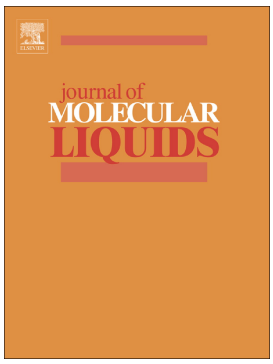

Machine learning models for the prediction of diffusivities in supercritical CO2 systems

José P.S. Aniceto, Bruno Zêzere, Carlos M. Silva

PII: S0167-7322(21)00007-6

DOI: https://doi.org/10.1016/j.molliq.2021.115281

Reference: MOLLIQ 115281

To appear in: *Journal of Molecular Liquids*

Received date: 30 November 2020

Revised date: 30 December 2020

Accepted date: 3 January 2021

Please cite this article as: J.P.S. Aniceto, B. Zêzere and C.M. Silva, Machine learning models for the prediction of diffusivities in supercritical CO2 systems, *Journal of Molecular Liquids* (2021), https://doi.org/10.1016/j.molliq.2021.115281

Machine learning models for the prediction of diffusivities in supercritical CO_2 systems

José P.S. Aniceto, Bruno Zêzere, and Carlos M. Silva

CICECO, Department of Chemistry, University of Aveiro, 3810-193 Aveiro, Portugal

Abstract

The molecular diffusion coefficient is fundamental to estimate dispersion coefficients, convective mass transfer coefficients, etc. Since experimental diffusion data is scarce, there is significant demand for accurate models capable of providing reliable diffusion coefficient estimations.

In this work we applied machine learning algorithms to develop predictive models to estimate diffusivities of solutes in supercritical carbon dioxide. A database of experimental data containing 13 properties for 174 binary systems totaling 4917 data points was used in the training of the models. Five machine learning algorithms were evaluated and the results were compared with three commonly used classic models.

The best results were found using the Gradient Boosted algorithm which showed an average absolute relative deviation (AARD) of 2.58 % (pure prediction). This model has five parameters: temperature, density, solute molar mass, solute critical pressure and solute acentric factor. For the same dataset, the classic Wilke-Chang equation showed AARD of 12.41 %.

1 Introduction

The knowledge of transport properties is required for the design, simulation and scale-up of rate controlled separations and chemical reactions. The binary diffusivities at infinite dilution, D_{12}, are fundamental to estimate important quantities like dispersion coefficients, convective mass transfer coefficients, and catalysts efficiency factors [1–3]. Over the past few years the so-called "green solvents" have been gaining more attention in both academia and industry, and from them we may detach supercritical carbon dioxide (SC-CO_2) [4, 5]. This solvent has been extensively used in

supercritical extraction (SFE), namely for extraction of compounds from vegetable matrices [5]. In this context, diffusivity coefficients data become extremely important. However, the knowledge of D_{12} in SC-CO_2 is still limited in terms of solutes and operating conditions, requiring accurate models capable of interpolating D_{12} and also predicting this property when no data is available [6].

Among the most well known models one can cite the hydrodynamic equation of Wilke-Chang [7], published in 1955, which is still widely used due to its simplicity since it only requires information on solvent viscosity, solute molecular mass, solute volume at normal boiling point and operating conditions. This equation has also been modified in several occasions giving rise, for instance, to the Lai-Tan equation [8], an empirical modification of the first one, specifically devised for SC-CO_2 systems. Furthermore, some other hydrodynamic based equations have been proposed and published for D_{12} estimation in SC-CO_2, which will not be addressed in this work but can be found elsewhere [9, 10]. Regarding correlative models, one can cite the 2-parameter correlation of Dymond-Hildebrand-Batschinski (DHB) [11–13], based on free-volume theory, which is specially useful when some data of a given system is available allowing to both interpolate and extrapolate data for the desired condition.

Recently Artificial Intelligence has been applied in several fields of chemical engineering, for instance, for the estimation of physical properties of various compounds. Artificial neural network (ANN) models have been used to calculate the diffusion coefficients of pure compounds in water [14, 15]. Vaz *et al.* [16] proposed an ANN based correlation for the estimation of self-diffusion coefficients as function of residual entropy and a molecular chain length parameter, which provided an average absolute relative deviation of 9.13 % for a large database with molecular dynamic and experimental values for hard-sphere, Lennard-Jones, hard-sphere chain, and real fluids composed of polar, nonpolar, symmetrical and asymmetrical molecules. Feed forward neural networks have also been developed to estimate the Fick diffusion coefficient in binary liquid systems [17]. Likewise, Eslamloueyan and Khademi [18] proposed a method based on a feed forward three-layer neural network to predict binary diffusion coefficient of gases at atmospheric pressure based on the critical temperature, critical volume and molecular weight of each component in the mixture. A genetic function approximation (GPA) derived model containing five parameters has been proposed to predict diffusion coefficient of non-electrolyte

organic compounds in air at 25 °C [19]. The estimation of the binary diffusion coefficients of liquid hydrocarbons at infinite dilution and in concentrated solutions has also been accomplished with Multi-layer perceptron (MLP) neural networks and an adaptive neuro-fuzzy inference system (ANFIS) [20].

In this work we applied machine learning algorithms, such as decision tree, nearest neighbors and ensemble methods to develop models for the prediction of binary diffusion coefficients of supercritical CO_2 systems. A large database of experimental data, covering small and large, polar and nonpolar solute molecules, was collected and used in the training of the models. The results were compared with the Wilke-Chang, Lai Tan and DHB equations, extensively used for this purpose.

2 Theory, Database and Methods

The methodology used in this work to develop machine learning (ML) models for the prediction of diffusivities can be summarized in the following steps: (i) variable (feature) selection; (ii) learning algorithms selection; (iii) data splitting into training and testing sets; (iv) data scaling; (v) hyper-parameters optimization; and (vi) final model evaluation. These steps are detailed below. The ML models were compared with the classic models of Wilke-Chang [7], Lai-Tan [8] and Dymond-Hildebrand-Batschinski [11–13] shown in Section 2.4.

2.1 Database

The database used in this work has been updated and extended from the one previously published by Vaz *et al.* [10]. It is composed by 174 binary SC-CO_2 systems (solvent/solute) totaling 4917 data points and contains information on 13 properties, as shown in Table 1. It covers a wide range of temperatures and pressures: 283.15–398.15 K and 67–3500 bar, respectively.

Supercritical CO_2 densities were computed by the correlation of Pitzer and Schreiber [21] when they were not provided by the authors. The viscosities of SC-CO_2 were estimated by the correlation of Altunin and Sakhabetdinov [22] whenever necessary. The solute molar volumes at normal boiling point were estimated by Tyn–Calus equation [23] (Equation 4). The missing critical constants were estimated by Joback [23–25], Somayajulu [26], Klincewicz [27, 28], Ambrose [27, 29], and Constantinou-Gani [30] methods. The acentric factors, when not available, were estimated by the Lee-Kesler [31] and Pitzer [32] equations. The Lennard-Jones diameter and

energy were taken from Silva and Liu [13] and when not available were estimated by Equations 8 and 9 from Liu *et al.* [33].

Detailed information on the database used, including pure compound properties and reduced temperature and pressure ranges, is presented in the Supplementary Material Table SM1.

Table 1: Properties and variables available in the database of diffusivities of several solutes in SC-CO_2.

