Model-based optimization approaches for pressure-driven membrane systems

Rizki, Zulhaj; Ottens, Marcel

DOI
10.1016/j.seppur.2023.123682

Publication date
2023

Document Version
Final published version

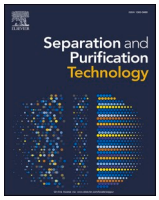
Published in
Separation and Purification Technology

Separation and Purification Technology 315 (2023) 123682

Contents lists available at ScienceDirect

Separation and Purification Technology

journal homepage: www.elsevier.com/locate/seppur



Review

Model-based optimization approaches for pressure-driven membrane systems

Zulhaj Rizki *, Marcel Ottens

Department of Biotechnology, Delft University of Technology, van der Maasweg 9, 2629HZ Delft, The Netherlands

ARTICLE INFO

Keywords:
Membrane
Modeling
Optimization
Neural networks
Mechanistic models

ABSTRACT

Membrane technology is commonly used within food, bio- and pharmaceutical processes. Beside single-stage membranes, multi-stage membrane systems are become more popular to improve separation performance. In this review, we present a unified four-phase model-based optimization framework to optimize these systems, using mechanistic models, empirical models including machine learning models, or a combination of them. We begin by providing a general overview and outlining the steps to construct each phase in the framework. The importance of each stage and critical points to consider are discussed. We then provide detailed information for each phase, including the governing equations from known literature models. Finally, we explore the platform's potential applications and outlook. Despite the great potential of an integrated approach, studies thus far focus either on extensive membrane modeling with brute-force optimization via simple comparison or on meticulous optimization using an oversimplified membrane model. We believe that the integrated framework can bridge the well justified approaches in both filtration modeling and mathematical optimization and help in designing multi-unit processes.

1. Introduction

Membrane separation technology is often used for food, bio- and pharmaceutical processes because of its mild conditions, resulting in limited product quality degradation [1–3], its ability to process a massive volume of streams, and its economic benefits [4]. Moreover, successful implementation of membrane processes has been reported for applications involving macromolecules such as proteins and carbohydrates [5–8], medium-sized molecules such as oligosaccharides [9,10], and small compounds such as sugars and salts [11–15].

As a disadvantage, the achievable purity is an issue for membrane technology. A single-unit membrane cannot produce the same high-purity product as more selective processes such as chromatography [16,17] or thermal based processes like evaporation and crystallization. Yet, economic considerations and the advantage of mild conditions motivate researchers to keep improving the performance of membrane processes. This can be done in three ways: (1) improving the selectivity by modifying the membrane properties during fabrication [18,19], (2) improving the mass transfer by modifying the module design [20,21], and (3) optimizing the process design, for example by utilizing multi-stage membrane systems [7,22–25]. Each approach has been well developed and involves multidisciplinary research, which includes (not limited to) material science, process engineering, mathematics and computer science. In this paper, we focus on the third approach.

Optimizing the design of a membrane process may include the selection of optimum process parameters (e.g. *trans*-membrane pressure, TMP, or process temperature) of a single membrane unit or the use of multiple membrane systems that can be optimized in the later stage as well. A membrane system of this configuration comprises multiple (similar or different types of) membranes such that it gives an enhanced performance compared to a single membrane unit [22–27]. One membrane process may function as a pre-treatment for another membrane process in order to yield a better overall separation compared to a direct feeding. A classic example of such a multiple membrane system is a series configuration where the outlet stream of each membrane is fed to the next membrane to improve the product purity. For purification of permeable compounds, the additional membrane is configured next to the permeate stream [28], while for concentration of rejected molecules, the additional membrane is configured next to the retentate stream [23]. Dilution of this retentate stream before feeding it to the next membrane is often done to wash the permeable compounds. Such a process is well known as diafiltration [29,30].

Membranes in a series configuration (with or without diafiltration) only further process one of the two outlet streams of a membrane. The

* Corresponding author.

https://doi.org/10.1016/j.seppur.2023.123682
Received 7 December 2022; Received in revised form 20 March 2023; Accepted 22 March 2023
Available online 28 March 2023

other stream is usually collected and discharged as a waste stream or fed to another unit to recover valuable compounds. This practice increases the product loss, thus reducing the yield. To overcome this issue, a practice of recycling the other stream and feeding it back to the previous stage has been applied. The stream can be fed directly to the adjacent unit, making it a counter-current cascade configuration (or simply cascade) [26,31,32]. An illustration of different configurations of the membrane system can be seen in Fig. 1.

Optimizing membrane systems can be done from two perspectives: (1) finding the optimum set of process parameters, and (2) selecting the optimum configuration. The optimum process parameters are often determined for an individual unit and later used within the full membrane system. This practice is easier due to the smaller number of parameters involved during the optimization procedure. In some cases, the optima of a single unit are also the optima for the full system. However, it can be found that the global objective of a membrane system differs from the objective of its individual contributing unit. In those cases, a holistic optimization for the complete system is necessary.

Designing an optimum membrane system can be challenging. It requires insight into the behavior of each unit and how a change in each process parameter affects the overall performance of the membrane system. Brute-force optimization can be done by testing various combinations of process parameters (e.g. using factorial design) and comparing the results. The combination that gives the highest performance is then selected as the optimum. Considering other possible configurations of the system further complicates the search for optimum conditions [27]. Testing all possible combinations experimentally can be very laborious and may not guarantee that the real optimum is among the tested combinations.

Modeling can aid the optimization process and save a lot of time and resources. With a proper model, the outcome of membrane processes can be simulated with given process parameters. The optimum can then be selected from the best simulated result [26,27]. Due to the huge number of possible combinations of process parameters and configurations, the search for an optimum set-up can still be challenging and time consuming. Applying an optimization algorithm can then help to find the optimum faster [36–39].

In this paper, we review modeling approaches that have been studied for pressure driven membrane processes and integrate them with optimization algorithms. While we focus on selected applications within food, bioprocesses and pharmaceutical areas, similar principles are expected to apply for other areas. An integrated framework bridging membrane filtration modeling and mathematical optimization is then discussed as a general guideline to approach complex optimization problems in the context of membrane systems. This framework is similar to the general framework used in process system engineering [40] giving special attention to multi-stage pressure-driven membranes.

2. Optimization framework

The optimization of a membrane system using models can be divided into two parts: modeling the membrane itself and programming the optimization problem. Each part has become the subject of intensive research in separate fields. However, an integration of both approaches is underrepresented in the literature.

An integrated modeling approach can be achieved by sequentially combining the membrane and optimization models, with the membrane model serving as input for the optimization algorithm. Prior to model development, it is crucial to define a clear and aligned problem scope to synchronize the objectives of both models. A membrane system is comprised of multiple filtration units, making a single filtration model the building block for the larger membrane system model. This integrated approach can be broken down into four phases, which are discussed in the following sections and summarized in Fig. 2.

2.1. Phase 1: System definition

The first phase in the modeling framework is to define the system itself. This system definition can reflect the following phases to ensure the integration between models that will be developed in each phase.

To define the filtration process, we first set up the model mixture. How many components are being considered, and which ones are present in which streams? It is common to lump and weight-average multiple components [27,41]. It is also common to neglect minor components which are a priori known to have little or no effect on the chosen functionality of the filtration system. This model mixture has to be the simplest representation that still captures the desired complexity to be observed. Knowing this mixture definition, we can roughly decide the filtration process (e.g. micro-, ultra- or nanofiltration, or reverse osmosis). A general guideline for this selection has been widely published and presented as a filtration spectrum of (pressure-driven) membrane processes (Fig. 3).

