The modelling of lead removal from water by deep eutectic solvents functionalized CNTs: artificial neural network (ANN) approach

Seef Saadi Fiyadh, Mohammed Abdulhakim AlSaadi, Mohamed Khalid AlOmar, Sabah Saadi Fayaed, Ako R. Hama, Sharifah Bee and Ahmed El-Shafie

ABSTRACT

The main challenge in the lead removal simulation is the behaviour of non-linearity relationships between the process parameters. The conventional modelling technique usually deals with this problem by a linear method. The substitute modelling technique is an artificial neural network (ANN) system, and it is selected to reflect the non-linearity in the interaction among the variables in the function. Herein, synthesized deep eutectic solvents were used as a functionalized agent with carbon nanotubes as adsorbents of Pb²⁺. Different parameters were used in the adsorption study including pH (2.7 to 7), adsorbent dosage (5 to 20 mg), contact time (3 to 900 min) and Pb²⁺ initial concentration (3 to 60 mg/l). The number of experimental trials to feed and train the system was 158 runs conveyed in laboratory scale. Two ANN types were designed in this work, the feed-forward back-propagation and layer recurrent; both methods are compared based on their predictive proficiency in terms of the mean square error (MSE), root mean square error, relative root mean square error, mean absolute percentage error and determination coefficient (R²) based on the testing dataset. The ANN model of lead removal was subjected to accuracy determination and the results showed R² of 0.9956 with MSE of 1.66 × 10⁻⁴. The maximum relative error is 14.93% for the feed-forward back-propagation neural network model.

Key words | carbon nanotubes, deep eutectic solvents, feed forward, lead ions, neural network, water treatment

INTRODUCTION

The removal of heavy metal ions from water has been a crucial step to curb the resulting environmental problem. Any presence of heavy metals in water is recognized as a threat to both human health and aquatic organisms (Wang & Chen 2006). In addition, the non-biodegradability of heavy metals to their tendency to build up in living organisms may lead to various disease. Heavy metals may be present in the solution as free ions or in the form of molecules, and chelate metal ligands in any water streams (Salisu et al. 2016). Lead is known to be one of the primary heavy metals discharged into the environment by battery manufacturing, metal electroplating, pigment and dye industries (Majumdar et al. 2010). The consumption of such contaminated water may affect the kidney, brain, liver and central nervous system, which subsequently will lead to irreversible brain damage, weakness of muscles and nervous disorders (Geetha et al. 2015). Research has been done to prove that the adsorption technique is one of the effective methods to extract metal ions from water solution (Pimentel et al. 2007). The effectiveness of adsorption is very dependent on the selection of appropriate process conditions, including the mass of sorbent, duration of the process, pH and temperature of the system (Lourie & Gjengedal 2011). Many studies have been done on different materials to be used...
as an adsorbent to extract metal cations from water, for example, activated carbon (Chen & Wu 2004), clay minerals (Oubagaranadin & Murthy 2010), biomaterials (Gupta et al. 2006) and Pistacia vera shells (Yetilmezsoy & Demirel 2008). However, these adsorbents have not given satisfying results.

Researchers have suggested carbon nanotubes (CNTs) to be used as effective adsorbents to extract numerous pollutants such as dyes, metal ions, phenols, aniline, drugs and other contaminants (Ibrahim et al. 2016). The qualities of CNTs such as large surface area and diameter and the shorter equilibrium time than other materials contribute to its applications. CNTs have the potential to be used in a variety of applications due to its distinctive electrical, physical and chemical properties.

CNTs have been successfully used to extract various heavy metals, for instance chromium, copper, zinc, lead, cadmium, arsenic and mercury (Ihsanullah et al. 2016). Researchers have reported that CNTs are efficient and significantly adsorb lead more than copper and cadmium in the suitable pH value. Also, the presence of different ions, the strength of ions and the pH value are the major criteria that influence the adsorption of Pb\(^{2+}\) (Kabbashi et al. 2009).

Moreover, CNTs showed a great affinity for interaction with different compounds (Ihsanullah et al. 2016). Therefore, functionalization is the key to improve the activity of CNTs. The conventional functionalization usually involves harsh acids and non-environmentally friendly chemicals with complicated processes. Consonantly, the need for environmentally friendly functionalization agents with simple chemical processes is crucial (AlOmar et al. 2017b).

Recently, deep eutectic solvents (DESs) have gained an enormous amount of interest due to their involvement in many applications. DESs were first introduced as a low-cost development or replacement for ionic liquids (ILs). DESs have many advantages over ILs in terms of availability of the raw materials and easy synthesis with minimum environmentally harmful waste (AlOmar et al. 2016c). Therefore, DESs have conquered many fields of science. Lately, DES involvement in many nanotechnology-related fields has included media for synthesis of nanoparticles (Chen et al. 2014; Chakrabarti et al. 2015; Jia et al. 2015; Xu et al. 2016), electrolyte in nanostructure sensors (Zheng et al. 2014), electrolyte in nanoparticle deposition (Abbott et al. 2009; Gu & Tu 2011; Renjith et al. 2014) and as a functionalization agent of CNTs (AlOmar et al. 2016a, 2017a). AlOmar et al. (2016a, 2016b, 2016c) have used choline chloride based DESs as a functionalization agent of CNTs to prepare a sufficient adsorbent of Pb\(^{2+}\) ions. Consequently, the dataset prepared from that work has been implemented for the modelling in this work (AlOmar et al. 2016b).

