



Artificial intelligence vs. classical approaches: a new look at the prediction of flux decline in wastewater treatment

Hossein Mashhadi Meighani^a, Amin Dehghani^a, Fatemeh Rekabdar^{b,*}
Mahmood Hemmati^b, Iraj Goodarznia^a

^aDepartment of Chemical and Petroleum Engineering, Sharif University of Technology, Tehran, Iran

^bPolymer Science and Technology Division, Research Institute of Petroleum Industry (RIPI), Tehran, Iran
Tel. +98 21 48253151; email: rekabdarf@ripi.ir

Received 22 November 2012; Accepted 2 February 2013

ABSTRACT

This study compares the performance of three different approaches to modeling namely the classical pore-blocking models, artificial neural networks (ANN) and the novel genetic programming (GP) approach. Among the available models proposed by Hermia, standard pore-blocking and cake filtration models were opted because of their better fitness with experimental measurements. A feedforward backpropagation network using Bayesian Regulation as well as Levenberg–Marquardt training methods was developed based on the experimental results. Network inputs include the controlling parameters of permeate flux namely: temperature, transmembrane pressure, crossflow velocity, pH, and filtration time. The architecture and internal parameters of the network have substantial effect on the prediction performance of the ANN. Hidden layers and neuron numbers were regulated using trial-and-error approach. The individual program proposed by GP, which has satisfied the required fitness value after 500 generations, had a depth of 10 among a population of 700 individuals. Relative error with respect to experimental results was used to compare the aforementioned models. It was found that ANN outperformed pore-blocking and GP models. The GP-based model had an acceptable coincidence with the experimental data and its ability to correlate the input and target variables by a mathematical relation showed the high potentiality of GP as a modeling tool.

Keywords: Flux decline prediction; Artificial neural network; Genetic programming; Pore blocking model

1. Introduction

The increasing production of wastewater especially oily wastewater, in both residential and industrial areas, such as refinery plants and production units associated with the increasing strictness of environmental regulations calls for the development of new

and more efficient techniques for disposal or reuse of wastewater [1].

Traditional techniques used in an oil-in-water emulsion separation, for example, centrifugation, de-emulsification, gravity settlement, and air flotation, have some operational difficulties and also do not lead to the purities of interest [2,3].

*Corresponding author.

Recently, membrane filtration has been playing a fundamental role in the separation processes and has been broadly implemented worldwide for its potentialities, high efficiency, relatively high ability in transporting specific components, operational cost-effectiveness, and its easy scaling up features. Although these superiorities have been approved in many industrial and laboratory applications, some disadvantages are still assigned to the membrane processes, for example, fouling of their pore spaces due to concentration polarization, and chemical interaction with water constituents. This phenomenon, which causes a permeate flux decline, is the main limitation of membrane processes [4].

In response, being familiar with such a critical issue and benefiting from some knowledge on the fouling of membrane and its associated factors is of much vitality. Modeling the membrane filtration processes has recently been the subject of much interest and accounted as a challenging issue in many studies.

With the complexities such as hydrodynamic dependency and complicated nature of cake layer, there has been experienced a common poor knowledge in description of the phenomena happening in the filtration process. This significantly weakens the phenomenological models, and as a result, more attention has been paid to the semi-empirical and black box models [5,6]. In this regard, Hermia's pore-blocking models have been exploited by many of the researchers. Mohammadi et al. [3] have used a modified form of pore-blocking model to investigate the permeate flux decline in the synthesized oily water separation by reverse osmosis membrane. Their results show an inconsistency between experimental measurements and model predictions. In the process of oily emulsion microfiltration by mullite ceramic membrane and using the pore blocking model, Abbasi et al. [7] studied the effects of pressure, cross flow velocity (CFV), temperature, and oil concentration on flux decline. They observed that different pore-blocking models had better results in different times of filtration. Vela et al. [8] also obtained the same results as that of Abbasi et al. [7] for different operating conditions. However, it is a well-known fact that the parameters associated with both non-linear correlations and pore-blocking models could not be generalized to represent variation in flux decline with feed concentrations and pressure differentials [2].

Modeling techniques based on the direct analysis of experimental data (descriptive models) appear to be the promising alternatives for the models that use phenomenological hypotheses (i.e. knowledge-based models) [5]. Artificial neural networks (ANN) are the most famous and popular black box modeling tool in

this respect. The reason lies in the fact that ANN has been widely used to predict the permeate flux recently.

Sargolzaei et al. [9] showed the capability of ANN model to predict the starch removal performance using a hydrophilic polyethersulfone. Aydiner et al. [10] compared performance of ANN and Koltuniewicz's method in modeling of flux decline rate of crossflow microfiltration of a mixture in presence of phosphate and fly ash. The study concluded that both methods had satisfactory results. Benefiting from the genetic algorithm (GA) to choose the initial connection weights and biases of backpropagation neural network (BPNN), Ming et al. [11] employed BPNN to predict membranes pervaporation performances at different operating condition. Sahoo and Ray [5] utilized GA to optimize configuration of a radial basis function network on predicting flux decline in crossflow membranes. Shokrian et al. [12] showed that one can use ANN to predict separation factor of C_3H_8 from CH_4 and H_2 by synthesized membrane. Curcio et al. [13] constructed a three layer ANN which was used to model the ultrafiltration of BSA solutions under pulsating conditions. Nandi et al. [2] delivered an ANN model, which later compared with conventional pore blocking model to predict the permeate flux of dead-end filtration of oil-in-water emulsions.

In neural network modeling, there is a relationship between function complexity and network size in such a way that optimal number of hidden layers in the network grows proportionally by the complexity of the problem [14]. In genetic programming (GP), however, there is no need to have any prior knowledge neither on the physics of the problem nor on design of the model at hand. Moreover, the ANN does not return any function from the model, while GP results in a mathematical correlation between the parameters. Recent studies have focused more on the GP modeling. Hong and Bhamidimarri [15] used GP in order to model the dynamic performance of municipal activated sludge waste water treatment plants. The model had a relative mean square error (RMSE) on the training set of 1.34 mg/L and an RMSE on the testing set of 1.57 mg/L. RMSE value of 1.57 mg/L and 81.4% of R^2 for the testing data indicates very satisfactory performance of the model. Hwang et al. [16] found the pattern of fouling variation in filtration of drinking water using GP. The model used some of the operating conditions such as flow rate and filtration time and some of the feed water quality parameters like turbidity and algae pH, as the input variables. They developed a successful model for the pattern of membrane resistance during the operational period, though the dramatic increase in the membrane resistance was

not predicted at all. Shokrkar et al. [17] studied the treatment of oily wastewaters with synthesized mullite ceramic microfiltration membranes and proposed a new approach for modeling of membrane flux using GP. The population size in their model was 500 individuals, number of generations was 90, and the maximum depth for the trees was assumed to be 7. The results thus obtained from their model represented a good agreement with the experimental data, having an average error of less than 5%. Some of the works which have implemented the GP approach are briefly described in Table 1.

In this study, the experimental results reported by Salahi et al. [20] have lend the authors a hand on the comparative assessment of three different modeling methods including pore blocking models, ANN, and GP. The overall performance, accuracy, advantages, and blind spots of each method are investigated. The main criteria applied for the performance study are considered as the run time and ease of use of the models. Relative error is the cornerstone of accuracy evaluation and according to the output format; the overall strengths and weaknesses of the models are estimated within the scope of the article.

