

A New Artificial Network Approach for Membrane Filtration Simulation

J. Vivier and A. Mehablia*

CNRS-IAARC Centre National de la Recherche Scientifique 3,
rue Michel-Ange 75794 Paris cedex 16 – France

Original scientific paper

Received: October 14, 2011

Accepted: September 8, 2012

To improve traditional neural networks, the present research used the wavelet network, a special feedforward neural network with a single hidden layer supported by the wavelet theory. Prediction performance and efficiency of the proposed network were examined with a published experimental dataset of cross-flow membrane filtration. The dataset was divided into two parts: 70 samples for training data and 330 samples for testing data. Various combinations of transmembrane pressure, filtration time, ionic strength and zeta potential were used as inputs of the wavelet network to predict the permeate flux. The initial network led to a wavelet network model after training procedures with fast convergence within 30 epochs. Further, the wavelet network model accurately depicted the positive effects of either transmembrane pressure or zeta potential on permeate flux. Moreover, comparisons indicated the wavelet network model produced better predictability than the back-forward backpropagation neural network and the multiple regression models. Thus the wavelet network approach could be employed successfully in modeling dynamic permeate flux in cross-flow membrane filtration.

Key words:

Artificial Neural Network, membrane, simulation

Introduction

Applications of membrane technology in water and wastewater treatment have been growing steadily in recent years (Brindle and Stephenson,¹ Liao *et al.*,² Visvanathan *et al.*³). Many full-scale facilities are running with immersed microfiltration or ultrafiltration membranes mainly because membranes can offer higher efficiency in liquid-solid separation (Visvanathan *et al.*,³ 2000). However, the effectiveness of membrane separation is greatly affected by fouling the decline in permeate flux due to accumulation of colloidal matter, organic molecules, sparingly soluble inorganic compounds and microorganisms on membrane surfaces or in membrane pores (Defrance *et al.*,⁴ Song,⁵ Zhang and Song⁶). Fouling is an unavoidable deleterious phenomenon in membrane filtration and makes membrane technology less competitive in many applications. Therefore, understanding membrane fouling and developing controlling methods are crucial for the future of membrane technologies. To optimize the operation of membrane filtration, many modeling methods have been used to predict the permeate flux decline in recent years. Among these methods, theoretical models have been established using numerous factors, which include feed-water characteristics, membrane properties and operational conditions (such as trans-membrane pressure, cross-flow velocity and temperature) (Li and Elimelech⁷). The-

oretical models can produce highly detailed and complex descriptions of membrane fouling; however, they are computationally intensive and demand a high-level modeling expertise (Bowen *et al.*⁸). Moreover, the mechanisms of membrane fouling and the effect of key factors on fouling are still unclear or quite controversial (Tarleton and Wakeman⁹), which makes theoretical models designed with some necessary assumptions. Such assumptions result in models that are only valid for certain types of feed-water under certain conditions. In order to control membrane filtration effectively and easily, alternative methods are desired. Because available information often consists of measured data of several factors and permeate flux, the alternatives are expected to depend not on deep insight into complicated mechanisms, but on full utilization of available information. Furthermore, such an alternative must have the ability to determine the connections between input and output only by analyzing process data. Fortunately, data driven models, such as artificial neural networks (ANNs), could act as these alternatives. ANNs are mathematical algorithms that simulate the capacity of the biological brain to solve complex problems, and have the ability to approximate almost any function in a stable and efficient way (Chattopadhyay and Bandyopadhyay¹⁰). ANNs have already been extensively applied to modeling membrane filtration (Chen and Kim¹¹ Cheng *et al.*¹²). In all types of ANNs, feed-forward back-propagation neural networks (BPNNs) and radial basis function networks