New techniques such as the artificial neural network (ANN) have been considered as a less complicated model in the sophisticated biological network. An ANN system, as a substitute technique of modelling, has been selected in order to represent the non-linear function relationship among variables. The ANN techniques do not require any mathematical induction since the ANN analyses and recognizes the patterns in a series of inputs and outputs of a dataset without any prior assumptions about their characteristics and interrelations (Mandal et al. 2009). The speciality of the ANN to generalize and identify the pattern of any non-linear and complex development makes it an influential modelling means. A neural network has the ability to extract complicated data that are beyond the capability of being observed by a human or any computer technique. Experiments have been successfully performed to use ANN to model the adsorption of lead ions by Pistacia vera L. shells (Yetilmezsoy & Demirel 2008), the removal of Lanuset Red G on Chara contraria (Mjalli et al. 2007), Lanuset Red G on walnut husk removal efficiency (Çelekli et al. 2012), and the intercalated tartrate-Mg-Al layered double hydroxides as an adsorbent (Yasin et al. 2014). Several studies have recently been conducted on water quality prediction models (Wu & Xu 2011; Chibole 2013). Moreover, there some research has been conducted in different areas, for example, modelling the fermentation media optimization (Desai et al. 2008) and modelling of a microwave-assisted extraction method (Khajeh & Moghaddam 2011).

**Problem statement**

The ANN is used to predict the adsorbent capacity for Pb\(^{2+}\) from water by using a set of experimental data that have been prepared in advance. The main advantages of using the ANN are its precision and efficacy in apprehending the non-linear relationship between the variables of multi-input or output in a complicated system. Moreover, the availability in abundance and easy handing of an ANN makes it an economic and good option to predict Pb\(^{2+}\) adsorption. From an economical perspective, an ANN can be utilised as a substitute for CNTs, which are relatively higher in cost, to study the adsorption process. Furthermore, the adsorption prediction model can play a key role in providing relevant information related to the input variables. In addition to that, the development of such models can be considered as low-cost, and reduce the engineering effort.
Objective

The aims of this study are as follows. An ANN model will be created to establish the relationship that exists between the adsorbent dosage, concentration of Pb²⁺, pH and contact time to predict the DES-CNTs adsorption capacity of Pb²⁺ from water solution based on the experimental dataset prepared in laboratory scale (AlOmar et al. 2016b). (2) The adsorption capacity of the DES-CNTs adsorbent for Pb²⁺ will be predicted by using the ANN model and compared with the experimentally measured values. (3) Two neural network types will be designed and compared based on the performance of the network.

MATERIALS AND METHODS

Experimental

In a previous study (AlOmar et al. 2016b), a novel Pb²⁺ adsorbent was prepared based on pristine CNTs oxidized with KMnO₄, and then functionalized by the DES consisting of choline chloride: triethylene glycol at mixing ratio of 1:2. The preparation of the adsorbent was in two stages: the primary oxidation involved sonication of pristine CNTs (P-CNTs) with KMnO₄ for 2 h at 65 °C, and later the resulting oxidized CNTs (K-CNTs) were sonicated with DES for 5 h at 65 °C to produce KTEG-CNTs. The adsorbent was comprehensively characterized by conducting the Raman shift using Raman spectroscopy. The functional groups associated with the functionalization process were analysed using Fourier transform infrared spectroscopy. The surface charge, surface area and surface morphology were investigated by zeta potential, Brunauer-Emmett-Teller surface area field emission scanning electron microscopy and transmission electron microscopy respectively. The structural phase was also investigated by obtaining the X-ray diffraction profile. Moreover, a batch adsorption study was performed at ambient condition. Adsorbent dosage, initial concentration, pH value and contact time were taken as variables to the response of adsorption capacity of KTEG-CNTs. One hundred and fifty-eight points were taken to study the influence of each parameter and the interaction among them on the adsorption capacity. The range used for each parameter is listed in Table 1. The work flow is demonstrated in Figure 1.

Design of ANN structure

The Neural Network Toolbox R2014a of MATLAB was used to predict the adsorption capacity of functionalized CNT to adsorb Pb²⁺ from water solution. A total of 158 experimental datasets were prepared and used to create the ANN model. The experimental variables are initial concentration of Pb²⁺, adsorbent dosage, pH and contact time.

ANNs are a sophisticated statistical approach created to behave similar to the human nervous system by developing a logical model containing an interconnective neurons system in a computing network (Kurt & Kayfeci 2009; Hemmat Esfe et al. 2015). The neural network is used to resolve complicated test models such as pattern recognition, classification and estimation.

The supervised and the unsupervised are the two major types of ANNs that can be used in classification or regression. For the supervised model, the network is trained in order to adjust the optimum weight values between neurons, which makes it able to produce the desired output value(s) after taking different number of instructing data from the previous experimental examples. For the unsupervised model, there is no preferred design value during the introducing of the input to the structure. The supervised method was applied in this work.

