Desalination and Water Treatment
www.deswater.com

doi: 10.5004/dwt.2017.20768

75 (2017) 34–44
May

Nonlinear autoregressive with exogenous input (NARX) modelling for fouling prediction in industrial dead-end ultrafiltration process

Mohamed Bin Shams

Department of Chemical Engineering, University of Bahrain, P.O. Box 32038, Isa Town, Bahrain, Tel. +973 (1) 7876050; Fax: +973 (1) 7680935; email: mshams@uob.edu.bh

Received 23 December 2016; Accepted 27 March 2017

ABSTRACT

In this paper, a nonlinear autoregressive with exogenous input (NARX) model is developed to predict the fouling in industrial-scale dead-end ultrafiltration (UF) system. For better representation of the operating regions, experimental data for training and testing were generated using design of experiments, namely, the central composite design. The NARX model, which is basically a recurrent neural networks, was fed with four inputs: the forward pressure, backwash pressure, forward time and backwash time to predict the forward flow rate of a single 8 inch industrial UF membrane. The dynamic network learned the input–output mappings with great proficiency and showed decent accuracy in predicting forward flow rates for operating conditions different than that used during the training phase. Using different operating condition scenarios, the correlation coefficients (*r*) of the parity plots were ranged from 98.59% to 99.47%. In addition, using an informative and large data set allows smaller NARX network architecture to be built. The proposed NARX model provides an easy, quick-to-build and efficient time-dependent framework for fouling prediction that can be used for advanced process control algorithms and process optimization, in particular for large-scale industrial UF systems.

Keywords: Ultrafiltration; NARX; Recurrent neural networks; Fouling; Design of experiments

1. Introduction

Worldwide demand of clean water is rising especially in regions with water scarcity due to the rapid growth in population and economy as well as due to the ongoing deterioration of freshwater resources. Together with the reduction in cost and energy consumption, membrane-based technology becomes a preferred choice in most of the recent water production and waste treatment processes [1]. Ultrafiltration (UF) becomes a preferred option for chemical recovery, pre-desalination, wastewater reclamation, juice concentration, dairy making, medical usage and in treating oily wastewater [1,2]. Concentration polarization and fouling are major problems shared by all membrane separation processes and limits its performance [3]. However, for UF system, concentration polarization plays a dominant role during the filtration of low molecular weight solutes or macromolecules [4]. Flux losses due to concentration polarization can be recovered using backwashing. However, more serious type of fouling is the one which occurs as a result of the deposition of the foulant on the porous surface or inside the porous structure of the membrane leading to a decrease in forward flow rate. Researchers were able to distinguish between reversible and irreversible fouling according to their responses to certain cleaning methods [5].

There are different forms of fouling in UF systems, e.g., biological fouling, scaling, colloidal, organic and inorganic fouling. Fouling can be developed through different mechanisms, namely, adsorption, pore blocking and cake/gel formation [6]. Several parameters affect the rate and the extent of fouling in UF system, e.g., nature and concentration of solutes and solvents, membrane materials and its surface characteristics, the pore size and its distribution, the hydrodynamics of the membrane modules, the cross-flow velocity (CFV), transmembrane pressure (TMP), pH, ionic environment and the cleaning protocols implemented [1,3,7]. It is important to predict the effect of the aforementioned parameters and their interactions on the productivity of the UF system under different fouling conditions. Therefore, several Darcy's-based relationships were found in the literature which has the form given by Eq. (1):

$$J = \frac{\Delta p - \Delta\pi}{\mu(R_m + R_{\text{ads}} + R_{\text{rev}} + R_{\text{irrev}})} \quad (1)$$

where J is the volumetric flux, Δp is the TMP, $\Delta\pi$ difference in osmotic pressures, R_{ads} is the resistance due to surface or pore adsorption, R_m is the empirically measured membrane resistance, R_{rev} is the reversible resistance (not present after physical/chemical cleaning), R_{irrev} is the irreversible resistance and μ is the dynamic viscosity of the permeate. If the UF system is operated at constant TMP, then Eq. (1) shows that increasing ($R_{\text{ads}} + R_{\text{rev}} + R_{\text{irrev}}$) with time will lead to flux decline [6]. Fouling, unlike concentration polarization, is considered to be a function of time. Therefore, a dynamic modelling framework is more appropriate to predict the performance of UF system under fouling condition. Ho and Zydney [8] proposed a model to predict the forward flow rate through a fouled membrane at any given filtration time. The volumetric flow rate through the fouled membrane Q was given as:

$$Q = Q_0\left[\exp\left(-\frac{\alpha\Delta PC_b}{\mu R_m}t\right) + \int_0^t \frac{\alpha\Delta PC_b}{\mu(R_m + R_p)} \times \exp\left(-\frac{\alpha\Delta PC_b}{\mu R_m}t_p\right)dt_p\right] \quad (2)$$

where Q_0 is the initial volumetric flow rate through the clean membrane, α is the pore blockage parameter, ΔP is the TMP, C_b is the bulk colloidal concentration, μ is the solution viscosity, R_m is the clean membrane resistance, R_p is the resistance of protein/colloidal deposit over a particular region of the membrane, t is current time and t_p is time at which a particular region was first covered or blocked by an aggregate. In spite of its theoretical basis, Eq. (2) requires the determination of several parameters which are usually difficult to be precisely and rapidly estimated, e.g., t_p. Furthermore, only forward filtration stage is considered in Eq. (2) while backwashing was not addressed. The complexity and nonlinear nature of Eq. (2) and its counterparts available in literatures motivates the use of data-driven or empirical-based models for performance predictions of membrane processes.

Several investigations have considered the use of static neural network (NN) models for studying and controlling membrane separation processes. Using data collected from five desalination plants in the Gaza Strip, Aish et al. [9] developed multilayer perceptron and radial basis function neural networks for one week ahead forecasting. Total dissolved solids concentrations and permeate flow rate in the Gaza Strip desalination plants were predicted as outputs. Water temperature, pH, conductivity and pressure were used as inputs for both networks [9]. Recently, Cao et al. [10] developed a static feedforward NNs to study the performance of a vacuum membrane distillation. The statistically validated NN model showed that the effect of the vacuum pressure is the greatest on the permeate flux followed by the feed inlet temperature. Hamachi et al. [11] developed a recurrent NN to predict the permeate flux and deposit thickness of a cross-flow microfiltration of a bentonite suspension. The inputs to their model were the TMP, CFV and concentration of the suspension [11]. Although the wide range of static NNs applications in membrane separation processes, few studies have considered the application of recurrent NNs for performance prediction and control of large-scale UF systems. While feedforward NNs use supervised learning algorithms to update its weights, self-organizing map (SOM) is considered as an unsupervised learning network [12]. Ki et al. [13] used SOM to analyze a large data sets simulated using large number of hollow fibre direct contact membrane distillation. The SOM was found useful in grouping the simulation input parameters with respect to their thermal, geometrical and fluid dynamic influences on the mass and heat transfers [13]. Tayyebi and Alishiri [14] proposed a nonlinear inverse model control strategy based on NNs for a simulated Multistage flash (MSF) desalination plant. In particular, three NNs were used to control the top brine temperature, level of last stage and brine salinity. Set-point tracking results indicated that NN-based controllers are robust tools for MSF plants. Barello et al. [15] developed a static NN-based correlation for estimating water permeability constant (K_w) of hollow fibre and spiral wound membranes]. The proficiency of their models was verified against experimental data from literature and was found useful in predicting K_w for any salinity and operating pressure under fouling conditions. Badrnezhad and Mirza [16] proposed a hybrid model that combines NN and genetic algorithms (GAs) for cross-flow UF of oily wastewaters [16]. While the role of NN was to predict the flux decline under various operating parameters, GA was used to optimize the operating parameters for a desired permeate flux. The validation shows good agreements while treating oily wastewater from Tehran refinery.