2. Theory and model description

2.1. Pore-blocking model

As a function of the solute characteristics, mainly particle size, four different mechanisms of fouling are probable, namely complete pore-blocking model, standard pore-blocking model, intermediate pore blocking model, and cake filtration model. The models have been proposed by Hermia [21] based on the constant

pressure dead-end filtration; however, the acceptable coincidence of their prediction with the experimental results of cross flow filtration has proved them applicable in such processes [22–25].

In these models, the general equation is as follows:

$$\frac{dJ}{dt} = -K_j(J - J_0)^{2-n} \quad (1)$$

where $n = 2.0$ for complete pore blocking; $n = 1.5$ for standard pore blocking; $n = 1.0$ for intermediate pore blocking, and $n = 0$ for cake filtration; J represents the permeate flux, K_j is a constant, and J_0 is the limiting flux. The equations of the pore-blocking model are given in the Table 2.

In these equations, the constants K_b and K_i are related to the blocked surface area per unit permeate volume; A represents the membrane surface; K_s is a function of retained particles volume per unit permeate volume, and K_c depends on both cake resistance and concentration [3].

Complete pore blocking occurs when the solute particles are larger than the mean pore size of the membrane. The particles or molecules finally deposit

Table 2
Pore blocking characteristic models

Characteristic equation	Model
$Ln(J) = Ln(J_0) - K_b t$	Complete pore blocking
$\frac{1}{J} = \frac{1}{J_0} + K_i A t$	Intermediate pore blocking
$\frac{1}{J^{0.5}} = \frac{1}{J_0^{0.5}} + K_s t$	Standard pore blocking
$\frac{1}{J^2} = \frac{1}{J_0^2} + K_c t$	Cake filtration

Table 1
Summary of research on the membrane filtration modeling using GP

Reference	Membrane process	GP model characteristics	Error	Year
Shokrkar et al. [17] (Prediction of permeation flux decline during MF of oily wastewater using GP)	Mullite ceramic membrane, MF	500 population, 90 generation and maximum tree depth of 7	0.2–13%	2011
Okhovat and Mousavi [18] (Modeling of arsenic, chromium and cadmium removal by nanofiltration process using GP)	Polymeric (Polyamide) membrane, NF	200 population, 250 generation and maximum tree depth of 50	RMSE = 0.005–0.02	2011
Suh et al. [19] (Application of GP to develop the model for estimating membrane damage in the membrane integrity test using fluorescent nanoparticle)	Synthesized membrane, MF	140 population, 220 generation and maximum tree depth of 7	MAE = 0.83	2011
Hwang et al. [16] (Prediction of membrane fouling in the pilot-scale microfiltration system using GP)	Polymeric (PVDF) membrane, MF	–	8%	2009

on the surface of membrane and the available paths through the membrane would successively reduce. In standard pore blocking mechanism, smaller particles or molecules plug the internal pores of membrane and the reduction of pore volume will be proportional to the filtered permeate volume. The intermediate blocking occurs when the membrane mean pore size is nearly the size of particles or molecules. In this model, each particle can block the membrane pores and deposit on the previously deposited particles. Particles greater than pore volume in size, accumulate on the surface of membrane and gradually a porous concentrated layer called “cake” will cover the surface. This layer dramatically affects the membrane performance. Cake filtration model describes the variation of permeate flux in this fouling mechanism. Salahi et al. [20] used the pore-blocking models and reported a better prediction of standard pore-blocking and cake filtration models. R^2 values of these models have been summarized in Table 3. As it can be traced, in early times, standard pore-blocking model has a better fitness, while as time elapses, cake filtration model shows a higher accuracy in prediction.

2.2. Artificial neural network

ANN models are the algorithms used for cognitive tasks, such as learning and optimization. They are powerful tools used in modeling complex systems that seek to simulate human brain behavior by processing data on a trial and error basis. Then it tries in learning how to avoid repeating an error the next time a similar situation occurs. Neural networks have proved to be promising solutions for approximating complex and non-linear functions; they have also exhibited a great accuracy in complex systems when there is a lack of rigorous knowledge on their behavior [26].

Table 3
 R^2 values of pore blocking models [20]

Experiment number	Standard pore blocking	Complete pore blocking	Intermediate pore blocking	Cake filtration
1	0.9321	0.8677	0.9748	0.9993
2	0.9060	0.8732	0.9568	0.9991
3	0.9232	0.8349	0.9607	0.9974
4	0.9232	0.8732	0.9607	0.9974
5	0.9060	0.8349	0.9568	0.9991
6	0.9188	0.8458	0.9678	0.9999
7	0.9110	0.8400	0.9606	0.9996
8	0.9196	0.8681	0.9583	0.9970
9	0.8710	0.8027	0.9241	0.9852

Neural networks are configured to obtain a desired set of outputs, by the identification of the connection and strengths between given input and desired output. According to the desired output, neural networks can be designed to perform the classification, regression predictions, or desired tasks. The former task requires neural networks to label correctly the relationship between input attributes with one class of a predefined set of classes (e.g. classifying cells for cancer diagnosis), while the latter requires to map an n-dimensional input vector with a real value variable (e.g. temperature forecast). The main difference between these two networks is their output representation, since both can be modeled with the same learning method and architecture [27].

In order to use a neural network for prediction, it is necessary to define its topology (i.e. number of layers and neurons) and additional variables such as weights and bias of each layer as well. These variables are obtained during the learning process. Various methods for network learning are available; among them is a method which should set the weights and bias of the network explicitly, and requires a good knowledge of the process to be modeled. Another method is by feeding the neural network with patterns and iteratively changing its weights in order to adjust the network to the provided patterns [28].

One of the most common neural networks architectures is the Feedforward backpropagation (FFBP); the reason lies in the fact that it has been employed successfully to model many different tasks. Backpropagation networks describe how a network with neurons connected in only one direction (forward) is trained. Backpropagation is a type of supervised training that provides sample inputs and outputs to the network. In FFBP networks, actual outputs are compared against predicted outputs and so, error is calculated. Based on the resulted error, and starting from the output layer, the backpropagation algorithm adjusts the weights until the input layer is reached [29]. Fig. 1 depicts a general schematic of how an ANN works.

A successful BPNN requires the determination of internal parameters such as network architecture and initial weights to meet the required performance [30].

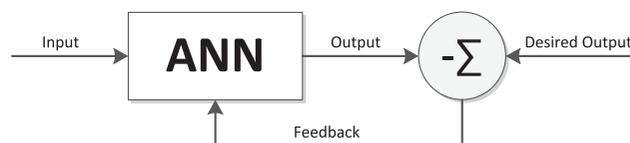


Fig. 1. Artificial neural network, training and learning concepts.

An ineffective design of the network will last into some unreliable consequences. Finding a suitable architecture and the corresponding weights of the network is a complex task due to the lack of theoretical parameters or optimal values and requires the trial and error approach using different initializations and architecture [31].

In initial steps of training, errors of both training and validation data are reduced. After several steps, the error of training data decreases while that of validation data increases. As a result, the network is over-trained and its generality decreases. Hence, the training process must be continued until the validation data error decreases. Testing data set is used to test the trained network for unseen patterns (The results of which are known to the researcher but not used in the training procedure). The network generalizes well when it sensibly interpolates these new patterns [9].

