Desalination 308 (2013) 102–110

Contents lists available at SciVerse ScienceDirect

Desalination

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

Artificial neural network model for desalination by sweeping gas membrane distillation

M. Khayet [*], C. Cojocaru

Department of Applied Physics I, Faculty of Physics, University Complutense of Madrid, Avda. Complutense s/n 28040, Madrid, Spain

ARTICLE INFO

Article history:
Received 10 February 2012
Received in revised form 22 June 2012
Accepted 24 June 2012
Available online 17 July 2012

Keywords:
Sweeping gas membrane distillation
Desalination
Artificial neural network
Optimization

ABSTRACT

Sweeping gas membrane distillation process (SGMD) has been used for desalination and its performance index, defined as the product of the distillate flux and the salt rejection factor, has been modeled using artificial neural network (ANN) methodology. A feed-forward ANN has been developed for prediction of the performance index based on a set of 53 different experimental SGMD tests. A feed solution of 30 g/L sodium chloride was used in all experiments and the salt rejection factors were found to be greater than 99.4%. The individual and interaction effects of the input variables, namely the feed inlet temperature, the feed flow rate or the feed circulation velocity, and the air flow rate or the air circulation velocity, on the SGMD performance index have been investigated. The optimum point was determined by means of Monte Carlo simulation. The obtained optimal conditions were a feed inlet temperature of 69 °C, an air flow rate of 34.5 L/min (i.e. 2.02 m/s air circulation velocity) and a feed flow rate of 160 L/h (i.e. 0.155 m/s liquid circulation velocity). Under these operating conditions a performance index of 1.493×10^{-3} kg/m^2.s has been achieved experimentally being the maximal SGMD performance index obtained inside the region of experimentation.

1. Introduction

During the last years artificial neural network (ANN) modeling was used frequently in various separation and technological applications, mainly due to their powerfulness for solving complex multiple regression problems [1]. The ability of ANN for mapping non-linear relationships between the inputs and outputs of a system or a process has extended the field of applications of this modeling tool.

Several studies have been considered for the application of ANN in modeling of various processes in membrane technology [1–10]. A feed-forward ANN was developed by Abbas and Al-Bastaki [1] for the prediction of a reverse osmosis (RO) performance using a FilmTec SW30 membrane for desalination of various salt solutions ranging between brackish water and seawater salinities. Purkait and collaborators [2] employed two ANN models for prediction of the permeate flux when treating a leather plant effluent by nanofiltration (NF) process followed by reverse osmosis (RO). Zhao and co-workers [3] have performed a comparison between a modified solution diffusion model and ANN to predict RO/NF water quality effluent. Yangali-Quintanilla et al. [4] also used ANN to predict the rejection of neutral organic compounds by NF and RO using polyamide membranes. Libotean and collaborators [5] proposed an ANN with back-propagation to forecast the performance of an RO plant and for potential use in operational diagnostics. Al-Abri and Hilal [6] developed an ANN model for simulation of a combined humic substance coagulation and membrane filtration. Huaiqun and Kim [7] used a radial basis function neural network approach for prediction of permeate flux decline in crossflow membrane filtration of a colloidal suspension. Darwish, et al. [8] used ANN for simulation of NF of sodium chloride and magnesium chloride solutions. Sahoo and Ray [9] worked on the prediction of permeate flux decline in crossflow membranes using ANN and genetic algorithms. Mhurchú and Foley [10] employed the dead-end filtration of yeast suspensions by correlating specific resistance and permeate flux data using artificial neural networks.

In our previous paper [11] we compared the ANN model with response surface model (RSM) in terms of prediction and optimization of desalination by RO process. The ANN model was found to be more adequate in prediction of the RO performance index than the RSM empirical model in a wide range of salt concentrations.

It is worth quoting that the application of neural network modeling in membrane distillation (MD) is very limited. MD is a thermally driven process mainly dealing with water vapor transport through non-wetted porous hydrophobic membranes [12]. This process demonstrates to be successfully applied in desalination of seawater or brackish waters. Various MD configurations can be considered to apply the driving force, which is the transmembrane vapor pressure, i.e. direct contact membrane distillation (DCMD), air gap membrane distillation (AGMD), vacuum membrane distillation (VMD) and sweeping gas membrane distillation (SGMD) [12].

Recently we have reported on the development and application of an ANN model to predict the AGMD desalination performance index [13]. The ANN model was used for optimization of the AGMD process and the following optimal conditions were obtained, an air gap

[*] Corresponding author. Tel.: +34 91 3945185; fax: +34 91 3945191.
E-mail address: khayetm@fis.ucm.es (M. Khayet).

0011-9164/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.desal.2012.06.023

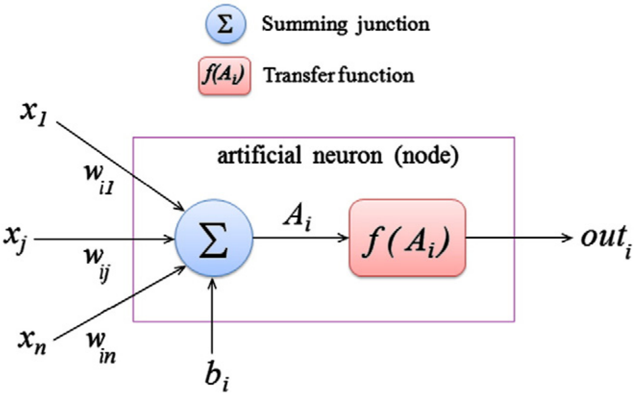

Fig. 1. The structure of an artificial neuron (node).

thickness of 3.0 mm, a condensation temperature of 13.9 °C, a feed inlet temperature of 71 °C and a feed flow rate of 205 L/h. Under these conditions a maximum experimental performance index of 51.075 kg/m^2.h was achieved experimentally.

In this study, attempts are made to develop an ANN model for the prediction of the SGMD performance index. To the best of our knowledge, there is no ANN model reported for SGMD configuration, so far. This MD configuration, which is rarely studied (i.e. about 4.5% of the papers dealing with MD published up to December 2010 in refereed journals [12]), consists of a gas that sweeps the permeate side of the membrane carrying the evaporated molecules away from the permeate side of the membrane pores. Condensation of the vapor takes place outside the membrane module. Therefore, external condensers are required to collect the vapor in the permeate side stream, complicating in this way the system design and increasing its cost. Similar to the other MD configurations, the feed liquid solution is maintained at a higher temperature than the temperature in the permeate side [12–15].

2. ANN theoretical

ANN is a computational model that was inspired by the structure and functional aspects of biological neural networks. The artificial network is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. It consists of an interconnected group of artificial neurons that processes information using a connectionist approach [16].