Property	Units	Description
D_{12}	$cm^2\ s^{-1}$	Diffusion coefficient
T	K	Temperature
P	bar	Pressure
ρ_1	$g\ cm^{-3}$	Solvent density
μ_1	cP	Solvent viscosity
M_2	$g\ mol^{-1}$	Molar mass of solute
$T_{c,2}$	K	Critical temperature of solute
$T_{bp,2}$	K	Boiling point temperature of solute
$P_{c,2}$	bar	Critical pressure of solute
$V_{c,2}$	$mol\ cm^{-3}$	Critical volume of solute
w_2	-	Acentric factor of solute
$\sigma_{LJ,2}$	Å	Lennard-Jones diameter of solute
$\varepsilon_{LJ,2/k_B}$	K	Lennard-Jones energy of solute

2.2 Machine learning model development and optimization

Model features selection: Model features were selected from the properties and variables available in the database presented in Table 1. Features were excluded from the model until no collinearity above a defined threshold of 0.65 was present. To select the variable to exclude from a pair of collinear variables, the correlation with D_{12} and the ease of obtaining that variable were taken into account. Variables with low correlation with diffusivity were also removed.

Training and testing sets: In Machine Learning, data is usually divided into a training set, used for learning and fitting of the model, and a testing set, used to evaluate the fitted model after learning. Information from the testing set is never utilized during learning. Training and testing data sets were created by randomly splitting data base points 70 % into a training set and 30 % into testing sets. These data sets were kept unaltered for the evaluation of all models, guaranteeing the same input data for all.

Scaling: Feature scaling is usually beneficial to most learning algorithms as it often improves model robustness and training speed [34]. Scaling is accomplished through normalization or standardization of the features. Normalization consists in transforming the real range of values into a standard range (*e.g.* [0,1] or [−1,1]). Standardization consists in scaling variables so that they follow a standard normal distribution (mean of zero and standard deviation of one). In this work, properties were normalized to the [0,1] range using scikit-learn MinMaxScaler.

Hyper-parameter optimization: Unlike model parameters, which are fitted to the data when a model is trained, hyper-parameters are not learned from data and must be defined before training. They are configuration options of a given learning algorithm, usually with a numerical value, that influence how the model behaves. In this work, hyper-parameters were optimized by grid search with 4-fold cross-validation (using scikit-learn *GridSearchCV*). This method performs an exhaustive test of all hyper-parameters in a previously defined grid and evaluates the resulting model performance *via* k-fold cross-validation. The cross-validation technique avoids further reduction of the training set to create a validation set. Instead, the training set is split into k subsets and the model is trained using data from $k-1$ of the folds and tested on the remaining data. This is repeated using each $k-1$ combination of folds for training and the best hyper-parameters are those of the model with the best average performance. The tested hyper-parameters for each learning algorithm used, as well as the best hyper-parameters, are shown in Table 2.

Table 2: Tested and best hyper-parameter values for each machine learning algorithm. All remaining hyper-parameters were left at their default values.

ML Algorithm	Hyper-parameter	Values Tested	Best

k -Nearest Neighbors	Number of neighbors	3; 4; 5; 6; 7; 10; 12; 15	3
	Algorithm	auto; ball_tree; kd_tree; brute	kd_tree
	Leaf size (BallTree or KDTree algorithm)	1; 2; 3; 5; 10; 15; 30	1
	Weight function	uniform; distance	distance
Decision Tree	Quality of a split metric	mse; mae	mae
	Split strategy	best; random	best
	Minimum number of samples per leaf	0.1; 1; 2; 5	1
	Minimum number of samples to split a node	0.1; 2; 4	2
	Minimum weighted fraction required for leaf node	0; 0.1; 0.5	0
	Maximum number of features for split	auto; sqrt; log2; None	auto
	Minimum impurity decrease	0; 0.5; 2	0
Random Forest	Quality of a split metric	mse; mae	mae
	Number of estimators	5; 10; 15; 20; 30; 50; 100; 150	15
	Minimum number of samples per leaf	0.1; 1; 2; 5	1
	Minimum number of samples to split a node	0.1; 2; 4	4
	Minimum weighted fraction required for leaf node	0; 0.1; 0.5	0
	Maximum number of features for best split	auto; sqrt; log2; None	None

	Minimum impurity decrease	0; 0.5; 2	0
Gradient Boosted	Loss function	ls; lad; huber	lad
	Number of trees used in the boosting process	100; 500; 900; 1100; 1500	500
	Maximum depth of each tree	2; 3; 5; 10; 15	10
	Minimum number of samples per leaf	1; 2; 4; 6; 8	10
	Minimum number of samples to split a node	2; 4; 6; 10	1
	Maximum number of features for split	auto; sqrt; log2; None	sqrt

mse: mean squared error; mae: mean absolute error; ls: least square regression; lad: least absolute deviation; huber: a combination of ls and lad.

2.3 Machine Learning algorithms

Five ML algorithms were evaluated for the prediction of binary diffusivities: a Multilinear Regression, a k-Nearest Neighbors model, a Decision Tree algorithm, and two Ensemble Methods (Random Forest and Gradient Boosted). All models were implemented using the Python machine learning library scikit-learn version 0.22.1 [35]. In the following a brief description of each one is presented.

Multilinear Regression: A simple Ordinary Least Squares Multilinear Regression was used as a baseline model for the prediction of binary diffusivities. In a linear regression model, the real value, y, is a linear combination of features X_i weighted by coefficients b_i:

$$Y = b_0 + \sum_{i=1}^{p} X_i b_i \tag{1}$$

where Y is the predicted output, b_0 is the intercept or bias term, X_i are the input variables, b_i are the model coefficients, and p represents the number of parameters. The coefficients are optimized to minimize the residual sum of squares between the observed and the

calculated targets by the linear approximation [36]. This model was implemented using the *LinearRegression* class in scikit-learn.

Nearest Neighbors Regression: k -Nearest Neighbors (kNN) is a non-parametric method and one of the simplest machine learning algorithms. It operates by finding the k closest training examples (x_i) to each new input (x) and returns the average of their responses y_i .

$$Y(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i \tag{2}$$

where $N_k(x)$ is the neighborhood of x defined by the k closest points, where the Euclidean distance is usually used as a distance metric between samples [36]. Implementation was done using the *KNeighborsRegressor* in scikit-learn.

Decision Tree Regression: Decision Tree are models that use the training data to build a graph (tree) of simple decision rules that is used to analyse features. The prediction of the target variable is performed by following this tree, choosing the branches that return true values until an output (leaf) node is reached. This model was implemented using *DecisionTreeRegressor* in scikit-learn.

Ensemble methods: Occasionally, the algorithms presented above, due to their simplicity, cannot produce an accurate model for the problem at hand. In these cases other methodologies are required such as ensemble learning or neural networks. In ensemble methods a large number of simple models are trained and their predictions are combined to obtain a high accuracy model, providing improved generalizability and robustness over a single model [36]. There are two main types of ensemble methods. Averaging ensemble methods, like the Random Forest algorithm, average the predictions of several independently trained weak models. Boosting ensemble methods, such as the Gradient Boosted model, iteratively build multiple models in which each new learner mitigates the bias of the previous model. Random Forests and Gradient Boosted models are based on Decision Trees and have proven to be effective for regression on numerous cases [37]. Both ensamble methods were applied in scikit-learn using the *RandomForestRegressor* and *GradientBoostingRegressor* classes.

