Desalination and Water Treatment
www.deswater.com
2 (2009) 170–183
1944-3994/1944-3986

Studies on the application of different ANNs to predict permeate flux in rotating disk membrane modules: A case study with MATLAB™

Chiranjib Bhattacharjee[a]*, Dwaipayan Sen[a], Projjwal Sarkar[a], S. Datta[a], P.K. Bhattacharya[b]

[a]*Department of Chemical Engineering, Jadavpur University, Kolkata 700 032, India*
Tel: +91 92 3056 2975; Fax: +91 33 2414 6007; email: cbhattacharyya@chemical.jdvu.ac.in
[b]*Department of Chemical Engineering, Indian Institute of Technology, Kanpur 208 016, India*

Received 28 April 2007; Accepted 28 July 2008

ABSTRACT

Stirred rotating disk membrane modules (RDMM) have been found to provide maximum flux enhancement due to minimization of the concentration polarization layer by shear effect. But unfortunately rigorous theoretical modeling of RDMM has not been developed yet, which greatly affects the simulation and optimization of this device under applications in many fields. In this perspective, the work reported in this article was carried out to develop an efficient artificial neural network (ANN) capable of predicting permeate flux in case of RDMM. Different ANNs, like Feedforward, Elman, Hopfield and Radial Basis Network, were investigated to find the optimum network configuration and corresponding network parameters have been estimated. As a case study, the focus was given to ultrafiltration (UF) of casein whey, a complex protein mixture. For this purpose the UF of whey was carried out in RDMM at different pH and transmembrane pressures, with and without rotating the membrane. A detailed comparative study was made on the validation results between the predicted flux and corresponding experimental values for different ANNs investigated. Based on the results, it was concluded that the Radial Basis Network was most accurate in terms of flux prediction and stability for UF using RDMM.

Keywords: Rotating disk membrane module; Ultrafiltration; Flux; pH; Transmembrane pressure; Neural network

1. Introduction

Membrane separation involves partially separating a feed containing a mixture of two or more components by use of a semipermeable barrier (the membrane) through which one or more of the species moves faster than another or other species [1]. In this paper the emphases have been given on the ultrafiltration (UF) membrane separation in the whey treatment process using a rotating disk membrane module (RDMM), and the subsequent investigation for optimum artificial neural network (ANN) configuration. UF is a pressure-driven membrane technique that uses porous membranes for the separation of material in the 1×10^{-9} m–10×10^{-6} m size range, or stated otherwise, compounds with molecular weights in excess of 1,000 [2]. One of the major problems that is associated with any confectionery industry is discarding waste-water which contains much valuable protein such as α-lactalbumin, β-lactoglobulin, etc. So the primary necessity is to build up a treatment scheme so that all the proteins from this casein whey can be recovered and the treated water can be reused or recycled. In a recent communication [3], fractional UF using RDMM, followed by ion-exchange membrane chromatography, was suggested for partial recovery of different whey proteins. RDMM is an

*Corresponding author.

efficient device in terms of its hydrodynamics, but the major problem lies in the lack of an available rigorous theoretical model, making simulation of UF performances and further scale-up difficult.

The UF membrane process is commonly attributed to mainly two different problems: one is membrane fouling [4] and the other one is concentration polarization [5]. Fouling occurs when the membrane is physically obstructed either by a build-up of particulates on the surface or by membrane compaction, whereas concentration polarization refers to the formation of a concentration boundary layer adjacent to the membrane, characteristics of any pressure-driven membrane separation process, which results in a local increase in the osmotic pressure and the reduction of the permeate flux.

To alleviate this problem, new module designs are being investigated such as spiral-wound modules, vibrating shear enhanced process systems (VSEP), and rotating disk or rotating disk stack modules [6]. Recently, Luhui et al. [7] claimed that RDMM could be the most efficient device showing high permeate flux. In this membrane module the generated concentration boundary layer or gel layer is swept away by the rotation of stirrer on the membrane and by the rotation of membrane itself. So in this module flux declination is less comparable to any other modules.

To describe the polarization phenomena in UF, several models were developed so far. Usually all of them could be classified into the following three categories: (a) resistance-in-series model, (b) gel polarization model and (c) osmotic pressure model. According to the gel polarization model [8], a gel layer is formed on the membrane surface, limited by solubility limit of the solute. As shown by Wijman et al. [9], the three models mentioned above predict almost equivalent permeate fluxes under a steady-state condition, especially at higher concentrations. On the other hand, it was shown by Danes et al. [10] that no classical model as stated above could explain the experimental data with good fit. Due to this reason, most of the models that are reported in literature are usually composite in nature, not based directly on any classical models as listed above.

Due to lack of a proper mathematical model for describing flux declination, particularly in case of RDMM due to its highly complex hydrodynamics, a general approach has been made in this article by introducing the concept of the Artificial Neural Network (ANN). This technique is quite different from its peer Artificial Intelligence (AI), which is a very common name in the world of computer gaming programs. ANN is a training-based network. The experimental points are used for the training of the network. If the network could be trained properly and accurately, then the outcome of the network will resemble the real world. So it should be a vivid realization to all of us that if the network is not trained properly, i.e. if the supplied information to the network is not accurate, then there could be a large margin of error between the expected result and simulated result.

The work reported in this article was carried out to simulate permeate flux for the UF of casein whey in RDMM by using ANN, and subsequently, to investigate the best network configuration. The main objective was to establish a generic model, i.e. ANN, to study the hydrodynamics of a rotating disk membrane module and to study the performance for different types of networks, leading to identification of most optimized and efficient network. For this purpose networks that were used to do a comparative study are the Elman, [11], Feedforward [12], Radial Basis [13] and Hopfield [14]. MATLAB-6.5 ™ (Release 13, Mathworks) was used for carrying out all the investigations. In the case of the Feedforward and Elman networks, the number of hidden layers and number of nodes in the hidden layer were also varied in order to observe the variations in the network outputs. The main reasoning to choose casein whey as the feed for RDMM is because of the complex nature of the protein mixture; it offers an additional dependence on solution pH, which affects the permeate flux. The alteration of the pH was found to have a strong and complex relation with permeate flux because of its relation with isoelectric point, as well as due to some conformational changes (isomerization), mostly of β-lactoglobulin, a major constituent of whey proteins. The treatment objective in this study was separation and purification of whey proteins from casein whey. For this, we adopted a two-stage methodology, pretreatment (centrifugation and microfiltration), followed by UF. We used a rotating disk module, a high shear device for minimizing concentration polarization and consequent fouling. It is expected that if the proposed ANN could result in good prediction for this complex feed stream and can account for the variation of pH, transmembrane pressure (TMP) and membrane rotation, then it could be possibly treated as good validation.

2. Theoretical

The main purpose of this study is to make a comparison between different neural networks. Two different types of networks were investigated. One used a supervised learning algorithm and the other an unsupervised learning one. The Hopfield Network is an example that uses unsupervised learning and the Feedforward Network is based on supervised learning.

The basic network structure (Fig. 1) for any ANN is the Feedforward Network with a back-propagation gradient descent algorithm. Although the error back-propagation algorithm (EBP) [15] has been a significant milestone in

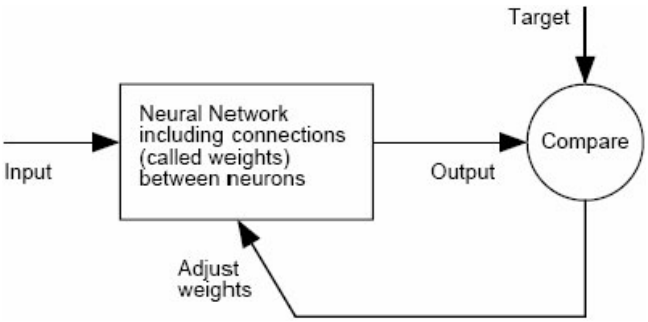

Fig. 1. Schematic representation of an Artificial Neural Network, which resembles a feedback system.

neural network research, it has been known as an algorithm with a very poor convergence rate. The most recent development occurring in this feedforward network solution is optimization of the network by the Levenberg–Marquardt optimization technique. It is widely accepted as the most efficient one in the sense of its accuracy level [19].

