

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/266203431>

# Estimating hydrogen sulfide solubility in ionic liquids using a machine learning approach

ARTICLE *in* JOURNAL OF SUPERCRITICAL FLUIDS THE · AUGUST 2014

Impact Factor: 2.37 · DOI: 10.1016/j.supflu.2014.08.011

---

CITATIONS

6

READS

75

---

## 6 AUTHORS, INCLUDING:



**Ali Shafiei**

University of Waterloo

**40** PUBLICATIONS **144** CITATIONS

[SEE PROFILE](#)



**Mohammad Ali Ahmadi**

National Iranian Oil Company (NIOC)

**113** PUBLICATIONS **919** CITATIONS

[SEE PROFILE](#)



**Alireza Baghban**

Petroleum University of Technology

**6** PUBLICATIONS **11** CITATIONS

[SEE PROFILE](#)



**Reza Soleimani**

Petroleum University of Technology

**11** PUBLICATIONS **88** CITATIONS

[SEE PROFILE](#)



# Estimating hydrogen sulfide solubility in ionic liquids using a machine learning approach



Ali Shafiei<sup>a, \*\*</sup>, Mohammad Ali Ahmadi<sup>b,\*</sup>, Seyed Hayan Zaheri<sup>b</sup>, Alireza Baghban<sup>c</sup>, Ali Amirkakhrian<sup>d</sup>, Reza Soleimani<sup>e</sup>

<sup>a</sup> Earth & Environmental Sciences, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1, Canada

<sup>b</sup> Department of Petroleum Engineering, Ahwaz Faculty of Petroleum Engineering, Petroleum University of Technology (PUT), PO Box 63431, Ahwaz, Iran

<sup>c</sup> Department of Chemical Engineering, University of Tehran, Tehran, Iran

<sup>d</sup> School of Chemical and Petroleum Engineering, Shiraz University, 71345 Mollasdra Avenue, Shiraz, Iran

<sup>e</sup> Young Researchers and Elite Club, Neyshabur Branch, Islamic Azad University, Neyshabur, Iran

## ARTICLE INFO

### Article history:

Received 13 June 2014

Received in revised form 31 July 2014

Accepted 1 August 2014

Available online 21 August 2014

### Keywords:

Ionic liquids

Hydrogen sulfide

Solubility

Prediction

Particle swarm optimization

Artificial neural network

## ABSTRACT

For the design and development of new processes of gas sweetening using ionic liquids (ILs), as promising candidates for amine solutions, an amazing model to predict the solubility of acid gases is of great importance. In this direction, in the current study, the capability of artificial neural networks (ANNs) trained with back propagation (BP) and particle swarm optimization (PSO), to correlate the solubility of H<sub>2</sub>S in 11 different ILs have been investigated. Different structures of three-layer feed forward neural network using acentric factor ( $\omega$ ), critical temperature ( $T_c$ ), critical pressure ( $P_c$ ) of ILs accompanied by pressure ( $P$ ) and temperature ( $T$ ), as input parameters, were examined and an optimized architecture has been proposed as 5–9–1. Implementation of these models for 465 experimental data points collected from the literature shows coefficient of determination ( $R^2$ ) of 0.99218 and mean squared error (MSE) of 0.00025 from experimental values for PSO-ANN predicted solubilities while the values of  $R^2 = 0.95151$  and MSE = 0.00335 were obtained for BP-ANN model. Therefore, through PSO training algorithm we are able to attain significantly better results than with BP training procedure based on the statistical criteria.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

The removal of such these acid gases, i.e. CO<sub>2</sub> and H<sub>2</sub>S, from natural gas is of great importance for operational, economical, and environmental as well as health and well-being reasons. Among various treatment methods have been reported for the removal of impurities (acid gases) and purify natural gases, the gas-liquid absorption in amine based solvents, is one of the most commonly used processes for the removal of acid components in industrial natural gas treatment and sweetening plants [1–10].

Recently, ionic liquids (ILs), have gained considerable attention as a new class of non-aqueous solvents [11], with high potential

to be used as novel media for gas sweetening processes [12] as published data show that ILs are proficient in absorbing H<sub>2</sub>S from natural gas, as well as selectively separating H<sub>2</sub>S from carbon dioxide (CO<sub>2</sub>) [7]. In addition to the selective absorption of undesirable substances (i.e. of CO<sub>2</sub> and H<sub>2</sub>S) from the natural gas, a variety of industrial applications of ionic liquids have been proposed such as ILs in nuclear fuel reprocessing [13], in cellulose processing [14] and etc.

ILs are melted salts which are comprised of cationic and anionic species. Unlike conventional salts such as sodium chloride, ILs have a little affinity to form crystals owing to their asymmetrical and bulky cation assembly [7]. These salts are even liquid at or near room temperatures, and for that reason, are also called room temperature ionic liquids (RTILs) [15]. One of the greatest remarkable features of ILs for gas sweetening is their negligibly small vapor pressure which allows them to be regenerated and recycled to absorbing section with no significant losses into the gas stream results in negligible solvent make-up requirements [11]. Aside from this appealing characteristic, i.e. very low vapor pressure, non-flammability, lack of odor, high thermal, chemical and

\* Corresponding author at: Department of Petroleum Engineering, Ahwaz Faculty of Petroleum Engineering, Petroleum University of Technology (PUT), PO Box 63431, Ahwaz, Iran.

\*\* Corresponding author at: Earth & Environmental Sciences, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1.

E-mail address: [ahmadi6776@yahoo.com](mailto:ahmadi6776@yahoo.com) (M.A. Ahmadi).

electrochemical stability, their ability to be tailored by manipulating of cations and anions for a specific task are another advantages of ILs.

Understanding regarding the acid gases' solubility in ILs at various pressures and temperatures for prospective implementation in industrialized processes of natural gas sweetening in order to discover an intriguing IL for use as a gas separating agent and also for the scheme and processes of gas sweetening on the basis of ionic liquids, is a matter of supreme importance [9]. While some experimental data on the phase behavior of systems containing  $\text{CO}_2 + \text{ILs}$  and/or  $\text{H}_2\text{S} + \text{ILs}$  are accessible in literature (note that dissimilar to  $\text{CO}_2 + \text{ILs}$  mixtures,  $\text{H}_2\text{S} + \text{IL}$  systems were not investigated extensively and reported actual data for  $\text{H}_2\text{S}$  solubility in ILs are limited and bear no resemblance to the experimental data related to  $\text{CO}_2$  solubility in ILs. See, for instance, Soriano et al. [16], who presented a relatively good review about works on solubility of  $\text{CO}_2$  in different ILs and references [4,17–20] for solubility of  $\text{H}_2\text{S}$  in various ILs), even more data are required for reliable process design. Owing to the complications of experimental works, time-consumption experiments, expensive and in some cases dangerous, it would be of great value to evolve robust models for predicting accurately the phase behavior of these kinds of systems over a wide range of operational conditions to facilitate the process design [15,21–24]. Additionally, to competently design the processes handling mixtures of acid gases in ionic liquids, knowing about the working settings to attain required acid gases solubility in the practical ionic liquid solvent is vitally needed [21].

In the past few years, several studies and attempts have been made to model the solubility of gas solubility in various ILs. A comprehensive recent review has been presented by Vega et al. [25] about several works on the modeling of gas solubility in ILs especially  $\text{CO}_2$ -IL systems using various modeling approaches. For  $\text{H}_2\text{S}$ , however, modeling studies like experimental investigations for solubility in ILs are not as much as those for  $\text{CO}_2$ .

However, all of the previously applied approaches such as equation of states (EOSs), suffer from disadvantages. The main shortcoming shared by almost all above-mentioned models is that they can be trusted exclusively for a particular system and not for a wide assortment of systems as we are interested in. As reported in the literature [26,27], group contribution methods, activity models, extended Henry's law, equations of state, straightforward correlations and other recommended models, which are applied to model the solubility of gases, have need of regulating variables which have to adjusted on the basis of actual recorded data samples and without adjustable variables and experimental data samples, most of aforementioned approaches are not reliable. Besides, the complexity and time-consuming accurate methods such as, SAFT models, modified Henry's law, etc. are other defaults [28].

Computational Intelligence paradigms are advanced procedures, accompanied by supreme flexibility. Artificial Neural Network (ANN), is an important embranchment of computational intelligence paradigms, can be a suitable alternative for gas solubility and phase equilibrium modeling. ANN, due to enjoy significant advantageous such as its high uniformity of analysis, parallelism, adaptivity, nonlinearity, fault tolerance and design and capability to challenge fuzzy and imprecise data [29], has aroused the interest of the scientific community and engineers. It eliminates the complex equations and provides a very simple way of dealing with a problem using mathematical works. They have been extensively employed in various areas of science and technology, such as research related to solubility modeling and phase equilibria [30].