Each artificial neuron is a unitary computational processor, which has a summing junction operator and a transfer function [17,18]. The connections between inputs, neurons and outputs consist of wei*ghts* and *biases*, which are considered parameters of the neural network. The summing junction operator of a single neuron (i) summarizes the weights and bias into a net input A_i known as argument. The transfer function of a single neuron converts the net argument A_i into the scalar output according to the scheme shown in Fig. 1 where w_{ij} is the connection weight, x_j is the input variable, i and j $\left(i=\overline{1,m};\ j=\overline{1,n}\right)$ are the integer indexes, m is the number of artificial neurons, n is the number of input variables and b_i is the bias of the single artificial neuron (i). The type of transfer functions frequently employed for solving multiple regression problems is summarized for a single artificial neuron in Table 1 together with the explicit relationship for the summing junction operator.

The most used transfer functions to solve multiple regression problems are the linear transfer function (purelin), log-sigmoid transfer function (logsig) and hyperbolic tangent sigmoid transfer function (tansig) [17–22].

The way in which the inputs, neurons and outputs are connected is known as *architecture* or *topology* of the neural network. Usually, the neurons of a network are grouped into several layers such as *hidden* and *output layers* consisting of, hidden and output neurons, respectively. In addition, the inputs can be considered as an additional layer. The most applied neural network topology for solving the multiple regression problems (i.e. function approximations) is the multi-layer feed-forward structure also known as multi-layer perceptron MLP [17–19,21]. This topology is detailed elsewhere [19]. The most important phase for building ANN model is the *training* of the network. During the training process the weights and biases of a feed-forward neural network are adjusted systematically in order to minimize the residual error between network outputs (predictions) and targets (experimental data) [11,23,24]. There is a variety of training algorithms. The most used classes of training methods for feed-forward neural networks are the back-propagation (BP) algorithms [11,17,18,23,24]. Training of ANN using BP algorithm is an iterative optimization process applied for performance function minimization by adjusting the network weights and biases appropriately. The most employed performance function is the *mean-squared-error* (*MSE*). In the case of a single response (output neuron), *MSE* may be written as [13,21]:

$$MSE=\frac{1}{N}\sum_{q=1}^{N}\left(Y_q^{(\exp)}-\hat{Y}_q^{(pred)}\right)^2 \tag{1}$$

where $Y_q^{(\exp)}$ is the experimental response (target), $\hat{Y}_q^{(pred)}$ is the predicted response by ANN (network output), N is the number of experimental data points and q is the iteration index (positive integer number). There are many variations of BP algorithm. Most of them use the gradient descent method for iterative updating of weights and biases until the convergence is satisfied. Normally, the gradient descent updating of the network weights (w_{ij}) and biases (b_i) using a single iteration of BP algorithm can be written as [25]:

$$w_{ij}^{(k+1)}=w_{ij}^{(k)}-\eta\frac{\partial E^{(k)}}{\partial w_{ij}^{(k)}} \tag{2}$$

$$b_i^{(k+1)}=b_i^{(k)}-\eta\frac{\partial E^{(k)}}{\partial b_i^{(k)}} \tag{3}$$

where E is the error function (e.g. *MSE*), η is the learning rate and k is the integer index indicating the epoch (iteration) in learning phase. Once the ANN was trained the optimal weights and biases are saved and the neural network model can be used for simulation and optimization [11,23].

Table 1
Transfer functions of artificial neurons used for solving multiple regression problems.

Description	Notation	Summing junction (input transformation)	Transfer function equation	Output range
Linear transfer function	*purelin*	$A_i=\sum_{j=1}^{n}x_j\cdot w_{i,j}+b_i$	$f(A_i)=A_i$	$-\infty\le f(A_i)\le+\infty$
Log-sigmoid transfer function	*logsig*	$A_i=\sum_{j=1}^{n}x_j\cdot w_{i,j}+b_i$	$f(A_i)=\frac{1}{1+\exp(-A_i)}$	$0\le f(A_i)\le1$
Hyperbolic tangent sigmoid transfer function	*tansig*	$A_i=\sum_{j=1}^{n}x_j\cdot w_{i,j}+b_i$	$f(A_i)=\frac{1-\exp(-A_i)}{1+\exp(-A_i)}$	$-1\le f(A_i)\le+1$

3. Experimental

In all experiments a commercial porous hydrophobic membrane TF450 provided by Gelman company has been employed. This membrane is made of polytetrafluorethylene (PTFE) supported by a polypropylene (PP) net. Its main characteristics as specified by the manufacturer are the total thickness, PTFE/PP (178 μm), the mean pore size (0.45 μm), the fractional void volume or porosity (80%) and the liquid entry pressure of water (LEP_w, 137.8 kPa).

For preparation of the feed salt solution, sodium chloride NaCl (puriss) provided by Sigma-Aldrich was used. The electrical conductivity of the feed and distillate solutions was measured using a conductivimeter 712 Ω Metrohm. Then the salt concentration was determined from the electrical conductivity using a previous calibration. Finally, the salt rejection factor was determined based on salt concentrations in feed and distillate solutions.

The experimental set-up of SGMD configuration used is shown in Fig. 2. A plate and frame membrane module (1) has been used consisting of two chambers, one for the feed and the other for distillate. The chambers are made from silicone separators and placed between two acrylic manifolds. The polymeric membrane (2) is placed between the two chambers and has an effective area of 5.53×10^{-3} m^2. Both chambers have a thickness of 4.3 mm (length = 86.3 mm; width = 6.83 mm). The flow paths within the sweeping-gas module were of counter-flow type. The temperatures of the liquid feed and the sweeping air were measured at the inlet and outlet of the membrane module by Pt100 sensors connected to a digital multimeter FLUKE HYDRA. To avoid membrane pore wetting, the pressure difference through the membrane was controlled with two manometers placed at the inlets of the module. The liquid was circulated from the feed reservoir (3) through the system by a circulation pump MasterFlex 7529-20 (4). The feed flow rate was measured with a flowmeter Tecfluid (5) and the temperature of the circulating liquid was controlled by a heat exchanger (6) connected to a thermostat (Techne TU-16D). The sweeping air flow rate was maintained by means of a compressor (Fiac 1001, PS 11 bar) (7). A valve (8) was employed to adjust the flow rate that is measured with a flowmeter Gilmont 51501 (9). Two humidifiers connected in series (10a and 10b) were used for water saturation of the air. The heat exchanger (11) connected to a thermostat (Techne RB-12A) was used to adjust the temperature of the sweeping air. A trap for water retention (12) was connected before air could enter the module. A condenser (13) controlled by a thermostat (PolyScience Recirculator) was used for condensation of the produced water vapor. The distillate was collected in a graduated cylinder (14). The distillate flux was determined experimentally by weighing the obtained distillate during a predetermined time using a digital balance (ANDGF-1200 with precision of 0.01 g) (15). To minimize the heat loss, the containers and pipes of the SGMD set-up were thermally insulated.