Another type of network structure in ANN, called Recurrent Networks, nowadays is drawing lot of interest. It is a kind of network which exhibits the same input–output behavior [20]. Two important networks in this category are Elman and Hopfield. The Elman Network is a two-layer back-propagation network with the addition of the feedback connection from the output of the hidden layer to its input. The Elman Network can be considered in such a fashion that the input layer is as if divided into two segments. The name of this extra segment is called a context unit. It is connected with the hidden layer with a one-to-one node connection [21]. So the number of hidden units is equal to the number of context units. Hopfield neural networks have been widely used as associative memory or to solve optimization problems [22]. There are many versions of the Hopfield networks [14,23–26]. Basically the Hopfield Network follows a feedback mechanism. The network is recursive in that the output is fed back as the input once the network is in operation.

A radial basis function (RBF) [27] is another type of feed-forward ANN. Typically in an RBF network, there are three layers: one input, one hidden and one output layer. In RBF networks, one major advantage is that if the number of input variables is not too high, then learning is much faster than other type of networks. However, the required number of the hidden units increases geometrically with the number of the input variables. It becomes practically impossible to use this network for a large number of input variables.

3. Experimental

A series of UF runs were performed with an objective to obtain the permeate flux as a function of different operating conditions dominating in UF of casein whey using a rotating disk membrane module. The major parameters considered were membrane rotation speed, TMP and solution pH.

3.1. Materials

Casein whey was obtained from local sweet-meat industries situated in and around Kolkata, India. The pH of the raw casein whey varied from 3 to 4, depending upon the quantity of excess acid present in the whey resulting from acid caseination. The sweet-meat industries, in most of the cases, used hydrochloric acid or its equivalent for casein precipitation. Coomassie Brilliant Blue (G-250) for Bradford protein assay (Pierce Biotechnology, Rockford, USA), was obtained through Hysel India (New Delhi, India). Sodium hypochloride (NaOCl), sodium hydroxide (NaOH) (used for membrane cleaning) and ethanol (used for membrane storage) were purchased from Merck (Mumbai, India).

3.2. Feed pretreatment

In order to prevent any possibility of membrane fouling, the suspended casein particles and colloidal matter (mostly fat) were removed by centrifugation followed by microfiltration. Centrifugation was carried out in a research centrifuge model TC 4100D (Remi, Mumbai, India) with a speed of 12,500 rpm, giving a relative centrifugal force of 156.8 N for a period of 30 min. After centrifugation, the sample was subjected to microfiltration (MF) using an all glass vacuum filtration unit (Sartorius, Göttingen, Germany), fitted with an oil-free portable vacuum pump (Sartorius, Göttingen, Germany, model ROC 300 with moisture trap), with a polyether sulfone (PES) membrane (0.047 m diameter, pore size 0.45 $\times 10^{-6}$ m) as the filter media. The permeate from the MF was adjusted for pH by adding calculated amount of 1 N hydrochloric acid or 1 N sodium hydroxide, as required, to produce the feed for the subsequent UF run. The isoelectric point of casein whey was 4.6 [28]. Accordingly, the pH was set to 2.8 and 5.5 to facilitate the study of UF performance below and above the isoelectric point of casein whey.

3.3. Membrane and module

UF of pretreated casein whey was carried out batch wise in a stirred rotating disc module using the PES membrane. The module, made of SS316, was manufactured by Gurpreet Engineering Works (Kanpur, India) as per the specified design. The module (Fig. 2) was equipped with two motors with speed-controllers to provide rotation of the stirrer (not used in this study) and membrane housing.

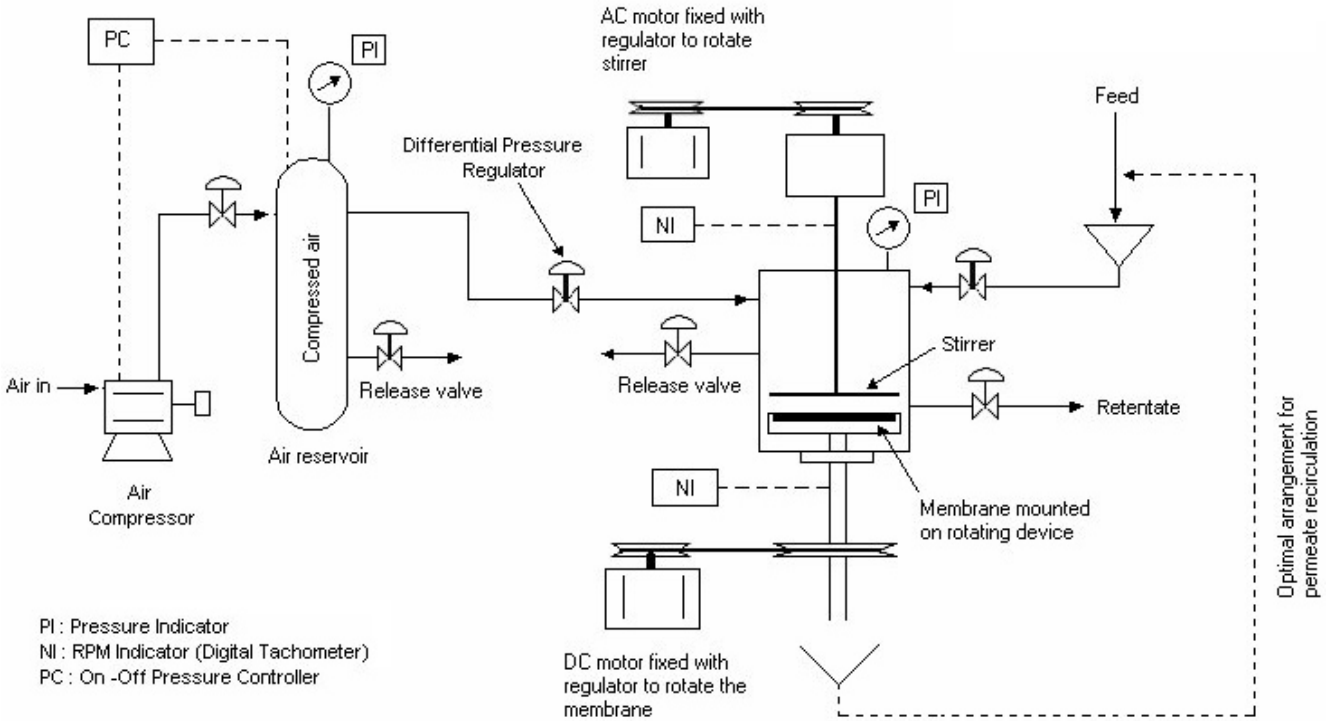

Fig. 2. Schematic diagram of the experimental set-up.

A digital tachometer was used to measure the rotational speed of the membrane. The set-up was equipped with an arrangement for recycling the permeate to the feed cell to run it in continuous mode with constant feed composition. The later mode was not investigated in this study. The module was also equipped with an outside water-jacket, through which provision for water circulation was available. An adequate mechanical sealing mechanism was provided to prevent leakage from the rotating membrane assembly up to a pressure of 1 MPa. An air compressor was used to provide compressed air for pressurization of the cell. An intermediate air reservoir fitted with on-off controller based on pressure sensor was provided which maintains the pressure within reservoir between 1–1.2 MPa. A differential pressure regulator was used to set the pressure at the desired level within the module. The complete schematic diagram of the rotating disk module set-up is given in Fig 2. The PES membranes (flat disk of 0.076 m diameter) of 30 kg/kmol molecular weight cut-off (MWCO) were imported from Millipore (Bedford, USA) through its Indian counterpart (Millipore India). The flat-sheet membrane operable in a pH range of 1–14 has an actual diameter of 0.076 m, whereas the effective diameter was 0.056 m.

3.4. Membrane compaction and water run

Prior to experiments, the membrane was subjected to compaction for about an hour with ultra-pure water at a pressure of 0.9 MPa, higher than the highest operating pressure to prevent any possibility of change of membrane hydraulic resistance during UF. Once the water flux becomes steady with no further decrease, it was concluded that full compaction of the membrane has taken place. The membrane was washed thoroughly with distilled water after every run with casein whey to remove any deposited fouling layer, which was followed by water runs to determine the extent of fouling. The water fluxes obtained from such studies were found to be within 2% of initial water flux, thus showing minimum fouling resulting from the proposed separation scheme. The ultra-pure deionised water used in this study was obtained from the Arium 611DI ultrapure water system (Sartorius, Göttingen, Germany). The feed to this Arium 611DI was taken from a usual laboratory distillation unit.

