Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Modelling of a Propylene Glycol Production Process With Artificial Neural Networks: Optimization of the Architecture

Emilio Alba-Robles[a], Oscar Daniel Lara-Montaño[b], Fernando Israel Gómez-Castro[a*], Jahaziel Alberto Sánchez-Gómez[a], Manuel Toledano-Ayala[b]

[a] Universidad de Guanajuato, Departamento de Ingeniería Química, Guanajuato, Guanajuato, México
[b] Universidad Autónoma de Querétaro, Facultad de Ingeniería, Querétaro, Querétaro, México
* Corresponding Author: fgomez@ugto.mx.

ABSTRACT

Chemical process models often involve high non-linearity due to thermodynamic and kinetic relationships, with non-convex bilinear terms adding complexity to process optimization. Recently, data-driven models, particularly artificial neural networks (ANNs), have gained traction for representing chemical processing units. The predictive accuracy of ANNs depends on data quality, variable interactions, and network architecture, the latter being an optimization challenge itself. This study proposes and evaluates two strategies to optimize ANN architecture for modeling a propylene glycol production process from glycerol. The process includes a reactor and two distillation columns, with training data generated through simulation in Aspen Plus by varying design and operating variables. Two approaches are compared: random ANN structure generation and architecture optimization using the ant colony algorithm, a method suitable for discrete problems. Decision variables include the number of hidden layers and neurons per layer, with the objective of minimizing mean squared error. Both methods yield ANN predictions with high accuracy (R^2 > 99.9%), closely matching simulation data. However, the ant colony algorithm achieves a better fit despite slower convergence, demonstrating its effectiveness in fine-tuning ANN architecture for chemical process modeling.

Keywords: Artificial Intelligence, Artificial Neural Networks, Glycerol, Stochastic Optimization, Network Architecture

INTRODUCTION

Rigorous models for process equipment are built upon well-established physical and chemical principles, such as design equations, mass and energy balances, thermodynamic relationships, and, in some cases, cost equations. These models enable the precise representation of system behavior across various operating conditions, offering a robust foundation for process analysis and optimization.

Despite their advantages—such as high accuracy, clear interpretability, and alignment with physical laws, as highlighted in numerous studies—rigorous models have significant limitations. Researchers have pointed out challenges including the high cost of constructing these models, particularly for large and complex systems. Additionally, they require a deep understanding of the system and often involve extensive fine-tuning to achieve accurate predictions [1]. Moreover, the computational demands of rigorous models can be prohibitive, especially when simulating intricate or highly dynamic processes.

These challenges have accelerated interest in data-driven models, which address many of these limitations. Data-driven science, often referred to as the fourth paradigm [2], is transforming how we approach research and problem-solving. Montáns et al. [3] highlight that, unlike rigorous models, data-driven approaches rely on vast amounts of empirical data to uncover patterns, trends, and insights, bypassing the need for exhaustive system-

specific knowledge. These methods enable faster model development and adaptation, particularly for complex, nonlinear systems.

The development of data-driven models is an area of significant interest in Process Systems Engineering. An example of this can be seen in the work by Kwon et al. [4], where data-driven models are applied to the analysis of processing units for controlling distillation columns. Among the various types of data-driven models, machine learning models stand out due to their ability to identify complex patterns without the need for explicit formulation.

Cavalcanti et al. [5] conducted an extensive review compiling a range of successful studies on the application of Artificial Intelligence (AI) in Process Engineering. This comprehensive study covers various fields, particularly showing how ANNs can replace complex phenomenological models, offering the advantage of reduced computational effort.

For an ANN model to perform satisfactorily, both the quantity and quality of the data used for training must be appropriate. Regarding quantity, the performance of ANN models is enhanced as the amount of data available for training increases [6]. Of course, the number of data points required will depend on the complexity of the system being modeled by the neural network. Likewise, the quality of the information used for training the network is crucial. The use of such strategies for process design applications presents a challenge, as the units necessary for generating data to train the network are not available. One strategy that can be employed under such circumstances is the use of rigorous models for data generation [7].