The 158 datasets were allocated into training and testing sets, consisting of four inputs (adsorbent dosage, pH, contact time and initial concentration) and one output (adsorption capacity), and the testing files contained only the output parameter that were not operated for the training processes. The data were subdivided as defined percentages to prepare separate datasets for training and testing processes of the ANN model. Nevertheless, the division was organized on the basis that the training data form the major share of the latter. Subsequently, the data were switched within the spreadsheet and analysis was done to invalidate the presence of an existing combination of trend and the inherent characteristics within the data (Zhang & Govindaraju 2005; Sarangi & Bhattacharya 2005; Pakhale et al. 2015).

A total of two types of neural network were designed to analyse the feed-forward back-propagation (BP) and the layer recurrent (LR), to develop a neural network. The number of neuron(s), layer(s), training and testing sets and the type of transfer function need to be determined carefully.

Table 1 | The range of input and output parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adsorbent dosage (g)</td>
<td>5</td>
<td>20</td>
</tr>
<tr>
<td>Initial concentration of Pb²⁺ (ppm)</td>
<td>3</td>
<td>60</td>
</tr>
<tr>
<td>pH</td>
<td>2.7</td>
<td>7</td>
</tr>
<tr>
<td>Contact time (min)</td>
<td>3</td>
<td>900</td>
</tr>
<tr>
<td>Uptake capacity (mg/g) (output)</td>
<td>7.12</td>
<td>294.5</td>
</tr>
</tbody>
</table>
The suitable training algorithm can only be determined upon the identification of the complexity of the problem, the number of data points in the training set, the value of biases and weights in the network and the maximum error target. Six training functions, as presented in Table 2, were used and compared based on their performance to select the best suitable training function in both the feed-forward BP and the LR. For all training function, three hidden layers were selected for the feed-forward BP and five hidden layers for the layer recurrent neural network (LRNN); the number of hidden layers were selected by trial and error to design the best neural network structure.

Similarly, selection of the optimum number of hidden neurons to be used in the network is one of the major challenges for neural networks; using imprudent hidden neurons will lead to an overfitting problem. This will cause the network to over-estimate the complexity of the goal problem. It significantly affects the generalization performance, which causes a significant deviation in predictions. In the network optimization, two hidden neurons were used as the first hypothesis and then the number of neurons was increased up to 15 for both the feed-forward BP and the LR. Determination of the optimum number of hidden neurons to avoid the overfitting problem is critical in function estimation using a neural network.

The transfer function is one of the most important factors in the model creation; in this study three different transfer functions are used (TANSIG, PURELIN and LOGSIG) to choose the optimum one for the model.

**Feed-forward BP**

The learning algorithm is usually used, in the feed-forward BP ANN application, which used the BP system as the gradient descent technique to minimize network error. Each layer in the BP ANN has several neurons and each neuron transmits input values and processes to the next layer. As shown in Figure 2, the value of the input variable is multiplied by the connection weights \( w_{ij} \) which connects the input to the hidden layer.

The FFBP models consist of input layers, hidden layers and output layer in a multilayer neural network. The input layer consists of \( I \) nodes, the hidden layer contains \( J \) nodes and the output layer consists of the \( K \) nodes. Consequently, the network output \( z_k \) can be written as:

\[
z_k = f \left( b_{ok} + \sum_{j=1}^{J} b_{jk} \left( a_{oj} + \sum_{i=1}^{I} a_{ij} x_i \right) \right)
\]

The \( f \) is the transfer function in Equation (1) or the activation function, \( b_{jk} \) and \( a_{ij} \) (\( i = 1,2,3; \ j = 1,2,3; \ f = 1,2,3, k \)).

---

**Table 2** | The selected training functions

<table>
<thead>
<tr>
<th>Name of training function</th>
<th>Training function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quasi-Newton BP</td>
<td>Trainbfg</td>
</tr>
<tr>
<td>Bayesian regularization BP</td>
<td>Trainbr</td>
</tr>
<tr>
<td>Powell–Beale conjugate gradient BP</td>
<td>Traincgbr</td>
</tr>
<tr>
<td>Polak–Ribiere conjugate gradient BP</td>
<td>Traincgp</td>
</tr>
<tr>
<td>Fletcher–Reeves conjugate gradient BP</td>
<td>Traincgf</td>
</tr>
<tr>
<td>Levenberg–Marquardt BP</td>
<td>Trainlm</td>
</tr>
</tbody>
</table>
are the weight values, $x_i$ is the input number, and $a_{oi}$ and $b_{ok}$ are the deviation. The $f$ function in Equation (1) is a type of mapping rule to transfer the neurons from the weighted input to output; also it is a strategy type to introduce the non-linear into the feed-forward BP neural network (FBNN) (Kothari & Agyepong 1996).

There are plenty of FBNN transfer functions in the BP unit. The following transfer function selected principles used as a monotonous non-decreasing, differentiable and continuous function. In this work the most universal binary logistic sigmoid transfer function is used and it is written as follows:

$$f(x) = \frac{1}{1 + e^{-x}}$$ (2)

The optimal parameters can be selected by adjusting the weight values of the network as the FBNN owned by a supervisory learning algorithm technique (El-Shafie & El-Manadely 2011), and ‘optimal’ means the difference between the target values or actual values $t_k$ and the achieved minimum or target network output $z_k$, that is:

$$E = \frac{1}{2} \sum_{k=1}^{k} (z_k - t_k)^2$$ (3)

where $E$ refers to the error between the actual values $t_k$ and the network output $z_k$.

