International Scholarly Research Network
ISRN Chemical Engineering
Volume 2012, Article ID 572421, 7 pages
doi:10.5402/2012/572421

*Research Article*

# Application of Artificial Neural Network in Simulation of Supercritical Extraction of Valerenic Acid from *Valeriana officinalis* L.

**Amir Rabiee Kenaree and Shohreh Fatemi**

*School of Chemical Engineering, College of Engineering, University of Tehran, P.O. Box 11365–4563, Tehran 14174, Iran*

Correspondence should be addressed to Shohreh Fatemi, shfatemi@ut.ac.ir

Received 12 September 2012; Accepted 24 October 2012

Academic Editors: L. Jiang, S. Kaneco, and A. Yu



Application of artificial neural network (ANN) has been studied for simulation of the extraction process by supercritical $CO_2$. Supercritical extraction of valerenic acid from *Valeriana officianalis* L. has been studied and simulated according to the significant operational parameters such as pressure, temperature, and dynamic extraction time. ANN, using multilayer perceptron (MLP) model, is employed to predict the amount of extracted VA versus the studied variables. Three tests, validation, and training data sets in three various scenarios are selected to predict the amount of extracted VA at dynamic time of extraction, working pressure, and temperature values. Levenberg-Marquardt algorithm has been employed to train the MLP network. The model in first scenario has three neurons in one hidden layer, and the models associated with the second and the third scenarios have four neurons in one hidden layer. The determination coefficients are calculated as 0.971, 0.940, and 0.964 for the first, second, and the third scenarios, respectively, demonstrating the effectiveness of the MLP model in simulating this process using any of the scenarios, and accurate prediction of extraction yield has been revealed in different working conditions of pressure, temperature, and dynamic time of extraction.

## 1. Introduction

Valerian essential oils or extracts of valerian root have since long been used as sedatives. Therefore, extensive studies have been performed on the extract of the valerian root in recent years [1, 2]. These studies have revealed antispasm and sedative properties of the valerian [3–5], and attributed medical properties of the valerian mainly to valerenic acid (VA) [5]. Due to these findings, VA is used in formulation of many drugs and cosmetic products. Different methods have been employed for the extraction of the valerian root extract [6, 7], which are divided into two major categories. Hydrodistillation as the first category is inexpensive and easy to implement. However, its main disadvantage lies in operating at the boiling point of water, leading to the loss of many water-sensitive or temperature-sensitive combinations. Supercritical fluid extraction (SFE) in the second category has been recently used for the extraction of the natural materials to a great extent [6–9]. Working at lower temperatures, fast speed of the process, use of clean inexpensive fluid, and easy separation of the products are the main suitable features of SFE. Although the SFE is more costly compared to other methods such as hydrodistillation, its distinctive properties have made it so popular in food and drug industries.

The previously mentioned features of the SFE have drawn attention of scientific and industrial bodies to this technique. Owing to the importance of accurate simulation in design and control of the extraction process, different models have been proposed for simulation and prediction of the supercritical fluid extraction process. Traditional modeling methods, which are based on the balance of mass, energy, and momentum, require the accurate data of the process [10–13]. These are time-consuming analytical methods which encounter considerable error when there is insufficient knowledge about processes. On the other hand, computational intelligence-(CI-) based methods do not require exact description of the process mechanism

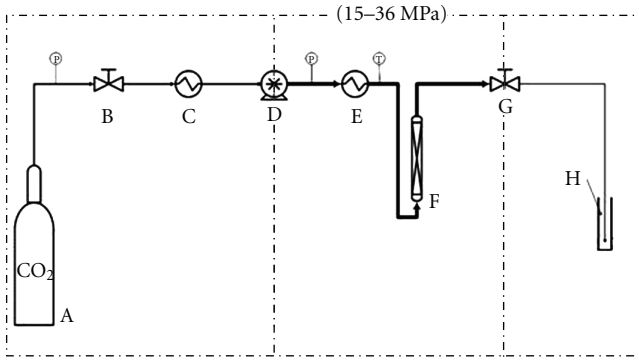


Figure 1: Schematic diagram of SFE extraction. (A) $CO_2$ supply tank; (B) $CO_2$ tank valve; (C) condenser; (D) $CO_2$ pump; (E) heater; (F) extractor; (G) restrictor valve; (H) collecting vial.