To anticipate the model being extended into a multi-stage system and later optimized, we need to define the configuration of the multi-stage system. A further question will be if the optimization will be performed within one fixed configuration or if multiple configurations will

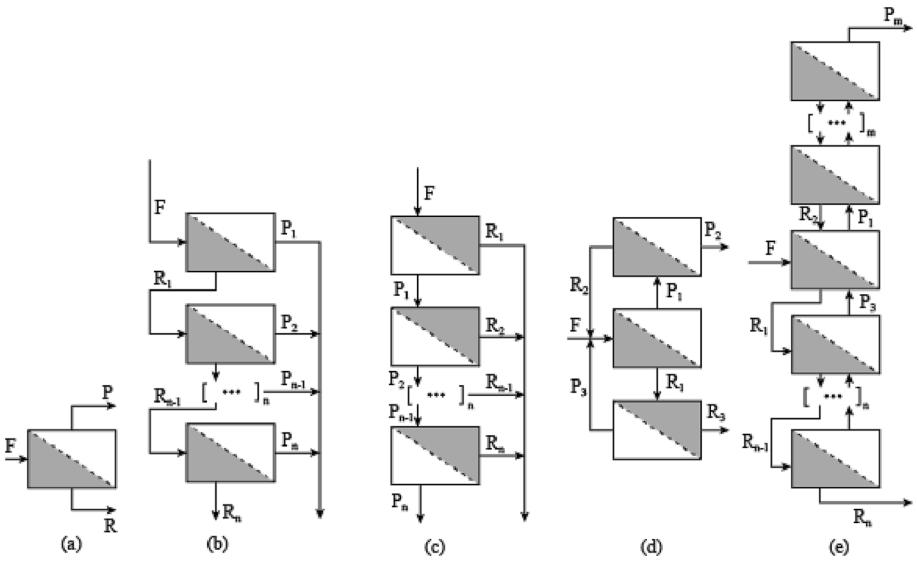


Fig. 1. Illustration of streams and configurations of membrane systems: (a) Single-stage membrane, (b) Sequential concentration, adapted from Cordova et al. [23], (c) Sequential purification, adapted from Ebrahimi et al. [28], (d) Three-stage ideal cascade, adapted from Lightfoot [22], and (e) (General) counter-current cascades, adapted from Caus et al. [33], Abejon et al. [34], and Rizki et al. [35]. F, P, and R represent the feed, permeate, and retentate streams, respectively. Subscripts indicate stage numbers.

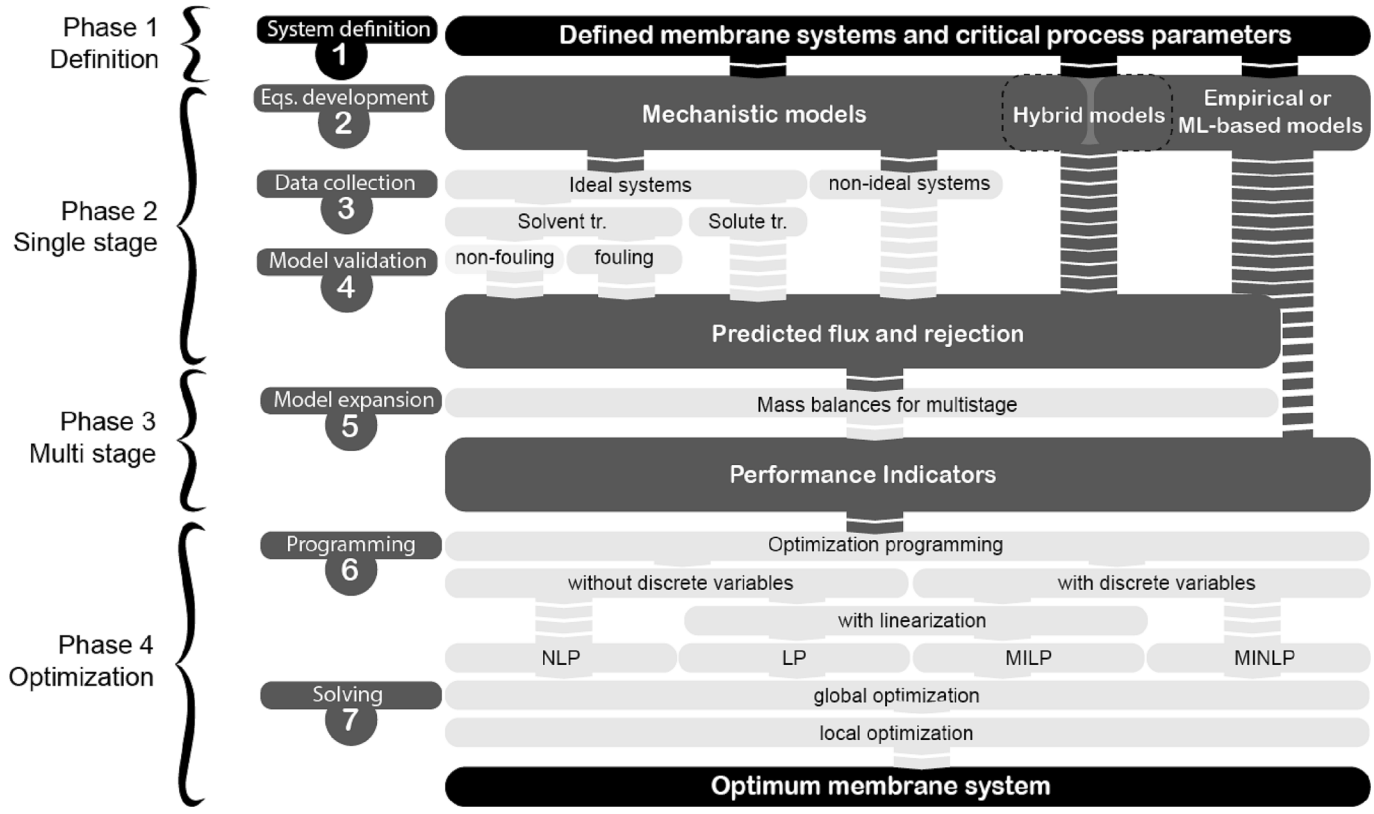


Fig. 2. Model-based optimization framework for multi-stage membrane systems.

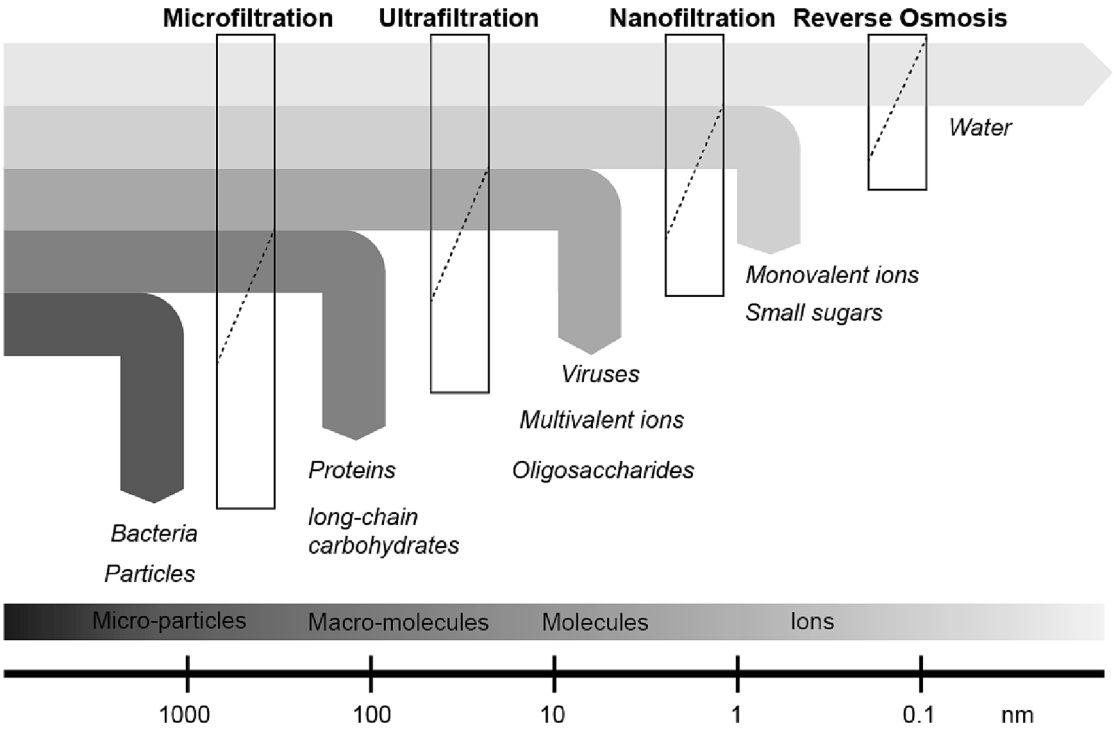


Fig. 3. Filtration spectrum adapted from various commercial membrane specifications dedicated to food, bioprocesses and pharmaceutical applications.

be considered. Head-to-head comparison of several predefined configurations is commonly used for brute-force optimization [26,27,41]. This does not guarantee the global optima are found; there may be another alternative configuration that gives a better separation performance. Adopting a technique from process system engineering, we can develop a superstructure, which is a general structure that represents all possible configurations [42–44]. This superstructure employs logical variables so that a combination of those yields each considered configuration. When

considering multiple configurations to be optimized later, this superstructure needs to be defined in this phase.

