Journal of Molecular Liquids 372 (2023) 121195

Contents lists available at ScienceDirect

Journal of Molecular Liquids

journal homepage: www.elsevier.com/locate/molliq

Novel mathematical and polypharmacology predictions of salicylsalicylic acid: Solubility enhancement through $SCCO_2$ system

Peijun Zhang [a,*], Mustafa Fahem Albaghdadi [b], Sabah Auda AbdulAmeer [c], Abdulmalik S. Altamimi [d], Ali Zeinulabdeen Abdulrazzaq [e], Hayder chailibi [f], Salema K. Hadrawi [g], Hassan Falih Hamdan [h], Farag M.A. Altalbawy [i,j], Amal M. Alsubaiyel [k,*]

[a] School of Computer Science, Xijing University, Xi'an, Shaanxi 710123, China
[b] Information Technology Unit, Al-Mustaqbal University College, Babylon 51001, Iraq
[c] Ahl Al Bayt University, Kerbala, Iraq
[d] Department of Pharmaceutical Chemistry, College of Pharmacy, Prince Sattam Bin Abdulaziz University, PO Box 173, Alkharj 11942, Saudi Arabia
[e] Department of Dental Industry Techniques, Al-Noor University College, Bartella, Iraq
[f] College of Medical Techology, Al-Farahidi University, Iraq
[g] Refrigeration and Air-conditioning Technical Engineering Department, College of Technical Engineering, The Islamic University, Najaf, Iraq
[h] Mazaya University College, Iraq
[i] Department of Chemistry, University College of Duba, University of Tabuk, Duba 71911, Saudi Arabia
[j] National Institute of Laser Enhanced Sciences (NILES), University of Cairo, Giza 12613, Egypt
[k] Department of Pharmaceutics, College of Pharmacy, Qassim University, Buraidah 52571, Saudi Arabia

ARTICLE INFO

Article history:
Received 8 November 2022
Revised 27 December 2022
Accepted 30 December 2022
Available online 3 January 2023

Keywords:
Solubility improvement
Machine learning
Simulation
Model prediction

ABSTRACT

Over the last decades, significant drawbacks of organic solvents such as high toxicity have motivated the scientists to find more eco-friendly solvents. Supercritical fluids (SCFs), especially $SCCO_2$, are known as a promising class of solvent, which have shown their indisputable potential of application in industrial-based pharmaceutical activities due to possessing various advantages such as high abundancy, low cost, and insignificant toxicity. Machine Learning (ML) is considered as a numerical approach to estimate drug solubility in pharmaceutical industry. The purpose of this manuscript is to estimate the solubility of salicylsalicylic acid in $SCCO_2$ and compare it with experimental data using machine learning (ML) approach. A regression problem with 32 input vectors is the subject of this study, which is being conducted. This dataset contains two input features (P and T) and one output feature. We utilized Decision Tree (DT), K-nearest neighbor (KNN), and Multilayer perceptron (MLP) regression models as the first time for salicylsalicylic acid, which had error rates of 1.10E-01, 1.07E-01, and 7.13E-01, respectively, when using the MAPE measure. In addition, the R-squared scores for the DT, KNN, and MLP models are 0.974, 0.996, and 0.809, respectively. The third statistic is MAE, in which the error rates of models are 5.27E-05 for DT, 5.53E-05 for KNN, and 2.61E-04 for MLP. The error rates of DT, KNN, and MLP are all 5.27E-05. Finally, KNN was the most general model, with optimal values of P = 400, T = 338.0, and Y = 0.00388 being obtained.

1. Introduction

Development of innovative experimental and analytical techniques for facilitating the production of novel and effective drugs is an important need in recent pharmaceutical industry [1,2]. Due to the existence of disparate technological/operational challenges towards appropriate incorporation of safe manufacturing processes with mainstream therapeutic application, scientific areas of drug need promising approaches and thus, novel pharmacological development processes [3].

In current decades, true selection of solvent plays an important role in the pharmaceutical industry. Despite noteworthy advantages of organic solvents in chemical/pharmaceutical industry such as good performance and ease of use, the emergence of various drawbacks including high vapor pressures and flammability, great toxicity, and atmospheric pollution has confined their application [4,5]. To overcome the limitations of organic solvents, the use of supercritical fluids (SCFs), which possess the properties of both gaseous and liquid states, have been of great interest.

* Corresponding authors.
E-mail addresses: zhangpj2006@126.com (P. Zhang), asbiel@qu.edu.sa (A.M. Alsubaiyel).

https://doi.org/10.1016/j.molliq.2022.121195
0167-7322/

SCFs have shown the performance of their usage in disparate scientific fields like extraction, targeted drug delivery, chromatography, purification, and separation [6–8]. Among different types of SCFs, supercritical carbon dioxide ($SCCO_2$) has been widely applied in industrial-based pharmaceutical activities owing to having disparate advantages such as high abundancy, low cost, good biodegradability, and insignificant toxicity [9].

Application of artificial intelligence (AI) technique to numerically anticipate the drug solubility results obtained by experimental investigations has been interesting among researchers. AI approach is currently used to solve minor but important challenges in drug development industries due to its brilliant capability to generate meaningful insights [1]. Machine learning (ML) tools are progressively replacing analytical modeling in scientific domains. These techniques are used to solve a variety of issues, including decision trees, artificial neural networks, and other linear and non-linear models [10–12].

We have a dataset with two input features and one output in this study. In this work, three models are chosen: Multiple Layers Perceptron, Decision Tree, and K-nearest Neighbors.