*Correspondence Email: amine.mehablia@cnrs-iaarc.com

(RBFNs) have been proven the two most attractive methods for modeling membrane filtration until now (Al-Abri and Hilal,¹³ Aydiner *et al.*,¹⁴ Sahoo and Ray¹⁵). The BPNN was first used to predict the dynamic process of membrane fouling during cross-flow microfiltration of cane sugar. They studied the effect of the BPNN hidden structure and different distributions of training data on the predictability, and they obtained satisfactory accuracy. In following studies, BPNNs have been shown to be sometimes better suited and more useful for high nonlinear processes of membrane filtration (Chellam,¹⁶ Cinar *et al.*,¹⁷ Curcio *et al.*,¹⁸ Ni Mhurchu and Foley¹⁹). Unfortunately, there is a lack of reliable rules for the choice of the hidden structure of BPNNs and the common way is by trial and error. Moreover, initial parameters of BPNNs are often assigned random values, which cause slow convergence during the training process. On the other hand, using the experimental data of Fabish *et al.*,²⁰ Chen and Kim¹¹ found that RBFNs achieved better predictions than BPNNs. However, Sahoo and Ray¹⁶ pointed out that the results of Chen and Kim¹² were obtained through ANNs with un-optimized structures. With the same experimental data of Fabish *et al.*,²⁰ they showed that there was no apparent difference in predictability between BPNNs and RBFNs when the structures of both type of networks were optimized by genetic algorithms (GAs). Although the GA optimized networks may improve the predictability of ANNs, Cheng *et al.*¹² stated that the GAs are inclined to encounter the pitfall of overfitting when the number of neurons is too large because of their intrinsic randomness. Cheng *et al.*¹² presented a modified RBFN (MRBFN) which showed good convergence resulting from a well initialized structure. They constructed the MRBFN by choosing neurons from a set of candidate neurons, which were predetermined according to uniform partitions of the domain of interest at different levels. However, their method could encounter the problem of dimension curse when the input dimension increases, because the increasing dimension leads to an exponential increase of the uniform partitions of the domain. In particular, the dimension curse becomes more severe when higher levels are carried out in the case of higher input dimension. Despite the vast efforts to improve the efficiency of ANNs in membrane processes as above, there are still some problems, for instance, lack of specific methods or the danger of encountering lower efficiency to determine the initial network structures and the initial parameters. Therefore, a more efficient type of ANNs will be preferred to model the complex process of membrane filtration. Wavelet networks have shown to be a promising alternative to traditional ANNs (Zhang and Benveniste²¹) and

many studies describe wavelet networks as a powerful tool for function approximation (Zhang and Benveniste,²¹ Zhang *et al.*²²), system identification (Billings and Wei,²³ Sjoberg *et al.*²⁴), automatic control (Sanner and Slotine,²⁵ Oussar *et al.*²⁶), etc. Such network is a special 3-layer feed forward network as shown in Fig. 1 which is calculated as follow:

$$\hat{y} = \sum_{i=1}^s \psi(x) \quad (1)$$

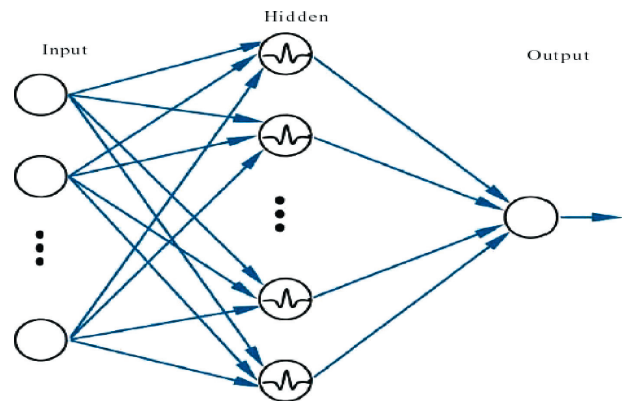


Fig. 1 – The structure of wavelet network

Where, \hat{y} is the output; x is the input; ω is the connecting weight; ψ is the hidden-layer neuron; s is the number of hidden neurons. As a distinguishing characteristic, ψ is a wavelet function and is often called a wavelon. The wavelet is a function that satisfies an admissibility condition and it has a property of compact support, namely, nonzero values only in a finite domain (Daubechies²⁸). A single wavelet $\psi(x)$ called a mother wavelet can generate family of wavelets by dilating and translating itself as following:

$$\left\{ \psi_{m,n}(x) = \alpha^{\left(\frac{1}{2}\right)^{dn}} \psi(\alpha^{-n}x - m\beta) : n \in Z, m \in Z^d \right\} \quad (2)$$

Where α and β are the dilation and translation step sizes (typically $\alpha = 2$, $\beta = 1$), respectively, m is the translation index and n is the dilation level. Each wavelet in the family (2) has finite support. Support refers to the region where the function is nonzero. The wavelet function of the family (2) can exhibit a support that is compact or rapidly vanishing (Daubechies²⁷). The wavelet family can be used as tools for function analysis and synthesis. The concept of a wavelet family is very similar to a set of sine functions at different frequencies used in Fourier analysis. The family (2) is used as a set of candidate wavelets that constitute the original form of eq. 1, i.e., the initial wavelons are chosen from the