In order to establish optimum configuration of neural networks and related parameters such as weights and biases, several researchers applied different optimization methods such as back propagation (BP) [31], genetic algorithm [32], particle swarm optimization (PSO) [33–35], hybrid particle swarm optimization

and genetic algorithm (HPSOGA) [36,37], unified particle swarm optimization (UPSO) [38], imperialist competitive algorithm (ICA) [39,40], and pruning algorithm (PA) [41] have been employed.

In this study, we got to grips with the application of ANN trained with PSO predicting the solubility of  $\text{H}_2\text{S}$  in 1-hexyl-3-methylimidazolium hexa-fluoro-phosphate ([hmim][PF6]), 1-hexyl-3-methyl-imidazolium-tetra-fluoro-borate ([hmim][BF4]), 1-ethyl-3-methylimidazolium ethyl-sulfate ([emim][EtSO4]), 1-octyl-3-methylimidazolium bis (tri-fluoro-methyl-sulfonyl) imide ([C8mim][Tf2N]), 1-hexyl-3-methylimidazolium bis (tri-fluoro-methyl-sulfonyl) imide ([C6mim][Tf2N]), 1-octyl-3-methylimidazolium hexa-fluoro-phosphate ([C8mim][PF6]), 1-butyl-3-methylimidazolium hexa-fluoro-phosphate ([bmim][PF6]), 1-butyl-3-methylimidazolium tetra-fluoro-borate ([bmim][BF4]), 1-butyl-3-methylimidazolium bis(tri-fluoro-methyl-sulfonyl)imide ([bmim][Tf2N]), 1-ethyl-3-methylimidazolium hexa-fluoro-phosphate ([emim][PF6]) and 1-ethyl-3-methylimidazolium bis(tri-fluoro-methyl-sulfonyl)imide ([emim][Tf2N]) at different temperatures and pressures. To the best of our knowledge, thus far, there is neither publication on the prediction of  $\text{H}_2\text{S}$  solubility in ILs using ANN nor using any other computational intelligence paradigms.

## 2. Theory

### 2.1. Artificial neural network

According to Haykin [29], a neural network (NN) is an immensely counterpart allocated processor fabricated of straightforward processing units, which has an accepted tendency for storage experimental knowledge and being it accessible for application.

Whereas several types of ANNs exist, the most generic kind of ANNs is the multi-layer feed-forward NN (MLFFNN), which comprises a group of interrelated neurons organized in different layers. The layers include: output layer, hidden layer, and input layer, each of which contains an assembly of neurons. Each of these neurons share same output and input associations, nevertheless, which doesn't communicate with the further neurons creating the same layer. The connections between the neurons are rigorously turned in a certain approach: from the input towards the output with no feed-backing among the neurons. The neurons throughout the output and hidden layers measure the net inputs by execution a biased sum of the targets receiving from the preceding layer. On the other hand, the outputs are determined by utilizing a transfer function on the net inputs. So, the neuron's weighted input is translated to output signals by applying transfer function. The three transfer functions which are shortly described as follows are the most commonly used transfer functions:

- Linear transfer function (purelin),

$$\varphi(n) = n \quad (1)$$

is often employed in the output layer. Outputs in the span of  $+\infty$  to  $-\infty$  can be obtained from this transfer function. The main interest of the ML-FFNN model, belong to its nonlinear sigmoid functions such as tansig and logsig, which are mostly utilized in its hidden layer.

- Log-Sigmoid transfer function (logsig)

$$\varphi(n) = \frac{1}{1 + e^{-n}} \quad (2)$$

is an effortlessly differentiable and commonly nonlinear transfer function, which is employed for various engineering

implementations. The logsig function creates outputs between 1 and 0 as the net input of neuron refers to positive and negative infinity, correspondingly.

- Hyperbolic Tangent Sigmoid (tansig)

$$\varphi(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (3)$$

is a S-shaped transfer function that generates output in the range of  $-1$  to  $+1$ . It is also differentiable and frequently used in engineering applications. In all of above-mentioned equations "n" represent the net input signal and " $\varphi$ " stands for activation function.

The architecture of the three-layer feed-forward ANN illustrated in Fig. 1 has input layer (layer 1), hidden layer (layer 2) and output layer (layer 3). In this figure, the last layer's output,  $a^2$ , is assumed as the network output of interest. Thus, this output is considered as  $y$ , which could be either a vector or a scalar, varying with the case under study. As depicted in Fig. 1 the input vector  $P$ , which is an  $R$  length input vector, is denoted at the left by the solid vertical bar. This network has  $S^1$  neurons in the hidden layer and  $S^2$  neurons in its third layer (output layer). The third layer is assumed to be  $S^2 = 1$ . Accordingly, the network has  $R$  inputs  $P$ ,  $S^1$  outputs from the first hidden layer  $a^1$  and  $S^2 = 1$  outputs from the output layer,  $a^2 = y$ . These vectors can be described by the below expression:  $P = [p_1 \dots p_R]^T$ ,  $a^1 = [a_1^1 \dots a_{S^1}^1]^T$  and  $a^2 = y$ .

The components of every layer include the weight matrix  $w^i$ , the production operation, the vector of bias  $b^i$ , the summer, and  $\varphi^i$  which represents the activation function. The activation function transforms the weighted input of neuron to its output activation, where the superscripts  $i$  is assigned to the  $i$  layer. The matrices' sizes are demonstrated merely underneath their matrix parameter names. This representation allows us to apprehend the constructions of the layers and their associated matrix calculation. In all the layers (input layer, hidden layer and output layer), the bias and

weight matrices can be formulated as follows:

$$w^1 = \begin{bmatrix} w_{1,1}^1 & \dots & w_{1,R}^1 \\ \vdots & \ddots & \vdots \\ w_{S^1,1}^1 & \dots & w_{S^1,R^1}^1 \end{bmatrix},$$

$$w^2 = \begin{bmatrix} w_{1,1}^2 & \dots & w_{1,S^1}^2 \end{bmatrix}, \quad b^1 = [b_1^1 \dots b_{S^1}^1]^T \text{ and } b^2 = [b_1^2]$$

where the superscripts 1 and 2 are assigned to the hidden and output layers, respectively. Bias denotes an additional input attached to neurons. It permanently has a magnitude of 1 and handled such as other weights [42]. The motivation for attaching the bias factor is that it tolerates an illustration of marvels having limits [43]. The weights used in the model are indicative of the adjustable coefficients existing inside the network which specify the strength of the input sign [44]. Throughout this model, every component of the input vector  $P$  enters the network through the  $S^1 \times R$  weights matrix  $w^1$  in the hidden layer. A coefficient 1 put forward the neurons of the hidden layer as an input. Then the constants are multiplied by a bias, which is illustrated as  $b^1$  vector. The aforementioned vector is added to the weighted inputs,  $w^1 P$ , to generate the net input  $n^1$  vector with dimension of  $S^1$  as follows:

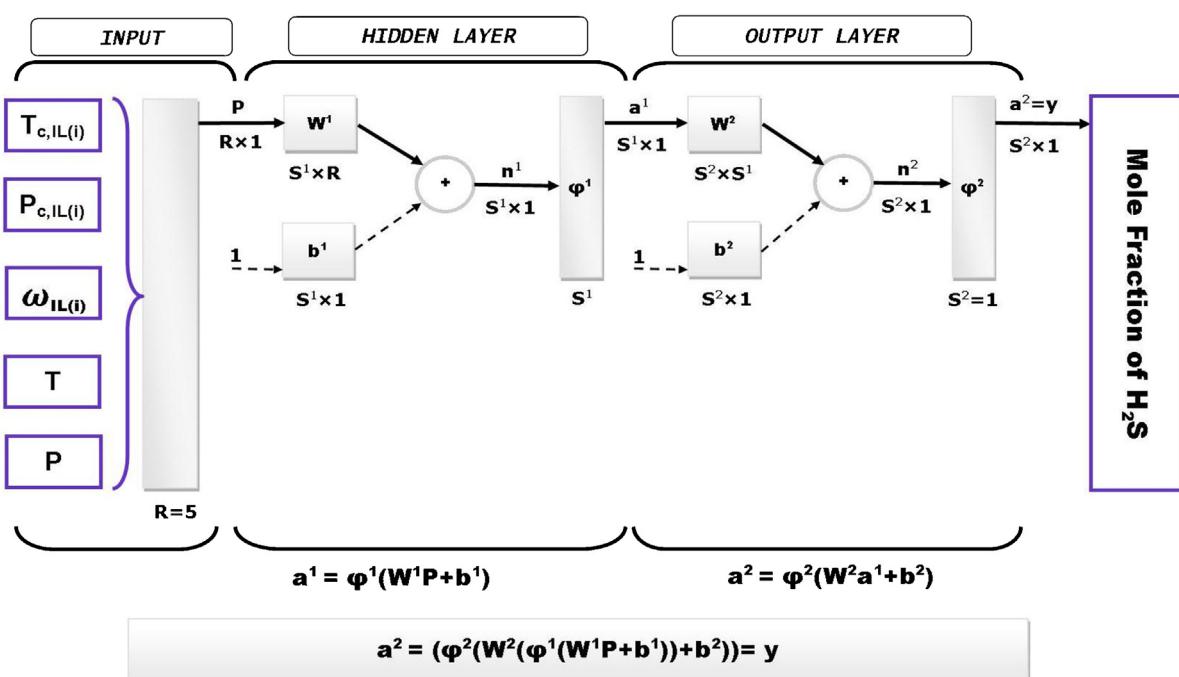
$$n^1 = w^1 P + b^1 \quad (4)$$

This summation is the representation of the transfer function  $\varphi^1$ . The output of the hidden layer  $a^1$  is a column vector, which can be described by the following equation:

$$a^1 = \varphi^1(w^1 P + b^1) \quad (5)$$

where  $\varphi^1$  represents the activation function of the hidden layer. The similar approach is continual in the output layer. Note that, the output of hidden layer is the input for the following layer (output layer). The net input of the third layer (output layer) is shown in Eq. (6). In this equation,  $a^2$  demonstrates the general output of the network. This value is shown in Eq. (7).