2.2.1. Architecture and model basis of the proposed neural network

As mentioned earlier, the BP algorithm is based on a learning rule by which the weights are evolved in order to minimize the mean of squared differences between the desired and actual values of the output neurons. The standard BP algorithm suffers from a few drawbacks such as the risk to converge in local minima and long computational time. In order to improve its performance, two different types of high-performance BP training algorithms employing different optimization techniques are used in this study. These algorithms are Levenberg–Marquardt (LM) and Bayesian Regularization (BR) algorithms that brief explanations are offered in the following.

LM method. Similar to quasi-Newton methods, the LM algorithm was designed to approach a second-order training speed without having to compute the Hessian matrix. The method combines the best features of the Gauss–Newton method and the steepest-descent method, but avoids many of their limitations [32].

BR method. This method is the modification of the LM training algorithm to produce a well-generalized network. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network that can generalize well. This algorithm can train any network as long as its weights, inputs, and activation functions have derivative functions. In the configuration of a neural network model, one of the most important factors is to determine the number of hidden layers to be used and also the number of neurons located in each layer. Although some researchers suggest that solely one

hidden layer is usually sufficient [33,34], the introduction of additional hidden layers allows for the fit of a larger variety of target functions and enables the approximations of complex functions with fewer connection weights [35]. Hecth and Nielsen [36] suggested that the upper limit for the number of hidden layer neurons should be smaller than $2N_i + 1$, where N_i is the number of input neurons, considered in order to ensure that ANNs are able to approximate any continuous function.

In data analysis, numerous combinations of network geometry with two and three number of hidden layers were tested. For each hidden layer, various combinations having 3–20 neurons were tried. Each case was examined using both LM and BR training algorithms. The results of MSEs for all the investigated layer combinations are tabulated in Table 4.

The networks trained by LM algorithm demonstrate a better prediction performance compared with those by BR algorithm.

The best simulation results were obtained with model architectures of 5-3-1, 5-20-1 and 5-10-10-1 accompanied by LM as the training method. The first architecture had a fewer number of layers well as lower MSE and was hence selected as the optimum model. The architecture of a multi layered network used in this work is presented in Fig. 2.

Settings used in the development of the neural network are summarized in Table 5.

Table 4
MSEs of various examined neural networks

Model structure (Hidden Layers and Output Layer)	Transfer function	Learning algorithm	MSE	NMSE
5-30-1	TTP	BR	110.56	0.413
5-20-1	TTP	BR	47.65	0.178
5-10-1	TTP	BR	267.46	1.000
5-5-1	TTP	BR	81.82	0.306
5-3-1	TTP	BR	208.22	0.779
5-30-1	TTP	LM	0.91	0.003
5-20-1	TTP	LM	0.44	0.002
5-10-1	TTP	LM	2.64	0.010
5-5-1	TTP	LM	2.13	0.008
5-3-1	TTP	LM	1.28	0.005
5-10-5-1	TTTP	BR	28.20	0.105
5-10-10-1	TTTP	BR	32.43	0.121
5-10-5-1	TTTP	LM	1.52	0.006
5-10-10-1	TTTP	LM	1.40	0.005

BR: Bayesian regulation; LM: Levenberg–Marquardt; T: Tansig; P: Pureline; MSE: Mean Square Error; NMSE: Normalized Mean Square Error.

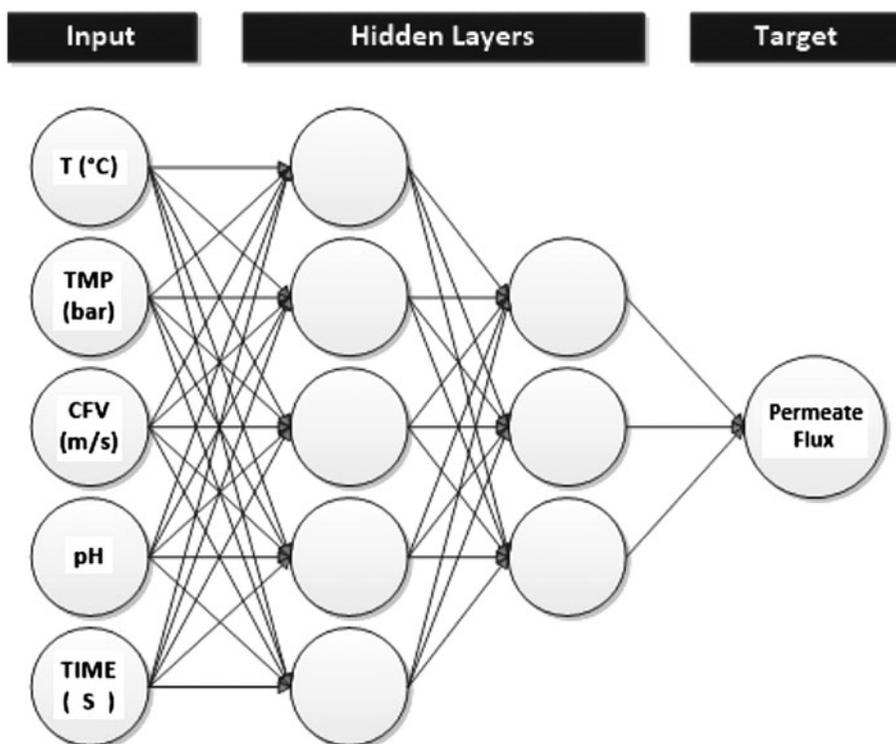


Fig. 2. Architecture of the developed backpropagation multilayer ANN used for the modeling of permeate flux.

Table 5
Settings used in the development of the neural network

General information		Neuron numbers				Transfer functions		
Network type	Training function	Layer no.	Hidden Layer #1	Hidden Layer #2	Output Layer	Hidden Layer #1	Hidden Layer #2	Output Layer
FFBP	TRAINLM	4	5	3	1	TANSIG	TANSIG	PURELIN
Performance function	Epochs	max fail	mem_reduc	min_grad	mu	mu_Inc.	mu_dec	mu_max
MSE	150	75	1	1×10^{-10}	1×10^{-3}	10	0.1	$1 \times 10^{+10}$

FFBP: Feed-forward back propagation; TRAINLM: Levenberg–Marquardt training function; MSE: Mean squared error.

Having chosen the backpropagation learning method, input parameters have to be normalized between 0 and 1 to prevent any numerical overflow. In this study, we normalized the input data according to Eq. (2),

$$P_{Normalized} = \frac{P - P_{min}}{P_{max} - P_{min}} \quad (2)$$

where $P_{Normalized}$ is the normalized value of parameter P , P_{min} and P_{max} are respectively the minimum and maximum values of each parameter.

One of the most widely used transfer function is tan-sigmoid (tansig) and is used in this work in hidden layers according to Eq. (3).

$$tansig(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

tansig function maps the inputs in the range of -1 to $+1$.

In the output layer, pure line function is used as the transfer function. This function produces the output equal to its input variable.