The desalination performance index (Y) of the SGMD process has been determined experimentally using the following relationship:

$$Y = J \times \frac{RF}{100} \tag{4}$$

where J denotes the distillate flux (kg m^{-2} s^{-1}) and RF the salt rejection factor (%).

4. Results and discussions

The experimental data used to construct the ANN model for SGMD process are summarized in Table 2. A total number of 53 different experiments were employed. It was found that the salt rejection factor was varied between 99.64% and 99.97% and most of the values are near 99.9%. Therefore, the ANN model was developed for the SGMD performance index (Y). About 80% from all these available data were used for training and 20% for validation and test. As inputs of the neural network three variables were considered, i.e. the feed inlet temperature ($T_{f,in}$), the feed liquid flow rate or liquid circulation velocity (U_w) and the air flow rate or air circulation velocity (U_a). In all experiments the feed liquid concentration (NaCl) and the temperature of the sweeping air ($T_{a,in}$) at the module inlets were maintained constants, 30 g/L and 20 °C, respectively. The desalination performance index (Y) was considered as response (target). To avoid overfitting, both input variables and response were normalized before training. The input variables were normalized so that they can vary in the range [0–1] according to the following relationship [13]:

$$x_j = \frac{\left(z_j - z_j^{min}\right)}{\left(z_j^{max} - z_j^{min}\right)} \tag{5}$$

where x_j refers to the normalized input variable, while z_j, z_j^{min} and z_j^{max} are the actual, minimum and maximum values of the input variable.

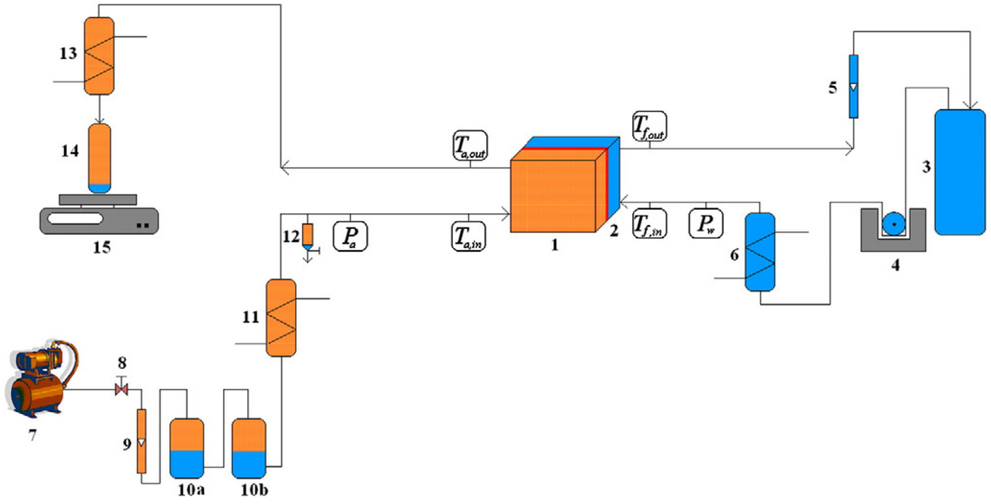

Fig. 2. SGMD experimental set-up. (1) SGMD module; (2) flat sheet membrane; (3) feed container; (4) peristaltic pump; (5) water flowmeter; (6) heat exchanger; (7) air compressor; (8) valve; (9) air flowmeter; (10a,10b) humidifiers; (11) heat exchanger for cooling or heating the air at desired temperature; (12) trap for water retention; (13) condenser; (14) permeate container; (15) balance.

Table 2
Experimental data used for the construction of the ANN model of SGMD process.

Run		Input variables			Responses			
Trial	Type	$T_{f,in}$ (°C)	U_a (m/s)	U_w (m/s)	J (kg m^{-2} s^{-1})×10^{-3}	RF (%)	Y (kg m^{-2} s^{-1})×10^{-3}	Y_n
1	Training	68	1.932	0.200	1.106	99.962	1.1056	0.800
2	Training	54	1.932	0.200	0.636	99.876	0.6352	0.438
3	Training	68	0.966	0.200	0.650	99.881	0.6492	0.449
4	Training	54	0.966	0.200	0.459	99.919	0.4586	0.302
5	Training	68	1.932	0.140	0.969	99.924	0.9683	0.694
6	Training	54	1.932	0.140	0.740	99.784	0.7384	0.517
7	Training	68	0.966	0.140	0.582	99.942	0.5817	0.397
8	Training	54	0.966	0.140	0.326	99.927	0.3258	0.200
9	Training	70	1.449	0.170	1.016	99.709	1.0130	0.729
10	Training	52	1.449	0.170	0.499	99.919	0.4986	0.333
11	Training	61	2.028	0.170	0.841	99.929	0.8404	0.596
12	Training	61	0.869	0.170	0.499	99.945	0.4987	0.333
13	Training	61	1.449	0.206	0.989	99.452	0.9836	0.706
14	Training	61	1.449	0.134	0.634	99.920	0.6335	0.437
15	Training	61	1.449	0.170	0.670	99.903	0.6694	0.464
16	Training	68	1.449	0.170	0.919	99.963	0.9187	0.656
17	Training	54	1.449	0.170	0.542	99.924	0.5416	0.366
18	Training	65	1.884	0.170	0.988	99.954	0.9875	0.709
19	Training	58	1.014	0.170	0.528	99.920	0.5276	0.355
20	Training	65	1.014	0.170	0.647	99.944	0.6466	0.447
21	Training	58	1.884	0.170	0.698	99.911	0.6974	0.486
22	Training	65	1.594	0.194	0.849	99.955	0.8486	0.602
23	Training	58	1.304	0.146	0.780	99.954	0.7796	0.549
24	Training	65	1.304	0.146	0.626	99.938	0.6256	0.431
25	Training	61	1.738	0.146	0.893	99.936	0.8924	0.636
26	Training	58	1.594	0.194	0.686	99.928	0.6855	0.477
27	Training	61	1.159	0.194	0.610	99.940	0.6096	0.418
28	Training	54	0.966	0.170	0.361	99.904	0.3607	0.227
29	Training	68	0.966	0.170	0.579	99.928	0.5786	0.395
30	Training	54	1.932	0.170	0.679	99.936	0.6786	0.471
31	Training	68	1.932	0.170	1.024	99.962	1.0236	0.737
32	Training	54	1.449	0.140	0.549	99.949	0.5487	0.372
33	Training	68	1.449	0.140	0.846	99.925	0.8454	0.600
34	Training	54	1.449	0.200	0.562	99.946	0.5617	0.382
35	Training	68	1.449	0.200	0.966	99.951	0.9655	0.692
36	Training	61	0.966	0.140	0.473	99.948	0.4728	0.313
37	Training	61	1.932	0.140	0.896	99.853	0.8947	0.638
38	Training	61	0.966	0.200	0.409	99.922	0.4087	0.264
39	Training	61	1.932	0.200	0.804	99.925	0.8034	0.567
40	Training	65	1.690	0.185	0.956	99.956	0.9556	0.685
41	Training	57	1.690	0.185	0.688	99.905	0.6873	0.478
42	Validation	65	1.207	0.185	0.704	99.812	0.7027	0.490
43	Validation	57	1.207	0.185	0.543	99.914	0.5425	0.367
44	Validation	65	1.690	0.155	0.873	99.955	0.8726	0.621
45	Validation	57	1.690	0.155	0.639	99.899	0.6384	0.441
46	Validation	65	1.207	0.155	0.600	99.860	0.5992	0.410
47	Validation	57	1.207	0.155	0.559	99.936	0.5586	0.379
48	Test	66	1.449	0.170	0.887	99.968	0.8867	0.632
49	Test	56	1.449	0.170	0.625	99.968	0.6248	0.430
50	Test	61	1.738	0.170	0.993	99.639	0.9894	0.711
51	Test	61	1.159	0.170	0.552	99.930	0.5516	0.374
52	Test	61	1.449	0.188	0.802	99.874	0.8010	0.566
53	Test	61	1.449	0.152	0.664	99.917	0.6634	0.460

