

Crossflow Microfiltration of Aqueous Suspensions with Guar and Xanthan Gums: Identification of Solutions Using Artificial Neural Networks

Matheus Nonis Passerini¹ , Érica Regina Filletti^{1*} 

Abstract

Artificial Neural Networks (ANNs) are mathematical models used in the computational area that act in an analogous way to the central nervous system of living beings, which possess the ability of acquiring knowledge in a technique called machine learning, allowing them to recognize patterns and be used in numerous applications. Therefore, the objective was to develop Artificial Neural Networks capable of identifying aqueous suspensions with Guar and Xanthan gums (widely used in the food industry) during the crossflow microfiltration process. The ANNs were trained in the supervised learning algorithms `trainscg`, `trainlm` and `traingd`, all in the 70/15/15 model, for a range of five to fifteen neurons in the hidden layer, whose datasets were found in the literature, referring to temperature, flow velocity, pressure, transmembrane flow rate, time and membrane pore size. The software used to implement the ANNs was MATLAB and the evaluation criteria consisted of the analysis of the parameters confusion matrix, error histogram, performance and ROC curve. In summary, ten ANNs had satisfactory performances, presenting confusion matrices with accuracies above 98.8%, error histogram graphs being Gaussian centered at 0, decaying performance curves with stopping criterion equal to 6 errors in the validation set and ROC graphs similar to a square with vertices at (0,0), (1,0), (0,1) and (1,1), results considered satisfactory in the literature.

Keywords

Artificial Neural Networks — Crossflow Microfiltration — Guar — Xanthan.

¹ Department of Engineering, Physics and Mathematics, Institute of Chemistry - São Paulo State University - UNESP Rodovia Araraquara Jaú, Km 01 - s/n, Campos Ville, ZIP Code: 14800-903 – Araraquara, Brasil

*Corresponding author: erica.filletti@unesp.br

Received: 03 December 2023 — Revised: 21 February 2024 — Accepted : 01 May 2024 — Published: 01 May 2024

1. Introduction

An Artificial Neural Network (ANN) is a mathematical computer model that simulates the central nervous system of intelligent living beings, such as *Homo sapiens*. ANNs are capable of unravelling physical and mathematical patterns through a technique called Machine Learning, where previously known data is provided to the computer and it will come up with a model that best adjusts to the problem based on trial and error [Filletti, 2007, Filletti and Selegim Jr, 2010].

Artificial Neural Networks currently have numerous applications, such as optimizing industrial processes [Gul et al., 2020], predicting events [Chen et al., 2020], voice commands and personalizing advertisements [Viktoratos and Tsadiras, 2021,

Wang et al., 2022, Mykhailichenko et al., 2022].

On the other side, crossflow filtration is a unitary operation widely used in the industry which aims to separate phases of a solution based on the principle of the pressure difference caused by the perpendicular movement between the permeate (substances passing through the membrane) and the retained (substances kept under the membrane) [Belfort et al., 1994, Chew et al., 2020].

Crossflow filtration can be divided into subclasses according to the pore size of the membrane used for separation, being crossflow microfiltration one of them. The prefix "micro" refers to the set of pores between 0.1 and 0.2 μm in size [Barros, 2018].

This unit operation is widely used in the food industry

(filtering fermented broths, clarifying, separating and removing unwanted substances) [Proni et al., 2020], sewage treatment (removing impurities from water to make it potable) [Hube et al., 2021], cleaning and cosmetics (optimizing solutions) [Kürzl et al., 2022], among others.

The use of Artificial Neural Networks in the field of cross-flow filtration has already shown positive results in the literature, as pointed out by [Proni et al., 2020, Jokić et al., 2020, Sekulić et al., 2017], who developed ANNs capable of estimating the permeate flow of an açai-based drink, helping to separate the microbial biomass cultivated from *Bacillus velezensis* bacteria and predicting the performance of the microfiltration process for heavy metal ions, respectively.