While defining the system, it is also important to consider the process variables. These can reflect parameters which are varied and parameters which are kept constant in the set-up. Our model, in principle, relates these variables to certain performance parameters. The optimization will later aid in selecting the best values for these variables. Which of these variables become decision variables will be decided during optimization programming in a later step. However, we need to make sure that these decision variables exist in the model.

2.2. Phase 2: Single-stage filtration modeling

Having defined the system, we next develop a single-stage filtration model. As commonly done for other types of processes, membrane filtration modeling can follow three approaches [45–47]: mechanistic modeling, machine learning (ML)-based modeling, or hybrid modeling, which is a trade-off between the other two approaches. Mechanistic models are process-specific, developed by considering the physical phenomena that happen within the process under consideration. ML-based models, on the other hand, are based on an input–output relationship [48,49], so that a similar modeling technique can be adopted for any type of process.

These three types of models come with their own benefits and drawbacks (Fig. 4). An ideal model should have the desired benefits and tolerable drawbacks. A process scientist may prefer a mechanistic model because it represents a good understanding of the process. However, such a model may require a longer time to develop and to solve.

Over the past decades, pressure-driven membrane processes have been well established and a good understanding of the underlying working principles has been achieved [50–52]. Therefore, it should not be difficult to develop a mechanistic model. Challenges may emerge when attempting to model a new system. Information may be lacking with regard to the model mixture, the membrane itself, or interactions between them. This may require a dedicated study into the new phenomena, or the development of empirical relationships to represent the missing information, which can then be combined into a hybrid model. However, the main limitation of mechanistic models of pressure-driven membrane processes is the availability of either computational power or experimental data. Fig. 4 shows that a mechanistic model is relatively complex compared to other types of models. Extensive computing resources are needed as the model employs sets of many physical equations that may include ordinary or partial differential equations. Even though numerous solvers are readily available, solving this system of equations may still take a considerable amount of time, especially when more variables are involved, for example when working with multiple components or when expanding the model to a multi-stage system.

Fig. 4 shows a general distinction between the three types of models. The mechanistic and ML-based models occupy the opposite ends of the evaluation criteria spectrum. Hybrid models exist in between the two, with a broad range of possibilities depending on the formulation.

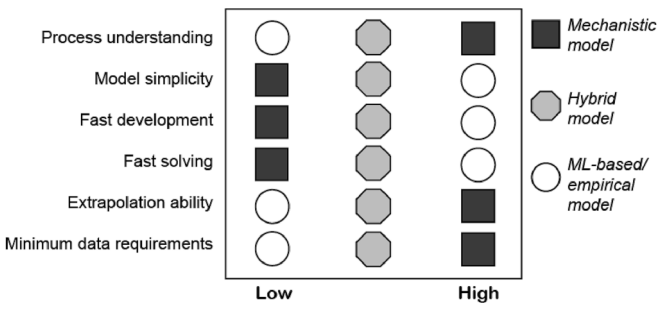


Fig. 4. Graphical comparison of Mechanistic models (black filled squares), hybrid models (grey filled octagons) and ML-based models (open circles).

Despite the clear distinctions presented in Fig. 4, real-world models may deviate from this simplified classification. The performance of the models themselves may vary depending on available resources. For example, access to advanced modeling tools and/or supercomputers may alleviate the computational resources bottleneck and thus aid the development and solving of the models. These advanced tools, however, may not be available or accessible to every-one.

Although it requires less data, a mechanistic model may require not only the process input and output data but also system characterization data. This characterization may include the determination of certain process parameters or material properties under certain conditions. Such a characterization of a new system is often more difficult and expensive to perform than simply gathering process input–output data, as required for ML-based models.

Regardless of the choice of model, the development of a single-stage filtration model follows three steps (steps 2 to 4 in the model-based optimization framework, Fig. 2): equations development, data collection, and model validation. These steps are discussed next.

Step 2 - Equations development.

Once the system and parameters are set, we need to define the relationship between these parameters. This step is the core of the modeling effort, where we use everything we know about the system to formulate equations used to simulate the filtration process. Primary goal of the single-stage filtration modeling is to predict fluxes and solute rejections [53–57]. Having predicted these parameters, other output parameters of the filtration model can be calculated simply by considering total and component mass balances.

Before developing the governing equations, one should consider to what level the model will incorporate such equations. Mechanistic models and empirical models lay at opposite ends of the model complexity scale. In addition, both mechanistic and empirical models can still be decomposed and formulated into several levels of complexity. This closely depends on the problem definition.

Empirical models employ an input–output equation without giving attention to what is really happening in the system. The equation is traditionally selected based on the shape of the input–output plot. A simple equation can rarely capture the actual input–output relationship of a filtration system unless within an exceptionally narrow observation range. Capturing more complex input–output relationships may require the use of more advanced empirical models, such as composites of several equations, or neural networks.

Governing equations in a mechanistic model represent physical phenomena. These phenomena fall within two categories: ideal and non-ideal systems (Fig. 2). Most works on filtration modeling study ideal systems and only a few studies [58–60] have been reported on non-ideal filtration systems. From an industry perspective, this area is less attractive because it involves impracticalities such as difficult maintenance or very low fluxes, affecting the cost level. From a modeling perspective, a non-ideal system is more complicated because the stronger interactions between the solutes require more rigorous computation.

Ideal filtration systems operate within dilute conditions, so that the inter-component interactions can be neglected. The governing equations can then be decomposed into solvent and solute transport, which may affect each other due to convection and concentration polarization. Even so, the effect may be simpler than observed in non-ideal systems.

While formulating solvent transport equations, it is also important to consider if the system is fouled or not. In both cases, we may observe a regime where the flux decreases with time until a steady state is reached. In systems without fouling, the steady-state regime can still be an interesting operational regime [53,61,62]. In contrast, fouled systems can severely decrease the filtration performance so that the process needs to be stopped and the system cleaned [3,63–65] before or not long after steady state is reached. Therefore, dynamic models are more relevant for representing filtration systems with fouling. For non-fouled systems, it is optional to use dynamic or steady-state models, depending on the purpose. One obvious fact to be considered is that steady-state

models are easier to formulate and solve because of the absence of the time variable.

Step 3 – Data collection

After completing the second step, the model may still be incomplete due to missing information, such as material properties, process-related properties or parameters that are not predefined by the equations. Such information is usually available within a database in literature, e.g. a chemical engineering handbook, or from previous research studying (close to) similar systems. When no explicit values are available, an approximation is needed. Fortunately, approximations have been published to estimate certain material and process-related properties using more general attributes, such as molecular weight and process temperature [62,66–68]. These estimations are useful to predict properties of uncommon materials that still belong to a certain group, for example, proteins and carbohydrates [69,70]. As an alternative to those approximations, molecular dynamics simulations have been reported [71,72] to be useful in predicting some physical properties of biomolecules. This approach makes use of properties of substances at a molecular level and simulates how such molecules behave in a mixture.

If the unknown parameters cannot be retrieved from a database or approximated, they can be characterized experimentally. This step is usually compulsory for new systems. A common parameter to be characterized experimentally is the membrane pore size distribution. Experimental characterization can be done by direct measurements or by fitting data using regression. Direct measurement techniques include (nano- or microscopic) imaging and phase change measurements. For a review of available techniques, see for example Hilal et al. [73]. Performing such measurements may not be easy due to the expertise needed to operate analytical instruments or simply due to the cost involved. Parameter fitting [74,75] is a more user-friendly characterization technique with arguably less accuracy compared to direct measurements. It makes use of data that are usually more common or easier to observe. Although the values of fitted parameters can be biased by other parameters that are probably not well defined in the system, they can still be useful enough to predict certain outcomes.