family (2). For a set of training data, only part of wavelets in the family (2) are useful for constituting the wavelet network model because most wavelets may cover no data due to their compact support (Zhang²⁸). In addition, the set of training data in practical cases is sparse, and this sparseness leads to a further reduction in the number of candidate wavelets in family (2). Clearly, both the wavelet compact support and the sparseness of training data make the selection of initial wavelets in eq. 1 more convenient. Once a given number of initial wavelets have been chosen, the initial value of the connecting weight (ω) can be determined by linear regression (Zhang²⁸). Compared with traditional ANNs, wavelet networks obviously offer a highly efficient approach for constructing the initial form of networks. Many studies have exhibited the high efficiency of wavelet networks according to different constructing methods, such as the orthogonal least square (OLS) algorithm and backward elimination algorithm (Oussar and Dreyfus,²⁹ Zhang²⁸). In particular, Zhang²⁸ has reported that wavelet networks converged fast during training processes because of their better initialization, and wavelet networks already offered a comparable accuracy even in their initial form. Specific constructing methods, better initialization of internal parameters and fast convergence, all of them make wavelet networks more suitable than traditional ANNs for modeling complex nonlinear processes. Although the great benefit of wavelet networks has been successfully applied to many fields, any wavelet network application to membrane systems has not been reported yet. In this paper, the wavelet network approach was used to model a process of cross-flow membrane filtration based on the published data set of Bowen *et al.*⁸ The purpose of the current investigation was to demonstrate the high efficiency of a wavelet network; to study its predictability of permeate flux under various operating conditions and to compare its capability with those of BPNN and the conventional multiple regression (MR) method.

Materials and methods

Experimental data

This study used the experimental data published in Bowen *et al.*³⁰ In the experiments, spherical silica colloids with a mean particle diameter of 65 nm were used to perform cross-flow membrane ultrafiltration at five different combinations of pH and ionic strength (I). For each combination of pH and I , the permeate flux was measured under five different transmembrane pressures (ΔP) as shown in Fig. 4. In this paper, I , ζ , ΔP and time (t) acted as the input of wavelet networks to predict the dy-

namic permeate flux. Out of 400 data samples, 70 samples were chosen as training data such that more samples were located in regions of greatest curvature, as shown in Fig. 3. The other 330 samples were used as testing data. Prior to being put in the wavelet network, the values of all experimental data were normalized between -1 and 1 as below:

$$x_{norm} = \frac{(x - x_{cent})}{x_{half}} \quad (3)$$

Where, $x_{cent} = (x_{max} + x_{min})/2$ is the center of variable Interval, $x_{half} = (x_{max} - x_{min})/2$ is the half of interval length, x_{max} is the maximum value of a variable and x_{min} is the minimum value of a variable. On the other hand, a corresponding denormalization was done to achieve a reasonable permeate flux after an output of the network was attained. The normalization method was derived from Z-score normalization (Priddy and Keller³⁰). Such normalization can minimize bias within the wavelet network for one feature over another. It can also speed up training time by starting the training process for each feature within the same scale.

Constructing and training the wavelet network

According to eq. 1, a so-called Mexican hat wavelet (Zhang²⁸) was chosen as a mother wavelet with the following form:

$$\psi(x) = (d - \|x\|^2) e^{-\|x\|^2/2} \quad (4)$$

Where, d is the dimension of x , $\|x\|^2 = xx^T$. Then a family of candidate wavelets was determined by the mother wavelet and the training data. Then, further development was done by selecting different numbers of wavelets from the candidate family according to the OLS algorithm and then, the number of wavelets was determined by the generalized cross-validation (GCV) criterion. The GCV, a modified form of cross-validation, is a popular method for choosing the smoothing parameter. An important early reference is Craven and Wahba.³¹ The GCV criterion consists of estimating the expected performance of the model evaluated with fresh data based on the required data (Zhang²⁸). Thus an initial form of the wavelet network model was achieved. In the end, the initial wavelet network was trained to get the final wavelet network model. The detailed steps are listed below:

1) Definition of wavelet support: For $\psi_{m,n}(x)$ in the family (2), denoted by $S_{m,n}$ its support, i.e.,

$$S_{m,n} = \{x \in R^d : |\psi_{m,n}(x)| > 0.662 \max |\psi_{m,n}(x)|\} \quad (5)$$

2) Generating the family of candidate wavelets: With the definition of $S_{m,n}$, each training point x_k among N training points is examined within five dilation levels ($n = 0, 1, 2, 3, 4$) to find the wavelets whose supports contain x_k . Then the index set of the found wavelet, I_k , is defined by:

$$I_k = \{(m, n) : x_k \in S_{m,n}\} \quad (6)$$

The union of I_k for all x_k in the family (2) gives the indices of candidate wavelets whose supports contain at least one training data point. This results in the family of candidate wavelet

$$W = \{\psi_{m,n} : (m, n) \in I_1 \cup I_2 \dots I_N\} \quad (7)$$

Let L be the number of wavelets in W . For convenience of the following presentation, the double index (m, n) is replaced by a single index $j = 1, 2, \dots, L$, i.e.,

$$W = \{\psi_1, \psi_2, \dots, \psi_L\} \quad (8)$$

3) Determination of the initial wavelet network: The initial wavelet network is achieved by a hybrid algorithm (Zhang²⁸) where, the OLS algorithm determines the relative optimal combination of a certain number of candidate wavelets and then the GCV criterion points out the most reasonable combination of a certain number of candidate wavelets. The hybrid algorithm can be done as follows:

Algorithm:

$$I = (1, 2, \dots, L)$$

$$p_j = \psi_j \text{ for all } j \in I;$$

$$l_0 = 0, q_{l_0} = 0;$$

Begin-loop

For $i = 1 : L$ $p_j = p_j - (\psi_j^T q_{l_{i-1}}) q_{l_{i-1}}$, for all $j \in I$;

$$I = I - \{j : p_j = 0\};$$

If I is empty, set $s = i - 1$ and break the loop;

$$l_j = \arg \max_{j \in I} (p_j^T y)^2 / (p_j^T p_j);$$

$$\alpha_{ji} = \psi_{l_i}^T q_{l_j} \text{ for } j = 1, \dots, i - 1;$$

$$\alpha_{ii} = \sqrt{p_{l_i}^T p_{l_i}};$$

$$q_{l_i} = \alpha_{ii}^{-1} p_{l_i};$$

$$\tilde{\omega}_{l_i} = q_{l_i}^T y;$$

Which is an upper triangle matrix

$$A_i = \begin{bmatrix} \alpha_{11} & \dots & \alpha_{1i} \\ \vdots & \ddots & \vdots \\ 0 & \dots & \alpha_{ii} \end{bmatrix}$$

$$\begin{bmatrix} \omega_{l_1} \\ \vdots \\ \omega_{l_i} \end{bmatrix} = \begin{bmatrix} \tilde{\omega}_{l_1} \\ \vdots \\ \tilde{\omega}_{l_i} \end{bmatrix} \times A_i^{-1}$$

$$\hat{y}_i(x) = \sum_{i=1}^i \omega_{l_i} \psi_{l_i}(x)$$

which is the resulting wavelet network;

$$GCV_i = \frac{1}{N} \sum_{k=1}^N (\hat{y}_i(x_k) - y_k) \left(1 + \frac{2s}{N}\right);$$

$$I = I - \{l_i\};$$

End loop:

$$s = \arg \min_{i \in \{1 \dots L\}} GCV_i.$$

As a result, the initial wavelet can be achieved with the form of eq. 1.

4) Training the wavelet network: A common backpropagation training method is applied to fit the training data, as well as possible in this paper. The details about back-propagation method can be found in Haykin.³² The performance of the training process is evaluated with normalized square root of mean square (NSRMSE)

$$NSRMSE = \frac{\sqrt{\frac{1}{K} \sum_{k=1}^K (y_k - \hat{y}_k)^2}}{\sqrt{\frac{1}{K} \sum_{k=1}^K (y_k - \bar{y})^2}} \text{ with } \bar{y} = \frac{1}{K} \sum_{k=1}^K y_k.$$

In addition, the NSRMSE was used as the stopping criteria for training ANN, and it was also a performance index of model prediction in this paper.

Results and discussion

The wavelet network model

By examining each training input x_k consisting of $I, \zeta, \Delta P$ and t , a family of 152 candidate wavelets was achieved and each of the wavelets had a support covering at least one training input. By comparison, the number of candidate wavelets would be: $(2^4)^0 + (2^4)^1 + (2^4)^2 + (2^4)^3 + (2^4)^4 = 69905$.

If the candidate wavelets were attained by the technique of uniform partition of input space that was described in Cheng *et al.*¹² Note that the exponent 4 of 2^4 in parenthesis is the dimension of input x_k . Clearly, the number of 69905 wavelets would be a dimension curse. Therefore, even if generating the 69905 wavelets was not tedious, the big number would certainly cause higher cost of computation in the following selection of network neurons. However, with the number of 152 candidate wavelets in this paper, the proposed method was not very sensitive to the input dimension of training data. In fact, such insensitivity resulted from the compact support of wavelets and the sparseness of training data, both of which were useful in avoiding the superfluous wavelets to be included in the candidate family.