Recent research has focused on optimizing artificial neural network (ANN) architectures to enhance their predictive performance and computational efficiency. Evolutionary algorithms and metaheuristic approaches have been widely explored to automate the design of ANN structures, allowing for improved adaptability to complex systems [8]. In process modeling applications, strategies incorporating statistical criteria for model selection, such as the Akaike Information Criterion (AIC), have been employed to refine ANN architectures and enhance their predictive capabilities [9].

In this work, ANNs are proposed for modeling chemical processing units, employing two different strategies for architecture optimization. The first strategy relies on randomly generated configurations, while the second utilizes the ant colony optimization algorithm [10]. In both approaches, key performance parameters are recorded to enable a subsequent comparison of their effectiveness. The study begins with an overview of the case study, followed by a detailed description of the methodology used to generate the data and evaluate the proposed optimization strategies. Finally, the results are analyzed and discussed, highlighting the advantages of each approach.

CASE STUDY

Sánchez-Gómez et al. [11] proposed various production schemes for propylene glycol, including conventional and reactive distillation-based configurations. In this study, the focus is on the conventional processing scheme, which is presented in Figure 1.

The feed to the process consists of a crude glycerol stream, represented as a 1:1 mass ratio mixture of glycerol and water, with a total flow rate of 100 kmol/h. This stream enters a reactor where glycerol reacts with hydrogen to produce propylene glycol. However, a secondary reaction also occurs, forming ethylene glycol as a byproduct.

The reactor effluent is separated using a direct distillation sequence, where propylene glycol is obtained as the distillate product from the second column, achieving a molar purity of at least 99%. Further details about the process are provided in the work by Sánchez-Gómez et al. [11]. The initial conditions for the process are given in Table 1.

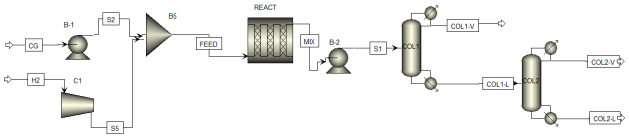

Figure 1. Propylene glycol production process from glycerol and hydrogen.

METHODOLOGY

The analysis has been divided in two stages: (i) data generation, and (ii) development of artificial neural networks. These stages are described below.

Information Generation

To gather sufficient data for neural network modeling, the analysis begins with an initial simulation in Aspen Plus of the conventional propylene glycol production process (Figure 1), operating under the initial conditions outlined in Table 1. The focus is on modeling the reactor and the two distillation columns using artificial neural networks.

Python programming enables interaction with Aspen Plus simulations [12]. A customized code has been developed to automatically access and modify variables within the simulation, identified as input variables in Table 1. The simulation was executed under various conditions, generating response variable values for 500 random designs.

To define feasible ranges for input variables,

Table 1: Manipulated and response variables for sensitivity analysis.

Reactor			
Manipulated	**Initial Values**	**Range**	**Response**
Diameter	0.1 m	1 to 6 m	Heat Load
Length	10 m	3 to 10 m	Composition
Operation temperature	225 °C	200 to 250 °C	
Columns 1 and 2			
Manipulated	**Initial Values**	**Range**	**Response**
Number of Stages (NS)	20 (COL1) 40 (COL2)	10 to 35 25 to 50	Diameter
Feed Stage	5 (COL1) 20 (COL2)	From 2 to NS	Composition at bottoms and dome
Reboiler Heat Load	320 kcal/s (COL1) 183 kcal/s (COL2)	157 to 788 kcal/s 91 to 457 kcal/s	Condenser heat load
Reflux ratio	0.5 (COL1) 2.6 (COL2)	0.1 to 10	Temperature at S1, S2, SN and SN-1

evaluation intervals were established based on the initial process conditions (Table 1), ensuring the design and operational parameters align with practical standards.

Non-converging or thermodynamically inconsistent simulations were filtered out, leaving a dataset of 351 valid simulations for subsequent economic analysis. For each case, the total annualized cost (TAC) of the process was calculated using Guthrie's method, annualizing equipment costs over 10 years [13].

At the end of this process, the dataset is ready for training and testing the neural networks. An important aspect of the data used to train an artificial neural network is its quality. In this case, the data to be used meets this characteristic, as they come from rigorous simulations conducted in a commercial simulator, Aspen Plus. This simulator is widely recognized in the industry for its ability to accurately and reliably model chemical processes. By using such a simulator, it is ensured that the generated data are representative of actual operational conditions, if adequate datasets and sub-models are employed.