The target Y was normalized in the range varying between 0.2 and 0.8 using the following equation [13,26]:

$$Y_n = (1-2\Delta)\frac{(Y-Y_{min})}{(Y_{max}-Y_{min})}+\Delta \qquad (6)$$

where Y_n, Y_{min} and Y_{max} are the normalized, minimum and maximum levels of the target Y and Δ is a quantity set as 0.2 in order to ensure the normalized interval [0.2–0.8]. Such an interval gives the network limited extrapolation capability for the extended range [0–1]. The values of the normalized target Y_n are listed in the last column of Table 2.

In order to build the ANN model our code was developed using MATLAB software and the standard functions included in the neural network toolbox. The normalized values of the inputs and target were used to feed and train the neural network. In this case, the Levenberg–Marquardt back-propagation algorithm (LM-BP) was employed [18]. In order to optimize the neural network architecture, the computations started using one neuron in the hidden layer as the initial guess. Afterwards, the number of neurons was increased and the performance function *MSE* was calculated as shown in Fig. 3. As can be seen, the increment of the number of neurons results in a minimal value of the error function *MSE* for a network topology with 9 neurons in the hidden layer. Further increase of the number of neurons leads to a slight increase of the *MSE*. Therefore, in this case the optimal architecture of the ANN model includes three inputs (i.e. variables), one hidden layer with 9 neurons and one output layer with a single neuron (Fig. 4). Thus, the notation of the developed feed-forward neural network may be written as MLP (3:9:1). Note that, all neurons of the hidden layer have the log-sigmoid transfer function (*logsig*) while the single neuron from the output layer has the linear (*purelin*) transfer function. As it can be observed in Fig. 4,

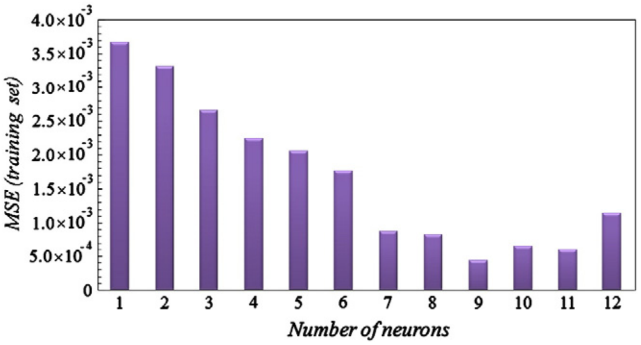

Fig. 3. Dependence between the network performance (*MSE*) and the numbers of neurons in hidden layer using LM-BP algorithm.

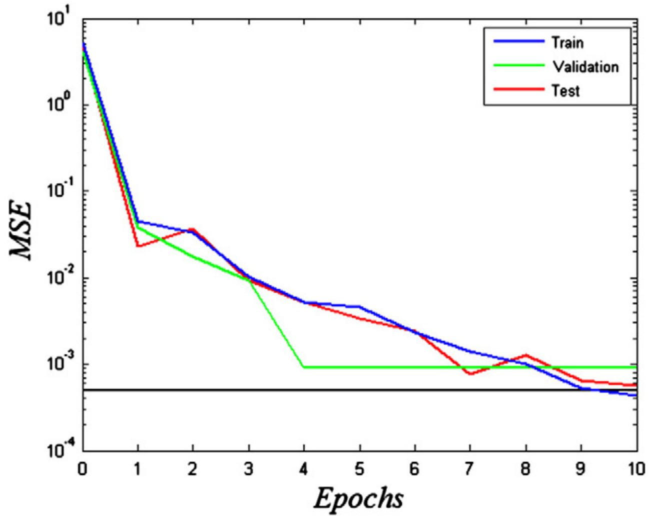

Fig. 5. Evolution of network performance (*MSE*) during training phase using LM-BP algorithm; goal is 5×10^{-4} while performance is 4.29×10^{-4}.

the connections between the inputs, the neurons and the output consist of weights and biases that are the parameters of MLP (3:9:1). It should be pointed out that the biases connected to all artificial neurons from the hidden and output layers (Fig. 4) play a similar role to the offset terms in multiple regression models.

The training of the neural network MLP (3:9:1) has been carried out by adjusting the values of weights and biases in order to minimize the performance function *MSE*. Fig. 5 shows the evolution of the *MSE* during training phase when LM-BP algorithm has been applied. It was observed that after 10 epochs (iterations) the training performance of 4.29×10^{-4} was achieved. Therefore, the training convergence has been satisfied since the goal was set at 5×10^{-4}. For validation and test subsets the performance function after 10 epochs was lower than 1×10^{-3}, which is acceptable.