As stated earlier, the amount of published works that are available in the literature on the application of recurrent NNs to membrane processes in general and large-scale industrial UF system in particular remains to be limited. In addition, the use of design of experiment (DoE) for NN modelling and in particular for fouling prediction is scarce. Accordingly, a rapid and precise data-driven approach that dynamically predicts the forward/filtrate flow rate through fouled membrane could be an attractive alternative. Specially, if the operating conditions such as backwashing flow, backwashing pressure, durations of forward filtration and backwashing are considered. Therefore, in this paper, a nonlinear autoregressive with exogenous input (NARX) model is proposed for predicting the long-term performance of an industrial-scale dead-end UF system operated under constant TMP. To reduce the risk of the well-known extrapolation problem encountered when NN-based models are used, the proposed NARX-based NN model was trained using a data set that spans a range of operating conditions determined systematically using one of the DoE techniques, namely, the central composite design (CCD).

2. Recurrent neural network modelling: architecture and learning algorithm

NNs can be classified into two main categories, namely feedforward networks (or static networks) and feedback networks (or recurrent networks) [12]. For static feedforward NNs, there are no feedback paths. A set of input–output data collected from the process to be modelled is presented to the NN. The differences between the actual output and the one predicted by the NN model are minimized while adjusting the network weights (synaptic) using an optimization algorithm, e.g., steepest descent [17]. On the other hand, recurrent NN is characterized by the presence of a feedback. That is, the model architecture allows the outputs from one or more nodes, e.g., the output layer, to be fed back to one or more nodes in previous layers, e.g., hidden or input layers. The feedback structure of the recurrent NN provides storage capabilities to the output nodes through dynamic states, therefore, providing the network with some sort of memory. The latter makes recurrent NN very useful for modelling and identifying time-dependent phenomena such as fouling. Similar to feedforward NN, the basic elements of the computation in recurrent NN are the neurons. These are simply a processors that take the weighted sum of the inputs from other nodes and apply nonlinear mapping (activation function) before passing the output to the next neuron. The output o_s of a single neuron s having R inputs is given as:

$$o_s = f\left(\sum_{r=1}^{R} w_{sr} x_r - \theta_s\right) \tag{3}$$

where f is the node's activation function, $x_1, x_2, \ldots, x_R$ are the node's inputs, w_{sr} is the connection weight between the r-th input and the s-th neuron and θ_s is the threshold for the s-th node. Depending on the problem and the location of the node within the layer, several differentiable activation functions (f) can be used, e.g., sigmoid, hyperbolic tangent sigmoid and linear functions. Fig. 1 illustrates the profile of three activation functions commonly used for NNs applications. For an arbitrary unit in a recurrent network, the activation at a given time t is given as:

$$o_i(t) = f\left(\sum_j w_{ij}\, o_j(t-1) + x_i(t-1)\right) \tag{4}$$

where $o_i(t)$ is the current output signal of node i, $o_j(t-1)$ is the output signal of the node j at previous time sample and w_{ij} is the connection weight between node i and node j. Multiple layers of neurons with nonlinear transfer functions provide the network with the required capabilities to learn the nonlinear relationships between input and output vectors. The NARX network is a special variant of recurrent dynamic networks. More specifically, NARX model is a generalization of the well-known autoregressive with exogenous inputs model used for time series modelling [18]. NARX model is gives as:

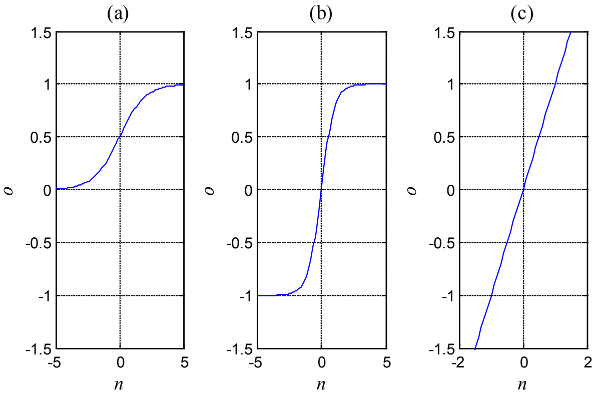

Fig. 1. Typical profiles of three common activation functions: (a) log-sigmoid transfer function, (b) tan-sigmoid transfer function and (c) linear transfer function.

$$y(k+1) = f[y(k),\cdots,y(k-m);u(k),\cdots,u(k-n)] + \varepsilon_i \tag{5}$$

where $y(k+1)$ is the output of network at time $(k+1)$, and $y(k)$, $y(k-1)$, …, $y(k-m)$ are the delayed outputs and can be thought of as network inputs along with the input signal $u(k)$ and its delayed variants $u(k-1)$, …, $u(k-n)$. ε_i is independently and identically distributed error component. A diagram of the NARX network is shown in Fig. 2, with one-layer feedforward network. For multiple inputs and outputs, the input $u(k)$ and output $y(k)$ can be multidimensional vectors. The training algorithm for the NARX network is an extension of the standard backpropagation algorithm of the static feedforward NN. The extension is based on the fact that the target outputs of the NARX model are available during the training phase of the network, and therefore, one could create what is called a series-parallel architecture by eliminating the feedback path [12]. As can be seen in Fig. 3, the series-parallel architecture uses the target outputs instead of feeding back the estimated output from the network during the training phase. A main advantage of such an arrangement is that the resulting network has feedforward architecture in which the classical static backpropagation learning algorithm (BPL) can be used to train the original parallel configuration of the NARX network (Fig. 2). The BPL is based on the gradient descent technique for solving optimization problems, that is, the minimization of the network cumulative error E_c with respect to the networks weights. E_c represents the sum of n squared errors $E(k)$, where $E(k)$ is the difference between the target output from the i-th output neuron $t_i(k)$ and the corresponding network prediction $o_i(k)$, q is total output neurons and n is the number of training patterns presented to the network for learning purposes, that is:

$$E_c = \sum_{k=1}^{n} E(k) = \frac{1}{2}\sum_{k=1}^{n}\left(\sum_{i=1}^{q}\left(t_i(k) - o_i(k)\right)^2\right) \tag{6}$$

BPL is designed to iteratively update the connection weights in the direction of the gradient decent of E_c. For offline learning at which all the training patterns are presented to the network at once, the BPL can be summarized as follows:

$$\min_{\mathbf{w}} E_c = \min_{\mathbf{w}} \frac{1}{2}\sum_{k=1}^{n}\left(\sum_{i=1}^{q}\left(t_i(k) - o_i(k)\right)^2\right) \tag{7}$$

where the vector $\mathbf{w}$ denotes the network weight vector, that is, the interconnection weights among all the neurons of the network. The backpropagation has its name due to the fact that the error signal is used to update the weight (w_{ij}) which connects a neuron j in a layer and the neuron i at the preceding layer get its value from the knowledge of the error of the following layer [12]. That is, at every training cycle, the values of the weights are propagated backward from the output layer all the way to the input layer.