2.4 Classic D_{12} models

In this article the obtained ML models were compared with three widely known equations from the

literature, the predictive equations of Wilke-Chang [7] and Lai-Tan [8], and the correlation of Dymond-Hildebrand-Batschinski (DHB) [11–13]. The models are briefly described in the following.

Wilke-Chang equation: It is an empirical modification of the Stokes-Einstein relation and is given by:

$$D_{12} = 7.4\times10^{-8}\frac{(\phi M_1)^{0.5}T}{\mu_1 (V_{\mathrm{Tc,bp},2})^{0.6}} \tag{3}$$

where subscripts 1 and 2 represent solvent (CO_2) and solute, respectively, M_1 (g mol^{-1}) is the molar mass of solvent, μ_1 (cP) is the viscosity of the solvent, T (K) is the temperature, ϕ (nondimensional) is the association factor of the solvent (1.0 for the case of CO_2), and $V_{\mathrm{TC,bp},2}$ (cm^3 mol^{-1}) is the solute molar volume at normal boiling temperature. The last can be estimated by the Tyn-Calus relation [7, 38] when no experimental data is available:

$$V_{\mathrm{TC,bp},2} = 0.285\times V_{c,2}^{1.048} \tag{4}$$

being $V_{c,2}$ the critical volume of the solute.

Lai-Tan equation: It is a modification of the Wilke-Chang equation and was specifically devised for tracer diffusion coefficients in SC-CO_2. It is described as:

$$D_{12} = 2.50\times10^{-7}\frac{M_1^{0.5}T}{(10\times\mu_1)^{0.688}V_{c,2}^{0.284}} \tag{5}$$

Dymond-Hildebrand Batschinski model (DHB): It is a free-volume based model frequently adopted to describe transport properties in nonpolar systems. The equation is [11–13]:

$$D_{12} = B_{\mathrm{DHB}}\sqrt{T}(V_1 - V_{\mathrm{D}}) \tag{6}$$

where V_1 (cm^3 mol^{-1}) is the molar volume of the solvent, B_{DHB} (cm^3 mol^{-1} K$^{-1/2}$) is a parameter characteristic of the solute-solvent pair, and V_{D} (cm^3 mol^{-1}) is the minimum solvent molar volume required for diffusion. The last two are the adjustable parameters of the equation.

2.5 Model evaluation

The performance of the proposed models was evaluated using the average absolute relative deviation (AARD), which was always calculated in order to assess the goodness of fittings and predictions. It is given by:

$$\text{AARD}(\%) = \frac{100}{\text{NDP}} \sum_{i=1}^{\text{NDP}} \left| \frac{D_{12}^{\text{calc}} - D_{12}^{\text{exp}}}{D_{12}^{\text{exp}}} \right|_i \tag{7}$$

where superscripts calc and exp denote calculated and experimental values, and NDP is the number of data points. In addition to this weighted average, the simple arithmetic average of the AARD values of all systems (AARDarithmetic) was also calculated. The minimum and maximum system AARD are reported as an indication of the performance of the best and worst systems. Likewise, the AARD metric was also applied to the classic models used for comparison.

3 Results and discussion

Model development started with the selection of the relevant properties and variables from Table 1 (feature selection), followed by the selection of the machine learning algorithm and finally the comparison of the best machine learning models with the Wilke-Chang, Lai-Tan and DHB equations.

3.1 Properties and variables selection

A feature selection process was conducted to identify appropriate variables and properties for the model. Figure 1 shows the correlation matrix (in the form of heat map) for the SC-CO_2 data set, where values represent absolute Pearson correlation. Collinear quantities were excluded from the model by analyzing the Pearson correlations and setting a correlation coefficient threshold of 0.65. For each pair of quantities with a correlation above this value, usually the one with lower correlation with D_{12} was removed from the model. The simplicity of the model was also taken into account when selecting/excluding variables. For instance, when analyzing $T_{\text{bp},2}$ and M_2 pair, which presents a correlation of 0.88, $T_{\text{bp},2}$ has slightly higher correlation with D_{12}; however, $T_{\text{bp},2}$ was excluded in favor of M_2 as information on M_2 is immediate and rigorously calculated, thus endowing the model with greater simplicity.

Figure 1: Pearson correlation heat map for all properties and variables available for the supercritical CO_2 model.

Following this procedure, the selected properties/variables for the model were temperature, pressure, density, solute molar mass, solute critical pressure and solute acentric factor. Upon further testing, pressure was also excluded as its effect upon the model performance was negligible. This is consistent with the fact that pressure and temperature effects are both included in the density effect, thus at least one of these variables should be theoretically unnecessary. Table 3 shows the final properties/variables chosen for the supercritical CO_2 machine learning model (ML SC-CO_2), as well as those embodied in the Wilke-Chang and Lai-Tan equations used for comparison. The DHB correlation relies on V_1 and two fitted parameters (B_{DHB} and V_{D}), presented in Section 2.4.

The widespread use of the classic models such as the Wilke-Chang equation owes to their ease of applicability. Wilke-Chang and Lai-Tan equations require variables and parameters which are simple to obtain as temperature, viscosity, critical volume and molecular mass. The ML model here proposed relies on similarly simple quantities (temperature, density, molecular mass, and critical pressure) but also on the acentric factor whose information is less easily available. However, this property can be estimated using either the Pitzer [32] or Lee-Kesler [31] equations. The former requires knowledge of $T_{\mathrm{c},2}$, $P_{\mathrm{c},2}$ and vapor pressure while the later requires information on $T_{\mathrm{c},2}$, $P_{\mathrm{c},2}$ and $T_{\mathrm{bp},2}$.

Table 3: System properties/variables used in each model.

Parameters	Proposed model	Classic models		
	ML SC-CO_2	Wilke-Chang	Lai-Tan	DHB
T	✓	✓	✓	
P				
ρ_1	✓			
μ_1		✓	✓	
M_2	✓			
$T_{\mathrm{c},2}$				
$T_{\mathrm{bp},2}$				
$P_{\mathrm{c},2}$	✓			

$V_{c,2}$		✓	✓	
w_2	✓			
$\sigma_{LJ,2}$				
$\varepsilon_{LJ,2/k_B}$				
M_{CO_2}		✓	✓	
V_1				✓
B_{DHB}				✓[a]
V_D				✓[a]
Count	5	4	4	3

[a] Requires fitting to experimental data.

3.2 Machine Learning model selection

Five machine learning algorithms were applied in this study covering several types of supervised learning models: Multilinear regression (linear model), k-Nearest Neighbors, Decision Tree, Random Forest (averaging ensemble method) and Gradient Boosted (boosting ensemble method).

In Figure 2 the predicted diffusivities are plotted *versus* the experimental values for the test set for the five machine learning algorithms. The best results are achieved using the Gradient Boosted algorithm (Figure 2e) with AARD = 2.58 %. Note this is the pure prediction deviation calculated for the test set, not used in model training. Remaining algorithms ranked, from lower to higher AARD, as: Random Forest (4.14 %), k-Nearest Neighbors (4.77 %), Decision Tree (4.89 %), and Multilinear Regression (15.81 %). Similar behavior is observed for the k-Nearest Neighbors, Decision Tree, Random Forest and Gradient Boosted models (Figures 2b, 2c, 2d, and 2e) which present random distribution along diagonal, while the Multilinear Regression shows heavy underfitting for low and high D_{12} values.