Development of Artificial Neural Networks (ANNs)

To determine whether the generated data was sufficient, an ANN was developed and trained using Python and the TensorFlow library. For this initial evaluation, a network with two hidden layers, each containing 32 neurons, was employed. The Adam optimization algorithm was used to adjust the weights of the connections between neurons, with the ReLU (Rectified Linear Unit) activation function. In all cases, the output variable was the total annualized cost (TAC) of the equipment being modeled. The dataset was split into training and validation data, using percentages of 80% and 20%, respectively. The results of this preliminary assessment were satisfactory, indicating that the amount of data generated was adequate.

The next step involves identifying the network configuration that minimizes the mean squared error (MSE) of the predictions across the entire dataset. The MSE was quantified using the following objective function:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(real_i - pred_i)_i^2 \quad (1)$$

Two strategies were evaluated to solve this optimization problem:

(i) Random generation of network architectures.

(ii) A metaheuristic optimization algorithm based on the Ant Colony Optimization (ACO) approach.

Random Generation

The random generation strategy is widely used as an effective approach for identifying the optimal neural network architecture [14]. This strategy begins by defining the limits for the number of layers and the number of neurons per layer. A code was implemented to randomly generate network architectures within these limits, train them, and evaluate their performance using MSE as the primary metric. Among all evaluated architectures, the one with the lowest MSE was selected. A total of 150 network architectures were tested, with additional performance metrics recorded, such as the coefficient of determination (R^2), prediction error percentage, and computation time.