The name "MLP" refers to a neural network with many layers of perceptron. MLPs are forward-feeding artificial neural networks. MLP includes minimum of three layers from inputs, outputs, and hidden layers. The nodes are not activated in in the first layer; rather, the nodes in this layers represent the data point. If the data point is reflected by a vector of length *d*, then the input layer will include *d* nodes [13,14].

Estimation problems benefit from the simplicity of the k-Nearest Neighbor (KNN). Furthermore, because it is a wasteful algorithm that does not generalize from the training subset, all the training subset is retained during the testing stage. It is through comparison with the training data that the KNN regressor learns [15–17].

Classification and regression problems can be solved with the help of a decision tree, which has roots in machine learning. A benefit of decision tree compared to another classification systems is that it employs an ordered or hierarchical decision-making structure instead of simply grouping characteristics (or bands) together. The decision tree is a hierarchical and interpretable model for dealing with various machine learning problems. We begin at the decision tree's root and work our way down the tree based on the value of each feature in each node's subtree. We repeat this process until we reach the last node or leaf [18–20].

2. Data set

In the dataset used for this work (Table 1) [21], pressure and temperature are inputs and a single output is shown through 32 values.

Table 1
The whole data set. Re-used from [21] with permission from Elsevier.

No	x1 = P	x2 = T	y
1	120	308	1.07×10^{-4}
2		318	7.07×10^{-5}
3		328	6.12×10^{-5}
4		338	3.77×10^{-5}
5	160	308	1.98×10^{-4}
6		318	2.17×10^{-4}
7		328	1.87×10^{-4}
8		338	1.66×10^{-4}
9	200	308	2.47×10^{-4}
10		318	3.59×10^{-4}
11		328	5.14×10^{-4}
12		338	5.93×10^{-4}
13	240	308	2.95×10^{-4}
14		318	4.83×10^{-4}
15		328	6.85×10^{-4}
16		338	1.02×10^{-3}
17	280	308	3.88×10^{-4}
18		318	6.74×10^{-4}
19		328	1.05×10^{-3}
20		338	1.60×10^{-3}
21	320	308	4.71×10^{-4}
22		318	8.59×10^{-4}
23		328	1.39×10^{-3}
24		338	2.11×10^{-3}
25	360	308	5.35×10^{-4}
26		318	9.58×10^{-4}
27		328	1.69×10^{-3}
28		338	2.52×10^{-3}
29	400	308	5.77×10^{-4}
30		318	1.28×10^{-3}
31		328	2.25×10^{-3}
32		338	3.88×10^{-3}

3. Methodology

3.1. Decision tree regression

The growing popularity of DT can be attributed to several factors, including their ease of use and interpretation, their low time complexity, and their ability to be represented graphically. A DT is a collection of constraints that are applied in a logical order, starting at the root and working their way up the tree to the leaf or branch at the end [22,23]. Hierarchical tree structures are more transparent than neural networks (ANNs), making them more straightforward to understand and comprehend. The assessment aspects to maximize inter-node heterogeneity when constructing the DT from a dataset [11].

Classification and regression trees are two distinct approaches to data mining (RT). For the intended purpose, the theoretical foundation of RT is presented here in a brief overview. Multiple regressions and recursive partitioning of the dataset are used to generate the DT. Each rule of the tree's internal node repeats the data splitting process until a previously specified stop condition is met, beginning from the root node. It is attached to each of the leaves or terminal nodes a simple regression model that is only applicable to that node. Using pruning to reduce the tree's structural complexity will help it generalize better once the induction process is complete. Pruning criteria can be based on the number of cases in a node.

Before the DT can be activated, it's necessary to select the best possible measurement vectors. The process begins with a binary split of the dependent feature or the root node, with the sub nodes being "purer" than the root node. The DTs go through all the possible splits to find the one that maximizes the resulting tree's 'purity,' and this is how they do it (s*) [18,20,24].

$$\Delta i(s,t) = i(t) - p_L i(t_L) - p_R i(t_R)$$

It's s that divides the node into left and right child nodes, and each of these has a proportion of p_L and p_R in it, as shown in the equation. Prior to splitting, $i(t)$ which is impurity measurement; after splitting, $i(t_L)$ and $i(t_R)$ are criterion of impurity; then finally, $\Delta i(s,t)$ calculates impurity reduction caused by split *s*.

Impurity can be measured in a variety of ways. Gain-ratio, Gini index, and Chi-square are a few common examples. In this study, the Gini index (the most common) is used to measure $i(t)$.

$$I_G\left(t_{X(x_i)}\right) = 1 - \sum_{j=1}^{m} f\left(t_{X(x_i)}, j\right)^2$$

To calculate the percentage of examples with the value x_i belonging to node t, we divide $f(t_{X(x_i)},j)$ by the total number of nodes. Based on Gini impurity index, the DT splitting criterion is to select the attribute with the lowest (IG).

