

## Article

# Application of CO<sub>2</sub> Supercritical Fluid to Optimize the Solubility of Oxaprozin: Development of Novel Machine Learning Predictive Models

Saad M. Alshahrani <sup>1,\*</sup>, Ahmed Al Saqr <sup>1</sup>, Munerah M. Alfadhel <sup>1</sup>, Abdullah S. Alshetaili <sup>1</sup>, Bjad K. Almutairy <sup>1</sup>, Amal M. Alsubaiyel <sup>2,\*</sup>, Ali H. Almari <sup>3</sup>, Jawaher Abdullah Alamoudi <sup>4</sup> and Mohammed A. S. Abourehab <sup>5,6,\*</sup>

- <sup>1</sup> Department of Pharmaceutics, College of Pharmacy, Prince Sattam Bin Abdulaziz University, P.O. Box 173, Al-Kharj 11942, Saudi Arabia
- <sup>2</sup> Department of Pharmaceutics, College of Pharmacy, Qassim University, Buraidah 52571, Saudi Arabia
- <sup>3</sup> Department of Pharmaceutics, College of Pharmacy, King Khalid University, Abha 62529, Saudi Arabia
- <sup>4</sup> Department of Pharmaceutical Sciences, College of Pharmacy, Princess Nourah bint Abdulrahman University, Riyadh 145111, Saudi Arabia
- <sup>5</sup> Department of Pharmaceutics, Faculty of Pharmacy, Umm Al-Qura University, Makkah 21955, Saudi Arabia
- <sup>6</sup> Department of Pharmaceutics and Industrial Pharmacy, College of Pharmacy, Minia University, Minia 61519, Egypt
- \* Correspondence: sm.alshahrani@psau.edu.sa (S.M.A.); asbiel@qu.edu.sa (A.M.A.); maabourehab@uqu.edu.sa (M.A.S.A.)

**Abstract:** Over the last years, extensive motivation has emerged towards the application of supercritical carbon dioxide (SCCO<sub>2</sub>) for particle engineering. SCCO<sub>2</sub> has great potential for application as a green and eco-friendly technique to reach small crystalline particles with narrow particle size distribution. In this paper, an artificial intelligence (AI) method has been used as an efficient and versatile tool to predict and consequently optimize the solubility of oxaprozin in SCCO<sub>2</sub> systems. Three learning methods, including multi-layer perceptron (MLP), Kriging or Gaussian process regression (GPR), and k-nearest neighbors (KNN) are selected to make models on the tiny dataset. The dataset includes 32 data points with two input parameters (temperature and pressure) and one output (solubility). The optimized models were tested with standard metrics. MLP, GPR, and KNN have error rates of  $2.079 \times 10^{-8}$ ,  $2.173 \times 10^{-9}$ , and  $1.372 \times 10^{-8}$ , respectively, using MSE metrics. Additionally, in terms of R-squared, they have scores of 0.868, 0.997, and 0.999, respectively. The optimal inputs are the same as the maximum possible values and are paired with a solubility of  $1.26 \times 10^{-3}$  as an output.

**Keywords:** optimization; solubility; mathematical modeling; green chemistry; machine learning



**Citation:** Alshahrani, S.M.; Saqr, A.A.; Alfadhel, M.M.; Alshetaili, A.S.; Almutairy, B.K.; Alsubaiyel, A.M.; Almari, A.H.; Alamoudi, J.A.; Abourehab, M.A.S. Application of CO<sub>2</sub> Supercritical Fluid to Optimize the Solubility of Oxaprozin: Development of Novel Machine Learning Predictive Models. *Molecules* **2022**, *27*, 5762. <https://doi.org/10.3390/molecules27185762>

Academic Editor: Francisco Torrens

Received: 1 August 2022

Accepted: 17 August 2022

Published: 6 September 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the last years, disparate scientific investigations have been conducted on advanced targeted drug delivery systems owing to the need of pharmaceutical industries. Indeed, developing appropriate methodologies for particle engineering with the aim of controlling particle size is of great importance due to the drastic impact of this parameter on the drug delivery route [1–3].

A supercritical fluid (SCF) is identified as any fluid above critical pressure/temperature, where its density follows the behavior of liquids, but its viscosity and diffusivity follow a manner between liquid and gas. Moreover, SCFs possess a surface tension near zero. Considering their brilliant transport characteristics, SCFs have been of great interest for application in various industrial activities like extractions, chromatography, and particle engineering [4–8]. SCFs can be an appropriate option for poisonous and explosive light hydrocarbons and organic solvents [9–13]. Amongst various SCFs, it seems that carbon dioxide SCF (CO<sub>2</sub>SCF) can be considered the only commonly applied “green solvent” due to its very low flammability, inert nature, simplicity of utilization, and low threshold limit value

(TLV). It is worth noting that the TLV amount of CO<sub>2</sub>SCF is significantly more eco-friendly and less poisonous than acetone (TLV = 750 ppm) or pentane (TLV = 600 ppm) [14].

Nowadays, the development of mathematical modeling and numerical simulations to compare the experimental (real) results with predicted ones is an important and efficient activity towards moving the quality-by-design (QbD) paradigm in the pharmaceutical industry [15–17]. Artificial intelligence (AI) is a novel and promising technique for developing predictive models in disparate industrial processes, such as membrane-based separation, crystallization, coating, and chemical reactions [18–21].

Machine learning (ML) methods are gradually replacing traditional computing methods in a variety of scientific disciplines. These problem-solving strategies include neural networks, ensemble models, and tree-based models. Machine learning models may now be used to study many difficulties by several initial properties and several final amounts. The correlation among initial amount and final values are derived by these methods [22–24]. In this work, three distinct methods including GPR, KNN, and MLP are selected to make models on the available dataset.

GPR has recently gotten much attention as a powerful statistical technique for data-driven modeling. GPR's popularity stems partly from its theoretical connection to Bayesian nonparametric statistics, infinite neural networks, kernel approaches in machine learning, and spatial statistics [25,26].

The name “MLP” refers to a multi-layer perceptron-based neural network. MLPs are forward-feeding artificial neural networks. MLP has at least three levels: inputs, outputs, and hidden layers. The input layer nodes are not active; instead, the input layer nodes represent the data point. The input layer will have  $d$  nodes if the data point is presented by a  $d$ -dimensional vector [27,28].

