Machine learning-based prediction of CO2 fugacity coefficients: Application to estimation of CO2 solubility in aqueous brines as a function of pressure, temperature, and salinity

Article *in* International Journal of Greenhouse Gas Control · September 2023
DOI: 10.1016/j.ijggc.2023.103971

5 authors, including:

Rupom Bhattacherjee
Oklahoma State University - Stillwater
5 PUBLICATIONS **4** CITATIONS
SEE PROFILE

Kodjo Botchwa y
Oklahoma State University - Stillwater
7 PUBLICATIONS **16** CITATIONS
SEE PROFILE

Goutam Chakraborty
Oklahoma State University - Stillwater
CITATIONS **42** PUBLICATIONS **607** CITATIONS
SEE PROFILE

Prem Bikkina
Oklahoma State University - Stillwater **157**
PUBLICATIONS **3,241**
SEE PROFILE

Machine Learning-Based Prediction of CO_2 Fugacity Coefficients: Application to Estimation of CO_2 Solubility in Aqueous Brines as a Function of Pressure, Temperature, and Salinity

Rupom Bhattacherjee[a,b], Kodjo Botchway[a], Jack C. Pashin[c], Goutam Chakraborty[a], Prem Bikkina[b]

[a] Spears School of Business, Oklahoma State University, Stillwater, OK, 74078, United States

[b] School of Chemical Engineering, Oklahoma State University, Stillwater, OK, 74078, United States

[c] Boone Pickens School of Geology, Oklahoma State University, Stillwater, OK, 74078, United States

Abstract

Fugacity is a fundamental thermodynamical property of gas and gas mixtures to determine their behavior and dynamics in complex systems. Fugacity can be deduced experimentally from the measurements of volume as a function of pressure at constant temperature or calculated iteratively using analytical equations of states (EOS). Experimental measurement is time-consuming, and analytical models based on EOS are computationally demanding, especially when an approximate but quick estimation is desired. In this work, machine learning (ML) is employed as a viable alternative to analytical EOSs for quick and accurate approximation of CO_2 fugacity coefficients. Five different ML algorithms are used to estimate the fugacity coefficients of pure CO_2 as a function of pressure ($\leq$ 2000 bar) and temperature ($\leq$ 1000 °C). A combination of experimental and pseudo-experimental (obtained from an analytical EOS) data of CO_2 fugacity coefficients is used to train, validate, and test the models. The best results were found using the Extreme Gradient Boosting algorithm, which showed a mean square error of only 0.0002 in the validation data and an average deviation of only 1.3% in the test data (pure prediction). To quantify the effectiveness of the machine learning techniques, results from the best-performing model are compared with two state-of-the-art analytical models. The ML model with significantly less computational complexity showed similar accuracy to the analytical models. The estimated fugacity data are then used to compute the CO_2 solubility in aqueous NaCl solution of different concentrations, and a maximum deviation of only 3.2% from the experimental data is observed.

1. Introduction

The fugacity of gas, often expressed by the fugacity coefficient (ratio of fugacity and pressure), is the pressure of the substance corrected for the non-ideality in its behavior (e.g., some level of interaction exists between gas molecules). The real gas pressure and fugacity are connected with fugacity coefficient, a dimensionless number that measures how far away the gas is from ideal conditions. When the fugacity coefficient is 1, there would be no interaction between the molecules, and the gas will behave as an ideal gas. If the fugacity coefficient is less than 1, molecules are attraction dominant; hence the effective pressure exerted by the gas molecules will be less than the ideal gas pressure. Similarly, fugacity coefficient greater than 1 indicates the

molecules are repulsion dominant, and the effective pressure is higher than the pressure exerted by the ideal gas molecules.

The term fugacity was first coined by Lewis (1908) to replace the mechanical partial pressure of gas or gas mixtures with effective partial pressure, and since then, it has been a very critical thermodynamical property of gas or a mixture of gases to compute their chemical equilibrium.

Fugacity is directly related to the chemical potential (μ) of the substances (Eqs. 1 and 2), and thus dictates the preference of the component for one phase over others. The differential change in the chemical potential between two states of slightly different pressure but equal temperature for a real gas can be explained by the ideal gas law if the pressure term is replaced by fugacity, as shown in equations 1 and 2. Readers are referred to the study by Hurai et al. (2015) to get an in-depth understanding of fugacity and fugacity coefficient.

$$(for\ ideal\ gas)\quad \int_{\mu_0}^{\mu} d\mu = \int_{P_o}^{P} V_m dP = \int_{P_o}^{P} \frac{RT}{P} dP = RT\ ln\ P/P_0 \qquad (1)$$

$$(for\ real\ gas) d\mu = RT \ln f/f_0 \qquad (2)$$

Where, R is the gas constant, V_m is fluid's molar volume, and P_0 and f_0 are reference pressure and fugacity, respectively.

Fugacity can be measured experimentally (Bruno, 1985; Frost and Wood, 1997) or estimated using different equations of state (EOS) (Duan et al., 1992; Holland and Powell, 1991; Spycher and Reed, 1988). There have been a number of empirical or semi-empirical EOSs developed to estimate the fugacity of gas in pure form or as a mixture with other fluids, such as Redlich-Kwong EOS (Redlich and Kwong, 1949), several modifications of Redlich-Kwong (de Santis et al., 1974; Flowers, 1979; Holloway, 1977), Peng-Robinson EOS (Peng and Robinson, 1976), and Virial EOS (Mason and Spurling, 1969).

Redlich and Kwong equation is an empirical Van-der-Waals type cubic equation that relates temperature, pressure, and volume of gases to estimate the thermodynamical properties of fluids. Several modifications of this equations were proposed to improve the estimation. However, the original Redlich-Kwong equation, along with some of its modifications, is reported to be less accurate in estimating fugacity values near critical conditions (Tarakad et al., 1979). The Peng-Robinson equations, another type of EOS devised to model gas fugacity, even though enables a more accurate estimation of fugacity in the liquid-vapor boundary than the Redlich-Kwong equations, they are more intricate in nature (Appelo et al., 2014). The only EOS with a better theoretical foundation to represent the properties of pure and mixed gases is the Virial equations (Mason and Spurling, 1969) which have been used extensively to estimate thermodynamical properties, including gas fugacity or fugacity coefficients (Bai et al., 2021; Chueh and Prausnitz, 1967; Dhamu et al., 2021; Duan et al., 1992; Schultz et al., 2010; Spycher and Reed, 1988). Spycher and Reed(1988) presented a second-order Virial EOS in terms of pressure and temperature to estimate the fugacity of pure and mixture of gases. Their model has the ability to be efficiently

implemented in other numerical models where pressure and temperature are the primary variables. Duan et al. (1992) formulated a fifth-order Virial expansion to estimate the fugacity coefficient of pure CO_2, CH_4, H_2O, and their mixtures. Comparison of their estimations with a large amount of experimental data for pure systems revealed that the EOS is capable of providing a very accurate estimation (deviation below 2.5%) of CO_2 fugacity coefficients for a wide range of temperatures (up to 1000 °C) and pressure (up to 3500 bar). Their fugacity EOS was later used by Duan and Sun (2003) and recently by Bhattacherjee et al. (2022) to estimate the CO_2 solubility in pure water and aqueous NaCl solution for geological storage applications. However, the EOS was presented in a very complex form and required a number of parameters to be evaluated.

Recently, machine learning or data-driven methods have become increasingly popular in various fields, and chemical engineering is no exception. Machine learning models are computationally less challenging to deploy than EOS-based models, and depending on the experimental data available to train the models, these can be used for any regression or classification problems with minimal error and less run-time requirement. Jirasek et al. (2020) developed a probabilistic matrix factorization model to predict the activity coefficient, a measure of the non-ideality of liquid mixtures. Their model had significantly less mean square error than UNIQUAC Functional-group Activity Coefficients (UNIFAC), one of the most conventional physical methods of predicting activity coefficients. Zhang et al. (2018) used a back-propagation neural network (BPNN) and a general regression neural network (GRNN) to provide an ultra-fast prediction method for the thermodynamic properties (e.g., solubility, density, and viscosity) of CO_2 in the solutions of Potassium Lysinate.