Fig. 2 shows the varying GCV with respect to the number of selected wavelets according to the OLS algorithm. The GCV criterion was applied to the number of the selected wavelets, which was a model order determination problem. This criterion of minimum GCV avoided the danger of overfitting the training data. The minimum GCV was attained at the number 12, i.e., the combination of the chosen 12 wavelets were used to constitute the initial wavelet network with a relative optimal approximation. Because it would be very tedious to tell the optimal combination of a certain number of wavelets from all the subsets of the same size of the candidate family, the chosen 12 wavelets were determined by stepwise selection method. This method first selected one wavelet which was optimal for fitting the training data, then the second one such that it was optimal while cooperating with the first selected one, then the third one which was similarly selected, and so on. In this way, the stepwise selection by the OLS algorithm successfully made a trade-off between optimality and efficiency.

The obtained initial network fitted the training data with an NSRMSE of 0.103. Such smaller

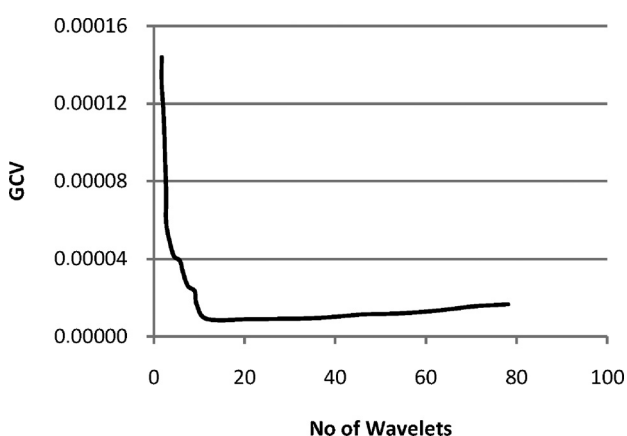


Fig. 2 – GCV vs selected wavelets

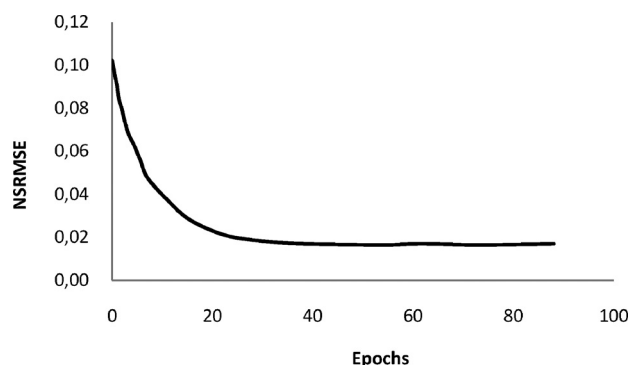


Fig. 3 – NSRMSE vs epochs

NSRMSE agrees with Zhang²⁸ who reported that the OLS strategy could offer satisfactory accuracy. Consequently, the OLS strategy and the GCV criterion could lead to an efficient constructing procedure and a better initialization for the wavelet network. With the initial wavelet network, the common backpropagation was used as the training method. During the training procedure, NSRMSE drastically dropped during the first 30 epochs (Fig. 3). Clearly, the training process converged so fast even in an ordinary back-propagation training method. The fast convergence verified again the better initialisation of the wavelet network approach.

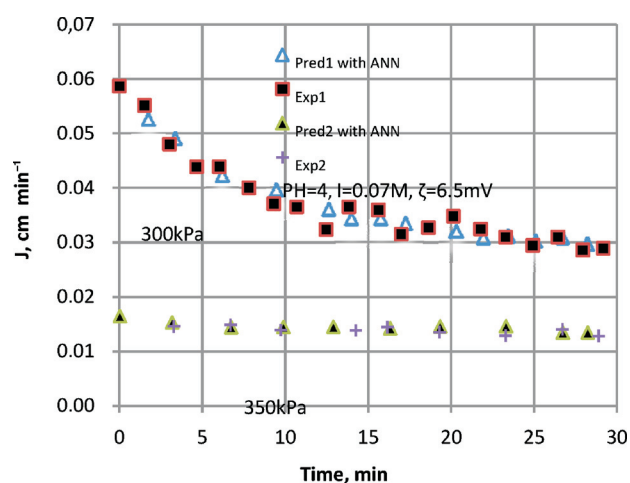


Fig. 4 – Effect of physico-chemical and hydrodynamic parameters on the flux

With the initial wavelet network, the common back-propagation was used as the training method. During the training procedure, NSRMSE drastically dropped during the first 30 epochs (shown in Fig. 3). Clearly, the training process converged so fast even in an ordinary back-propagation training method. The fast convergence verified again the better initialization of the wavelet network approach.