Each ML algorithm hyper-parameters were optimized as described in Section 2.2. A grid search with 4-fold cross-validation was applied, using at least 3 levels for numeric hyper-parameters and all available options for non-numeric hyper-parameters. A detailed description of the tested and best hyper-parameter values for each machine learning algorithm is provided in Table 2. In the case of the best model (Gradient Boosted), it uses the least absolute

deviation as the loss function; 500 boosting stages; a minimum number of samples required to split of 10; maximum depth of 10; leafs with a minimum of 1 sample; and a maximum number of features considered when splitting equal to the square root of the total number of features; all remaining hyper-parameters were left at their default values. The best model will be henceforth denoted by ML-GB SC-CO_2 (Machine Learning Gradient Boosted model applied to D_{12} in SC-CO_2).

Figure 2: Predicted *versus* experimental diffusivities for the test set using different machine learning algorithms: a) Multilinear Regression; b) k-Nearest Neighbors; c) Decision Tree; d) Random Forest; and e) Gradient Boosted.

3.3 Comparison with classic models

The final proposed model (ML-GB SC-CO_2) contains five parameters: temperature, solvent density, solute molar mass, solute critical pressure and solute acentric factor. It provided an AARD of 2.58 % for the test set, which contains 108 systems and 1476 data points.

Table 4 compares this model performance with the classic models of Wilke-Chang, Lai-Tan, and DHB, in terms of global AARD, arithmetic average of systems ($AARD_{arithmetic}$), as well as minimum and maximum system AARD ($AARD_{min}$ and $AARD_{max}$). The DHB equation takes one less system since not enough points were available in the training set to fit its two parameters. Overall the new ML-GB SC-CO_2 model outperforms the classic models, with only the DHB equation attaining comparable results, which my be attributed to the two embodied parameters. With regard to the maximum system AARD, ML-GB SC-CO_2 shows a maximum deviation of 17.27 %, slightly below the one provided by DHB (19.54 %) and significantly below those of the remaining classic models. Additionally, one should keep in mind that the ML model is universal while the DHB equation is system-specific, *i.e.* requires two parameters previously fitted to available data of each system.

The performance of the classic models is presented in Figure 3 in terms of predicted *versus* experimental diffusivities. It is interesting to note that the Lai-Tan equation, a modification of the Wilke-Chang equation specifically developed for supercritical CO_2 systems, presents the worst results with an AARD of 26.01 %. This inferior performance is because the Lai-Tan equation was

obtained by re-optimizing the Wilke-Chang frontal coefficient and μ_1 and $V_{c,2}$ exponents using only 141 experimental points from 8 systems [8]. Consequently, such weaker support gives rise to higher errors when Lai-Tan equation deviates from its original pressure and temperature conditions and type of molecules.

Table 4: Comparison of the performance of the ML-GB SC-CO_2 model for the prediction (test set) of diffusivities in SC-CO_2 with the classic models: number of test systems, number of test points (NDP), global AARD, arithmetic average of systems AARD ($AARD_{arithmetic}$), minimum and maximum AARD ($AARD_{min}$ and $AARD_{max}$).

Model	Systems	NDP	AARD (%)	AARDarithmetic (%)	AARDmin (%)	AARDmax (%)
ML-GB SC-CO_2	168	1476	2.58	2.17	0.31	17.27
Wilke-Chang	168	1476	12.41	14.00	2.40	54.04
Lai-Tan	168	1476	26.01	22.97	1.56	84.25
DHB	167	1473	4.27	4.03	0.26	19.54

Figure 3: Predicted *versus* experimental diffusivities for the test set using the (a) Wilke-Chang, (b) Lai-Tan, and (c) DHB equations.

Detailed results for each system in the test and train sets are presented in Table 5, specifying the solute, number of data points and global AARD, for the ML-GB SC-CO_2 model and the three classic models adopted for comparison. The best results were obtained for the n-butylbenzene (0.31 %), n-decane (0.34 %) and 1-naphthol (0.44 %) systems, while the worst ones correspond to the m-xylene system (17.27 %) followed by the 1-methylnaphthalene (10.53 %) and 1-hexadecene (9.02 %) systems.

Figures 4 and 5 illustrate the $D_{12} / T^{0.5}$ dependence on solvent molar volume (free-volume theory) and Stokes-Einstein plots, respectively, using experimental and predicted (ML-GB SC-CO_2 model) diffusivities for two systems: acetone and 1-methylnaphthalene. The main idea

behind such representations is to demonstrate that the ML-GB SC-CO_2 model conserves the classical trends verified in the Stokes-Einstein and free-volume representations. As can be observed in both figures, the expected trends are kept, conserving the linear dependency of $D_{12}/T^{0.5}$ *versus* V_1 and D_{12} *versus* $T\mu_1^{-1}$. In fact this dependency can be quantified by the coefficient of determination (R^2) obtained scoring a value of 0.9509 (acetone) and 0.9681 (1-methylnaphthalene) for the Stokes-Einstein relation, and 0.9772 (acetone) and 0.9443 (1-methylnaphthalene) for the free-volume plot. In the case of 1-methylnaphthalene one sees that the model still follows the expected linear trends in the Stokes-Einstein and free-volume plots, notwithstanding the larger deviations of some data, mainly the last point. This causes the higher AARD found for this system.

Table 5: Calculated results (AARD) for the diffusivities of solutes in supercritical CO_2 for every system in the test and train sets achieved by the ML-GB SC-CO_2 model and the classic models used for comparison. Systems sources and ranges of temperature, pressure and densities are reported in Table SM2 in Supplementary Material.

				AARD (%)							
				ML-GB SC-CO_2		Wilke-Chang		Lai-Tan		DHB	
Solute	Total	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train
α-linolenic acid	56	13	43	1.39	0.69	13.19	14.56	30.94	32.70	2.62	2.83
α-pinene	30	13	17	3.17	1.54	6.51	5.91	13.77	13.40	3.10	2.91
α-tocopherol	82	28	54	1.47	0.77	25.86	26.75	30.85	31.63	1.96	2.23
β-carotene	90	25	65	1.35	0.79	15.79	14.53	66.97	65.89	2.36	2.30
β-pinene	15	2	13	4.87	1.35	5.03	12.75	4.65	5.14	10.31	3.62