The computational power of Machine Learning also provides the ability to try different in-house algorithms on the same dataset, track their effectiveness, make necessary modifications, and select the most appropriate model: a trial-and-error roadmap not readily possible with the EOS models. Mohamadian et al. (2022) compared the performance of several machine learning algorithms, including extreme gradient boosting (XGB), multilayer perceptron (MLP), K-nearest neighbor (KNN), and internal genetic algorithm (GA) to estimate the solubility of CO_2 in the aqueous solution of NaCl as a function of pressure, temperature, and salinity. Abdolbaghi et al. (2019) applied four machine learning algorithms: particle swarm optimization (PSO), multilayer perceptron (MLP), hybrid-adaptive neuro-fuzzy inference system (hybrid-ANFIS), and coupled simulated annealing-least square support vector machine (CSA-LSSVM) to predict the viscosity of pure CO_2 at high temperature and pressure conditions. Machine learning has also been used in several other studies to estimate different thermodynamical and PVT properties of fluids such as viscosity (Amar et al., 2020), solubility (Menad et al., 2019; Mesbah et al., 2018; Nabipour et al., 2020), density (Lin and Seraj, 2022; Syah et al., 2021), diffusivities (Amar and Ghahfarokhi, 2020; Aniceto et al., 2021), and interfacial tension (Amooie et al., 2019; Safaei-Farouji et al., 2022; Vo-Thanh et al., 2022)

In this study, five different machine learning algorithms are used to develop models to estimate the fugacity coefficient of pure CO_2 as a function of temperature and pressure for the temperature

range of 0-1000 °C and pressure up to 2000 bar. Models are trained and validated on the experimental data collected by Angus et al. (1976) and Rhyzenko and Volkov (1971) and estimated data from Duan et al. (1992). The performance of the final model is tested on a separate dataset containing only experimental data, and the results are compared with two state-of-the-art thermodynamical models of estimating fugacity.

Predicted fugacity data are used to estimate the solubility of CO_2 in pure water and aqueous NaCl solutions using the solubility model developed by Duan and Sun (2003) at the temperature and pressure conditions usually reported in geological storage sites of CO_2. The original Duan and Sun model of solubility uses a fifth-order Virial EOS to estimate the CO_2 fugacity coefficients. This work intends to reduce the computational complexity of their model by estimating fugacity coefficients using machine learning frameworks. Such estimation can be used to understand the solubility trapping potential of CO_2 in depleted oil and gas reservoirs and saline aquifers.

2. Theory, Database, and Methods

The methodology used to develop the machine learning models for this study can be summarized in the following steps: (i) Database formation; (ii) learning algorithm selection; (iii) splitting data into training, validation, and test sets; (iv) data scaling; (v) hyper-parameter tuning; (vi) model evaluation; and (vii) selection of the best-performing model. Besides, predicted fugacity values are used to estimate the CO_2 solubility in pure water and aqueous NaCl solution using the solubility model developed by Duan and Sun (2003). Each of these steps is described in detail in the following subsections.

2.1 Database

The availability of experimental data on the CO_2 fugacity coefficient is very limited. Angus et al. (1976) reviewed and tabulated some available experimental PVT data for pure CO_2, including density, fugacity/pressure ratio (fugacity coefficient), and compressibility factor. However, these experimental data were limited to pressure up to 1000 bar only. Another great source of experimental data, provided by Rhyzenko and Volkov (1971) for CO_2 fugacity, also covers a shorter range of pressure, 80 to 1000 bar only.

This work aims to develop a model that can estimate the fugacity coefficients for temperatures up to at least 200 °C and pressure up to 2000 bar. These temperature and pressure ranges are selected based on the typical reservoir pressure and temperature encountered on the subsurface CO_2 storage sites. Temperature-wise, the experimental data are adequate to build the models, but pressure-wise, the data would not be enough to meet the objective of this study. Planning ahead of this scenario, to complement the experimental data, we chose to generate pseudo-experimental data of CO_2 fugacity coefficient for P>1000 bar using a current state-of-the-art analytical model developed by Duan et al. (1992).

The analytical model developed by Duan et al. (1992) is one of the most accurate thermodynamical models available to estimate the fugacity coefficients of pure CO_2. It covers an extensive range of pressure (up to 3500 bar) and temperature (up to 1000 °C). Moreover, CO_2 solubility in pure water and aqueous NaCl solutions estimated with their fugacity values reported to be very close to or within the experimental uncertainty (Bhattacherjee et al., 2022; Duan and Sun, 2003). Therefore, Duan's model was used to generate pseudo-experimental fugacity data for P >1000 bar. These estimated data were merged with the data from Angus et al. (1976) and Rhyzenko and Volkov (1971) to create a database of 640 data points for training, validating, and testing the models. The combined dataset covers a wide range of temperatures (up to 1000 °C) and pressures (up to 2000 bar). Table 1 shows the source, type, temperature, and pressure ranges of the data used to develop the database for fugacity prediction.

Table 1: Source, type, and T&P ranges of the data used to develop the database for this study.

Source	Data Type	Temperature	Pressure
Angus et al., 1976	Experimental	0-820 °C	1-1000 bar
Rhyzenko and Volkov, 1971	Experimental	400-1000 °C	800-1000 bar
Duan et al., 1992	Estimated	0-1000 °C	1000-2000 bar

2.2. Machine Learning Model Development and Optimization

2.2.1. Machine Learning Algorithms

This study used five different Machine Learning algorithms to predict the fugacity coefficients of CO_2. The algorithms employed were: Linear Regression (LR), Decision Tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGB), and different kernels (e.g., linear, polynomial, and Radial Based Function (RBF)) of Support Vector Machines (SVM). These algorithms were adopted using python libraries and packages for efficient utilization and further optimization. A description of the logic behind each model is presented as follows.

2.2.1.1 Linear Regression

Linear regression (LR) is one of the simplest machine learning models employed in predicting target values. The model makes a classification or regression calculation based on the value of a linear combination of features and their associated weights or parameters.

The algorithm is mathematically represented as:

$$y = \beta_0 + \beta_1 * x + ... + \beta_n x_n \quad (3)$$

where y is the set of output values from the algorithm, x is the set of input features fed into the model, and βs are the best parameters assigned to the features in such a way that the model prediction equation has the least amount of error between the predicted and actual target values.

To confirm the optimum selection of these parameters, we would need to use the training data and define a function that measures the quality of predictions for each value of β. This function is called the cost function. The cost function helps us to figure out the best possible values for β_0 and β_1, which would provide the best fit line for the data points. Since we want the best values for β_0 and β_1, we convert this search problem into a minimization problem where we would like to minimize the error between the predicted value and the actual value. The function is given as:

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred.y - y_i)^2 \quad (4)$$

where the idea is to minimize the sum of errors between the squared value of the difference between the predicted and actual values. The final selected β values would have the least cost function.

2.2.1.2. Decision Tree Regression

The decision tree (DT) is a supervised learning algorithm that builds the regressions or classification models in a tree-like structure based on decisions and all possible results and outcomes. The prediction of the target variable is followed by the tree, where the outputs at the individual nodes are determined, and these estimates further determine the branches. This modeling technique is generally preferred due to its ability to work well with data with missing or noisy data points without compromising the accuracy of estimation or prediction. It can also be ensembled like in random forest modeling to create even more efficient models.