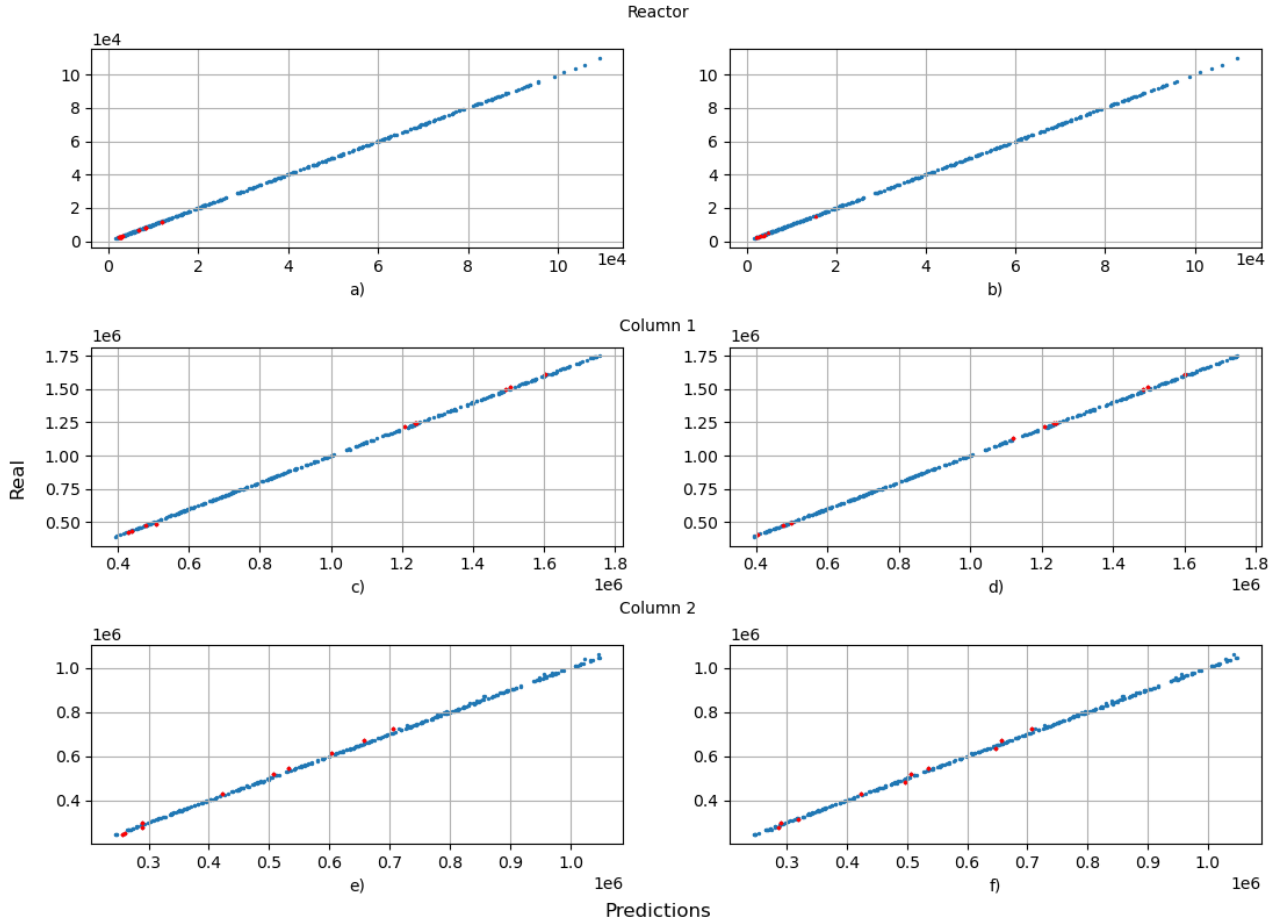

Figure 2. Rigorous simulation data (y-axis) vs predictions (x-axis). (a) Reactor with ACO, (b) Reactor with RS, (c) Column 1 with ACO, (d) Column 2 with ACO, (e) Column 2 wirh RS.

Metaheuristic Optimization Algorithm

Existing ACO algorithm implementations, such as the ACOR algorithm in the *Mealpy* library [15], has been considered. This version of ACOR provides solutions for both integer and continuous variables. However, due to specific needs in this study, the original Ant Colony Optimization algorithm was implemented in Python to address integer-based decisions. The algorithm parameters, along with the ranges for decision variables, are summarized in Tables 2 and 3. These last values were used consistently across both solution strategies.

The number of ants and iterations was set to ensure a total of 150 networks were trained, enabling a fair comparison between the random generation strategy and the metaheuristic approach. Just as in the random generation strategy, metrics such as MSE, R^2, prediction error percentage, and computational time were recorded and used to evaluate and compare the performance of both approaches.

RESULTS

As a result of the analysis conducted six neural network models were obtained, two for each process unit under study, corresponding to each architecture optimization strategy.

Figure 2 shows a comparison of the predictions obtained from both strategies. The data obtained through the Ant Colony Optimization algorithm is presented in the left column, while the corresponding results from Random Search are shown on the right. For each case, the 10 predictions with the highest error percentages can be observed as red points.

Table 2: Parameters for the Ant Colony Optimizer.

Strategy	Iterations	Population	n_best	decay	alpha	beta
ACO	25	6	5	0.95	1	1
RG	150	-	-	-	-	-

Table 3: Ranges for decision variables.

Variable	Range
Number of layers	[1, 4]
Number of neurons	[1, 100]

Table 4 summarizes the performance measures recorded for each model, as well as the architecture that minimized the MSE. In the last column (Architecture), the first number represents the number of hidden layers, while the information in brackets represents the number of neurons per layer. Among the generated models, only the reactor model obtained through the random search strategy exhibited predictions with an error percentage greater than 5%. In contrast, the remaining models predicted across the entire dataset with an error below 5%. In terms of the values of R^2, values higher than 99.9% are obtained for all the cases. However, it is observed that the Ant Colony Optimizer allowed faster convergence, particularly for the model of the second distillation column. It is important to mention that, for both strategies, a high number of neurons per layer are obtained. This is a relevant aspect to be considered for future research, since it is desired to obtain models with a good fitting, but with simpler architectures.

CONCLUSIONS

In this study, artificial neural networks are used to represent the performance of equipment from a chemical process, particularly the production of propylene glycol from renewable glycerol. Six artificial neural network models were obtained, each achieving accuracy greater than 99%. Several parameters were recorded to analyze their performance and subsequently compare the strategies.