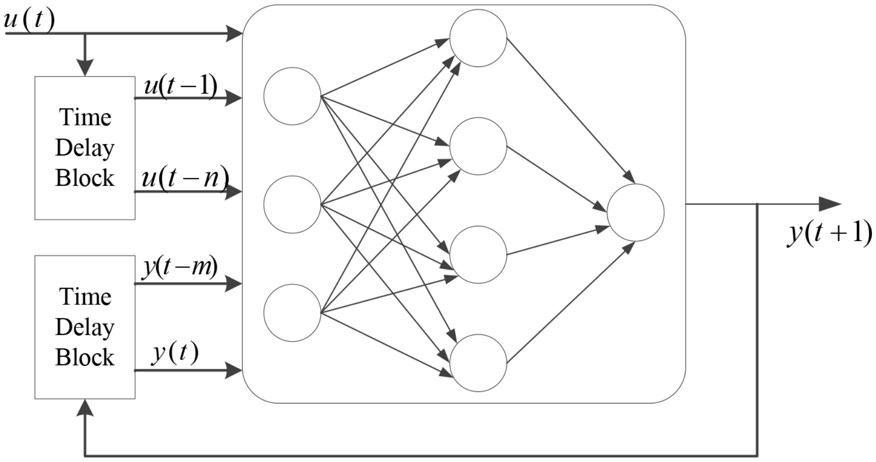

Fig. 2. NARX-based feedforward neural network (parallel architecture). The inclusion of time delays and feedbacks gives the network dynamic capabilities.

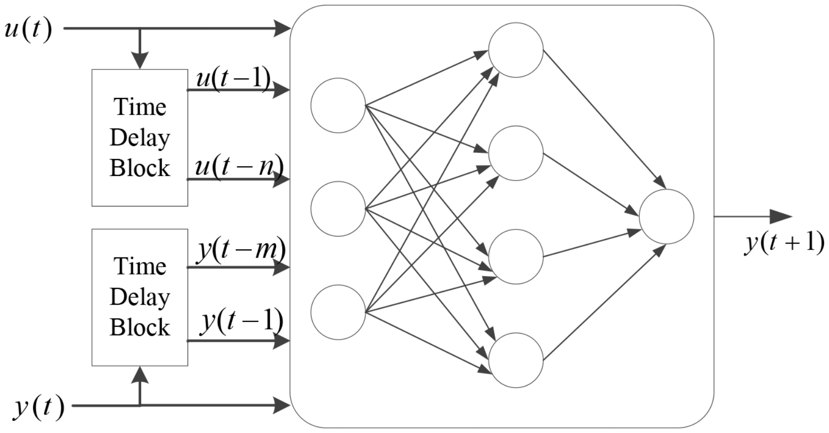

Fig. 3. NARX-based feedforward neural network (series-parallel architecture).

3. Experiment and method

3.1. Ultrafiltration setup and solution preparation

The industrial-scale UF system used in this study has been assembled at the University of Bahrain (Fig. 4). The system consists of 8 inch X-FLOW XIGA 40 hydrophilic membrane composed of a blend of polyvinylpyrrolidone and polyethersulfone. The 1.5275 m membrane has asymmetric microporous structure with a nominal pore size of 20 nm and a total membrane area of 40 m^2. The element outer diameter, permeate collector inner diameter and the hydraulic membrane diameter are 200, 42.6 and 0.8 mm, respectively. Two centrifugal variable speed pumps (Grundfos Lenntech) are used to vary between the forward (filtration) and backwash mode of operations. The speeds of the pumps were manually adjusted to achieve the required pressures. Several pneumatic valves (Georg Fischer) were used to facilitate different operation modes of the process. Turbidity and pH meters were provided to monitor the quality of the feed and product streams. In addition, the pressures of the feed, product and backwash streams were monitored through pressure transmitters (Endress+Hauser) while the corresponding flow rates were monitored using magnetic flow transmitters (Endress+Hauser). Bourdon gauges and rotameters were provided for on-site monitoring. Measurements from all transmitters and the operating conditions of the pumps and valves were hard wired to the Allen-Bradley programmable logic controller (PLC) (MicroLogix 1200). The PLC in turn was connected to an supervisory control and data acquisition system (RSLogix 500) which provides full automation of the pilot UF plant. The sampling frequency of the SCADA system was 1 s. A simulated wastewater using suspended Aerosil 200 was synthesized. The latter is a trade name of Degussa AG which consists of hydrophilic fumed silica (99.8% purity silicon dioxide) with average particle size of 12 nm and specific surface area of 200 m^2/g. The amount of silicon dioxide (Aerosil 200) was mixed with a tap water and agitated for about 20–30 min. After ensuring that the suspended particles were consistently agitated, the system was started through the SCADA system. The SCADA program allows forward and backwash intervals to be entered as well as starting the pumps and operating the required valves. The feed was pumped through the membrane. A series of pneumatic valves directed the feed stream within the system before the filtrate was collected in the product's tank. Once the allotted forward time was elapsed, backwash begins. During the backwashing, part of the filtrate in the product tank was used to clear the filter cake formed during the forward phase. The retentate was fed back to the feed tank. Backwashing was required to maintain the efficiency of the membrane and reduce the resulting flux decline. Backwashing is considered as an energy-efficient approach to prevent flux decline as opposed to increasing feed's pressure. Although there are many operating parameters (input factors) that can influence the performance of the UF system, forward pressure, backwash pressure, forward time, the backwash time and the concentration of foulant in the feed stream were considered in the current study. The extent of fouling was realized through the flux decline observed by the filtrate of the UF system. In order to accelerate the onset of fouling, a very high concentration of the foulant was used (4,000 ppm) in all experiments.