Unlike a mechanistic model, all parameters required for empirical or ML-based models should be fitted. The number of these parameters varies between models. Advanced empirical models like neural networks employ more parameters [76] (also known as weights). Statistically, the required data points to fit parameters increase with the number of the fitted parameters, which makes sense in relation to the degree of freedom. This may also be the reason that neural network models require more data points than other types of models.

Step 4 – Model validation

After completing the missing information by direct measurements of fitting, the model is technically complete and able to predict an outcome based on input information. To understand how good this prediction is, the model needs to be validated. This assessment of the model performance is done by comparing the model outcome with a real dataset. This requires separate experimental datasets that are distinct from the ones used for model development, such as those used for parameter characterization. The validation data set can be obtained from similar experiments, but it is important to carefully design these experiments to ensure that the validation set accurately represents the intended scope and application domain of the model. This challenge becomes even more significant when dealing with limited resources for performing the experiments.

The assessment may employ some statistical metrics (e.g. R^2, root mean squared errors, residuals) for an objective measure. In addition to that, visual plots are often used to see if the models can capture the general shape of the data. The result of the assessment can lead to a decision whether the model is acceptable or needs some improvements. The model could be improved by, for example, incorporating additional mechanisms or tuning the hyperparameters (e.g. number of hidden layers or neurons) for neural network models.

2.3. Phase 3: Expansion to multi-stage models

With a validated model, we can expand the model to describe a larger system comprising multiple membranes. This phase is often less complicated than the other phases in the optimization framework of Fig. 2. Most hurdles have probably been solved in phase 2 (single-stage membrane) with a good foundation in phase 1 (system definition). The main challenge in phase 3 is to formulate a correct mass balance within the system and then to solve it. With a good definition of the configuration (with or without superstructure) in phase 1, this task should be a relatively easy. However, solving the mass balance can still be problematic for dynamic models, especially with recycles. This circumstance creates an inter-dependency between streams that makes it impossible to solve sequentially. A simultaneous solving may be computationally expensive. As the number of variables increases linearly with the number of stages, the increase in computing time can be an even more severe complication.

With a proper formulation, a multi-stage model should be able to predict stream conditions at any point in the system. This information can then be used to formulate performance indicators to assess the filtration performance holistically. Common indicators include product purity, yield, productivity, and separation factors, which in turn can be translated into production cost and revenue, the indicators used for process optimization in phase 4. Note that an ML-based model can directly predict these performance indicators, skipping the intermediate prediction of each stream.

2.4. Phase 4: Optimization

Optimization means finding an optimum condition of a system. This procedure consists of three steps [77]: (1) formulating the knowledge of a system, (2) finding a measure of effectiveness or objective function, and (3) finding the optimum point. The first two steps can be considered as (mathematical) formulation steps, exploiting the available knowledge about the process. The third step is the real optimization step and usually done by selecting an appropriate optimization solver from the available options.

As step 1 (formulating the knowledge of multi-stage membrane systems) has already been performed in the previous phases, the optimization phase is mostly about steps 2 and 3.

Step 6 – Optimization programming

In this step, the optimization problem is formulated mathematically. This formulation, in general, aims to optimize (minimize or maximize) an objective function by varying decision variables within certain constraints. The decision variables can be chosen from a pool of variables that are employed in the filtration model. The objective is usually derived from the performance indicators. At this stage, the developed model should be able to relate decision variables to performance indicators.

Filtration systems are rarely assessed by a single performance indicator, complicating the optimization formulation because of multiple objectives. Nonetheless, multi-objective optimization has been studied thoroughly, and some strategies are available [27,78–80]. A similar approach can then be applied to filtration systems [27,37]. Alternatively, a single-objective optimization can still be performed with respect to one chosen indicator while designating the other indicators as constraints. Another alternative is to convert all indicators into a single metric, such as profit.

During programming, optimization constraints need to be formulated as well. These constraints can be relationships between variables or system boundaries. Bounding variables may simplify the optimization problem by narrowing the search domain, while adding constraints may complicate the optimization problem, requiring an extra strategy for solving. Implicitly, all governing equations from the previous phases are optimization constraints. Thus, the optimization model may already give a satisfactory solution without the need for additional (explicit)

formulation of constraints. In some cases, these constraints do need to be explicitly defined. The challenge is then to explicitly formulate the constraints in terms of the decision variables.

Multiple categories of optimization models can be discerned by considering the linearity of the objective and constraints, and the involvement of discrete variables. In general, models for membrane systems involve an obvious non-linear behavior, which makes the optimization problem a non-linear programming (NLP) problem. Some cases also incorporate discrete parameters in the optimization problem, for example, for superstructure optimization, which complicates the problem into a mixed-Integer non-linear programming (MINLP) problem. MINLP is considered to be the most complicated programming, and may require more advanced resources to solve. It can be wise to simplify the system via relaxation [83,84], linearization [81], or evaluating continuous variables [82]. In doing so, the problem is transformed into a mixed-integer linear programming (MILP) or NLP problem, which is probably easier to solve.

Step 7 – Solver selection and problem solving

The last part of optimization is to solve the problem itself, using an appropriate solver from a large pool of available optimization solvers. It is often possible that some general solvers are capable of solving various problems.

2.5. Surrogate model/hybrid optimization

Regardless of the choice of model (mechanistic, ML-based, or hybrid), an optimized solution for a multi-stage membrane system can be found by following the four-phase optimization framework. While mechanistic models reflect a good understanding of the system and require (relatively) long execution times, ML-based models have the opposite traits. These traits are carried over to the optimization phase.

Optimization is, in principle, an iterative procedure to find the optimum condition. In each iteration, a calculation is done, and repeated in the next iteration until the optimum solution is found. This implies that at every iteration, the membrane model is called to predict the outcome with altered input values. The execution of the optimization model then depends on the execution of the membrane model. The execution time of a single prediction depends on the complexity of the model. Inclusion of a dynamic model instead of a steady-state model, and recycles instead of sequences may increase the execution time of the model. The involvement of more stages in the system can also contribute to a longer execution time. Therefore, it is beneficial to accelerate the calculation to solve the whole optimization problem within an acceptable time frame. Attempts to do so have been reported in the literature by means of surrogate modeling, also known as hybrid optimization [83–85]. This approach combines both models within the optimization framework (Fig. 5). Initially, a mechanistic model is developed. This model is then used to generate a dataset to train an ML-based model. This trained

