

# Application of ANN modeling for oily wastewater treatment by hybrid PAC-MF process

Mohsen Abbasi\*, Yaser Rasouli and Peyman Jowkar

Department of Chemical Engineering, School of Chemical and Petroleum Engineering, Persian Gulf University, Bushehr, 75169, Iran

(Received July 25, 2017, Revised August 10, 2017, Accepted August 19, 2017)

**Abstract.** In the following study, Artificial Neural Network (ANN) is used for prediction of permeate flux decline during oily wastewater treatment by hybrid powdered activated carbon-microfiltration (PAC-MF) process using mullite and mullite-alumina ceramic membranes. Permeate flux is predicted as a function of time and PAC concentration. To optimize the networks performance, different transfer functions and different initial weights and biases have been tested. Totally, more than 850,000 different networks are tested for both membranes. The results showed that 10:6 and 9:20 neural networks work best for mullite and mullite-alumina ceramic membranes in PAC-MF process, respectively. These networks provide low mean squared error and high linearity between target and predicted data (high  $R^2$  value). Finally, the results present that ANN provide best results ( $R^2$  value equal to 0.99999) for prediction of permeation flux decline during oily wastewater treatment in PAC-MF process by ceramic membranes.

**Keywords:** artificial neural network; ceramic membranes; oily wastewater treatment; microfiltration; powdered activated carbon

## 1. Introduction

Developments in oil and gas refinement industries have increased the pollutant production rate and as a sequence, wastewater production, dramatically. The wastewater leaving industrial plants mostly include poisonous materials extremely harmful to the environment which may affect ecosystems irreversibly (Rácz *et al.* 2015, Razavi *et al.* 2015). In spite of such ecological problems, wastewater treatment is usually ignored for economic issues, or due to technological limitations, the pollutants may not be filtered as required. Therefore, it is essential to design new processes to treat industrial wastewater effluents both economically and efficiently (Maddah and Chogle 2015). Several main processes have been proposed for wastewater treatment including flotation (Zhu and Zhou 2014), coagulation (da Conceição *et al.* 2015), electrostatic and electro-coagulation processes (Ganesan *et al.* 2013, Djahida *et al.* 2014), biological treatment (Ratanatamskul *et al.* 2012, Lester *et al.* 2013), oxidation processes (He *et al.* 2013) and membrane separation technologies (Amaral *et al.* 2016, Kasim *et al.* 2016). A more complete review of these processes is available in literature (Patil *et al.* 2016).

Among these wastewater treatment methods, membrane processing method and particularly microfiltration (MF), is considered as a promising method able to remove micro-particles, micro-organisms, macro-molecules and bacteria while removal of such materials may not be possible by conventional methods (Abadi *et al.* 2011). MF membranes

are made of different materials such as: Alumina (Chen *et al.* 2015), mullite (Abbasi *et al.* 2010), silica (Ghouil *et al.* 2015), zirconia (Werner *et al.* 2014), etc.

In addition to very high thermal and chemical stability, mullite ceramic MF membranes are very cheap and can be easily prepared from low cost kaolin clay (Abbasi *et al.* 2010). Membrane low preparation cost is a very important factor for industrializing mullite membranes for wastewater treatment processes. Mullite and mullite-alumina ceramic MF membrane have been successfully used for wastewater treatment in a hybrid PAC-MF process in our previous work (Abbasi *et al.* 2011). Addition of alumina have given sufficient properties to mullite membranes for wastewater treatment while keeping the membrane preparation cost low enough to be used industrially.

Artificial neural network (ANN) is a simplified model of human brain. This network is made up of many interconnected blocks called neuron. These neurons, are aligned in layers. There are three main layers: Input layer, getting input signals from the surroundings; hidden layer, the main calculations are done here; and output layer, which combines calculations into result signals (Fig. 1(b)). By the use of such structure, neural network breaks complicated tasks to many simple tasks, computable by simple neurons. ANNs have been successfully used in different areas. In chemical engineering area, ANNs can be used for data prediction (non-linear regression), classification, association, conceptualization, filtering and optimization (Shokrkar *et al.* 2011). For prediction purpose, ANNs can be fitted on data with high non-linearity, where very complicated equations might be needed to fit data. The ANN training step is, however, much simpler than finding a suitable equation for non-linear regression. There are many uses of ANNs in prediction and optimization data

\*Corresponding author, Professor  
E-mail: [m.abbasi@pgu.ac.ir](mailto:m.abbasi@pgu.ac.ir)

(Nourouzi *et al.* 2011, Shokrkar *et al.* 2011, Chatteraj *et al.* 2014). Membrane processes, as well as many other engineering processes, show high non-linearity. Therefore, ANNs can be used to predict the membrane process operational parameters, both easily and accurately. A limited number of studies in the literature are focused on modeling of membrane performance in desalination and wastewater treatment processes by ANNs (Porrizzo *et al.* 2013, Cao *et al.* 2016).