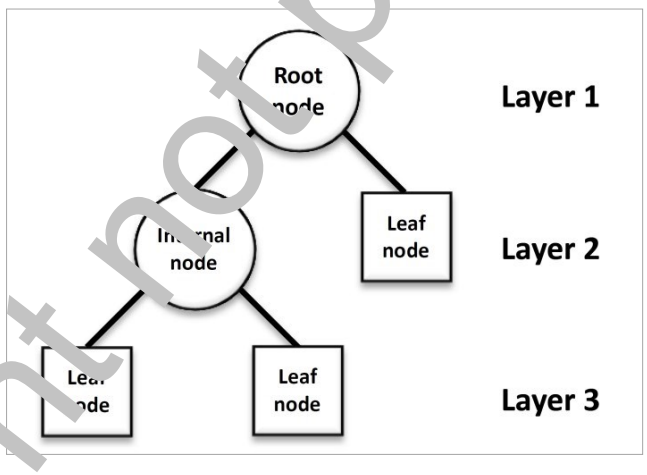

Figure 1: Basic decision tree (Hoffman, 2020).

The decision tree works to define the splits in the node such that the information gained from the resulting nodes is maximized. This information gain can be described as the net difference between the impurity in the root node and all the branching leaf nodes from that root, as seen in Figure 1. There are different criteria that can be used to mathematically determine this impurity difference: the entropy and the Gini index or the Gini impurity.

2.2.1.3. Random Forest Regression

The random forest (RF) regression model is an algorithm employed using ensemble learning, which essentially is a method that uses the combined predictions and estimations from multiple machine learning models to result in a more accurate regression of a target variable. This algorithm is based on constructing several trees in a particularly random manner and combining the predictions form the resulting models.

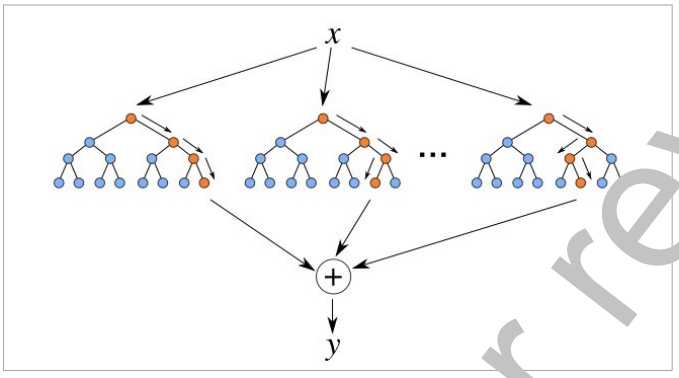

Figure 2: Random Forest Tree Regression illustration (Bakshi, 2020).

The model for a random forest prediction from this idea can be denoted by a base regression tree given as,

$$\{r_n(X, \Theta_m, D_n), m \geq 1\}, \quad (5)$$

where values for Θ are independently and identically distributed outputs of Θ based on the data set D_n and the independent variable, X.

Assuming multiple decision trees, the predictions from all the different ensembles using different hyperparameters are averaged, as shown in Figure 2.

The aggregated form of these random trees is estimated as

$$\{r_n(X, D_n) = E_\Theta[X, \Theta_m, D_n]\}, \quad (6)$$

where E_Θ is representative of the expectation of the random variable, dependent on X and the overall data, D_n. The averagely estimated prediction from all the models constitutes the random forest prediction value.

2.2.1.4. Extreme Gradient Boost

The principle behind the Extreme Gradient Boost (XGB) algorithm also follows the same principle as ensemble learning. XGB also trains many models to be able to arrive at an average prediction across all the models. The Boosting process performs in such a way that it identifies and minimizes

the disadvantages of the individual decision trees. In this, the prediction of a target variable y is given as,

$$y = \sum_{k=1}^{K} f_k(x_i), f_k \epsilon F \quad (7)$$

where for calculations, the K represents the number of trees in the ensemble model, the f_k is a function that maps the values of x to y in functional space F, and F is a set of possible Classification and Regression Trees. The objective function that we want to maximize or minimize in the prediction of the target variables is given as,

$$obj(\theta) = \sum_{i}^{n} l(y_i, y_{pi}) + \sum_{k=1}^{K} w(f_k) \quad (8)$$

where the first summation term is the training loss function which is the difference between the predicted target value from the model and the actual value. The second summation term also represents the complexity of the model in fitting the data. This function is defined by the regularization parameter used to ensure that the model is not overfitting or underfitting to the training datasets.

2.2.1.5. Support Vector Machines

The main ideas of support vector machines (SVM) are classification problems. However, the packages and adaptations made in different software enable them to handle regression problems properly and use the support vector regression algorithms. These adaptations are also contained in Python's "sklearn" machine learning packages. The SVM algorithm works in such a way that it finds a plane or, in this case, a hyperplane that can separate two sets of data points with the highest possible sense of purity (or, in this case, the separation margins). These respective hyperplanes can be expressed as $\pm y = \omega^T x + b$, the w term represents the slope of the plane. Then the objective function can change the optimization problem into:

$$(\omega^*, b^*) = argmax \frac{2}{||w^T||} y_i * (\omega^T x_i + b_i) \geq 1 \quad (9)$$

This simplification of the overall optimization parameter reduces the computations required to arrive at a global minimum. However, the solution for this is in a non-convex solution form, which is not preferred for this solution since the algorithm can get stuck at a local minimum instead of reaching the global minimum. Therefore, the non-convex equation is modified into a convex solution as,

$$(\omega^*, b^*) = argmin \frac{||w^T||^2}{2} y_i * (\omega^T x_i + b_i) \geq 1 \quad (10)$$

These approximation functions work for linear separators. However, the SVM package in the python language employs the use of kernels. These kernels are the functions in which the data points can be represented and would be separable. These could be linear, polynomial, or RBF (radial basis function).

2.2.2. Data Splitting

In machine learning, the dataset is split into a training and validation set to prevent the model from overfitting. The model is trained on the training data, and the validation set is never utilized during the training process. Instead, the validation set is used to evaluate the performance of the trained model on the unseen data and tune the model parameters accordingly. A separate test set is also used to score the data with the final model. The test set is different from the validation set in a way that it has never been used in tuning the model parameters and gives an unbiased estimate of the skill of the final model.

In this study, first, the test set is formed by randomly sampling 10% of the experimental data. The pseudo-experimental fugacity values estimated using the analytical model were not included in the test set. This was done so that the performance of the ML models could be compared with Duan's analytical model (Duan et al., 1992) on the test data. Because the pseudo-experimental data were derived from Duan's model, the comparison would be skewed in favor of Duan's model if the test set included the pseudo-experimental data.

Training and validation sets were created by randomly splitting 80% of the remaining data into the training set and 20 % into validation. The random state was kept constant to keep the dataset unaltered so that each model could be trained and validated on the same data. Table 2 summarizes the data used to develop the models in this study.

Table 2: Summary of the database used to train, validate, and test the models used in this study.

	Train			Validation			Test		
	T/°C	P/bar	Fugacity coefficient	T/°C	P/bar	Fugacity coefficient	T/°C	P/bar	Fugacity coefficient
count	461	461	461	115	115	115	64	64	64
mean	280.9	786.5	0.76	336.4	625.2	0.79	509.27	400	0.99
std	349.6	610.9	0.42	378.3	626.1	0.41	301.5	305	0.2
min	0	1	0.13	0	1	0.14	80	50	0.38
25%	40	200	0.37	50	50	0.39	320	100	0.95
50%	90	700	0.77	100	400	0.97	455	300	1.02
75%	500	1200	1.06	500	1000	1.01	805	600	1.12
max	1000	2000	1.82	1000	2000	1.8	1000	1000	1.28

2.2.3 Feature Scaling

Scaling is a very critical part of data pre-processing in machine learning and is used to bring all the features to the same standard so that the algorithm does not give any preference to any significantly large number (Burkov, 2019). Some of the machine learning algorithms that utilize

the distance between two observations to make decisions (e.g., SVM, principal component analysis, K-nearest neighbors) are very sensitive to the magnitude of the numbers and require scaling. Scaling is also required for algorithms that use gradient descent as the optimization technique, such as the case for linear regression, logistic regression, and neural networks. Rule based algorithms such as decision tree, random forest, or gradient-boosted decision tree, however, are not affected by scaling.