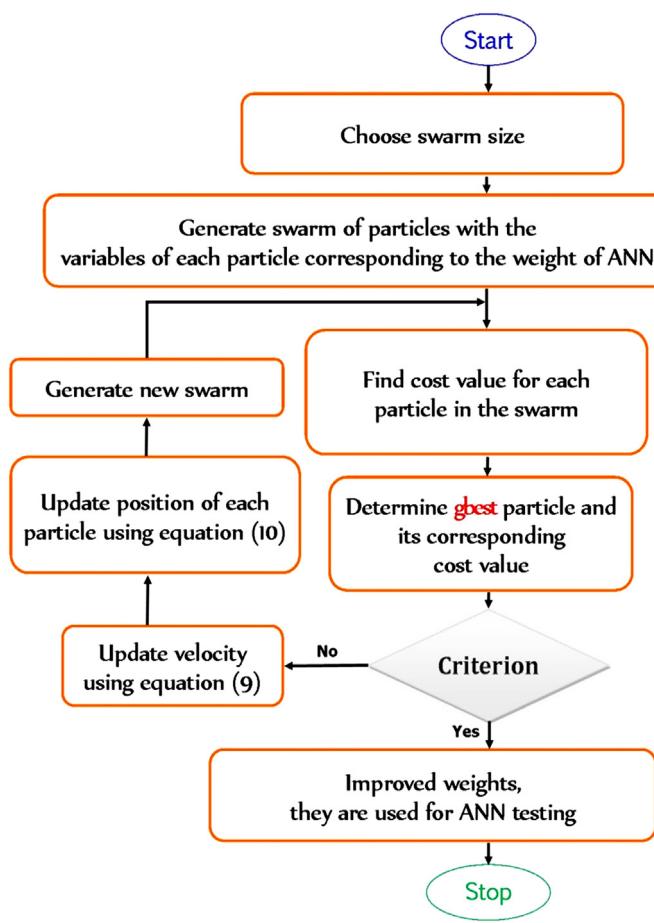
$$n^2 = w^2 a^1 + b^2 \quad (6)$$



**Fig. 1.** Schematic diagram of the three layer feed-forward neural network used in this study for prediction of H<sub>2</sub>S solubility in ILs.

**Table 1**Temperature, pressure and H<sub>2</sub>S solubility range of used ILs in this study.

| No. | Ionic liquid               | Temperature range (K) | Pressure range (MPa) | H <sub>2</sub> S solubility range (mole fraction) | No. of data points | Reference |
|-----|----------------------------|-----------------------|----------------------|---|--------------------|-----------|
| 1   | [hmim][PF6]                | 303.15–343.15         | 0.138–1.09           | 0.05–0.441  | 34                 | [12]      |
| 2   | [hmim][BF4]                | 303.15–343.15         | 0.111–1.1            | 0.06–0.499  | 33                 | [12]      |
| 3   | [emim][EtSO <sub>4</sub> ] | 303.15–353.15         | 0.1137–1.2704        | 0.012–0.118                                       | 36                 | [14]      |
| 4   | [C8mim][Tf <sub>2</sub> N] | 303.15–353.15         | 0.0935–1.9119        | 0.063–0.7355                                      | 47                 | [43]      |
| 5   | [C6mim][Tf <sub>2</sub> N] | 303.15–353.15         | 0.0685–2.0168        | 0.0368–0.7012                                     | 57                 | [43]      |
| 6   | [C8mim][PF6]               | 303.15–353.15         | 0.0845–1.9584        | 0.0463–0.6972                                     | 48                 | [15]      |
| 7   | [bmim][PF6]                | 303.15–343.15         | 0.123–1.011          | 0.044–0.405                                       | 42                 | [16]      |
| 8   | [bmim][BF4]                | 303.15–343.15         | 0.0608–0.836         | 0.03–0.354  | 42                 | [16]      |
| 9   | [bmim][Tf <sub>2</sub> N]  | 303.15–343.15         | 0.0944–0.916         | 0.051–0.51  | 44                 | [16]      |
| 10  | [emim][PF6]                | 333.15–363.15         | 0.1449–1.933         | 0.032–0.359                                       | 40                 | [44]      |
| 11  | [emim][Tf <sub>2</sub> N]  | 303.15–353.15         | 0.1077–1.686         | 0.049–0.609                                       | 42                 | [44]      |

**Fig. 2.** Flow diagram of PSO-based ANN procedure.

$$a^2 = \varphi^2 (w^2 a^1 + b^2) = \varphi^2 (w_{1,1}^2 a_1^1 + w_{1,2}^2 a_2^1 + \dots + w_{1,s^1}^2 a_{s^1}^1 + b_1^2) \quad (7)$$

where  $\varphi^2$  represents the activation functions third layer. It should be noted that  $w^i$  and  $b^i$  are both adaptive variables of the neuron. The dominant premise of NNs is that the adjustability of factors such as  $w^i$  and  $b^i$ , allows the network to exhibit the desired behavior. Thus, throughout training process, the biases and weights of the proposed NN model are systematically modified to minimize the NN objective function. The most frequent objective function for feed-forward NNs is mean square error (MSE). The system have to acquire by learning and adjusting to organized characteristics in the input configuration. This is accomplished by adjusting to statistical symmetries of configurations from the input training data.

Back-propagation, which is a controlled training approach based on the steepest descent principle, is one of the most frequently utilized methods for training feed-forward NNs [45,46]. In the BP training algorithm the network biases and weights are iteratively updated in the trend that the objective function MSE declines most promptly. In general, a single iteration of this algorithm to generate the new weights and biases can be written as:

$$W_k = W_{k-1} - \eta \text{grad}_{k-1} \quad (8)$$

here, the  $\eta$  is called the learning rate,  $W_{k-1}$  denotes a vector of existing biases and weights,  $\text{grad}_{k-1}$  represents the recent gradient of the objective function (MSE). For the details of arithmetical characteristics on BP training methods, valuable references have been provided in the literatures [46,47]. However, there is the possibility of getting trapped on suboptimal solutions during performing supervised training by BP algorithm since it is a gradient-based procedure. Given this problem, there is a need to apply robust global optimization schemes such as population-based stochastic optimization algorithms, enables the training process to escape from entrapment in local optimum in instances where the BP algorithm converges prematurely.

## 2.2. Particle swarm optimization

PSO is regarded as a global optimization technique which was first expressed by Kennedy and Eberhart [48]. The concept following the approach was enthused by the communal behavior of birds, such as swarm of insects and bird flocking. PSO is similar to the genetic algorithms (GAs) in which it starts with a crowd of a randomly generated population and the algorithm explores for optimal by modifying population. Though, dissimilar the GAs, PSO does not have evolution operators such as mutation and crossover. In PSO approach the population is composed of particles. Every particle travels almost the surface of cost (hyper surface that displays the cost) for all possible parameter values) with an adaptable velocity

**Table 2**

The critical properties and acentric factors of ILs used in this study.

| Compound                   | T <sub>c</sub> (K) | P <sub>c</sub> (MPa) | $\omega$ |
|----------------------------|--------------------|----------------------|----------|
| [hmim][PF6]                | 754.3              | 1.55                 | 0.8352   |
| [hmim][BF4]                | 679.1              | 1.79                 | 0.9258   |
| [emim][EtSO <sub>4</sub> ] | 1061.1             | 4.04                 | 0.3368   |
| [C8mim][Tf <sub>2</sub> N] | 1311.9             | 2.1                  | 0.4453   |
| [C6mim][Tf <sub>2</sub> N] | 1287.3             | 2.39                 | 0.3539   |
| [C8mim][PF6]               | 800.1              | 1.4                  | 0.9069   |
| [bmim][PF6]                | 708.9              | 1.73                 | 0.7553   |
| [bmim][BF4]                | 632.3              | 2.04                 | 0.8489   |
| [bmim][Tf <sub>2</sub> N]  | 1265               | 2.76                 | 0.2656   |
| [emim][PF6]                | 663.5              | 1.95                 | 0.6708   |
| [emim][Tf <sub>2</sub> N]  | 1244.9             | 3.26                 | 0.1818   |