3.2. K Nearest neighbor regression

For a simple model for classification or regression, try k-Nearest Neighbor regression. To add insult to injury, it's a poor algorithm that can't handle data that it hasn't seen before because it doesn't generalize from the training set. As new examples are identified, the KNN regressor compares them to the original training data [16]. Consider $T = \{(x_1, y_1), \cdots, (x_N, y_N)\}$ represent the training data with distance parameter d, $x_i = (x_{i1}, x_{i2}, \cdots, x_{im})$ represent the i^{th} instance indicated by m input attributes and the corresponding output y_i. Also, N stands for the size of dataset. It is required to compute the d_i distance between an unseen data point x and each sample x_i in T and sort the d_i distances. If d_i is ranked i^{th}, then $NN_i(x)$ is the matching instance for di, and $y_i(x)$ is the target it is aiming for. Finally, the prediction $\hat{y}$ of x is the mean of the results of k nearest neighbors to $\times$, i.e. $\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_i(x)$ [17]. To briefly describe the processes of the KNN regression algorithm, we can list these steps [17]:

- Inputs: training samples $\{x_i, y_i\}, x_i$: input parameters, y_i: the output, x: the test data point targeted for estimation
- Algorithm:
  - o determine distance $D(x, x_i)$ to each training data point x_i
  - o choose k nearest data points $x_{i1} \cdots x_{ik}$ and their outputs $y_{i1} \cdots y_{ik}$
  - o output:

$$\hat{y} = f(x) = \frac{1}{K}\sum_{j=1}^{k} y_{i_j}$$

3.3. Multilayer perceptron

In 1943, the notion of artificial neural networks was developed [25]. Later, in 1958, the perceptron was introduced as the first practical artificial neural network [26]. Since 1986, neural networks have grown in popularity [27].

Because they are modeled after the nervous system, neural networks use neurons as their primary building block. Layers of artificial neurons (nodes) form neural networks by wires. Networks like these can identify previously unseen patterns by learning from both their input patterns and from the mistakes they make along the way. Based on the connections, neuron model, and weight adjustment methods, there are a variety of neural networks that can be built [28]. Artificial neural network (ANN) procedures like the Multilayer Perceptron (MLP) can be used to model any smooth calculable functional relationship between parameters and outputs [29].

Updates and optimizations based on job complexity allow for a flexible approach to the size of hidden layers. The artificial neurons in the MLP system are organized in a three-layered network. The input data is utilized to build an output parameter, which is then mixed with computations from the hidden layer, as it passes through a series of input and hidden layers [14,30].

Input weights for neurons are calculated using the following equation:

$$z = x_1 w_1 + \cdots + x_n w_n = X^T W$$

There are a variety of continuously differentiable functions that can be used to compute the activation function as $f(z)$, including the more recent ReLU that is frequently used in deep learning. Using the findings from the perceptron, a classifier can then "guess" the right label for the input data by activating the appropriate activation functions. After gathering the results using back propagation, the weights are modified for other unseen input vectors [31,32].

3.4. Polypharmacology and repurposing of Salsalate

Salsalate smile code (C1=CC=C(C(=C1)C(=O)OC2=CC=CC=C2C(=O)O)O) was generated from pubchem (https://pubchem.ncbi.nlm.nih.gov/)then inserted into Super-PRED (https://prediction.charite.de/) the obtained predicted targets are supported in supplementary data files. Target predictions performed by super-pred are based on fingerprints similarity with different protein ligands. probability" and "model accuracy are used as two different scores used in target prediction process, additionally 10-fold cross-validation process is applied to assure getting accurate results.

3.5. Molecular docking studies

Computer aided drug design is extensively applied in drug discovery and molecular docking is a beneficial method to provide us with binding affinity, protein-ligand interactions, and binding mode information. In the current research work we used CB-Dock (https://clab.labshare.cn/cb-dock/php/dockingresult.php) to perform molecular docking of Salsalate into the putative protein targets. The pdb files obtained from super-pred results are used in docking simulations.

4. Results

For the mentioned models, we need to tune their critical hyperparameters, which in this research, with the help of grid search [33], their values have been found with more than 2000 executions. Some standard regression metrics are then used to evaluate the final models.

The coefficient of determination, commonly known as the R^2 score, can be described as the proportion of change in one variable that be able to illustrate through another variable. The coefficient of determination is computed using the following equation.

$$r^2 = 1 - \frac{\sum (y - \hat{y})^2}{\sum \left(y - \bar{y}\right)^2}$$

When the value of y is determined from the observed value, the value of $\hat{y}$ is computed from the forecasted value, $\bar{y}$, and the mean of y is calculated.

MAE is determined using this equation, which averages the difference between forecasted and actual values. Also, MAPE metric is defined below:

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N}$$

$$MAPE = \left(\frac{\sum_{i=1}^{N} \left|\frac{y_i - \hat{y}_i}{y_i}\right|}{N}\right) \times 100$$

Figs. 1, 2, and 3 compare the real amount and the estimated amount via three predictive models including DT, KNN and MLP. In the below graphs, blue points show the predicted amount in the training phase, the red points illustrate the test phase, and the green line shows the real amount. Comparing the obtained results of DT, KNN and MLP predictive models in Table 2 introduces the KNN model as the most general and precise estimator because most neighborhood points are close to the green line and the test