Relative error between ANN output and experimental data used for the sake of performance evaluation of the implemented ANN model was calculated via Eq. (4).

$$\text{Error \%} = \frac{O_{ANN} - O_{EXP}}{O_{EXP}} \times 100 \quad (4)$$

2.3. Genetic programming

The inspiration of Koza by Darwin’s theory, which explains the ability of computers in solving the problems by some training and validating algorithms, resulted in the introduction of GP [37]. GP is a progressive approach that generates a model program as a function of the parameters affecting a target variable

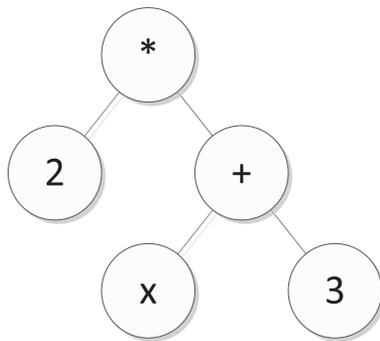


Fig. 3. Tree-like structure of an individual.

on the basis of biological evolution. In GP, the programs are presented in tree-like structures in which the leaves of the tree represent the independent variables or integers, while the nodes having their arguments as branches are the functions. The following tree shown in Fig. 3, demonstrates a $2 \times (3 + x)$ program in the population with three terminals and two functions [38].

The procedure of developing a nonlinear model in GP begins with generation of a random population using the available methods. The population is referred to the number of individual programs. Forming the individuals in the next generation is the second step which consists of two main methods namely Reproduction and Cross over. In reproduction, old individuals copy themselves to a new population without any changes and in cross over, the genetic materials from two individuals are mixed to form the offsprings. Fig. 4 is a representation devoted to this issue. The most important genetic operation in GP is cross over. The reason lies in the fact that it is the main source of having new individuals in the next generation. There are also other genetic operations such as mutation, edition, encapsulation, permutation, and decimation.

After substitution of the new population, it is required to specify how good an individual is in the population. The fitness values of all the individuals are determined using an appropriate fitness function and the best ones are selected for the next population.

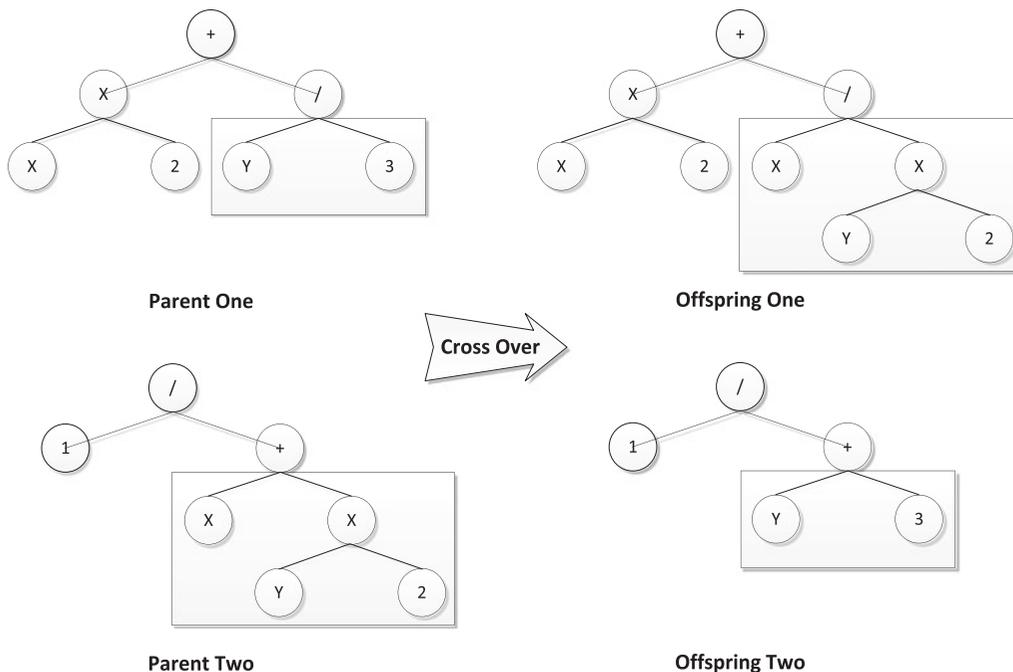


Fig. 4. Cross over operator in action representing original parents, cross over points and generated offsprings.

These steps are continued until the termination criterion is satisfied. This criterion may be either fitness value, maximum depth of trees or number of generations. Fig. 5 is a flow chart of GP.

2.3.1. Specifications of the proposed GP model

In the experiment design section, nine series of tests have been conducted in order to obtain the variation of permeate flux. The permeate flux is regarded as a function of some controlling parameters such as transmembrane pressure (TMP), feed temperature, CFV, pH, and filtration time [20]. The objective of presented modeling is hence to find the mathematical relation between these parameters. In this model, the selected function set was {"+", "-", "*", "exp", "sqrt"}. After normalizing the variables, 75% of them were introduced to the program as training data. Mean square error (MSE) between experimental values and those of predicted from the GP model was used as the fitness function. Population size was set to 700, the number of generations was 500, and the maximum depth of each individual was 10 to avoid the bloating phenomenon that tree depth increases without any improvement in the fitness [39]. The genetic operations were set to be only cross over and mutation and their probabilities were 60 and 40%, respectively. Lexictour method was chosen as the selection method and the expected number of children was assumed to be Rank 89 meaning that the expected number of children for each individual is based on its rank in the population and on the state of the algorithm (how far it is from the maximum allowed generation).

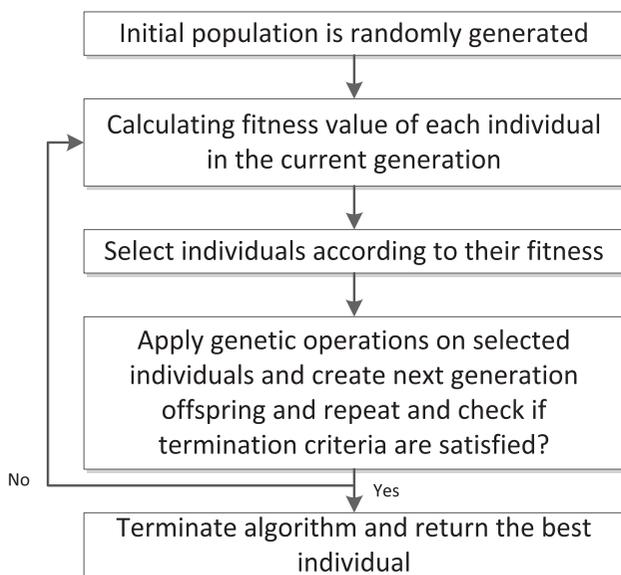


Fig. 5. GP flowchart.

The MATLAB®GP toolbox published by Sara Silva was used in this study.

3. Results and discussion

This section is comprised of four parts. First, the developed GP-based mathematical relation between target variable (permeate flux) and input parameters is presented. Next, permeate flux predictions using pore blocking, GP, and ANN models are demonstrated. The accuracies of the proposed models are discussed in terms of relative error and R² parameter in the third part. The last part is devoted to the complete investigation of selected models considering performance, accuracy, advantages, and disadvantages.