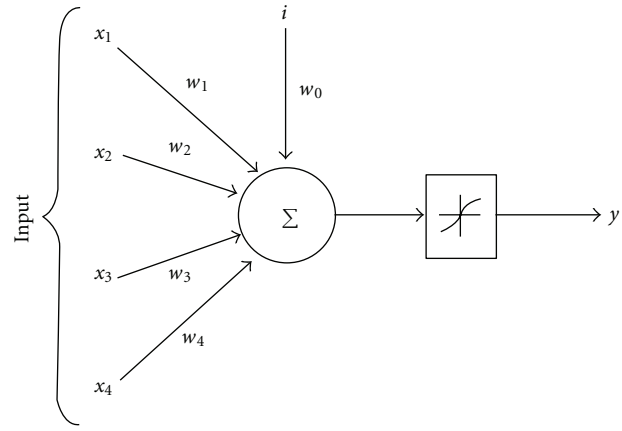


Figure 2: General architecture of a neuron.

[14–16] and can simulate the system's output with reasonable accuracy, only using experimental data. These models are mainly applied when process complexities hinder analytical modeling of the process.

Artificial neural networks (ANNs) as a type of CI-based models were inspired by parallel structure of the neural computations in human brain. The overall structure of the ANN model is determined by an algorithm or by the operator. The network's parameters are tuned by learning algorithms and experimental data in order to minimize the output error. In this paper, three different MLP-based models in three scenarios are proposed for simulation of supercritical extraction of the roots of valerian (*Valeriana officinalis* L.). These models are constructed to perform predictions at the new time, temperature, and pressure (which are not present in training data). Temperature, pressure, solvent flow rate, and particle size are considered as the input variables, while the yield of the extracted valerenic acid is regarded as the output variable. In each scenario, the training, validation, and testing data sets are selected considering the scenario's goal. Here, Levenburg-Marquardt algorithm is used to train the network [17]. Different scenarios are studied to approve the capability and reliability of ANN in simulation of the studied process. The literature survey rarely revealed the studies about application of ANN for simulation of supercritical extraction of essential oil [18, 19].

## 2. Supercritical Fluid Extraction

Compressing a fluid over its critical pressure ($P_C$) or heating a gas over its critical temperature ($T_C$) produces supercritical fluid. In this state, the fluid looks like a compressed gas or a dispersed liquid. Such fluids do not possess the specific properties of the gasses or liquids. Their viscosity and density are close to those of gasses and liquids, respectively. Separation technique which uses this type of fluids is termed supercritical fluid extraction (SFE). The general schematics of SFE process are depicted in Figure 1.

As shown in Figure 1, the extraction fluid such as carbon dioxide is first cooled in condenser and turned into liquid state. The obtained liquid is then pumped into the heater and is heated up to $T_C$. The heated fluid is then pumped to extractor. The extraction vial is filled with dried plant powder, and its temperature is constant. After the extraction, the fluid goes into the collecting vial. Then, its temperature and pressure are reduced in collecting vial. Therefore, the extraction fluid is exhausted, and the extraction yield remains in the collecting vial.

In this study, the experimental data of the extraction process of VA from valerian plant by supercritical fluid of carbon dioxide has been used. In this process, changes in pressure and temperature cause changes in fluid's density and viscosity and finally result in the different extraction efficiencies. Change in particles' size affects mass transfer coefficients, and extraction rate is affected by fluid's flow rate. Generally, increase in pressure, temperature, and solvent flow rate and decrease in particles' size improve the extraction efficiency.

## 3. The Neural Network

Inspired by biological natural systems, ANN is comprised of a series of simple processing units, called neurons, and intended to solve complex problems. Figure 2 shows the general architecture of a neuron.

It is evident in this figure that the input $x_i$ is formed by the weighted summation of the input variables plus a constant value of $(I \times \omega_0)$ referred to as bias. Then, the neuron's output signal, $\hat{y}_i$, is determined by an activation function $\Phi(x)$. The activation function is normally in the form of linear, sigmoid, or tangent hyperbolic functions. The activation functions of all hidden layer neurons are considered to be same [17]. Hence, the mathematical description of the calculation of the $i$th neuron's output with $p$ input variables is stated as follows:

$$x_i = \sum_{j=0}^{p} \underline{\omega_j}\ \underline{u_j},$$
$$\hat{y}_i = \Phi(x_i), \tag{1}$$