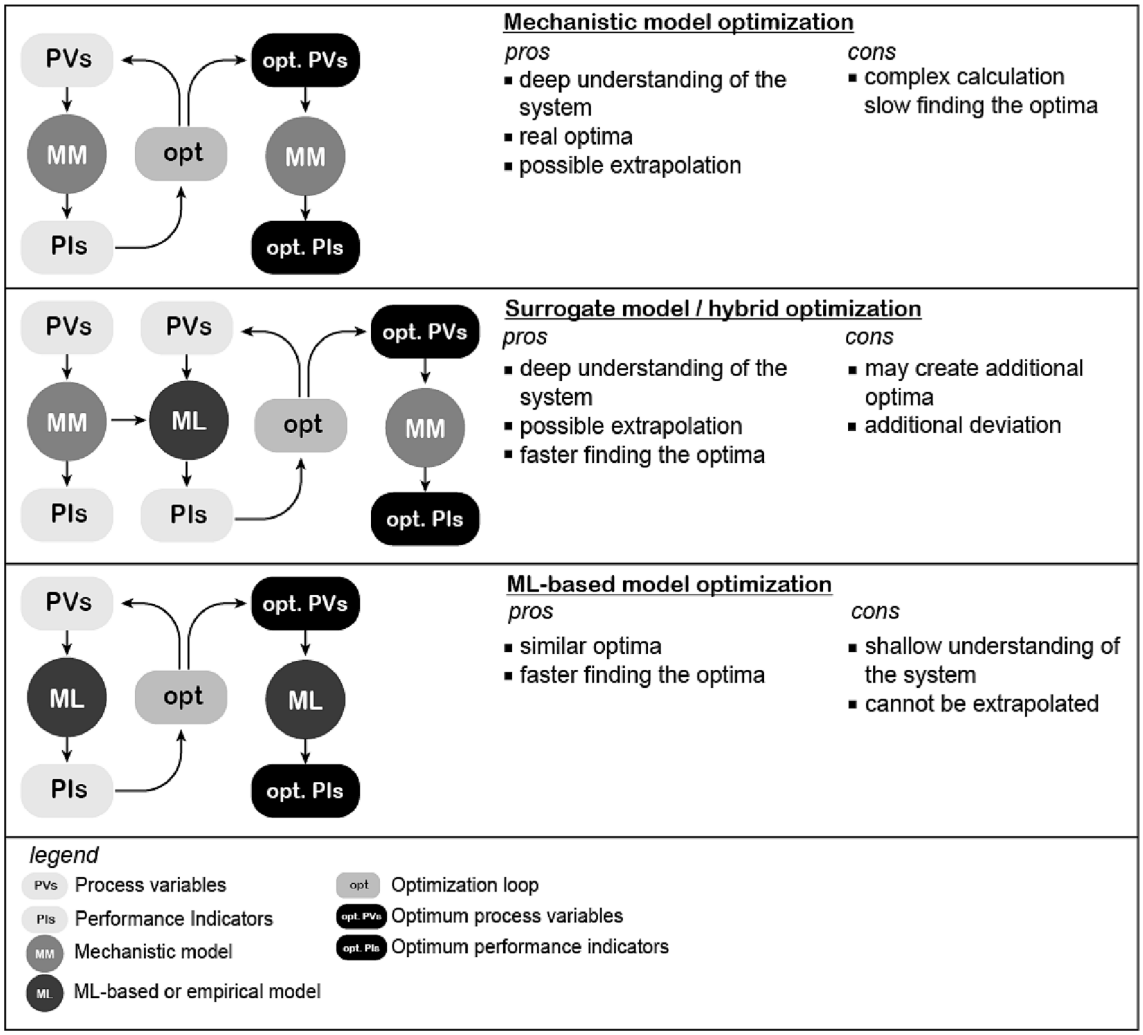


Fig. 5. Summary of the mechanistic model, surrogate model/hybrid optimization, and ML-based model optimization schemes, outlining their working principles as well as their advantages and disadvantages.

model is then used within the optimization loop. The solution from this optimization is finally fed back to the mechanistic model to get a more precise prediction.

Hybrid optimization offers advantages over both mechanistic and ML-based models, and suffers none of their shortcomings. The mechanistic models are utilized for their deep process understanding, generating continuous predictions despite requiring a smaller dataset for development. This data requirement concerns the system characterization and validation. Once developed, mechanistic models can generate as many data as needed within the validity range, which can then be used to train ML-based models. Developing ML-based models without the help of mechanistic models requires numerous experiments to generate the required dataset. Nevertheless, ML-based models are utilized for their fast calculation, thus accelerating the optimization algorithms.

The hybrid optimization approach also comes with pitfalls. It employs not one but two models, which means an additional step. Each additional step in modeling approaches contributes to possible deviations propagating into the final outcome. In the hybrid approach, an ML-based model suffers from a deviation originating from the mechanistic model that produces its training set, which already has a deviation from the real experimental values. This may lead to the objective function having a different profile than those formulated by only mechanistic or ML-based models. Despite different profiles, the global optimum will be similar and can be tested for both models. A global-optimization solver can be used to ensure that the optimization solver is not trapped within local optima.

2.6. Current modeling state

Within the optimization framework (Fig. 2), the most knowledge is needed for step 2: the governing equations for single-stage models. This is reflected in the number of studies into single-stage membrane modeling. The various mechanisms involved in pressure-driven filtration are now well understood. Typical equations used in the models have also been developed over decades. In recent years there have not been many contributions to the fundamental models. Recent studies examine a wider area emerging from new applications.

A good understanding of well-established filtration models commonly comes from studies of simplified systems [53,61,86]. While this type of study is indeed useful as a foundation for working with new systems, additional considerations, especially with regard to the solute transport, are needed to make existing models work for new applications. These aspects are addressed by recent membrane modeling studies, and may include corrections or additional parameters that represent the deviations of new systems with respect to the well-studied simplified systems. Examples are the incorporation of non-spherical solutes [67], pore size distribution [31], and mass transfer estimation [87,88].

The membrane model is constantly evolving to accommodate new applications in the field of circular processes. With the current trend towards waste valorization, innovative processes are being developed to extract valuable components from waste streams. Membrane processes are renowned for their robustness and high productivity, making them a preferred choice over other downstream techniques.

Membrane processes are frequently operated in multiple stages, with fewer studies available for multi-stage systems compared to single-stage systems. The reason for this is that when the membranes serve similar purposes and operate under similar conditions, a single-stage model with multiplication factors is sufficient to predict the behavior of the overall system. It becomes more challenging when the membranes in the system have different functions. For instance, in a three-stage cascade model [22,25,27,62], the membrane system is used for both purification and concentration, requiring a more complex model to account for the different functions. This complexity can be mitigated by applying similar operating conditions for each membrane performing a similar function, reminiscent of a multi-stage distillation process [32,89,90]. Conversely, the challenge becomes more pronounced when individual modules have the freedom to operate with varying process parameters [35,39,91,92], resulting in an inhomogeneous membrane system.

Comprehensive studies spanning all four phases of the optimization framework (Fig. 2) are rare. Regardless, membrane modeling studies in the literature indicate the need for identifying optimum conditions. Some studies include simple attempts at optimization by limiting the alternatives, for example reducing the optimization problem into a binary correlation between one process variable and one performance indicator [93,41]. Other attempts involve discretizing some alternatives and comparing their performance indicators [27,94]. These attempts, however, exempt other variables in the search domain by making such a simplification. The ‘pseudo’ optimum coming out of the comparison is not guaranteed to be the global optimum; the real optimum can come from other exempted alternatives.

Application of advanced optimization programming to solve multi-stage membrane systems has been reported in the literature [36,96,97]. Those studies, however, disregard the complexity of single-stage membrane modeling, using an over-simplified representation and focusing on optimization programming. These attempts can arguably be considered as empirical modeling efforts, following all four the phases reviewed in this article. A better representation of the membrane model can be found in a study where the input–output relationship of single membranes is described and fed to a multistage model, after which it is solved as an MINLP problem [39]. As the input–output relationship is generated from a mechanistic model, this study resembles a hybrid optimization approach.

Separately, all four phases discussed in this section have been intensively studied, often within different disciplines. Studies that integrate the four phases remain scarce. Fig. 6 illustrates the number of scientific articles discussing pressure-driven membranes and optimization. It can be seen that only a small portion of these articles attempt to optimize the problem. The number of articles that explore multi-stage membranes is even smaller. On the other hand, the number of studies of the optimization algorithm alone is 700 times larger than the number of optimization studies dedicated to the membrane process.

This section has presented an overview of such an integrated approach. The governing models in each phase will be discussed in more detail next.

3. Filtration models

The primary goal of filtration models used in process optimization is to predict the outcome of a membrane process with known process parameters at given feed values. In a steady-state condition, this prediction can be done by solving general mass balances for both overall streams (Fig. 1a) and each component (Eqs. (1) and (2)). Here, the mass flow rate of each stream is indicated by $\dot{m}$ and the mass concentration of a component by C_i. The subscript f indicates feed, p permeate, and r retentate.

$$\dot{m}_f = \dot{m}_p + \dot{m}_r \tag{1}$$

$$\dot{m}_f C_{f,i} = \dot{m}_p C_{p,i} + \dot{m}_r C_{r,i} \tag{2}$$

The feed condition (flow rate and composition) is known in most cases. That leaves Eqs. (1) and (2) with four unknowns. Solving those equations requires two more known parameters, which is the core of any filtration model. The first parameter needed is the permeate flow rate that can be calculated from the volumetric flux, J_v, of the membrane as follows.

$$\dot{m}_p = J_v \rho A \tag{3}$$

Here, ρ is the material density and A is the unit dimension.