3.5. Methodology

Experiments were carried out batch wise in the rotating disc membrane module at different operating conditions, starting each time with an initial pretreated feed volume of 350×10^{-6} m^3. The temperature within the cell was maintained at 30°C using water circulation through outside jacket on the module (not shown in Fig. 2) from a constant temperature bath (Julabo constant temperature batch/circulator, model FK30/31-ME). The objective of this study was to get complete dependence of flux on solution pH, membrane rotation speed and TMP.

Accordingly, the solution pH was varied at 2.8 and 5.5 and the TMP was fixed at 0.4, 0.5, 0.6, and 0.7 MPa. The membrane rotation speed was set at 0 (fixed membrane), 300 rpm and 600 rpm. The membrane rotation speed was controlled through a speed controller, regulating the power supply to motor. The membrane rotational speed was checked with the help of a digital tachometer working on stroboscopic principle during experiment. All these constituted total 24 runs, each of nearly 30 min duration, about 60% of which have been used for training of different networks and the remaining ones were used for model validation. After each experiment, the membrane was thoroughly washed for 20 min under running water, then soaked in deionised water for 6 h and then again thoroughly washed for 20 min. In each case, the water flux was regained by about 98% of its original value, suggesting the cause of flux decline to be either osmotic pressure limited or due to a reversible fouling layer.

4. Results and discussion

Using mainly membrane separation of casein whey, 86% of valuable proteins are recovered from the discarded water from the sweet-meat industries, which can be used for different purposes later on. Another side of this treatment scheme is that BOD and COD values are reduced by almost 99% of their initial values, which are well within the permissible limit of Pollution Control Board and can be discarded safely. The main purpose of this paper was a thorough comparative study of whey flux with different types of networks and to reach a conclusion on the performance of different networks. The Feedforward Network is the simplest network in ANN. The Elman Network resembles both the characteristics of the Feedforward Network and a recurrent network. A comparative study of these two networks is given in Table 1. The mean square error (MSE) [29] comparison curves for these two networks are shown in Fig. 3. From the analysis of Table 1 it is observed that for learning rate 0.000044 and momentum constant 0.60, the MSE for the Elman network is lowest, i.e., 9.67846×10^{-6} and the MSE for the Feedforward Network is 9.95536×10^{-6} at the cost of the number of epochs. Again, for the Feedforward Network the minimum value for MSE is obtained for a learning rate of 0.000020 and a momentum constant of 0.10. In this case the MSE is lowest and the number of epochs is 1027, which is also less tthan he number of epochs obtained for the previous case where the Elman Network is optimized. But we are compromising with the learning rate and momentum constant value which will guide the speed of the simulation process towards the minimum error value. Thus, to make this simulation process fast, we have considered the learning rate value of 0.000044 and momentum constant of 0.60 where the MSE that is obtained for the Elman Network is the lowest between all the MSEs that are obtained for any kind of network. Thus it can be considered for the optimum network parameters for this system. With these parameters a comparative study was made between the Elman and Feedforward Networks, shown in Fig. 4. Convenient ±10% error lines are drawn. If the points that are obtained are within this margin of error, then it can be concluded that the network is well simulated. From this curve, it is very clear that the Feedforward Network was not optimized for a learning rate of 0.000044 and momentum constant of 0.60.

In case of the Elman network the basic parameters for simulating the network are learning rates, momentum constants and number of nodes in the hidden layers. The basic difference between the Elman and Feed Forward Networks is that in case of the Elman Network, the

Table 1
Comparative study of MSE and Epoch obtained for the Elman and Feedforward Networks with 100 nodes in the hidden layer and with the *traingdx* training function

Learning rate	Momentum constant	MSE × 10^6		Epoch	
		Elman	Feedforward	Elman	Feedforward
0.000044	0.60	9.67846	9.95536	3871	345
0.000020	0.10	9.81543	9.04807	6109	1027
0.000020	0.20	9.98809	9.98467	6353	2047
0.000020	0.23	9.91746	9.98007	3133	3809
0.000020	0.25	9.88521	9.98096	3732	5530
0.000020	0.15	9.97764	9.98489	4698	4754
0.000020	0.14	9.98617	9.98933	6049	1755
0.000010	0.10	9.94225	9.94809	5521	2075
0.000015	0.10	9.91343	9.91230	6643	1710
0.000020	0.60	9.93943	9.95281	3128	3384

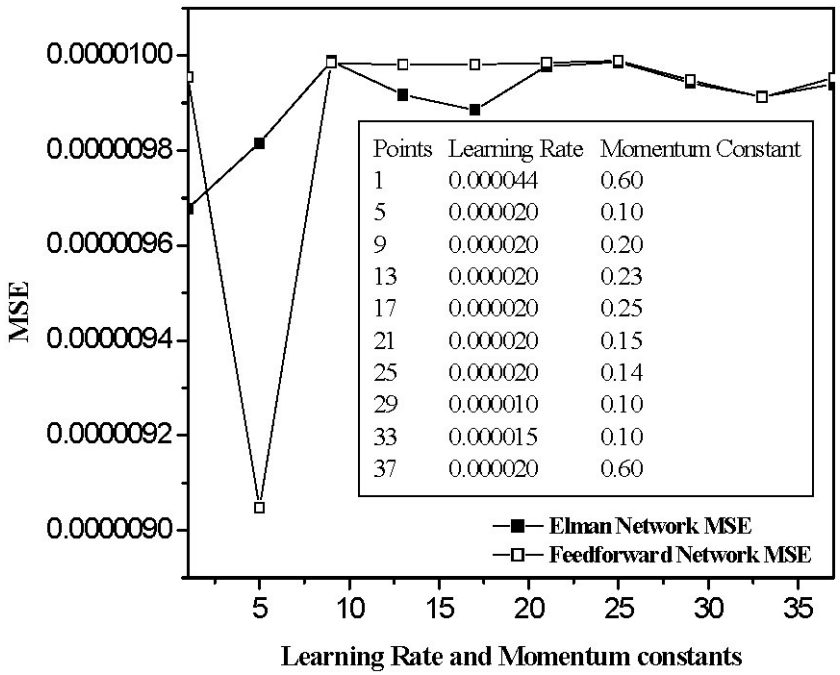

Fig. 3. Comparative study of MSE and Epoch obtained for the Elman and Feedforward Networks with 100 nodes in the hidden layer and with the *traingdx* training function.

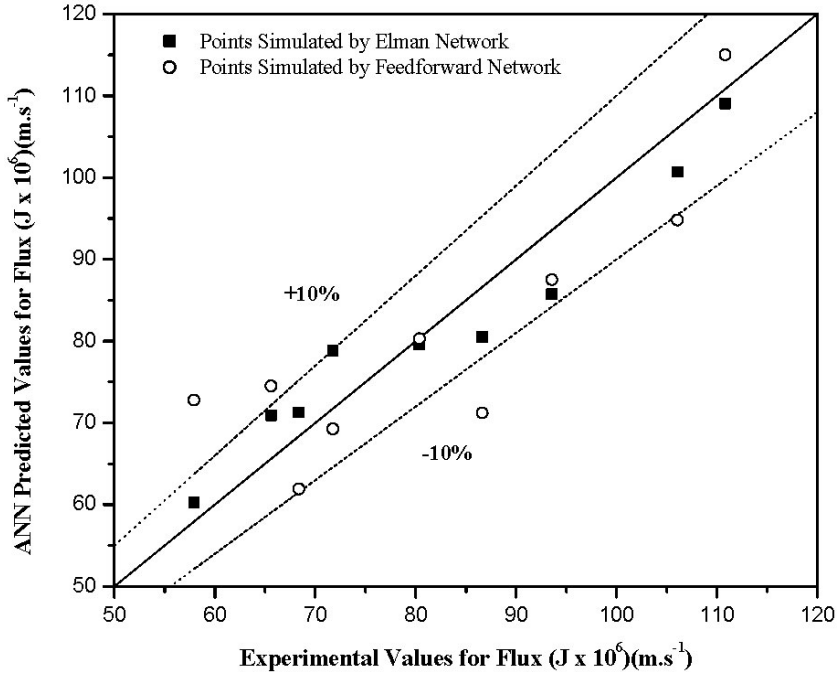

Fig. 4. Comparative study between Elman Network, Feedforward Network and experimental points.

output from the first layer is supplied as feedback to the first layer again. Thus the first layer of the Elman Network has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Thus even if two Elman networks, with the same weights and biases, are given identical inputs at a given time step, their outputs will be different. The network is trained for the data set given in Table 2. The transfer function that is used for the hidden layer is *tansig* and the transfer function that is used for the output layer is *purelin*; the *traingdx* function is used to train the network.