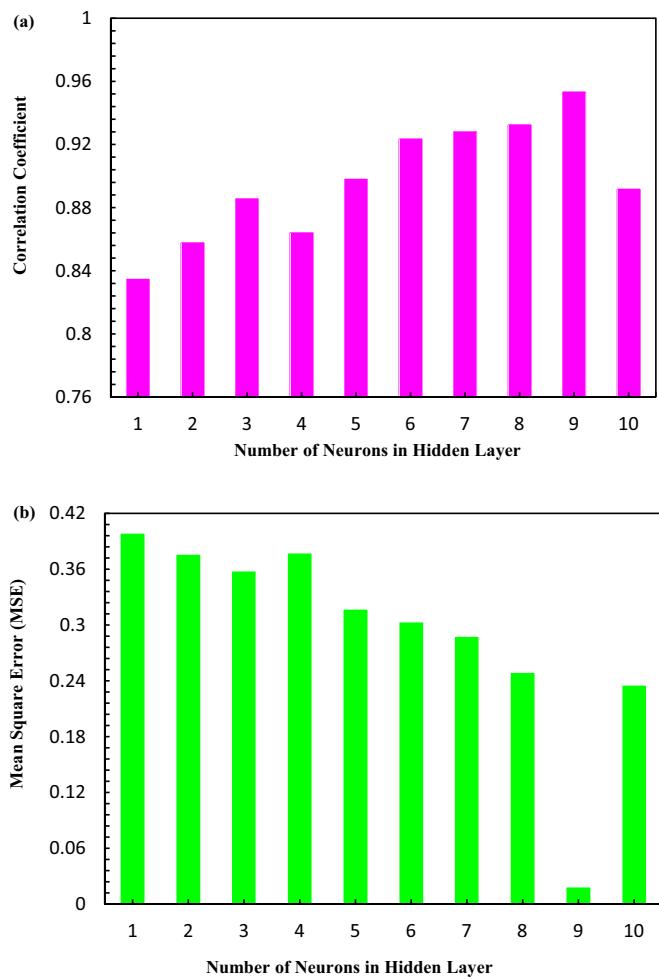


Fig. 3. Variation of: (a) R and (b) MSE with the number of hidden neurons.

[49]. In each iteration, the velocity and position of the particles are updated using the following equation:

$$v_i^{n+1} = \omega v_i^n + c_1 r_1^n [x_{i,p}^n - x_i^n] + c_2 r_2^n [x_g^n - x_i^n] \quad (9)$$

$$x_i^{n+1} = x_i^n + v_i^{n+1} \quad (10)$$

In which  $n$  represents the iteration number,  $i$  denotes the index of the particle,  $v$  stands for the particle velocity,  $x$  denotes the

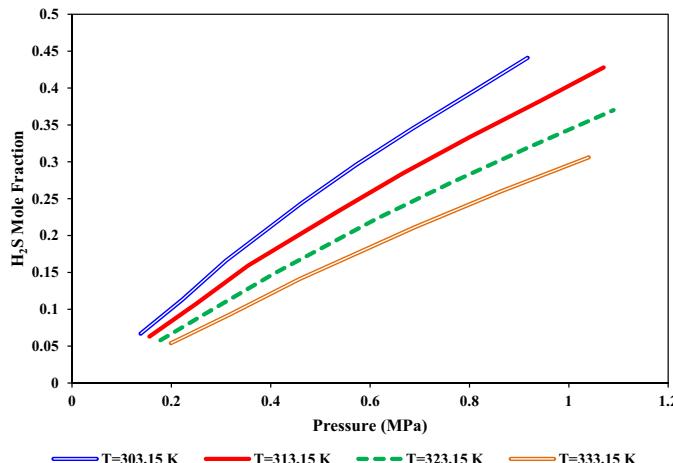


Fig. 4. H<sub>2</sub>S solubility in [hmim][BF<sub>4</sub>] versus pressure at different temperature.

position of particle,  $x_{i,p}$  represents the finest position of the particle  $i$  has visited since the first time step (pbest),  $x_g$  is the best global position, obtained thus far by any particle in the swarm (gbest),  $c_1$  and  $c_2$  are the acceleration factors related to pbest and gbest, respectively, and typically these values are set to 2,  $r_1$  and  $r_2$  are random values consistently scattered in the span [0,1] [50], and  $\omega$  is the inertia weight, introduced by Shi and Eberhart [51], that controls the exploitation and exploration of the exploration space [52]. Generally, the inertia weight is calculated by means of linear decreasing methodology where an initially large inertia weight is linearly decreased to a small value [53]:

$$\omega^n = \omega^{\max} = \left( \frac{\omega^{\max} - \omega^{\min}}{n_{\max}} \right) n \quad (11)$$

where  $\omega^{\max}$ ,  $\omega^{\min}$ ,  $n$  and  $n_{\max}$  are the initial inertia weight, the final inertia weight, current number of iterations and total number of iterations (maximum number of iteration used in PSO), respectively. Usually  $\omega^{\max}$  and  $\omega^{\min}$  values are equal to 0.9 and 0.4, respectively [33,34,48,53].

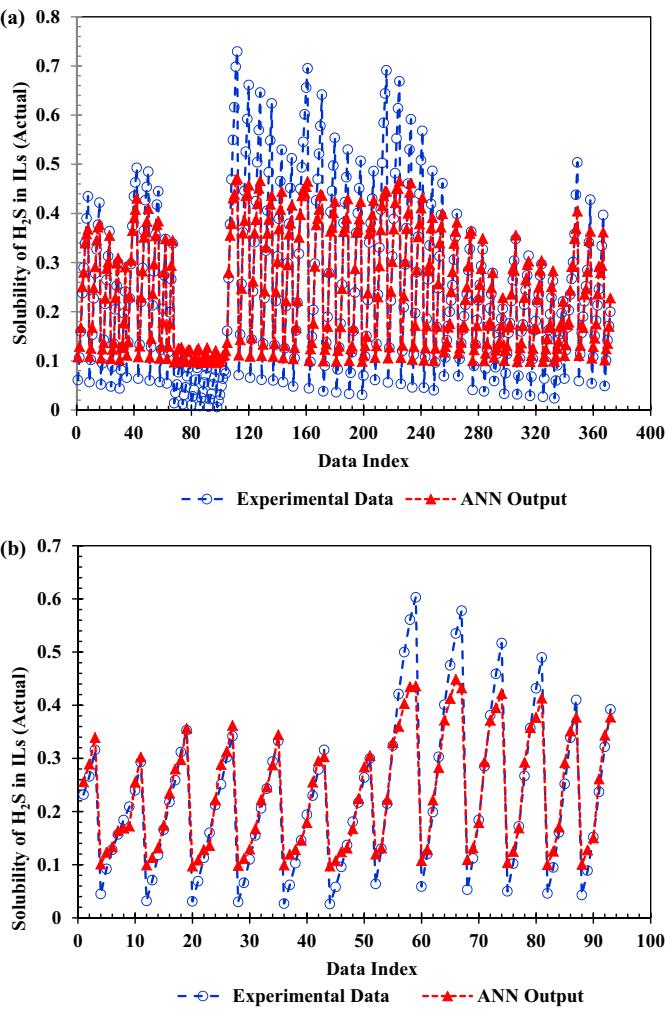
The PSO method modifies iteratively the vector of velocity for each particle then complements the aforementioned velocity to the position of particle. Velocity modifications are subjected to both the best global candidate route related with the final cost ever found by any particle in the swarm and the best personal candidate route connected with the final cost ever found by the particle. If the finest personal result has a cost less than the existing global route, then the finest personal route substitutes the finest global route [49]. As a matter of fact, personal and global lowest cost values and positions are updated by comparing the newly evaluated cost values against the previous personal and global lowest cost values, and replacing the lowest cost values and positions as necessary. Afterward, the velocity and position of the particles are updated according to above equations (Eqs. (9) and (10)). This technique is iteratively replicated pending a condition is satisfied, typically an acceptable cost value or a maximum number of iterations (generations) is accomplished. It should be mentioned that, cost evaluation is conducted by supplying the candidate solution to the cost function.

One of the first applications of PSO was to train neural networks [54,55]. Applying PSO in training neural networks have exhibited that the PSO is a viable alternative to neural network training. Thenceforth, several researchers have further investigated the capability of PSO as a training algorithm for a number of various neural networks. Inquiries have also shown for specific applications that neural networks trained using PSO give better results [53].

The main advantage of PSO algorithm over gradient-based algorithms such as BP lies in the dynamic interactions among the particles, which represent candidate solutions [56]. The particle makes adjustments by integrating particle experience with the discoveries of others [57]. Thus it is a social-psychological model of knowledge management [54]. While in BP, the particle makes judgments about stimuli through adjustments based only on personal experience [57].