The central idea of  $k$ -nearest neighbors (KNN) models is that they use the similarity of input data attributes to generate forecasts using other points that are most like the first. More specifically, it retains the entire training data during the testing phase [29,30].

## 2. Data Set

The used dataset of this study was taken from [31] which only has 32 data vectors. Each vector contains two input parameters (temperature and pressure) and one output (solubility). The dataset is shown in Table 1 and pairwise distribution of parameters is displayed in Figure 1.

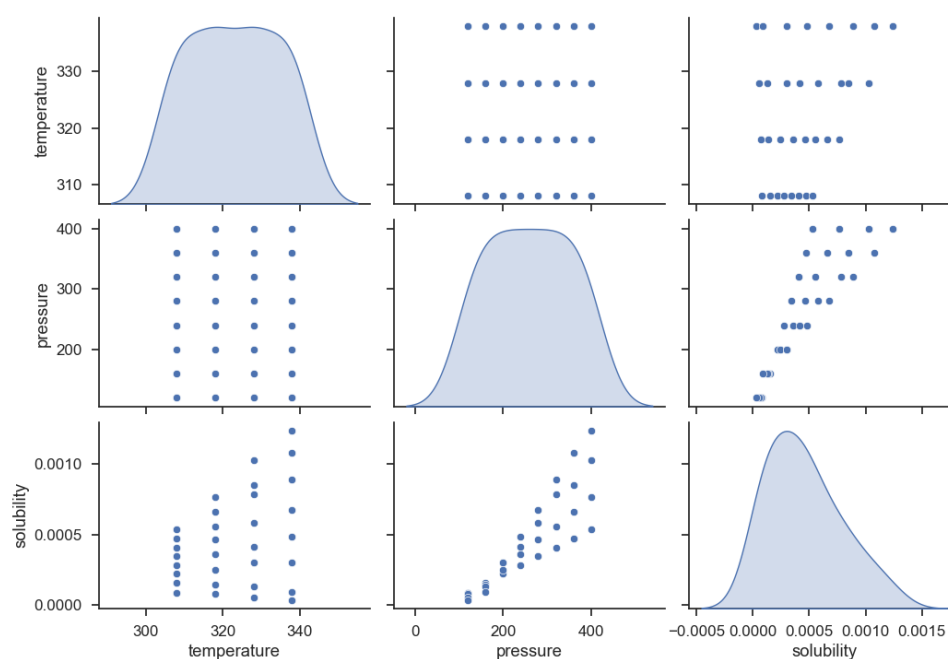


Figure 1. Pairwise Distribution of variables.

**Table 1.** The dataset based on input and output.

No.	Temperature (K)	Pressure (bar)	Solubility (Mole Fraction)
1	$3.08 \times 10^2$	$1.20 \times 10^2$	$8.19 \times 10^{-5}$
2	$3.08 \times 10^2$	$1.60 \times 10^2$	$1.58 \times 10^{-4}$
3	$3.08 \times 10^2$	$2.00 \times 10^2$	$2.24 \times 10^{-4}$
4	$3.08 \times 10^2$	$2.40 \times 10^2$	$2.80 \times 10^{-4}$
5	$3.08 \times 10^2$	$2.80 \times 10^2$	$3.44 \times 10^{-4}$
6	$3.08 \times 10^2$	$3.20 \times 10^2$	$4.06 \times 10^{-4}$
7	$3.08 \times 10^2$	$3.60 \times 10^2$	$4.73 \times 10^{-4}$
8	$3.08 \times 10^2$	$4.00 \times 10^2$	$5.33 \times 10^{-4}$
9	$3.18 \times 10^2$	$1.20 \times 10^2$	$7.55 \times 10^{-5}$
10	$3.18 \times 10^2$	$1.60 \times 10^2$	$1.41 \times 10^{-4}$
11	$3.18 \times 10^2$	$2.00 \times 10^2$	$2.45 \times 10^{-4}$
12	$3.18 \times 10^2$	$2.40 \times 10^2$	$3.56 \times 10^{-4}$
13	$3.18 \times 10^2$	$2.80 \times 10^2$	$4.64 \times 10^{-4}$
14	$3.18 \times 10^2$	$3.20 \times 10^2$	$5.58 \times 10^{-4}$
15	$3.18 \times 10^2$	$3.60 \times 10^2$	$6.60E \times 10^{-4}$
16	$3.18 \times 10^2$	$4.00 \times 10^2$	$7.66 \times 10^{-4}$
17	$3.28 \times 10^2$	$1.20 \times 10^2$	$5.34 \times 10^{-4}$
18	$3.28 \times 10^2$	$1.60 \times 10^2$	$1.28 \times 10^{-4}$
19	$3.28 \times 10^2$	$2.00 \times 10^2$	$3.02 \times 10^{-4}$
20	$3.28 \times 10^2$	$2.40 \times 10^2$	$4.14 \times 10^{-4}$
21	$3.28 \times 10^2$	$2.80 \times 10^2$	$5.82 \times 10^{-4}$
22	$3.28 \times 10^2$	$3.20 \times 10^2$	$7.87 \times 10^{-4}$
23	$3.28 \times 10^2$	$3.60 \times 10^2$	$8.51 \times 10^{-4}$
24	$3.28 \times 10^2$	$4.00 \times 10^2$	$1.03 \times 10^{-3}$
25	$3.38 \times 10^2$	$1.20 \times 10^2$	$3.3 \times 10^{-5}$
26	$3.38 \times 10^2$	$1.60 \times 10^2$	$9.09 \times 10^{-5}$
27	$3.38 \times 10^2$	$2.00 \times 10^2$	$2.98 \times 10^{-4}$
28	$3.38 \times 10^2$	$2.40 \times 10^2$	$4.81 \times 10^{-4}$
29	$3.38 \times 10^2$	$2.80 \times 10^2$	$6.77 \times 10^{-4}$
30	$3.38 \times 10^2$	$3.20 \times 10^2$	$8.89 \times 10^{-4}$
31	$3.38 \times 10^2$	$3.60 \times 10^2$	$1.08 \times 10^{-3}$
32	$3.38 \times 10^2$	$4.00 \times 10^2$	$1.24 \times 10^{-3}$

### 3. Methodology

#### 3.1. Gaussian Process Regression

Based on the Bayesian theory, it is possible to consider the GPR as a random process that employs the Gaussian processes to implement a nonparametric regression [32,33]. In this case, according to the Gaussian distribution, the probability distribution over function ( $x$ ) for each input is determined as follows:

$$f(x) \sim GPR(m(x), k(x, x')) \quad (1)$$

Here  $\mu(x)$  and  $\Sigma(x, x')$  represent the mean and covariance functions, respectively. These functions are computed using the following equations:

$$\begin{cases} \mu(x) = E(f(x)) \\ \Sigma(x, x') = E[(f(x) - \mu(x))(f(x') - \mu(x')))] \end{cases} \quad (2)$$

In which  $E(\cdot)$  denotes the expectation value. In practice, the value of  $\mu(x)$  is usually considered equal to zero for simplifying the process of calculation. It should be noted that this assumption leads to erroneous results [32]. For describing the correlation degree between an expected target value of the training data set and the predicted target according to the resemblance of the respective inputs, the  $\Sigma(x, x')$  is also called the kernel function.