3.2. Data collection and pre-processing

Different operating condition levels have different effects on the productivity of UF system as well as rate in which fouling develops. For efficient representation of possible range of operating conditions, DoE was employed in collecting the data required to train and validate the NARX-based NN. DoE becomes a necessary approach for data collection especially when significant interaction among factors/operating conditions exists. While characterizing the UF system under study, it was found that there is a significant interaction effect between the level of backwashing-pressure and backwashing-time on the filtrate production and hence the extent of fouling [7]. It would be impossible to detect such interaction using the widely used one-factor-at-time approach [19]. Therefore, collecting data within the framework of DoE to train the NARX-based NN provides an informative basis that not only considers the individual factor effects but also considers the interactions among these factors. Different designs techniques have been developed for different purposes. However, in the current study, and in order to span a wide range of possible operating conditions, CCD was used. CCD design contains an imbedded 2^4 factorial design, with centre points that are augmented with a group of extremes points (axial points) for possible curvature estimation [19]. Table 1 shows the levels of the four input factors used to generate the data for training and validating the NARX model.

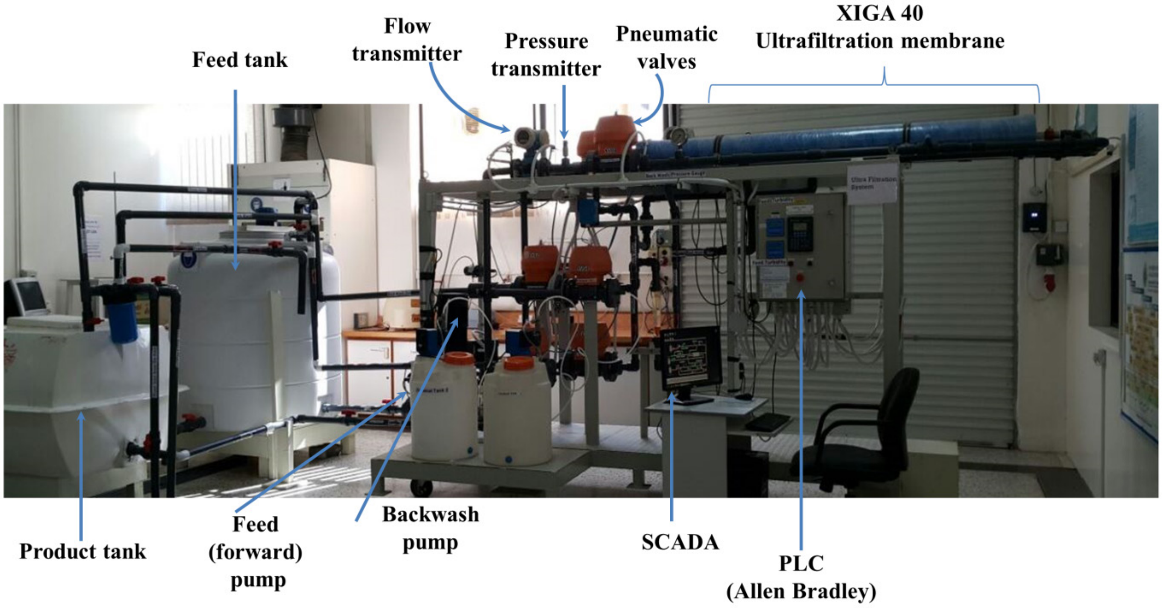

Fig. 4. The 8 inch industrial-scale UF system. The system is fully automated (Allen Bradely PLC) and operated through an SCADA system (RSLogix 500).

Table 1
The levels of the four operating conditions used to train and validate the NARX-based neural network

Uncoded (coded)	A: forward pressure (psi)	B: backward pressure (psi)	C: forward time (s)	D: backwash time (s)
High (+1)	38	26	240	90
Low (–1)	23	12	120	30
Centre points (0)	31	19	180	60
Axial (+2)	46	33	300	120
Axial (–2)	16	5	60	0

It is worth to emphasize here that fouling is a time-dependent phenomenon. The trained NARX-based NN is aimed to learn the nonlinear correlation patterns between the four operating conditions, namely, forward pressure, backwash pressure, forward time, backwash time and their effects on the forward flow rate as they evolve in time. Because different level combinations have different effect on the forward flow rate and the extent at which fouling evolved, it was found that for some experiments, few minutes were only required before the flow was completely ceased. Therefore, different number of data points was generated for different experiment as shown in the last column in Table 2. Fig. 5 depicts the concatenated time series of the six measured time-series collected from 31 experiments. Responses generated using level combinations in Table 1 represent a data bank from which one can select data to train the NARX-based NN. Accordingly, and in order to train the proposed NARX-based NN, input/output data sets from the first nine experiments in Table 1 were concatenated. MATLAB® Neural Network Toolbox was used to train and validate the proposed NARX-based NN. In identifying NN models, normalization of the input and output data sets is a critical pre-processing step. The latter is particularly important when the operating variables are not of the same order of magnitudes. If no normalization was carried out, the learning algorithm would assign large weight for input with large numerical values and small weight for inputs with small numerical values. Accordingly, small input factor will be less represented in the NARX-based NN even if it exhibits significant variations. Another aspect of normalization concerns the activation functions. For example, consider the log-sigmoid and tan-sigmoid transfer functions in Fig. 1. It can be seen that it is difficult for the activation function to distinguish between two different large values of n, since both of them will produce output values equal to 1.0 (e.g., sign (40) vs. sign (400)). To evade such difficulties, it was recommended to

Table 2
The design table for 31 experiments used to train the NARX-based neural network (16 full factorial experiments + 7 centre points + 8 axial points)

Exp. No.	Forward pressure (psi)	Backwash pressure (psi)	Forward time (s)	Backwash time (s)	No. of experimental data points
1	0	2	0	0	1,024
2	–1	1	–1	–1	1,499
3	1	–1	1	–1	799
4	2	0	0	0	1,049
5	0	0	0	–2	349
6	–1	–1	1	1	899
7	0	0	–2	0	1,149
8	1	1	–1	–1	1,137
9	–1	1	1	1	949
10	–1	–1	–1	1	899
11	1	–1	1	1	889
12	1	–1	–1	1	849
13	1	1	1	–1	629
14	1	–1	–1	–1	749
15	0	0	0	0	749
16	–1	–1	–1	–1	749
17	1	1	1	1	899
18	0	0	0	0	749
19	–1	1	–1	1	649
20	0	0	0	0	749
21	0	0	0	0	749
22	–1	1	1	–1	549
23	–1	–1	1	–1	599
24	1	1	–1	1	719
25	0	0	2	0	719
26	0	0	0	0	749
27	–2	0	0	0	699
28	0	0	0	0	749
29	0	0	0	0	749
30	0	0	0	2	669
31	0	–2	0	0	749