Furthermore, guar and xanthan gums are polysaccharides that come from the endosperm of the seed of the plant *Cyamopsis tetragonoloba* and from the colony of bacteria of the genus *Xanthomonas*, respectively [Borges and Tondo, 2008, Castañeda Ovando et al., 2020]. Both have a range of applications, especially in the food industry, where they are used as thickeners, emulsifiers, and stabilizers in the preparation of various types of aliments, such as tomato and salad sauces, gluten-free foods, ice cream, fruit juices, as well as personal hygiene products.

Based on that, the objective of this work was to develop Artificial Neural Networks capable of identifying aqueous suspensions containing guar and xanthan gums during the crossflow microfiltration process. The purpose is to create an alternative tool that can help both in controlling an important chemical process in the industry and in identifying the proficiency of industrial batches.

2. Experimental

2.1 Datasets

The input data from the datasets consists of values for temperature (T in $^{\circ}\text{C}$), pressure (P in kPa), flow velocity (v in $\text{m}\cdot\text{s}^{-1}$), time (t in min), and membrane pore size. These input variables result in a respective transmembrane flow rate (TFR in $\text{h}^{-1}\cdot\text{m}^{-2}$) [Queiroz, 2004]. Furthermore, each set of data is accompanied by its respective class (output data) according to the type of aqueous suspension it references: (1 0) for guar and (0 1) for xanthan. An example of a data set can be seen in Table 1.

2.2 Artificial Neural Networks

The Artificial Neural Networks were implemented using MATLAB software with the help of the *nstart* tool. The programming code for the ANN was obtained using this tool, where the number of neurons in the hidden layer and the supervised learning algorithm were changed so that various different combinations could be tested to solve the problem.

The ANNs were separated into two large groups according to the size of the membrane pores in the datasets:

- Group 1: Contains the data for the membrane pore size of $0.4\ \mu\text{m}$. The input data matrix is 5×310 and the

Table 1. Some input datasets with their respective class 2/8

Temp.	Pres.	Flow Velocity	Time	Memb. Pore size	Transmemb. Flow rate	Class	
($^{\circ}\text{C}$)	(kPa)	(m/s)	(min)	(μm)	(1/h.m)		
25	300	3.7	15	0.2	6.63	1	0
25	300	3.7	25	0.2	5.43	1	0
25	300	3.7	15	0.2	35.22	0	1
25	300	3.7	21	0.2	24.56	0	1
40	300	3.7	5	0.2	96.14	1	0
40	300	3.7	33	0.2	16.72	0	1
25	400	2.6	10	0.4	12.18	1	0
25	400	2.6	30	0.4	15.47	0	1
25	500	2.6	20	0.4	4.41	1	0
25	500	2.6	25	0.4	20.74	0	1
40	300	3.7	10	0.3	8.44	1	0
40	300	3.7	15	0.3	4.19	1	0
40	300	3.7	10	0.3	34.61	0	1
40	300	3.7	15	0.3	24.95	0	1
25	400	4.7	20	0.4	5.93	1	0
25	400	4.7	30	0.4	3.80	1	0
25	500	4.7	40	0.4	13.20	0	1
25	400	5.7	5	0.4	106.10	0	1
25	400	5.7	20	0.4	28.57	0	1
25	500	5.7	30	0.4	4.56	1	0

output matrix is 2×310 .

- Group 2: Contains data relating to membrane pore sizes of 0.2 and $0.3\ \mu\text{m}$. The input data matrix is 6×85 and the output matrix is 2×85 .

In group 2, therefore, the pore size entered as a variable for the ANN processing, while in group 1, the value of $0.4\ \mu\text{m}$ was kept fixed. This choice was made since the type of membrane found in the literature in these two cases is different: in group 1, the membrane is of the single-channel type, as for in group 2, multichannel [Queiroz, 2004]. This difference directly interferes on the others output data, affecting the accuracy of the ANN in a hypothetical junction.

2.3 Training Algorithms

As for the supervised training algorithm used to create the ANN, *trainscg*, *traingd* and *trainlm* were tested, each for a range of 5 to 15 neurons in the hidden layer and with a 70/15/15 split of the database (70% training, 15% validation and 15% testing). The *traingd* algorithm did not obtain any satisfactory results, so its use was discarded for this article.