### 3. Executions of ANN training using PSO algorithm

As mentioned, PSO can be utilized to train a neural network. In this case, every particle denotes a vector of weight (including weights and biases), and the mean square error (MSE) is employed as an objective function throughout this method. It is worth to point out that biases and weights are updated without executing any deviation signs, or any gradient info in contrast to training using BP algorithm, which results in simple determination if the size of the network rises. Indeed, an algorithm that relies only on gradient information can be challenging to apply while no gradient info is accessible for each activation functions. Weights are also not



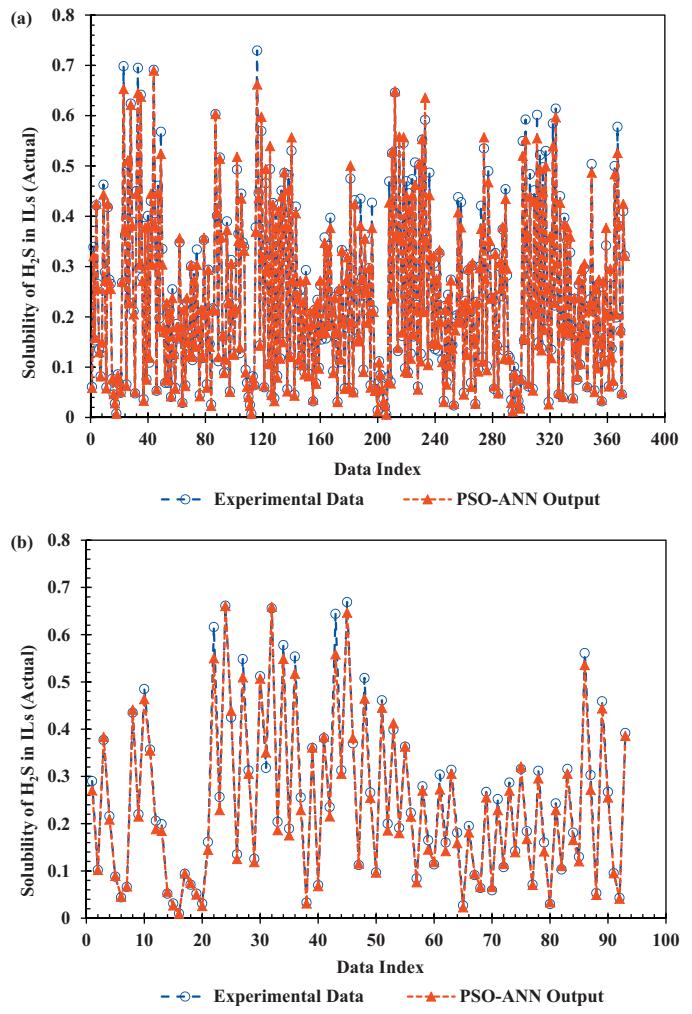
**Fig. 5.** Actual versus BP-ANN predicted mole fraction of  $\text{H}_2\text{S}$ : (a) training set and (b) testing set.

modified per training configuration. The PSO position and velocity modifier formulas (Eqs. (10) and (9)) are utilized to update biases and weights, subsequently which the training category is employed to specify the cost of a particle [53]. The aim of employment of the PSO is to minimize the objective function [34]. The flowchart of the employment of PSO algorithm for training the proposed neural network is shown in Fig. 2.

#### 4. Data preparation

To construct a truthful network, using dependable experimental data points is absolutely crucial, and to that end a literature survey is performed to collect experimental solubility of hydrogen sulfide in various ionic liquids data base at the wide span of pressure and temperature. Finally, a total of 465 data for 11 ILs were collected from 6 different references. The studied ionic liquids, mole fraction of  $\text{H}_2\text{S}$  in each ionic liquid and their corresponding range of temperatures, pressures accompanied with the references are presented in Table 1. The thermodynamic properties of the compounds included in the current study are illustrated in Table 2. In this table,  $T_c$  refers to the critical temperature,  $P_c$  is stands for the critical pressure, and  $w$  is the acentric factor. The data bases for the ionic liquids were gathered from the literature [58].

The acentric factor ( $w$ ), the critical temperature ( $T_c$ ), and the critical pressure ( $P_c$ ) of ionic liquids, to discriminate between different compounds, accompanied by the pressure ( $p$ ), and temperature ( $T$ )



**Fig. 6.** Actual versus PSO-ANN predicted mole fraction of  $\text{H}_2\text{S}$ : (a) training set and (b) testing set.

are considered as input variables and the solubility of  $\text{H}_2\text{S}$  in terms of mole fraction is regarded as output.

#### 5. Results and discussion

As mentioned, PSO-ANN is employed to approximate the non-linear relationship exiting between input variables and the output variable. Among 465 experimental  $\text{H}_2\text{S}$  solubility data points reported in the literature, 372 points of data bank are entered as training set and the residual 93 data points as the testing set (unseen data), which were not employed during the process of training, were introduced into the PSO-ANN in order to assess predictive power of the PSO-ANN. Establishing network structure is of vital importance for processing capability of the network. However, the optimal number of hidden layers and the optimal number of neurons throughout every layer, be subject to the application of the network and there is no direct approach for specification of them [59]. Hornik illustrated that multi-layer feed-forward NNs with one hidden layer and adequate neurons can connect each output to any input to a subjective point of precision [60]. Too few hidden neurons reduce the ability of the network to model while too much hidden neurons bring about poor generalization for unseen data. In this respect, by trial and error method ten different 5- $x$ -1 configurations ( $x$  varies from 1 to 10) were utilized with the intention of find the optimum network. Through rising the number of hidden neurons, the  $R^2$  and MSE were determined

**Table 3**

Details of trained ANN with PSO for the prediction of H<sub>2</sub>S solubility in ionic liquids.

| Type   | Value/comment |
|--|---------------|
| Input layer                                  | 5             |
| Hidden layer                                 | 9             |
| Output layer                                 | 1             |
| Hidden layer activation function             | Logsig        |
| Output layer activation function             | Purelin       |
| Number of data used for training             | 372           |
| Number of data used for testing              | 93            |
| Number of max iterations                     | 500           |
| c <sub>1</sub> and c <sub>2</sub> in Eq. (8) | 2             |
| Number of particles                          | 20            |

as shown in Fig. 3. Results demonstrate that the neural network with 9 neurons in its hidden layer is optimum number of hidden neurons as gave the minimum deviation. Table 3 represents several scenarios in adaption of network and PSO training variables. As a comparison, in addition to PSO algorithm, BP is also used to train ANN with the identical data samples employed throughout the PSO-ANN approach in which the value of momentum correction factor 0.001 and learning coefficient 0.7 are chosen.

To specify effectiveness and integrity of used smart approaches in estimation aforementioned target of this research, different statistical criteria are employed. These statistical parameters are calculated as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i^{\text{exp}} - y_i^{\text{pre}})^2 \quad (12)$$

$$R = \frac{\sum_{i=1}^N (y_i^{\text{exp}} - \bar{y}^{\text{exp}})(y_i^{\text{pre}} - \bar{y}^{\text{pre}})}{\sqrt{\sum_{i=1}^N (y_i^{\text{exp}} - \bar{y}^{\text{exp}})^2 \sum_{i=1}^N (y_i^{\text{pre}} - \bar{y}^{\text{pre}})^2}} \quad (13)$$

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i^{\text{exp}} - y_i^{\text{pre}})^2}{\sum_{i=1}^N (y_i^{\text{exp}} - \bar{y}^{\text{exp}})^2} \quad (14)$$

In which N represents the total number of data samples includes either training, testing or whole data set (input and output pairs),  $y_i^{\text{exp}}$  is the actual (experimental) at the sampling point i,  $y_i^{\text{pre}}$  is the ith output of the correlation,  $\bar{y}^{\text{exp}}$  and  $\bar{y}^{\text{pre}}$  are the average of the actual and predicted data.

Before explanation of the results gained with aforementioned smart connectionist techniques, it is worth to determine actual behavior of hydrogen sulfide solubility versus pressure at different temperature. Fig. 4 depicts the real trend of hydrogen solubility in [hmim][BF4] versus relevant pressure at different temperatures. As demonstrated in Fig. 4, through rising the temperature hydrogen solubility in [hmim][BF4] dramatically decreased at constant pressure. In this study, a proposed smart method should be sensitive to the temperature effect. In other words, PSO-ANN results have to same behavior versus pressure and temperature.

Predicted and actual values of the solubilities at training and testing stages for both PSO-ANN and BP-ANN methods are evaluated as demonstrated in Figs. 5 and 6. As shown in Fig. 6, the results of the PSO-ANN method in assessment with real samples illustrates an excellent accuracy of PSO-ANN method, while BP-ANN prediction results are not as satisfactorily precise as PSO-ANN results (see Fig. 5).