In a regression problem, the prior distribution of outputs  $y$  is defined as follows:

$$y \sim N(0, \Sigma(x, x) + \sigma_n^2 I_n) \quad (3)$$

where  $N(\cdot)$ , and  $\sigma_n$  specify a normal distribution and the noise term, respectively. It is assumed that a similar Gaussian distribution exists between the testing subset  $x'$  and training subset  $x$ . In this case, the forecast outputs  $y'$  would track a joint prior distribution through the training output  $y$  as [34]:

$$\begin{bmatrix} y \\ y' \end{bmatrix} \sim N\left(0, \begin{bmatrix} \Sigma(x, x) + \sigma_n^2 I_n & \Sigma(x, x') \\ \Sigma(x, x')^T & \Sigma(x', x') \end{bmatrix}\right) \quad (4)$$

Here  $\Sigma(x, x)$ ,  $\Sigma(x', x')$ , and  $\Sigma(x, x')$  denote the covariance matrices between input variables from the training set, testing set, and training-testing sets, correspondingly.

In the training process, with the help of the  $n$  points, some hyper-parameters  $\theta$  present in the covariance function are optimized to warranty the application of GPR. Minimizing the negative log marginal likelihood  $L(\theta)$  is a way to reach an optimized answer as [35]:

$$\begin{cases} L(\theta) = \frac{1}{2} \log[\det \lambda(\theta)] + \frac{1}{2} y^T \lambda^{-1}(\theta) y + \frac{n}{2} \log(2\pi) \\ \lambda(\theta) = \Sigma(\theta) + \sigma_n^2 I_n \end{cases} \quad (5)$$

As the optimized settings of the hyper parameters of GPR are determined, the forecast output  $y'$  is calculated at dataset  $x'$  by determining the related conditional distribution  $p(y' | x', x, y)$  as:

$$p(y' | x', x, y) \sim N(y' | \bar{y}', \text{cov}(y')) \quad (6)$$

with:

$$\begin{cases} \bar{y}' = \Sigma(x, x')^T [\Sigma(x, x) + \sigma_n^2 I_n]^{-1} y \\ \text{cov}(y') = \Sigma(x', x') - \Sigma(x, x')^T [\Sigma(x, x) + \sigma_n^2 I_n]^{-1} \Sigma(x, x') \end{cases} \quad (7)$$

In which  $\bar{y}'$  represents the related mean values of the forecast.  $\text{cov}(y')$  represents a variance matrix to determine the uncertainty range of these forecasts. These equations of GPR are explained in detail in [32].

### 3.2. Multilayer Perceptron Neural Networks

The feed-forward neural networks that include several latent layers are known as the multi-layer perceptron (MLP). One of the ways to develop the broadly employed MLP is the training rule of back propagation, which is rooted in the learning rule of error-correction (it is equivalent to traveling in the minus orientation of the immediate deviation according to the error function, that decreases the mistakes) [36,37].

The rule of back propagation includes two methods:

- First, the input vector is involved to the multilayer network, and its influences are transferred to the output levels over the hidden (middle) layers. Then, the final vector created on the latent class generates the genuine response of MLP.

- Next, in the backward path, the MLP settings are updated and regulated. The rule of error-correction will be followed in the implementation of this regulation. Furthermore, in the middle layers, weights of neurons are adjusted to reduce the difference between the neural network's predicted results and its actual results [38,39].

When the ANN is developed, the data will be basically split into two periods of trial. No rules are available to minimize the size of training and test datasets [40,41].

In MLP, the procedure starts only with initial class and proceeds up to the nerve cells in the final class to yield some results. A hidden class is any layer that exists between these two layers (input and output). The activation functions, solver function, and quantity of hidden layers are hyperparameters in this algorithm that should be optimized. The output formulation for the MLP model with one hidden layer and a single output is as follows [42,43]:

$$\tilde{\mathbf{y}} = \delta_2 \left( \sum_{i=1}^m (\mathbf{w}_i^{(2)} \delta_1(\mathbf{X})) + \mathbf{b}^{(2)} \right) \mathbf{X} = \sum_{j=1}^n (\mathbf{x}_j \mathbf{w}_{x_j}^{(1)}) + \mathbf{b}^{(1)} \quad (8)$$

Here,  $\tilde{\mathbf{y}}$  represents the estimation vector of the MLP model,  $m$  indicates the data vector amount in the entire data set,  $n$  denotes initial amount details in the dataset, and  $\mathbf{x}_j$  is  $j^{\text{th}}$  feature vector  $\mathbf{w}^{(2)}$  reflects the weights among the latent class and the final class, whereas  $\mathbf{w}^{(1)}$  indicates the weights of initial attributes linked to the latent class.  $\delta_2$  is the activation obligation for the final class [44]. In addition, the neurons' activation function is  $\delta_1$  in the latent class.  $\mathbf{b}^{(2)}$  and  $\mathbf{b}^{(1)}$  represent the bias vectors in the final class and all latent classes [45].