The second parameter needed to solve Eqs. (1) and (2) is the rejection coefficient, which describes the relation between the concentrations of

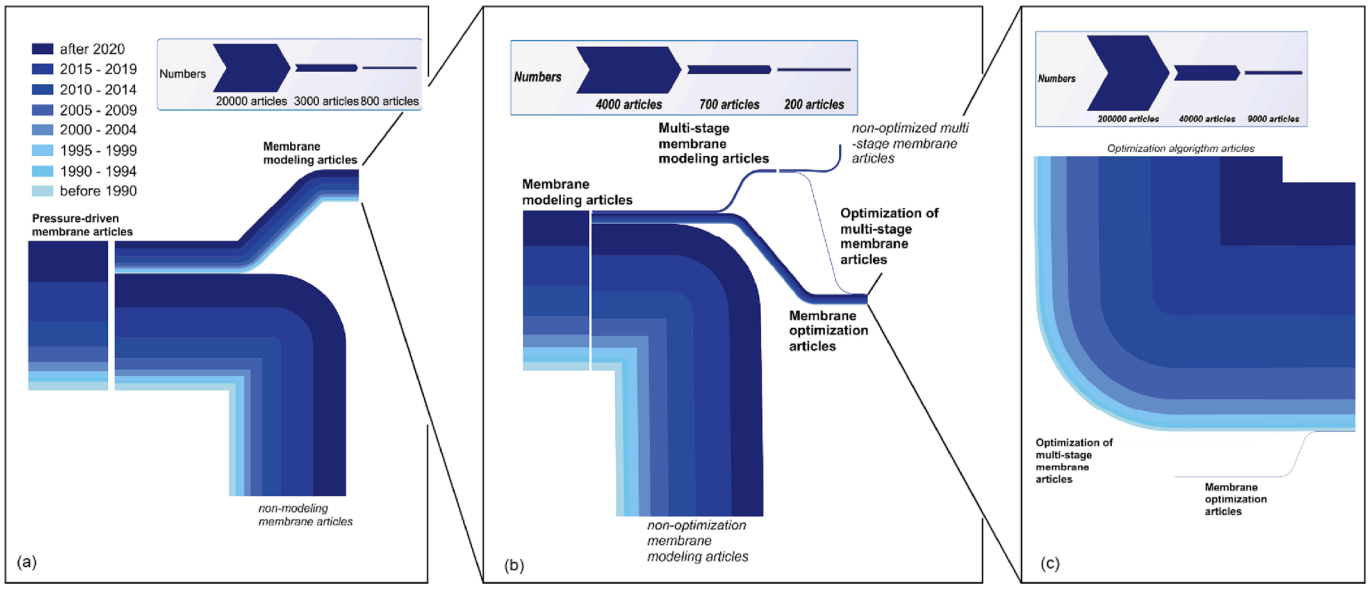


Fig. 6. Sankey diagram of the number of articles related to optimization modeling of multi-stage pressure-drive membranes. The figure is constructed using the Scopus database accessed on 8 February 2023 with keywords "membrane and filtration and pressure" and refined by keywords "model", "multi-stage or cascade or system", and "optimization". (a) Number of articles about pressure-driven membranes, both with and without models. (b) Number of articles about pressure-driven membrane modeling that further discuss multi-stage membrane system and optimization. (c) Number of articles on optimization algorithms, including discussions on membrane optimization.

the solute in permeate and retentate. Because this parameter concerns two observable concentrations, the term 'observed rejection', R_o, is more commonly used (Eq. (4)).

$$R_{o,i} = 1 - \frac{C_{p,i}}{C_{r,i}} \tag{4}$$

The volumetric flux, J_v, and rejection coefficient, $R_{o,i}$, are two key parameters to be modeled in any filtration model. This is not always straightforward, and often requires solving a system of equations simultaneously. The number of equations to be solved depends on the phenomena that are considered in the models. Neglecting those phenomena, both flux and rejection coefficient can also be modeled using ML-based models. The next sections summarize the mechanistic and ML-based models available in the literature.

3.1. Mechanistic models

Most works on mechanistic membrane modeling investigate ideal systems. The models describe the transport of both solvent and solutes, and can be decomposed into several sequences reflecting the

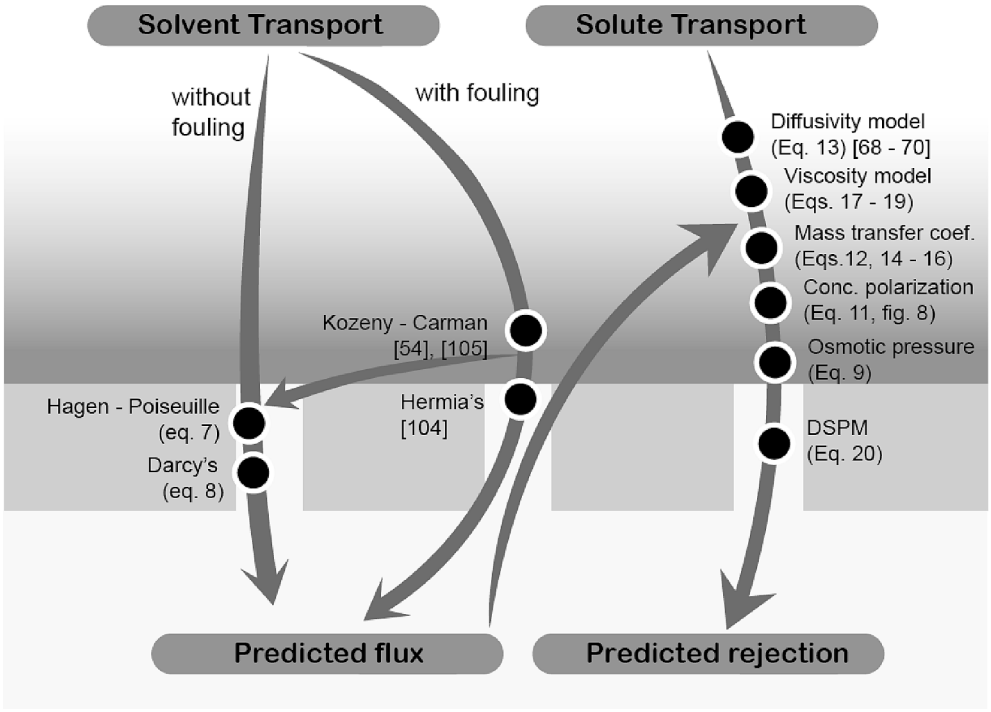


Fig. 7. Schematic overview of mechanistic filtration modelling approaches reported in the literature for ideal systems.

mechanisms at play during filtration. Fig. 7 shows the important sequences in mechanistic modeling of ideal filtration systems.

In general, the solvent flux is expressed using a well-known permeability equation [97] for membrane processes (Eq. (5)), with L_p the membrane permeability and ΔP the *trans*-membrane pressure (TMP).

$$J_v = L_p \bullet \Delta P \tag{5}$$

The permeability is often broken down into more contributing factors. Such factors are commonly expressed in Darcy's equation [98] (Eq. (6)) or Hagen-Poiseuille equation [52] (Eq. (7)). Darcy's equation expresses the effect of viscosity, η, on the flux and total resistance, R_T. The Hagen-Poiseuille equation implies the contribution of membrane radius r and thickness δ to its resistance, ruling out other resistances in a system without solute.

$$J_v = \frac{\Delta P}{\eta R_T} \tag{6}$$

$$J_v = \frac{r^2 \Delta P}{8 \eta \delta} \tag{7}$$

As the pore size gets smaller, transport is no longer ideal as assumed in the Hagen-Poiseuille equation. Regardless, the equation is still commonly used to model tighter membranes (e.g. nanofiltration) with an addition of correction factors. One correction that is commonly used in nanofiltration models is the viscosity correction due to the presence of narrow pores. Studies [53,67] proposed a higher viscosity inside a narrow pore because of the formation of a water layer of one molecule thick on the surface of the pore. Another correction for flux models in nanofiltration membranes stems from the non-uniformity of the pores, inferring average and pore-size distribution parameters. In most cases, using the average pore diameter is sufficient and already able to give a good prediction [53,61]. Additionally incorporating the pore size distribution may give a better prediction, as demonstrated in the work of Bowen and Welfoot [31].