The regression plots, as shown in Figs. 7 and 8, display the results of both PSO-ANN and BP-ANN methods with respect to actual data for test and training categories. The dashed line in every figure demonstrates the faultless output-model outputs = actual values. The solid line demonstrates the finest fitted linear regression line between model results and actual values. For a faultless fit, the data have to lie over a 45 degree line, Y = X, where the model results are

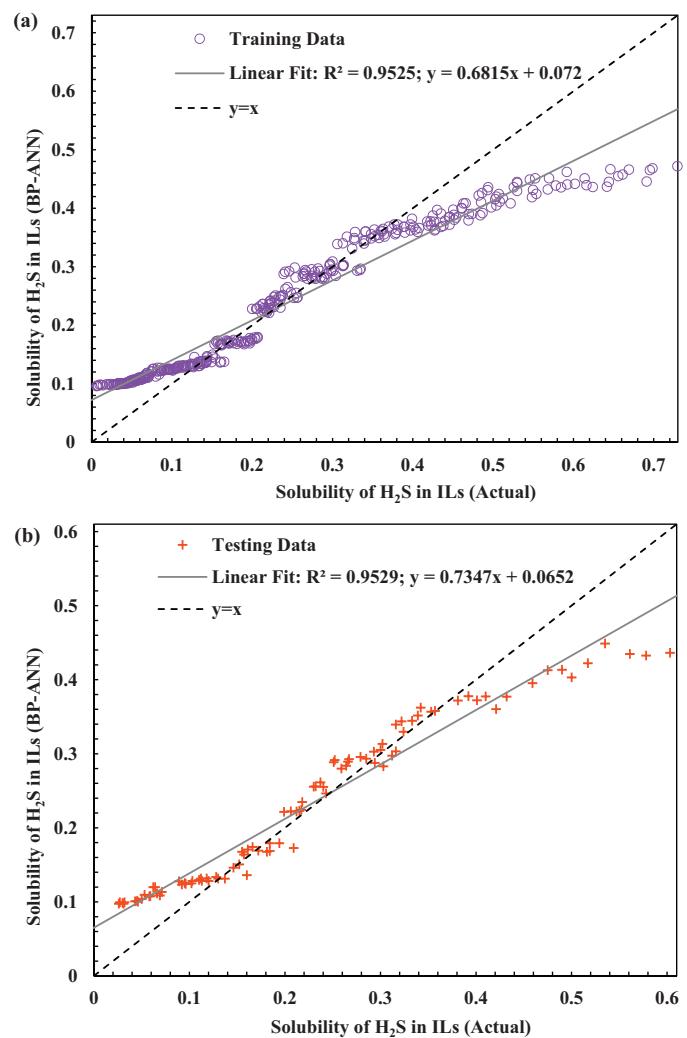


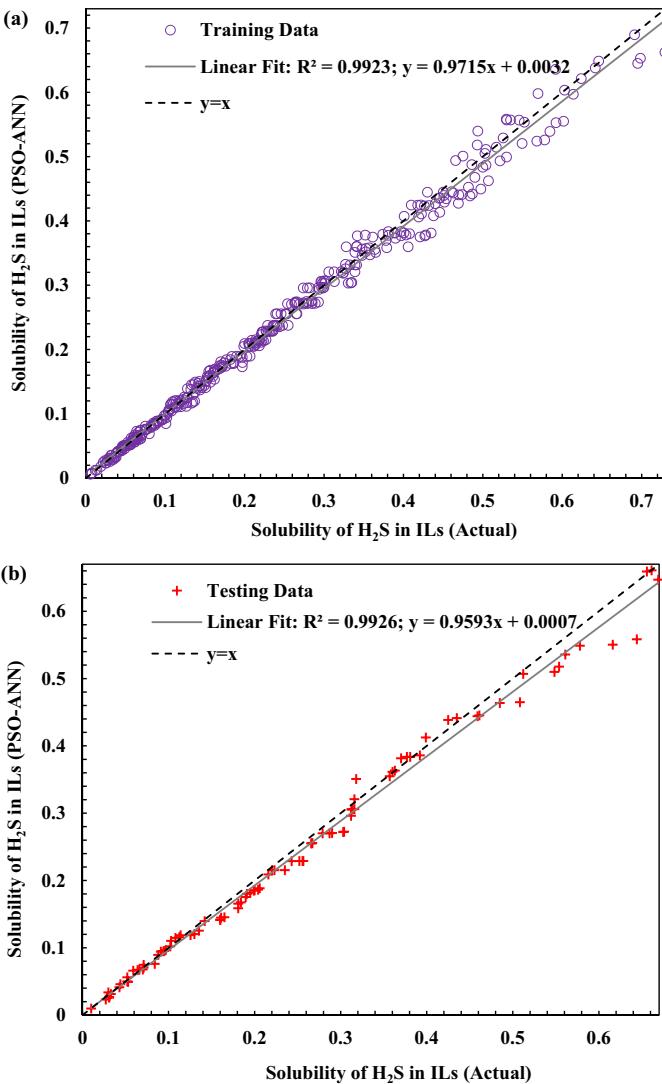
Fig. 7. Regression plots for the solubility of H<sub>2</sub>S prediction using BP-ANN model: (a) training set and (b) testing set.

identical to the actual values. For this issue, the fit is practically reliable for both data categories (i.e. training and testing). Therefore, the PSO-ANN outputs tracks the actual values very well for training, and testing phases, and the R is over 0.99 for the testing and training data category (see Fig. 8). As can be seen, compared with the BP-ANN, PSO-ANN yields better prediction results. Moreover, as depicted in Fig. 7, BP-ANN method is under estimated the lower boundary of hydrogen solubility in ILs; however, it is over estimated the upper boundary of hydrogen solubility in ILs. Despite the good agreement between BP-ANN results and intermediate boundary of H<sub>2</sub>S solubility in ILs, BP-ANN method is not capable to estimate H<sub>2</sub>S solubility correctly for entire boundary of H<sub>2</sub>S solubility. From statistical point of view, Eqs. (15) and (16) are generated as the linear regressions for the test data assortment for BP-ANN and PSO-ANN Methods correspondingly, as following as:

$$y = 0.7347x + 0.0652; \quad R^2 = 0.9529 \quad (15)$$

$$y = 0.9593x + 0.0007; \quad R^2 = 0.9926 \quad (16)$$

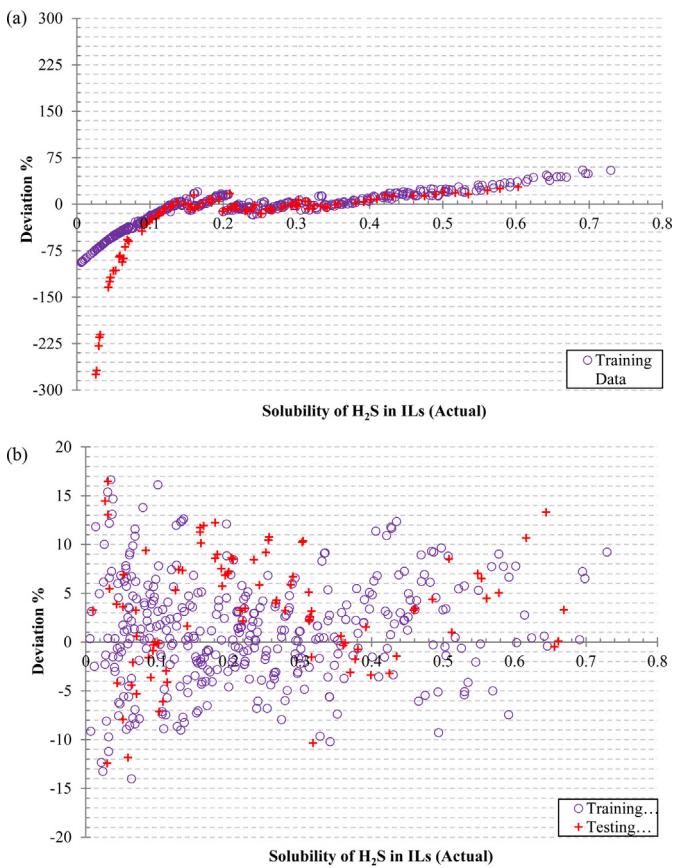
As demonstrated in Fig. 8, the regression lines overlays with the diagonal line, as an outcome of a slope magnitude close to 1 and infinitesimal magnitude of the intercept, this means that, PSO-ANN method estimate the real values of hydrogen sulfide solubility in different ILs at high precision and integrity rate.



**Fig. 8.** Regression plots for the solubility of H<sub>2</sub>S prediction using PSO-ANN model: (a) training set and (b) testing set.