After training phase the optimal values of the network weights and biases were saved and the developed ANN has been used for simulation and optimization of the SGMD process. Table 3 presents the optimal values of the network weights and biases in vector–matrix format after computation with LM-BP algorithm. Thus, the ANN model developed for the prediction of the SGMD performance index can be presented mathematically as an input–output composite mapping:

$$\hat{\mathbf{Y}}_n(\mathbf{x}) = \boldsymbol{\psi}^{(2)}\left(\mathbf{LW}^{(2,1)}\boldsymbol{\psi}^{(1)}\left(\mathbf{IW}^{(1,1)}\mathbf{x}+\mathbf{b}^{(1)}\right)+b^{(2)}\right) \tag{7}$$

where $\hat{\mathbf{Y}}_n$ denotes the vector of the normalized output (network predictions), $\mathbf{x}$ is the vector of the input variables, $\boldsymbol{\psi}^{(1)}$ is the vector of *logsig* transfer function corresponding to the hidden layer (layer-1), $\boldsymbol{\psi}^{(2)}$ is the vector of *purelin* transfer function corresponding to the output layer (layer-2), $\mathbf{IW}^{(1,1)}$ is the input weight matrix, $\mathbf{LW}^{(2,1)}$ is the layer weight vector, $\mathbf{b}^{(1)}$ is the bias vector, and $b^{(2)}$ is the bias scalar.

The agreement between the target (experimental observations) and the network output (predictions) for training, validation, and test sets is

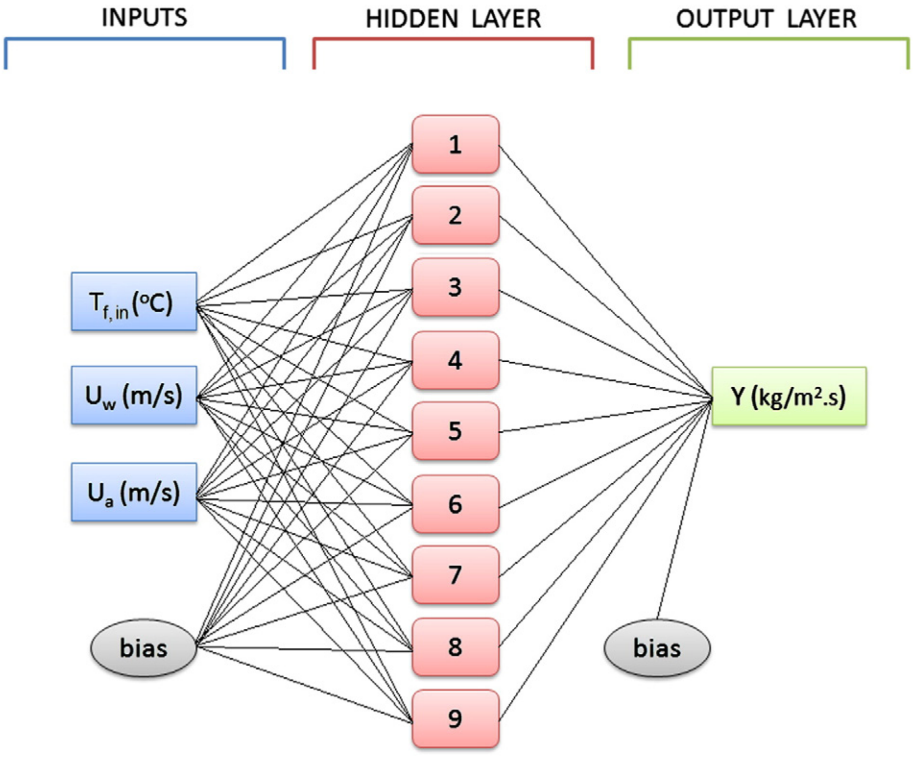

Fig. 4. Optimal topology of the developed ANN model used for prediction of the SGMD performance index.

Table 3
Optimal values of the network weights and biases computed by means of LM-BP algorithm.

Input weight matrix, IW	**IW**{1,1}=		
{Destination: hidden layer			
Source: inputs}			
	−11.3157	−4.7519	9.4878
	−5.9269	1.11766	7.6865
	−5.7622	−11.1475	−9.6669
	12.0438	5.3625	−7.2883
	5.9918	3.7665	−5.4772
	9.215	8.6715	4.4463
	−12.3579	6.9493	1.4116
	7.6707	−1.8324	5.9155
	−10.4228	7.0446	−3.4597
Bias vector, b	**LW**{2,1}^{T}=		
{Destination: hidden layer}			
	10.9375		
	2.2155		
	14.8345		
	−5.5984		
	−9.48		
	−4.12229		
	−2.3598		
	1.842		
Layer weight vector, LW	**LW**{2,1}^{T}=		
{Destination: output layer			
Source: hidden layer}			
	−0.6597		
	0.2486		
	−0.1759		
	0.0933		
	−05746		
	0.3838		
	−0.2305		
	−0.1517		
	0.1402		
Bias scalar, b	b{2}=0.7267		
{Destination: output layer}			

shown in Fig. 6. The overall correlation coefficient including training, validation and test, is $r^2 = 0.80$, while the correlation coefficient for validation is higher ($r^2 = 0.93$). These values reveal a satisfactory prediction of the experimental data by means of the developed ANN of MLP (3:9:1) type. In addition, the average relative error has been computed to appraise the goodness-of-fit between the ANN model and experiment. This average relative error is of 8.66% revealing a reasonable neural network model.

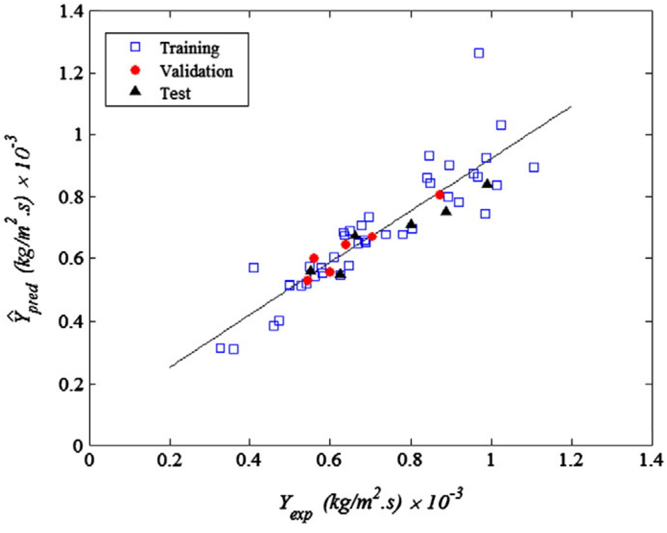

Fig. 6. Network outputs (predictions of SGMD performance index) plotted against the experimental observations (target).

The trained neural network was applied to plot 3D and 2D diagrams showing the simultaneous effects of two variables on the SGMD performance index.

Fig. 7 shows the influence of the air circulation velocity (U_a) and the feed inlet temperature ($T_{f,in}$) on the performance index $\left(\hat{Y}\right)$. As can be seen from Fig. 7, increasing both U_a and $T_{f,in}$ leads to an enhancement of $\hat{Y}$. An interaction effect between U_a and $T_{f,in}$ was detected. For example, at a high values of U_a and $T_{f,in}$, $\hat{Y}$ increases considerably due to the synergetic effect between these two input variables.