Table 2
Network input parameters and target training values

pH	TMP (MPa)	rpm[a]	Target values[b] for network training
2.8	0.4	0	68.74
5.5	0.4	0	56.15
5.5	0.5	0	59.20
5.5	0.5	100	69.04
2.8	0.5	0	81.47
2.8	0.5	100	97.53
2.8	0.4	100	79.71

[a]Indicates membrane rotation. 0 = no rotation; 100 = rotation.
[b]Indicates whey flux ($\times 10^6$ $m^3.m^{-2}.s^{-1}$) obtained experimentally at these specified conditions.

The following parameters that are set for this network: number of nodes: 100, learning rate: 0.000044, momentum constant: 0.60, and MSE: 0.00001. Interestingly, what we found is that if the network is given the same input and simulated weights and biases, then there is a chance that the network will respond differently. This has happened because of the changed feedback state. For the above parameter set-up, it was found that at the initial moment the network did not converge and the MSE calculated for this case was equal to 59.1958. But for the second time when the network was assigned to obtain weights and biases, the network converged and for this case the network converged with MSE 9.67846×10^{-6}. From this it is clear that the network differs with the feedback state. Fig. 5 shows the comparison between two training iterations. Fig. 6 shows the comparative study of whey flux for different learning rates and momentum constants.

The above simulation made for the Elman and Feedforward Networks is based on a single hidden layer. Figs. 7 and 8 show the performance of the Feedforward and Elman Networks respectively on the basis of two and hidden layers, respectively. Fig. 9 shows a comparative study of Elman and Feedforward networks for optimized network condition with two hidden layers. In this case, the simulated values were chosen for which the network showed small MSE. For the Elman Network the number of nodes in the first hidden layer was taken as 75 and the number of nodes in the second hidden layer was taken as 25.

In the case of the Feedforward Network two types of training functions were used. One was *traingd* and the other *traingdx*. In the case of the Feedforward Network for a single hidden layer, the number of nodes varied at 100, 50 and 90. It is found that the network predicted the flux near to the experimental values at learning rate equal to

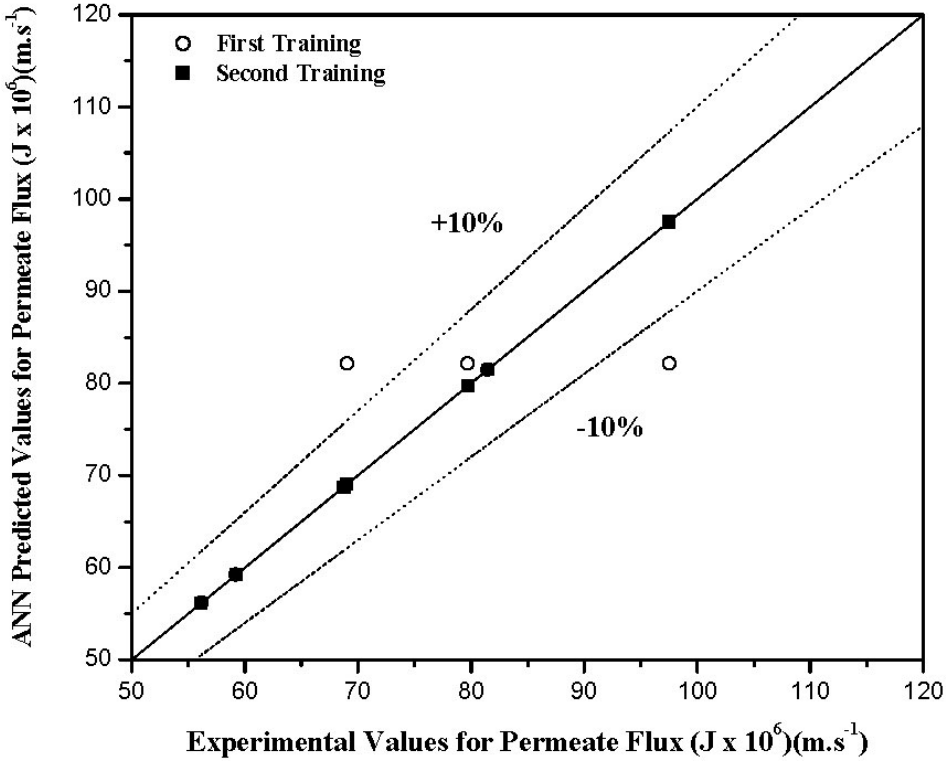

Fig. 5. Curve shows that even with assigning the same layer weights and biases, the network output can be different. Here First Training shows the initial training and Second Training shows the training after making an assignment to the layer weights and biases.

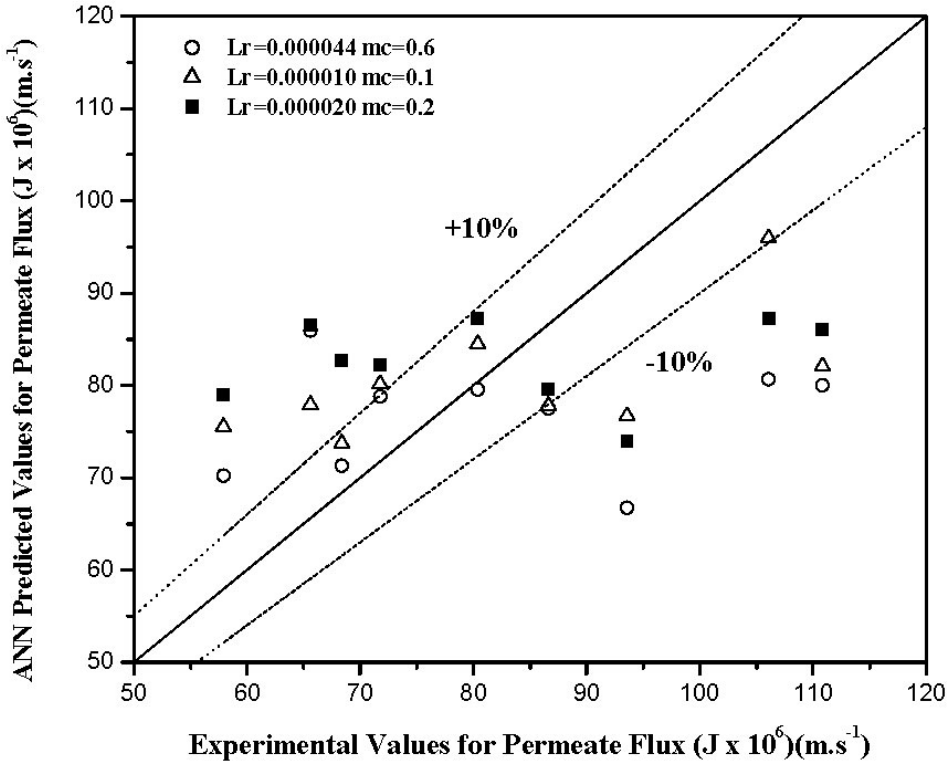

Fig. 6. Curve shows the comparative study of whey flux simulated by the Elman Network for different momentum constants and learning rates.