The percent deviation between actual data and models outputs ( $\% \text{deviation} = ((n_{\text{H}_2\text{S}}^{\text{Actual}} - n_{\text{H}_2\text{S}}^{\text{Predicted}}) / n_{\text{H}_2\text{S}}^{\text{Actual}}) \times 100$ ) are shown in Fig. 9. The percent deviation of PSO-ANN outputs form actual values recline in the span –14.02% to 16.63%, the average absolute deviation is 4.58%, and the magnitude of minimum relative deviation is 0.00%, while the percent deviation of BP-ANN outputs form actual values lie in the range –274.88% to 55.13%, the average absolute deviation is 23.59%, and the magnitude of minimum relative deviation is 0.00%. 90.75 and 44.73% of data for PSO-ANN and BP-ANN are in the range of  $\pm 10\%$  deviation. As demonstrated in Fig. 9, PSO-ANN can estimate solubility of hydrogen sulfide in different ILs much superior than BP-ANN. Superior robustness of PSO-ANN method is also confirmed by evaluating MSE, R (coefficient of determination), and R (correlation coefficient), of two models in Table 4.

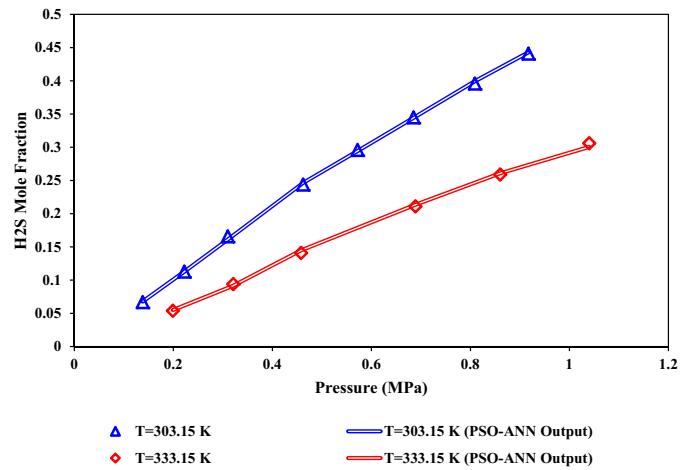
Finally, Fig. 10 demonstrates the sensitivity of the outputs gained from proposed smart technique for [hmim][BF4] system versus corresponding pressure at  $T = 303.15\text{ K}$  and  $333.15\text{ K}$ . Moreover, actual values of H<sub>2</sub>S solubility in [hmim][BF4] are marked in the aforementioned figure. As illustrated in Fig. 10, PSO-ANN approach has high level of integrity and accuracy in comparison with corresponding actual data samples.



**Fig. 9.** Relative deviation of predicted H<sub>2</sub>S solubility values using: (a) BP-ANN model and (b) PSO-ANN model versus experimental ones.

**Table 4**  
Comparison between the performances of PSO-ANN and BP-ANN model.

|     | PSO-ANN    |           |              | BP-ANN     |           |              |
|-----|------------|-----------|--------------|------------|-----------|--------------|
|     | Train data | Test data | Overall data | Train data | Test data | Overall data |
| $R$ | 0.99616    | 0.99628   | 0.99608      | 0.97596    | 0.97616   | 0.97545      |
| $R$ | 0.99234    | 0.99258   | 0.99218      | 0.95250    | 0.95288   | 0.95151      |
| MSE | 0.00023    | 0.00034   | 0.00025      | 0.00368    | 0.00205   | 0.00335      |



**Fig. 10.** Comparison of PSO-ANN outputs and real values of H<sub>2</sub>S solubility in [hmim][BF4] at  $T = 333.15\text{ K}$  and  $303.15\text{ K}$ .

## 6. Conclusion

In this study, a three layer feed forward neural network is employed for prediction of the hydrogen sulfide solubility in various ILs based on experimental results gathered from the previous published papers. For proposing a reliable model, 11 various ILs were examined. The acentric factor ( $\omega$ ), the critical pressure ( $P_c$ ), and the critical temperature ( $T_c$ ) of ILs together with the pressure ( $P$ ), and temperature ( $T$ ) are used as input variables. Two distinct training methods are used in this study for training the network: back propagation (BP) and particle swarm optimization techniques (PSO). The results demonstrated the superiority PSO-ANN over BP-ANN in training of neural network, based on statistical criteria represents quality of models outputs by virtue of PSO algorithm is able to hill-climb out of local optima while BP is a deterministic algorithm that depends only on gradient information are very prone to get stuck on a local optima. The PSO-ANN method is accomplished of estimating the solubility of hydrogen sulfide in the considered ILs as it yields a low average absolute deviation of 4.58% considering all 465 experimental data points. The proposed PSO-ANN model can be a viable alternative to thermodynamic models in the literature as it is computationally inexpensive, simple and easy to use along with high accurate results.