There are a number of ways scaling can be accomplished. The two most common methods are normalization and standardization of data. Normalization works by transforming the range of the values into a standard range, such as [0,1]. Standardization, on the other hand, transforms the data so that they follow a standard normal distribution with a mean of zero and a standard deviation of one. This study used standardization to scale the features using the StandardScaler feature of Python's Scikit-learn library (Pedregosa et al., 2011). The mean and median of the training set are used to scale the entire data.

2.2.4. Hyper-parameter Tuning

Hyper-parameters are different from the model parameters in the sense that they are not learned from the data fitted to the algorithm and must be defined prior to training the model. Hyper-parameter tuning is a model optimization technique that involves assigning different classes or numerical values to the parameters required to configure the learning algorithms and choosing a set of optimal hyper-parameters values to define the model architecture. In this work, optimal hyper-parameters were chosen by searching the hyper-parameters space for optimal values using the Grid Search technique (Pedregosa et al., 2011). Grid search evaluates different models developed with each possible combination of given hyper-parameter values and selects the one that produces the best results. This work uses GridSearchCV class from Python's Scikit-Learn library for hyper-parameter optimization with 3-fold cross-validation. Table 3 lists all the hyper-parameter values tested for each algorithm and the ones that produced the least validation error. Linear Regression (LR) model was exempted from the hyper-parameter tuning because the purpose of the LR model was just to set a base for the other models.

Table 3: Tested and best performing hyper-parameters values for each learning algorithm. Values for the remaining parameters were set to default.

ML Algorithm	Hyper-parameter	Values Tested	Best
SVM	C	1, 10, 100, 1000	100
	Gamma*	1, 0.1, 0.01, 0.001, scale	1
	Kernel	rbf, poly, linear	rbf
Decision Tree	Selection criterion	mse*, mae*	mae
	Split strategy	best, random	best
	Minimum number of samples to split a node	2, 3, 4….10	2

	Maximum depth of each tree	8, 10, 12….20	14
	Minimum number of samples per leaf	1, 2, 3….10	1
	Maximum number of leaf nodes	5, 20, 100, None	None
Random Forest	Number of trees used	100, 200, 300….1000	700
	Maximum number of features for split	auto, sqrt	auto
	Maximum depth of each tree	10, 20, 30....110	100
	Minimum number of samples to split a node	2, 5, 10	2
	Minimum number of samples per leaf	1, 2, 4	1
	Bootstrap	True, False	True
XGBoost	Number of trees used	100, 500, 1000	100
	Maximum depth of each tree	3, 6, 10	6
	Learning rate	0.01, 0.05, 0.1, 0.2, 0.3	0.3
	Fraction of features used to train each tree	0.3, 0.7, 1	1
	Gamma*	0, 1, 2	0
	Reg alpha*	0, 1, 2	0
	Reg lambda*	1, 2, 3	1
	Fraction of training samples used to train trees	0.1, 0.5, 1	1
	Tree construction algorithm	Exact, approx., hist	exact

*Gamma (SVM) determines how far the influence of a single training example reaches; C trades off the accuracy of the model for the simplicity of the decision function to avoid overfitting. Details on the hyper-parameters can be found in the Scikit-learn documentation (Pedregosa et al., 2011). For XGB, gamma defines the minimum loss reduction required to make a split; Reg alpha and Reg lambda are the L1 and L2 regularization terms, respectively (Chen et al., 2018).

*mse= mean squared error; mae= mean absolute error

2.3. Model Evaluation

The performances of the proposed models were evaluated using the mean squared error (MSE) in the training and validation data and the R^2 value in the validation data. Equations 11 and 12 are used to calculate MSE and R^2 values. In addition to that, predictions from the models are plotted against the validation data to graphically compare the performance of the models in estimating CO_2 fugacity coefficients.

$$MSE = \left(\frac{1}{n}\right) * \Sigma(actual - forecast)^2 \qquad (11)$$

$$R^2 = 1 - \frac{RSS}{TSS} \qquad (12)$$

Where,

actual = original or observed fugacity coefficients

forecast = Fugacity coefficients predicted using the developed models

n = number of observations
RSS= sum of squares of residuals, or the variability of the dataset explained by the model
TSS= Total sum of squares, or the total variability of the dataset

The best-performing model was then used to score the test data, and the results were compared with two classic thermodynamical models developed by Spycher and Reed (1988) and Duan et al. (1992) for CO_2 fugacity coefficients.

2.4. CO_2 Solubility Estimation

Predicted CO_2 fugacity data were used to estimate the solubility of CO_2 in different salinities of NaCl brine using the correlations developed by Duan and Sun (2003). The results were then compared with the available experimental data.

2.4.1. Solubility Correlations by Duan and Sun

Duan and Sun (2003) developed a set of thermodynamical correlations to estimate CO_2 solubility in water and NaCl brine of different salinities at temperatures ranging from 273–533 K and pressure from 0–200 MPa. According to their correlation, CO_2 solubility in brine or water can be estimated from the following equation:

$$ln\frac{y_{CO_2}P}{m_{CO_2}} = \frac{\mu_{CO_2}}{RT} - ln\phi_{CO_2} + \sum_c 2\lambda_{CO_2-c}m_c + \sum_a 2\lambda_{CO_2-a}m_a + \sum_c\sum_a \zeta_{CO_2-a-c}m_c m_a \qquad (13)$$

Where m_{CO_2} is the solubility of CO_2 expressed in moles of CO_2 per kg of water or brine, y_{CO_2} is the mole fraction of CO_2 in the vapor phase and can be estimated using equation (13), P is the pressure in bar, $\lambda\ and\ \zeta$ are second-order and third-order interaction parameters, respectively. $\frac{\mu_{CO_2}}{RT}$ is the dimensionless standard chemical potential, ϕ_{CO_2} is the fugacity potential of CO_2 in the vapor phase of CO_2-H_2O mixture, m is the molality of the brine (for pure water m=0), and a and c are the valence of anions and cations, respectively. Values for λ, ζ, and $\frac{\mu_{CO_2}}{RT}$ at different temperatures and pressure can be calculated using equation 15 and table 4. Values for the fugacity coefficient are estimated using the model developed in this study. It should be noted that, our model estimates the fugacity coefficient of pure CO_2, whereas equation 5 requires CO_2 fugacity coefficients in the vapor phase of the CO_2-H_2O mixture. However, the fugacity coefficient of pure CO_2 differs very little from that in the vapor phase of CO_2-H_2O mixture in the temperature and pressure range of this study (Duan et al., 1992). Therefore, fugacity coefficient data for solubility estimation can be predicted using the fugacity model developed in this study.

$$y_{CO_2} = \frac{P - P_{H_2O}}{P} \qquad (14)$$

$$Par\,(T,P) = c_1 + c_2T + \frac{c_3}{T} + c_4T^2 + \frac{c_5}{630-T} + c_6P + c_7PlnT + \frac{c_8P}{T} + \frac{c_9P}{630-T} + \frac{c_{10}P^2}{(630-T)^2} + c_{11}TlnP \quad (15)$$

In equation 14, P_{H_2O} is the pure water pressure in bar, which can be calculated using Eqn. (16) and Eqn. (17). Values for c_1 to c_{11} for different interaction parameters are listed in table 4.

$$P_{H_2O} = \frac{P_cT}{T_c}\left[1 + c_1(-t)^{1.9} + c_2t + c_3t^2 + c_4t^3 + c_5t^4\right] \quad (16)$$

$$t = \frac{T - T_c}{T_c} \quad (17)$$

T in Eqn. (15) to Eqn. (17) is the temperature in K, T_c and P_c are the critical temperature and pressure of water. Values for parameters c_1-c_5 are listed in table 5.