Effect of I , ζ and ΔP on permeate flux

Fig. 4 shows the measurements and the predictions of the wavelet network model for permeate flux during cross-flow ultrafiltration of silica colloids at five pressures for each of the five combinations of pH and I . The wavelet network model excellently offered predictions for both training and testing data. Moreover, the nonlinearity of the flux time profiles was well reproduced by the wavelet network. As shown in Fig. 4, the predictions, as well as the measurements accurately described permeate flux behavior greatly changing with the variations of ΔP , pH, and I . The same plot could have been generated for each scenario but it has been confined to one case to show the trend. For each combination of pH and I , the permeate flux was proportional to ΔP and the ultimate flux at higher transmembrane pressure was higher. These phenomena are due to the greater driving force for permeate flux by higher ΔP (Faibish *et al.*²⁰). Meanwhile, higher ΔP led to the steeper flux decline at the earlier stage of the filtration. This trend is attributed to the higher convective mass of colloids toward the membrane surface and a more densely packed cake layer at higher pressure (Faibish *et al.*²⁰). On the other hand, for each combination of pH and I , the lowest P was observed with the least flux decline for both experimental data and predictions. This observation is an index of operation approaching the critical flux below which colloids do not deposit on the membrane surface (Faibish *et al.*³⁰). The predictions also accurately described the positive effect of ζ on permeate flux. Also, an increasing flux followed a decreasing I with the same pH and, in the latter, the enhancement in permeate flux was preceded by an increasing pH with the same I . In fact, both of the above operations led to an increasing ζ (not considering the sign) of colloids (Bowen *et al.*⁸). The enhancements in permeate flux by ζ in Fig. 4, all the results are in qualitative agreement with the published reports by Huisman *et al.*³³ and McDonogh *et al.*³⁴). These authors explained the effect of ζ qualitatively by reasoning that high zeta potentials increase inter-parti-

cle repulsion, thus causing less deposition (thinner cake layers) and more permeable cake layers.

Consequently, the thinner and more permeable cake layers lead to an enhancement of permeate flux. Such an enhancement became more significant in this paper because of the colloidal particles with a smaller mean diameter 65 nm. Faibish *et al.*²⁰ have reported that for particles with diameters smaller than about 100 nm, the effect of interaction repulsion on permeate flux becomes dominant. Among all the predictions and experiments, the combination of pH = 9, I = 0.030 M and ζ = -80.5 mV led to the highest permeate flux for each ΔP in earlier stage. Besides, the effect of the highest ζ , lower I also enhanced the permeate flux for case E, which is in accordance with the report by Elzo *et al.*³⁵ who stated that lower I corresponds to larger distances between particles and such distance results in more permeable cake layers. The predictions in case E agree well with the theoretical explanation. All the case scenarios are listed below.

Case A: pH = 4, I = 0.072 M, ζ = 6.3 mV (Fig. 4)

Case B: pH = 4, I = 0.030 M, ζ = 17.9 mV

Case C: pH = 4, I = 0.0077 M, ζ = 24.0 mV

Case D: pH = 7, I = 0.031 M, ζ = 46.7 mV

Case E: pH = 9, I = 0.030 M, ζ = 80.5 mV

Performance comparison

This section describes the superiority of the wavelet network over BPNN under the same modeling conditions. The BPNN model used the tanh function as hidden neurons. Moreover, a comparison with the conventional MR method was used to show the better initialization of the wavelet network. Two MR models were built with the identical training data. One is a linear regression equation with the following form:

$$J = 2.488 \cdot 10^{-2} - 2.401 \cdot 10^{-1} \{I\} - 2.230 \cdot 10^{-4} \{\zeta\} + 1.376 \cdot 10^{-4} \{\Delta P\} - 8.353 \cdot 10^{-4} \{t\}.$$

Table 1 – Comparisons of R^2 and NSRMSE for different models

Model type	R^2		NSRMSE	
	training data	testing data	training data	testing data
Initial wavelet network	0.990	0.685	0.685	0.610
The wavelet network model	0.997	0.965	0.965	0.196
The BPNN model	0.997	0.914	0.068	0.294
The linear MR model	0.752	0.800	0.494	0.480
The nonlinear MR model	0.809	0.848	0.433	0.492

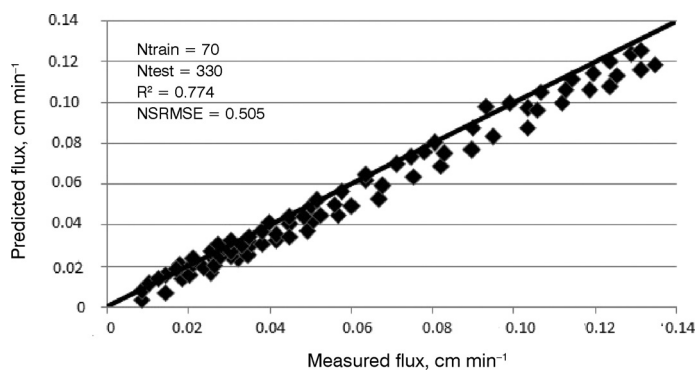


Fig. 5 – The initial wavelet network approach

Another is a nonlinear multiple equation with the following form:

$$\begin{aligned}
 J = & 2.956 \cdot 10^{-2} - 2.312 \cdot 10^{-1} \{I\} - \\
 & - 2.235 \cdot 10^{-4} \{\xi\} + 1.467 \cdot 10^{-4} \{\Delta P\} - \\
 & - 2.747 \cdot 10^{-4} \{t\} + 6.288 \cdot 10^{-5} \{t^2\}.
 \end{aligned}$$