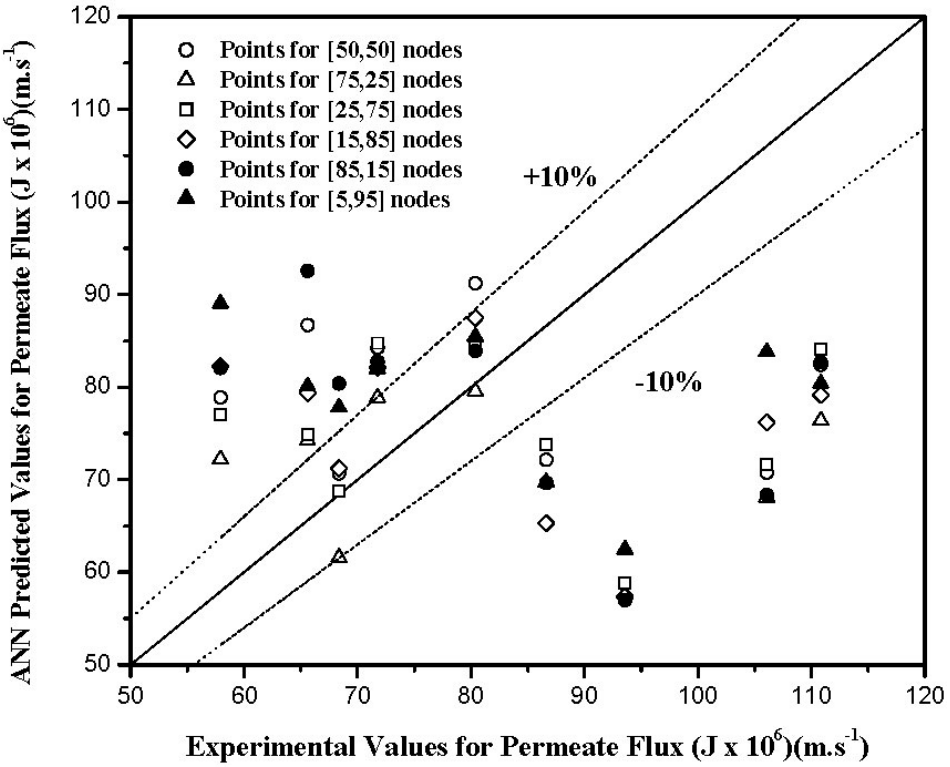

Fig. 7. Difference between simulated points obtained in the Elman Network for two hidden layers keeping different numbers of nodes in the respective layers. In the tile, the numbers within the third bracket show the number of nodes in the hidden layer; e.g., [25,75]: number of nodes in first hidden layer is 25, number of nodes in second hidden layer is 75.

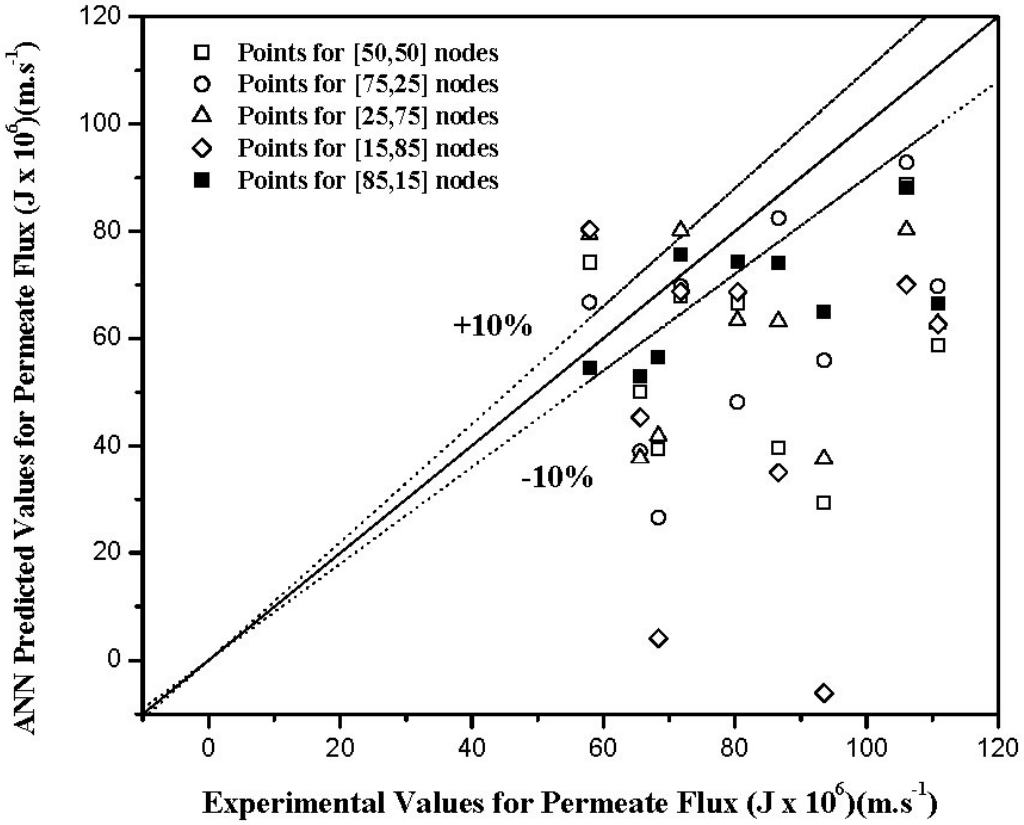

Fig. 8. Difference between simulated points obtained in the Feedforward Network for two hidden layers keeping different numbers of nodes in the respective layers. In the tile the numbers within the third bracket show number of nodes in the hidden layer.

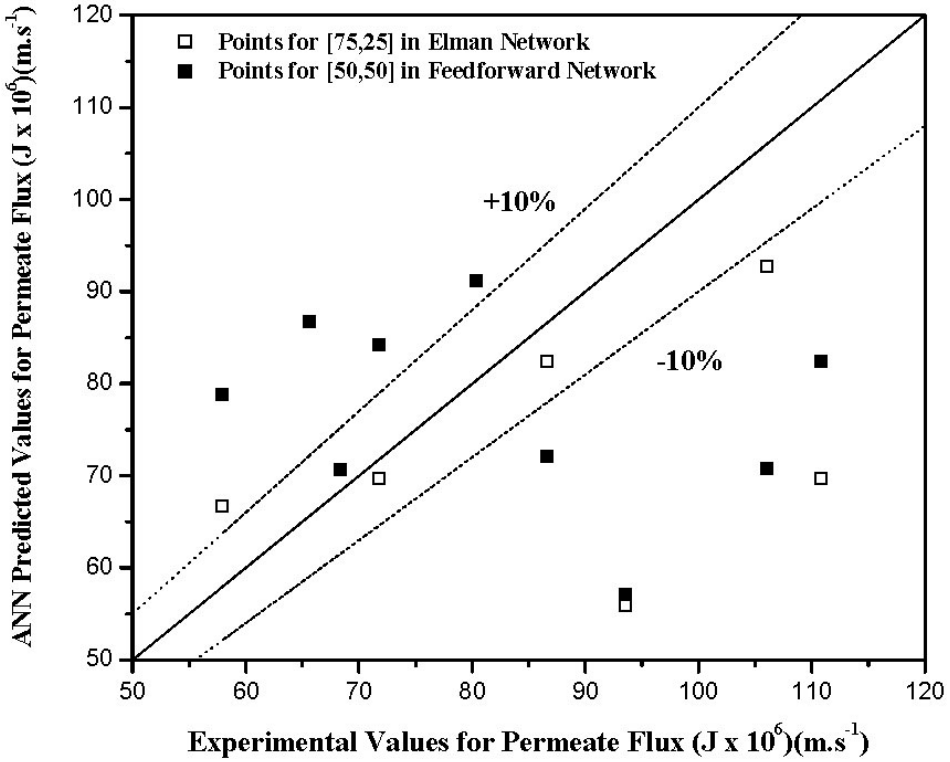

Fig. 9. Comparative study of Elman Network and Feedforward Network for the two hidden layer system.