The Scaled Conjugate Gradient Backpropagation (SCG) algorithm is a supervised learning algorithm that adjusts the weights of ANN from the steepest descent of the error function E , which depends on all weights of ANN, using a direction that produces convergence generally faster than the steepest descent direction, maintaining the minimum values achieved in all previous stages. This direction is called conjugate direction [Barros, 2018]. However, this does not necessarily produce the fastest convergence.

In this algorithm, the error function E can be approximated by the expansion of the second-order Taylor series, as shown in Equation 1 [Møller, 1993]:

$$E_{qw}(y) = E(w) + E'(w)^T y + \frac{1}{2} y^T E''(w) \quad \text{Eq.1}$$

where w represents the ANN weight vector and y represents a neighborhood of w . Besides that, to determine the minimum of $E_{qw}(y)$, it is necessary to find the critical points of the function, given by Equation 2:

$$E'_{qw}(y) = 0 \quad \text{Eq.2}$$

The biggest advantage of using SCG is the speed at which the training is completed due to the step size scaling mechanism, avoiding long line searches per learning iteration [Møller, 1993].

The Levenberg-Marquardt algorithm is a combination of two other error backpropagation methods: the Descending Gradient method and the Gauss-Newton method. It is often used to solve non-linear least squares problems [Custódio et al., 2019].

As with the Descending Gradient method, the Levenberg-Marquardt algorithm is iterative; however, it has the advantage of being able to choose the best result among the methods that make it up. According to [Custódio et al., 2019], the Levenberg-Marquardt algorithm tends to be very fast for training Artificial Neural Networks but requires a large amount of memory.

Let $d_i(x)$ be the desired response from neuron i and y_i be the obtained response from the ANN. Then, the error e_i is given by Equation 3:

$$e_i = y_i - d_i \quad \text{Eq.3}$$

where i in C , and C is the set of all neurons in the ANN. Equation 4 shows how the Gauss-Newton method works [Próni et al., 2020]:

$$\Delta x_k = [J^T J]^{-1} J^T e \quad \text{Eq.4}$$

where the vector $x = (x_1, \dots, x_n)$ represents the weights of the ANN, $e = (e_1, \dots, e_n)$, with e_i given by Equation 3, and J is the Jacobian matrix given by the derivative of the error with respect to each synaptic weight x_i . The Levenberg-Marquardt modification is shown in Equation 5:

$$\Delta x_k = [J^T J + \mu I]^{-1} J^T e \quad \text{Eq.5}$$

where I is the identity matrix and $\mu > 0$ is called the Levenberg-Marquardt parameter.

The effect of the additional matrix μI is to add μ to each eigenvalue of $J^T J$. Since the matrix $J^T J$ is positive semi-definite (therefore, the minimum possible eigenvalue is zero), any small but numerically significant positive value of μ will be sufficient to restore the augmented matrix and produce a

downward search direction [Próni et al., 2020].

2.4 Evaluation criteria

The evaluation criteria consisted on the following parameters considered to be by the literature [Filletti, 2007]:

- Confusion matrix accuracy above 98
- Histogram of the error: a graphical model of the pertinence by the error being a Gaussian curve centered on zero;
- Performance: decaying cross entropy curves per epoch with stopping criterion equal to 6 errors in the validation set (empirically verified as the best stopping criterion);
- ROC graph: graph of percentage of hits per percentage of errors being close to a square with vertices at (0,0), (0,1), (1,1) and (1,0).

The ANNs that fit these criteria were chosen for discussion in this article and will be presented in the next section.

3. Results

3.1 Group 1 *Trainscg*

The Artificial Neural Networks that performed satisfactorily with the *trainscg* training algorithm in group 1 (0.4 μm) were the ANNs whose hidden layer contained 6, 7, 8 and 12 neurons in the hidden layer. All of them achieved accuracy above 99% and the errors were mostly concentrated in the training datasets. It should also be noted that the 12-neuron ANN obtained 100% accuracy in all datasets, as shown in confusion matrix (Figure 1a), making it the ANN whose training algorithm is the *trainscg* of group 1 with the highest accuracy. Furthermore, Figure 1b shows the ROC graphic for the same ANN. Since the ANN was 100% in its predictions, the shape of the ROC graphic was perfectly a square with vertices at coordinates (0,0), (0,1), (1,1) and (1,0), which shows that the ANN did not present any false positives.