### 3.3. KNN

The KNN regression, starts to learn by contrasting the identified data points with the training dataset [30]. To explain this method, we assume  $T = \{(x_1, y_1), \dots, (x_N, y_N)\}$  is the training data and  $x_i = (x_{i1}, x_{i2}, \dots, x_{im})$  indicates the  $i^{\text{th}}$  data point with its  $m$  input features and the output is  $y_i$ .  $N$  represents the quantity of data points. It must calculate the  $d_i$  between a test instance  $x$  and every sample  $x_i$  in  $T$  and sort the  $d_i$  distance by its value for a test sample  $x$ . If  $d_i$  rates in the  $i^{\text{th}}$  position, the  $d_i$  matching sample is referred to as the  $i^{\text{th}}$  close to  $NNi(x)$ , then the target is denoted as  $y_i(x)$ . Lastly, the estimation  $\hat{y}$  of  $x$  denotes the average of the regression outputs of  $k^{\text{th}}$  close to  $x$ , i.e.,  $\hat{y} = \frac{1}{k} \sum_{i=1}^k y_i(x)$ . The KNN workflow regression algorithm is as follows [29,46]:

- Inputs: training input vectors  $\{x_i, y_i\}$ ,  $x_i$ : input features,  $y_i$ : real-valued output, testing point  $x$  to predict
- Algorithm:
  - calculate distance  $D(x, x_i)$  to every training example  $x_i$
  - select  $k^{\text{th}}$  nearest input vector  $x_{i1} \dots x_{ik}$  and their outputs  $y_{i1} \dots y_{ik}$
  - output:

$$\hat{y} = f(x) = \frac{1}{K} \sum_{j=1}^k y_{ij} \quad (9)$$

## 4. Results

In this section, after examining the adjusted hyper-parameters stated in the previous section, the final models will be generated and compared with the criteria in this field to evaluate and analyze the results of the suggested models with the data.  $R^2$ -score and MSE metrics are used in this study:

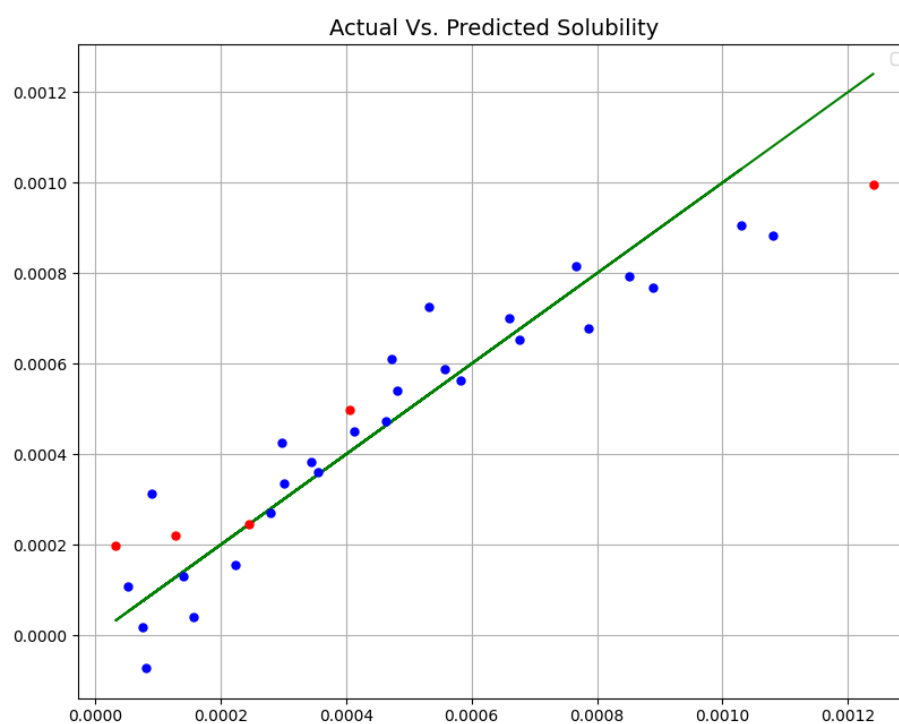
- For effective regression models, the R-square ( $R^2$ ) score is an actuarial metric. These graphs demonstrate a varies amount of percentage among related and non-aligned variables. It is critical to be able to quickly quantify the 0 to 100% difference among the related vary and the regression model.

- A mean squared error is one other standard metric for calculating the output of regression methods. MSE squares the points on the regression line. If the value's negative sign is deleted and larger variances are given more weight, the squared value becomes significant. The lower the mean fault, the better match you will detect. The sooner, the best.

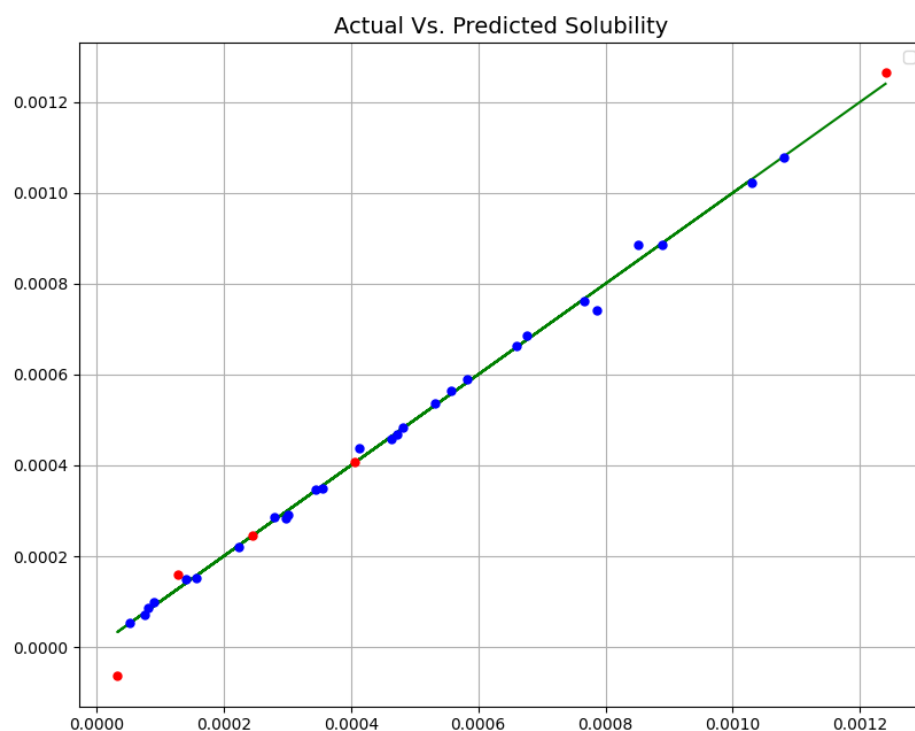
Table 2 enlists the final outcomes of all developed predictive models. Additionally, Figures 2–4 schematically compare actual (experimental) and predicted (model-based) values via three proposed models (MLP, GPR, and KNN). In all diagrams, blue dots show forestalled amount, red dots show forestalled amount in trial, and the green line shows the real amount. According to this table and figures, we chose the GPR model as the most accurate model among these three models. Although the KNN model also shows close and accurate results, it can be seen in the test data that two points have a distance from the real values, and it has more MAPE.