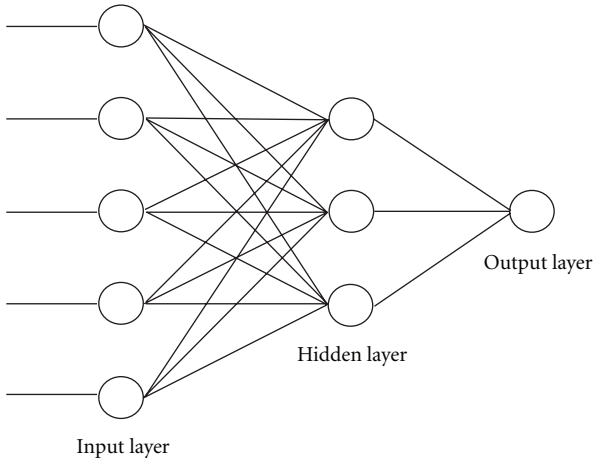


Figure 3: The general structure of an MLP network with 5 inputs, 3 neurons in one hidden layer, and an output layer.

where $\underline{u_j} = [u_0, u_1, u_2, \ldots, u_p]^T$ and $\underline{\omega_j} = [\omega_0, \omega_1, \omega_2, \ldots, \omega_P]$ are input and weighting vectors, respectively. In Figure 2, $w_0$ is the bias constant, considered as unity.

*3.1. Multilayer Perceptron Model.* Multilayer perceptron (MLP), as the most well-known ANN model, consists of an input layer, one or several hidden layers, and an output layer. The MLP's structure is illustrated in Figure 3.

In MLP network, the input units just transfer the incoming data without performing any processing task on them. The input data after passing the input layer reach the hidden layer which can be formed by one or more layers. Having processed in hidden layer neurons, these processed data are transferred to the output layer. A tangent hyperbolic or sigmoid transfer function is normally employed in MLP's hidden layer. In this paper, a tangent hyperbolic activation function has been used.

Consider

$$\Phi(x_i) = \frac{1 - e^{-x_i}}{1 + e^{-x_i}}, \tag{2}$$

where $x_i$ is the weighted summation of the inputs and $\Phi(x_i)$ is the output of the $i$th neuron. The activation function in the output layer is usually considered to be linear, computing the linear combination of its inputs. Now, the total output of the network with one hidden layer, $M$ neurons in hidden layer and one neuron in output layer, is calculated as follows:

$$\hat{y}_{\text{out}} = \sum_{i=1}^{M} \omega_i \hat{y}_i + (\omega_0 \times 1), \tag{3}$$

where $(\omega_0 \times 1)$ is associated with the output neuron's bias.

Obviously, all possible structures must be tried to find the best performing network. Therefore, the different numbers of the network layers and neurons in each layer and different activation functions must be considered. The number of neurons in input layer and output layer are same as the number of the input and output variables, respectively. Number of layers and hidden layer neurons is determined depending on the complexity of the problem and desired accuracy. Hence, the weighting coefficients and bias constants are the only unknown parameters of the network. The procedure of tuning these unknown parameters in order to minimize the output error is termed training. Neural network training is normally carried out in a supervised manner. In this training method, input data are passed through the network, and the output is calculated. Next, the difference between the network output and the real values, regarded as network error, is computed. Then, the training algorithm improves the weighting coefficients and bias constants such that the obtained error is reduced. This process continues up to achieving the desired error. In this paper, Levenburg-Marquardt training algorithm is employed to train the MLP network [20].

*3.2. Levenburg-Marquardt Algorithm.* Newton optimization methods have been established based on the second-order Taylor expansion around the old weight vector. Direct Newton method encounters difficulty in computation of the Hessian matrix. Therefore, scaling factor methods such as Levenbrug-Marquardt algorithm have been proposed. Although the Levenbrug-Marquardt algorithm and Newton method both have been set up on the Hessian matrix, the former has been designed to approach second-order training speed without having to compute the Hessian matrix. Consider the function $V(w)$. This function should be minimized with respect to the vector of networks parameters $(w)$. Vector $w$ is updated as follows:

$$\begin{aligned} \Delta w &= -\left[\nabla^2 V(w)\right]^{-1} \cdot V(w), \\ w(k+1) &= w(k) + \Delta w, \end{aligned} \tag{4}$$

where $\nabla^2 V(w)$ is Hessian matrix and $\nabla V(w)$ represents $V(w)$ gradient. If $V(w)$ is regarded as squared sum of error, then we have

$$V(W) = 0.5 \sum_{r=1}^{N} e_r^2(w). \tag{5}$$

Now, the Hessian matrix $\nabla^2 V(w)$ and the gradient $\nabla V(w)$ are defined using Jacobean matrix as follows:

$$\nabla V(w) = J^T(w) \cdot e(w), \tag{6}$$

$$\nabla^2 V(w) = J^T(w) \cdot J(w) + \sum_{r=1}^{N} e_r(w) \cdot \nabla^2 e_r(w), \tag{7}$$

where

$$J(w) = \begin{bmatrix} \dfrac{\partial e_1(w)}{\partial w_1} & \dfrac{\partial e_1(w)}{\partial w_2} & \cdots & \dfrac{\partial e_1(w)}{\partial w_{N_P}} & \\ \dfrac{\partial e_2(w)}{\partial w_1} & \dfrac{\partial e_2(w)}{\partial w_2} & \cdots & \dfrac{\partial e_2(w)}{\partial w_{N_P}} & \cdots \\ \vdots & \vdots & \ddots & \vdots & \\ \dfrac{\partial e_N(w)}{\partial w_1} & \dfrac{\partial e_N(w)}{\partial w_2} & \cdots & \dfrac{\partial e_N(w)}{\partial w_{N_P}} & \end{bmatrix}_{(N \times N_P)}, \tag{8}$$

and $\underline{w} = [w_1, w_2, \ldots, w_{N_P}]$ is the vector of network parameters.

It is obvious from (8) that the size of Jacobean matrix is $N \times N_p$, where $N$ and $N_P$ are the number of algorithm executions and network parameters, respectively. In Gauss-Newton method, the second term in (7) is assumed to be zero. Hence, the $w$ is updated based on the following relation:

$$\Delta w = \left[J^T(w) \cdot J(w)\right]^{-1} \cdot J^T(w) \cdot e(w). \tag{9}$$

The difference between Levenburg-Marquardt algorithm and Gauss-Newton method can be stated as follows:

$$\Delta w = -\left[J^T(w) \cdot J(w) + \mu \cdot I\right]^{-1} \cdot J^T(w) \cdot e(w), \tag{10}$$

where $I$ represents an $N_p \times N_p$ identity matrix. In each iteration when $V(w)$ decreases, $\mu$ is multiplied by $\mu_{\text{inc}}$. However, in case of decrease in $V(w)$, $\mu$ is divided by $\mu_{\text{dec}}$. Therefore, $w$ is stated as follows:

$$w(k+1) = w(k) - \left[J^T(w) \cdot J(w) + \mu \cdot I\right]^{-1} \cdot J^T(w) \cdot e(w). \tag{11}$$

Levenburg-Marquardt algorithm as an appropriate optimization tool for the small- and medium-sized networks (with less than hundred parameters) has been employed in this paper.

*3.3. Structure of the Multilayer Perceptron Model.* Simulation process by MLP model includes several steps. Experimental data are divided into three sets in first step. The largest set, termed training set, is comprised of 70 percent of the whole data. Validation set, as the second set, consists of 20 percent of the whole data. The last 10 percent of the experimental data are allocated to test set. Training set is used for tuning the network parameters. To validate our model, the validation set is employed, and the test set is used to perform prediction.

Different structures, including different number of layers, neurons, as well as different activation functions, must be examined in order to find the best performing network. The type of activation functions must be determined in next step. We have used tangent hyperbolic and linear activation functions in hidden layer and output layer, respectively. The network structure is decided on in the third step. Number of input layer neurons equals the number of input variables. On the other hand, number of output layer neuron is same as the number of output variables. Hence, only the number of hidden layer and their neurons must be determined. To do this, first, a network with one neuron in its hidden layer is constructed. Then, the other neurons are added one by one until the predetermined desired value of error is reached. Now, the network's parameters are tuned by training algorithm. For the networks with one hidden layer, total number of weighting coefficients and biases is computed based on the following equation:

$$N = M(P+1) + Q(M+1), \tag{12}$$

where $M$, $P$, and $Q$ are the number of hidden layer neurons, inputs, and outputs, respectively. As the size of the network increases, the number of network parameters also increases. Therefore, there is an optimal value for the network's size. The closer the number of parameters to the number of data, the less general network will be produced. Therefore, the number of network parameters should not exceed half of the number of training data.

Theoretically speaking, training error in ANNs can approach to zero. However, this situation is not desirable at all because it reduces network's flexibility. It means that the knowledge acquired through the training process is not applicable to other data, resulting in a large value of error. This is called overtraining. To prevent this undesirable circumstance, validation error is computed after each execution. When this error begins to rise, the training process is terminated.