Coefficients of both equations were proven significant by the *t*-statistic. Several other multiple polynomial regression models were also tested, but their coefficients showed worse significance by the *t*-test. Table 1 shows the coefficients of multiple determination R^2 and the NSRMSE for permeate flux obtained with the initial wavelet network, the wavelet network model, the BPNN model, the linear MR model and the nonlinear MR model. The wavelet network model not only provided better results in fitting, but also the best in prediction, with a R^2 of 0.997 for the training data and 0.965 for the testing data. The BPNN model provided results much better than the MR models. The nonlinear MR model slightly improves in the basis of the linear MR model, but its prediction ability is not comparable to the BPNN and wavelet networks. This supposes that the existing high non-linearity of membrane filtration is a challenge for the MR method. In particular, it can be stated that the NSRMSE obtained by the wavelet network model is much smaller than that obtained with the other three methods. The superiority of the wavelet network model in fitting and prediction resulted from its better initialization, because the initial wavelet network has already provided better results with a R^2 of 0.990 and an NSRMSE of 0.103 for the training data. Moreover, the number of wavelons of the initial wavelet network, which was evaluated by GCV criterion, led to good generalization performance of the wavelet network model with an NSRMSE of 0.196 for the testing data.

Fig.4 (case A) shows the predicted versus measured flux for the initial wavelet network, the wavelet network model, the BPNN model, and the MR

models. Most of the data shown in case B fall within the lines with $\pm 10\%$ relative error. Only 12 out of 330 predicted testing data lie outside the region enclosed by the dotted lines, indicating that 96.4 % of the data are within the $\pm 10\%$ relative error range. In case C, 81 out of 330 predicted testing data is not confined within the two dotted lines. In case D, only 40 % of the data lie within the $\pm 10\%$ relative error range, which is mainly because the complex nonlinear membrane filtration process could not be presented by a simple linear method.

Moreover, the simple linear description led to 9 negative results. Although the nonlinear MR model was slightly better than the linear MR model, it provided 13 negative predictions, which suggest that the MR method could not rationally depict the membrane filtration.

Conclusion

The present study shows that a wavelet network approach could be used to predict permeate flux decline of cross-flow ultrafiltration of colloidal suspensions as a function of operating conditions. The wavelet network approach provided the benefit of efficient constructing procedures by fully utilizing the sparseness of training data and the compact support property of wavelets. In particular, in case of multiple input dimensions (4 dimensions in this paper), this approach could avoid the curse of dimensionality for the choice of hidden neurons. Moreover, a better initialization with an NSRMSE of 0.103 by the OLS algorithm and GCV selection criterion led to fast convergence within 30 epochs during the training procedure. The wavelet network model excellently described the nonlinear variation of permeate flux under different operating conditions. The predictions described the positive effect of ΔP on permeate flux, and moreover, an increasing ξ was followed by an enhancement in permeate flux. Such an accurate prediction ability of wavelet networks could certainly be helpful in optimizing practical operations of membrane filtration. Further, the wavelet network model offered such satisfactory accuracy that almost 96.4 % of predictions for the testing data deviated measured data within the $\pm 10\%$ relative error range. Meanwhile, the comparisons of the performances of the initial wavelet network, the wavelet network model, the BPNN model, the linear MR model, and the nonlinear MR model confirmed the superiority of wavelet networks.

As it is shown, efficient construction, better initialization, faster convergence and higher accuracy, all have proved that the wavelet network is a promising alternative to traditional ANNs in model-

ing complex membrane filtration processes. As a modeling tool, wavelet networks could be used for both observation of membrane system performance and evaluation of experimental conditions. In addition, this modeling technique could be applied as a simulation tool to improve the operating conditions of other water and wastewater treatment systems that involve highly nonlinear processes.