**Table 2.** Outputs based on  $R^2$  and MSE.

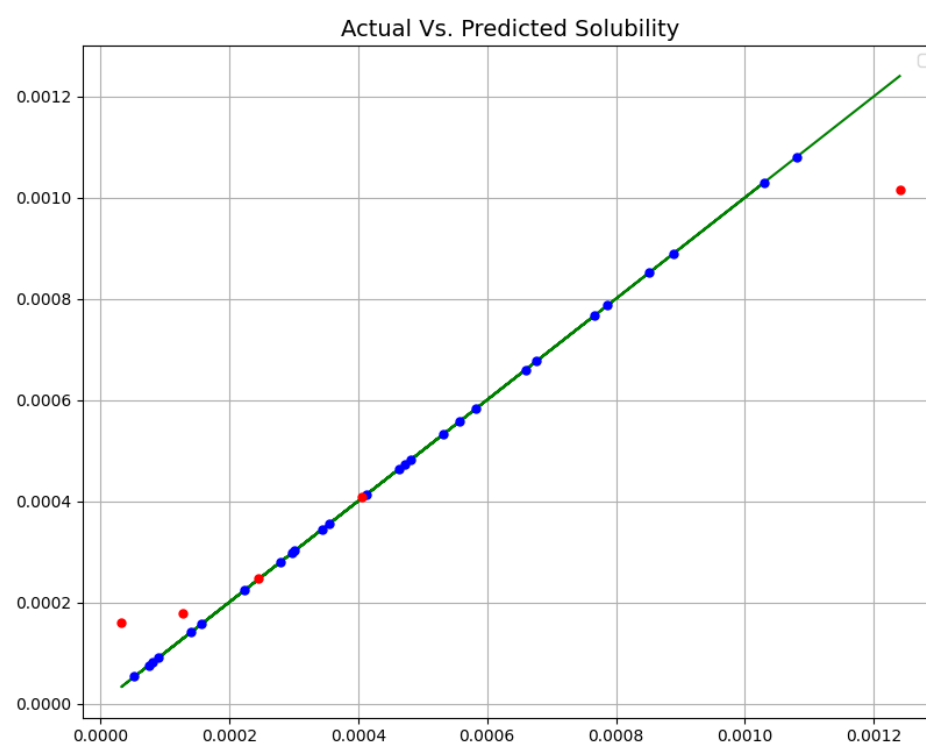
Models	$R^2$	MSE
MLP	0.868	$2.079 \times 10^{-08}$
GPR	0.997	$2.173 \times 10^{-09}$
KNN	0.999	$1.372 \times 10^{-08}$



**Figure 2.** Actual Vs. Predicted Solubility (mole fraction) (MLP).



**Figure 3.** Real Vs. Forestalled Solubility (mole fraction) (GPR).



**Figure 4.** Real Vs. Forestalled Solubility (mole fraction) (KNN).

The simultaneous influence of input values (temperature and pressure) on the solubility of oxaprozin is shown in Figure 5. If one of those parameters (temperature or pressure) keep constant, by changing the other one, two-dimensional Figures 6 and 7, have been provided can illustrate this fact. The optimized parameters are illustrated in Table 3.

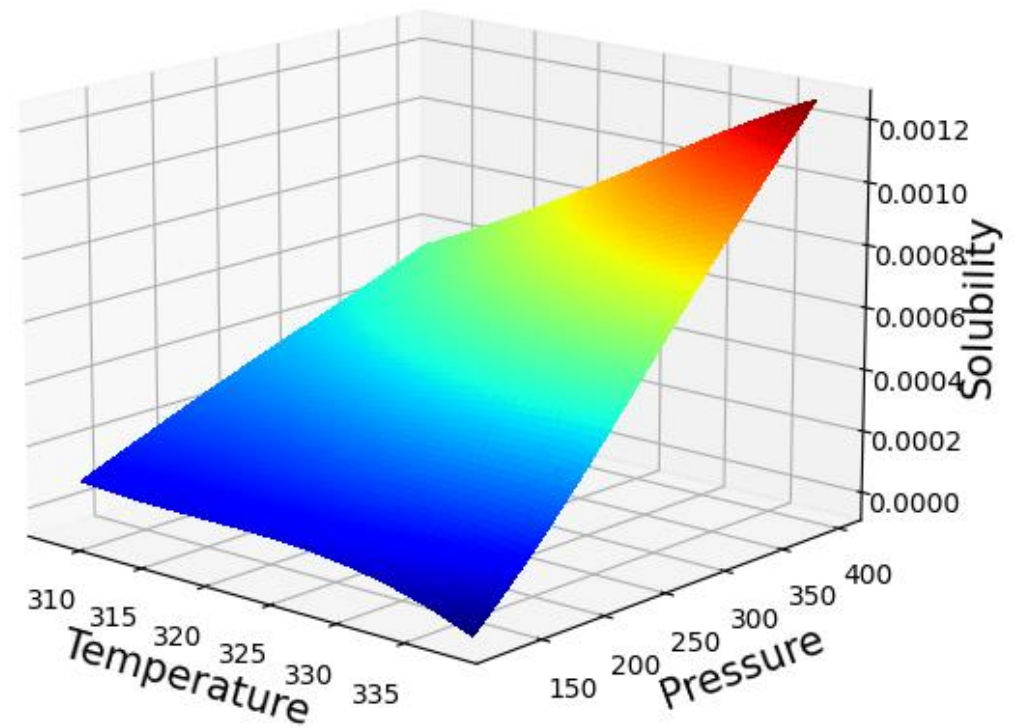


Figure 5. 3D projection with GPR Model (pressure, bar/temperature, K/solubility, mole fraction).