γ -linolenic acid	142	37	105	1.23	0.81	7.63	7.84	36.35	36.36	2.40	2.08
γ -linolenic acid ethyl ester	41	13	28	2.70	1.43	8.57	6.16	25.50	23.30	5.45	4.98
γ -linolenic acid methyl ester	52	16	36	2.41	1.05	11.91	14.07	13.29	22.89	7.04	7.71
1,1,1,5,5,5-hexafluoroacetylacetone	15	5	10	3.67	1.35	20.18	18.34	35.06	30.84	4.87	4.24
1,1'-dimethylferrocene	68	25	43	1.71	0.77	11.89	12.32	18.72	18.69	3.91	3.73
1,2-dichlorobenzene	15	4	11	2.60	0.80	7.27	6.75	19.82	16.72	3.05	1.91
1,2-diethylbenzene	15	4	11	3.03	0.61	8.49	5.36	17.80	17.09	3.71	2.30
1,3,5-trimethylbenzene	34	8	26	3.05	1.38	6.31	6.26	14.14	15.95	6.04	3.49
1,3-dichlorobenzene	4	1	3	2.11	1.08	11.78	11.43	25.83	23.22	0.95	1.17
1,3-divinylbenzene	15	6	9	2.10	0.74	5.14	2.80	18.51	15.30	1.60	1.52
1,4-diethylbenzene	15	4	11	1.92	0.55	7.23	5.08	17.14	18.39	6.30	1.96
15-crown-5	29	13	16	2.26	0.93	6.91	8.62	13.51	16.37	6.44	5.95
1-hexadecene	11	6	5	9.02	0.75	16.52	15.75	16.91	26.19	10.37	14.81
1-methylnaphthalene	11	2	9	10.53	0.79	30.84	18.02	39.65	27.32	19.53	3.21
1-naphthol	11	3	8	0.44	0.45	5.43	5.90	3.57	4.99	1.29	0.97
1-phenyldodecane	15	5	10	1.16	0.90	6.15	7.37	51.24	45.14	2.31	3.98

1-phenylethanol	15	2	13	3.45	0.46	16.31	9.39	25.44	25.31	7.94	1.99
1-phenylhexane	15	6	9	1.21	0.64	8.23	7.05	23.16	24.19	2.79	2.88
1-phenyloctane	15	1	14	1.13	0.78	11.44	8.45	20.17	27.86	6.10	3.37
1-propanol	17	4	13	3.70	0.40	8.70	17.50	7.04	3.70	4.19	2.45
2,2,4,4-tetramethyl-3-pentanone	9	2	7	6.29	0.94	29.08	26.41	26.46	19.18	1.38	0.64
2,3-dimethylaniline	15	5	10	2.43	0.70	11.17	18.47	33.10	33.58	3.12	2.37
2,4-dimethyl-3-pentanone	8	2	6	6.37	1.13	12.32	10.90	15.63	20.98	2.31	2.51
2,4-dimethylphenol	15	6	9	1.83	0.53	5.66	11.92	22.43	25.96	4.85	2.91
2,6-dimethylaniline	15	3	12	0.81	0.65	12.43	11.23	28.21	28.02	1.42	3.87
2,6-dimethylnaphthalene	6	0	6	n.d.	3.36	n.d.	7.15	n.d.	18.68	n.d.	4.24
2,7-dimethylnaphthalene	6	1	5	6.64	4.02	8.09	6.67	8.66	17.51	7.99	3.74
2-bromoanisole	15	2	13	0.93	0.48	17.40	16.39	30.07	30.79	3.58	3.68
2-butanone	40	13	27	1.70	0.47	3.60	5.87	5.24	4.46	2.34	3.08
2-ethyltoluene	15	3	12	3.68	0.64	8.69	9.01	10.26	10.60	4.53	3.71
2-fluoroanisole	15	6	9	2.47	0.48	21.21	16.65	25.37	27.51	3.59	1.81
2-heptanone	11	1	10	6.24	0.52	34.47	29.67	32.84	22.10	1.05	1.93
2-methylanisole	15	5	10	5.03	3.22	9.09	9.97	22.5	23.0	4.30	2.56

								0	8		
2-naphthol	16	3	13	0.94	0.24	7.32	7.96	10.81	10.69	2.45	1.68
2-nitroanisole	15	3	12	2.54	0.74	12.55	11.20	33.27	29.90	4.15	2.25
2-nonanone	10	3	7	1.96	1.55	36.44	35.47	26.47	24.66	3.23	2.13
2-pentanone	23	4	19	4.09	0.48	7.92	3.71	3.76	1.96	5.31	1.71
2-phenyl-1-propanol	15	6	9	2.13	0.78	10.53	9.19	28.73	30.93	3.96	2.20
2-phenylethanol	15	4	11	1.02	0.22	4.16	11.34	24.09	27.77	3.89	2.68
2-phenylethyl acetate	15	10	5	3.61	0.42	8.19	9.52	36.18	36.75	4.34	1.80
2-propanol	18	7	11	1.31	0.52	10.50	8.98	6.15	8.59	3.71	1.65
3-ethyltoluene	15	9	6	2.08	0.53	10.60	13.31	8.85	9.46	7.21	3.12
3-fluorophenol	4	0	4	n.d.	0.61	n.d.	13.15	n.d.	24.26	n.d.	1.00
3-methylbutylbenzene	15	4	11	0.70	0.73	4.16	6.85	18.08	21.46	4.06	2.69
3-nitrotoluene	15	3	12	1.97	0.97	2.40	4.40	13.10	19.74	6.40	3.91
3-pentanone	46	12	34	1.59	0.57	7.76	8.47	4.95	3.97	2.78	2.59
3-phenyl-1-propanol	15	7	8	2.16	0.61	4.78	7.45	28.80	25.79	4.26	1.92
3-phenylpropyl acetate	15	5	10	1.75	0.72	3.92	8.72	39.50	37.33	5.21	3.37
4-ethyltoluene	15	2	13	2.71	0.41	10.97	6.92	10.40	13.08	4.90	2.78
4-heptanone	9	1	8	0.74	0.41	36.46	36.54	28.58	29.82	0.39	0.48

4-methylanisole	15	3	12	3.82	2.96	17.29	17.58	29.03	32.51	2.11	3.64
5-nonanone	12	1	11	0.48	0.76	31.24	34.59	11.63	21.64	0.73	1.17
5-tert-butyl-m-xylene	31	13	18	1.34	0.49	8.85	8.23	19.73	19.15	3.69	3.97
6-undecanone	13	5	8	6.02	1.27	36.61	36.47	20.92	18.74	3.40	2.00
AA ethyl ester	48	13	35	0.72	0.40	15.20	15.15	31.90	29.88	1.60	0.98
acetone	213	74	139	2.91	1.18	4.89	5.99	10.20	10.54	5.44	5.38
acridine	6	2	4	4.36	0.77	6.93	3.93	29.14	26.27	3.85	2.93
adamantanone	8	2	6	3.02	0.45	12.37	20.10	13.53	15.43	11.00	0.89
allylbenzene	15	8	7	3.32	0.44	6.62	3.93	17.80	14.64	5.53	1.76
aniline	15	6	9	6.59	0.42	37.92	30.28	34.25	33.24	3.83	2.21
anisole	15	3	12	1.48	0.59	6.25	7.60	16.47	16.27	1.90	3.28
anthracene	22	8	14	3.48	0.67	10.61	10.25	12.94	15.66	1.87	1.31
arachidonic acid	75	23	52	1.58	0.92	9.51	9.78	41.35	41.00	2.88	2.18
behenic acid ethyl ester	17	5	12	0.64	0.60	21.03	21.47	33.10	30.56	0.93	0.84
benzene	249	84	165	5.64	1.68	9.40	8.44	8.93	9.54	7.12	8.13
benzoic acid	35	7	28	4.78	0.50	11.02	9.97	15.05	15.70	5.37	6.83