In the following study, data collected from a wastewater treatment setup, presented elsewhere (Abbasi *et al.* 2011), is processed using a feed forward (perceptron) artificial neural network. The data is collected in optimum temperature and pressure. An ANN training program is developed to test networks, different in number of neurons and hidden layers, transfer functions and initial conditions to find the network that fits best. The results show that a two hidden layer ANN with 10:6 neurons, i.e., 10 and 6 neurons in first and second hidden layers, respectively, gives the best prediction for mullite membrane permeate flux and a two hidden layer ANN with 10:6 neurons gives the best prediction for mullite-alumina membrane permeate flux. The parameters of best networks are presented so that other researchers can easily benefit from the results of this paper.

## 2. Artificial neural network

### 2.1 Network structure

An artificial neural network is a network consisted of many single neurons connected together. As a simplified model of human brain, in an ANN, neurons are aligned in layers, connected together in different ways. There are three main network topologies based on type of connections between neurons: (1) intralayer, connecting two neurons in the same layer; (2) interlayer, connecting neurons in two different layers; and (3) recurrent, connecting a neuron to itself. Within the interlayer topology, there are two options: Feedback and feedforward connections. It is shown that for prediction purpose, a feedforward interconnected network (Fig. 1(b)) works best (Baughman and Liu 2014). Therefore, this architecture is used in this paper to estimate permeate flux of mullite and mullite alumina membranes as a function of time and PAC concentration in PAC-MF hybrid process. Schematics of this network is presented in Fig. 1(b).

As Fig. 1(b) shows, a feed forward neural network includes an input layer, at least one hidden layer and one output layer. In the input layer, the number of neurons is the same as number of input variables, i.e., two neurons for time and PAC concentration. Each neuron in this layer is connected to all neurons in the next layer.

The second layer, called hidden layer, is the place where computations are done. The input of any neurons in this layer is received from previous layer and output is calculated by Eq. (1) (see Fig. 1(a)).

Where  $w_{ij}$  is the weight factor for  $i^{\text{th}}$  input,  $I_i$ , of the  $j^{\text{th}}$  neuron and  $b_j$  and  $O_j$  are bias and output of the  $j^{\text{th}}$  neuron, respectively.

$$O_j = f\left(\sum_{i=1}^n (w_{ij} I_i) + b_j\right) \quad (1)$$

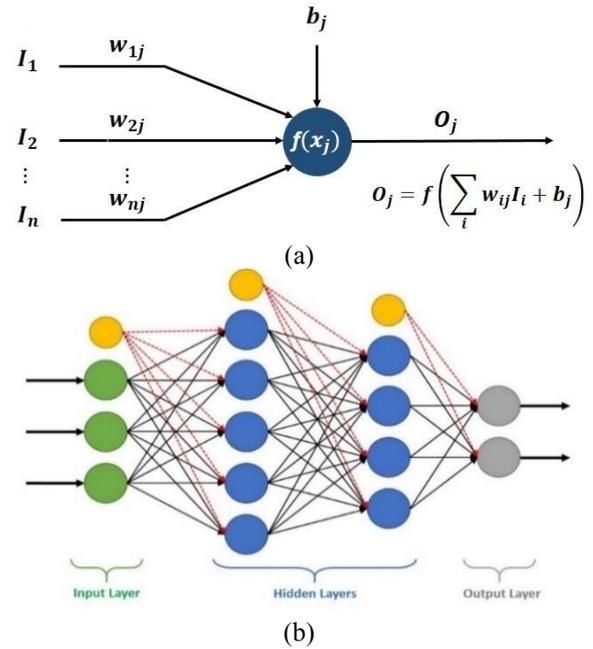


Fig. 1 (a) Basic structure of a single artificial neuron, (b) Basic structure of a feed forward neural network. Smaller yellow circles indicate bias blocks

Each input of a neuron is multiplied by the corresponding weight factor. In the next step, a bias is added to the summation of all these weighted inputs to result a total input for that neuron. The weights determine how much an individual input can affect the output of the neuron and bias determines how a single neuron can affect the network final output. Weights and biases can have either positive or negative values. The obtained input signal is then passed to a transfer function  $f(x)$  and the value of this function will be the final output of that neuron,  $O_j$ .

The transfer function can be any mathematical function but there are several common functions used commonly. A list of these functions is provided in Table 1. In this paper, Tan-Sigmoid, Log-Sigmoid, Linear and Radial Basis transfer functions are used for hidden and output layers.

### 2.2. Data preprocessing

Neural networks are strongly dependent on the data used for training step. In this step, if the input and output variables are not of the same order of magnitude, some variables may appear to have more significance than they actually do. The training algorithm has to compensate for order-of-magnitude differences by adjusting the network weights, resulting in large weight values for small inputs and vice versa (Baughman and Liu 2014).

This is not efficient in many of the training algorithms. To eliminate this effect, the data can be normalized. There are different normalization procedures. In this paper, the data is normalized to fall in the interval (-1, 1). The normalization function is as follow.

$$x_{i, Norm} = \frac{2 \times (x_i - x_{Min})}{(x_{Max} - x_{Min})} - 1 \quad (2)$$

Table 1. Common transfer functions used in artificial neural networks

Name	Equation	Plot
Hard-Limit	$f(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$	
Symmetric Hard-Limit	$f(x) = \begin{cases} 1 & x > 0 \\ -1 & x \leq 0 \end{cases}$	
Positive Linear	$f(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$	
Linear	$f(x) = x$	
Log-Sigmoid	$f(x) = \frac{1}{1 + e^x}$	
Tan-Sigmoid	$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$	
Radial Basis	$f(x) = e^{-x^2}$	

Where  $x_{Min}$  and  $x_{Max}$  are the minimum and maximum values of data, respectively and  $x_{i, Norm}$  is the normalized value corresponding to  $x_i$ . After the data is normalized, the network is trained by new normalized data and the network output is reverse-normalized to real values. In this way, weights or biases are not affected by either large or small values of data and network will be trained properly.