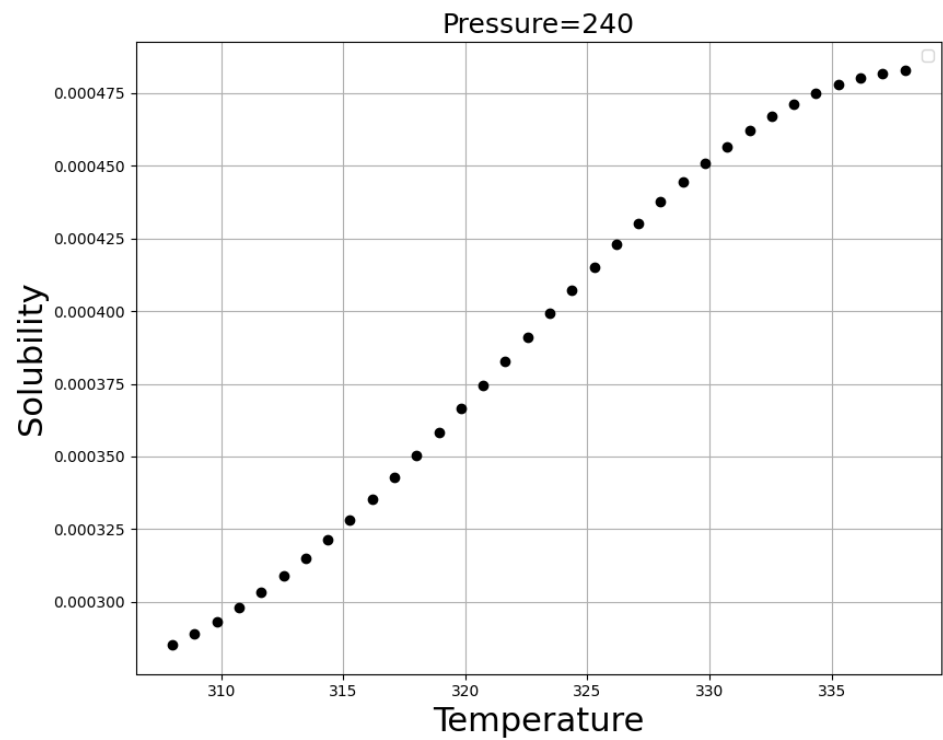


Figure 6. Trends for Temperature (K).



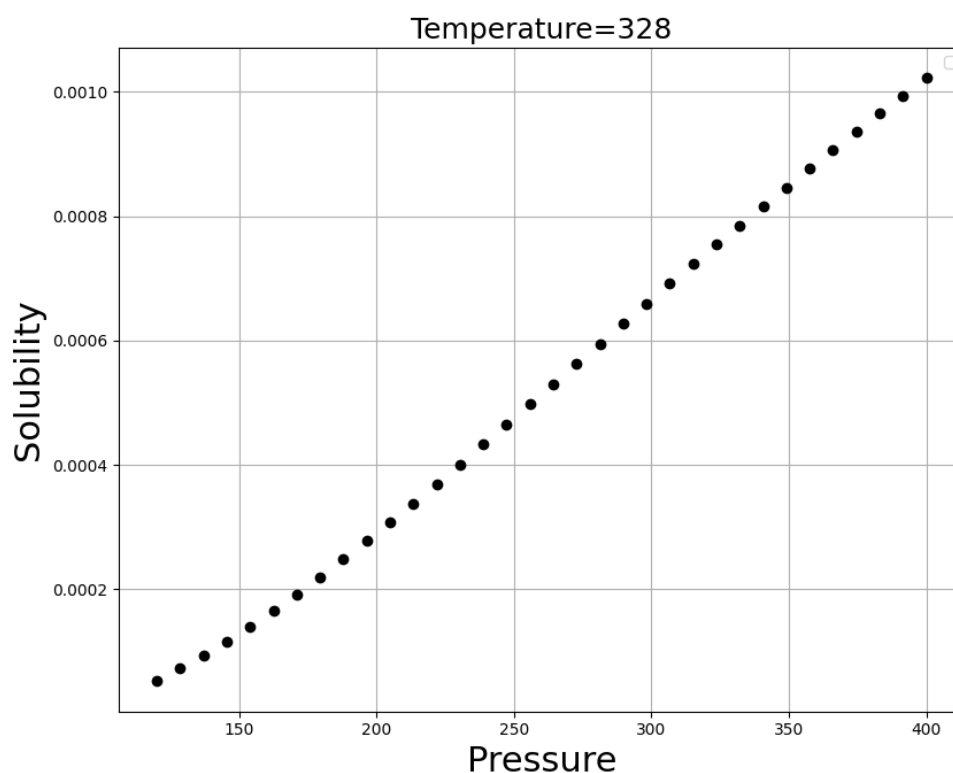


Figure 7. Trends for Pressure (bar).

Table 3. Optimized amount of inputs and output.

Temperature (K)	Pressure (bar)	Solubility
$3.38 \times 10^2$	$4.00 \times 10^2$	$1.26 \times 10^{-3}$

It is clear from Figures 6 and 7 that an increase in the pressure causes a significant enhancement in the solubility value of a drug, which can be attributed to the increase of the molecular compression of solvent and improvement in the solubilizing power of SCCO<sub>2</sub> [47–49]. Figure 7 demonstrates approximately five times improvement in the solubility of drugs by increasing the pressure from 120 to 410 bar. Regarding temperature, it can be said that this parameter has the opposite effect on the two competing parameters. An increase of temperature decreases the density of SCCO<sub>2</sub>, while increasing the sublimation pressure. Therefore, true analysis of these two parameters at the pressures below and above the cross-over pressure seems to be vital. At pressures below the cross-over pressure, the impact of density reduction prevails over the positive role of sublimation pressure and thus, an increase in temperature is equal to solubility reduction in SCCO<sub>2</sub> fluid. By increasing the pressure to values higher than the cross-over pressure, the positive role of sublimation pressure prevails over the destructive role of density reduction and thus, an increase in the temperature considerably increases the oxaprozin solubility in the supercritical solvent. As presented in Table 3, the optimum values of pressure and temperature to gain the greatest value of solubility are predicted to be 400 bar and 338 K, respectively.