3.1. Development of GP model

Normalized values of the input parameters consisting of filtration time, temperature, TMP, CFV, and pH were introduced for the development of GP model and the initial generation was created with a population size of 700 and poor fitness value as expected. Evolutionary algorithm was working properly, as observed, since the values of MSE were decreasing. Having satisfied the termination criteria (number of generations), the GP model would be achieved. The following is the best GP model:

$$P_1 = (-X_1 + X_2X_5 + 3X_2 - X_3) + X_5(X_2X_3 - X_2X_4) + X_2X_4 - e^{e\sqrt{X_3}}$$

$$P_2 = \sqrt{P_1 - (2X_1 + X_2(1 - 3X_3))\sqrt{-X_1 + X_2(X_4 + 1) - 2e^{X_3}} - X_4(X_1 - 3X_2)}$$

$$P_3 = \sqrt{X_3(X_2 - X_1) - X_1X_4 + X_2X_3} - \sqrt{X_2X_3 + X_2 - X_1} + \sqrt{X_1 - \sqrt{X_1}X_3X_4\sqrt{e^{X_4}}} + \sqrt{\sqrt{X_3} + e^{X_3}}$$

$$P_4 = -\sqrt{-X_1 + X_2(2X_4 + 1) - e^{X_5}} - X_2 - X_3X_4e^{\frac{X_4}{2}} - \sqrt{\sqrt{X_1}(X_3X_4(-\sqrt{X_3X_4} - 1) - 1) + X_1 + X_2X_3(1 - X_4^2)}$$

$$P = P_2 + P_3 + P_4$$

where X₁=filtration time (s), X₂=feed temperature (°C), X₃=TMP (bar), X₄=CFV (m/s), X₅=pH and P=permeate flux (l/m²h). In this formulation, √X will return zero if X is negative and equals roots of X if it is positive.

3.2. Permeate flux prediction

Eight sets of the experimental data, obtained from the literature [20], were utilized to investigate the accuracy of the pore blocking, GP, and ANN models. The operating conditions and experiment sets are briefly described in Table 6.

Having constructed the aforementioned models, their prediction performances were evaluated. Three experimental sets associated with the model prediction and experimental values are shown in Figs. 6–8. Obviously, in the initial steps where the permeate flux values are sharply declining, all models have lower accuracies and among them, pore-blocking model exhibits the worst outcome. As time passes, the rate of permeate flux decline decreases due to fouling of the pore volume, and in this interval, the experimental measurements and predicted values are in an excellent agreement. Generally, the ANN model presented a better prediction precision; the reason lies in the fact that ANN is a powerful interpolator.

Through all the experiment sets, three permeate fluxes have been chosen in the range of initial, middle,

Table 6
Operating conditions of filtration experiments

Experiment set	T ($^{\circ}\text{C}$)	TMP (bar)	CFV (m/s)	pH
1	25.0	3.0	0.75	7
2	25.0	4.5	1.25	10
3	37.5	1.5	0.75	10
4	37.5	3.0	1.25	4
5	37.5	4.5	0.25	7
6	50.0	1.5	1.25	7
7	50.0	3.0	0.25	10
8	50.0	4.5	0.75	4

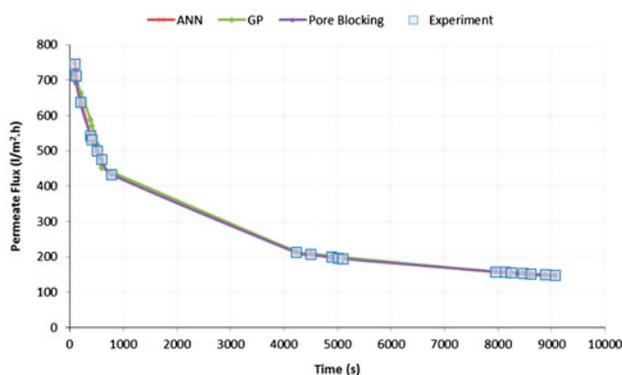


Fig. 6. Permeate flux prediction by pore blocking, GP and ANN models (temperature = 25°C , TMP = 4.5 bar, CFV = 1.25 m/s and pH = 10).

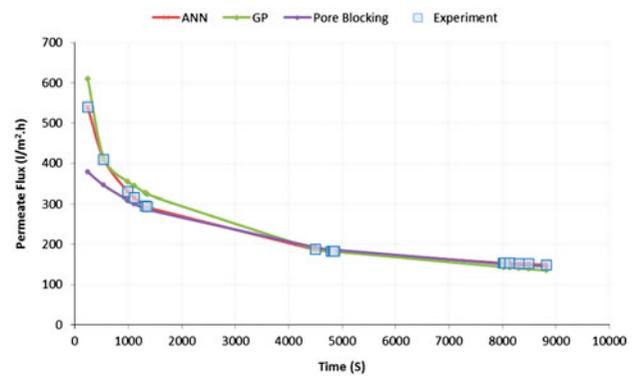


Fig. 7. Permeate flux prediction by pore blocking, GP and ANN models (temperature = 50°C , TMP = 4.5 bar, CFV = 0.75 m/s and pH = 4).

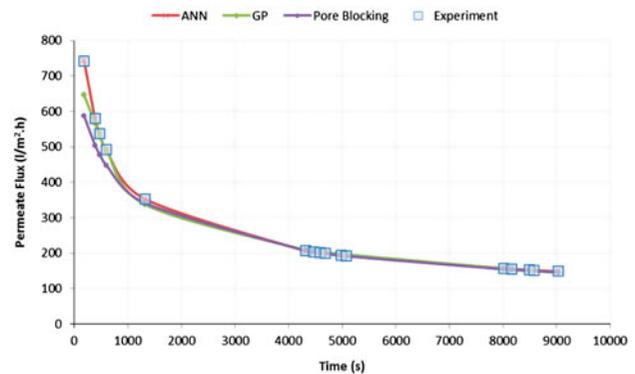


Fig. 8. Permeate flux prediction by pore blocking, GP and ANN models (temperature = 37.5°C , TMP = 3 bar, CFV = 1.25 m/s and pH = 4).

and last times and the experimental measurements, modeled values, and their related errors are shown in Table 7.

3.3. Error analysis

The regression lines for ANN, GP, and pore blocking models are shown in Figs. 9–11. The respective values of R^2 are 0.9999, 0.9723, and 0.9799. As clearly can be seen in Fig. 9, the ANN model and experimental measurements are in a complete coincidence. Although R^2 values obtained for pore blocking model and GP are in the same range, the data points of the GP model are more scattered around the regression line and one can say that pore-blocking model is of higher accuracy. In lower permeate flux values, both GP and pore-blocking models are in good agreement with experimental measurements; however, as the permeate flux goes up, GP predicted values show more deviation from unit slope line. It should be

Table 7
Experimental measurement, model values and related errors of experiments

Experiment set	EXP	Output (l/m ² h)			Relative error (%)		
		ANN	GP	Pore blocking	ANN	GP	Pore blocking
1	445.70	444.42	388.12	449.31	0.286	12.919	0.809
	416.90	417.02	371.15	419.65	0.028	10.973	0.660
	366.00	367.88	355.29	368.16	0.515	2.926	0.592
2	344.80	346.08	345.99	346.19	0.370	0.344	0.403
	335.40	336.37	341.14	336.58	0.290	1.710	0.352
	315.30	315.08	327.70	315.80	0.069	3.933	0.158
3	311.70	311.21	324.42	312.06	0.156	4.079	0.116
	286.10	284.26	305.58	286.19	0.642	6.809	0.030
	278.10	275.77	301.30	278.05	0.836	8.343	0.017
4	167.60	167.27	167.58	167.48	0.197	0.014	0.074
	163.30	163.20	164.04	163.11	0.059	0.455	0.114
	157.30	157.62	158.91	157.16	0.202	1.021	0.086
5	154.50	155.02	156.41	154.42	0.338	1.233	0.049
	122.70	123.05	119.37	122.76	0.289	2.715	0.052
	121.40	121.61	117.40	121.44	0.177	3.292	0.037
6	120.70	120.91	116.43	120.80	0.172	3.538	0.084
	118.10	117.95	112.26	118.12	0.130	4.942	0.021
	117.30	117.07	111.00	117.34	0.199	5.370	0.031
7	745.10	751.11	701.16	716.39	0.807	5.897	3.853
	711.20	717.56	692.45	690.84	0.894	2.636	2.862
	637.70	638.86	663.28	627.99	0.182	4.011	1.522
8	543.20	541.44	586.60	541.02	0.323	7.990	0.401
	531.10	529.74	570.27	529.76	0.255	7.374	0.252
	500.00	499.30	517.99	499.78	0.141	3.599	0.045