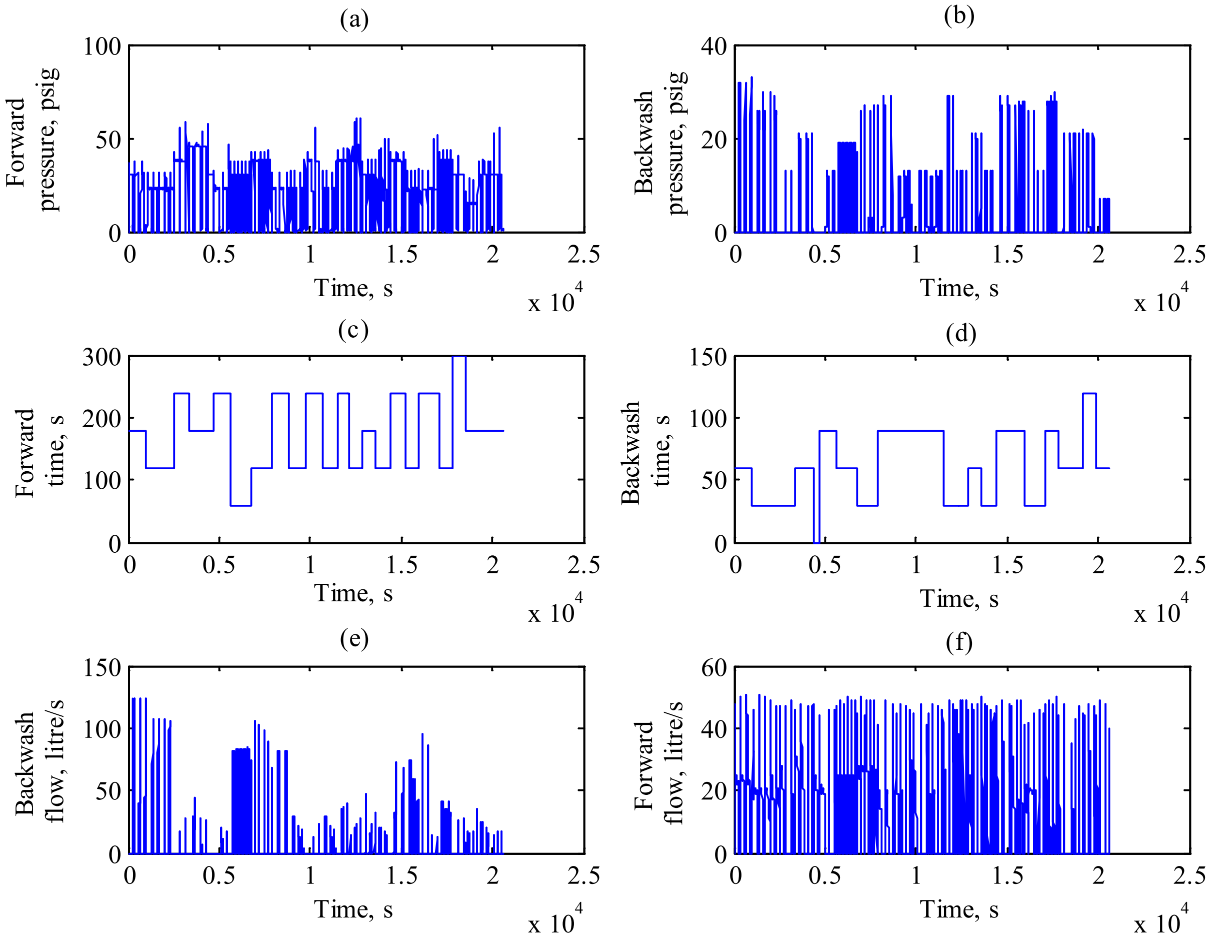

Fig. 5. Time series of the input–output data sets. (a) Forward pressure, (b) backwash pressure, (c) forward time, (d) backwash time, (e) backwash flow and (f) forward flow.

normalize the input/output data sets. Before presenting the data to the NARX-based NN, Neural Network Toolbox automatically normalizes the data set between limits of –1 and +1, having the average value set at 0. The data were normalized according to the following formula:

$$x_{i,\mathrm{norm}} = \frac{2(x_i - x_{i,\min})}{x_{i,\max} - x_{i,\min}} - 1 \tag{8}$$

where x_i is an input or target variable, $x_{i,\min}$ is the minimum value of the variable and $x_{i,\max}$ is the maximum value of the variable. The normalization technique in Eq. (8) utilizes the complete range of the transfer function and hence every variable in the data set has a similar distribution range.

4. Results and discussion

To train and validate the NARX model, a data set consists of 31 experiments was made available as shown in Table 2. The DoE approach was adopted in the current study for data collection. DoE provides a systematic approach to span the entire feasible operating range of the UF membrane system and in particular useful when interactions between input factors are significant. In addition, the DoE approach aims to avoid the well-known extrapolation problems associated with the NN-based models [17]. A general practice in the DoE approaches is to choose the high, low and extreme levels of a factor far enough such that a change in the response can be observed, however, not too far apart such that the expected physical operating ranges of the system are exceeded. For example, the forward and back wash pressures are constrained by the maximum pressure that the membrane can withstand. Accordingly, it is expected that the well-known extrapolation problem associated with NN modelling will be less severe since the most feasible operating space was considered during the design. As a result, the predictions of the NARX network are expected to be within the space considered during the DoE.

To illustrate the range of forward flow rate responses for different level combinations of the four input parameters of the UF system (forward pressure, backwash pressure, forward time and backwash time), Fig. 6 depicts four selected experiments from Table 2. Fig. 6(a) shows the forward and backwash flows for Exp. No. 2. For the latter, forward pressure, backwash pressure, forward time and backwash time were 23 psi, 26 psi, 120 s and 30 s, respectively. The forward and backwash flows of Exp. No. 5 are depicted in Fig. 6(b) where forward pressure, backwash pressure, forward time and