As for the histogram of the error of each of these ANNs, they all have the same format, a graph of error per frequency characterized by being a Gaussian curve centered on 0. The histogram of the *trainscg* ANN of group 1 with 12 neurons in the hidden layer shows the smallest errors between the ANN responses and the real values of guar and xanthan (Figure 2).

In addition, it should be noted that almost all of the Artificial Neural Networks tested in group 1, whose training algorithm was *trainscg*, obtained accuracy above 95%, with the exception of the 5-neuron ANN, whose accuracy was 90%.

3.2 Group 1 *trainlm*

The Artificial Neural Networks (ANNs) that performed satisfactorily with the *trainlm* training algorithm in group 1 (0.4 μL) were those with 5, 6, 9, and 10 neurons in the hidden layer, all with 100% accuracy in the confusion matrix, with Figure 3a showing the 10-neuron ANN. In addition, due to the

Crossflow Microfiltration of Aqueous Suspensions with Guar and Xanthan Gums: Identification of Solutions Using Artificial Neural Networks



Figure 1. Group 1 *trainscg* ANN with 12 neurons in the hidden layer confusion matrix (a), and ROC graphic (b).

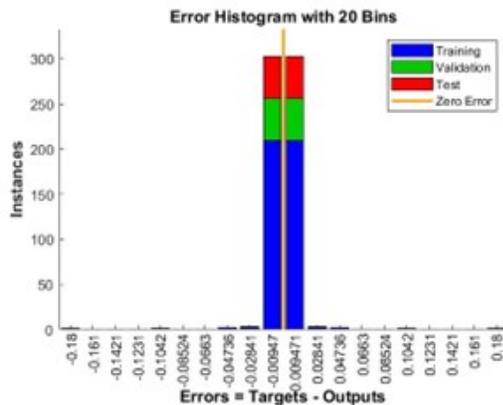


Figure 2. Group 1 *trainscg* ANN with 12 neurons in the hidden layer error histogram.

100% accuracy in the ANN predictions, the ROC graphic was exactly the same of Group 1 *trainscg* ANN with 12 neurons in the hidden layer (Figure 2b), for the same reason.

As with the group 1 *trainscg*, all the histograms have the same shape, a graph of error per instance characterized by being a Gaussian curve centred on 0. The histogram of the error of the 10-neuron group 1 *trainlm* ANNs shows the smallest errors between the ANN responses and the actual guar and xanthan values (Figure 3).



Figure 3. Group 1 *trainlm* ANN with 10 neurons in the hidden layer confusion matrix (a), and error histogram (b).

It should be noted that most of the errors of the 10-neuron ANN were in the order of 10^{-5} , characterizing it as the Artificial Neural Network with the best possible performance of all the groups. The performance graph (mean square error versus epoch) illustrates this in Figure 4:

3.3 Group 2 *trainscg*

In group 2, the *trainscg* training algorithm did not show satisfactory results. The best performance ANN was the one with 11 neurons in the hidden layer.

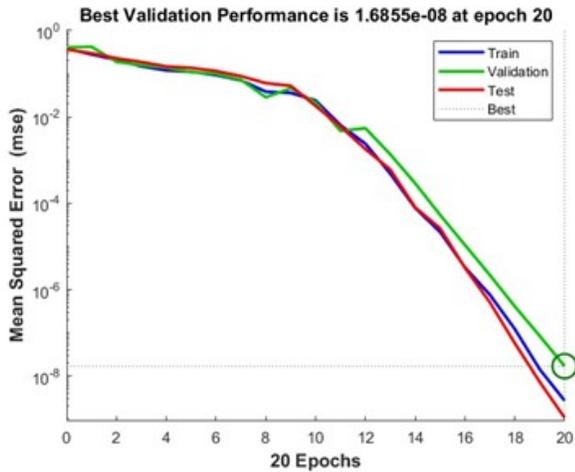


Figure 4. Group 1 *trainlm* ANN with 10 neurons in the hidden layer performance graphic.