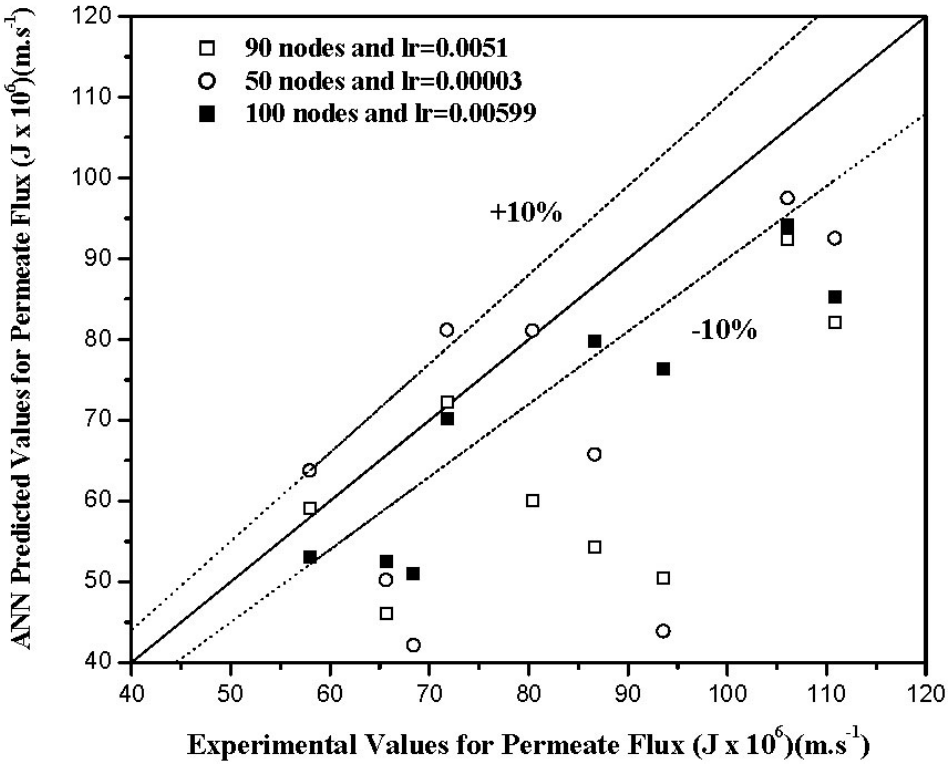

Fig. 10. Comparative study of various simulated points for Feedforward Network with the *traingdx* function and different number of nodes in the hidden layer and learning rates.

0.00599 taking training function *traingdx* for 100 nodes in the hidden layer. For 50 and 90 nodes in the hidden layer we did not obtain any results close to the experimental values, but for 50 nodes in the hidden layer we obtained simulated values quite near to the experimental values for a learning rate equal to 0.00003 and for the training function *traingdx*. For 90 nodes, the values were considered for a learning rate equal to 0.0051 with the same training function *traingdx*. The comparative curve is shown in Fig. 10. For 90 nodes the MSE was equal to 0.0046218. For 50 nodes the MSE was equal to 9.92735×10^{-6} and that for 100 nodes was 9.97659×10^{-6}. Epochs taken for these three cases were 619,802 and 294 respectively.

Another training function that was used for this Feedforward Network is *traingd*, but it is found that this training function was not going to be satisfactory. The simulated values obtained using this training function are quite far from the experimental points.

In the case of the RBF network, two types of Radial basis functions, *newrbe* and *newrb*, were used to train the network. The basic difference between these two types of functions is for the first case, neurons in the hidden layer are generated in equal number with input vectors. For the second case the neurons are generated with process. How far the simulation is going to be accurate depends on a constant in the Radial Basis Network called SPREAD. Here fixing goal, i.e. MSE was set equal to 0.002, SPREAD was taken as adjustable parameter to find out for which SPREAD the sum square error (SSE) is minimal. Different values of SPREAD were taken (1, 2, 3, 4 and 5).

It was found that SSE was minimal for SPREAD equal to 4. Fig. 11 shows the variation of the flux with different SPREADs. It gives a vivid picture of how the simulated values were close enough to the target value for SPREAD = 4. Therefore, this SPREAD is that value for which the input will have a maximum overlapping region. As previously stated, with the *newrb* function of the Radial Basis Network, the *radbas* neuron is generated online. If that MSE is reached below the set goal value (here 0.002) or the number of added neurons exceeds the default number (25) of the *radbas* neuron, then the generation of the network will stop. From the output of the generated network, the number of *radbas* neurons was found to be 6, whereas as per exact design of the Radial Basis Network the number of *radbas* neurons would be exactly equal to the number of input vectors. Here in the following matrix called "layer", the layer weight was initially assigned between different *radbas* neurons and input nodes in the network. It was observed that initial layer weights are equal to the supplied input values. Here six *radbas* neurons were formed instead of seven *radbas* neurons (supplied input data set). This is exactly what is featured by the *newrb* function:

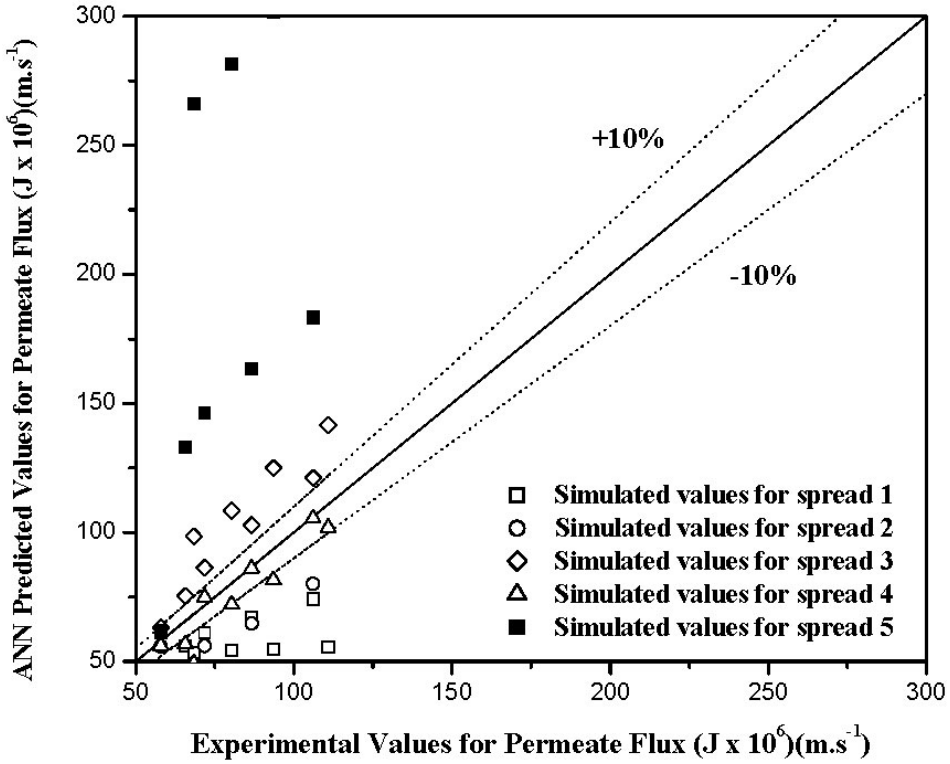

Fig. 11. Deviations of the simulated values by the *newrb* network from the target values for different SPREADs.

$$\text{layer} = \begin{bmatrix} 2.8200 & 5.0000 & 100.0000 \\ 2.8200 & 5.0000 & 0 \\ 2.8200 & 4.0000 & 100.0000 \\ 2.8200 & 4.0000 & 0 \\ 5.5500 & 4.0000 & 0 \\ 5.5500 & 5.0000 & 0 \end{bmatrix} \quad \text{bias} = \begin{bmatrix} 0.2081 & 0.2081 & 0.2081 & 0.2081 & 0.2081 & 0.2081 \end{bmatrix}^T$$

From the bias values it is clear that biases were set as per the formula $b = \frac{0.8326}{4} = 0.2081$. The number of initial layers and biases gives an idea about the number of *radbas* neurons that can be in the *radbas* layer. Customarily, the algorithm of the radial basis is that the *radbas* neurons are formed equal in number to the number of input vectors. So it was expected that the number of rows for biases will be exactly equal to the number of input vectors. In this work the simulation was completed with six *radbas* neurons.

