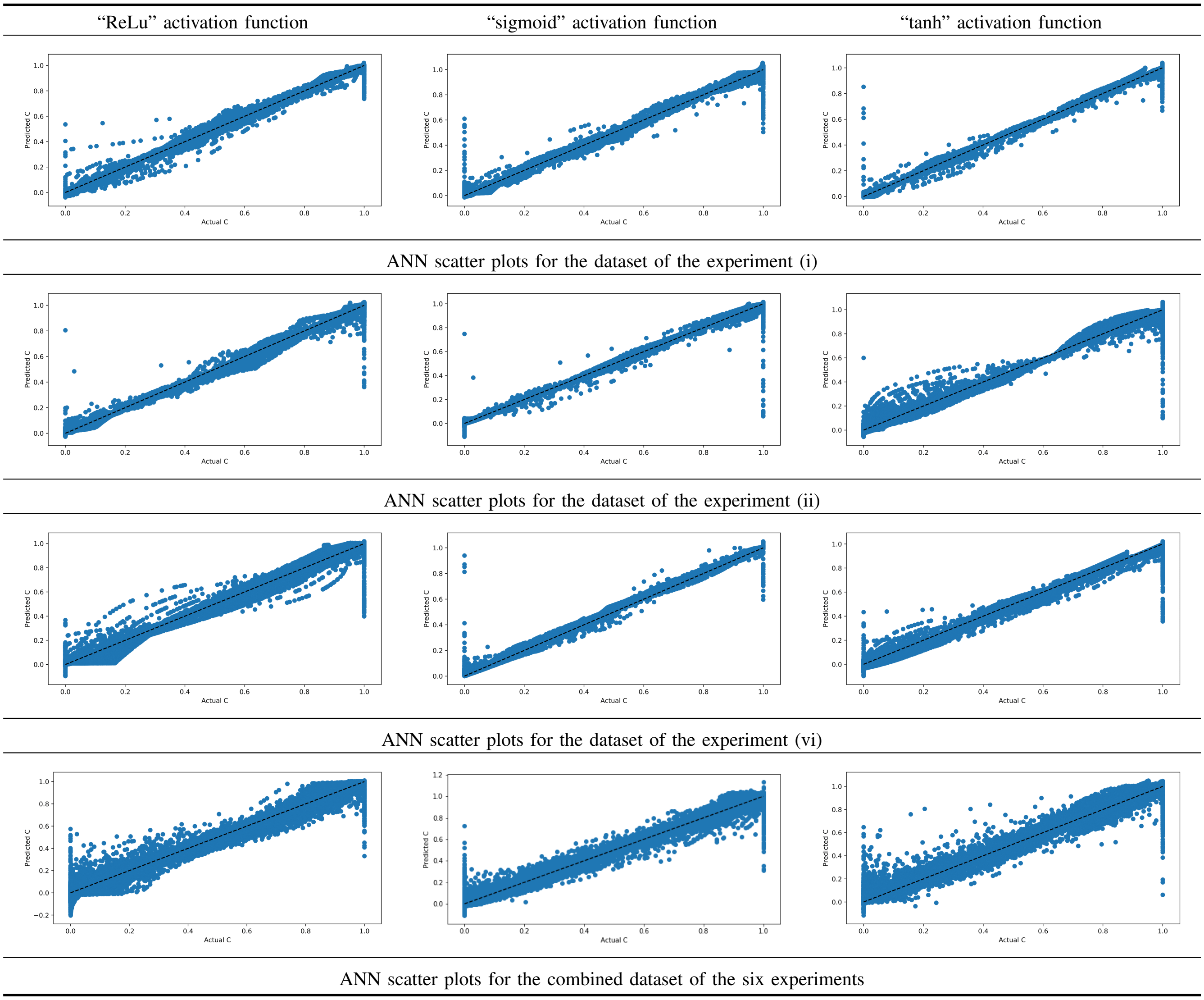
**Table 13.** Evaluation metric of ANN models with different activation function of the combined dataset of all the six experiments.

| Metric | ANN (tanh) | ANN (sigmoid) | ANN (ReLu) |
|---|---|---|---|
| RMSE | 0.033885 | 0.028778 | 0.034945 |
| MSE | 0.001148 | 0.000828 | 0.001221 |
| MAE | 0.017075 | 0.013217 | 0.015759 |
| $R^2$ | 0.994414 | 0.995971 | 0.994059 |

- Max depth: The maximum depth of an individual tree in the ensemble. Increasing the depth will result in overfitting.
- Learning rate: The rate of updating the weights. As the learning rate increases, the model learns faster but with the risk that the model might miss the global minima. As the learning rate decreases, it could be difficult for the model to converge. A learning rate of 0.3 is used in the ANN model.
- N estimators: The number of trees used in the ensemble.