In addition to data normalization, it is crucial to randomize the data so that the training, validation and test data is properly distributed over the entire data range. To eliminate the effect of data randomization on network performance, data is randomized several times, each used separately to find randomized data set that presents best network performance.

### 2.3. Training the network

There are different training techniques for neural networks including supervised and unsupervised training (Suykens *et al.* 2012). For prediction purpose, where input and target data are available, supervised training is preferred. A well-known supervised training algorithm is back propagation (Hecht-Nielsen 1989, Erb 1993). There are different modifications to this algorithm for either faster convergence or better performance. Levenberg-Marquardt back propagation and Bayesian Regulation back

propagation are two important modifications to this algorithm (MacKay 1996). In this study, different neural networks with different structures are trained using fast Levenberg-Marquardt (LM) backpropagation and accurate Bayesian Regulation (BR) backpropagation to find the best possible network.

Besides the training algorithm, the initial weights and biases used in training step have strong effect on network performance. While different procedures like genetic algorithm have been proposed (Chang *et al.* 2012), there is no efficient general rule to find best initial conditions and a trial and error procedure is still a basic method and is followed here. For trial and error procedure, each network structure is trained several times with different initial weights and biases to find the best initial conditions.

The network performance is evaluated by mean squared error (MSE). MSE is calculated by Eq. (3)

$$MSE = \frac{1}{N} \sum_{i=1}^n (x_{Exp} - x_{Pred})^2 \quad (3)$$

In this equation,  $x_{Exp}$  and  $x_{Pred}$  are experimental and network-predicted data and  $N$  is number of data samples. In each training epoch, MSE is calculated for training, validation and test data. The best network is identified by lowest mean squared error on test data.

Also, linear regression coefficient,  $R^2$  is calculated by Eq. (4) to represent the linearity between target and predicted data.

$$R^2 = \frac{\sum_{i=1}^n (x_{Exp} - \overline{x_{Exp}})^2 - \sum_{i=1}^n (x_{Exp} - x_{Pred})^2}{\sum_{i=1}^n (x_{Exp} - \overline{x_{Exp}})^2} \quad (4)$$

In the computer program which we developed, the training process is terminated by one the following criteria:

I. If the MSE of validation data starts to increase, the training process will be terminated if no reductions in MSE is observed by 20 next epochs. The 20 epochs will guarantee that network is not trapped in local minima.

II. If the change of MSE on validation data in two sequent epochs drops to less than  $10^{-5}$ , the training process will be terminated.

In either case, the network weights and biases that result the lowest MSE on validation data, are stored in network structure file.

### 3. Result and discussion

Table 2a Details of best trained networks for mullite ceramic membrane

Layer	Input	Hidden 1	Hidden 2	Output	
Neurons	2	10	6	1	
Transfer function	-	Tan-Sigmoid	Tan-Sigmoid	Linear	
MSE	Training	Validation	Test	Total	R2
	0.002084	0.003273	0.003761	0.002511	0.9999

Table 2b Details of best trained networks for mullite-alumina ceramic MF membrane.

Layer	Input	Hidden 1	Hidden 2	Output
Neurons	2	9	20	1
Transfer function	-	Tan-Sigmoid	Tan-Sigmoid	Linear
MSE	Training Validation Test	Total	R2	
	0.053266 0.083400 0.082464	0.062114	0.9999	

Variation of flux decline with time for oily wastewater treatment in hybrid PAC-MF process with mullite and mullite-alumina membranes are adopted from our previous work (Abbasi *et al.* 2011) and different neural networks are fitted on this data to find the best network architecture for prediction of permeate flux as a function of time and PAC concentration in membranes at optimal temperature and pressure. The Levenberg-Marquardt and Bayesian Regulation backpropagation training algorithms were used to train ANNs. Generally, in this study Levenberg-Marquardt backpropagation method represented lower MSE error with faster convergence. The training/validation/test data ratio was kept at 70/15/15 percent for all networks. In hidden and output layers, Tan-Sigmoid, Log-Sigmoid, Linear and Radial Basis transfer functions were used but experiments showed that Tan-Sigmoid and Linear transfer functions for hidden and output layers respectively, will produce the best results. Consequently, for all the results represented here, same transfer functions are used and Levenberg-Marquardt backpropagation method is used to train networks. To eliminate the effect of initial weights and biases, each network architecture is trained several times with different random initial weights and biases and the initial conditions resulting lowest mean squared error for test data, are used for final training. The networks are considered to have one or two hidden layers with 1 to 20 neurons for mullite and 1 to 30 neurons for mullite-alumina membranes. This is equal to almost 1,350 different network architectures, for both mullite and mullite-alumina membranes. Accounting different initial weights and biases and transfer functions for each architecture, more than 850,000 networks are tested totally. The MSE of test data and MSE of total data for different network architectures are shown in Fig. 2. For better representation of network with lowest error, the inverse of MSE is plotted based on number of neurons in first and second layers. Note that zero neurons in second layer is equal to a single hidden layer network. As Fig. 2 shows, for mullite membrane, a 10:6 neural network, i.e., 10 and 6 neurons in first and second hidden layers, respectively, results the lowest both MSE for test data (Fig. 2(a)) and for total data (Fig. 2(b)) and for mullite-alumina membrane, a 9:20 network results the lowest both MSE for test data (Fig. 2(c)) and for total data (Fig. 2(d)). Details of these networks are presented in Table 2.