## 4. Application of MLP to Simulation of Valerenic Acid Extraction Process

*4.1. Results.* Eighty experimental data, collected through 10 executions of the extraction process, are used to investigate the efficiency of MLP in simulating the process of VA extraction. Pressure, temperature, particle size, and solvent flow rate of supercritical fluid were kept constant during each execution. The output data were measured at 1.5, 5, 9.5, 15, 21.5, 29, 37.5, and 47 min. Specifications relating to each execution are tabulated in Table 1.

Regarding the experimental data, it is expected that predicting the extraction yield at new time, new pressure, or new temperature (which were not present in training process) exhibits small error. On the other hand, prediction at new particle size or new solvent flow rate brings about more error. Therefore, three different testing scenarios were designed to forecast the amount of extraction yield at new time, pressure, and temperature. In the first scenario, the amount of extracted VA at 15 minutes of each execution was picked out as testing set. Validation data were selected randomly from other data. The remaining data were allocated to training set. In the second scenario, the fourth execution was chosen for testing purposes. Validation data were selected randomly from other data. The remaining data were assigned to training set. The fifth execution was chosen as the testing set in the third scenario. Like other scenarios, validation data were selected randomly from other data, and the remaining data were allocated to training set. Obviously, the first scenario was focused on prediction at the new time, while the second and third scenarios were chosen to predict the yield at new pressure and temperature, respectively.

Then, different network structures for each scenario were trained by Levenburg-Marquardt algorithm [20]. The final MLP networks associated with each scenario were evaluated by $R^2$ error criterion, defined as follows:

$$R^2 = 1 - \frac{\sum_i^n \left(y_i^{\text{exp}} - y_i\right)^2}{\sum_i^n \left(y_i^{\text{exp}} - y_{\text{mean}}^{\text{exp}}\right)^2}, \tag{13}$$

Table 1: Operational conditions of the experiments.

| Number of execution | Pressure (MPa) | Temperature (K) | Flow rate of supercritical fluid (mL/min) | Particle size (mm) |
|---|---|---|---|---|
| 1 | 15 | 310 | 1.1 | 0.295 |
| 2 | 22 | 310 | 1.1 | 0.295 |
| 3 | 29 | 310 | 1.1 | 0.295 |
| 4 | 36 | 310 | 1.1 | 0.295 |
| 5 | 29 | 318 | 1.1 | 0.295 |
| 6 | 29 | 326 | 1.1 | 0.295 |
| 7 | 29 | 334 | 1.1 | 0.295 |
| 8 | 22 | 310 | 0.5 | 0.295 |
| 9 | 22 | 310 | 1.1 | 0.09 |
| 10 | 22 | 310 | 1.1 | 1.00 |

Table 2: The results of the constructed networks with their corresponding error.

| Scenario | Number of neurons in first layer | $R^2$ (training set) | $R^2$ (validation set) | $R^2$ (test set) |
|---|---|---|---|---|
| 1 | 3 | 0.987 | 0.981 | 0.971 |
| 2 | 4 | 0.988 | 0.970 | 0.940 |
| 3 | 4 | 0.977 | 0.972 | 0.964 |

where $n$ represents number of data. $y_i^{\text{exp}}$, $y_{\text{mean}}^{\text{exp}}$, and $y_i$ are experimental values, mean experimental values, and calculated values, respectively. The obtained results are summarized in Table 2.

*4.2. First Scenario.* In this scenario, obtained data at 15 minutes of the each execution were chosen as test set. Validation data were selected randomly from other data. The remaining data were allocated to training set. Training, validation, and testing data sets consist of fifty, twenty, and ten members, respectively. The best performing MLP structure for this scenario had three neurons in one hidden layer. The constructed network is comprised of 22 parameters, which is reasonable regarding the number of training data. The value of $R^2$ criterion for this network was 0.971. Figure 3 presents an illustrative comparison between obtained and real values of extracted VA associated with the first scenario.

*4.3. Second Scenario.* The fourth execution was selected for the second scenario. Validation data were selected randomly from other data. The remaining data were allocated to training set. Training, validation, and testing sets consist of 54, 18, and 8 members, respectively. This scenario was intended to predict the amount of the extraction yield at new pressure. The best MLP structure in this scenario has 4 neurons in one hidden layer and 29 parameters. The obtained value of $R^2$ in this scenario was 0.94. Figure 5 and Table 3 present a detailed comparison between predicted and real values.