## 8. Conclusions

Nanotechnology is a promising tool for managing various petroleum engineering problems. Nanoparticles can be used in



ANN scatter plots for the dataset of the experiment (i)

ANN scatter plots for the dataset of the experiment (ii)

ANN scatter plots for the dataset of the experiment (vi)

ANN scatter plots for the combined dataset of the six experiments

**Fig. 19.** Pore fractal dimensions based on FHH model.

EOR to escalate oil production to meet the energy demand. Nanoparticles can change the properties of the reservoir and the formation; therefore, they are utilized in the area of EOR through the injection of nanoparticles into the formation and monitoring their impact on the recovery factor. Thus, this work generates artificial datasets using the mathematical continuum models validated against experimental results from the literature. The machine learning algorithms are performed on three datasets, and another dataset is generated by combining all the datasets. Scikit-learn library is used to investigate data preprocessing, correlation, and feature importance of datasets. Moreover, the GridSearchCV algorithm is applied to optimize hyperparameters tuning. DT, RF, GBR, and ANN models are applied to predict the nanoparticles concentration in porous media. Furthermore, the performance of the predictive models is assessed using mean absolute error, R-squared correlation, mean squared error, and root mean squared error. It is found that the RF models and DT achieved high performance to predict the nanoparticles concentration for all the predictive models from all the four datasets compared to other developed forecasting models.

## Conflict of interest

The authors declare no competing interest.

## References

Abdelfatah, E., Pournik, M., Shiau, B. J. B., et al. Mathematical modeling and simulation of nanoparticles transport in heterogeneous porous media. Journal of Natural Gas Science and Engineering, 2017, 40: 1-16.

Alvarado, V., Manrique, E. Enhanced oil ecovery concepts, in Enhanced Oil Recovery, edited by V. Alvarado and E. Manrique, Boston, Gulf Professional Publishing, pp. 7-16, 2010.

Alvarado, V., Manrique, E. Enhanced oil recovery: An update review. Energies, 2010b, 3: 1529-1575.

Benamar, A., Ahfir, N. D., Wang, H., et al. Particle transport in a saturated porous medium: Pore structure effects. Comptes Rendus Geoscience, 2007, 339(10): 674-681.

Bradford, S. A., Yates, S. R., Bettahar, M., et al. Physical factors affecting the transport and fate of colloids in saturated porous media. Water Resources Research, 2002, 38(12): 63-1-63-12.

Breiman, L. Random forests. Machine Learning, 2001, 45: 5-32.

Changdar, S., Saha, S., De, S. A smart model for prediction of viscosity of nanofluids using deep learning. Smart Science, 2020, 8(4): 242-256.

Cushing, R. S., Lawler, D. F. Depth filtration: Fundamental investigation through three-dimensional trajectory analysis. Environmental Science and Technology, 1998, 32(23): 3793-3801.

Daribayev, B., Akhmed-Zaki, D., Imankulov, T., et al. Using machine learning methods for oil recovery prediction. ECMOR XVII, European Association of Geoscientists and Engineers, 2020, 2020(1): 1-13.

Álvarez del Castillo, A., Santoyo, E., García-Valladares, O. A new void fraction correlation inferred from artificial neural networks for modeling two-phase flows in geothermal wells. Computers and Geosciences, 2012, 41: 25-39.

El-Amin, M. F., Salama, A., Sun, S. Numerical and dimensional analysis of nanoparticles transport with two-phase flow in porous media. Journal of Petroleum Science and Engineering, 2015, 128: 53-64.

El-Amin, M. F., Salama, A., Sun, S. Modeling and simulation of nanoparticles transport in a two-phase flow in porous media. Paper SPE 154972 Presented at the International Oilfield Nanotechnology Conference and Exhibition, Noordwijk, The Netherlands, 12-14 June, 2012a.