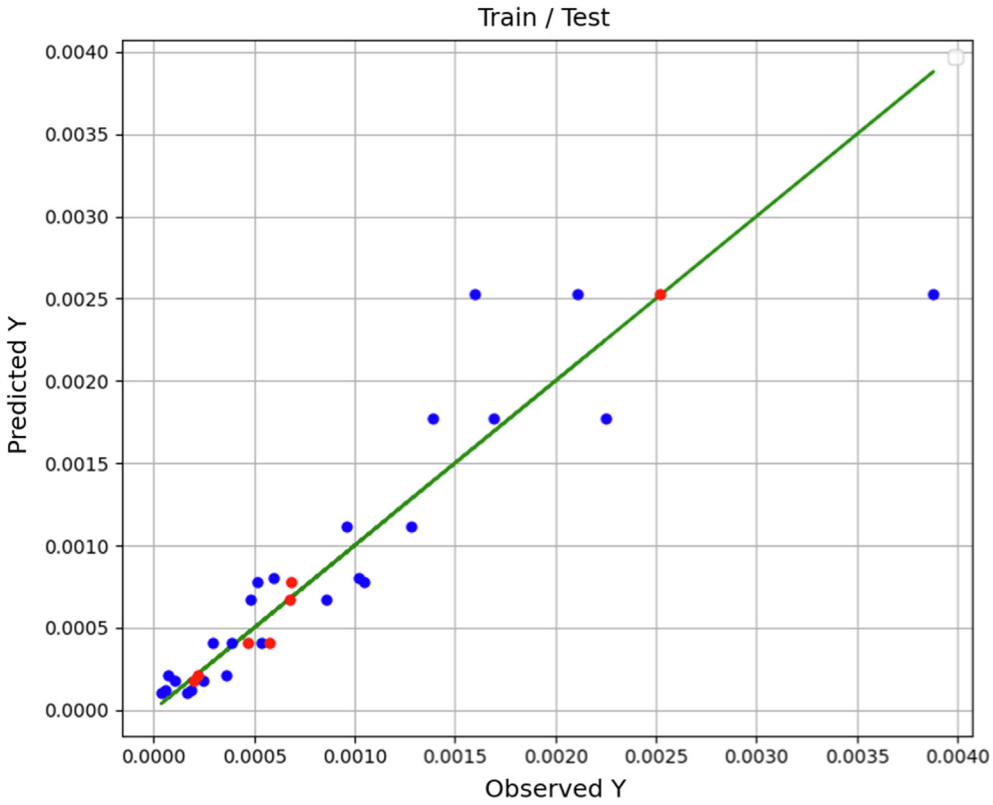

Fig. 1. Estimated vs actual values (DT Model).

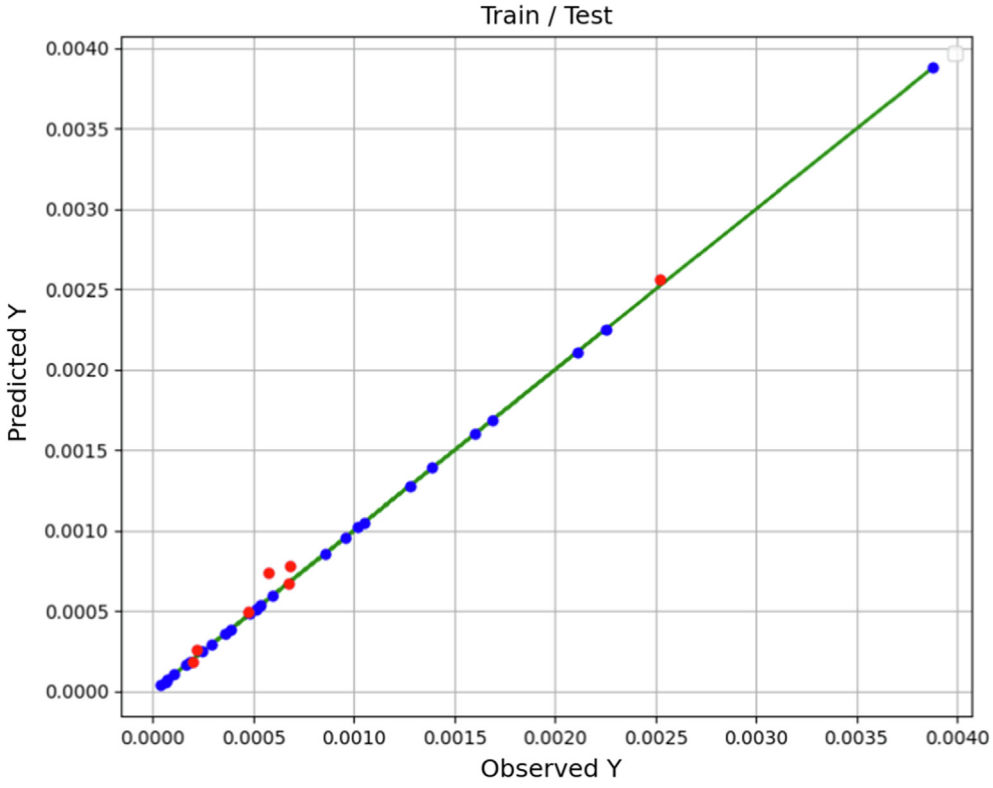

Fig. 2. Estimated vs actual values (KNN Model).