The analysis of each single network structure indicated that network performance after training step, depends on selection of training, validation and test data significantly.

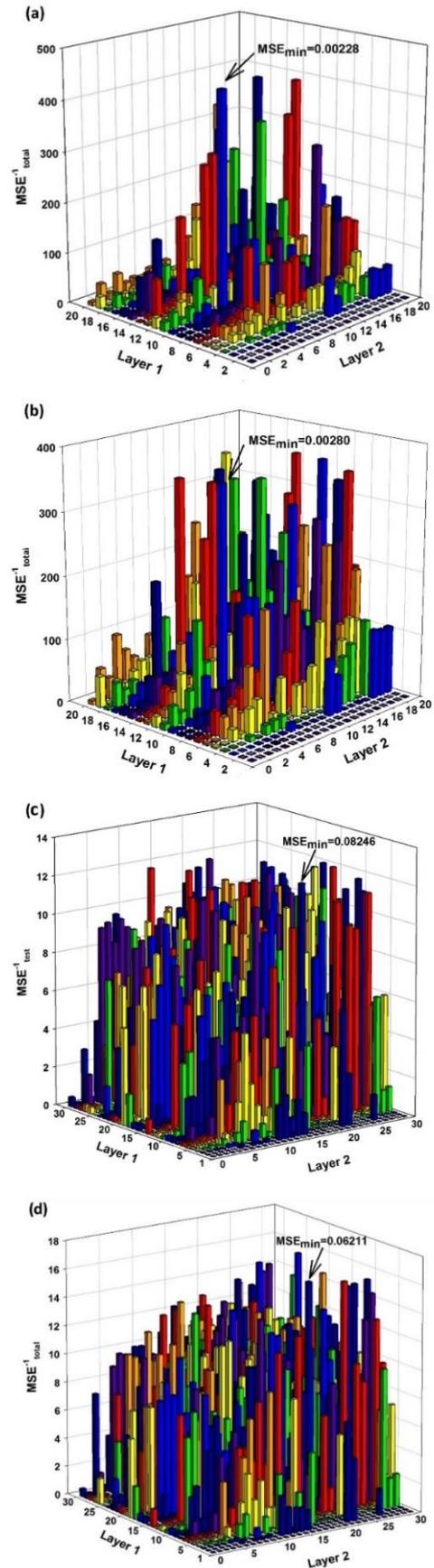


Fig. 2 Mean Squared Error (MSE) for mullite membrane (a) test and (b) total data and for mullite-alumina (c) test and (d) total data

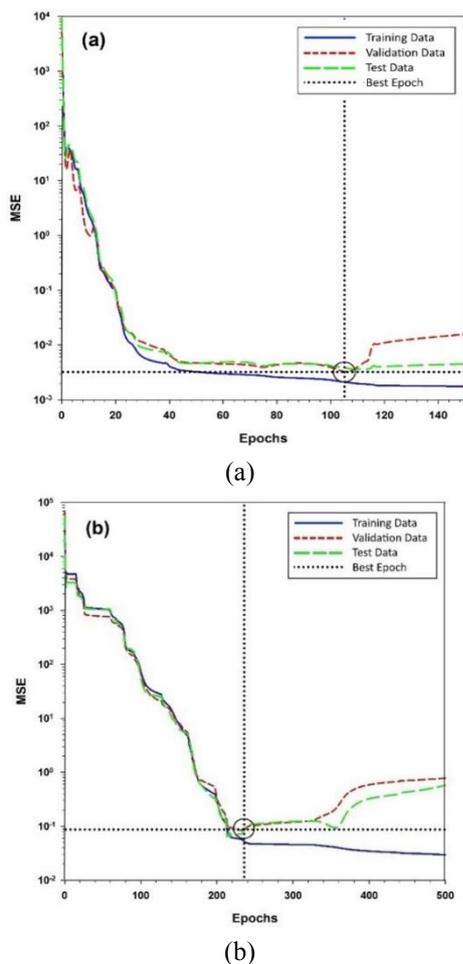


Fig. 3 Mean Squared Error (MSE) for (a) mullite membrane and (b) mullite-alumina membrane on each training epoch. Circle mark shows best epoch in training step

To optimize the performance of the selected networks from previous step, data was randomized for 100 times and each time, the networks were trained to find the best possible data classification. The optimized data classification is used to obtain the network training results presented hereinafter.