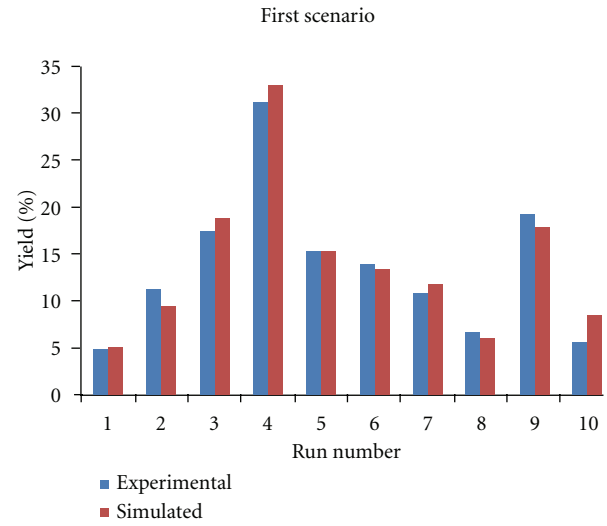


Figure 4: Comparison between real and predicted yield (%) after 15 min extraction in first scenario.

*4.4. Third Scenario.* In our last scenario, the fifth execution data were used. Validation data were selected randomly from other data. The remaining data were assigned to the training set. In this scenario, training, validation, and testing sets consist of 54, 18, and 8 members, respectively. This scenario was designed to predict the amount of extraction yield at new temperature. The optimal network for the third scenario was constructed with 4 neurons in one hidden layer and 29 parameters. The $R^2$ value for this scenario was 0.964. The comparison between real and predicted values has been presented in Figure 6 and Table 4.

The performed comparisons in Figures 3, 4, and 5 and Tables 2–4 show that the network's outputs are quite accurate. Thus, the MLP model can be applied to predict extraction yield at new time, pressure, and temperature.

## 5. Discussion

As stated in previous section, the best MLP models for the first, second, and third scenarios had three, four, and four

Table 3: Numerical comparison between real and predicted yield values in second scenario.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Real value | 14.03 | 21.32 | 27.65 | 31.14 | 37.75 | 42.44 | 45.65 | 46.36 |
| Prediction | 19.06 | 22.61 | 26.74 | 31.13 | 35.67 | 40.43 | 45.54 | 51.08 |

Table 4: Numerical comparison between real and predicted yield values in third scenario.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Real value | 6.01 | 9.39 | 13.35 | 15.14 | 16.74 | 19.33 | 20.42 | 21.14 |
| Prediction | 8.32 | 10.12 | 12.49 | 15.10 | 17.44 | 19.16 | 20.37 | 21.47 |

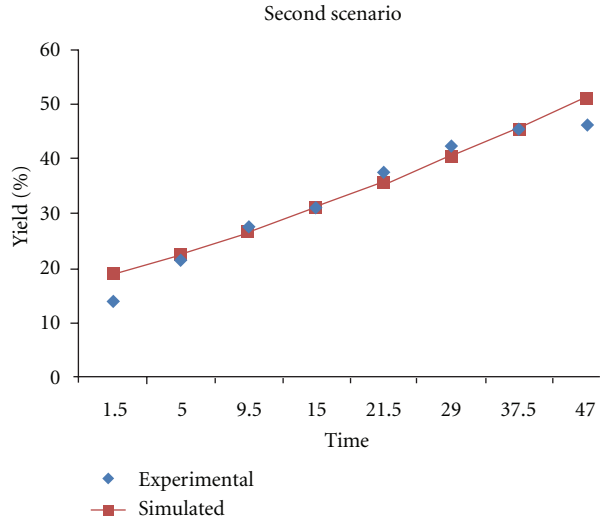


Figure 5: Comparison between real and predicted yield (%) in second scenario.

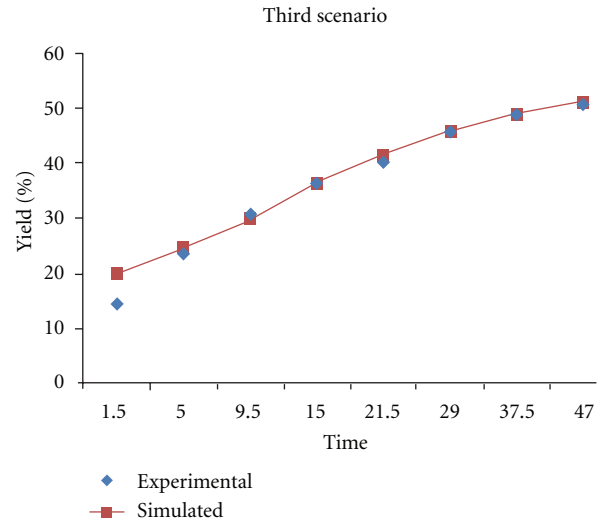


Figure 6: Comparison between real and predicted yield (%) in third scenario.