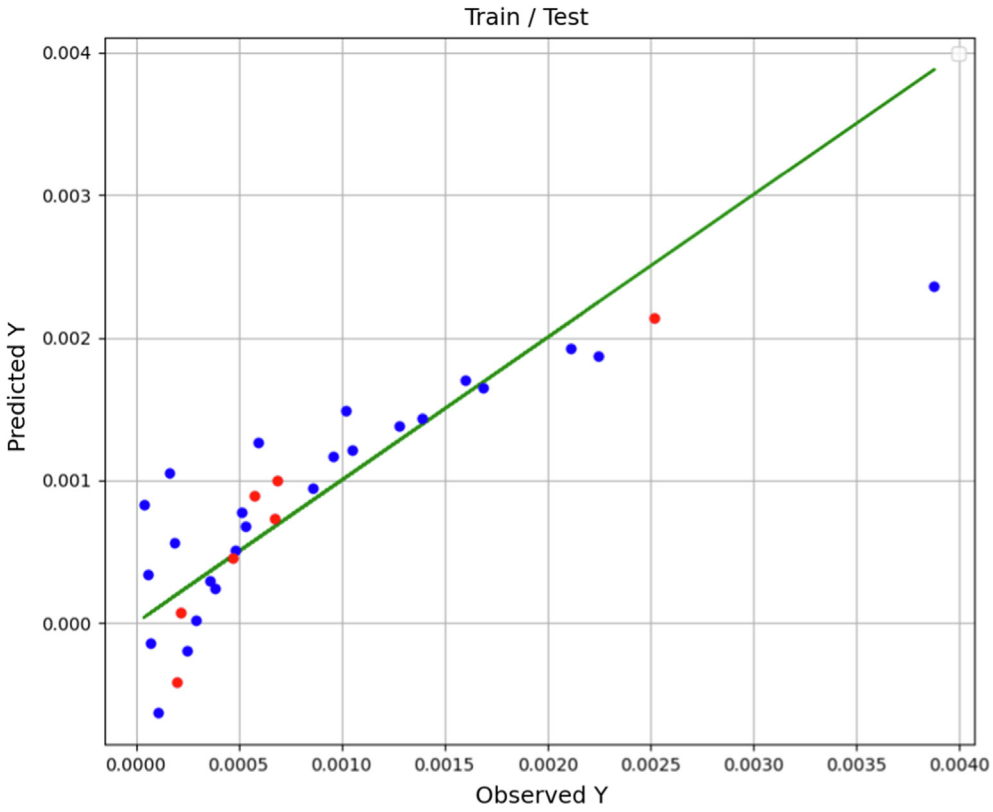

Fig. 3. Estimated vs actual values (MLP Model).

Table 2
Results.

Models	MAE	R^2	MAPE
DT	5.27E-05	0.974	1.10E-01
KNN	5.53E-05	0.996	1.07E-01
MLP	2.61E-04	0.809	7.13E-01

points are not too far from the line. Additionally, the R^2 value of this model is 0.996, which is significantly higher than other developed models.

Fig. 4 presents the final 3D result based on the KNN method to evaluate the effect of P and T on the solubility of salicylsalicylic acid as the only output. Furthermore, 2D changes of pressure and temperature against the solubility of salicylsalicylic acid are schematically illustrated in Figs. 5 and 6. For all considered isotherms, there is a direct connection between the pressure change and salicylsalicylic acid solubility. It means that increment of pressure intensifies the density of $SCCO_2$ system and therefore, increases the solubility amount of salicylsalicylic acid in $SCCO_2$. Although the relationship between pressure and solubility seems to be direct and clear, the existence of paradoxical factors has complexified the analysis of the temperature effect on the solubility. The solvent density and the sublimation pressure can be taken account as competing parameters, which significantly change the amount of salicylsalicylic acid solubility. Reduction of density (negative effect on solubility) and increment of the sublimation pressure (positive effect on solubility) are two paradoxical effects, which can be happened by increasing the temperature. Therefore, the net impact of the competing parameters can specify the desirable/undesirable role of temperature on the solubility. By analyzing the figures, it can

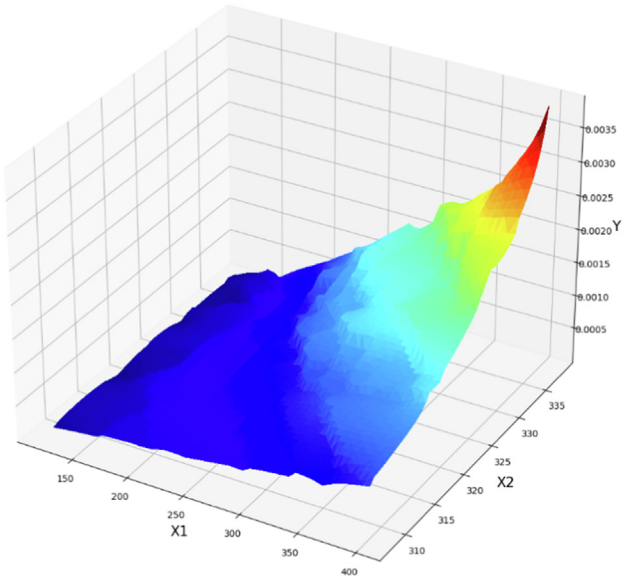

Fig. 4. 3D design of inputs/outputs (KNN Model).

be found that at the pressures higher than cross-over pressure (CP), the effect of sublimation pressure is stronger than the density. Therefore, increase of temperature results in the increment of solubility. For the pressures less than the CP, the impact of density overcomes the effect of sublimation pressure. It means that when temperature increases at the pressure lower than CP, the density of the $SCCO_2$ system reduces and consequently the solubility of salicylsalicylic acid declines. Considering Table 3, it is understood that