benzyl acetate	15	6	9	1.29	0.81	7.49	7.99	29.40	27.21	2.70	3.35
benzylacetone	15	5	10	1.69	0.93	5.93	6.32	30.63	30.38	3.47	4.27
biphenyl	24	7	17	2.43	0.55	10.44	10.00	10.21	9.97	2.81	3.35
bromobenzene	21	11	10	3.48	1.09	5.39	7.34	12.63	11.21	4.36	4.63
butyric acid ethyl ester	16	5	11	1.65	0.99	3.80	4.64	5.24	7.19	1.99	1.85
caffeine	25	5	20	1.41	0.78	24.73	18.00	31.57	27.70	5.65	7.08
capric acid ethyl ester	16	4	12	3.57	0.56	13.02	13.61	17.09	16.64	0.83	1.65
caprylic acid ethyl ester	16	4	12	2.02	0.92	9.87	10.35	16.20	13.71	2.18	1.44
chlorobenzene	21	5	16	2.71	0.80	4.57	6.18	8.99	11.08	1.46	4.06
chromium(III) acetylacetonate	104	35	69	3.31	1.41	15.95	15.36	45.69	44.86	7.17	6.69
chrysene	4	3	1	4.79	0.87	15.63	17.76	17.65	19.73	n.d.	n.d.
citral	15	5	10	2.43	0.67	10.00	7.95	7.57	12.99	5.14	3.99
cobalt(III) acetylacetonate	38	13	25	1.68	0.90	12.06	11.25	47.02	47.32	1.57	2.49
copper(II) trifluoroacetylacetonate	12	5	7	6.92	0.60	31.29	41.24	46.08	57.78	10.53	1.90
cycloheptanone	8	3	5	3.91	0.76	25.74	22.98	14.56	21.39	7.41	1.45
cyclononanone	8	3	5	3.69	0.82	17.91	17.45	22.70	22.68	2.36	2.78

cyclopentanone	8	2	6	4.88	0.24	19.03	20.73	9.43	9.67	1.61	0.81
DHA ethyl ester	65	23	42	0.86	0.41	17.03	17.47	30.98	30.97	1.35	1.44
DHA methyl ester	17	2	15	0.90	0.57	17.80	17.31	25.14	32.55	0.85	0.94
dibenzo-24-crown-8	28	9	19	0.92	0.51	12.30	12.93	52.92	50.69	2.05	1.96
dibenzyl ether	15	5	10	0.88	0.53	3.70	6.13	35.80	39.34	5.35	2.51
diethyl ether	17	2	15	4.76	0.83	12.13	10.33	4.36	8.64	2.39	6.06
diisopropyl ether	15	3	12	7.14	0.58	9.94	6.44	7.66	12.47	4.73	8.47
diolein	9	5	4	2.56	0.74	23.56	23.85	49.02	48.03	1.74	1.82
Disperse blue 14	47	14	33	3.52	1.29	20.18	20.84	20.18	20.10	3.98	2.17
Disperse orange 11	65	18	47	3.08	1.80	20.01	20.46	14.09	14.98	3.57	3.70
D-limonene	15	4	11	2.38	0.75	10.72	8.80	7.75	7.03	4.80	3.77
docosahexaenoic acid (DHA)	63	22	41	1.18	0.60	9.10	7.42	49.04	47.06	2.12	1.53
eicosapentaenoic acid (EPA)	55	15	40	1.30	0.54	7.23	8.00	41.19	41.86	2.07	1.65
EPA ethyl ester	48	20	28	0.52	0.44	14.74	15.14	30.66	29.40	0.99	1.12
EPA methyl ester	17	6	11	1.40	0.43	17.37	17.36	30.00	30.74	0.51	0.46
ethanol	24	7	17	2.09	0.92	15.75	9.46	8.94	9.98	3.76	2.59
ethyl acetate	16	5	11	3.77	0.47	23.95	14.70	5.22	7.88	6.00	7.99
ethyl benzoate	15	7	8	4.04	0.73	4.43	3.39	21.22	28.05	3.97	1.95

ethylbenzene	15	4	11	0.68	0.69	7.65	7.36	4.89	4.81	2.67	2.31
eugenol	15	3	12	3.05	1.03	18.44	17.00	33.78	41.17	7.21	1.63
ferrocene	107	30	77	1.90	0.79	16.64	17.74	17.56	17.21	6.48	6.88
fluorobenzene	15	3	12	6.71	0.68	11.27	10.98	8.50	11.41	4.38	4.24
geraniol	4	0	4	n.d.	0.76	n.d.	3.34	n.d.	35.80	n.d.	0.38
hexachlorobenzene	14	4	10	4.16	0.60	6.75	12.69	14.09	14.22	4.17	4.49
Ibuprofen	99	27	72	1.85	0.73	9.35	10.05	18.91	17.90	4.14	4.13
iodobenzene	20	4	16	2.76	0.61	3.06	10.50	17.31	21.75	3.51	2.72
i-propylbenzene	36	6	30	1.01	0.91	7.19	9.06	7.82	7.49	3.05	2.23
isobutylbenzene	15	8	7	1.99	0.59	3.94	5.63	15.27	19.71	3.20	1.98
L-carvone	27	10	17	1.83	0.88	2.85	4.13	24.25	24.50	2.58	2.69
linalool	15	4	11	2.52	0.77	10.19	6.16	9.72	10.77	3.02	4.42
linoleic acid	71	27	44	1.09	0.83	9.54	9.68	37.33	38.41	3.54	3.98
linoleic acid methyl ester	20	3	17	2.03	0.77	13.41	15.80	37.30	37.44	1.91	1.53
L-menthone	23	5	18	1.74	1.13	5.67	5.04	21.60	19.28	2.65	2.85
methanol	10	7	3	2.55	0.20	14.11	23.06	18.31	17.85	3.96	0.18
monoolein	11	4	7	0.65	0.69	6.78	9.81	41.6	44.5	1.53	1.26

								8	7			
m-xylene	12	5	7	17.27	0.78	27.01	11.10	18.50	15.00	10.54	9.75	
myristic acid ethyl ester	16	8	8	1.86	0.95	15.35	16.60	25.14	22.16	2.86	1.58	
myristoleic acid	42	11	31	1.50	0.68	5.20	5.86	30.85	31.63	1.55	3.10	
myristoleic acid methyl ester	81	19	62	4.45	1.18	8.84	10.92	13.57	15.24	9.66	10.47	
N-(4-methoxybenzylidene)-4-n-butylaniline	5	0	5	n.d.	0.33	n.d.	17.85	n.d.	3.72	n.d.	0.33	
naphthalene	114	35	79	3.16	0.92	11.24	10.66	8.20	9.75	7.24	10.36	
n-butylbenzene	15	4	11	0.31	0.59	8.28	5.57	19.26	15.59	6.62	2.32	
n-decane	5	1	4	0.34	0.16	36.55	38.99	17.82	21.93	2.93	1.08	
n-dodecane	5	0	5	n.d.	0.31	n.d.	40.87	n.d.	19.59	n.d.	2.99	
n-heptane	5	2	3	1.50	0.11	27.24	30.02	15.10	21.12	2.25	0.02	
n-hexane	5	3	2	0.85	0.28	21.48	25.05	13.61	20.68	7.30	0.00	
nitrobenzene	23	4	19	2.18	0.74	7.10	8.42	19.26	18.96	3.27	3.01	
n-nonane	5	0	5	n.d.	0.20	n.d.	36.45	n.d.	21.11	n.d.	1.43	
n-octane	5	1	4	1.59	0.36	32.10	33.74	16.25	21.38	6.82	0.41	
n-pentane	5	1	4	0.84	0.32	10.82	13.83	9.27	12.81	2.67	1.56	