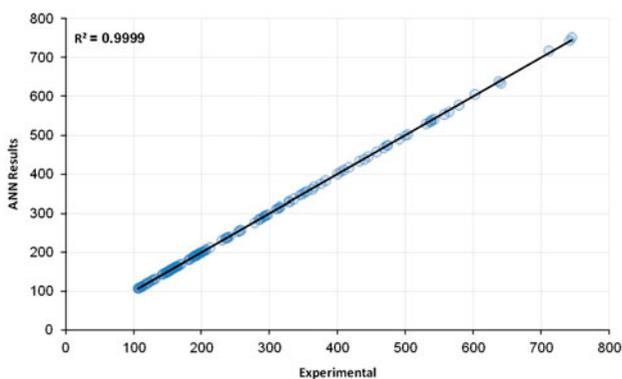


Fig. 9. ANN model values vs. experimental measurements.

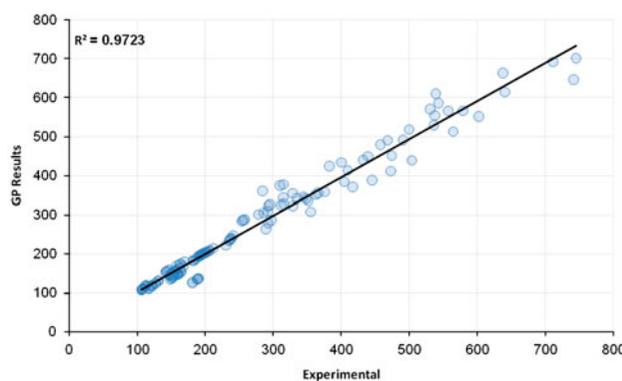


Fig. 10. GP model values vs. experimental measurements.

noted that approximately the same values of R^2 for the models could not be the only judging factor in comparing their precisions. To have a better view of their accuracies, relative errors were calculated and are presented in Table 7.

In Figs. 12–14, relative error of all the modeling predictions has been represented with respect to each

other. It is clear from Fig. 12 that relative errors of GP and ANN models lie in the range of 0–30 and 0–1.2%, respectively. Fig. 13 shows the values of pore-blocking model relative and ANN model error. As shown in Fig. 13, it is clear that the ANN model has a greater accuracy in comparison with pore-blocking model. Fig. 14 also represents the same orders of relative

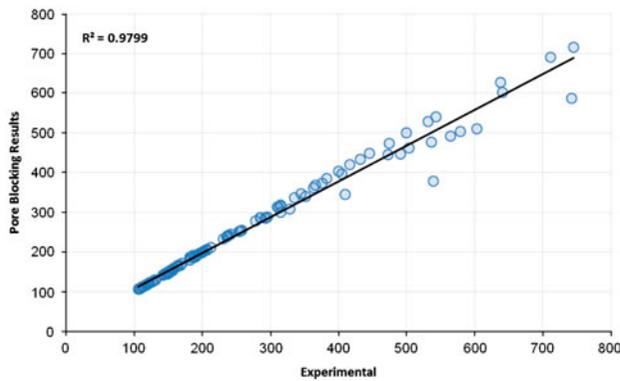


Fig. 11. Pore blocking model values vs. experimental measurements.

error for both pore-blocking and GP models. Although in both Figs. 12 and 13, the ANN relative error are much less than GP and pore-blocking models, a closer look at Fig. 13 shows that data points are more accumulated near the horizontal axis which means that pore-blocking model has a lower relative error compared to GP. According to Figs. 12 and 13, most of ANN prediction errors are less than 0.4% and in Fig. 14, it is observed that relatively high numbers of GP and pore-blocking errors lie in the range of 0–10 and 0–5%, respectively.

3.4. Performance evaluation

In Section 3.1 to 3.3, accuracies of the proposed models were evaluated from different points of view, for example, prediction precision in different time intervals and for various values of permeates flux. R^2 values of all these models were studied and it was found that ANN predicted values accurately matched experimental measurements. GP and pore-blocking

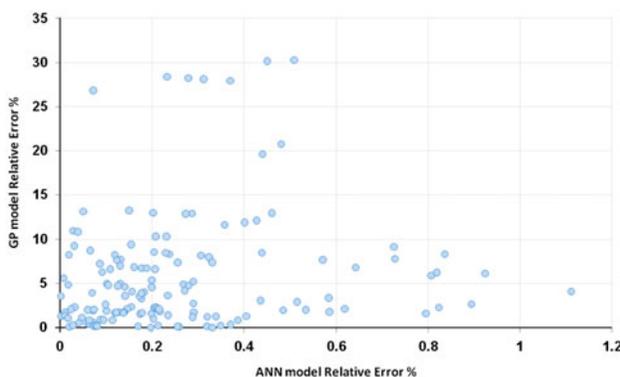


Fig. 12. GP model relative error vs. ANN model relative error.

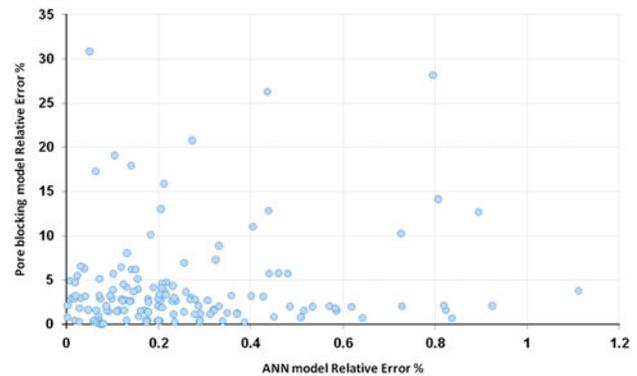


Fig. 13. Pore blocking model relative error vs. ANN model relative error.

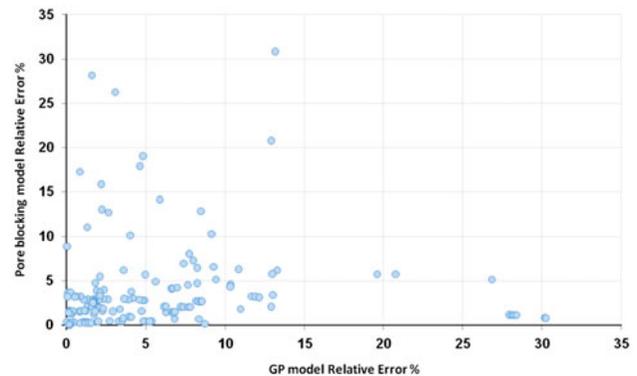


Fig. 14. Pore blocking model relative error vs. GP model relative error.

models had the same values of R^2 . Also, relative errors of the models were analyzed with respect to each other and it was concluded that the ANN errors begin from near zero values up to 1.2%. These limits for GP and pore-blocking model were zero and 30%, respectively. Deeper analysis of the obtained results showed that, most of the model predicted values for ANN, pore blocking, and GP had relative errors below 0.4, 5, and 10%, respectively. More detailed analysis of relative error frequencies are depicted in Figs. 15–17 which further confirm the observations. These figures show how many data points lie in each relative error interval. The vertical axis is number of data points and the horizontal axis depicts relative error.