The presence of solutes affects the solvent transport by reducing the driving force while increasing the resistance at the same time. Solutes can be partially or totally retained by a membrane, creating a concentration difference between the two sides of the membrane. This difference triggers a reverse solvent flux due to the osmotic pressure difference. The total solvent flux can then be formulated as a resultant of both fluxes by adding the osmotic pressure correction, $\Delta\pi$, as a driving force in Eqs. (5) to (7). This modification of Eq. (6) is shown in Eq. (8).

$$J_v = \frac{\Delta P - \Delta\pi}{\eta R_T} \tag{8}$$

The osmotic pressure difference is concentration dependent and can be approached in several ways. One of the most utilized models for osmotic pressure is van't Hoff model (Eq. (9)) that assumes an ideal solution. In this model, the osmotic pressure is proportional to the difference in total molar concentration of all solutes between the two sides of the membranes, ΔC_t, and the operating temperature, T, with R the ideal gas constant.

$$\Delta\pi = \Delta C_t \, i \, R \, T \tag{9}$$

The accumulation of retained solute on the membrane surface leads to a reduction in solvent flux, which is referred to as fouling. The fouling mechanism varies from system to system, and depends on the solutes and their interaction with the membrane. It is typically incorporated into the flux model using a resistance-in-series approach [63,64,99]. In this approach, the fouling solute creates an additional resistance, thus reducing the solvent flux. The total resistance, R_t, (Eq. (10)) for solvent transport can be broken down into the resistance from the membrane itself, R_m, and the resistance from fouling, R_f.

$$R_t = R_m + R_f \tag{10}$$

In most cases, the fouling resistance is much higher than the membrane resistance, so that the membrane resistance is often neglected in the calculation of filtration flux involving fouling. The fouling resistance can be formulated differently depending on the fouling mechanism. The formulation may also encompass multiple resistances, as multiple mechanisms (hybrids) may occur at the same time. The mechanisms underlying fouling have been modeled and reviewed in the literature; some models have been well explained, such as Hermia's model [100] for various mechanisms, or the Kozeny-Carman and Ergun equations [52,101] for cake formation.

Despite being a crucial area of study in membrane filtration, models for fouling are mostly limited to the resistance-in-series model and Hermia's model, which dates back to the 1980 s. Both models are semi-empirical and reflect limited knowledge of the underlying phenomena. Hermia's model proposes a unique relationship for each mechanism, but the physical principles behind these relationships are still not fully understood. This knowledge gap has spurred researchers to study fouling phenomena using more advanced process monitoring and more sophisticated modeling techniques, such as computational fluid dynamics.

The transport of solutes during filtration processes consists of transport towards the membrane surface and transport across the membrane. The transport towards the surface occurs because of convection and diffusion, and strongly depends on the hydrodynamic design of the membrane module. In addition to convection and diffusion, transport across the membrane is also affected by the membrane itself. The effect can appear either physically (e.g. size exclusion due to pore size) or chemically (e.g. chemical interaction between active groups on the membrane surface and the solutes) [53,62,102].

Due to convection, the flow of solvent drags along the solute from the retentate side to the membrane surface and finally across the membrane. However, the membrane can (partially) reject certain solutes, creating an accumulation of solutes on the membrane surface. This accumulation triggers a back-diffusion towards the bulk retentate region where the concentration is lower. Such phenomena are well known as concentration polarization [103–105] (Fig. 8). Estimating solvent flux without considering this phenomenon will result in an overestimated flux due to a lower estimate of osmotic pressure difference.

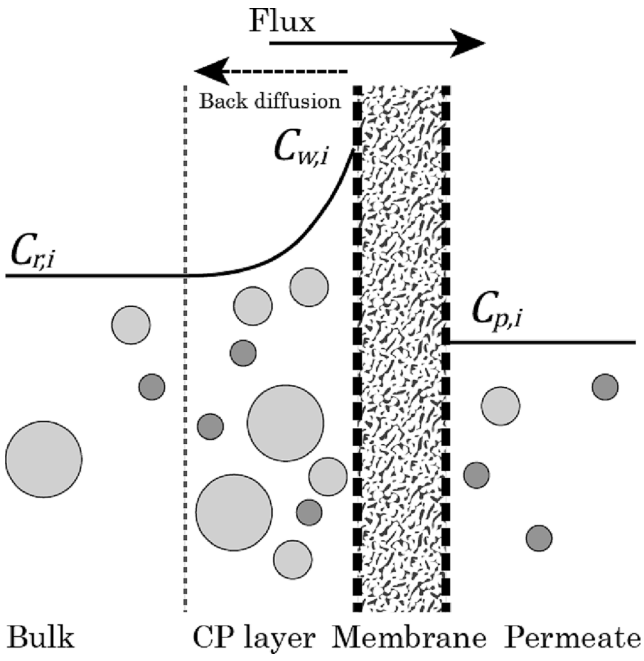


Fig. 8. Schematic illustration of concentration polarization (CP) phenomena. Solid lines indicate concentration profiles for component i with subscripts r = retentate, w = wall, and p = permeate.

After the system reaches an equilibrium, the convection and back diffusion effects will be balanced. In this state, a relationship between concentrations at the retentate bulk, $C_{r,i}$, at the membrane wall, $C_{w,i}$, and at the permeate, $C_{p,i}$, can be formulated (Eq.11). This equation can be solved if the mass transfer coefficient, k_i, is known.

$$\frac{C_{w,i} - C_{p,i}}{C_{r,i} - C_{p,i}} = \exp\left(\frac{J_v}{k_i}\right) \tag{11}$$

The mass transfer coefficient in a convective flow is related to the diffusivity coefficient of convective mass transport, D_i, and the length of the transport pathway, expressed as the hydraulic diameter, d_h, by the dimensionless Sherwood number (Eq. (12)).

$$Sh_i = \frac{k_i d_h}{D_i} \tag{12}$$

The diffusivity coefficient of several solutes are available in the literature and in databases [106–110]. These data usually provide information for common substances such as sucrose or glucose. Common proteins such as lysozyme are sometimes included as well. For uncommon solutes, for which diffusivity data are not available, an approximation may be needed such as the Stokes-Einstein equation [68] (Eq. (13)).

$$D_i = \frac{k_B T}{6 \pi \eta r_{S,i}} \tag{13}$$

This equation can be used to estimate the diffusivity coefficient at a certain temperature, T, for a solution with known viscosity, η, and (Stokes or hydrodynamic) radius, r_S, of the solute. The Stokes radius can be estimated by considering the shape of the molecules involved. For small molecules, a spherical shape can be a good approximation. Thus, the radius can be easily calculated from the molar volume. Molar volume data can be available in literature or estimated from the solute density. Density data are in general more easily to access compared to diffusivity data. For more complex biomolecules, such as carbohydrates and proteins, the size approximation becomes more complicated. To address this issue, some researchers develop semi-empirical relationships to estimate the diffusivity coefficients for uncommon biological compounds. These relationships, however, are mostly studied for certain types of solutes. The application of these relationships can be limited to those types. Some empirical relationships are available in the literature, and frequently summarized and tabulated [62,66,67,111].

The Sherwood number is related to other dimensionless numbers: the Reynold, *Re*, and Schmidt, *Sc*, numbers. The relationships between those dimensionless numbers depend on the hydrodynamics of the systems. The hydrodynamic behavior is often not straightforward and rarely explained mechanistically. Therefore, many empirical relationships are proposed in literature that are valid only for specific regimes and flow patterns. The most common form of the empirical relationship for those dimensionless numbers is expressed as a power law (Eq. (14)).

$$Sh = a\, Re^b Sc^c \tag{14}$$

where

$$Re = \frac{\rho\, u_v\, d_h}{\eta} \tag{15}$$

$$Sc_i = \frac{\eta}{\rho\, D_i} \tag{16}$$

The constants used in this power law depend on the system and are only valid within certain ranges. One common form was expressed by Chilton and Colburn [112] with $a = 0.023$, $b = 0.8$, and $c = 0.33$. This relationship is valid for turbulent flows, which is found in most membrane separation practices.

The Sherwood relationship may depend on the system design, including the flow regime and the module. In addition to the constants proposed by Chilton and Colburn [112], other values for these constants have been proposed by other researchers with multiple validity conditions. Some models use a different form than the power model. Other Sherwood relationships have been described in the literature and tabulated in numerous publications, for example in the works of Gekas and Hallstrom [113], Lisdonk et al. [114], and Bandini and Morelli [115], among other publications.