n-pentylbenzene	31	10	21	3.48	0.51	8.25	8.43	16.42	17.53	4.51	2.58
n-propylbenzene	60	13	47	3.44	0.87	7.41	13.12	9.47	7.64	4.77	4.56
n-tetradecane	5	1	4	6.13	0.28	31.93	40.49	1.56	14.91	11.57	0.78
n-undecane	5	3	2	6.66	0.50	40.84	41.08	22.16	21.29	2.94	0.00
oleic acid	19	4	15	1.47	0.30	11.96	9.51	40.57	39.87	2.44	2.14
oleic acid ethyl ester	5	2	3	3.16	0.40	9.75	3.04	28.85	29.33	2.74	0.23
oleic acid methyl ester	21	5	16	2.85	0.42	8.94	6.78	29.82	28.23	5.15	3.31
palladium(II) acetylacetonate	125	41	84	1.26	0.62	20.59	22.59	38.41	37.88	5.22	4.62
palmitic acid ethyl ester	17	5	12	1.67	0.53	14.84	15.27	31.30	27.90	0.26	0.75
p-dichlorobenzene	13	4	9	3.73	0.48	10.94	10.46	13.61	18.53	5.69	2.87
phenanthrene	25	7	18	6.74	1.05	14.93	15.69	4.20	5.83	5.04	4.57
phenol	109	27	82	2.31	0.48	21.40	21.14	9.58	10.62	5.55	4.47
phenylacetic acid	16	2	14	1.74	0.62	5.82	4.02	17.69	15.58	1.83	1.86
phenylacetylene	15	5	10	1.49	0.72	6.18	8.61	16.45	15.84	1.69	1.58
phenylbutazone	78	20	58	1.78	0.69	9.65	7.67	34.13	32.18	5.29	5.98
phenylmethanol	15	4	11	1.51	0.53	14.14	13.83	22.23	22.78	3.28	2.26
p-xylene	7	1	6	0.91	2.19	7.05	6.46	1.95	4.95	1.14	4.02

pyrene	21	9	12	2.99	1.08	8.64	10.58	24.57	16.72	3.74	3.32
sec-butylbenzene	15	4	11	1.63	0.65	3.86	4.03	22.73	18.95	4.68	2.83
squalene	5	1	4	2.35	1.30	33.70	29.88	18.67	26.75	3.93	1.15
stearic acid	4	2	2	2.09	1.00	50.13	50.51	20.71	21.43	0.77	0.00
stearic acid ethyl ester	17	5	12	0.62	0.59	24.10	24.53	29.64	24.06	1.27	0.99
styrene	15	6	9	4.28	0.68	6.82	4.43	12.89	15.94	5.13	3.94
tert-butylbenzene	15	5	10	1.38	0.75	8.26	7.74	15.66	13.21	3.33	3.95
tetrahydrofuran	15	2	13	4.36	0.30	8.08	17.18	2.38	12.21	12.29	4.50
thenoyltrifluoroacetone	15	2	13	1.88	1.13	35.57	29.37	44.66	48.09	2.75	3.43
toluene	41	10	31	3.26	0.68	10.33	8.44	4.57	4.65	9.01	8.26
triarachidonin	27	8	19	1.56	0.82	18.29	17.15	70.02	70.36	0.67	0.88
trierucin	101	30	71	2.22	0.67	15.40	12.80	80.45	83.10	2.68	3.06
trifluoroacetylacetone	15	6	9	3.62	0.36	2.88	4.30	12.28	10.43	3.14	1.46
trinervonin	38	13	25	1.54	0.60	16.15	16.86	84.25	82.66	2.84	2.97
triolein	14	6	8	2.11	0.44	30.59	28.66	54.80	53.40	3.95	2.57
ubiquinone CoQ10	80	27	53	2.22	0.84	12.49	14.61	71.68	71.48	4.00	4.21

vanillin	15	5	10	2.36	1.40	11.45	13.23	23.79	25.05	2.18	2.08
vitamin K1	17	8	9	2.37	1.93	25.75	28.57	31.30	31.12	2.45	2.56
vitamin K3	22	4	18	3.05	1.27	8.83	9.75	12.64	11.89	4.75	2.56
water	24	3	21	5.73	0.25	54.04	56.51	21.57	12.07	3.21	4.32

n.d.: not determined.

Figure 4: Experimental and calculated diffusivities (ML GB SC-CO_2 model) in terms of free-volume theory coordinates for (a) acetone and (b) 1-methylnaphthalene.

Figure 5: Experimental and calculated diffusivities (ML-GB SC-CO_2 model) in terms of Stokes-Einstein coordinates for (a) acetone and (b) 1-methylnaphthalene.

4 Conclusions

In this work a machine learning model for the prediction of binary diffusivities in SC-CO_2 was developed. This model was trained and validated by splitting a database containg 13 properties of 174 systems and 4917 points into training and test sets. Several learning algorithms were tested (Multilinear Regression, k- Nearest Neighbors, Decision Tree, Random Forest and Gradient Boosted). The best results were found using the Gradient Boosted algorithm, which presented an average deviation of 2.58 % for the test set. This model takes five input properties/variables which are readily available for multiple solutes: temperature, solvent density, solute molar mass, solute critical pressure and solute acentric factor. Results were compared with the classic diffusivity equations of Wilke-Chang, Lai-Tan, and Dymond-Hildebrand-Batschinski, which demonstrated worse performance for the same data with deviations of 12.41 %, 26.01 % and 4.27 %, respectively. Although the Dymond-Hildebrand-Batschinski model shows similar performance, it requires *a priori* experimental data to fit the system parameters, which is not always possible.

Acknowledgements

This work was developed within the scope of the project CICECO-Aveiro Institute of Materials, UIDB/50011/2020 & UIDP/50011/2020, financed by national funds through the Foundation for Science and Technology/MCTES, as well as the Multibiorefinery project (POCI-01-0145-FEDER-016403). Bruno Zêzere thanks FCT for PhD grant SFRH/BD/137751/2018.

References

[1] P C Wankat. *Rate-controlled separations*. Blackie Academic & Professional, Great Yarmouth, 1994.

[2] Eduardo L G Oliveira, Armando J D Silvestre, and Carlos M. Silva. Review of kinetic models for supercritical fluid extraction. *Chemical Engineering Research & Design*, 89(7A):1104–1117, 2011.

[3] James J. Carberry. Chemical and catalytic reaction engineering. *McGraw-Hill*, 1971.

[4] Denis Prat, John Hayler, and Andy Wells. A survey of solvent selection guides. *Green Chemistry*, 16(10):4546–4551, oct 2014.

[5] M M R de Melo, A J D Silvestre, and C M Silva. Supercritical fluid extraction of vegetable matrices: applications, trends and future perspectives of a convincing green technology. *The Journal of Supercritical Fluids*, 2014.

[6] Bruno Zêzere, Ana L. Magalhães, Inês Portugal, and Carlos Manuel Silva. Diffusion coefficients of eucalyptol at infinite dilution in compressed liquid ethanol and in supercritical CO_2 ethanol mixtures. *Journal of Supercritical Fluids*, 133:297–308, mar 2018.