To produce an output vector $Y = (y_1, y_2, \ldots, y_p)$ for the ANN which is close as possible to target vector $T = (t_1, t_2, \ldots, t_p)$ a learning, also named as training, process is employed to find the optimum bias vectors $V$ and the optimum weight matrices $W$ that reduce the error, established in advance, which typically has the from:

$$E = \sum_{\rho} (y_i - t_i)^2$$ (4)

Here, $y_i = \text{ANN output}$; $t_i$ is the target output $T$; $\rho = \text{output value of nodes and} \ P = \text{training patterns number}$. The training is a process by which the connection weights of the ANN are adapted by a continuous process of simulation through the situation in which the network is embedded.

The input data are normalized in the range of (0 to 1) form to avoid the overfitting. It is realized that all the units at the same layer do not connect to each other in the adopted FBNN structure, and the connections between the developed layers can be expressed based on the weighted coefficient (El-Shafie et al. 2007).

The weighted signals and bias from the input neurons are summed by the hidden neurons and then projected through the transfer function. In the FFBP algorithm, the inputs are forwarded into the network until the end of the network, outputs are initiated and compared to the target value and the error is calculated (Kurt & Kayfeci 2009; Hemmat Esfe et al. 2015).

The BP learning is to establish the relationship between the target data and the input data that are usually assigned
with a random initial weight and later updating them by comparing the results between the actual and target values. In the diversity of research using neural computations, different transfer functions were used depending upon problem non-linearity and the complexity of data, in order to design the proper network.

**Layer recurrent**

The artificial recurrent neural networks (RNNs) perform different classes of computational modelling which is usually created by more or less detailed analogy with the biological brain module. In the LR, many abstract neurons, which also may be called processing elements or units, are interconnected by likewise synaptic links or connections, which enable activation to pass through the network.

The LRNN is almost the same as the forward neural network (FNN) except that both the hidden and output layer of the LRNN have recurrent connection associated with a tap delay, which is different to the FNN. In addition to the input space, the RNN works on an internal state space, which is a trace of what has already been processed by the network. Neurons in the RNN can be connected to any other neurons in the same or a previous layer. The RNNs consist of input layer, hidden layer, and output layer, with activation function, feedback connection weights, and interconnection weights. Figure 3 illustrates the flow of the input samples in the LRNN architecture.

The first hidden layer of the LRNN is connected to the inputs and the following layer, which assembles the network output and has a connection from the previous layer. The input weights of the hidden layer come from the input samples, and the following layer has a weight that comes from the preceding hidden layer. Although the hidden layer does not directly affect the external environment, it has great influence on the following layer or the output layer of the ANN.

The LRNN is categorized by the appearance of a backward connection in the hidden and output layers providing backward connection initiated from each output of hidden layer connected to one of the weights of the input layer by the context unit. Moreover, the backward connection from the output of the hidden layer, as presented in Figure 3, is occupied by the real output during the training processes of the LRNN. The number of hidden layers, number of neurons and the function of transfer types are the most important network parameters when considering the network structure.

**Evaluation indicators for simulation models**

Two competing neural networks have been developed, the feed-forward BP and LR for modelling, in this study. The total of 158 experimental data were divided into two subsets of testing and training (Desai et al. 2008; Lee et al. 2011) for developing the ANN model. The experimental variables are Pb²⁺ concentration, adsorbent dosage, pH, and contact time. The assessment of multicriteria was carried out. Therefore, the FNN and RNN model performance was determined by comparing the actual data and the simulated data. The simulation behaviour of each model was evaluated by using the root mean square error (RMSE), relative error (RE), mean square error (MSE), relative root mean square error (RRMSE) and the mean absolute percentage error.
Formulas for calculating MSE, RMSE, RE, RRMSE, and MAE are given below:

\[
\text{MSE} = \frac{1}{n} \sum_{t=1}^{n} (D_{a(t)} - D_{f(t)})^2
\]

(5)

\[
\text{RMSE} = \left[ \frac{1}{n} \sum_{t=1}^{n} (D_{a(t)} - D_{f(t)})^2 \right]^{\frac{1}{2}}
\]

(6)

\[
\text{RRMSE} = \left[ \frac{1}{n} \sum_{t=1}^{n} \left( \frac{D_{a(t)} - D_{f(t)}}{D_{a(t)}} \right)^2 \right]^{\frac{1}{2}}
\]

(7)

\[
\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{D_{a(t)} - D_{f(t)}}{D_{a(t)}} \right| \times 100
\]

(8)

\[
R^2 = 1 - \frac{SS_{res}}{SS_{tot}}
\]

(9)

\[
\text{RE} = \frac{D_{a(t)} - D_{f(t)}}{D_{a(t)}} \times 100
\]

(10)

where

\[D_{a(t)} = \text{the actual value}\]

\[D_{f(t)} = \text{the simulated value}\]

\[SS_{res} = \text{the regression sum of squares}\]

\[SS_{tot} = \text{the sum of squares of residuals}\]

Generally, MSE, RMSE, RE, MAPE and RRMSE equations were based on the obtained result by comparing the evaluated error of the actual and simulated model. The model with the smallest error is considered to be the best.