Table 4: T-P coefficient values for the interaction parameters in Eqn. (13) and Eqn. (15)

T-P coefficient	$\frac{\mu_{CO_2}}{RT}$	λ_{CO_2-Na}	$\zeta_{CO_2-Na-Cl}$
C_1	28.94	-0.41	3.36e-4
C_2	-0.04	6.08e-4	-1.98e-5
C_3	- 4770.67	97.53	
C_4	1.03e-5		
C_5	33.81		
C_6	9.04e-3		
C_7	-1.15e-3		
C_8	-0.31	-0.02	2.12e-3
C_9	-0.09	0.02	-5.24e-3
C_{10}	9.33e-4		
C_{11}		1.41e-5	

Table 5: Parameters for Eqn. (16)

C_1	-38.64
C_2	5.89
C_3	59.88
C_4	26.65
C_5	10.64

2.4.2. CO_2 Solubility Experimental Data

Model estimated CO_2 solubility data for NaCl brines of different salinities were compared with the experimental data. The sources of our collection of solubility data for NaCl brines are listed in Table 6. These sources have solubility values presented in different units such as molality, mole fraction, mass fraction, etc. These solubility units were all converted to molality (mol/kg) to have a common unit. A reliability assessment of the sources was performed by comparing the data from different sources at similar temperatures, pressure, and salinities. Data sourced from Ellis and Golding (1963) were excluded from the comparison as the data points were significantly off the trend. Data by Kiepe et al. (2002) were not used as their experimental data were higher than others at similar temperatures and pressure. Zhao et al. (2015) had the data only at 50 °C and 150 bar. Since this pressure, temperature, and salinities of their dataset were already covered by other measurements, their data were also discarded. The solubility data of Duan and Sun (2003) were excluded as those data were estimated rather than experimentally measured.

Table 6: Source of data collected for CO_2 solubility in NaCl brine of different salinities.

Source	T, °C	P, bar	Salinities, mol/kg
Bando et al. (2003)	30-60	100-200	0.2-0.5
Carvalho et al. (2015)	20-80	33-143	0.5-2
Duan and Sun (2003)	0-260	0-2000	0-4
Ellis and Golding (1963)	175-228	16-92	0-2
Hou et al. (2013)	50-100	30-182	4
Kiepe et al. (2002)	40-80	20-100	0.5-4.3
Koschel et al. (2006)	50-100	50-190	1-3
Liu et al. (2011)	45	21-158	1.9
Messabeb et al. (2016)	50-150	50-202	0-3
Mohammadian et al. (2022)	25-100	1-202	0-0.25
Nighswender et al. (1989)	80-160	20-100	0.17
Rumpf et al. (1994)	40-160	20-96	4
Takenouchi and Kennedy (1965)	150-200	100-1400	1.09-4.27
Wang et al. (2019)	30-80	30-300	1-3
Yan et al. (2011)	50-140	50-400	1-5
Zhao et al. (2015)	50	150	1-3

The final data set, comprising 890 experimental data points, covers pressure up to 1400 bar and temperature up to 200 °C. Out of the total 972 experimental data points collected, only 163 observations have $P \geq 200$ bar, and only 12 with $P \geq 1000$ bar. In fact, the most high-pressure CO_2 solubility data are from only four sources (Bando et al. (2003), Takenouchi and Kennedy (1965),

Yan et al. (2011), and Wang et al. (2019)) with only Takenouchi and Kennedy (1965) having the solubility data at pressure ≥1000 bar.

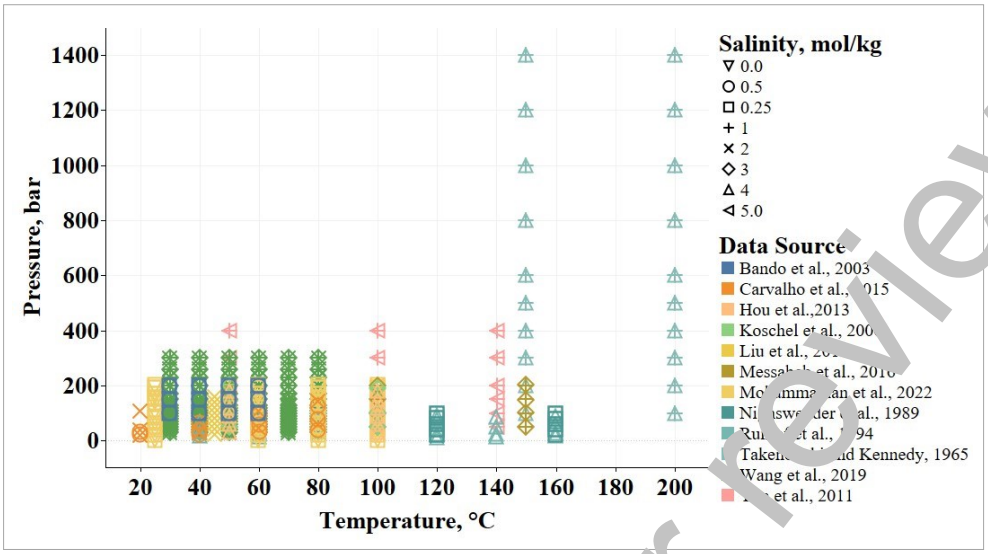

Figure 1: P and T range of accepted experimental CO_2 solubility data in NaCl brines of different salinities.

3. Results and Discussion

3.1. Machine Learning Model Selection

Five different supervised machine learning algorithms were used in this study to model CO_2 fugacity coefficient as a function of temperature and pressure: Linear Regression, Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF: averaging ensemble method), and Extreme Gradient Boosting (XGB: boosting ensemble method). The hyper-parameters of the algorithms were optimized using the grid search cv technique mentioned in section 2.2.4.

Table 7 compares the value of matrices used to evaluate the performance of the developed models, and figure 4 shows the plots for predicted fugacity coefficient values versus the actual values from the validation set. Note that the validation data set is not used during the training; hence the plots show pure prediction deviations from the actual values. Each developed model, except the naïve linear regression did a fair job approximating the CO_2 fugacity coefficients. The R^2 values were close to 1, and the mean square errors (MSE) were also very minimum. However, there were some instances of heavy overestimation and underestimation of fugacity coefficients from RBF SVM, DT, and RF models, as appears in figure 3. The best fit to the diagonal line was obtained from the XGB model. XGB model also produced the lowest validation MSE and highest R^2 value in the validation data. The remaining models ranked based on validation MSE, from lower to higher as: RF, DT, RBF SVM, and LR. The DT model even though exhibited the least MSE in the training data among the models, the difference between the training and validation MSE is too high to rely upon the model for final prediction. Hence, XGB is chosen as the best-performing model. Figure

5 shows the fugacity coefficient prediction from the XGB model and its comparison with the validation data.