Fig. 8 illustrates the influence of the liquid circulation velocity (U_w) and $T_{f,in}$ on $\hat{Y}$. As stated previously, the increase of $T_{f,in}$ leads to an increase of $\hat{Y}$, whereas the influence of U_w on $\hat{Y}$ is insignificant. Similar results were obtained in our previous study dealing with modeling SGMD using response surface methodology (RSM) [27]. From the RSM model, it was also concluded that the effect of the liquid flow rate on the distillate flux was less significant.

The influence of U_a and U_w on $\hat{Y}$ is shown in Fig. 9. A significant interaction effect between these two variables was observed. At low U_w values, $\hat{Y}$ increases with the increase of U_a. In contrast, for higher U_w values, the increment of U_a leads first to a slight increase of $\hat{Y}$ followed by a stationary zone and finally a slight decrease. On the other hand, the increase of U_w at low U_a values results in an enhancement of . However, for higher U_a values, the increase of U_w does not change $\hat{Y}$ significantly. Similar interaction effects have been detected

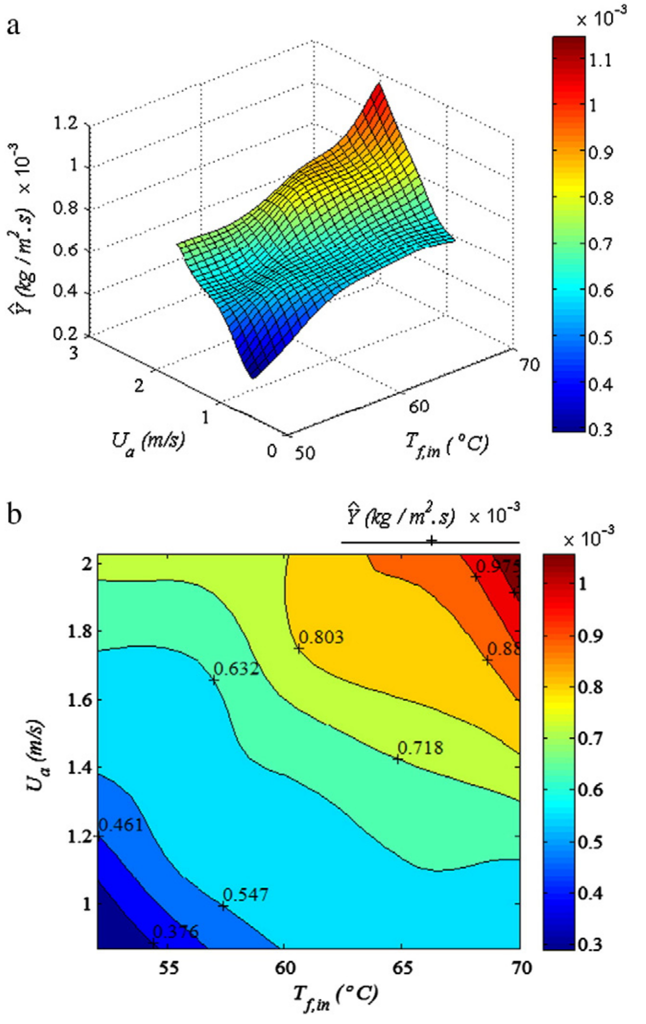

Fig. 7. SGMD performance index $\left(\hat{Y}\right)$ as a function of $T_{f,in}$ and U_a for $U_w = 0.165$ m/s, (a) 3D output surface and (b) 2D contour-lines map.

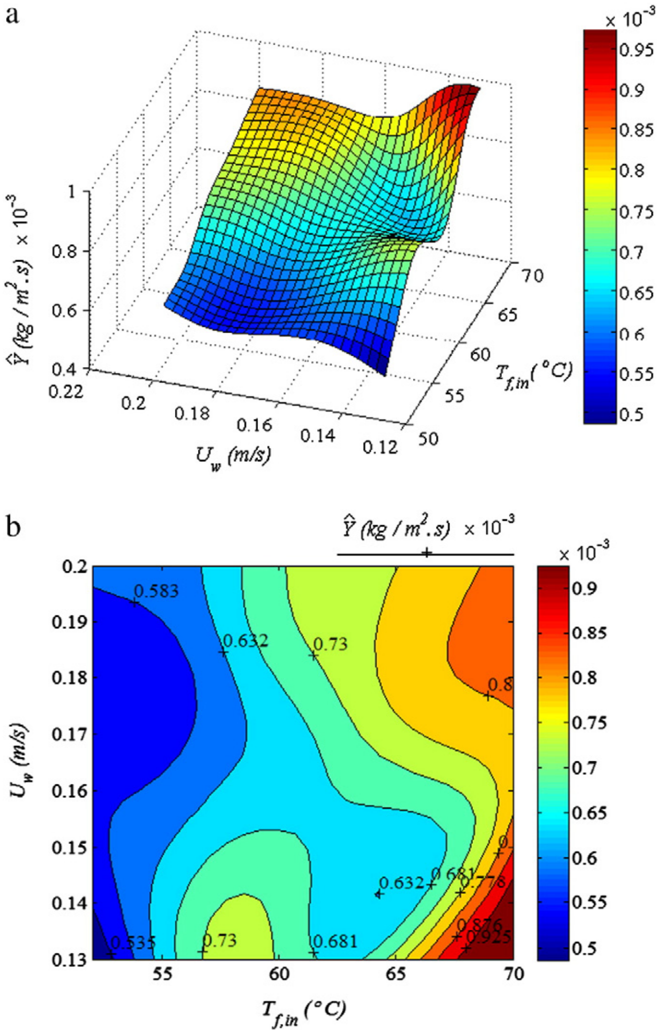

Fig. 8. SGMD performance index $\left(\hat{Y}\right)$ as a function of $T_{f,in}$ and U_w for $U_a = 1.45$ m/s, (a) 3D output surface and (b) 2D contour-lines map.