### RESULTS AND DISCUSSION

In this section, the used methods to select the optimal neuron number, training function, performance of the selected model, model performance evaluation and the RE between the predicted and the actual results are discussed.

#### Model performance evaluation

The functioning of each model was assessed by using the MSE, RMSE, RRMSE and MAPE. The values of each simulated method are tabulated in Table 3.

<table>
<thead>
<tr>
<th>Evaluation indicators</th>
<th>FBNN</th>
<th>LRNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>$1.66 \times 10^{-4}$</td>
<td>$7.2 \times 10^{-4}$</td>
</tr>
<tr>
<td>RMSE</td>
<td>$1.28 \times 10^{-2}$</td>
<td>$2.68 \times 10^{-2}$</td>
</tr>
<tr>
<td>RRMSE</td>
<td>$5.76 \times 10^{-2}$</td>
<td>$7.46 \times 10^{-2}$</td>
</tr>
<tr>
<td>MAPE</td>
<td>4.101</td>
<td>5.605</td>
</tr>
</tbody>
</table>

It is observed that the MSE of the FBNN model is $1.66 \times 10^{-4}$, which is a practical value and reflects the efficacy and higher accuracy in comparison to the LRNN model to simulate the relationship between initial concentration of Pb$^{2+}$, adsorbent dosage, pH and contact time to analyse the adsorbent capacity of KTEG-CNTs.

A simulation model is categorised to be reasonable and accurate if the MAPE value is below 30% and 5% respectively. Based on this acceptance criteria, the MAPE value for FBNN of 4.10% is considered as accurate, whereas the MAPE value of 5.60% for LRNN does not fall in the range of accuracy.

It is evident that the FBNN provides a better result in comparison to LRNN. Furthermore, the FBNN model resulted in RMSE and RRMSE values of $1.28 \times 10^{-2}$ and $5.76 \times 10^{-2}$ respectively, which is less than the results obtained in the LRNN model. The probability of error in the simulated value is low with the result of RMSE and RRMSE getting closer to zero. Therefore, these criteria further confirm that the FBNN method provided higher degree of accuracy in comparison to the LRNN method.

#### Training and testing dataset

This section discusses the part of a typical multilayer network workflow. In common practice, the data are first partitioned into two sets. The first set is termed as the training set, which is used to record the gradient and modify the network weights and biases. The testing set is the second subset, which is not applied during the training but is functional to compare various models and plot the errors of test sets. Generally, each BP training group starts with several initial weights and biases, and various divisions of data into training and testing sets. These different conditions can lead to varied solutions for the same problem. In this study, different training and testing percentages were used to find the optimal set with minimum error; the used sets are presented in Table 4.
At 70% training and 30% testing the MSE for the FBNN is $2.44 \times 10^{-2}$ and for the LRNN is $1.08 \times 10^{-1}$; when increasing the percentage of the training set to 75% with 25% for the test set, the MSE decreased to $9.73 \times 10^{-3}$ for the FBNN and $9.41 \times 10^{-2}$ for the LRNN. Furthermore, when the percentage of training was increased to 80% with a decrease in the percentage of the test set to 20%, there was an improvement in the result of the MSE, which was $5.87 \times 10^{-3}$ for FBNN and $2.96 \times 10^{-2}$ for the LRNN. The minimum MSE for the FBNN and LRNN was at 85% for the training and 15% for the test. The results demonstrate that the FBNN model is more accurate than LRNN as the values of the MSE for all the sets for the FBNN were lower than the MSE for LRNN.

### Neuron number optimization

The best structure of the ANN model and its specification difference are determined with reference to the smallest value of MSE of the test dataset. When increasing the number of the neurons, the network generates different MSE values for the testing dataset. Figure 4 illustrates the relationship between the number of neurons in each hidden layer and the MSE obtained.

For the feed-forward BP, three hidden layers were used. The MSE value of the network result is higher for the two (MSE $7.43 \times 10^{-2}$) and three ($7.60 \times 10^{-2}$) hidden layer neurons than those with four (MSE $6.51 \times 10^{-2}$), five (MSE $5.03 \times 10^{-2}$), six (MSE $4.25 \times 10^{-2}$) and seven (MSE $3.54 \times 10^{-2}$). The value of the MSE reduced significantly from $3.54 \times 10^{-2}$ to $1.48 \times 10^{-2}$ with the application of eight hidden neurons and continued to decrease with a subsequent rise in the neuron number from eight to ten. Therefore, the neural network consisting of 10 hidden neurons with MSE value of $1.66 \times 10^{-4}$ was selected as the best case based on the MSE value.