It is worth to mention that error analysis can hardly be the sole criterion in choosing the right model as various parameters such as ease of use, run time, giving an explicit relation between input and output parameters, etc. also have to be taken into account. With this vision, although GP and pore-blocking models have higher error percentages, they

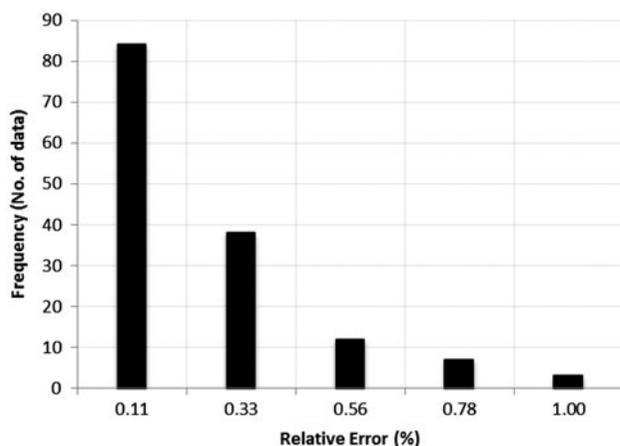


Fig. 15. Frequency of relative errors for ANN model.

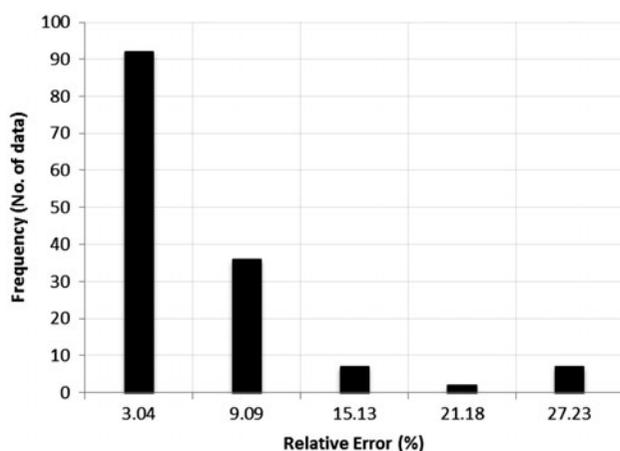


Fig. 16. Frequency of relative errors for GP model.

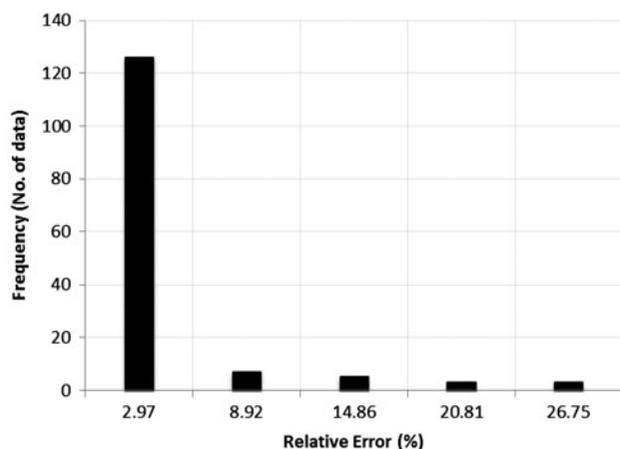


Fig. 17. Frequency of relative errors for pore blocking model.

return an explicit mathematical relation which could be used in any experimental or simulation study. Integration of ANN model into other studies comes with its downsides. Also, the fact that ANN is a poor extrapolator should not be overlooked.

Until today, pore-blocking model has been broadly used for modeling in the experimental works. This was due to the fact that this model gives a linear and simple formula that predicts the permeate flux. In some cases of controlling the permeate flux decline, having some knowledge of the physical factor that causes the fouling of membrane can well help us reduce fouling phenomena by changing the operation conditions or feed characteristics. Taking into account this fact, pore-blocking models are of particular importance since each model expresses a different fouling mechanism.

The main advantages of GP model are returning an explicit relationship which could be used anywhere and eliminating the need for arrangement of the model structure in contrary to ANN. This should be considered especially where the nature of the problem at hand is complicated. The main disadvantages of GP are the high run time and higher relative error compared with the other two models. Considering all the strengths and weaknesses of each model for the sake of constructing a model for membrane processes, ANN, GP, and pore-blocking models can each be taken into consideration.

4. Conclusion

A thorough comparison of three modeling techniques in permeate flux decline prediction was put under vast concentration in this article. These are pore-blocking model, GP, and ANN. The aim was to predict the permeate flux decline in membrane process. To achieve this aim, the experimental data were obtained from the published literature. Better pore-blocking models for the data were the cake filtration and standard pore-blocking models. In the case of ANN modeling, optimum structure of the network was obtained in a trial and error manner which is a network with two hidden layers each consisting of five and three neurons, respectively. FFBP learning method was employed to train the network. For the development of GP model, the population size of each generation was set to be 700 and the maximum depth of individuals was assumed 10. MSE was selected as the fitness function where the crossover and mutation were the genetic operations. Having developed all the prediction models, experimental measurements were modeled and relative error of each predicted value was calculated. Error analysis

showed that ANN was the least erroneous among the three. Returning an explicit relation and ease of use were the main features of pore-blocking model. GP not only gives a mathematical relation, but also has no need for any previous knowledge on the model structure. The key superiorities and blind spots of the methods were also fully reviewed. Therefore, it essentially depends on the case in hand that which method would return the best model to predict the permeate flux decline rate.

Acknowledgements

We would like to acknowledge and extend our heartfelt gratitude to Mr Salahi for his valuable help regarding permission for using his experimental results.