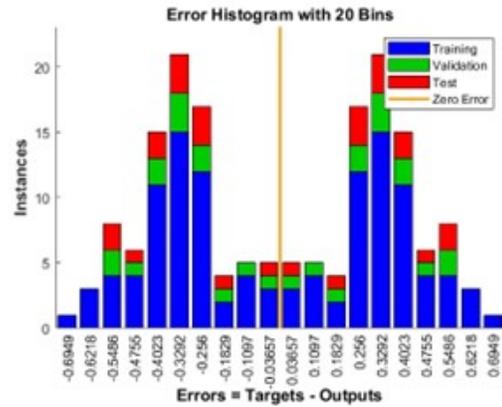
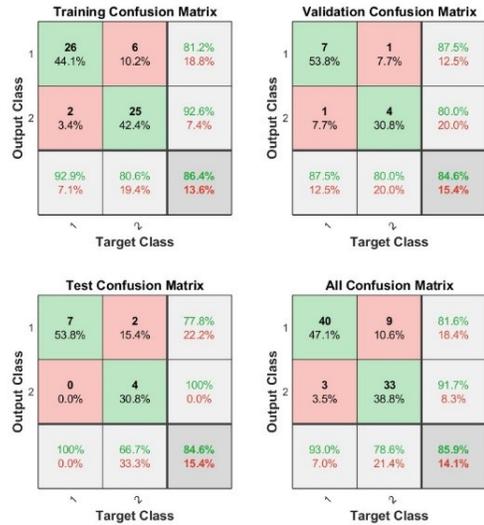


Figure 5. Group 2 *trainscg* ANN with 11 neurons in the hidden layer confusion matrix (a), and error histogram (b).

The group 2 *trainscg* ANN with 11 neurons in the hidden layer obtained 85.9% accuracy in the confusion matrix (Figure 5a) and generated an irregular error histogram graph (Figure 5b), whose errors are concentrated at 0.30, which means that the model generated by the training solved most of the results with a difference of 30% to the original datasets.

It can be seen in Figure 5a, in the Total Confusion Matrix, that the biggest error in the model generated was in the fourth block, id est, the ANN predicted xanthan (2), but the result was guar (1). Table 2 shows all the output data from the group 2 ANN whose training algorithm was *trainscg*, with (1 0) guar and (0 1) xanthan, whose ANN errors are highlighted in red.

Moreover, Figure 6 shows the ROC graphic for the Group 2 *trainscg* ANN with 11 neurons in the hidden layer. It can be seen that the graphic is different to the others, due to the 85% accuracy of this ANN. The graphic shows that, for the test set, the ANN did not present any false positive, but in the training set and validation set, the ANN committed some errors in its predictions.

It should also be noted that the average accuracy of the group 2 *trainscg* ANNs was 75%. This fact can be explained by one main reason: a huge difference between the size of the data set of group 1 and group 2 - while the first has 310 datasets, the second has only 85; - due to the poor result obtained in group 2, this difference in size can be seen as a "barrier" in terms of the number of the datasets for this problem in question, as this directly interferes with the accuracy of the ANN classification. (the more data available for the ANN to carry out its training, the better its performance).

3.4 Group 2 *trainlm*

The Artificial Neural Networks that performed satisfactorily with the *trainlm* training algorithm in group 2 (0.2 and 0.3 μm) were the ANNs whose hidden layer contained 7 and 9 neurons. The 7-neuron ANN in this subgroup obtained 100% accuracy

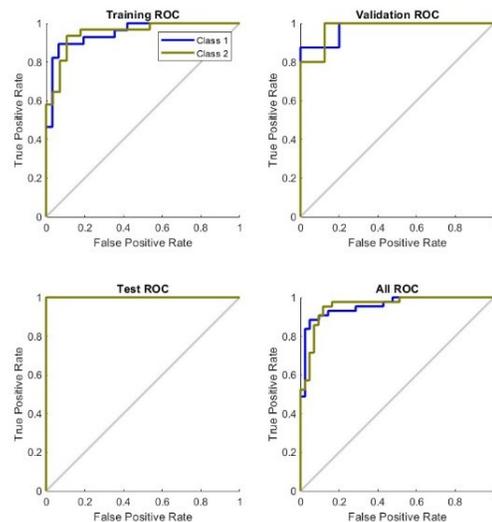


Figure 6. Group 2 *trainscg* ANN with 11 neurons in the hidden layer ROC graphic.