### Table 4 | The training and testing sets

<table>
<thead>
<tr>
<th>Training %</th>
<th>Testing %</th>
<th>MSE FBNN</th>
<th>MSE LRNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>30</td>
<td>$2.44 \times 10^{-2}$</td>
<td>$1.08 \times 10^{-1}$</td>
</tr>
<tr>
<td>75</td>
<td>25</td>
<td>$9.73 \times 10^{-3}$</td>
<td>$9.41 \times 10^{-2}$</td>
</tr>
<tr>
<td>80</td>
<td>20</td>
<td>$5.87 \times 10^{-3}$</td>
<td>$2.96 \times 10^{-2}$</td>
</tr>
<tr>
<td>85</td>
<td>15</td>
<td>$1.66 \times 10^{-4}$</td>
<td>$7.22 \times 10^{-4}$</td>
</tr>
<tr>
<td>90</td>
<td>10</td>
<td>$6.67 \times 10^{-2}$</td>
<td>$8.45 \times 10^{-2}$</td>
</tr>
</tbody>
</table>

**Figure 4** | The neuron number at each hidden layer with the MSE value.
Furthermore, when the neuron number increased to 13, the MSE value displayed a slight increment from $1.66 \times 10^{-4}$ to $4.98 \times 10^{-3}$. Subsequently, with further addition in the neuron number from 13 to 1, the result of the MSE is sharply increased. The increment might be assigned to the MSE characteristics input vector used and performance index used in this work.

For LR as shown in Figure 4, the number of hidden neurons was tested on five hidden layers as the best for the structure of the network. Two hidden neurons were used at the first try and showed $9.01 \times 10^{-2}$ MSE, while for three neurons the MSE decreased to $8.04 \times 10^{-2}$ and, for four the MSE then increased to $8.27 \times 10^{-2}$. When using five neurons there is a decline in the MSE to $7.05 \times 10^{-2}$; however, increasing the neurons to six, the MSE showed a slight decrease to $3.83 \times 10^{-2}$. The number of hidden neurons was later increased to seven and eight to improve the stabilization of the network, and the MSE decreased to $2.27 \times 10^{-2}$ and $2.06 \times 10^{-2}$ respectively; with the increasing of neurons to nine the MSE reached $1.94 \times 10^{-3}$. Furthermore, with 10 neurons the MSE decreased to $7.22 \times 10^{-4}$, which shows the best stabilization of the network. With 11 and 12 neurons, the MSE displayed a higher value of $1.96 \times 10^{-3}$ and $4.68 \times 10^{-3}$ respectively. Finally, with 13, 14 and 15 neurons, the MSE rises to $1.04 \times 10^{-2}$, $9.84 \times 10^{-3}$ and $1.86 \times 10^{-2}$. This confirm that using 10 neurons at each hidden layer shows the best performance of the network.

### Selection of the training function for FBNN and LRNN

It is laborious to find the fastest training algorithm for a given problem due to the complexity of the problem, which depends on various factors. This section discusses the comparison of the various training algorithms. Networks were trained on six different training functions and identified based on the $R^2$ value and the MSE. Six training functions shown in Table 5 were used and compared to select the best suitable training function in both the feed-forward BP and the LR.

For the feed-forward BP, the comparison study showed that the Bayesian regularization BP (trainbr) had resulted in the smallest value of MSE in comparison to different sets of algorithms such as the Levenberg-Marquardt BP (trainlm) algorithm. As shown in Table 5, the smallest MSE was about $1.66 \times 10^{-4}$, with $R^2$ of 0.9956 for trainbr function presented in Figure 5, which reflects a great performance of the network. This was followed by the trainlm with MSE of $6.79 \times 10^{-4}$. However, both trainrb and trainlm show a better behaviour than the other algorithms such as trainbfg, traincgb, traincgf and traincgp.

The structure and the combinatorial characteristic of the test data influence the results' optimality initiated by some BP algorithms. Therefore, the problem complexity was solved by the results of several analyses of training algorithms used for the comparison.

For the LR the first benchmark is using trainbfg as training function with tansig transfer function, which results in MSE of $1.28 \times 10^{-2}$, while for the trainbr training function the MSE is decreased to $7.22 \times 10^{-4}$, which shows a great performance of the network. However, by using a different training function such as traincgp and traincgf, the MSE is $4.44 \times 10^{-3}$ and $2.89 \times 10^{-2}$, which are greater than the MSE of trainbr training function, while the MSE for the traincgp is $1.32 \times 10^{-2}$. The performance of the trainlm also showed a great value of

| Table 5 | The training function, $R^2$ and MSE |
|-----------------|---------------------|---------------------|
| Training function | Feed-forward BP | LR |
|                  | MSE     | $R^2$ | MSE     | $R^2$ |
| Trainbfg         | $9.03 \times 10^{-3}$ | 0.7465 | $1.28 \times 10^{-2}$ | 0.856 |
| Trainbr          | $1.66 \times 10^{-4}$ | 0.9956 | $7.22 \times 10^{-4}$ | 0.990 |
| Traincgb         | $2.97 \times 10^{-3}$ | 0.9021 | $4.44 \times 10^{-3}$ | 0.956 |
| Traincgf         | $4.63 \times 10^{-2}$ | 0.4931 | $2.89 \times 10^{-2}$ | 0.663 |
| Traincgp         | $2.30 \times 10^{-2}$ | 0.6915 | $1.32 \times 10^{-2}$ | 0.891 |
| Trainlm          | $6.79 \times 10^{-4}$ | 0.9766 | $2.82 \times 10^{-3}$ | 0.968 |

![Figure 5](https://iwaponline.com/wst/article-pdf/76/9/2413/209046/wst076092413.pdf)
the MSE, $2.82 \times 10^{-3}$, which can be considered as one of the best suitable training functions for the network.