Overall, the random architecture generation strategy yielded slightly better results in terms of MSE and R^2. However, the Ant Colony Optimizer required less time than the random search approach to determine the optimal architecture.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support from Universidad de Guanajuato, Mexico, through project 189/2024. Additionally, CONAHCYT provided financial support through a postdoctoral fellowship granted to Oscar Daniel Lara Montaño (application number 4861420). The resources and academic environment provided by these institutions were fundamental to the successful completion of this research.

REFERENCES

1. Bikmukhametov T, Jäschke J. Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. *Comput Chem Eng* 138: 106834 (2020). https://doi.org/10.1016/j.compchemeng.2020.106834
2. Hey T, Tansley S, Tolle K. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research (2009)
3. Montáns FJ, Chinesta F, Gómez-Bombarelli R, Kutz JN. Data-driven modeling and learning in science and engineering. *C R Mécanique* 347: 845–855 (2019). https://doi.org/10.1016/j.crme.2019.11.009
4. Kwon H, Oh KC, Choi Y, Chung YG, Kim J.

Table 4: Performance measures recorded for each model.

Model	MSE	R^2	%e (top1)	Computational time (s)	Architecture
Reactor AC	17678.7572	0.9999788	3.8471	12270.27	4 [76]
Reactor RS	13854.5483	0.9999834	8.2299	12501.10	4 [89]
Column 1 AC	6670628.1656	0.9999589	4.8744	12767.27	4 [78]
Column 1 RS	6084556.8997	0.9999625	1.3685	15123.15	4 [79]
Column 2 AC	10504337.1583	0.9997871	4.3230	12196.61	4, [95]
Column 2 RS	10131389.0314	0.9997945	3.4354	12200.67	4 [93]

Development and application of machine learning-based prediction model for distillation column. *Int J Intell Syst* 36:1970-1997 (2021). https://doi.org/10.1002/int.22368

5. Cavalcanti FM, Kozonoe CE, Pacheco KA, De Brito Alves RM. Application of Artificial Neural Networks to Chemical and Process Engineering. IntechOpen (2021).
6. Aggarwal CC. Neural Networks and Deep Learning: A Textbook. Springer Nature (2023).
7. Mosqueda-Huerta ZJ, Lara-Montaño OD, Toledano-Ayala M, Gómez-Castro FI. Desarrollo de modelos de red neuronal artificial para el diseño de intercambiadores de calor de tubos y coraza. *Avances en Ingeniería Química* 3:PRO75-PRO80 (2024).
8. Ünal HT, Başçiftçi F. Evolutionary design of neural network architectures: a review of three decades of research. *Artif Intell Rev* 55:1723-1802 (2022). https://doi.org/10.1007/s10462-021-10049-5
9. Hernández-Vargas EA, Sánchez-Ramírez E, De La Fuente AR, Calderón-Alvarado MP, Segovia-Hernández JG. Advancing neural network architectures for time series forecasting: A sustainable approach to intensified biobutanol production. *Chem Eng Process* 194:109603 (2023). https://doi.org/10.1016/j.cep.2023.109603
10. Zhao H, Zhang C, Zheng X, Zhang C, Zhang B. A decomposition-based many-objective ant colony optimization algorithm with adaptive solution construction and selection approaches. *Swarm Evol Comput* 68:100977 (2022). https://doi.org/10.1016/j.swevo.2021.100977
11. Sánchez-Gómez JA, Gómez-Castro FI, Hernández S. Design and intensification of the production process of propylene glycol as a high value-added glycerol derivative. *Comput Aided Chem Eng* 52:1915-1920 (2023). https://doi.org/10.1016/B978-0-443-15274-0.50304-8
12. ten Hagen, R. W. AspenPlus Python Interface (Version 2.1) [Computer software] (2022). https://github.com/YouMayCallMeJesus/AspenPlus-Python-Interface
13. Turton R, Shaeiwitz JA, Bhattacharyya D, Whiting WB. Analysis, Synthesis, and Design of Chemical Processes. Pearson Professional (2018).
14. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media (2019).
15. Van Thieu N, Mirjalili S. MEALPY: An open-source library for latest meta-heuristic algorithms in Python. *J Syst Archit* 139:102871 (2023). https://doi.org/10.1016/j.sysarc.2023.102871