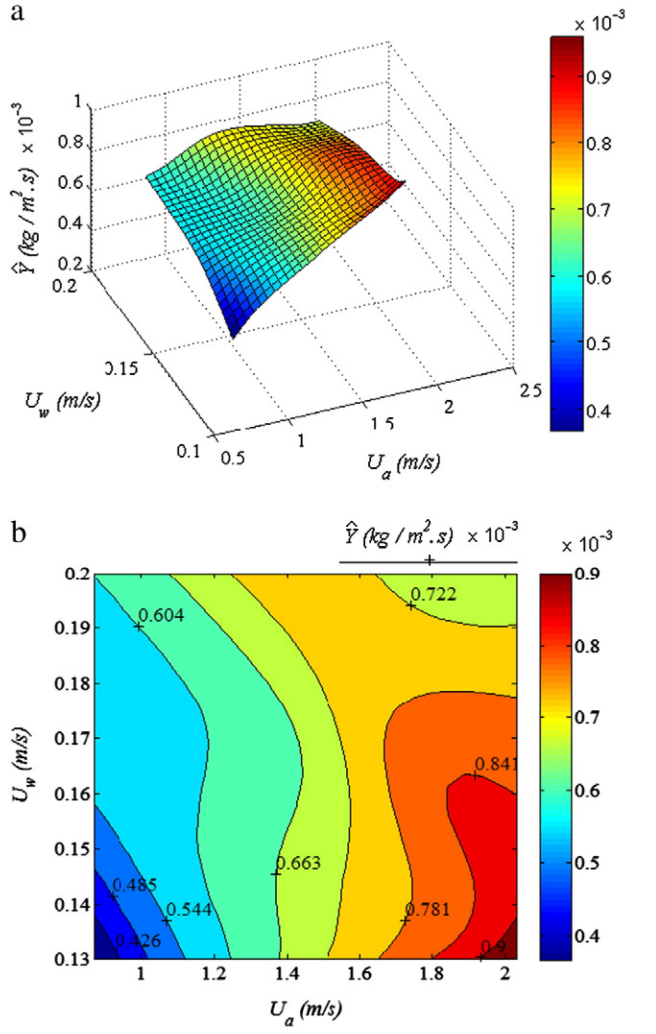

Fig. 9. SGMD performance index $\left(\hat{Y}\right)$ as a function of U_w and U_a for $T_{f,in} = 61$ °C, (a) 3D output surface and (b) 2D contour-lines map.

when RSM was used for SGMD modeling [27]. It was stated that this last effect was attributed to the formation of small air bubbles on the membrane surface contacting the liquid (i.e. feed/membrane surface) due to the high sweeping air flow rate. The presence of such bubbles produces an increase of the resistance to mass transfer and therefore a reduction of the permeate flux. As a consequence, operating under a high air flow rate above a critical point that depends on the liquid flow rate, its temperature and pressure should be avoided.

The constructed neural network model MLP (3:9:1) has been used to model and optimize the SGMD process. Monte Carlo simulation was employed for optimization. According to this algorithm, for each normalized input variable (x_j) the pseudo-random numbers (PRNs) are generated and thus a set of trial points is produced. Then, the network output is computed for each point using Eq. (7) and the maximal output value is determined by comparison. PRNs are generated by the computer processor being uniformly distributed in the normalized interval, PRN ∈ [0–1]. For instance, in MATLAB program, PRNs can be generated automatically using the standard function *rand*. In our case, Monte-Carlo simulations were carried out by multistage approach using a zoom-in technique in order to identify the optimal solution more accurately. At each stage a number of 10^4 points were generated randomly. The computed optimum condition given by the ANN model is summarized in Table 4. This condition corresponds to the following optimal values of the variables, $T_{f,in} = 69$ °C, $U_a = 2.02$ m/s (i.e. 34.5 L/min) and $U_w = 0.155$ m/s (i.e. 160 L/h). Table 4 also shows the experimental performance index together with the predicted one for the determined optimum condition. As it can be observed, the experimental response ($Y_{\exp} = 1.493 \times 10^{-3}$ kg/m².s) is slightly higher than the predicted one ($\hat{Y}_{pred} = 1.365 \times 10^{-3}$ kg/m².s) being the residual error 8.57%. It is to point out that, among all responses of the 53 different SGMD tests (Table 2), the performance index corresponding to the optimum operating condition is the highest one.

Table 4
Optimal conditions determined by means of the ANN model and Monte Carlo simulation.

Feed solution	NaCl (30 g/L)
Optimal operating conditions:	
$T_{f,in}$ (°C)	69
U_a (m/s)	2.02
U_w (m/s)	0.155
Responses:	
J (kg/m².s)	1.495×10^{-3}
RF (%)	99.903
$Y_{\exp}$ (kg/m².s)	1.493×10^{-3}
$\hat{Y}_{pred}$ (kg/m².s)	1.365×10^{-3}

In our recent published paper [27], using the same SGMD system, the obtained SGMD optimal condition by RSM model was a liquid circulation velocity of 0.16 m/s (i.e. 165 L/h liquid flow rate), an air circulation velocity of 2.11 m/s (i.e. 36 L/min air flow rate) and a temperature difference between the liquid and air at the module inlets of 54.3 °C. In the present study, the optimal temperature difference obtained by ANN model is lower (i.e. 49 °C), whereas the liquid and air flow rates are quite similar. The optimum performance index determined by RSM model is 1.5 times higher than that obtained by ANN model. This is attributed to the higher temperature difference determined by the RSM model.

5. Conclusions

An artificial neural network model was proposed to predict the SGMD performance index considering the most important inputs variables, namely the feed inlet temperature, the air circulation velocity and the liquid circulation velocity. A feed-forward neural network was developed by means of back-propagation training method. SGMD was applied for desalination of saline aqueous solution of 30 g/L NaCl concentration. The agreement between the experimental and the ANN predicted response, which is the SGMD performance index, was acceptable.

According to the ANN predictions, the most significant effect on the SGMD performance index is attributed to the inlet feed temperature followed by the sweeping air flow rate. The liquid flow rate is the less significant variable having a small influence on the SGMD performance index. An interaction effect between the air flow rate and the liquid flow rate has been detected and discussed.

The ANN model was used for optimization of SGMD process. The obtained SGMD optimal conditions by Monte Carlo simulation were as follows: an inlet feed temperature of 69 °C, a sweeping air flow rate of 34.5 L/min (i.e. air circulation velocity of 2.02 m/s), and a feed liquid flow rate of 160 L/h (i.e. liquid circulation velocity of 0.155 m/s). The predicted SGMD performance index under such optimum conditions is 1.3654×10^{-3} kg/m^2.s. According to the confirmation run the optimal performance index determined experimentally is 1.493×10^{-3} kg/m^2.s being the maximal response within all 53 SGMD tests performed in this study. However, this maximal performance index determined by ANN is smaller than that obtained by response surface methodology (RSM).