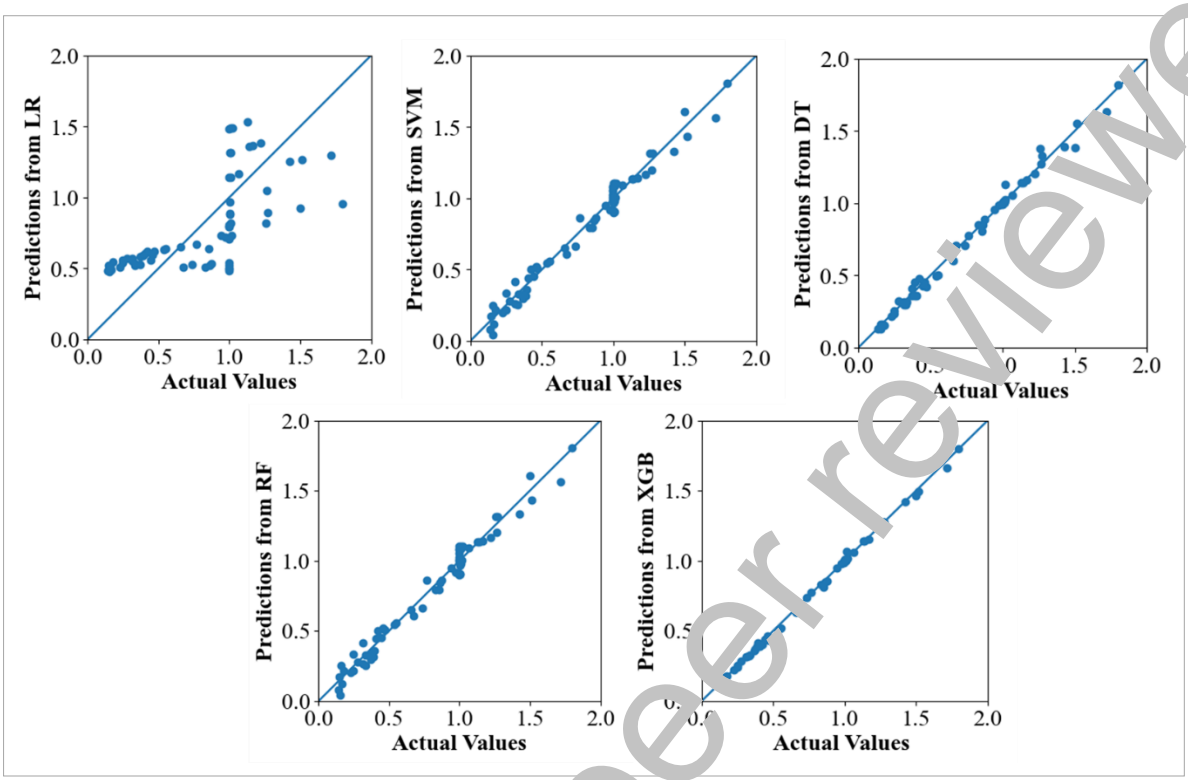

Figure 2: Actual versus predicted fugacity coefficients values for the ML models developed in this study.

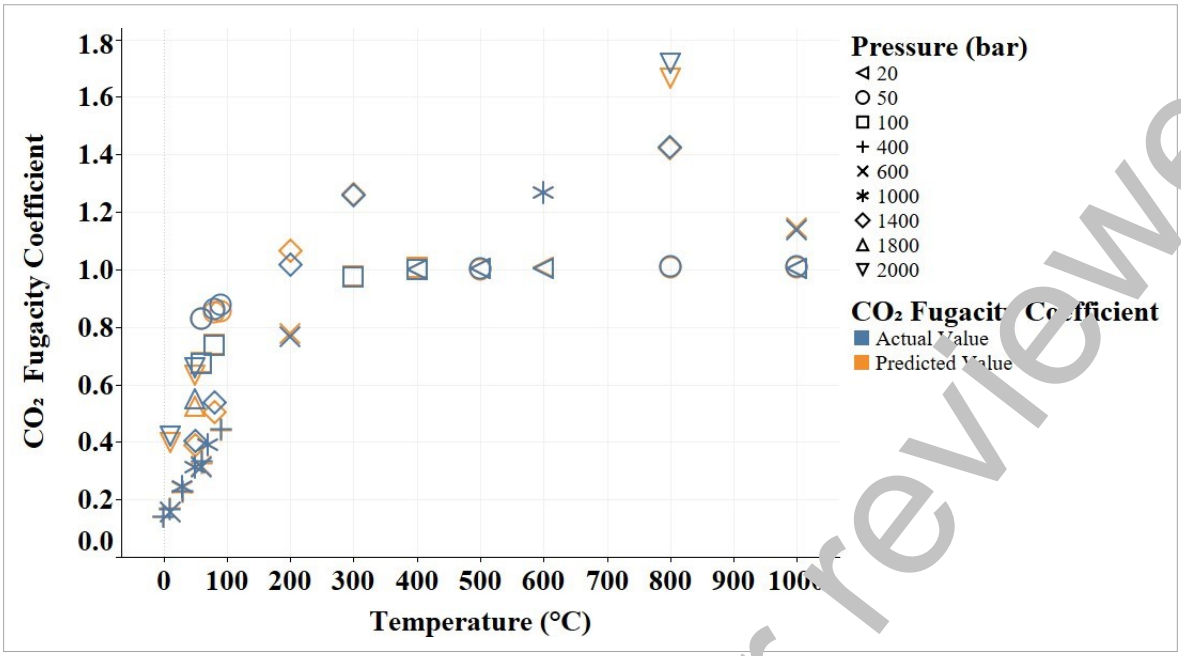

Figure 3: Prediction of CO_2 fugacity coefficient with the XGB model and its comparison with the actual data from validation set.

Table 7: Values of the evaluation matrices used to compare the machine learning models used in this study.

Model	Train MSE	Validation MSE	Validation R^2
Linear Regression	0.0729	0.0955	0.4297
RBF SVM	0.0035	0.0042	0.9748
Decision Tree	9.07×10^{-6}	0.0013	0.9923
Random Forest	0.0002	0.0008	0.9950
XGBoosting	2.02×10^{-5}	0.0002	0.9986

3.2. Comparison with Analytical Models

Figure 6 & Figure 7 compare the performance of the final proposed ML model (XGB) with two state-of-the-art analytical models developed by Spycher and Reed (1987) and Duan et al. (1992) for fugacity coefficients. The experimental data used in the comparison are from the test data set and have not been used in training or validating the models. Note that Spycher and Reed's model is applicable for temperature from 80 °C to 350 °C, up to 500 bars, and from 400 °C to 1000 °C,

up to 1000 bars. Duan's model has a broader range of temperature and pressure limit which is 0 to 1000 °C and 0 to 3500 bars, respectively. However, for the purpose of comparison, the pressure and temperature limit mutually shared by both models are selected, which is temperature from 0 °C to 1000 °C and pressure up to 1000 bars.

Overall, Spycher and Reed's model generates the fugacity values with the least deviation from the experimental values. Predictions from the XGB model and the estimations from Duan's model are slightly off at lower temperatures (<400 °C). The average deviation from the XGB model and Duan's model is 1.13% and 1.19%, respectively, whereas, for Spycher and Reed's model, the deviation is 0.78%. A slightly lower deviation from the Spycher and Reed's model might be due to the fact that the estimated values from Spycher and Reed's model are extracted from the plots reported in their studies using a plot-digitizing software. Even though the method is straightforward, a lack of precision can lead to some level of data extraction errors. Nevertheless, the deviations from the XGB model were minor, and with significantly less computational complexity it performed equally well as the two EOS-based analytical models.

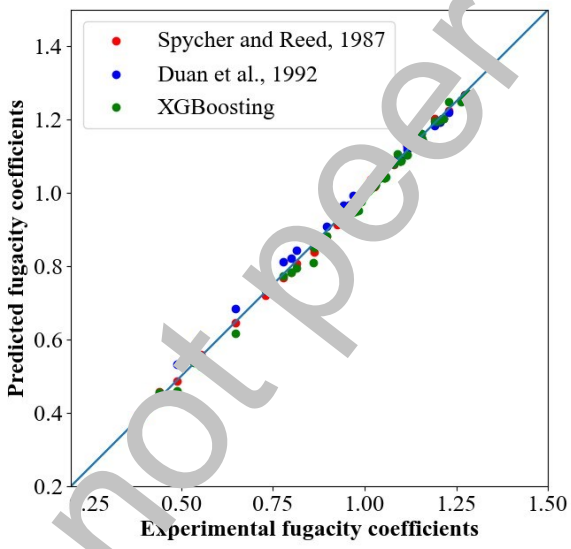

Figure 4: Experimental vs. predicted fugacity coefficients for test data.