The MSE for training, validation and test data based on number of epochs are presented in Fig. 3. As this Figure shows, for both mullite (10:6) and mullite-alumina (9:20) membranes, the MSE of all three data classifications (training, validation and test) are near to each other and admit that networks do not overfit data. The best epoch is marked with circle and that is where the training step is stopped. Note that after best epoch, validation and test MSE increase significantly and that is an obvious sign of network overfitting.

To investigate the performance and accuracy of selected networks, linear regression coefficient,  $R^2$ , is calculated for test data and as shown in Fig. 4,  $R^2$  values very close to unity (0.99999) are obtained for both mullite and mullite-alumina membranes. These values indicate that selected networks are accurate with reasonable error and can be easily used to estimate permeate flux of membranes as a function of time and PAC content in hybrid process.

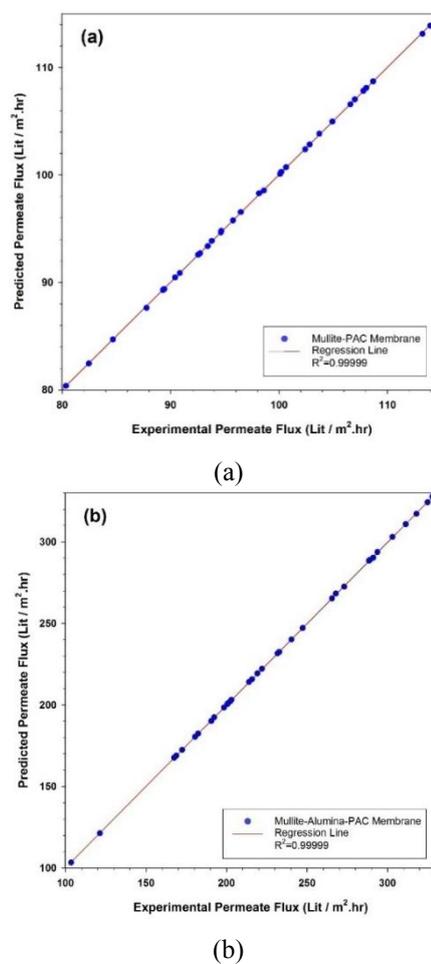


Fig. 4 Network-predicted vs. Experimental permeate flux for (a) mullite and (b) mullite-alumina membranes.  $R^2$  Values are obtained by performing linear regression on data

A comparison between network-predicted and experimental data is presented in Fig. 5. The data predicted by neural network is fitted on experimental data for different PAC concentrations at different times. There is a perfect match between neural network output and experimental data on overall time interval and PAC concentration range. This is in agreement with low MSE values and high  $R^2$  values obtained in Figs. 2 and 4.

In order to let other researchers use the results of this study, the optimized parameters of selected networks are presented in appendix 1 for mullite and mullite-alumina membranes. These parameters include weights and biases of all neurons in each network. Using these parameters, it is possible to easily simulate the trained networks and get results.

#### 4. Conclusions

In the present study, artificial neural networks are used to predict permeate flux of mullite and mullite-alumina ceramic membranes for oily wastewaters treatment in a hybrid PAC-MF process. The permeate flux is predicted as a function of time and PAC concentration. The feedforward neural network used in this study is considered to have one