In the first case, when the first *radbas* neuron was created, the layer weight was assigned equal to the following values of pH, TMP and rotational speed: pH = 2.8, TMP =0.5 MPa, rotational speed = 300 rpm. In the case of the Radial Basis Network with the *newrbe* function, the network generates the number of *radbas* neurons exactly equal to the number of input vectors. For the problem under consideration, we had seven input vectors, as follows:

$$p = \begin{bmatrix} 2.80 & 5.50 & 5.50 & 5.50 & 2.80 & 2.80 & 2.80^{*} \\ 4.00 & 4.00 & 5.00 & 5.00 & 5.00 & 5.00 & 4.00^{\#} \\ 0 & 0 & 0 & 100.0 & 0 & 100.0 & 100.0^{\$} \end{bmatrix}$$

where * is pH, # is TMP (MPa) and $ is the rotation indicator. Here SPREAD was taken as 2.42, and therefore the bias for the *radbas* was set up as $b = \frac{0.8326}{2.42} = 0.344$. The initial weights for the *radbas* neurons were set up numerically equal to the input vectors,

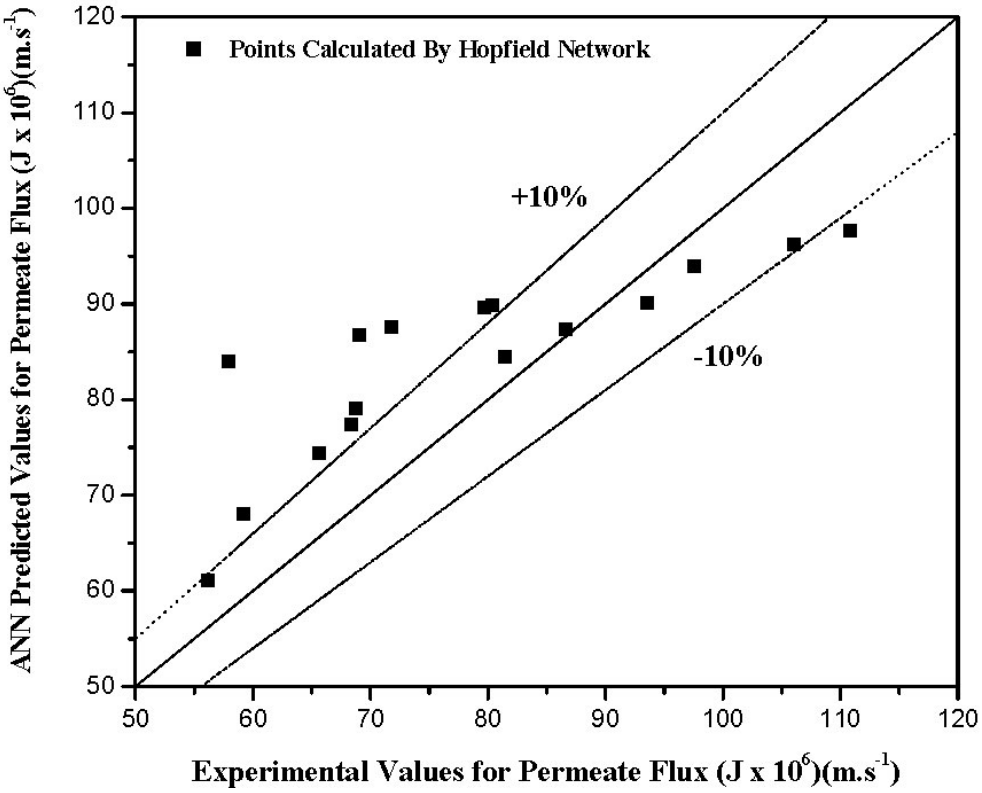

Fig. 12. Accuracy of the network. (Hopfield Network was not at all near to the points obtained from the experiments.)

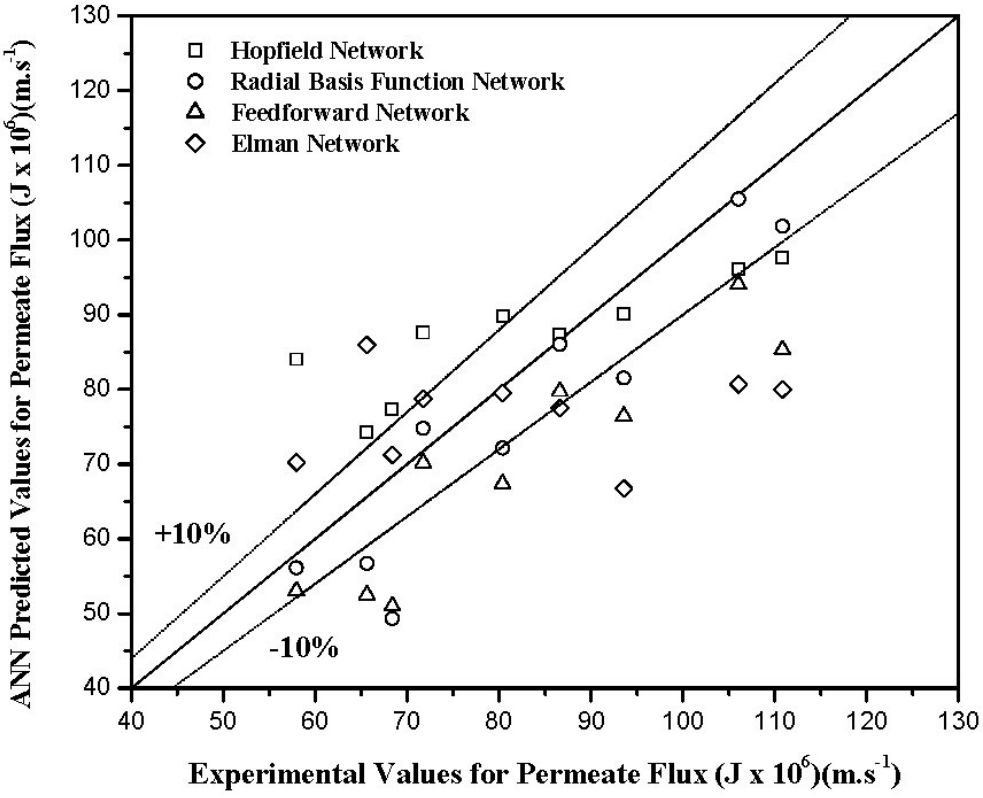

Fig. 13. Comparative study of predicted whey flux with a different network architecture.

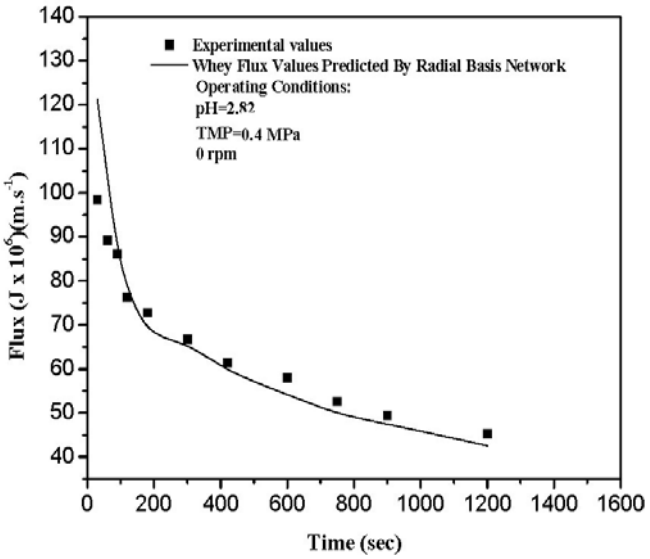

Fig. 14. Comparative study between whey flux obtained experimentally and predicted whey flux with the Radial Basis Network.

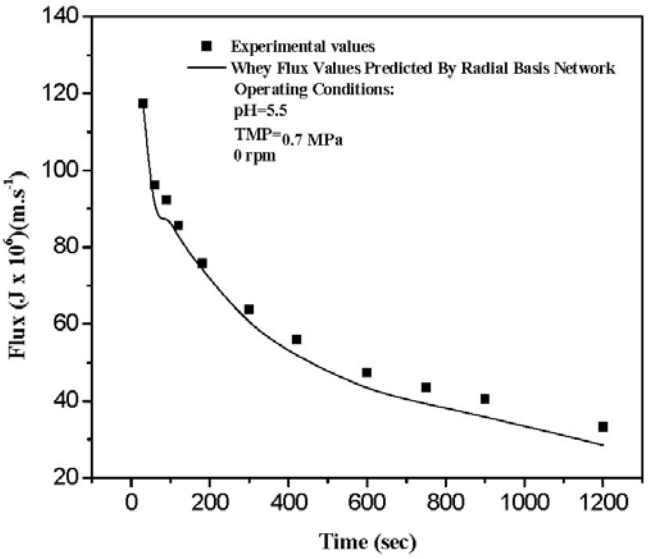

Fig. 15. Comparative study between whey flux obtained experimentally and predicted whey flux with the Radial Basis Network.