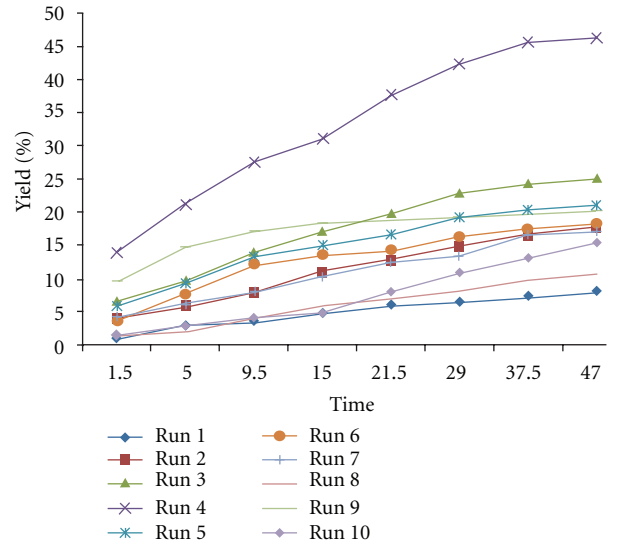


Figure 7: The experimental results (yield %), from the initial time to the end for all executions.

neurons in one hidden layer, respectively. In first scenario, several data points from each execution were present in training set, leading to a more simple structure. However, the whole data of the fourth execution were chosen as the testing set for the second scenario. Similarly, the fifth execution was assigned to the third scenario testing data. Hence, the prediction in second and third scenarios was more difficult, resulting in more complex structures. On the other hand, the first and third scenarios can be interpreted as interpolations. Quite contrary, the second scenario was an extrapolation. Referring to Table 1, it can be seen that the time variable takes the values of 1.5, 5, 9.5, 15, 21.5, 37.5, and 47 minutes The temperature variable takes the values of 130, 318, 326, and 334 K, and the pressure takes the values of 15, 22, 29, and 36 MPa. The data points related to 15 minutes of all executions were selected as the testing set in the first scenario. The whole data of the fourth execution, that is, at temperature of 334 K, were assigned to the second scenario testing set. In the third scenario, the whole data of the fifth execution, that is, at pressure of 29 MPa, were chosen as the testing data. Obviously, the extrapolation error is higher than that of interpolation. Therefore, the $R^2$ value for the second scenario is less than the first and third scenarios.

The error of MLP model can be attributed to three factors. The experimental error is the first factor. As shown in Figure 7, the experimental data curves do not have a smooth slope, and there are some breaking points. These breaking points are probably due to the measurement error in extraction yield. Network estimation is the second component of error. The predicted curves by the MLP network have a smooth slope, crossing through the middle

of the experimental data points. This leads to a difference between network's output and the experimental data. If the training error is minimized, the outputs of the network will fit the breaking points present in experimental data curves. Therefore, the measurement errors are introduced into network structure, resulting in inaccurate prediction of the testing data. Some other experimental factors that have been ignored are the third source of error.

## 6. Conclusion

Simulation of the complex processes such as supercritical extraction of valuable components from pharmaceutical plants requires the stochastic methods. In this work, application of multilayer perceptron model in ANN has been conducted for simulation of VA extraction by $CO_2$ supercritical fluid. To approve the reliability and capability of ANN, three testing sets in three different scenarios were selected. In order to find the optimal structure for each scenario, Levenburg-Marquardt training algorithm was employed. The $R^2$ values for these three scenarios were 0.971, 0.940, and 0.964, respectively. Small error values in all scenarios demonstrated the effectiveness of the MLP model in simulation of SFE process. It was concluded that MLP model can be well applied to simulate extraction process and optimize the operational conditions in the complex processes where there are no theoretical models.

## References


[1] X. Q. Gao and L. Björk, "Valerenic acid derivatives and valepotriates among individuals, varieties and species of Valeriana," *Fitoterapia*, vol. 71, no. 1, pp. 19–24, 2000.

[2] I. Zizovic, M. Stamenić, J. Ivanović et al., "Supercritical carbon dioxide extraction of sesquiterpenes from valerian root," *Journal of Supercritical Fluids*, vol. 43, no. 2, pp. 249–258, 2007.

[3] S. Khom, I. Baburin, E. Timin et al., "Valerenic acid potentiates and inhibits GABAA receptors: molecular mechanism and subunit specificity," *Neuropharmacology*, vol. 53, no. 1, pp. 178–187, 2007.