Crossflow Microfiltration of Aqueous Suspensions with Guar and Xanthan Gums: Identification of Solutions Using Artificial Neural Networks

(Figure 7a), while the 9-neuron ANN obtained 98.8%.

Figure 7b shows the error histogram of the 7-neuron ANN. The ANN reached estimated errors of 35%, despite its high precision. A plausible explanation for this high error in the histogram, which is much higher than the errors of group 1, whose training algorithm was also *trainlm*, is the discrepant difference in the number of datasets, as also happened with the *trainscg* algorithm.



Figure 7. Group 2 *trainlm* ANN with 7 neurons in the hidden layer confusion matrix (a), and error histogram (b).

Furthermore, the average for this subgroup was around 90% accuracy in the confusion matrix, which possibly indicates that if there were more datasets available in the literature, the precision would be higher and, therefore, the algorithm proved to be functional in this case as well.

Table 2. Group 2 *trainscg* ANN with 11 neurons in the 6/8 hidden layer output data.

Sample	ANN Response		Class Obtained by ANN		Expected Class	
1	0.7757	0.2243	1	0	0	0
2	0.5875	0.4125	1	0	0	0
3	0.5279	0.4721	1	0	0	0
4	0.5503	0.4497	1	0	0	0
5	0.5910	0.4090	1	0	0	0
6	0.6169	0.3831	1	0	0	0
7	0.6406	0.3594	1	0	0	0
8	0.8727	0.1273	1	0	0	0
9	0.8584	0.1416	1	0	0	0
10	0.7532	0.2468	1	0	0	0
11	0.6145	0.3855	1	0	0	0
12	0.3962	0.6038	0	1	0	0
13	0.4583	0.5417	0	1	0	0
14	0.6239	0.3761	1	0	0	0
15	0.7467	0.2533	1	0	0	0
16	0.8508	0.1492	1	0	0	0
17	0.9072	0.0928	1	0	0	0
18	0.9409	0.0591	1	0	0	0
19	0.7315	0.2685	1	0	0	1
20	0.5455	0.4545	1	0	0	1
21	0.4999	0.5001	0	1	0	1
22	0.5201	0.4799	1	0	0	1
23	0.5488	0.4512	1	0	0	1
24	0.5789	0.4211	1	0	0	1
25	0.0700	0.9300	0	1	0	1
26	0.3508	0.6492	0	1	0	1
27	0.4046	0.5954	0	1	0	1
28	0.3586	0.6414	0	1	0	1
29	0.3482	0.6518	0	1	0	1
30	0.2216	0.7784	0	1	0	1
31	0.2384	0.7616	0	1	0	1
32	0.2712	0.7288	0	1	0	1
33	0.3321	0.6679	0	1	0	1
34	0.4187	0.5813	0	1	0	1
35	0.5130	0.4870	1	0	0	1
36	0.6115	0.3885	1	0	0	1
37	0.6853	0.3147	1	0	0	1
38	0.7583	0.2417	1	0	0	1
39	0.7494	0.2506	1	0	0	1
40	0.7360	0.2640	1	0	0	1
41	0.7027	0.2973	1	0	0	1
42	0.6192	0.3808	1	0	0	1
43	0.6237	0.3763	1	0	0	1

Table 3. Table 2 - Continuation: Group 2 *trainscg* ANN with 11 neurons in the hidden layer output data.