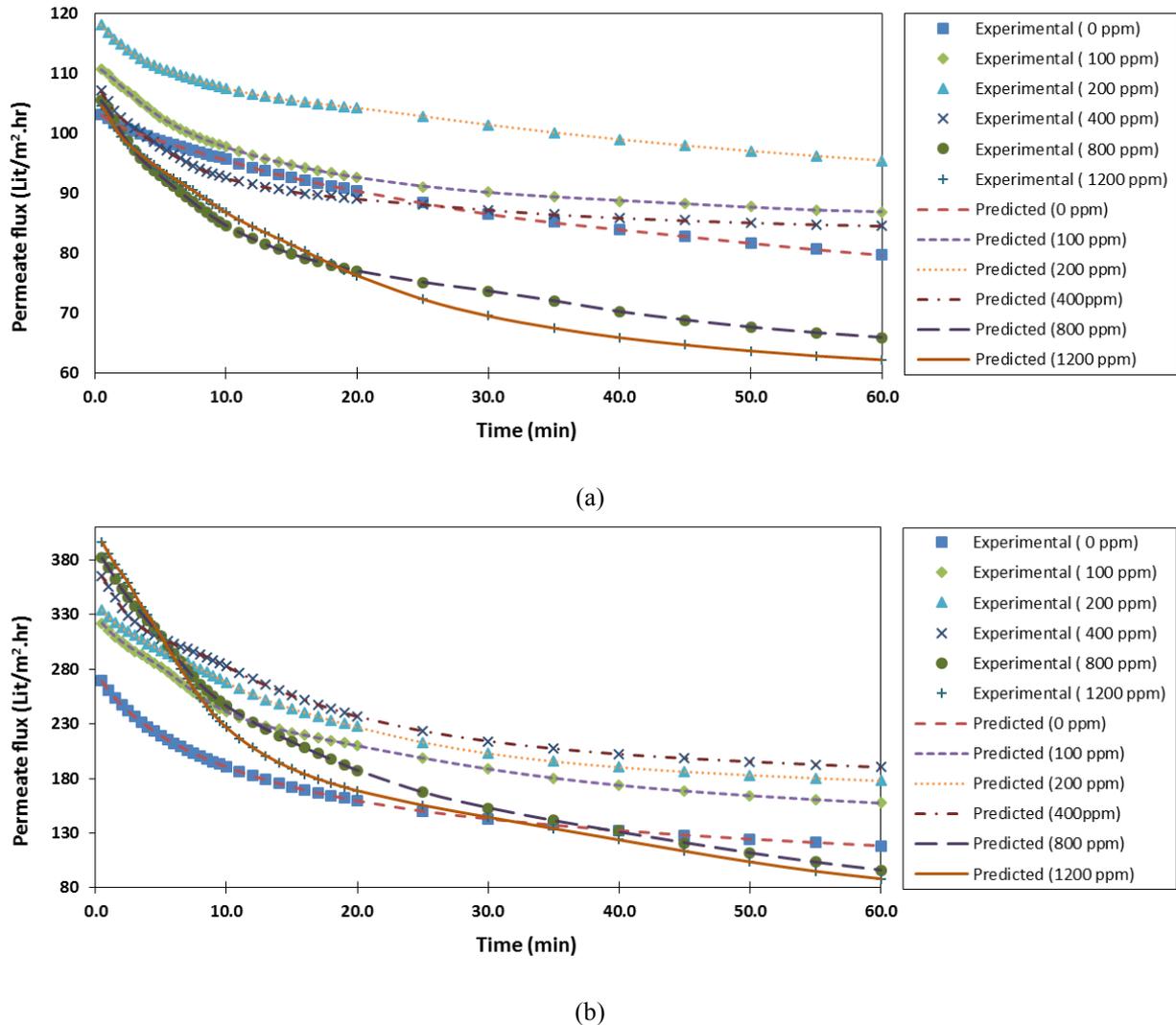


Fig. 5 Network predicted and experimental permeate flux as a function of time and PAC content in hybrid process for (a) mullite and (b) mullite-alumina membranes

or two hidden layers with zero to twenty or thirty neurons per each one. The Levenberg-Marquardt and Bayesian Regulation backpropagation training algorithms are used for ANN training. Generally for this specific case, Levenberg-Marquardt backpropagation method represented lower MSE error with faster convergence. To optimize the selected structure, different initial weights and biases, transfer functions and data classifications were tested. We found that a 10:6 and 9:20 ANN produce lowest MSE (for test and total data) for mullite and mullite-alumina ceramic membranes, respectively. The parameters of the trained networks are also presented and it is possible for other researchers to use the results of this study.

## References

- Abadi, S.R.H., Sebzari, M.R., Hemati, M., Rekabdar, F. and Mohammadi, T. (2011), "Ceramic membrane performance in microfiltration of oily wastewater", *Desalination*, **265**(1), 222-228.
- Abbasi, M., Mirfendereski, M., Nikbakht, M., Golshenas, M. and Mohammadi, T. (2010), "Performance study of mullite and mullite-alumina ceramic MF membranes for oily wastewaters treatment", *Desalination*, **259**(1), 169-178.
- Abbasi, M., Reza Sebzari, M. and T. Mohammadi (2011), "Enhancement of oily wastewater treatment by ceramic microfiltration membranes using powder activated carbon", *Chem. Eng. Technol.*, **34**(8), 1252-1258.
- Amaral, M.C.S., Andrade, L.H., Neta, L.S.F., Magalhães, N.C., Santos, F.S., Mota, G.E. and Carvalho, R.B. (2016), "Microfiltration of vinasse: Sustainable strategy to improve its nutritive potential", *Water Sci. Tech.*, **73**(6), 1434-1441.
- Baughman, D.R. and Liu, Y.A. (2014), *Neural Networks in Bioprocessing and Chemical Engineering*. Academic Press, Massachusetts, U.S.A.
- Cao, W., Liu, Q., Wang, Y. and Mujtaba, I. M. (2016), "Modeling and simulation of VMD desalination process by ANN", *Comput. Chem. Eng.*, **84**, 96-103.
- Chang, Y.T., Lin, J., Shieh, J.S. and Abbod, M.F. (2012), "Optimization the initial weights of artificial neural networks via genetic algorithm applied to hip bone fracture prediction", *Adv. Fuzzy Syst.*, **2012**, 6.
- Chattoraj, S., Mondal, N.K., Das, B., Roy, P. and Sadhukhan, B. (2014), "Carbaryl removal from aqueous solution by Lemna major biomass using response surface methodology and