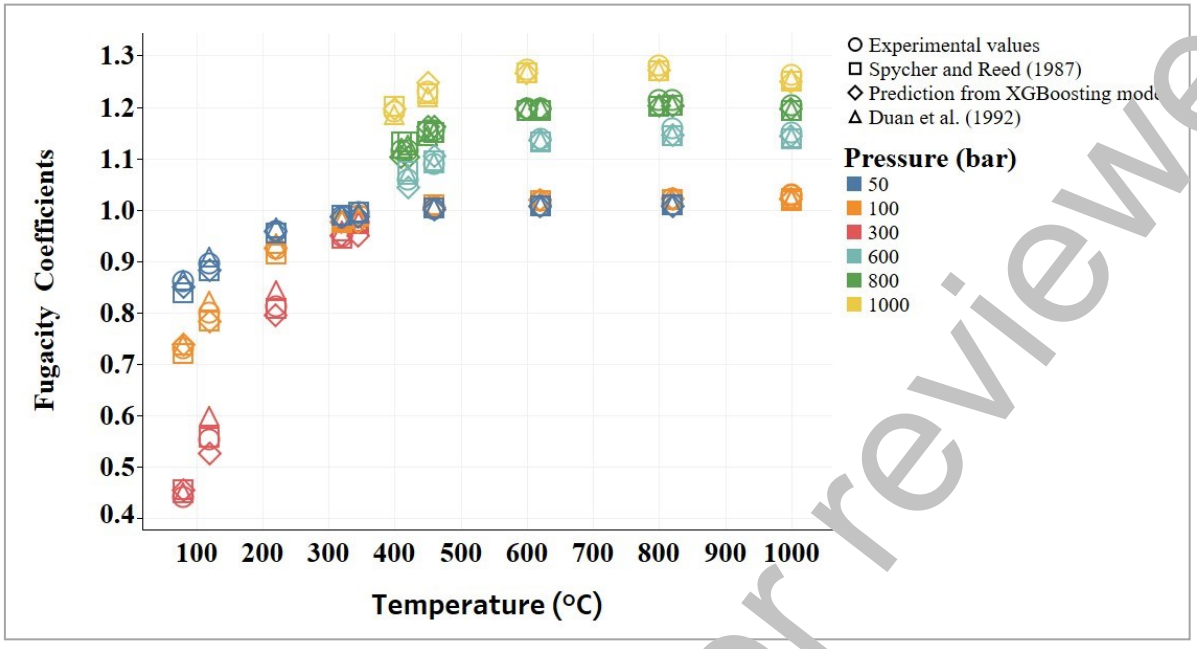

Figure 5: Comparison of the experimental data with the predictions from XGB model and two analytical models by Spycher and Reed (1987) and Duan et al. (1992).

3.3. Solubility Prediction

Fig. 8 (A, B, and C) compares the estimated CO_2 solubility data for different salinities of brine with the experimental data at 30 °C, 60 °C, and 80 °C, respectively. Due to the overall scarcity of experimental data at high pressure and temperature conditions, it was challenging to find enough isothermal data points (especially for T ≥ 100 °C) representing different salinities. Therefore, to compare the estimated solubility data at temperature ≥ 100 °C, data from different salinities (0-5 mol/kg) and temperatures (100-200 °C) are combined (Fig. 8D).

The estimated solubilities data are overall in good agreement with the available experimental data, indicating that fugacity coefficient values predicted from the XGB model are reliable for CO_2 solubility estimation for at least up to 200 °C, 1400 bar, and 5 molal NaCl solution. The average deviation from the experimental data was 2.08%, 2.03%, 1.5%, and 3.2% for 30 °C, 60 °C, 80 °C, and 100-200 °C, respectively.

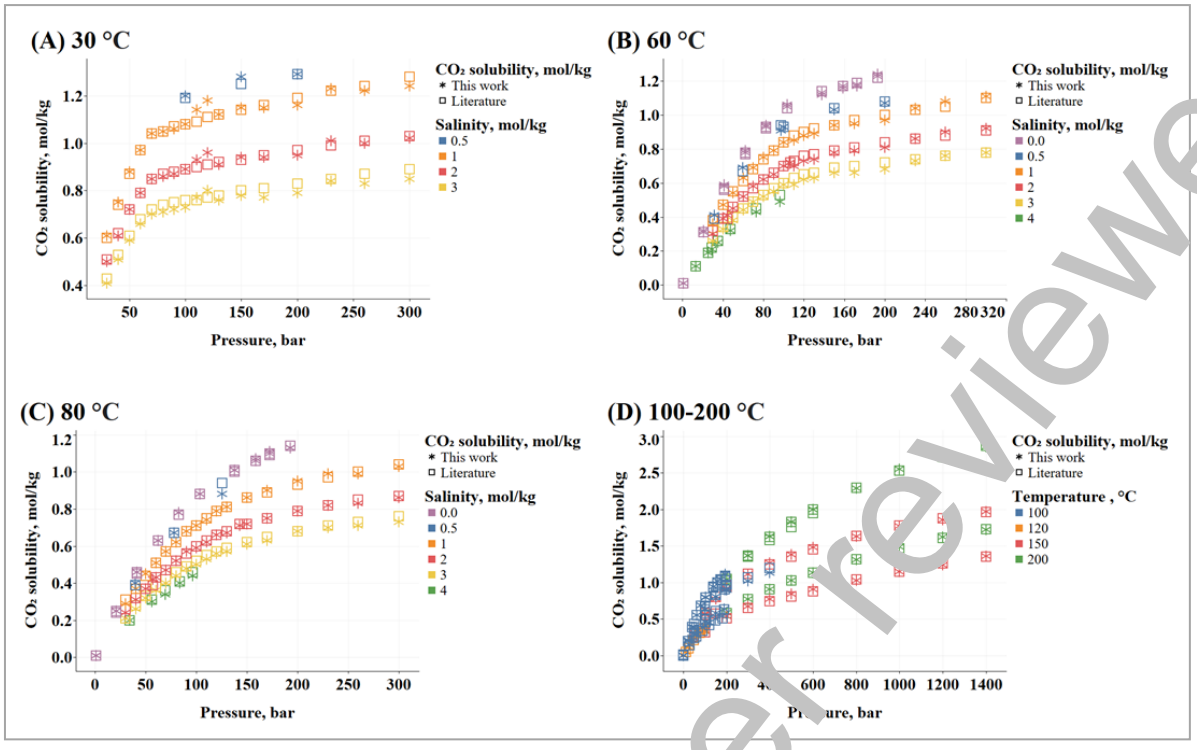

Figure 6: Comparison of estimated and experimental CO_2 solubility values at different pressures and temperatures

4. Conclusions

In this work, a machine learning approach for the prediction of CO_2 fugacity coefficient is developed to estimate CO_2 fugacity coefficient with similar accuracy but significantly lesser computational complexity than EOS based analytical models. Five different learning algorithms (Multilinear Regression, Support Vector Machine, Decision Tree, Random Forest, and Extreme Gradient Boost) are used to estimate the fugacity coefficient as a function of pressure and temperature. Extreme Gradient Boost model provided the prediction with the highest accuracy in the validation data. The developed model can be used to estimate CO_2 fugacity coefficient for temperature in the range of 0 to 1000 °C and pressure up to 2000 bars. The comparison between the ML model and the EOS-based analytical model suggest that the proposed model can be used as a substitution for the analytical models where a quick and approximate estimation of CO_2 fugacity coefficient is required.

The estimated fugacity coefficients are used to compute CO_2 solubility, one of the major applications of fugacity data, in NaCl brines of different salinities. The maximum average deviation from the experimental data ranged from 2.08 % to 3.2 % for pressures up to 1400 bar, temperature up to 200 °C, and concentrations up to 5 molal NaCl solution. The model developed, stored data with predicted fugacity, and the codes required to make the prediction for fugacity coefficients are provided in the Supplementary Material section.