Sample	ANN Response		Class Obtained by ANN		Expected Class	
44	0.6453	0.3547	1	0	0	1
45	0.6798	0.3202	1	0	0	1
46	0.7219	0.2781	1	0	0	1
47	0.7648	0.2352	1	0	0	1
48	0.8057	0.1943	1	0	0	1
49	0.8390	0.1610	1	0	0	1
50	0.3689	0.6311	0	1	0	1
51	0.6724	0.3276	1	0	0	1
52	0.7452	0.2548	1	0	0	1
53	0.7294	0.2706	1	0	0	1
54	0.7032	0.2968	1	0	0	1
55	0.7150	0.2850	1	0	0	1
56	0.7558	0.2442	1	0	0	1
57	0.8136	0.1864	1	0	0	1
58	0.8661	0.1339	1	0	0	1
59	0.9059	0.0941	1	0	0	1
60	0.9335	0.0665	1	0	0	1
61	0.9519	0.0481	1	0	0	1
62	0.2758	0.7242	0	1	0	1
63	0.5838	0.4162	1	0	0	1
64	0.5138	0.4862	1	0	0	1
65	0.4452	0.5548	0	1	0	1
66	0.3955	0.6045	0	1	0	1
67	0.3688	0.6312	0	1	0	1
68	0.3592	0.6408	0	1	0	1
69	0.3617	0.6383	0	1	0	1
70	0.3770	0.6230	0	1	0	1
71	0.4000	0.6000	0	1	0	1
72	0.4330	0.5670	0	1	0	1
73	0.4738	0.5262	0	1	0	1
74	0.0002	0.9998	0	1	0	1
75	0.3002	0.6998	0	1	0	1
76	0.3416	0.6584	0	1	0	1
77	0.3373	0.6627	0	1	0	1
78	0.3202	0.6798	0	1	0	1
79	0.3020	0.6980	0	1	0	1
80	0.2931	0.7069	0	1	0	1
81	0.2908	0.7092	0	1	0	1
82	0.3092	0.6908	0	1	0	1
83	0.3449	0.6551	0	1	0	1
84	0.3994	0.6006	0	1	0	1
85	0.4699	0.5301	0	1	0	1

4. Conclusions

Artificial Neural Networks are precise models for solving a wide range of problems, whether physical, chemical, mathematical or even philosophical, from simple ones such as predicting permeates to very complex problems such as voice detection, autonomous cars and so on.

This paper explored the concepts behind this fascinating area of computing and used the tool of Artificial Neural Networks to solve a relatively simple problem, requiring ANNs with few neurons in the hidden layer only, but very important for optimizing and controlling a unit operation that is widely used in industries, especially in the food industry, which is extremely necessary in today's system.

The ANNs created during this work proved that there is a pattern behind the problem of classifying aqueous suspensions subjected to crossflow microfiltration, which can be unfolded through the use of this tool. As a result, six of the ten Artificial Neural Networks shaped a model that was able to get 100% of the provided datasets, some with estimation errors of less than 10^{-3} .

In addition, *trainlm* proved to be the most suitable training algorithm for this case, as five of the six ANNs developed achieved maximum efficiency in terms of the confusion matrix. With regard to *trainscg*, the algorithm was efficient in solving the problem, but only when the amount of data is relatively large. Finally, the *traingd* algorithm did not fit well.

Therefore, considering the results presented in this report, Artificial Neural Networks proved to be a suitable tool for solving the proposed problem, the identification of aqueous suspensions during the crossflow microfiltration process with guar and xanthan gums, because, as well as being low cost and easy to apply, the ANNs were accurate in their results.

References

- [Barros, 2018] Barros, V. (2018). Avaliação do desempenho de algoritmos de retropropagação com redes neurais artificiais para a resolução de problemas não-lineares. Master's thesis, Federal Technological University of Paraná, Postgraduate in Science Computing.
- [Belfort et al., 1994] Belfort, G., Davis, R., and Zydney, A. (1994). The behavior of suspensions and macromolecular solutions in crossflow microfiltration. *Journal of Membrane Science*, 96(1-2):1-58.
- [Borges and Tondo, 2008] Borges, C. D. and Tondo, C. V. (2008). Xanthan gum: characteristics and operational conditions of production. *Semina: Ciências Biológicas e Da Saúde*, 29(2):171-188.
- [Castañeda Ovando et al., 2020] Castañeda Ovando, A., González-Aguilar, L. A., Granados-Delgadillo, M. A., and Chávez-Gómez, U. J. (2020). Goma guar: un aliado en la industria alimentaria. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías Del ICBI*, 7(14):107-111.
- [Chen et al., 2020] Chen, T., Kapron, N., and Chen, J. C.-Y. (2020). Using evolving ann-based algorithm models for accurate meteorological forecasting applications in vietnam. *Mathematical Problems in Engineering*, pages 1-8.
- [Chew et al., 2020] Chew, J. W., Kilduff, J., and Belfort, G. (2020). The behavior of suspensions and macromolecular

Crossflow Microfiltration of Aqueous Suspensions with Guar and Xanthan Gums: Identification of Solutions Using Artificial Neural Networks

solutions in crossflow microfiltration: An update. *Journal of Membrane Science*, 601:117865.