[7] C. R. Wilke and Pin Chang. Correlation of diffusion coefficients in dilute solutions. *AIChE Journal*, 1(2):264–270, jun 1955.

[8] Ching Chih Lai and Chung Sung Tan. Measurement of molecular diffusion coefficients in supercritical carbon dioxide using a coated capillary column. *Industrial and Engineering Chemistry Research*, 1995.

[9] Raquel V. Vaz, Ana L. Magalhães, and Carlos M. Silva. Improved hydrodynamic equations for the accurate prediction of diffusivities in supercritical carbon dioxide. *Fluid Phase Equilibria*, 2013.

[10] Raquel V. Vaz, Ana L. Magalhães, and Carlos M. Silva. Improved Stokes-Einstein based

models for diffusivities in supercritical CO_2. *Journal of the Taiwan Institute of Chemical Engineers*, 45(4):1280–1284, 2014.

[11] J. H. Dymond. Corrected Enskog theory and the transport coefficients of liquids. *The Journal of Chemical Physics*, 1974.

[12] J. H. Dymond, E. Bich, E. Vogel, W. A. Wakeham, V. Vesovic, and M. J. Assael. Dense Fluids. In Jürgen Millat, J. H. Dymond, and C. A. Nieto de Castro, editors, *Transport Properties of Fluids*, pages 66–112. Cambridge University Press, 1996.

[13] C.M. Silva and H. Liu. Modelling of Transport Properties of Hard Sphere Fluids and Related Systems, and its Applications. In *Theory and Simulation of Hard-Sphere Fluids and Related Systems*, pages 383–492. Springer, Berlin, 2008.

[14] F. Gharagheizi and M. Sattari. Estimation of molecular diffusivity of pure chemicals in water: A quantitative structure-property relationship study. *SAR and QSAR in Environmental Research*, 2009.

[15] Aboozar Khajeh and Mohammad Reza Rabaani. Diffusion coefficient prediction of acids in water at infinite dilution by QSPR method. *Structural Chemistry*, 2012.

[16] Raquel V. Vaz, Ana L. Magalhães, Daniel L.A. Fernandes, and Carlos M. Silva. Universal correlation of self-diffusion coefficients of model and real fluids based on residual entropy scaling law. *Chemical Engineering Science*, 2012.

[17] Reza Beigzadeh, Masoud Rahimi, and Seyed Reza Shabanian. Developing a feed forward neural network multilayer model for prediction of binary diffusion coefficient in liquids. *Fluid Phase Equilibria*, 2012.

[18] R. Eslamloueyan and M. H. Khademi. A neural network-based method for estimation of binary gas diffusivity. *Chemometrics and Intelligent Laboratory Systems*, 2010.

[19] Seyyed Alireza Mirkhani, Farhad Gharagheizi, and Mehdi Sattari. A QSPR model for prediction of diffusion coefficient of non-electrolyte organic compounds in air at ambient condition. *Chemosphere*, 2012.

[20] Alireza Abbasi and Reza Eslamloueyan. Determination of binary diffusion coefficients of hydrocarbon mixtures using MLP and ANFIS networks based on QSPR method. *Chemometrics and Intelligent Laboratory Systems*, 2014.

[21] Kenneth S. Pitzer and Donald R. Schreiber. Improving equation-of-state accuracy in the critical region; equations for carbon dioxide and neopentane as examples. *Fluid Phase*

Equilibria, 1988.

[22] V. V. Altunin and M. A. Sakhabetdinov. Viscosity of liquid and gaseous carbon dioxide at temperatures of 220 - 1300 K and pressures up to 1200 bar. *Thermal Engineering (English translation of Teploenergetika)*, 1972.

[23] Robert C. Reid, John M. Prausnitz, and Bruce E. Poling. *The Properties of Gases & Liquids*. McGraw-Hill, 4th edition, 1987.

[24] K.G. Joback. *A Unified Approach to Physical Property Estimation Using Multivariate Statistical Techniques*. Master thesis, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[25] K. G. Joback and R. C. Reid. Estimation of pure-component properties from group-contributions. *Chemical Engineering Communications*, 57(1-6):233–243, jul 1987.

[26] G. Raam Somayajulu. Estimation Procedures for Critical Constants. *Journal of Chemical and Engineering Data*, 1989.

[27] Ana L. Magalhães, Francisco A. Da Silva, and Carlos M. Silva. Free-volume model for the diffusion coefficients of solutes at infinite dilution in supercritical CO_2 and liquid H_2O. *Journal of Supercritical Fluids*, 74:89–104, 2013.

[28] K. M. Klincewicz and R. C. Reid. Estimation of critical properties with group contribution methods. *AIChE Journal*, 1984.

[29] Robert H. Perry and Don W. Green. *Perry's Chemical Engineers' Handbook*, volume Sixth edit. McGraw-Hill, 6th edition, 1997.

[30] Leonidas Constantinou and Rafiqul Gani. New group contribution method for estimating properties of pure compounds. *AIChE Journal*, 40(10):1697–1710, oct 1994.

[31] Byung Ik Lee and Michael G. Kesler. A generalized thermodynamic correlation based on three-parameter corresponding states. *AIChE Journal*, 1975.

[32] Kenneth S. Pitzer, David Z. Lippmann, R. F. Curl, Charles M. Huggins, and Donald E. Petersen. The Volumetric and Thermodynamic Properties of Fluids. II. Compressibility Factor, Vapor Pressure and Entropy of Vaporization. *Journal of the American Chemical Society*, 1955.

[33] Hongqin Liu, Carlos M. Silva, and Eugénia A. Macedo. New Equations for Tracer Diffusion Coefficients of Solutes in Supercritical and Liquid Solvents Based on the Lennard-Jones Fluid Model. *Industrial & Engineering Chemistry Research*,

36(1):246–252, 1997.

[34] Andriy Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.

[35] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, oct 2011.

[36] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[37] Andreas C. Müller and Sarah Guido. *Introduction to Machine Learning with Python: a guide for data scientists*. O'Reilly Media, Inc., 2016.

[38] B E Poling, J M Prausnitz, and J P O'Connel. *The Properties of Gases and Liquids*. McGraw-Hill, Singapore, 5th edition, 2000.

Declaration of interests

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

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

University of Aveiro, November 30th 2020

Dear Professor Stratos Pistikopoulos,

Editor-in-Chief of the Journal of Molecular Liquids

On behalf of all co-authors I hereby confirm that we have no conflict of interests with any researcher or entity, and that our work is being uniquely submitted to the *Journal of Molecular Liquids.*

Carlos Manuel Silva
Chem. Eng. Group | CICECO - Aveiro Institute of Materials
Associate Professor | Department of Chemistry | University of Aveiro
Campus Universitário de Santiago
3810-193 Aveiro | Portugal

E-mail: carlos.manuel@ua.pt | Web: www,egichem.com
Phone: +351 234 401 549 | Ext: 24928

Highlights

- New predictive model to estimate diffusivities in supercritical carbon dioxide.
- The new machine learning model was trained with a database of 174 binary systems.
- It was compared with several classical models, such as the Wilke-Chang equation.
- The machine learning model provided the best performance with errors of 2.58 %.