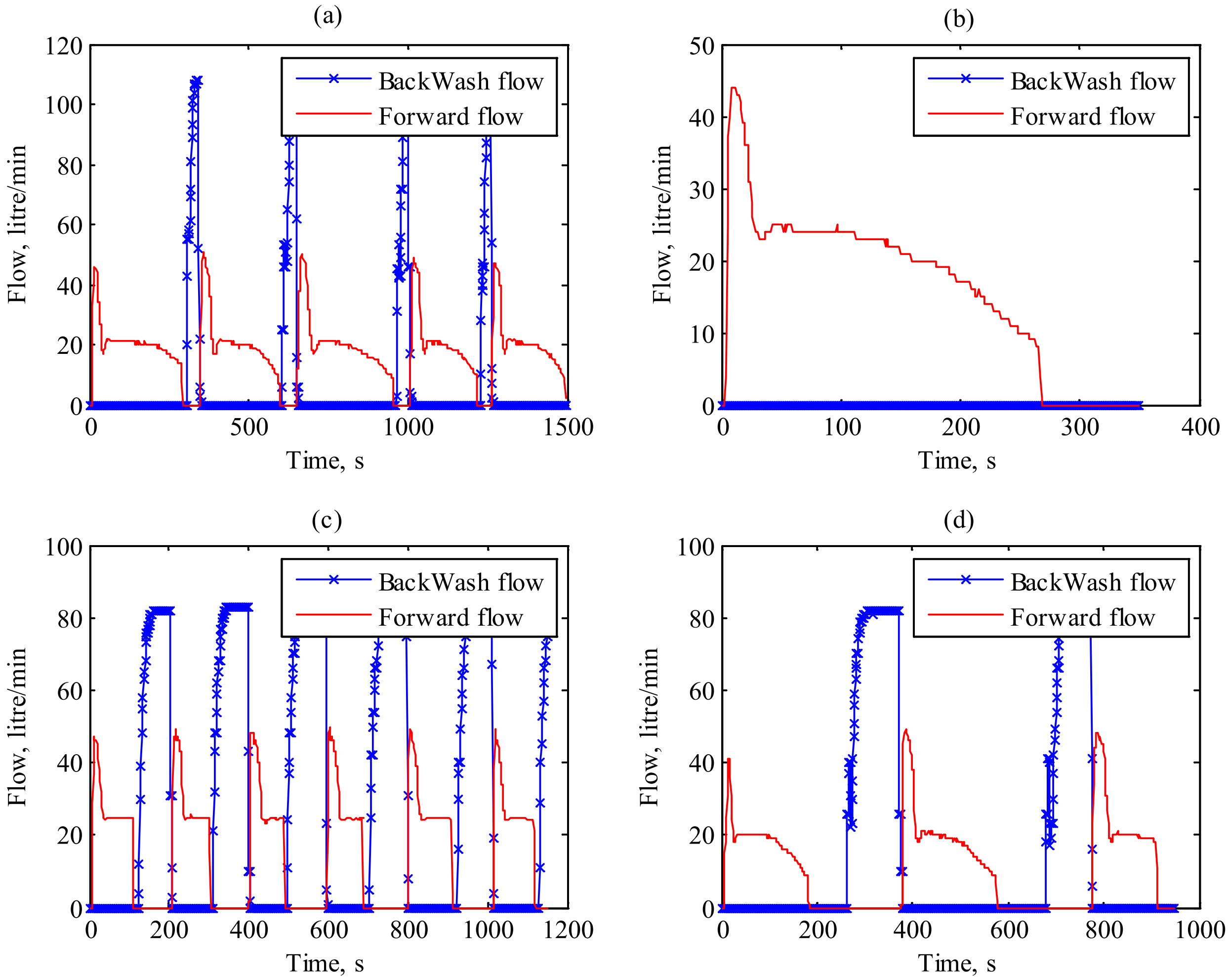

Fig. 6. Forward flow and backwash flows for four selected experiments are given in Table 2. (a) Exp. No. 2, (b) Exp. No. 5, (c) Exp. No. 7 and (d) Exp. No. 9.

backwash time were 31 psi, 19 psi, 180 s and 0 s, respectively. Note that in Exp. No. 5 no backwash was allowed. Therefore, and as expected for the high concentration (4,000 ppm) used in all experiments, complete blockage of the membrane was observed after 4 min. A vital observation is the different rate of flux declines observed under the different operating conditions. The different rate of declines actually makes the recurrent NN an appropriate choice for predicting the performance of the UF system. Fig. 6(c) demonstrates the case of frequent backwashing and Fig. 6(d) shows the case of delayed backwashing. In fact, determining the optimal time for backwashing is a trade-off between the required pumping energy and the preserving membrane's efficiency. Therefore, experiments in Table 2 gives a range of responses suitable to train the NARX model. Experiments 1–9 (Table 2) were selected to train the NARX model. The four inputs time-series and the output (forward flow rate) were presented to the network. Neural Network Toolbox® was used to design the network architecture. The network consists of one hidden and one output layers. Within the hidden layer, 10 neurons were used. The number of delays, that is, n and m in Eq. (5) were set to 2. The number of neurons and delays were chosen by trial and error. Sigmoid and linear transfer functions were used in the hidden and the output layers, respectively. The input time series were placed in a matrix **X** and the response in a matrix **Y**. The **X** matrix consists of 8,854 time-steps of five variables, that is, x_1 = forward pressure, x_2 = backwash pressure, x_3 = forward time, x_4 = backwash time and x_5 = back wash flow. The time-steps are 1 s apart. The **Y** matrix consists of 8,854 time-steps of one variable, that is, the forward flow rate. The 8,854 rows in the **X** and its corresponding responses in the **Y** matrix were divided up randomly.

Following the random division, 70% of the data set (6,198 points) was used to train the network. To assess the network's generalization capability during the training phase, that is, the ability of the network to interpolate new patterns whose outputs are known to the user but not to the network, 15% of the data set was used (1,328 points). The latter is essential for terminating the training phase and prevents overfitting the data. Finally, to test the resulting trained network, the remaining 15% of the data set was used (1,328 points). The mean squared error (MSE), that is, the average squared difference between the outputs and the targets was used as performance index for the training, validation and testing phases. The Levenberg–Marquardt backpropagation algorithm was used to train the network.

In spite of the closed loop architecture of the NARX model, backpropagation algorithm can still be used by opening the feedback loop (Figs. 2 and 3). The latter was made possible due to the fact that output at different times were available during the training phase. Therefore, the open-loop architecture can be used, where the application of the backpropagation algorithm becomes straightforward. The time required to train the network was 6 s using Intel® Core™ i5-2410M CPU with 4.0 GB RAM. The short training duration can be explained by looking at Fig. 7 where 18 epochs were only required to attain a small (MSE) value of 2.1163. The 18 iterations were required before the validation and testing errors stopped decreasing further. The actual vs. the predicted forward flow rate during the training, validation and testing phases are depicted in Fig. 8. For the given input–output data set, *r* = 0.9941, *r* = 0.99459 and *r* = 0.9943 for the training, validation and testing phases. As can be seen, most of the circles are close to the 45° line indicating that the NARX model was successful in learning the input–output patterns presented to the network. Fig. 9 shows the time series plots of the actual observations (test data set) vs. the predictions from the trained NARX model. A good agreement between the NARX network and the unseen data set was obtained.

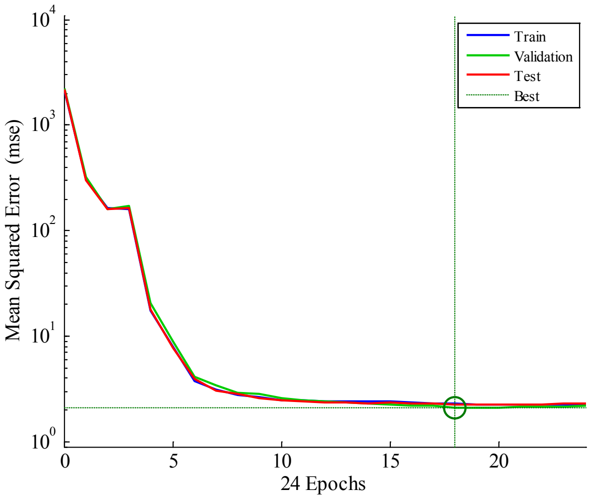

Fig. 7. Mean squared error (MSE) vs. the number of epochs. The best validation performance is 2.1163 at epoch 18.