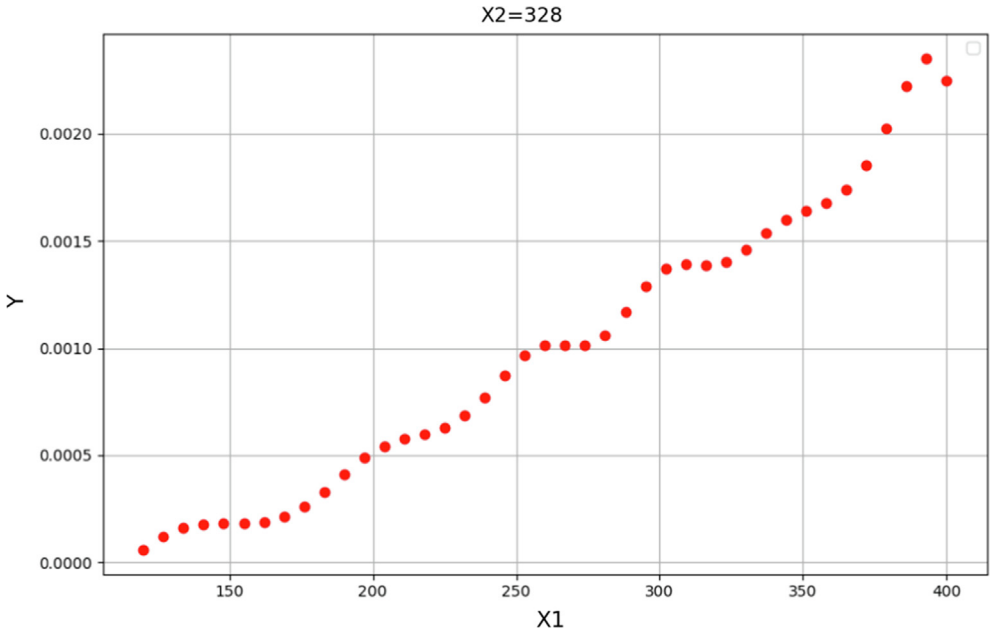

Fig. 5. Tendency of X1.

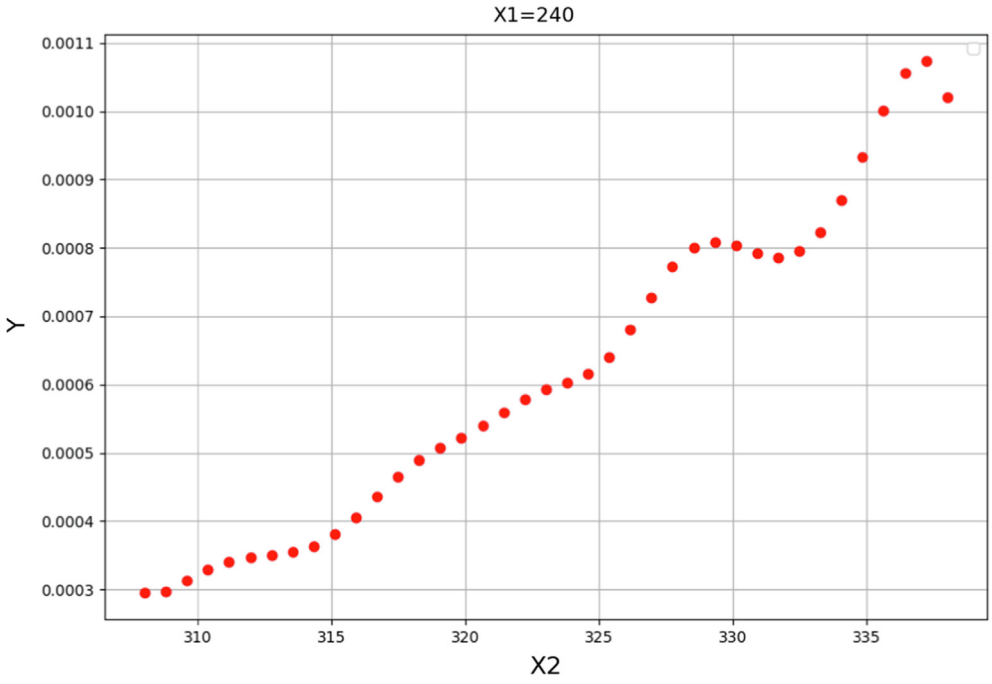

Fig. 6. Tendency of X2.

the pressure and the temperature of 400 bar and 338 K are the optimized parameters for achieving the greatest amount of salicylsalicylic acid solubility.

4.1. Polypharmacology and repurposing of Salsalate

Super-Pred obtained results revealed the ability of Salsalate to target several proteins based on its fingerprint similarity with their ligands. Table 4 illustrates the obtained results from Super-pred, we have selected proteins predicted to be targeted by Salsalate with probability percent more than 85 and the accuracy of the used model to be more than 90%. The obtained results with model

Table 3
Parameters optimization for the maximum response.

X1 = P	X2 = T	Y
400	338.0	0.00388

Table 4
Predicted putative protein targets for Salsalate obtained by Super-Pred.

Target Name	ChEMBL-ID	Pdb ID	Probability	Model accuracy
Transcription intermediary factor 1-alpha	CHEMBL3108638	4YBM	90.6%	95.56%
Cathepsin D	CHEMBL2581	4OD9	89.23%	98.95%
Muscarinic acetylcholine receptor M5	CHEMBL2035	6OL9	88.25%	94.62%
Transthyretin	CHEMBL3194	6SUG	88.04%	90.71%
Nuclear factor NF-kappa-B p105 subunit	CHEMBL3251	1SVC	87.5%	96.09%
DNA-(apurinic or apyrimidinic site) lyase	CHEMBL5619	6BOW	87.18%	91.11%
Glucose transporter	CHEMBL2535	6THA	85.1%	98.75%