## References

- [1] S. Mortazavi-Manesh, M. Satyro, R.A. Marriott, Modelling carbon dioxide solubility in ionic liquids, *Canadian J. Chemical Engineering* 91 (2013) 783–789.
- [2] M.O. Schach, R. Schneider, H. Schramm, J.U. Repke, Techno-economic analysis of postcombustion processes for the capture of carbon dioxide from power plant flue gas, *Industrial Engineering Chemistry Research* 49 (2010) 2363–2370.
- [3] X. Wang, T. Sun, J. Yang, L. Zhao, J. Jia, Low-temperature  $H_2S$  removal from gas streams with SBA-15 supported ZnO nanoparticles, *Chemical Engineering J.* 142 (2008) 48–55.
- [4] H. Sakhaeinia, V. Taghikhani, A.H. Jalili, A. Mehdizadeh, A.A. Safekordi, Solubility of  $H_2S$  in 1-(2-hydroxyethyl)-3-methylimidazolium ionic liquids with different anions, *Fluid Phase Equilibria* 298 (2010) 303–309.
- [5] M.M. Taib, T. Murugesan, Solubilities of  $CO_2$  in aqueous solutions of ionic liquids (ILs) and monoethanolamine (MEA) at pressures from 100 to 1600 kPa, *Chemical Engineering J.* 181 (182) (2012) 56–62.
- [6] D. Stirling, The Sulfur Problem: Cleaning up Industrial Feedstocks, The Royal Society of Chemistry, 2000.
- [7] J.E. Bara, Potential for hydrogen sulfide removal using ionic liquid solvents, in: A. Mohammad, Dr. Inamuddin (Eds.), *Green Solvents II, Properties and Applications of Ionic Liquids*, Springer Dordrecht Heidelberg, New York, London, 2012, pp. 155–168.
- [8] M.B. Shiflett, A. Yokozeiki, Separation of  $CO_2$  and  $H_2S$  using room-temperature ionic liquid [bmim][PF6], *Fluid Phase Equilibria* 294 (2010) 105–113.
- [9] R.J. Kidney, W.R. Parrish, *Fundamentals of Natural Gas Processing*, Taylor & Francis, Boca Raton, FL, 2006.
- [10] G. Astarita, D.W. Savage, A. Bisio, *Gas Treating With Chemical Solvents*, Wiley, New York, NY, 1983.
- [11] S. Mortazavi-Manesh, M. Satyro, R.A. Marriott, A semi-empirical Henry's law expression for carbon dioxide dissolution in ionic liquids, *Fluid Phase Equilibria* 307 (2011) 208–215.
- [12] J.F.G. Brennecke, E.J. Maginn, *Purification of Gas with Liquid Ionic Compounds*, University of Notre Dame du Lac, Notre Dame, IN, United States, 2003.
- [13] Ch. JagadeeswaraRao, K.A. Venkatesan, K. Nagarajan, T.G. Srinivasan, P.R. VasudevaRao, Electrodeposition of metallic uranium at near ambient conditions from room temperature ionic liquid, *J. Nuclear Materials* 408 (2011) 25–29.
- [14] R.P. Swatloski, S.K. Spear, J.D. Holbrey, R.D. Rogers, Dissolution of cellulose with ionic liquids, *J. American Chemistry Society* 124 (2002) 4974–4975.
- [15] A. Shariati, S.S. Ashrafmansouri, M. Haji Osbuei, B. Hooshdaran, Critical properties and acentric factors of ionic liquids, *Korean J. Chemical Engineering* 30 (2013) 187–193.
- [16] A.N. Soriano, B.T. Doma, M.H. Li, Carbon dioxide solubility in some ionic liquids at moderate pressures, *J. Taiwan Institute of Chemical Engineering* 40 (2009) 387–393.
- [17] C.S. Pomicelli, C. Chiappe, A. Vidis, G. Laurenczy, P.J. Dyson, Influence of the interaction between hydrogen sulfide and ionic liquids on solubility: experimental and theoretical investigation, *J. Physical Chemistry B* 111 (2007) 13014–13019.
- [18] F.-Y. Jou, A.E. Mather, Solubility of hydrogen sulfide in [bmim][PF6], *International J. Thermophysics* 28 (2007) 490–495.
- [19] A.H. Jalili, M. Safavi, C. Ghotbi, A. Mehdizadeh, M. Hosseini-Jenab, V. Taghikhani, Solubility of  $CO_2$ ,  $H_2S$ , and their mixture in the ionic liquid 1-octyl-3-methylimidazolium bis(trifluoromethyl)sulfonylimide, *Physical Chemistry B* 116 (2012) 2758–2774.
- [20] H. Sakhaeinia, A.H. Jalili, V. Taghikhani, A.A. Safekordi, Solubility of  $H_2S$  in ionic liquids 1-ethyl-3-methylimidazolium hexafluorophosphate ([emim][PF6]) and 1-ethyl-3-methylimidazolium bis(trifluoromethyl)sulfonylimide ([emim][TF2N]), *J. Chemical Engineering and Data* 55 (2010) 5839–5845.
- [21] A. Eslamimanesh, F. Gharagheizi, A.H. Mohammadi, D. Richon, Artificial neural network modeling of solubility of supercritical carbon dioxide in 24 commonly used ionic liquids, *Chemical Engineering Science* 66 (2011) 3039–3044.
- [22] X. Ji, H. Adidharma, Thermodynamic modeling of  $CO_2$  solubility in ionic liquid with heterosegmented statistical associating fluid theory, *Fluid Phase Equilibria* 293 (2010) 141–150.
- [23] J.S. Torrecilla, J. Palomar, J. García, E. Rojo, F. Rodríguez, Modelling of carbon dioxide solubility in ionic liquids at sub and supercritical conditions by neural networks and mathematical regressions, *Chemometry Intelligent Laboratory* 93 (2008) 149–159.
- [24] P.F. Arce, P.A. Robles, T.A. Graber, M. Aznar, Modeling of high-pressure vapor–liquid equilibrium in ionic liquids + gas systems using the PRSV equation of state, *Fluid Phase Equilibria* 295 (2010) 9–16.
- [25] L.F. Vega, O. Vilaseca, F. Llorell, J.S. Andreu, Modeling ionic liquids and the solubility of gases in them: recent advances and perspectives, *Fluid Phase Equilibria* 294 (2010) 15–30.
- [26] M. Safamirzaei, H. Modarress, Correlating and predicting low pressure solubility of gases in [bmim][BF4] by neural network molecular modeling, *Thermochimica Acta* 545 (2012) 125–130.
- [27] M. Safamirzaei, H. Modarress, Application of neural network molecular modeling for correlating and predicting Henry's law constants of gases in [bmim][PF6] at low pressures, *Fluid Phase Equilibria* 332 (2012) 165–172.
- [28] M.A. Ahmadi, R. Haghbakhsh, R. Soleimani, M. Bazrgar Bajestani, Estimation of  $H_2S$  solubility in ionic liquids using a rigorous method, *J. Supercritical Fluid* 92 (2014) 60–69.
- [29] S.S. Haykin, *Neural Networks A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, 1999.
- [30] M. Safamirzaei, H. Modarress, Modeling and predicting solubility of *n*-alkanes in water, *Fluid Phase Equilibria* 309 (2011) 53–61.
- [31] M. Lashkarbolooki, Z. Sadat Shafipour, A. Zeinolabedini Hezave, H. Farmani, Use of artificial neural networks for prediction of phase equilibria in the binary system containing carbon dioxide, *J. Supercritical Fluid* 75 (2013) 144–151.
- [32] G.B. Sahoo, C. Ray, Predicting flux decline in crossflow membranes using artificial neural networks and genetic algorithms, *J. Membrane Science* 283 (2006) 147–157.
- [33] S. Zendehboudi, M.A. Ahmadi, L. James, I. Chatzis, Prediction of condensate-to-gas ratio for retrograde gas condensate reservoirs using artificial neural network with particle swarm optimization, *Energy Fuels* 26 (2012) 3432–3447.
- [34] M.A. Ahmadi, S.R. Shadizadeh, New approach for prediction of asphaltene precipitation due to natural depletion by using evolutionary algorithm concept, *Fuel* 102 (2012) 716–723.
- [35] S. Zendehboudi, M.A. Ahmadi, A. Bahadori, A. Shafiei, T. Babadagli, A developed smart technique to predict minimum miscible pressure—eօr implications, *Canadian J. Chemical Engineering* 91 (2013) 1325–1337.
- [36] M.A. Ahmadi, S. Zendehboudi, A. Lohi, A. Elkamel, I. Chatzis, Reservoir permeability prediction by neural networks combined with hybrid genetic algorithm and particle swarm optimization, *Geophysical Prospecting* 61 (2013) 582–598.
- [37] M.A. Ahmadi, M. Golshadi, Neural network based swarm concept for prediction asphaltene precipitation due to natural depletion, *J. Petroleum Science and Engineering* 98 (99) (2012) 40–49.
- [38] M.A. Ahmadi, Neural network based unified particle swarm optimization for prediction of asphaltene precipitation, *Fluid Phase Equilibria* 314 (2012) 46–51.
- [39] M.A. Ahmadi, M. Ebadi, A. Shokrollahi, S.M.J. Majidi, Evolving artificial neural network and imperialist competitive algorithm for prediction oil flow rate of the reservoir, *Applied Soft Computing* 13 (2013) 1085–1098.
- [40] S. Zendehboudi, M.A. Ahmadi, O. Mohammadzadeh, A. Bahadori, I. Chatzis, Thermodynamic investigation of asphaltene precipitation during primary oil production: laboratory and smart technique, *Industrial Engineering Chemistry Research* 52 (2013) 6009–6031.
- [41] R. Reed, Pruning algorithms—A survey, *Neural Networks*, IEEE Transactions Neural Networks 4 (1993) 740–747.
- [42] M.A. Ahmadi, M. Ebadi, S.M. Hosseini, Prediction breakthrough time of water coning in the fractured reservoirs by implementing low parameter support vector machine approach, *Fuel* 117 (2014) 579–589.
- [43] M.A. Ahmadi, M. Ebadi, Evolving smart approach for determination dew point pressure through condensate gas reservoirs, *Fuel* 117 (2014) 1074–1084 (Part B).
- [44] M.A. Ahmadi, M. Ebadi, P.S. Marghamaleki, M.M. Fouladi, Evolving predictive model to determine condensate-to-gas ratio in retrograded condensate gas reservoirs, *Fuel* 124C (2014) 241–257.
- [45] K.A. Al-Shayji, *Modeling, Simulation and Optimization of Large-Scale Commercial Desalination Plant*, Virginia Polytechnic Institute and State University, Virginia, USA, 1998 (PhD thesis).
- [46] M.H. Beale, M.T. Hagan, H.B. Demuth, *Neural Network Toolbox™ User's Guide*, The MathWorks, Inc., 2012.
- [47] M.T. Hagan, H.B. Demuth, M. Beale, *Neural Network Design*, PWS Publishing Co., Boston, MA, 1996.
- [48] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proc. IEEE Int. Conf. on Neural Networks, IV, Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948.

- [49] R.L. Haupt, S.E. Haupt, Practical Genetic Algorithms, second ed., John Wiley & Sons, Inc., 2004.
- [50] D.M. Prata, M. Schwaab, E.L. Lima, J.C. Pinto, Nonlinear dynamic data reconciliation and parameter estimation through particle swarm optimization: application for an industrial polypropylene reactor, *Chemical Engineering Science* 64 (2009) 3953–3967.
- [51] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: *Proceedings of the IEEE Congress on Evolutionary Computation*, 1998, pp. 69–73.
- [52] S.N. Sivanandam, S.N. Deepa, *Introduction to Genetic Algorithms*, Springer-Verlag, Berlin, Heidelberg, 2008.
- [53] A.P. Engelbrecht, *Computational Intelligence: An Introduction*, second ed., John Wiley & Sons: University of Pretoria South Africa, 2007.
- [54] J. Kennedy, The particle swarm: social adaptation of knowledge, in: *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1997, pp. 303–308.
- [55] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: *Proceedings of the Sixth International Symposium on Micromachine and Human Science*, 1995, pp. 39–43.
- [56] R. Mendes, P. Cortez, M. Rocha, J. Neves, Particle swarms for feed forward neural network training, in: *Proc. Int. Joint Conf. Neural Networks (IJCNN '02)*, 2002, pp. 1895–1899, 2.
- [57] J. Kennedy, R. Eberhart, Y. Shi, *Swarm Intelligence*, Morgan Kaufmann, 2001.
- [58] J.O. Valderrama, P.A. Robles, Critical properties, normal boiling temperatures, and acentric factors of fifty ionic liquids, *Industrial Engineering Chemistry Research* 46 (2007) 1338–1344.
- [59] M.A. Ahmadi, M. Ebadi, A. Yazdanpanah, Robust intelligent tool for estimation dew point pressure in retrograded condensate gas reservoirs: application of particle swarm optimization, *J. Petroleum Science and Engineering* (2014), <http://dx.doi.org/10.1016/j.petrol.2014.05.023>.
- [60] K. Hornik, M. Stinchcombe, H. White, Multi-layer feed forward networks are universal approximations, *Neural Networks* 2 (1989) 359–366.