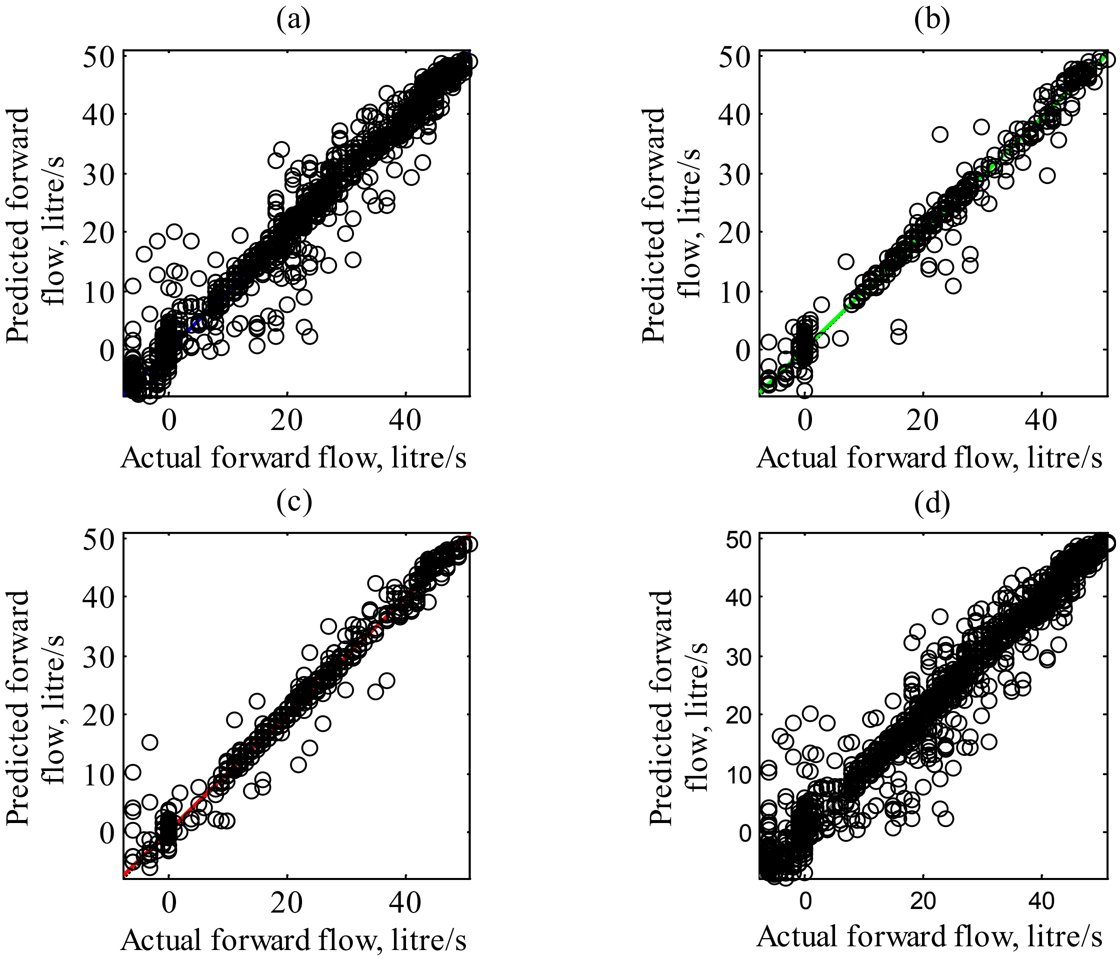

Fig. 8. Parity plots (predicted vs. actual forward flow rate). (a) Training: *r* = 0.9941, (b) validation: *r* = 0.99459, (c) test: *r* = 0.9943 and (d) overall: *r* = 0.9942.

To qualify the network, Fig. 10 depicts the autocorrelation function of the prediction errors, that is, the error term (ε_i) in Eq. (5). As can be noticed, the predictions errors were uncorrelated indicating the adequacy of the model. In addition, input-error cross-correlation function shows no unusual patterns. To further verify the generalization capability of the identified NARX model, the inputs from experiment 17 were scaled and presented to the trained NARX model. The data set for experiment 17 was completely new to the NARX model and consists of input matrix **X** of size 899 × 5 and response **Y** matrix of size 899 × 1. Fig. 11 demonstrates the generalization capability of the identified NARX model for the inputs from experiments 17. The NARX model shows decent performance in predicting the forward flow rate, hence the fouling responsible of the flux decline. The values of the correlation coefficient (*r*) and the MSE were estimated as 0.9947 and 2.7905, respectively. Table 3 gives the correlation coefficient (*r*) and the MSE of all the experiments in Table 2. As can be seen from Table 3, simulations of the trained NARX network yielded forward flow rates which are very close to the observed values. The latter can be perceived from the high correlation coefficient values (~0.99 for most of the runs).

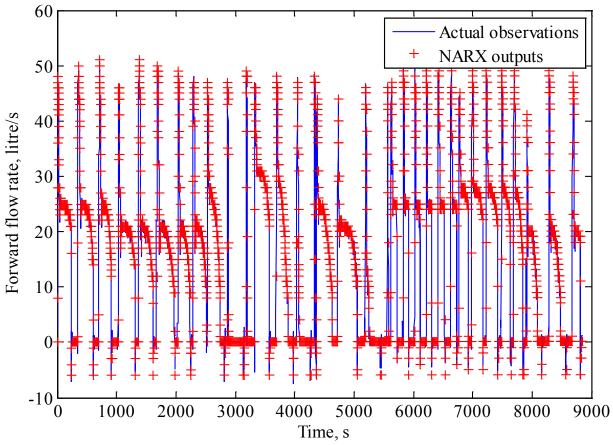

Fig. 9. The time series of the forward flow rate, observed vs. NARX predictions for the testing data set.

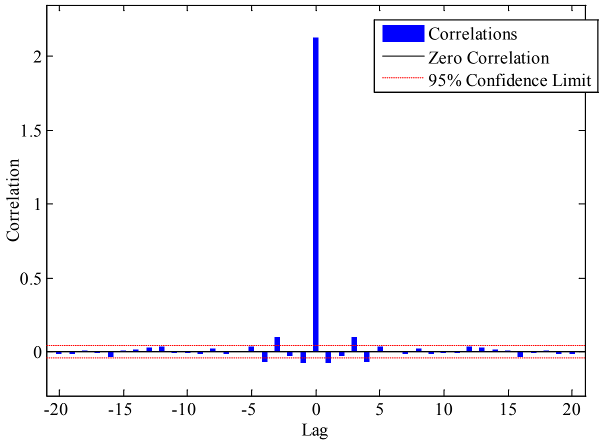

Fig. 10. Autocorrelation function of the prediction errors forms the NARX model.