It can be seen that the trainbr training function is the most suitable for both feed-forward BP and LR networks.

Relative error indication

RE is one of the indications of error in the model prediction values comparing the predicted values to the actual values; measurements and calculation can be characterized with regard to their precision and accuracy. The term accuracy can be defined as how closely the predicted value matches the actual value, whereas precision refers to how closely values match each other.

The highest RE value is found to be 14.93\% for the FBNN model and 18.67\% for the LRNN model calculated by Equation (10), which are considered as acceptable values.

Based on the results shown in Figure 6, it is observed that the error for all the testing dataset is less than 14.93\% for the FBNN, whereas the maximum error for the LRNN is less than 18.67\%. This indicates that the FBNN model is more accurate than the LRNN model. This proves the effectiveness and reliability of the proposed approach to extract features from input data. The hybrid FBNN algorithm network model is able to provide a perfect prediction of KTEG-CNTs as Pb\(^{2+}\) absorber from water. The uncertainty in this work might come from the accuracy of the initial concentration of Pb\(^{2+}\), amount of adsorbent dosage, and the aqueous solution pH adjustment, as the amount of the materials used is very small. Also, the humidity and the temperature of the room are not considered in this study, which might affect the accuracy of the results.

The effect of pH on the adsorption capacity

The pH is one of the very important factors which can affect the quantity and the form of Pb\(^{2+}\) in water, the interactions between the adsorbent and Pb\(^{2+}\), and the quantity and the form of the adsorbent surface sites (Chen & Wang 2007).

The pH effect on adsorption capacity was studied by mixing 12.5 mg adsorbent dosage with 5 mg/l concentration at 15 min contact time with pH values from 1 to 10. The experiment results show that the pH of the solution was an important factor affecting the adsorption efficiency. The pH increment led to significant increase in adsorption capacity until pH 5.0, then the adsorption capacity became steady with increasing pH. It is well known that at pH greater than 7.0, the dominant species of Pb\(^{2+}\) are Pb(OH)\(^{+}\) and Pb(OH)\(_2\). This complexation may occur due to the extensive presence of OH\(^{-}\) at this pH level, which resulted in a precipitation form (Gupta et al. 2011). In addition, the decreasing of H\(^{+}\) plays a significant role in the mechanism of Pb\(^{2+}\) adsorption due to the decreasing of competition on the active sites of the adsorbent. The agreement of the ANN model predictions in relation to pH is presented in Figure 7. From Figure 7, it can be noticed that the ANN model outputs showed almost the same behaviour as the experimental data, which proves
that the ANN model can predict the adsorption capacity for Pb$^{2+}$ removal from water satisfactorily.

The effect of adsorbent dosage on the adsorption capacity

The adsorbent dosage is one of the important parameters involved in the adsorption process. Adsorbent dosage effect on the Pb$^{2+}$ removal is examined by keeping the other factors constant, at time 10 minutes, pH 5.0, and 5 mg/l of Pb$^{2+}$ initial concentration. The Pb$^{2+}$ removal capacity is decreased from 47.46 mg/g to 19.704 mg/g after increasing the adsorbent dosage from 5 mg to 12.5 mg and is decreased from 19.704 mg/g to 12.392 mg/g after increasing the adsorbent dosage from 12.5 mg to 20 mg. The decrease in the uptake capacity with increasing adsorbent dosage might be attributed to the increase in the adsorbent surface area, leading to an increase of active sites (Kumar & Phanikumar 2013; Das et al. 2014). The ANN technique is used for the modelling and prediction of the obtained data from the experimental work; the prediction results show a good agreement with the experimental result trend. The ANN outputs and the experimental results as a function of adsorbent dosage versus the uptake capacity are presented in Figure 8.

The effect of initial concentration

The initial concentration being one of the factors involved in this work, the effect of initial concentration of Pb$^{2+}$ ions was studied by changing the initial concentration from 5 mg/l to 60 mg/l. The other factors were fixed at time 60 min, pH 2.7 and adsorbent dosage 5 mg. From the presented results in Figure 8, it can be seen that the uptake capacity of Pb$^{2+}$ ions at 5 mg/l concentration was 47.7 mg/g, whereas after increasing the Pb$^{2+}$ concentration to 60 mg/l the uptake capacity increased to 225.05 mg/g. This might be attributed to the increase in the driving force of the mass transfer which led to an increase in the uptake capacity of Pb$^{2+}$ ions from water solution. At low concentration, the Pb$^{2+}$ ions interact at the adsorbent active site, whereas at higher Pb$^{2+}$ concentration, the adsorbent active site will be saturated and the removal percentage will be lower (Hamza et al. 2013). The obtained data from the experimental work were trained and predicted by using the ANN modelling techniques. The ANN model prediction was found to be satisfactory for the experimental data observation. The experimental and predicted output of the ANN are presented in Figure 9.

The effect of contact time

The contact time being one of the involved parameters in the experimental work, the contact time effect was studied with varying the contact time from 5 min to 120 min. The other involved parameters were kept constant: initial concentration...
5 mg/l, adsorbent dosage 5 mg and pH 5. The uptake capacity at 5 min is 31.98 mg/g, whereas at 80 min the uptake capacity reached 48.1 mg/g; the maximum uptake capacity at the equilibrium time is 49.3 mg/g. It is clear from the results presented in Figure 10 that 90% removal occurred at 80 min (Kabbashi et al. 2013). This is due to the availability of vacant sites at the adsorbent surface; hence, the adsorption rate will be higher at the beginning of the reaction. The ANN model was used for the modelling and prediction of the obtained results; it can be seen from Figure 10 that the ANN model predicted the experimental data satisfactorily.