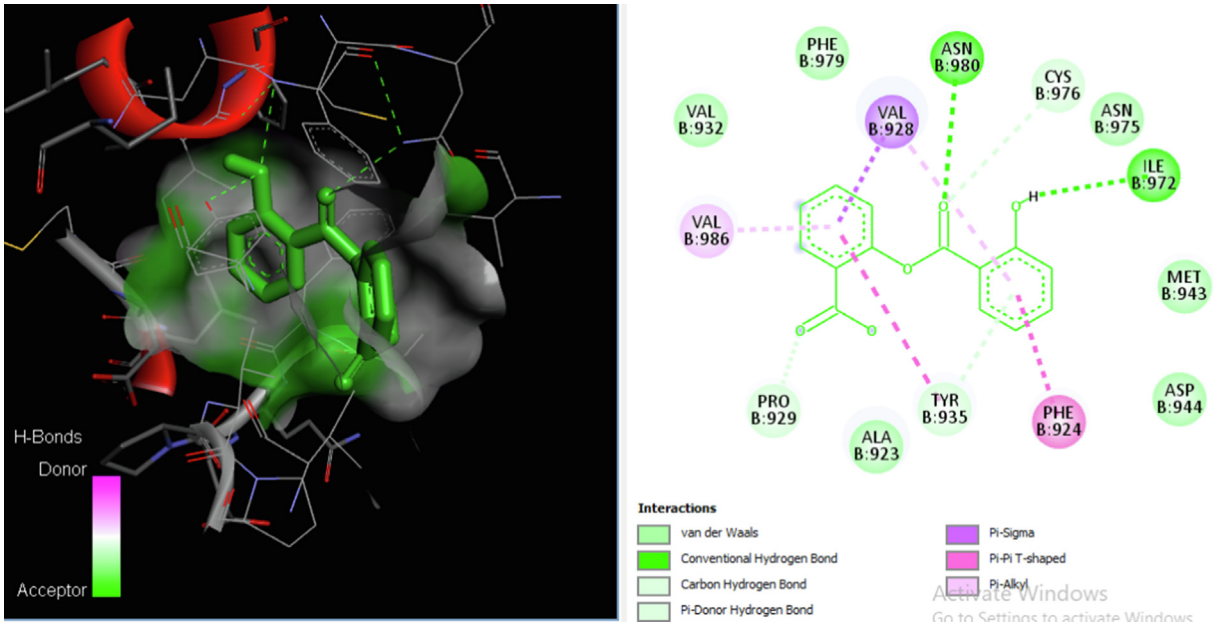

Fig. 7. 3D binding mode and 2D interactions of salsalate into Transcription intermediary factor 1-alpha protein binding site (Pdb ID: 4YBM).

accuracy more than 90% and probability more than 85% showed the ability of Salsalate to target the following proteins: Transcription intermediary factor 1-alpha, Cathepsin D, Muscarinic acetylcholine receptor M5, Transthyretin, Nuclear factor NF-kappa-B p105 subunit, DNA- (apurinic or apyrimidinic site) lyase and Glucose transporter.

4.2. Molecular docking studies

Further investigation through molecular docking studies was done using Transcription intermediary factor 1-alpha protein as a plausible target because the probability of targeting this protein with Salsalate exceeds 90% and the accuracy of the model exceeds 95% as revealed from super-pred, so we decided to focus on this crucial target to explore the binding mode, possible interactions, and affinity with Salsalate. The Pdb file used is 4YBM, the obtained results showed that salsalate has binding affinity score equal −7.8 Kcal/mol, additionally it has showed an ability to form hydrogen bonding and hydrophobic interactions also (Fig. 7).

5. Conclusion

Mathematical estimation of salicylsalicylic acid in $SCCO_2$ system and its comparison with obtained experimental data using machine learning (ML) approach is the main purpose of this manuscript. This research project is a regression problem with 32 input vectors, which is currently being investigated. Among the features in this dataset are two input features (the letters P and T) and one output feature. When utilizing the MAPE measure, we used Decision Tree (DT), K-nearest neighbor (KNN), and Multilayer perceptron (MLP) regression models, which had error rates of 1.10E-01, 1.07E-01, and 7.13E-01, respectively, when using the Decision Tree (DT) model. We also used Decision Tree (DT) and K-nearest neighbor (KNN) regression models to analyze the data. Furthermore, the R-squared scores for the DT, KNN, and MLP models are 0.974, 0.996, and 0.809, respectively, for the three models. The final statistic is MAE, in which the error rates of models are 5.27E-05 for DT, 5.53E-05 for KNN, and 2.61E-04 for MLP. The error rates of models are 5.27E-05 for DT, 5.53E-05 for KNN, and 2.61E-04 for MLP. 5.27E-05 is the error rate of the DT, KNN, and MLP algorithms. Finally, the KNN model was the most general, with optimal values of P = 400, T = 338.0, and Y = 0.00388 being achieved for the parameters.

Data Availability

All data are available within the published paper.

CRediT authorship contribution statement

Peijun Zhang: Supervision, Validation, Conceptualization, Data curation, Investigation, Methodology, Project administration,

Writing - original draft, Writing - review & editing. **Mustafa Fahem Albaghdadi:** Conceptualization, Formal analysis, Investigation, Methodology, Resources. **Sabah Auda AbdulAmeer:** Investigation, Resources, Data curation, Formal analysis. **Abdulmalik S. Altamimi:** Investigation, Resources, Formal analysis, Validation, Writing - original draft. **Ali Zeinulabdeen Abdulrazzaq:** Investigation, Resources, Software, Visualization, Writing - review & editing. **Hayder chailibi:** Investigation, Resources, Methodology, Software, Validation, Visualization. **Salema K. Hadrawi:** Investigation, Resources, Formal analysis, Validation, Conceptualization, Data curation, Software. **Hassan Falih Hamdan:** Investigation, Resources, Validation, Data curation, Methodology, Software, Visualization, Writing - original draft. **Farag M.A. Altalbawy:** Investigation, Resources, Validation, Software, Visualization, Writing - review & editing. **Amal M. Alsubaiyel:** Supervision, Validation, Conceptualization.