5. Acknowledgements

The National Energy Technology Laboratory (NETL) of the United States Department of Energy (DOE) provided funding (Award No. DE-FE0031557) for this study through the Southern States Energy Board. Cost share supporting this research was supplied by the project partners, including the SAS Institute. This paper is based upon work supported by the Department of Energy and was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed or represented that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendations, or favoring by the United States Government or any agency thereof. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

6. References

Amar, M.N. and Ghahfarokhi, A.J., 2020. Prediction of CO_2 diffusivity in brine using white-box machine learning. Journal of Petroleum Science and Engineering, 190: 107037.

Amar, M.N. et al., 2020. Modeling viscosity of CO_2 at high temperature and pressure conditions. Journal of Natural Gas Science and Engineering, 77: 103271.

Amooie, M.A. et al., 2019. Data-driven modeling of interfacial tension in impure CO_2-brine systems with implications for geological carbon storage. International Journal of Greenhouse Gas Control, 90: 102811.

Angus, S., Armstrong, B. and De Reuck, K., 1976. Carbon dioxide: International thermodynamic tables of the fluid state-3. Pergamon Press, New York, 3: 266.

Aniceto, J.P., Zêzere, B. and Silva, C.M., 2021. Machine learning models for the prediction of diffusivities in supercritical CO_2 systems. Journal of Molecular Liquids, 326: 115281.

Appelo, C., Parkhurst, D.L. and Post, V., 2014. Equations for calculating hydrogeochemical reactions of minerals and gases such as CO_2 at high pressures and temperatures. Geochimica et Cosmochimica Acta, 125: 49-67.

Bai, J., Zhang, P., Zhou, C.-W. and Zhao, H., 2021. Theoretical studies of real-fluid oxidation of hydrogen under supercritical conditions by using the virial equation of state. Combustion and Flame, 111945.

Bakshi, G., 2020. Random Forest Regression, https://levelup.gitconnected.com/.

Bhattacharjee, R. et al., 2022. Evaluating CO_2 Storage Potential of Offshore Reservoirs and Saline Formations in Central Gulf of Mexico by Employing Data-driven Models with SAS® Viya, Western Users of SAS Software, San Francisco.

Bruno, T.J., 1985. An apparatus for direct fugacity measurements on mixtures containing hydrogen. Journal of Research of the National Bureau of Standards, 90(2): 127.

Burkov, A., 2019. The hundred-page machine learning book, 1. Andriy Burkov Quebec City, QC Canada.

Chen, T., He, T. and Benesty, M., 2018. XGBoost documentation.

Chueh, P. and Prausnitz, J., 1967. Vapor-liquid equilibria at high pressures. Vapor-phase fugacity coefficients in nonpolar and quantum-gas mixtures. Industrial & Engineering Chemistry Fundamentals, 6(4): 492-498.

Dhamu, V., Thakre, N. and Jana, A.K., 2021. Structure-H hydrate of mixed gases: Phase equilibrium modeling and experimental validation. Journal of Molecular Liquids, 343: 117605.

Duan, Z., Møller, N. and Weare, J.H., 1992. An equation of state for the CH_4-CO_2-H_2O system: I. Pure systems from 0 to 1000 C and 0 to 8000 bar. Geochimica et Cosmochimica Acta, 56(7): 2605-2617.

Duan, Z. and Sun, R., 2003. An improved model calculating CO_2 solubility in pure water and aqueous NaCl solutions from 273 to 533 K and from 0 to 2000 bar. Chemical geology, 193(3-4): 257-271.

Frost, D.J. and Wood, B.J., 1997. Experimental measurements of the fugacity of CO_2 and graphite/diamond stability from 35 to 77 kbar at 925 to 1650 C. Geochimica et Cosmochimica Acta, 61(8): 1565-1574.

Hoffman, K., 2020. Decision Tree Hyperparameters Explained, ken-hoffman.medium.com.

Holland, T. and Powell, R., 1991. A Compensated-Redlich-Kwong (CORK) equation for volumes and fugacities of CO_2 and H_2O in the range 1 bar to 50 kbar and 100–1600 C. Contributions to Mineralogy and Petrology, 109(2): 265-273.

Hurai, V., Huraiová, M., Slobodník, M. and Thomas, R., 2015. Chapter 6 - Fluid Thermodynamics. In: V. Hurai, M. Huraiová, M. Slobodník and R. Thomas (Editors), Geofluids. Elsevier, pp. 171-230.

Jirasek, F. et al., 2020. Machine learning in thermodynamics: Prediction of activity coefficients by matrix completion. The journal of physical chemistry letters, 11(3): 981-985.

Lewis, G.N., 1908. The osmotic pressure of concentrated solutions, and the laws of the perfect solution. Journal of the American Chemical Society, 30(5): 668-683.

Lin, T. and Seraj, A., 2022. Evolving Machine Learning Methods for Density Estimation of Liquid Alkali Metals over the Wide Ranges. International Journal of Chemical Engineering, 2022.

Mason, E.A. and Spurling, T.H., 1969. The virial equation of state, 2. Pergamon.

Menad, N.A., Hemmati-Sarapardeh, A., Varamesh, A. and Shamshirband, S., 2019. Predicting solubility of CO_2 in brine by advanced machine learning systems: Application to carbon capture and sequestration. Journal of CO_2 Utilization, 33: 83-95.

Mesbah, M., Shahsavari, S., Soroush, E., Rahaei, N. and Rezakazemi, M., 2018. Accurate prediction of miscibility of CO_2 and supercritical CO_2 in ionic liquids using machine learning. Journal of CO_2 Utilization, 25: 99-107.

Mohammadian, E., Liu, B. and Riazi, A., 2022. Evaluation of Different Machine Learning Frameworks to Estimate CO_2 Solubility in NaCl Brines: Implications for CO_2 Injection into Low-Salinity Formations. Lithosphere, 2022(Special 12): 1615832.

Nabipour, N., Mosavi, A., Baghban, A., Shamshirband, S. and Felde, I., 2020. Extreme learning machine-based model for Solubility estimation of hydrocarbon gases in electrolyte solutions. Processes, 8(1): 92.

Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12: 2825-2830.

Peng, D.-Y. and Robinson, D.B., 1976. A new two-constant equation of state. Industrial & Engineering Chemistry Fundamentals, 15(1): 59-64.

Redlich, O. and Kwong, J.N., 1949. On the thermodynamics of solutions. V. An equation of state. Fugacities of gaseous solutions. Chemical reviews, 44(1): 233-244.

Safaei-Farouji, M. et al., 2022. Application of robust intelligent schemes for accurate modelling interfacial tension of CO_2 brine systems: Implications for structural CO_2 trapping. Fuel, 319: 123821.

Schultz, A.J., Shaul, K.R., Yang, S. and Kofke, D.A., 2010. Modeling solubility in supercritical fluids via the virial equation of state. The Journal of Supercritical Fluids, 55(2): 479-484.

Spycher, N.F. and Reed, M.H., 1988. Fugacity coefficients of H_2, CO_2, CH_4, H_2O and of H_2O-CO_2-CH_4 mixtures: A virial equation treatment for moderate pressures and temperatures applicable to calculations of hydrothermal boiling. Geochimica et Cosmochimica Acta, 52(3): 739-749.

Syah, R. et al., 2021. Implementation of artificial intelligence and support vector machine learning to estimate the drilling fluid density in high-pressure high-temperature wells. Energy Reports, 7: 4106-4113.

Tarakad, R.R., Spencer, C.F. and Adler, S.B., 1979. A comparison of eight equations of state to predict gas-phase density and fugacity. Industrial & Engineering Chemistry Process Design and Development, 18(4): 726-739.

Vo-Thanh, H., Amar, M.N. and Lee, K.-K., 2022. Robust machine learning models of carbon dioxide trapping indexes at geological storage sites. Fuel, 316: 123391.

Zhang, Z., Li, H., Chang, H., Pan, Z. and Luo, X., 2018. Machine learning predictive framework for CO_2 thermodynamic properties in solution. Journal of CO_2 Utilization, 26: 152-159.