## 5. Conclusions

This paper was prominently focused on the prediction of oxaprozin solubility in SCCO<sub>2</sub> fluid. To do this, machine learning (ML) techniques were employed to develop mathematical modeling and simulations to predict and optimize drug solubility. To make models on the small dataset, three learning methods were chosen: MLP, KNN, and GPR. There are 32 data points in the dataset, each with two input parameters (temperature and pressure) and one output parameter (solubility). Standard metrics were used to test

the optimized models. Using the MSE metric, MLP, GPR, and KNN have error rates of  $2.079 \times 10^{-8}$ ,  $2.173 \times 10^{-9}$ , and  $1.372 \times 10^{-8}$ , respectively. In addition, they have R-squared scores of 0.868, 0.997, and 0.999, respectively. The optimal inputs are identical to the maximum possible values, and the output is a solubility of  $1.26 \times 10^{-3}$ .

**Author Contributions:** S.M.A.: supervision, writing, software, editing, A.A.S.: writing, resources, editing, A.S.A.: editing, investigation, analysis, M.M.A.: visualization, methodology, software, writing, editing, B.K.A.: editing, analysis, validation, A.M.A.: writing, editing, investigation, analysis, A.H.A.: editing, resources, validation, J.A.A.: editing, analysis, investigation, M.A.S.A.: supervision, editing, writing, analysis. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data are available within the published paper.

**Acknowledgments:** The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IF-PSAU-2021/03/18826). The authors would like to thank the Deanship of scientific research at Umm Al-Qura University for supporting this work by grant code (22UQU4290565DSR62).

**Conflicts of Interest:** The authors declare no conflict of interest.

**Sample Availability:** Samples of the compounds are available from the authors.

## References

1. Taberner, A.; del Valle, E.M.M.; Galán, M.A. Supercritical fluids for pharmaceutical particle engineering: Methods, basic fundamentals and modelling. *Chem. Eng. Process. Process Intensif.* **2012**, *60*, 9–25.
2. Zhuang, W.; Hachem, K.; Bokov, D.; Ansari, M.J.; Nakhjiri, A.T. Ionic liquids in pharmaceutical industry: A systematic review on applications and future perspectives. *J. Mol. Liq.* **2021**, *349*, 118145.
3. Scherließ, R.; Bock, S.; Bungert, N.; Neustock, A.; Valentin, L. Particle engineering in dry powders for inhalation. *Eur. J. Pharm. Sci.* **2022**, *172*, 106158. [[PubMed](#)]
4. York, P. Strategies for particle design using supercritical fluid technologies. *Pharm. Sci. Technol. Today* **1999**, *2*, 430–440. [[CrossRef](#)]
5. Girotra, P.; Singh, S.K.; Nagpal, K. Supercritical fluid technology: A promising approach in pharmaceutical research. *Pharm. Dev. Technol.* **2013**, *18*, 22–38.
6. Chakravarty, P.; Famili, A.; Nagapudi, K.; Al-Sayah, M.A. Using supercritical fluid technology as a green alternative during the preparation of drug delivery systems. *Pharmaceutics* **2019**, *11*, 629.
7. Yang, G.; Li, Z.; Shao, Q.; Feng, N. Measurement and correlation study of silymarin solubility in supercritical carbon dioxide with and without a cosolvent using semi-empirical models and back-propagation artificial neural networks. *Asian J. Pharm. Sci.* **2017**, *12*, 456–463. [[CrossRef](#)]
8. Pitchaiah, K.; Lamba, N.; Deepitha, J.; Mohapatra, P.; Madras, G.; Sivaraman, N. Experimental measurements and correlation of the solubility of N, N-dialkylamides in supercritical carbon dioxide. *J. Supercrit. Fluids* **2019**, *143*, 162–170.
9. Knez, Ž.; Pantić, M.; Cör, D.; Novak, Z.; Hrnčič, M.K. Are supercritical fluids solvents for the future? *Chem. Eng. Process.-Process Intensif.* **2019**, *141*, 107532.
10. Nunes, A.N.; Roda, A.; Gouveia, L.S.F.; Fernández, N.; Bronze, M.R.R.; Matias, A.A. Astaxanthin extraction from marine crustacean waste streams: An integrate approach between microwaves and supercritical fluids. *ACS Sustain. Chem. Eng.* **2021**, *9*, 3050–3059.
11. Chrastil, J. Solubility of solids and liquids in supercritical gases. *J. Phys. Chem.* **1982**, *86*, 3016–3021. [[CrossRef](#)]
12. Bartle, K.; Clifford, A.; Jafar, S.; Shilstone, G. Solubilities of solids and liquids of low volatility in supercritical carbon dioxide. *J. Phys. Chem. Ref. Data* **1991**, *20*, 713–756. [[CrossRef](#)]
13. Su, C.-S.; Chen, Y.-P. Correlation for the solubilities of pharmaceutical compounds in supercritical carbon dioxide. *Fluid Phase Equilibria* **2007**, *254*, 167–173. [[CrossRef](#)]
14. Beckman, E.J. Supercritical and near-critical CO<sub>2</sub> in green chemical synthesis and processing. *J. Supercrit. Fluids* **2004**, *28*, 121–191.
15. Siepmann, J.; Siepmann, F. Mathematical modeling of drug dissolution. *Int. J. Pharm.* **2013**, *453*, 12–24. [[CrossRef](#)]
16. Ramteke, K.; Dighe, P.; Kharat, A.; Patil, S. Mathematical models of drug dissolution: A review. *Sch. Acad. J. Pharm* **2014**, *3*, 388–396.
17. Silveira, C.L.; Galvao, A.C.; Robazza, W.S.; Feyh, J.V.T. Modeling and parameters estimation for the solubility calculations of nicotinamide using UNIFAC and COSMO-based models. *Fluid Phase Equilibria* **2021**, *535*, 112970. [[CrossRef](#)]