The first 9 runs in Table 2 provide a rich and informative data set to train the NARX model. As can be seen from Table 2 for the first 9 runs, columns 2–5 spans different operating levels of the four input factors. In fact, this large and informative data set allowed the identification of a smaller NARX architecture (one neuron and one time delay). The main advantage of the latter is the short training time. The correlation coefficients of the NARX model using one neuron and one time delay element were 0.9917, 0.9921 and 0.9937 for the training, validation and testing, respectively. When the trained model was used to predict the forward flow rate using the input operating conditions for experiment 17, it yielded a correlation coefficient and MSE of 0.99402 and 2.7485, respectively. On the other hand, when as smaller data set consists of experiments 4–6 were used to train the NARX network (2,297 points), the correlation coefficients of the NARX model using one neuron and one time delay element was 0.9922, 0.9949 and 0.9912 for the training, validation and testing, respectively. The correlation coefficient for the parity plot for testing experiment 17 was 0.9791. Although the slight decrease in the correlation coefficient value, one can notice the ability of DoE trained NARX network in predicting the forward flow rate with good accuracy.

5. Conclusions

Predicting the impending fouling of UF membrane becomes an essential requirement in water treatment plants. Therefore, an NARX network was built to predict the forward flow rate and hence the rate of fouling of an industrial-scale UF system. The UF membrane system was systematically subjected to different operating conditions collected using DoE. The network inputs were the forward pressure, backwash pressure, forward time and backwash time. The NARX model showed high proficiency in predicting the forward flow of the UF system when data different than those used during the training phase is presented to the NARX model. The proposed NARX model could be of advantageous for advanced UF plant control algorithm and process optimization.

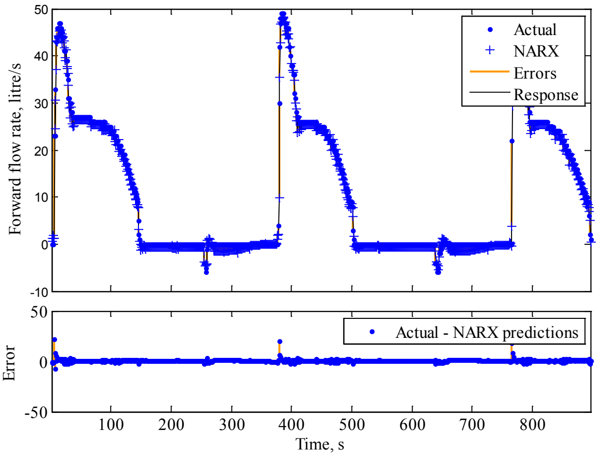

Fig. 11. The NARX predictions of the forward flow rate with inputs from experiment 17 (not used to train the network). The subplot below shows the errors (actual–NARX predictions).

Table 3
The generalization capability of NARX in predicting different experiment conditions in Table 2

Exp. No.	Correlation coefficient (*r*)	Mean squared error (MSE)	Exp. No.	Correlation coefficient (*r*)	Mean squared error (MSE)
10	0.9918	2.9846	19	0.9932	2.1634
11	0.9901	3.1561	22	0.9859	3.0933
12	0.9918	3.0118	23	0.9882	3.5310
13	0.9938	2.3978	24	0.9913	3.9111
14	0.9911	3.8940	25	0.9903	2.2508
15/18/20/21/26/28/29	0.9932	2.6549	27	0.9908	2.3709
16	0.9877	3.8242	30	0.9921	2.6867
17	0.9947	2.7905	31	0.9860	4.0347

Note: The mean squared error (MSE) and correlation coefficient (*r*) are for testing.

Acknowledgment

The author would like to acknowledge the Deanship of Scientific Research at University of Bahrain for the financial support.

References

[1] X. Shi, G. Tal, N.P. Hankins, V. Gitis, Fouling and cleaning of ultrafiltration membranes: a review, J. Water Process Eng., 1 (2014) 121–138.
[2] M. Padaki, R.S. Murali, M.S. Abdullah, N. Misdan, A. Moslehyani, M.A. Kassim, N. Hilal, A.F. Ismail, Membrane technology enhancement in oil–water separation. A review, Desalination, 357 (2015) 197–207.
[3] M.O. Nigam, B. Bansal, X.D. Chen, Fouling and cleaning of whey protein concentrate fouled ultrafiltration membranes, Desalination, 218 (2008) 313–322.
[4] J.D. Seader, E.J. Henley, Separation Process Principles, John Wiley & Sons, Inc., New Jersey, 2006.
[5] S.C. Tu, V. Ravindran, M. Pirbazari, A pore diffusion transport model for forecasting the performance of membrane processes, J. Membr. Sci., 265 (2005) 29–50.
[6] R. Field, Fundamentals of Fouling, Membranes for Water Treatment, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, 2010.
[7] M.B. Shams, F. Mumtaz, A. Murali, W. Akbar, N. Al-Bastaki, Application of statistical design to an industrial-scale dead-end ultrafiltration process, Chem. Eng. Technol., 38 (2015) 1315–1326.
[8] C.C. Ho, A.L. Zydney, A combined pore blockage and cake filtration model for protein fouling during microfiltration, J. Colloid Interface Sci., 232 (2000) 389–399.
[9] A.M. Aish, H.A. Zaqoot, S.M. Abdeljawad, Artificial neural network approach for predicting reverse osmosis desalination plants performance in the Gaza Strip, Desalination, 367 (2015) 240–247.
[10] W. Cao, Q. Liu, Y. Wang, I.M. Mujtabab, Modeling and simulation of VMD desalination process by ANN, Comput. Chem. Eng., 84 (2016) 96–103.
[11] M. Hamachi, M. Cabassud, A. Davin, M. Mietton Peuchot, Dynamic modelling of crossflow microfiltration of bentonite suspension using recurrent neural networks, Chem. Eng. Process., 38 (1999) 203–210.
[12] F.O. Karray, C.W. DeSilva, Soft Computing and Intelligent Systems Design: Theory, Tools and Applications, Addison-Wesley, London, 2004.
[13] S.J. Ki, H.J. Kim, A.S. Kim, Big data analysis of hollow fiber direct contact membrane distillation (HFDCMD) for simulation-based empirical analysis, Desalination, 355 (2015) 56–67.
[14] S. Tayyebi, M. Alishiri, The control of MSF desalination plants based on inverse model control by neural network, Desalination, 333 (2014) 92–100.
[15] M. Barello, D. Manca, R. Patel, I.M. Mujtaba, Neural network based correlation for estimating water permeability constant in RO desalination process under fouling, Desalination, 345 (2014) 101–111.
[16] R. Badrnezhad, B. Mirza, Modeling and optimization of cross-flow ultrafiltration using hybrid neural network-genetic algorithm approach, Ind. Eng. Chem. Res., 20 (2014) 528–543.
[17] D.R. Baughman, Y.A. Liu, Neural Networks in Bioprocessing and Chemical Engineering, Academic Press, Inc., San Diego, 1995.
[18] G. Box, G.M. Jenkins, G.C. Reinsel, Time Series Analysis: Forecasting and Control, Prentice-Hall International, Inc., New Jersey, 2008.
[19] D. Montgomery, Design and Analysis of Experiments, Wiley, New Jersey, 2009.