References

- [1] J.Y. Seo, A. Vogelpohl, Membrane choice for wastewater treatment using external cross flow tubular membrane filtration, *Desalination* 249 (2009) 197–204.
- [2] B.K. Nandi, A. Moparthy, R. Uppaluri, M.K. Purkait, Treatment of oily wastewater using low cost ceramic membrane: Comparative assessment of pore blocking and artificial neural network models, *Chem. Eng. Res. Des.* 88 (2010) 881–892.
- [3] T. Mohammadi, M. Kazemimoghadam, M. Saadabadi, Modeling of membrane fouling and flux decline in reverse osmosis during separation of oil in water emulsions, *Desalination* 157 (2003) 369–375.
- [4] H. Strathmann, L. Giorno, E. Drioli, *An Introduction to Membrane Science and Technology*, Institute on Membrane Technology, CNR-ITM, Consiglio nazionale delle ricerche, Rome, 2006.
- [5] G.B. Sahoo, C. Ray, Predicting flux decline in cross flow membranes using artificial neural networks and genetic algorithms, *J. Membr. Sci.* 283 (2006) 147–157.
- [6] S. Ghandehari, M.M. Montazer Rahmati, M. Asghari, A comparison between semi theoretical and empirical modeling of cross flow microfiltration using ANN, *Desalination* 277 (2011) 348–355.
- [7] M. Abbasi, M.R. Sebzari, A. Salahi, S. Abbasi, T. Mohammadi, Flux decline and membrane fouling in cross flow microfiltration of oil-in-water emulsions, *Desalin. Water Treat.* 28 (2011) 1–7.
- [8] M.C.V. Vela, S.A. Blanco, J.L. Garcia, E.B. Rodriguez, Analysis of membrane pore blocking models applied to the ultrafiltration of PEG, *Sep. Purif. Technol.* 62 (2008) 489–498.
- [9] J. Sargolzaei, M. Haghghi Asl, A. Hedayati Moghaddam, Membrane permeate flux and rejection factor prediction using intelligent systems, *Desalination* 284 (2012) 92–99.
- [10] C. Aydiner, I. Demir, E. Yildiz, Modeling of flux decline in crossflow microfiltration using neural networks: The case of phosphate removal, *J. Membr. Sci.* 248 (2005) 53–62.
- [11] M. Tan, G. He, X. Li, Y. Liu, C. Dong, J. Feng, Prediction of the effects of preparation conditions on pervaporation performances of polydimethylsiloxane(PDMS)/ceramic composite membranes by backpropagation neural network and genetic algorithm, *Sep. Purif. Technol.* 89 (2012) 142–146.
- [12] M. Shokrian, M. Sadrzadeh, T. Mohammadi, C₃H₈ separation from CH₄ and H₂ using a synthesized PDMS membrane: Experimental and neural network modeling, *J. Membr. Sci.* 346 (2010) 59–70.
- [13] S. Curcio, V. Calabr'o, G. Iorio, Reduction and control of flux decline in cross-flow membrane processes modeled by artificial neural networks, *J. Membr. Sci.* 286 (2006) 125–132.
- [14] I. Gómez, L. Franco, J.L. Subirats, J.M. Jerez, Neural network architecture selection: Size depends on function complexity artificial neural networks–ICANN 2006, in: S. Kollias, A. Stafylopatis, W. Duch, E. Oja (Eds.), vol. 4131, Springer, Berlin/Heidelberg, 2006, pp. 122–129.
- [15] Y.S. Hong, R. Bhamidimarri, Evolutionary self-organising modelling of a municipal wastewater treatment plant, *Water Res.* 37 (2003) 1199–1212.
- [16] T.M. Hwang, H. Oh, Y.K. Choung, S. Oh, M. Jeon, J.H. Kim, S.H. Nam, S. Lee, Prediction of membrane fouling in the pilot-scale microfiltration system using genetic programming, *Desalination* 247 (2009) 285–294.
- [17] H. Shokrkari, A. Salahi, N. Kasiri, T. Mohammadi, Prediction of permeate flux decline during MF of oily wastewater using genetic programming, *Chem. Eng. Res. Des.* doi:10.1016/j.cherd.2011.10.002.
- [18] A. Okhovat, S.M. Mousavi, Modeling of arsenic, chromium and cadmium removal by nanofiltration process using genetic programming, *Appl. Soft Comp.* 12 (2011) 793–799.
- [19] C. Suh, B. Choi, S. Lee, D. Kim, J. Cho, Application of genetic programming to develop the model for estimating membrane damage in the membrane integrity test using fluorescent nanoparticle, *Desalination* 281 (2011) 80–87.
- [20] A. Salahi, M. Abbasi, T. Mohammadi, Permeate flux decline during UF of oily wastewater: Experimental and modeling, *Desalination* 251 (2010) 153–160.
- [21] J. Hermia, Constant pressure blocking filtration laws: Application to power-law non-newtonian fluids, *Chem. Eng. Res. Des.* 60 (1982) 183–187.
- [22] T. Mohammadi, A. Esmaelifar, Wastewater treatment of a vegetable oil factory by a hybrid ultrafiltration-activated carbon process, *J. Membr. Sci.* 254 (2005) 129–137.
- [23] S.W. Lee, S.E. Park, S.J. Park, Effect of electrostatic force on surface charges of colloidal particles and retardation of membrane fouling, *J. Ind. Eng. Chem.* 13 (2007) 395–399.
- [24] S.M. Kumar, G.M. Madhu, S. Roy, Fouling behaviour, Regeneration options and on-line control of biomass-based power plant effluents using microporous ceramic membranes, *Sep. Purif. Technol.* 57 (2007) 25–36.
- [25] K.J. Hwang, T.T. Lin, Effect of morphology of polymeric membrane on the performance of cross-flow microfiltration, *J. Membr. Sci.* 199 (2002) 41–52.
- [26] S. Jhaes, F. Raza, Improving Gas Load Forecast Accuracy—A practical Approach. Paper PSIG-06B1 presented at the PSIG Annual Meeting, October 2003, Williamsburg, Virginia, pp. 11–13.
- [27] M. Rucha, P. Cortez, J. Neves, Evolution of Neural Networks for Classification and Regression, Universidade do Minho, Dep. Informática, Portugal, 2004.
- [28] B. Kröse, P. Van der Smagt, *An Introduction to Neural Networks*, eighth ed., The University of Amsterdam, Amsterdam, 1996.
- [29] I. Ferreira, A. Gammiero, M. Llamedo, Design of a Neural Network Model for Predicting Well Performance after Water Shutoff Treatments using Polymer Gels, Mexico, 2012 SPE 153908.
- [30] M. Al-Abri, N. Hilal, Artificial neural network simulation of combined humic substance coagulation and membrane filtration, *Chem. Eng. J.* 141 (2008) 27–34.
- [31] N. García-Pedrajas, D. Ortiz-Boyer, C. Hervás-Martínez, An alternative approach for neural network evolution with a genetic algorithm: crossover by combinatorial optimization, *Neural Netw.* 19 (2006) 514–528.
- [32] I. Dalkiran, K. Danisman, Artificial neural network based chaotic generator for cryptology, *Turk. J. Elec. Eng. Comp. Sci.* 18 (2) (2010) 225–240.
- [33] J.H. Garrett, D.J. Gunaratham, N. Ivezic, in: N. Kartam, I. Flood (Eds.), *Artificial Neural Networks for Civil Engineers: Fundamentals and Applications*, ASCE, New York, 1997, pp. 1–17.
- [34] A.G. El-Din, D.W. Smith, A neural network model to predict the wastewater inflow incorporating rainfall events, *Water Res.* 36 (2002) 1115–1126.

- [35] E. Toth, A. Brath, A. Montanari, Comparison of short-term rainfall prediction models for real-time flood forecasting, *J. Hydrol.* 239 (2000) 132–147.
- [36] R. Hetch-Nielsen, Kolmogorov's mapping neural network existence theorem, *Proceedings of the First IEEE International Joint Conference on Neural Networks*, San Diego, 1987, pp. 11–14.
- [37] J.R. Koza, *Genetic Programming: On the Programming of Computers by means of Natural Selection*, MIT press, Massachusetts, 1992.
- [38] H. Iba, T.K. Paul, Y. Hasegawa, *Applied genetic programming and machine learning*, (Crc press international series on computational intelligence), Taylor and Francis group, LLC, New York, 2010.
- [39] S.G.O. Silva, *Controlling Bloat: Individual and Population Based Approaches in Genetic Programming*, University of Coimbra Faculty of Sciences and Technology Department of Informatics Engineering, 2008.