Nomenclature

A	net input argument
b	bias term for a neuron
$\mathbf{b}$	bias vector for a layer
E	error function
J	distillate (permeate) flux
IW	input weight matrix
logsig	log-sigmoid transfer function (Matlab syntax)
LW	layer weight matrix
MSE	mean-squared-error (performance function)
n	number of input variables
N	total number of experimental trials
m	number of artificial neurons
PRN	pseudo random number
purelin	linear transfer function (Matlab syntax)
r^2	linear correlation coefficient
RF	rejection factor
$T_{a,in}$	air inlet temperature
$T_{f,in}$	liquid inlet temperature
ΔT	temperature difference
tansig	hyperbolic tangent sigmoid transfer function (Matlab syntax)
U_a	air circulation velocity
U_w	liquid circulation velocity
w	weight (neural network connection)
$\mathbf{x}$	vector of normalized input variables
x	normalized input variable
Y	performance index (experimental value/target)
$\hat{Y}$	performance index (predicted value/network output)
Y_n	normalized performance index
$\hat{\mathbf{Y}}_n$	vector of network output (normalized predicted values of performance index)
z	actual variable

Subscripts and Superscripts

exp	experimental value
i	positive integer number (iteration index)
j	positive integer number (iteration index)
k	positive integer number (iteration index)
max	maximal level
min	minimal level
pred	predicted value
q	positive integer number (iteration index)

Greek letters

Δ	positive quantity
η	learning rate
$\boldsymbol{\psi}$	vector of transfer function

Acknowledgments

The author (C. Cojocaru) is grateful to the Spanish Ministry of Science and Innovation for supporting the research grant (project SB2009-0009). The authors also acknowledge the financial support from the University Complutense of Madrid, UCM-BSCH (projects GR58/08 and GR35/10-A, UCM group 910336).

References

[1] A. Abbas, N. Al-Bastaki, Modeling of an RO water desalination unit using neural networks, Chem. Eng. J. 114 (1–3) (2005) 139–143.

[2] M.K. Purkait, V.D. Kumar, D. Maity, Treatment of leather plant effluent using NF followed by RO and permeate flux prediction using artificial neural network, Chem. Eng. J. 151 (1–3) (2009) 275–285.

[3] Y. Zhao, J.S. Taylor, S. Chellam, Predicting RO/NF water quality by modified solution diffusion model and artificial neural networks, J. Membr. Sci. 263 (1–2) (2005) 38–46.

[4] V. Yangali-Quintanilla, A. Verliefde, T.U. Kim, A. Sadmani, M. Kennedy, G. Amy, Artificial neural network models based on QSAR for predicting rejection of neutral organic compounds by polyamide nanofiltration and reverse osmosis membranes, J. Membr. Sci. 342 (1–2) (2009) 251–262.

[5] D. Libotean, J. Giralt, F. Giralt, R. Rallo, T. Wolfe, Y. Cohen, Neural network approach for modeling the performance of reverse osmosis membrane desalting, J. Membr. Sci. 326 (2) (2009) 408–419.

[6] M. Al-Abri, N. Hilal, Artificial neural network simulation of combined humic substance coagulation and membrane filtration, Chem. Eng. J. 141 (1–3) (2008) 27–34.

[7] H. Chen, A.S. Kim, Prediction of permeate flux decline in crossflow membrane filtration of colloidal suspension: a radial basis function neural network approach, Desalination 192 (1–3) (2006) 415–428.

[8] N.A. Darwish, N. Hilal, H. Al-Zoubi, A.W. Mohammad, Neural networks simulation of the filtration of sodium chloride and magnesium chloride solutions using nanofiltration membranes, Chem. Eng. Res. Des. 85 (4) (2007) 417–430.

[9] G.B. Sahoo, C. Ray, Predicting flux decline in crossflow membranes using artificial neural networks and genetic algorithms, J. Membr. Sci. 283 (1–2) (2006) 147–157.

[10] J.N. Mhurchú, G. Foley, Dead-end filtration of yeast suspensions: correlating specific resistance and flux data using artificial neural networks, J. Membr. Sci. 281 (1–2) (2006) 325–333.

[11] M. Khayet, C. Cojocaru, M. Essalhi, Artificial neural network modeling and response surface methodology of desalination by reverse osmosis, J. Membr. Sci. 368 (1–2) (2011) 202–214.

[12] M. Khayet, Membranes and theoretical modeling of membrane distillation: a review, Adv. Colloid Interface Sci. 164 (1–2) (2011) 56–88.

[13] M. Khayet, C. Cojocaru, Artificial neural network modeling and optimization of desalination by air gap membrane distillation, Sep. Purif. Technol. 86 (2012) 171–182.

[14] M. Khayet, M.P. Godino, J.I. Mengual, Theoretical and experimental studies ondesalination using the sweeping gas membrane distillation method, Desalination 157 (2003) 297–305.

[15] M. Khayet, M.P. Godino, J.I. Mengual, Thermal boundary layers in sweeping gas membrane distillation processes, AIChE J. 48 (2002) 1488–1497.

[16] Wikipedia (the free encyclopedia): Artificial Neural Network. available on-line at: http://en.wikipedia.org/wiki/Artificial_neural_network.

[17] M.T. Hagan, H.B. Demuth, M. Beale, Neural Network Design, PWS Publishing Co., Boston, 1996.

[18] H. Demuth, M. Beale, Neural Network Toolbox: For Use with MATLAB (Version 4.0), The MathWorks, Inc., 2004.

[19] A. Shahsavand, M. Pourafshari Chenar, Neural networks modeling of hollow fiber membrane processes, J. Membr. Sci. 297 (1–2) (2007) 59–73.

[20] M. Dornier, M. Decloux, G. Trystram, A. Lebert, Dynamic modeling of cross-flow microfiltration using neural networks, J. Membr. Sci. 98 (3) (1995) 263–273.

[21] M. Sadrzadeh, T. Mohammadi, J. Ivakpour, N. Kasiri, Separation of lead ions from wastewater using electrodialysis: comparing mathematical and neural network modeling, Chem. Eng. J. 144 (3) (2008) 431–441.

[22] H. Niemi, A. Bulsari, S. Palosaari, Simulation of membrane separation by neural networks, J. Membr. Sci. 102 (1995) 185–191.

[23] C. Cojocaru, M. Macoveanu, I. Cretescu, Peat-based sorbents for the removal of oil spills from water surface: application of artificial neural network modeling, Colloids Surf., A 384 (1–3) (2011) 675–684.

[24] K. Yetilmezsoy, S. Demirel, Artificial neural network (ANN) approach for modeling of Pb(II) adsorption from aqueous solution by Antep pistachio (*Pistacia Vera* L.) shells, J. Hazard. Mater. 153 (3) (2008) 1288–1300.

[25] I.N. da Silva, R.A. Flauzino, An approach based on neural networks for estimation and generalization of crossflow filtration processes, Appl. Soft Comput. 8 (1) (2008) 590–598.

[26] B. Sarkar, A. Sengupta, S. De, S. DasGupta, Prediction of permeate flux during electric field enhanced cross-flow ultrafiltration — a neural network approach, Sep. Purif. Technol. 65 (3) (2009) 260–268.

[27] M. Khayet, C. Cojocaru, A. Baroudi, Modeling and optimization of sweeping gas membrane distillation, Desalination 287 (2012) 159–166.