El-Amin, M. F., Subasi, A., Developing a generalized scaling-law for oil recovery using machine learning techniques. Procedia Computer Science, 2019, 163: 237-247.

El-Amin, M. F., Subasi, A. Forecasting a small-scale hydrogen leakage in air using machine learning techniques. Paper IEEE 9257718 Presented at 2020 2nd International Conference on Computer and Information Sciences (ICCIS), Aljouf, Saudi Arabia, 13-15 October, 2020a.

El-Amin, M. F., Subasi, A. Predicting turbulent buoyant jet using machine learning techniques. Paper IEEE 9257628 Presented at 2020 2nd International Conference on Computer and Information Sciences (ICCIS), Aljouf, Saudi Arabia, 13-15 October, 2020b.

El-Amin, M. F., Sun, S., Salama, A. Modeling and simulation of nanoparticle transport in multiphase flows in porous media: $CO_2$ sequestration. Paper SPE 163089 Presented at the Mathematical Methods in Fluid Dynamics and Simulation of Giant Oil and Gas Reservoirs, Istanbul, Turkey, 3-5 September, 2012b.

El-Amin, M. F., Sun, S., Salama, A. Enhanced oil recovery by nanoparticles injection: Modeling and simulation. Paper SPE 164333 Presented at the SPE Middle East Oil and Gas Show and Conference, Manama, Bahrain, 10-13 March, 2013.

Esfe, M. H., Bahiraei, M., Mahian, O. Experimental study for developing an accurate model to predict viscosity of CuO–ethylene glycol nanofluid using genetic algorithm based neural network. Powder Technology, 2018, 338: 383-390.

Fan, J. Numerical study of particle transport and deposition in porous media. Bretagne, INSA de Rennes, 2018.

Hansen, L. K., Salamon, P. Neural network ensembles. IEEE transactions on pattern analysis and machine intelligence, 1990, 12(10): 993-1001.

Irfan, S. A., Shafie, A. Artificial neural network modeling of nanoparticles assisted enhanced oil recovery, in Advanced Methods for Processing and Visualizing the Renewable Energy, edited by S. A. Irfan and A. Shafie, Springer, Singapore, pp. 59-75, 2021.

Jeong, S. W., Kim, S. D. Aggregation and transport of copper oxide nanoparticles in porous media. Journal of Environmental Monitoring, 2009, 11(9): 1595-1600.

Ju, B., Fan, T. Experimental study and mathematical model of nanoparticle transport in porous media. Powder Technology, 2009, 192(2): 195-202.

Kazemzadeh, Y., Shojaei, S., Riazi, M., et al. Review on application of nanoparticles for EOR purposes: A critical review of the opportunities and challenges. Chinese Journal of Chemical Engineering, 2019, 27(2): 237-246.

Khalilinezhad, S. S., Cheraghian, G., Karambeigi, M. S., et al. Characterizing the role of clay and silica nanoparticles in enhanced heavy oil recovery during polymer flooding. Arabian Journal for Science and Engineering, 2016, 41(7): 2731-2750.

Khalilinezhad, S. S., Cheraghian, G., Roayaei, E., et al. Improving heavy oil recovery in the polymer flooding process by utilizing hydrophilic silica nanoparticles. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects, 2017, 1–10.

Kong, X., Ohadi, M. M. Applications of micro and nano technologies in the oil and gas industry-an overview of the recent progress. Paper SPE 138241 Presented at the Abu Dhabi International Petroleum Exhibition and Conference, Abu Dhabi, UAE, 1-4 November, 2010.

Lashari, N., Ganat, T. Emerging applications of NANOMATERIALS in chemical enhanced oil recovery: Progress and perspective. Chinese Journal of Chemical Engineering, 2020, 28(8): 1995-2009.

Li, S. An experimental investigation of enhanced oil recovery mechanisms in nanofluid injection process. Norway, Norwegian University of Science and Technology, 2016.

Lippmann, R. An introduction to computing with neural nets. IEEE Assp magazine, 1987, 4(2): 4-22.

Liu, X., O'carroll, D. M., Petersen, E. J., et al. Mobility of multiwalled carbon nanotubes in porous media. Environmental science and technology, 2009, 43(21): 8153-8158.