- artificial neural network”, *J. Environ. Chem. Eng.*, **2**(4), 1920-1928.
- Chen, X., Zhang, W., Lin, Y., Cai, Y., Qiu, M. and Fan, Y. (2015), “Preparation of high-flux  $\gamma$ -alumina nanofiltration membranes by using a modified sol-gel method”, *Microporous Mesoporous Mater.*, **214**, 195-203.
- da Conceição, V.M., Ugri, A., Bonicontró, M.C., Silveira, C., Nishi, L., Vieira, M.F., de Jesus, B., Vieira, A.M.S. and Bergamasco, R. (2015), “Removal of excess fluoride from groundwater using natural coagulant *Moringa oleifera* Lam and microfiltration”, *Can. J. Chem. Eng.*, **93**(1), 37-45.
- Djahida, Z., Amel, B., Mourad, T.A., Hayet, D. and Rachida, M. (2014), “Treatment of a dye solophenyle 4GE by coupling electrocoagulation/nanofiltration”, *Membr. Water Treat.*, **5**(4), 251-263.
- Erb, R.J. (1993), “Introduction to backpropagation neural network computation”, *Pharmaceut. Res.*, **10**(2), 165-170.
- Ganesan, P., Lakshmi, J., Sozhan, G. and Vasudevan, S. (2013), “Removal of manganese from water by electrocoagulation: Adsorption, kinetics and thermodynamic studies”, *Can. J. Chem. Eng.*, **91**(3), 448-458.
- Ghouil, B., Harabi, A., Bouzerara, F., Boudaira, B., Guechi, A., Demir, M.M. and Figoli, A. (2015), “Development and characterization of tubular composite ceramic membranes using natural alumino-silicates for microfiltration applications”, *Mater. Charact.*, **103**, 18-27.
- He, X., Chai, Z., Li, F., Zhang, C., Li, D., Li, J. and Hu, J. (2013), “Advanced treatment of biologically pretreated coking wastewater by electrochemical oxidation using Ti/RuO<sub>2</sub>-IrO<sub>2</sub> electrodes”, *J. Chem. Technol. Biot.*, **88**(8), 1568-1575.
- Hecht-Nielsen, R. (1989), “Theory of the backpropagation neural network”, *Proceedings of the International Joint Conference on Neural Networks 1*, Washington, D.C., U.S.A., June.
- Kasim, N., Mohammad, A.W. and Abdullah, S.R.S. (2016), “Performance of membrane filtration in the removal of iron and manganese from Malaysia’s groundwater”, *Membr. Water Treat.*, **7**(4), 277-296.
- Lester, J., Jefferson, B., Eusebi, A.L., McAdam, E. and Cartmell, E. (2013), “Anaerobic treatment of fortified municipal wastewater in temperate climates”, *J. Chem. Technol. Biot.*, **88**(7), 1280-1288.
- MacKay, D.J. (1996), “Bayesian methods for backpropagation networks”, *Models of Neural Networks III*, Springer, New York, U.S.A.
- Maddah, H.A. and Choglem A.M. (2015), “Applicability of low pressure membranes for wastewater treatment with cost study analyses”, *Membr. Water Treat.*, **6**(6), 477-488.
- Nourouzi, M.M., Chuah, T.G. and Choong, T.S. (2011), “Optimisation of reactive dye removal by sequential electrocoagulation-flocculation method: Comparing ANN and RSM prediction”, *Water Sci. Technol.*, **63**(5), 984-994.
- Patil, D.S., Chavan, S.M. and Oubagaranadin, J.U.K. (2016), “A review of technologies for manganese removal from wastewaters”, *J. Environ. Chem. Eng.*, **4**(1), 468-487.
- Porrizzo, R., Cipollina, A., Galluzzo, M. and Micale, G. (2013), “A neural network-based optimizing control system for a seawater-desalination solar-powered membrane distillation unit”, *Comput. Chem. Eng.*, **54**, 79-96.
- Rácz, G., Kerker, S., Schmitz, O., Schnabel, B., Kovács, Z., Vatai, G., Ebrahimi, M. and Czermak, P. (2015), “Experimental determination of liquid entry pressure (LEP) in vacuum membrane distillation for oily wastewaters”, *Membr. Water Treat.*, **6**(3), 237-249.
- Ratanatamskul, C., Glingeysorn, N. and Yamamoto, K. (2012), “The BNR-MBR (Biological Nutrient Removal-Membrane Bioreactor) for nutrient removal from high-rise building in hot climate region”, *Membr. Water Treat.*, **3**(2), 133-140.
- Razavi, S.M.R., Miri, T., Barati, A., Nazemian, M. and Sepasi, M. (2015), “Industrial wastewater treatment by using of membrane”, *Membr. Water Treat.*, **6**(6), 489-499.
- Shokrkar, H., Salahi, A., Kasiri, N. and Mohammadi, T. (2011), “Mullite ceramic membranes for industrial oily wastewater treatment: Experimental and neural network modeling”, *Water Sci. Technol.*, **64**(3), 670-676.
- Suykens, J.A., Vandewalle, J.P. and de Moor, B.L. (2012), *Artificial Neural Networks for Modelling and Control of Non-linear systems*, Springer Science and Business Media, New York, U.S.A.
- Werner, J., Besser, B., Brandes, C., Kroll, S. and Rezwani, K. (2014), “Production of ceramic membranes with different pore sizes for virus retention”, *J. Water Process Eng.*, **4**, 201-211.
- Zhu, H.Y. and Zhou, M. (2014), “Air flotation method in the treatment of oily wastewater application of its progress”, *Adv. Mater. Res.*, **971-973**, 2044-2047.