[4] B. M. Dietz, G. B. Mahady, G. F. Pauli, and N. R. Farnsworth, "Valerian extract and valerenic acid are partial agonists of the 5-$HT_{5a}$ receptor in vitro," *Molecular Brain Research*, vol. 138, no. 2, pp. 191–197, 2005.

[5] D. Benke, A. Barberis, S. Kopp et al., "GABAA receptors as in vivo substrate for the anxiolytic action of valerenic acid, a major constituent of valerian root extracts," *Neuropharmacology*, vol. 56, no. 1, pp. 174–181, 2009.

[6] A. Safaralie, S. Fatemi, and F. Sefidkon, "Essential oil composition of *Valeriana officinalis* L. roots cultivated in Iran. Comparative analysis between supercritical $CO_2$ extraction and hydrodistillation," *Journal of Chromatography A*, vol. 1180, no. 1-2, pp. 159–164, 2008.

[7] Z. Hromádková, A. Ebringerová, and P. Valachovič, "Ultrasound-assisted extraction of water-soluble polysaccharides from the roots of valerian (*Valeriana officinalis* L.)," *Ultrasonics Sonochemistry*, vol. 9, no. 1, pp. 37–44, 2002.

[8] L. Zhiyi, L. Xuewu, C. Shuhua et al., "An experimental and simulating study of supercritical $CO_2$ extraction for pepper oil," *Chemical Engineering and Processing: Process Intensification*, vol. 45, no. 4, pp. 264–267, 2006.

[9] L. Vázquez, A. M. Hurtado-Benavides, G. Reglero, T. Fornari, E. Ibáñez, and F. J. Señoráns, "Deacidification of olive oil by countercurrent supercritical carbon dioxide extraction: experimental and thermodynamic modeling," *Journal of Food Engineering*, vol. 90, no. 4, pp. 463–470, 2009.

[10] J. M. Del Valle, J. C. De La Fuente, and D. A. Cardarelli, "Contributions to supercritical extraction of vegetable substrates in Latin America," *Journal of Food Engineering*, vol. 67, no. 1-2, pp. 35–57, 2005.

[11] S. Espinosa, M. S. Diaz, and E. A. Brignole, "Food additives obtained by supercritical extraction from natural sources," *Journal of Supercritical Fluids*, vol. 45, no. 2, pp. 213–219, 2008.

[12] W. Wu and Y. Hou, "Mathematical modeling of extraction of egg yolk oil with supercritical $CO_2$," *Journal of Supercritical Fluids*, vol. 19, no. 2, pp. 149–159, 2001.

[13] E. M. C. Reis-Vasco, J. A. P. Coelho, A. M. F. Palavra, C. Marrone, and E. Reverchon, "Mathematical modelling and simulation of pennyroyal essential oil supercritical extraction," *Chemical Engineering Science*, vol. 55, no. 15, pp. 2917–2922, 2000.

[14] M. J. Kamali and M. Mousavi, "Analytic, neural network, and hybrid modeling of supercritical extraction of $\alpha$-pinene," *Journal of Supercritical Fluids*, vol. 47, no. 2, pp. 168–173, 2008.

[15] M. Izadifar and F. Abdolahi, "Comparison between neural network and mathematical modeling of supercritical $CO_2$ extraction of black pepper essential oil," *Journal of Supercritical Fluids*, vol. 38, no. 1, pp. 37–43, 2006.

[16] M. Fullana, F. Trabelsi, and F. Recasens, "Use of neural net computing for statistical and kinetic modelling and simulation of supercritical fluid extractors," *Chemical Engineering Science*, vol. 55, no. 1, pp. 79–95, 2000.

[17] O. Nelles, *Nonlinear System Identification from Classical Approaches to Neural Networks and Fuzzy Models*, Springer, 2001.

[18] A. N. Mustapa, Z. Abdul Manan, and C. Y. M. Azizi, "Subcritical and supercritical fluid extraction: a critical review of its analytical usefulness," *Journal of Chemical and Natural Resources Engineering*, vol. 2, pp. 164–180, 2008.

[19] A. Moghadassi, S. M. Hosseini, F. Parvizian, I. Al-Hajri, and M. Talebbeigi, "Predicting the supercritical carbon dioxide extraction of oregano bract essential oil," *Songklanakarin Journal of Science and Technology*, vol. 33, no. 5, pp. 531–538, 2011.

[20] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.