18. Paul, D.; Sanap, G.; Shenoy, S.; Kalyane, D.; Kalia, K.; Tekade, R.K. Artificial intelligence in drug discovery and development. *Drug Discov. Today* **2021**, *26*, 80. [[CrossRef](#)]
19. Chen, Z.; Liu, X.; Hogan, W.; Shenkman, E.; Bian, J. Applications of artificial intelligence in drug development using real-world data. *Drug Discov. Today* **2021**, *26*, 1256–1264. [[CrossRef](#)]
20. Yang, J.; Du, Q.; Ma, R.; Khan, A. Artificial intelligence simulation of water treatment using a novel bimodal microporous nanocomposite. *J. Mol. Liq.* **2021**, *340*, 117296. [[CrossRef](#)]
21. Zheng, Y.; Wang, X.; Wu, Z. Machine Learning Modeling and Predictive Control of the Batch Crystallization Process. *Ind. Eng. Chem. Res.* **2022**, *61*, 5578–5592. [[CrossRef](#)]
22. Bishop, C.M.; Nasrabadi, N.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4.
23. Rodriguez-Galiano, V.; Sanchez-Castillo, M.; Chica-Olmo, M.; Chica-Rivas, M. Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geol. Rev.* **2015**, *71*, 804–818. [[CrossRef](#)]
24. Wang, H.; Lei, Z.; Zhang, X.; Zhou, B.; Peng, J. Machine learning basics. *Deep Learn.* **2016**, 98–164.
25. Rasmussen, C.E. Gaussian processes in machine learning. In *Summer School on Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 63–71.
26. Shi, J.Q.; Choi, T. *Gaussian Process Regression Analysis for Functional Data*; CRC Press: Boca Raton, FL, USA, 2011.
27. Noriega, L. Multilayer Perceptron Tutorial. Ph.D. Thesis, School of Computing, Staffordshire University, Stoke-on-Trent, UK, 2005.
28. Agirre-Basurko, E.; Ibarra-Berastegi, G.; Madariaga, I. Regression and multilayer perceptron-based models to forecast hourly O<sub>3</sub> and NO<sub>2</sub> levels in the Bilbao area. *Environ. Model. Softw.* **2006**, *21*, 430–446. [[CrossRef](#)]
29. Song, Y.; Liang, J.; Lu, J.; Zhao, X. An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing* **2017**, *251*, 26–34. [[CrossRef](#)]
30. Cover, T. Estimation by the nearest neighbor rule. *IEEE Trans. Inf. Theory* **1968**, *14*, 50–55. [[CrossRef](#)]
31. Khoshmaram, A.; Zabihi, S.; Pelalak, R.; Pishnamazi, M.; Marjani, A.; Shirazian, S. Supercritical process for preparation of nanomedicine: Oxaprozin case study. *Chem. Eng. Technol.* **2021**, *44*, 208–212. [[CrossRef](#)]
32. Rasmussen, C.E.; Nickisch, H. Gaussian processes for machine learning (GPML) toolbox. *J. Mach. Learn. Res.* **2010**, *11*, 3011–3015.
33. Ebden, M. Gaussian processes: A quick introduction. *arXiv* **2015**, arXiv:1505.02965.
34. Yang, Y.; Li, S.; Li, W.; Qu, M. Power load probability density forecasting using Gaussian process quantile regression. *Appl. Energy* **2018**, *213*, 499–509. [[CrossRef](#)]
35. Liu, D.; Pang, J.; Zhou, J.; Peng, Y.; Pecht, M. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. *Microelectron. Reliab.* **2013**, *53*, 832–839. [[CrossRef](#)]
36. Dibike, Y.B.; Solomatine, D.P. River flow forecasting using artificial neural networks. *Phys. Chem. Earth Part B Hydrol. Ocean. Atmos.* **2001**, *26*, 1–7. [[CrossRef](#)]
37. Hagan, M.T.; Demuth, H.B.; Beale, M. *Neural Network Design*; PWS Publishing, Co.: Boston, MA, USA, 1997.
38. Rafiq, M.; Bugmann, G.; Easterbrook, D. Neural network design for engineering applications. *Comput. Struct.* **2001**, *79*, 1541–1552. [[CrossRef](#)]
39. Goh, A.T.; Wong, K.; Broms, B. Estimation of lateral wall movements in braced excavations using neural networks. *Can. Geotech. J.* **1995**, *32*, 1059–1064. [[CrossRef](#)]
40. Mielniczuk, J.; Tyrcha, J. Consistency of multilayer perceptron regression estimators. *Neural Netw.* **1993**, *6*, 1019–1022. [[CrossRef](#)]
41. Zare, M.; Pourghasemi, H.R.; Vafakhah, M.; Pradhan, B. Landslide susceptibility mapping at Vaz Watershed (Iran) using an artificial neural network model: A comparison between multilayer perceptron (MLP) and radial basic function (RBF) algorithms. *Arab. J. Geosci.* **2013**, *6*, 2873–2888. [[CrossRef](#)]
42. Abdelbasset, W.K.; Elkholi, S.M.; Opulencia, M.J.C.; Diana, T.; Su, C.-H.; Alashwal, M.; Zwawi, M.; Algarni, M.; Abdelrahman, A.; Nguyen, H.C. Development of multiple machine-learning computational techniques for optimization of heterogenous catalytic biodiesel production from waste vegetable oil. *Arab. J. Chem.* **2022**, *15*, 103843. [[CrossRef](#)]
43. Ramchoun, H.; Ghanou, Y.; Ettaouil, M.; Janati Idrissi, M.A. Multilayer perceptron: Architecture optimization and training. *Int. J. Interact. Multimed. Artif. Intell.* **2016**, *4*, 26–30. [[CrossRef](#)]
44. Zhou, X.; Zhang, Y.; Mao, T.; Ruan, Y.; Gao, H.; Zhou, H. Feature extraction and physical interpretation of melt pressure during injection molding process. *J. Mater. Process. Technol.* **2018**, *261*, 50–60. [[CrossRef](#)]
45. Yang, J.; Zeng, X.-Q.; Ng, W.W.; Yeung, D.S. Computation of two-layer perceptron networks' sensitivity to input perturbation. In Proceedings of the 2008 International Conference on Machine Learning and Cybernetics, Kunming, China, 12–15 July 2008; pp. 762–767.
46. Devroye, L.; Györfi, L.; Krzyżak, A.; Lugosi, G. On the strong universal consistency of nearest neighbor regression function estimates. *Ann. Stat.* **1994**, *22*, 1371–1385. [[CrossRef](#)]
47. Knez, Z.; Skerget, M.; Sencar-Bozic, P.; Rizner, A. Solubility of nifedipine and nitrendipine in supercritical CO<sub>2</sub>. *J. Chem. Eng. Data* **1995**, *40*, 216–220. [[CrossRef](#)]
48. Méndez-Santiago, J.; Teja, A.S. The solubility of solids in supercritical fluids. *Fluid Phase Equilibria* **1999**, *158*, 501–510. [[CrossRef](#)]
49. Medina, I.; Bueno, J.L. Solubilities of 2-nitroanisole and 3-phenyl-1-propanol in supercritical carbon dioxide. *J. Chem. Eng. Data* **2000**, *45*, 298–300. [[CrossRef](#)]