Maghzi, A., Kharrat, R., Mohebbi, A., et al. The impact of silica nanoparticles on the performance of polymer solution in presence of salts in polymer flooding for heavy oil recovery. Fuel, 2014, 123: 123-132.

Manning, R. K. A technical survey of polymer flooding projects. The University of Texas at Austin, Austin, 1983.

Mcdowell-Boyer, L. M., Hunt, J. R., Sitar, N. Particle transport through porous media. Water Resources Research, 1986, 22(13): 1901-1921.

Mendez, K. M., Pritchard, L., Reinke, S. N., et al. Toward collaborative open data science in metabolomics using Jupyter Notebooks and cloud computing. Metabolomics, 2019, 15(10): 1-16.

Mohaghegh, S., Ameri, S. Artificial neural network as a valuable tool for petroleum engineers. Paper SPE 29220 Presented at SPE Internatlonal Soclety of Petroleum Engheers, Texas, USA, January, 1995.

Murphy, M. J. Experimental analysis of electrostatic and hydrodynamic forces affecting nanoparticle retention in porous media. Austin, University of Texas at Austin, 2012.

Ogolo, N. A., Olafuyi, O. A., Onyekonwu, M. O. Enhanced oil recovery using nanoparticles. Paper SPE 160847 Presented at the SPE Saudi Arabia Section Technical Symposium and Exhibition, Al-Khobar, Saudi Arabia, 8-11 April, 2012.

Park, J. J., Lacerda, S. H., Stanley, S. K., et al. Langmuir adsorption study of the interaction of CdSe/ZnS quantum dots with model substrates: Influence of substrate surface chemistry and pH. Langmuir, 2009, 25(1): 443-450.

Pirizadeh, M., Alemohammad, N., Manthouri, M., et al. A new machine learning ensemble model for class imbalance problem of screening enhanced oil recovery methods. Journal of Petroleum Science and Engineering, 2021, 198: 108214.

Shaniv, D., Dror, I., Berkowitz, B. Effects of particle size and surface chemistry on plastic nanoparticle transport in saturated natural porous media. Chemosphere, 2021, 262: 127854.

Subasi, A., El-Amin, M. F., Darwich, T., et al. Permeability prediction of petroleum reservoirs using stochastic gradient boosting regression. Journal of Ambient Intelligence and Humanized Computing, 2020.

Van Den Doel, K., Robinson, M., Stove, C., et al. Subsurface temperature measurement using electromagnetic waves and machine learning for enhanced oil recovery. Paper Presented at 82nd EAGE Annual Conference and Exhibition, Amsterdam, Netherlands, 8-11 December, 2020.

Van Poollen, H. K. Fundamentals of Enhanced Oil Recovery. New York, USA, Elsevier, 1980.

Wang, Y., Li, Y., Pennell, K. D. Influence of electrolyte species and concentration on the aggregation and transport of fullerene nanoparticles in quartz sands. Environmental Toxicology and Chemistry: An International Journal, 2008, 27(9): 1860-1867.

Wasan, D. T., Nikolov, A. D. Spreading of nanofluids on solids. Nature, 2003, 423(6936): 156-159.

You, J., Ampomah, W., Sun, Q. Development and application of a machine learning based multi-objective optimization workflow for $CO_2$-EOR projects. Fuel, 2020, 264: 116758.

Yousef, A. M., Kavousi, G. P., Alnuaimi, M., at al. Predictive data analytics application for enhanced oil recovery in a mature field in the Middle East. Petroleum Exploration and Development, 2020, 47(2): 393-399.

Youssif, M. I., El-Maghraby, R. M., Saleh, S. M., et al. Silica nanofluid flooding for enhanced oil recovery in sandstone rocks. Egyptian Journal of Petroleum, 2018, 27(1): 105-110.

Zhang, T. Modeling of nanoparticle transport in porous media. Austin, The University of Texas at Austin, 2012.

Zhang, X. D. A Matrix Algebra Approach to Artificial Intelligence. Singapore, Springer, 2020.

Zhang, Y., Haghani, A. A gradient boosting method to improve travel time prediction. Transportation Research Part C: Emerging Technologies, 2015, 58: 308-324.

Zhou, K., Li, S., Zhou, X., et al. Data-driven prediction and analysis method for nanoparticle transport behavior in porous media. Measurement, 2021, 172: 108869.