CONCLUSION

An ANN has been successfully used to predict the removal of Pb$^{2+}$ from aqueous solution by using DES-functionalized CNTs. The tansig transfer function was used in this study for modelling. Two different neural network types were developed in this work, the (BP ANN) and (LR ANN). Both models were created with the same aim function and restriction with the same structure of dataset. The optimal topology of the ANN was obtained during the training phase using the trainbr algorithm. The results showed that the network with 10 neurons in each hidden layer, with three hidden layers, showed the best performance. Moreover, the supervised type (multilayer feed-forward neural network) was selected in this study.

The MSE of the BP-ANN model prediction is $1.66 \times 10^{-4}$ with the $R^2$ of 0.9956. The favourable features of the ANN modelling technique included its generalization, efficiency and simplicity, which make it a preferable choice for the modelling of complex systems, such as removal of Pb$^{2+}$ ions from water processes.

ACKNOWLEDGEMENT

The authors express their thanks to the University of Malaya for funding this research, UMRG (RP017B-13AET).

REFERENCES


Khajeh, M. & Moghaddam, M. G. 2011 Comparison of response surface methodology and artificial neural network in predicting the microwave-assisted extraction procedure to determine zinc in fish muscles. *Food and Nutrition Sciences* **2** (8), 803–808.


Majumdar, S. S., Das, S. K., Chakravarty, R., Saha, T.,
Bandyopadhyay, T. S. & Guha, A. K. 2010 A study on lead
adsorption by Mucor rouxii biomass. Desalination 251 (1–3),
96–102.
Mandal, S., Sivaprasad, P. V., Venugopal, S. & Murthy, K. P. N. 2009
Artificial neural network modelling to evaluate and predict the
deformation behavior of stainless steel type AISI 304L during
Mjalli, F. S., Al-Asheh, S. & Alfadala, H. E. 2010 Use of artificial
neural network black-box modelling for the prediction of
wastewater treatment plants performance. Journal of
Oubagaranadin, J. U. K. & Murthy, Z. V. P. 2013 Isotherm
modelling and batch adsorber design for the adsorption of
Cu(II) on a clay containing montmorillonite. Applied Clay
Science 50 (3), 409–413.
Pakhale, G. K., Nale, J. P., Temesgen, W. B. & Mulunch, W. D.
2015 Modelling reference evapotranspiration using artificial
Pimentel, P. M., González, G., Melo, M. F. A., Melo, D. M. A.,
from aqueous solution by retorted shale. Separation and
Purification Technology 56 (3), 348–353.
Renjith, A., Roy, A. & Lakshminarayanan, V. 2014 In situ
fabrication of electrochemically grown mesoporous metallic
thin films by anodic dissolution in deep eutectic solvents.
Abd Karim, K. J. 2016 Removal of lead ions from aqueous
solutions using sodium alginate-graft-poly(methyl
methacrylate) beads. Desalination and Water Treatment
57 (33), 15353–15361.
Sarangi, A. & Bhattacharya, A. K. 2005 Comparison of artificial
neural network and regression models for sediment loss
prediction from Banha watershed in India. Agricultural
Water Management 78 (3), 195–208.
Wang, J. & Chen, C. 2006 Biosorption of heavy metals by
Saccharomyces cerevisiae: a review. Biotechnology Advances
Wu, G. & Xu, Z. 2011 Prediction of algal blooming using EFDC
model: case study in the Daoxiang Lake. Ecological
Modelling 222 (6), 1245–1252.
Xu, K., Wang, Y., Ding, X., Huang, Y., Li, N. & Wen, Q. 2016
Magnetic solid-phase extraction of protein with deep eutectic
solvent immobilized magnetic graphene oxide nanoparticles.
Talanta 148, 153–162.
Yasin, Y., Ahmad, F. B. H., Ghaffari-Moghaddam, M. & Khajeh,
M. 2014 Application of a hybrid artificial neural network–
genetic algorithm approach to optimize the lead ions removal
from aqueous solutions using intercalated tartrate-Mg–Al
layered double hydroxides. Environmental Nanotechnology,
Monitoring & Management 1–2, 2–7.
Yetilmazsoy, K. & Demirel, S. 2008 Artificial neural network
(ANN) approach for modelling of Pb(II) adsorption from
Zhang, B. & Govindaraju, R. S. 2003 Geomorphology-based
artificial neural networks (GANNs) for estimation of direct
runoff over watersheds. Journal of Hydrology 273 (1–4),
18–34.
Zheng, Y., Ye, L., Yan, L. & Gao, Y. 2014 The electrochemical
behavior and determination of quercetin in choline chloride/urea deep eutectic solvent electrolyte based on abrasively
immobilized multi-wall carbon nanotubes modified
electrode. International Journal of Electrochemical Science
9, 238–248.

First received 29 March 2017; accepted in revised form 19 June 2017. Available online 8 July 2017