- [Custódio et al., 2019] Custódio, C. A., Filletti, É. R., and França, V. v. (2019). Artificial neural networks for density-functional optimizations in fermionic systems. *Scientific Reports*, 9(1).
- [Filletti, 2007] Filletti, É. R. (2007). *Desenvolvimento de modelos neurais para o processamento de sinais acústicos visando a medição de propriedades topológicas em escoamentos multifásicos*. PhD thesis.
- [Filletti and Seleglim Jr, 2010] Filletti, É. R. and Seleglim Jr, P. (2010). Nonintrusive measurement of interfacial area and volumetric fraction in dispersed two-phase flows using a neural network to process acoustic signals — a numerical investigation.
- [Gul et al., 2020] Gul, M., Kalam, M. A., Mujtaba, M. A., Alam, S., Bashir, M. N., Javed, I., Aziz, U., Farid, M. R., Hassan, M. T., and Iqbal, S. (2020). Multi-objective-optimization of process parameters of industrial-gas-turbine fueled with natural gas by using grey-taguchi and ann methods for better performance. *Energy Reports*, 6:2394–2402.
- [Hube et al., 2021] Hube, S., J., W., Sim, L. N., Ólafsdóttir, D., Chong, T. H., and Wu, B. (2021). Fouling and mitigation mechanisms during direct microfiltration and ultrafiltration of primary wastewater. *Journal of Water Process Engineering*, 44:102331.
- [Jokić et al., 2020] Jokić, A., Pajčin, I., Grahovac, J., Lukić, N., Ikonić, B., Nikolić, N., and Vlajkov, V. (2020). Dynamic modeling using artificial neural network of bacillus velezensis broth cross-flow microfiltration enhanced by air-sparging and turbulence promoter. *Membranes*, 10(12):1–14.
- [Kürzl et al., 2022] Kürzl, C., Tran, T., and Kulozik, U. (2022). Application of a pulsed crossflow to improve chemical cleaning efficiency in hollow fibre membranes following skim milk microfiltration. *Separation and Purification Technology*, 302:122123.
- [Møller, 1993] Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4):525–533.
- [Mykhailichenko et al., 2022] Mykhailichenko, I., Ivashchenko, H., Barkovska, O., and Liashenko, O. (2022). Application of deep neural network for real-time voice command recognition. In *2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek)*, pages 1–4.
- [Proni et al., 2020] Proni, C., Haneda, R. N., and Filletti, E. R. (2020). Desenvolvimento de redes neurais artificiais para análise do fluxo de permeado de uma bebida à base de açaí no processo de microfiltração tangencial. *C.Q.D.- Revista Eletrônica Paulista de Matemática*, 17:189–205.
- [Queiroz, 2004] Queiroz, V. M. S. (2004). *Estudo experimental do escoamento e da concentração de mistura no processo de filtração tangencial de suspensões macromoleculares*. PhD thesis.
- [Sekulić et al., 2017] Sekulić, Z., Antanasijević, D., Stevanović, S., and Trivunac, K. (2017). Application of artificial neural networks for estimating cd, zn, pb removal efficiency from wastewater using complexation-microfiltration process. *International Journal of Environmental Science and Technology*, 14(7):1383–1396.
- [Viktoratos and Tsadiras, 2021] Viktoratos, I. and Tsadiras, A. (2021). Personalized advertising computational techniques: A systematic literature review, findings, and a design framework. *Information*, 12(11):480.
- [Wang et al., 2022] Wang, G., Tan, G. W.-H., Yuan, Y., Ooi, K.-B., and Dwivedi, Y. K. (2022). Revisiting tam2 in behavioral targeting advertising: A deep learning-based dual-stage sem-ann analysis. *Technological Forecasting and Social Change*, 175:121–345.