Table 3
Absolute average deviation (AAD) for different networks

pH	TMP (MPa)	rpm	AAD for Feedforward Network (%)	AAD for Radial Basis Function Network (%)	AAD for Elman Network (%)	AAD for Hopfield Network (%)
2.8	0.4	0	15.80	6.90	28.50	41.8
5.5	0.7	0	25.20	7.30	28.50	8.0

$$\text{layer}=\begin{bmatrix} 2.800 & 4.0000 & 0 \\ 5.500 & 4.0000 & 0 \\ 5.500 & 5.0000 & 0 \\ 5.500 & 5.0000 & 100.0000 \\ 2.800 & 5.0000 & 0 \\ 2.800 & 5.0000 & 100.0000 \\ 2.800 & 4.0000 & 100.0000 \end{bmatrix}$$

So it implies that the number of *radbas* neurons are exactly equal to the number of input data set.

Now in case of Hopfield Network the result that we obtained are not as accurate as we are getting for Radial Basis Network. The graphical results after simulation with Hopfield Network are shown in Fig. 12.

It is evident that among all the networks, Radial Basis Network reproduces experimental points with highest accuracy. Fig. 13 gives a comparative study of all the networks. Here the values of the network are taken for optimized network parameters. The figure vividly shows the idea of the network which is going to produce the most accurate estimation of experimental values. So from the above discussion, it could be assessed that the Radial Basis Network predicts almost accurate flux values for the different operating parameters under consideration.

Now it would be interesting to see whether this network can predict the actual flux decline profile or not. For this purpose, the Radial Basis Network with function *newrb* and with SPREAD = 4 were taken to predict the flux decline profile. Figs. 14 and 15 show a comparative flux decline profile between the experimental values and the network predicted values for operating conditions of pH= 2.8, TMP = 4 and pH = 5.5 and TMP = 7, respectively, without membrane rotation. It was found that the profile is almost identical with the profile that we obtained through the experiment.

Further, to elucidate the nature of deviations between the simulated and experimental values as obtained from different ANNs, absolute average deviations obtained from all such networks considered in this study under fixed operating conditions (pH = 2.8; TMP = 0.4 MPa;

rpm = 0 and pH = 5.5; TMP = 0.7 MPa; rpm = 0) are shown in Table 3. It is evident that accurate prediction of UF flux under this case paved the way for considering the Radial Basis Network as an optimized network for simulating complex UF behaviour.

5. Conclusions

The objective of this paper was to find optimum ANN architecture for predicting whey flux for UF of casein whey, a complex protein mixture in a RDM module, having complex hydrodynamics. To accomplish this, different neural network architectures of feedback and feedforward types were considered. It was found that among different networks with varying network parameters, the Radial Basis Network is the best network that can simulate the flux with the highest accuracy. Moreover, the experimental runs taken for network training and validation were based on whey protein, a complex solute mixture, whose behaviour also depends on solution pH, among other parameters. In fact, absolute average deviations as calculated using 40% of the total data set for validation were found to be 6.9% at pH = 2.8 and 7.3% for pH = 5.5. Accordingly, it is concluded that this network can be considered as a model network for predicting fluxes for UF in a RDM module.

6. Symbols

J_V — Permeate volumetric flux, $m.s^{-1}$
t — Operation time, s

References

[1] W.S. Ho and K.K. Sirkar, Membrane Handbook, Van Nostrand Reinhold, New York, 1992.
[2] B.E. Reed, W.L.R. Viadero and J. Young, Treatment of oily wastes using high-shear rotary ultrafiltration, J. Env. Eng., 123 (1997) 1234–1242.
[3] S. Bhattacharjee, S. Datta and C. Bhattacharjee, Studies on the fractionation of β-lactoglobulin from casein whey using ultrafiltration and ion-exchange membrane chromatography, J. Membr. Sci., 275 (2006) 141–150.
[4] M.D. Petala and A.I. Zouboulis, Vibratory shear enhanced processing membrane filtration applied for the removal of natural organic matter from surface waters, J. Membr. Sci., 269 (2006) 1–14.
[5] J.G. Pharoah, N. Djilali and G.W. Vickers, Fluid mechanics and mass transfer in centrifugal membrane separation, J. Membr. Sci., 176 (2000) 277–289.
[6] B. Culkin, A. Plotkin and M. Monroe, Solve membrane fouling problems with high shear filtration, Chem. Eng. Prog., 94 (1998) 29–33.
[7] L. Ding, O. Al-Akoum, A. Abraham and M. Y. Jaffrin, Milk protein concentration by ultrafiltration with rotating disk modules, Desalination, 144 (2002) 307–311.
[8] W.F. Blatt, A. Dravid, A.S. Michael and L. Nelsen, Solute polarization and cake formation in membrane ultrafiltration: causes, consequences and control techniques, Membrane Science and Technology, Plenum Press, New York, 1970, pp. 47–97.
[9] J.G. Wijmans, S. Nakao and C.A. Smolders, Flux limitation in ultrafiltration: osmotic pressure model and gel layer model, J. Membr. Sci., 20 (1984) 115–124.
[10] F.E. Danes, B. Boriou and S. Poyen, Effects of diffusion and osmosis on flux decline during ultrafiltration with total rejection on an unstirred batch cell, J. Membr. Sci., 50 (1990) 177–187.
[11] J Henriques, P. Gil, A. Dourado and H. Duarte-Ramos, Application of a recurrent neural network in online modelling of real-time systems, Orthodox Academy of Crete, Greece, 1999, pp. 1–6.
[12] C.C. Jeng and I.C. Yang, Practical implementation of back-propagation networks in a low-cost pc cluster, Neural Inform. Process., 4 (2004) 33–37.
[13] S. Chen, C.F.N. Cowan and P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks, 2 (1991) 302–309.
[14] J.J. Hopfield and D.W. Tank, Computing with neural circuits, Science, 233 (1986) 625–633.
[15] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning internal representations by error propagation, MIT Press, Cambridge, MA, 1986.
[16] T. Samad, Back-propagation improvements based on heuristic arguments, Proc. International Joint Conference on Neural Networks, Washington, 1990, pp. 565–568.
[17] A.A. Miniani and R.D. Williams, Acceleration of back-propagation through learning rate and momentum adaptation, Proc. International Joint Conference on Neural Networks, San Diego, CA, 1990, pp. 676–679.
[18] R.A. Jacobs, Increased rates of convergence through learning rate adaptation, Neural Networks, 1 (1988) 295–308.
[19] A. Salvetti and B.M. Wilamowski, Introducing Stochastic Process within the Backpropagation Algorithm for Improved Convergence, Artificial Neural Networks in Engineering, St. Louis, Missouri, 1994, pp. 13–16.
[20] A.A. Al-Falou and D. Trummer, Identifiability of recurrent neural networks, Vienna University of Technology, Econometric Theory, 19 (2003) 812–828.
[21] J.L. Elman, Finding structure in time, Cog. Sci., 14 (1990) 179–211.
[22] G. Galan-Marin, E. Merida-Casermeiro and J. Munoz-Pere, Modelling the competitive Hopfield Networks for the maximum clique problem, Comp. Ops. Res., 30 (2003) 603–624.
[23] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sci. USA, 1982, pp. 2554–2558.
[24] J.J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, Proc. Nat. Acad. Sci. USA, 1984, pp. 3088–3092.
[25] D.W. Tank and J.J. Hopfield, Collective computation in neuron like circuits, Sci. Am., 257 (1987) 104–114.
[26] K.A. Smith, D. Abramson and D. Duke, Hopfield neural networks for timetabling: formulations, methods, and comparative results, Comp. Ind. Eng., 44 (2003) 283–305.
[26] S. Chen, S.A. Billings and W. Luo, Orthogonal least squares methods and their application to non-linear system identification, Int. J. Control, 50 (1989) 1873–1896.
[28] M. Cheryan, Ultrafiltration and Microfiltration Handbook, Technomic Publications, Lancaster, PA, 1998.
[29] P. Gao, L. Zhang, K. Cheng and H. Zhang, A new approach to performance analysis of a seawater desalination system by an artificial neural network, Desalination, 205 (2007) 147–155.