CC

---

Nomenclature

---

ANN	Artificial Neural Network
B	Neuron bias
BR	Bayesian Regulation backpropagation
$f(x)$	Transfer function
I	Neuron input
LM	Levenberg-Marquardt backpropagation
MSE	Mean Squared Error
N	Number of data samples
O	Neuron output
PAC	Powdered Activated Carbon
R <sup>2</sup>	linear regression coefficient
$\omega$	Neuron weight
x	Individual sample data

---

Subscripts

---

Exp	Experimental data
Max	Maximum value
Min	Minimum value
Norm	Normalized data
Pred	Network-predicted data

---

**Appendix 1: Weights and biases of optimized ANNs for permeate flux prediction of mullite and mullite-alumina membranes in hybrid PAC-MF process**

Table A1 Weights between input and first layer of 10:6 network for mullite membrane

Input layer	Fist hidden layer									
	1	2	3	4	5	6	7	8	9	10
1	-1.930	0.700	-2.595	-0.247	0.009	-2.458	3.849	5.339	-4.393	-0.205
2	-3.777	-5.460	-5.086	4.382	-6.606	0.129	-3.712	-0.337	2.561	-4.532

Table A2 Weights between first a second layer and between second and output layer of 10:6 network for mullite membrane

First hidden layer	Second hidden layer						Second hidden layer	Output layer
	1	2	3	4	5	6		1
1	-1.301	-1.970	-1.099	0.076	-0.242	0.352	1	-14.457
2	-0.679	-3.732	-1.719	-2.582	2.617	-0.483	2	4.932
3	0.578	1.974	-1.753	0.873	-0.943	-0.984	3	-5.587
4	-2.430	-0.530	-1.376	-2.453	0.151	1.237	4	5.247
5	-2.713	2.450	-1.753	-1.665	-1.216	1.815	5	-2.393
6	-0.502	1.818	-1.049	-1.258	0.378	3.105	6	12.103
7	-0.432	1.123	-2.145	-2.018	0.219	0.106	-	-
8	-0.394	-0.022	3.809	-3.384	-0.463	0.904	-	-
9	-1.108	0.088	-2.055	-1.381	-1.469	0.515	-	-
10	-0.393	-1.237	-0.255	0.573	-0.372	0.704	-	-

Table A3 Neuron biases of 10:6 network for mullite membrane

Neuron	Hidden layer		Output layer
	1	2	-
1	4.484	-5.113	2.880
2	-5.113	2.880	-
3	2.880	2.624	-
4	2.624	-2.519	-
5	-2.519	2.048	-
6	-1.450	-	-
7	2.048	-	-
8	5.819	-	-
9	-3.892	-	-
10	4.322	-	-

Table A4 Weights between input and first layer of 9:20 network for mullite-alumina membrane

Input layer	Fist hidden layer								
	1	2	3	4	5	6	7	8	9
1	5.228	5.464	1.210	0.296	1.216	0.346	-1.090	0.743	4.353
2	-0.770	-3.226	-1.155	0.260	0.594	-4.196	-3.744	3.596	-0.727

Table A5 Weights between first and second layer of 9:20 network for mullite-alumina membrane

First hidden layer	Second hidden layer									
	1	2	3	4	5	6	7	8	9	10
1	2.223	2.293	0.214	0.827	1.529	0.759	0.914	1.377	0.725	0.283
2	4.088	1.864	4.302	1.177	1.178	0.555	2.086	0.889	0.504	0.500
3	1.672	0.788	0.134	0.963	-0.671	1.479	0.933	0.168	1.079	0.682
4	2.031	1.221	1.307	-0.208	0.354	-0.247	1.047	0.254	0.023	0.424
5	1.034	1.067	2.977	1.591	1.159	1.869	-0.439	0.086	0.209	0.118
6	-1.053	1.776	2.252	-1.057	0.686	-1.487	0.543	0.716	1.046	1.456
7	-0.305	0.692	-0.003	-0.853	0.223	0.552	0.905	0.541	0.916	0.869
8	4.059	1.853	2.057	-0.857	1.285	0.131	0.452	1.696	2.879	0.799
9	2.722	-0.781	1.160	0.400	0.878	0.452	-0.968	0.606	0.173	0.177
10	2.223	2.293	0.214	0.827	1.529	0.759	0.914	1.377	0.725	0.283
	11	12	13	14	15	16	17	18	19	20
1	0.282	0.282	0.282	0.282	0.283	0.282	0.283	0.283	0.283	0.283
2	0.501	0.502	0.502	0.501	0.500	0.501	0.500	0.501	0.501	0.501
3	0.683	0.683	0.683	0.683	0.683	0.683	0.682	0.683	0.683	0.683
4	0.425	0.426	0.426	0.425	0.424	0.425	0.424	0.424	0.424	0.424
5	0.117	0.117	0.117	0.117	0.118	0.117	0.118	0.118	0.118	0.118
6	1.457	1.459	1.459	1.458	1.456	1.458	1.455	1.456	1.456	1.456
7	0.869	0.869	0.869	0.869	0.869	0.869	0.869	0.869	0.869	0.869
8	0.798	0.796	0.796	0.797	0.799	0.797	0.799	0.798	0.799	0.799
9	0.176	0.175	0.175	0.176	0.177	0.176	0.178	0.177	0.177	0.177
10	0.282	0.282	0.282	0.282	0.283	0.282	0.283	0.283	0.283	0.283