Nomenclature

- \hat{y} – output
 x – input
 ω – connecting weight
 ψ – hidden-layer neuron
 α and β – dilation and translation step sizes d : dimension of x
 L – number of wavelets
 NSRMSE – performance index of model prediction
 s – number of hidden neurons
 $x_{cent} = (x_{max} + x_{min})/2$ – center of variable
 $x_{half} = (x_{max} + x_{min})/2$ – half of interval length

References

1. Brindle, K., Stephenson, T., *Biotech. Bioeng.* **49** (6) (1996) 701.
2. Liao, B. Q., Kraemer, J. T., Bagley, D. M., *Crit. Rev. Environ. Sci. Tech.* **36** (6) (2006) 48–9.
3. Visvanathan, C., Ben Aim, R., Parameshwaran, K., *Crit. Rev. Environ. Sci. Tech.* **30** (1) (2000) 1.
4. Defrance, L., Jaffrin, M. Y., Gupta, B., Paullier, P., Geaugey, V., *Biores. Tech.* **73** (2) (2000) 105.
5. Song, L. F., *J. Membr. Sci.* **139** (2) (1998) 183.
6. Zhang, M. M., Song, L. F., *Environ. Sci. Tech.* **34** (17) (2000) 3767.
7. Li, Q. L., Elimelech, M., *Environ. Sci. Tech.* **38** (17) (2004) 4683.
8. Bowen, W. R., Mongruel, A., *Chem. Eng. Sci.* **51** (18) (1996) 4321.
9. Tarleton, E. S., Wakeman, R. J., *Chem. Eng. Res. Des.* **71** (4) (1993) 399.
10. Chattopadhyay, S., Bandyopadhyay, G., *Switzerland. Int. J. Remote Sens.* **28** (20) (2007) 4471.
11. Chen, H. Q., Kim, A. S., **192** (1–3) (2006) 415.
12. Cheng, L. H., Cheng, Y. F., Chen, J. H., *J. Membr. Sci.* **308** (1–2) (2008) 54.
13. Al-Abri, M., Hilal, N., *Chem. Eng. J.* **141** (1–3) (2002) 27.
14. Aydiner, C., Demir, I., Yildiz, E., *J. Membr. Sci.* **248** (1–2) (2005) 53.
15. Sahoo, G. B., Ray, C., *Predicting J. Membr. Sci.* **283** (1–2) (2006) 147.
16. Chellam, S., *J. Membr. Sci.* **258** (1–2) (2005) 35.
17. Cinar, O., Hasar, H., Kinaci, C., *J. Biotech.* **123** (2) (2006) 204.
18. Curcio, S., Calabro, V., Iorio, G., *J. Membr. Sci.* **286** (1–2) (2006) 125.
19. Ni Mhurchu, J., Foley, G., *J. Membr. Sci.* **281** (1–2) (2006) 325.
20. Faibish, R. S., Elimelech, M., Cohen, Y., *J. Colloid Interf. Sci.* **204** (1) (1998) 77.
21. Zhang, Q. H., Benveniste, A., *IEEE T. Neural Network* **3** (6) (1992) pp 889.
22. Zhang, X. Y., Qi, J. H., Zhang, R. S., Liu, M. C., Xue, H. F., Fan, B. T., *Comput. Chem.* **25** (2) (2001) 125.
23. Billings, S. A., Wei, H. L., *A IEEE T. Neural Networ.* **16** (4) (2005) pp 862.
24. Sjöberg, J., Zhang, Q., H., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P. Y., Hjalmarsson, H., Juditsky, A., *Automatica* **31** (12) (1995) 1691.
25. Sanner, R. M., Slotine, J. J. E., *Int. J. Contr.* **70** (3) (1998) 405.
26. Oussar, Y., Rivals, I., Personnaz, L., Dreyfus, G., *Neurocomputing* **20** (1–3) (1998) 173.
27. Daubechies, I., *Ten lectures on wavelets*, SIAM, Philadelphia, (1992) 11.
28. Zhang, Q. H., technical report of Linköping University, LiTH-ISY-I1423. October (1992).
29. Oussar, Y.; Dreyfus, G., *Neurocomputing* **34** (1) (2000), 131.
30. Priddy, K. L., Keller, P. E., *Artificial neural networks: An introduction*, SPIE Press, (2005) 15.
31. Craven, P., Wahba, P., *Numer. Math.* **31** (4) (1979) 377.
32. Haykin, S., *Neural networks: A Comprehensive Foundation*, 2nd. Ed., Prentice Hall, (1999) 183.
33. Huisman, I. H., Dutre, B., Persson, K. M., Tragardh, G., *Desalination* **113** (1) (1997) 95.
34. McDonogh, R. M., Fane, A. G., Fell, C. J. D., *J. Membr. Sci.* **43** (1) (1989) 69.
35. Elzo, D., Huisman, I., Middelink, E., Gekas, V., *Colloid Surf. Physicochem. Eng. A.* **138** (2–3) (1998) 145.