Data availability

No data was used for the research described in the article.

Declaration of Competing Interest

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.

Acknowledgement:

Research were supported Natural Science Basic Research Program of Shaanxi (Program No. 2022JM-029) and The Fourteenth Five Year Plan for Educational Science in Shaanxi Province (Program No. SGH21Y0286) and Horizontal projects (Program No. 2022610002003337).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.molliq.2022.121195.

References

[1] V. Patel, M. Shah, A comprehensive study on artificial intelligence and machine learning in drug discovery and drug development, Intelligent Medicine (2021).
[2] C. Ma, Y. Peng, H. Li, W. Chen, Organ-on-a-chip: a new paradigm for drug development, Trends Pharmacol. Sci. 42 (2) (2021) 119–133.
[3] J. Rantanen, J. Khinast, The future of pharmaceutical manufacturing sciences, J. Pharm. Sci. 104 (11) (2015) 3612–3638.
[4] F.M. Kerton, Solvent systems for sustainable chemistry, Sustainable Inorg. Chem. (2016) 193.
[5] K. Grodowska, A. Parczewski, Organic solvents in the pharmaceutical industry, Acta Pol. Pharm. Drug Res. 67 (1) (2010).
[6] M.P. Razgonova et al., Supercritical fluid technology and supercritical fluid chromatography for application in ginseng extracts, Farmacia 67 (2) (2019) 202.
[7] L. Baldino, M. Scognamiglio, E. Reverchon, Supercritical fluid technologies applied to the extraction of compounds of industrial interest from Cannabis sativa L. and to their pharmaceutical formulations: A review, J. Supercrit. Fluids 165 (2020).
[8] S.D. Manjare, K. Dhingra, Supercritical fluids in separation and purification: A review, Mater. Sci. Energy Technol. 2 (3) (2019) 463–484.
[9] L.-H. Wang, S.-T. Lin, A predictive method for the solubility of drug in supercritical carbon dioxide, J. Supercrit. Fluids 85 (2014) 81–88.
[10] C.M. Bishop, N.M. Nasrabadi, Pattern recognition and machine learning, vol. 4, Springer, 2006.
[11] V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, M. Chica-Rivas, Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines, Ore Geol. Rev. 71 (2015) 804–818.
[12] H. Wang et al., Machine learning basics, Deep learning (2016) 98–164.
[13] L. Noriega, Multilayer perceptron tutorial, School of Computing, Staffordshire University, 2005.
[14] E. Agirre-Basurko, G. Ibarra-Berastegi, I. Madariaga, Regression and multilayer perceptron-based models to forecast hourly O3 and NO2 levels in the Bilbao area, Environ. Model. Softw. 21 (4) (2006) 430–446.
[15] S. Zhang, X. Li, M. Zong, X. Zhu, R. Wang, Efficient kNN classification with different numbers of nearest neighbors, IEEE Trans. Neural Networks Learn. Syst. 29 (5) (2018) 1774–1785.
[16] T. Cover, Estimation by the nearest neighbor rule, IEEE Trans. Inf. Theory 14 (1) (1968) 50–55.
[17] Y. Song, J. Liang, J. Lu, X. Zhao, An efficient instance selection algorithm for k nearest neighbor regression, Neurocomputing 251 (2017) 26–34.
[18] M. Mathuria, Decision tree analysis on j48 algorithm for data mining, Int. J. Adv. Res. Computer Sci. Software Eng. 3 (6) (2013).
[19] J.R. Quinlan, Learning decision tree classifiers, ACM Computing Surveys (CSUR) 28 (1) (1996) 71–72.
[20] L. Yang, S. Liu, S. Tsoka, L.G. Papageorgiou, A regression tree approach using mathematical programming, Expert Syst. Appl. 78 (2017) 347–357.
[21] W. Liu, R. Zhao, X. Su, A. Mohamed, T. Diana, Development and validation of machine learning models for prediction of nanomedicine solubility in supercritical solvent for advanced pharmaceutical manufacturing, J. Mol. Liq. 358 (2022) 119208.
[22] L. Breiman, et al., Classification and regression trees, Routledge, 2017.
[23] J.R. Quinlan, C4. 5: programs for machine learning, Elsevier, 2014.
[24] M. Xu, P. Watanachaturaporn, P. Varshney, M. Arora, Decision tree regression for soft classification of remote sensing data, Remote Sens. Environ. 97 (3) (2005) 322–336.
[25] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (4) (1943) 115–133.
[26] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (6) (1958) 386.
[27] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
[28] M.T. Hagan, H.B. Demuth, M. Beale, Neural network design, PWS Publishing Co., 1997.
[29] M.W. Gardner, S. Dorling, Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmos. Environ. 32 (14–15) (1998) 2627–2636.
[30] A.B. Nassif, D. Ho, L.F. Capretz, Towards an early software estimation using log-linear regression and a multilayer perceptron model, J. Syst. Softw. 86 (1) (2013) 144–160.
[31] M. Ture, I. Kurt, A. Turhankurum, K. Ozdamar, Comparing classification techniques for predicting essential hypertension, Expert Syst. Appl. 29 (3) (2005) 583–588.
[32] K. Crammer, et al., Online passive aggressive algorithms, 2006.
[33] P. Lerman, Fitting segmented regression models by grid search, J. Roy. Stat. Soc.: Ser. C (Appl. Stat.) 29 (1) (1980) 77–84.