In order to solve Eqs. (12) – (16) and estimate the mass transfer coefficient, we need to know the solution viscosity, η. In most cases, the viscosity depends on the concentration as the void fraction between molecules decreases when the concentration increases. A general relationship relating solution concentration and viscosity is commonly expressed in the Krieger-Dougherty [116] equation (Eq. (17)). This equation relates the relative viscosity (compared to the solvent viscosity) to the volume fraction of the solution, Φ.

$$\frac{\eta}{\eta_0} = \left(1 - \frac{\Phi}{\Phi_{max}}\right)^{\Phi_{max}[\eta]} \tag{17}$$

The volume fraction depends on the solute concentration. Estimating it may become more complicated for natural polymers such as proteins and carbohydrates. For that purpose, researchers have proposed simpler relationships that directly relate the relative viscosity to the concentration. These relationships are non-linear and commonly expressed as polynomial (Eq. (18)) [117] or exponential (Eq. (19)) [70,118] equations using regressed parameters determined from experiments.

$$\frac{\eta}{\eta_0} = 1 + a_1 c + a_2 c^2 + a_3 c^3 + \cdots$$

$$a_2 = k_1 a_1^2,\ a_3 = k_2 a_1^3,\ \cdots \tag{18}$$

$$\frac{\eta}{\eta_0} = a \exp(E X) \tag{19}$$

Unlike the transport on the membrane surface, few models have been developed to explain the solute transport across the membrane. The Donnan Steric Pore Model (DSPM) has been proven to effectively explain solute transport across the membrane, although corrections have been proposed for some parts of the equations. The DSPM is based on the Extended Nernst-Planck equation, which considers both diffusive and convective transport in the membrane. The diffusive transport occurs due to a gradient of the electro-chemical potential between the two sides of the membrane. The Nernst-Planck equation can be further divided into three parts: convection, diffusion due to the concentration, and diffusion due to the electrical potential, Ψ_m, (Eq. (20)). This equation employs a hindrance coefficient due to convection, $K_{c,i}$, and also diffusion, $K_{d,i}$, along with the solute diffusion coefficient for a diluted system,$D_{i,\infty}$.

$$J_i = K_{c,i}\, c_i\, J_v - K_{d,i}\, D_{i,\infty} \frac{dc_i}{dx} - z_i\, c_i\, K_d\, D_{i,\infty} \frac{F}{R_g T} \frac{d\psi_m}{dx} \tag{20}$$

For neutral solutes, the third term of Eq. (20) is nullified. The solute flux is then only dependent on the convection and diffusion due to the concentration gradient. For both terms, hindrance coefficients are employed, which proportionally relate the solute flux to each driving force. Dechadilok and Deen [119,120] commonly estimate these values using transport of a sphere along a cylindrical pipe, which is a similar assumption in the Hagen-Poiseuille equation.

The DSPM has been widely utilized to model filtration systems for both neutral and charged solutes. Solving Eq. (20) may not be straightforward. However, attempts to solve the equation have been reported for a wide range of filtration systems [53,61,67,121]. These works also successfully couple the DSPM with other models related to the solute transport on the membrane surface, confirming the robustness of DSPM.

In the presence of non-idealities, the flux formulation can reflect back to non-ideal mass transport. The solute flux can then be derived

thermodynamically, starting with the gradient of chemical activity as the driving force [60,122] (Eqs. (21) and (22)). Eqs. (21) and (22) are simplifications of Eq. (20) without considering drag coefficients and electrical interaction. Instead, the gradient of osmotic pressure is assumed to affect the solute flux in the non-ideal system. The osmotic pressure itself is related to the solute concentration. These equations have been validated for concentrated monoclonal antibody solutions in the work of Binabaji et al. [60].

$$J_i = J_v\, c_i - \frac{(D_i c_i)}{RT}\left(\frac{d\mu}{dx}\right) \tag{21}$$

$$\left(\frac{d\mu}{dx}\right) = \frac{1}{c_i}\left(\frac{d\Pi}{dc}\right)\left(\frac{dc}{dx}\right) \tag{22}$$

A similar approach to derive the flux equation based on the activity gradient was taken by Aguirre-Montesdeoca et al. [58]. They developed a filtration model for a concentrated mixture of oligosaccharides, utilizing the Maxwell-Stefan approach. This model is designed to handle more complex systems with multiple solutes that interact with each other, and where the solvent may also affect the transport of the solutes. Inclusion of these effects is the main difference between the Maxwell-Stefan equation and models for ideal solutions, such as Fick's law.

The works of Binabaji et al. and Aguirre-Montesdeoca et al. demonstrate approaches to model filtration for concentrated systems. However, membrane filtration for concentrated systems forms only a small part of existing liquid membrane processes, primarily due to low fluxes making the process economically unfeasible. Binabaji et al. showed that the flux almost reaches zero at the maximum observed concentration. Other factors such as the high osmotic pressure and high viscosity also make these systems difficult to handle from a technology point of view. To our knowledge, the works discussed here have not been extended towards other concentrated solutions. However, they provide additional options for expanding the operation range of membrane process modeling.

3.2. Empirical and Machine Learning-based models

As alternatives to detailed mechanistic models, empirical models can be used to predict the outcomes of membrane filtration processes. These models relate unknown parameters to known variables by employing empirical coefficients. The coefficients are normally fitted using experimental data obtained within a certain range of process conditions. Therefore, the models using such coefficients are only valid for systems operating in similar conditions.

Empirical models adopt common equations to relate input variables to predicted variables. The simplest equation correlates the input variable with the prediction variable in a linear fashion; an increase of the input variable yields a proportional increase in the other variable. Despite its very simple nature, the linear model is still satisfactory for several conditions. Some properties have linear dependencies on common variables, such as water permeability. In addition, linear behavior can also be observed when the window of observation is narrow and the effect small [62,123]. Even a complicated relationship between process variables can be simplified using a linear model by limiting the process range (e.g. a narrow window of process temperatures). Another technique to empirically utilize the linear relationship is by interpolating available data. Here, an unknown data point in between two available data is interpolated by assuming a linear relationship between them. This practice is also found to be satisfactory for certain conditions, and used in the literature [24,26].

For larger parameter ranges, the relationships between variables often lose their linearity, requiring the use of non-linear models. Many available non-linear models have been explained in the literature, including polynomial, exponential, and sigmoid models, among other models.

In addition to the different equations involved, different non-linear models can be distinguished by their unique shapes. A polynomial model can be identified by the presence of turning points (extremes) in the curve. The number of turning points also indicates the polynomial degree. A second-degree polynomial model (quadratic) has only one turning point, a third-degree polynomial model has two turning points, and so on. A rapid increase of the predicted value (J shape) can indicate an exponential relationship. The inverse behavior (a plateau) can be seen in a logarithmic model.

The selection of the empirical model is often done by looking at the shape of the relationship being considered. One can first plot the predicted values against their predictor and see what kind of shape appears. From there, the model can be chosen from the unique shape of the relationship. It often appears that the available data can be represented by multiple shapes, giving multiple candidates of non-linear models. The model selection is then commonly performed by statistically comparing the performance of such candidates by performing an error analysis [124]. Regardless of its usage in comparing models, an error analysis can give an indication of the model performance when no other models are available.

It is often found that the behavior of the predicted variables is not represented straightforwardly by any of the available non-linear models. In this case, a composite of multiple non-linear models (and linear models) can be utilized. This can be translated into a series of sequential models as if the process is performed in two steps; one step follows one model and the second follows another model. Whether the process actually happens in that sequence is not really important since it is not the main objective of the empirical models.

The concept of an artificial sequence of empirical models has found its way into neural network modeling, which is widely used in artificial intelligence development. A neural network model consists of multiple neurons that mimic the neuron structure in our brain. The neurons are connected to each other with activation functions [76]. This structure makes a neural network a huge composite empirical model.

The neurons are organized into several layers, including (1) an input layer, (2) one or multiple hidden layer(s), and (3) an output layer (Fig. 9). The input and output layers consist of a specific number of nodes that depends on the number of predictors being considered, plus the bias node, and the number of outputs that need to be predicted. For the hidden layers, on the other hand, there is no clear relationship between the number of layers and nodes, and the number of predictors and output variables. The structure of the networks, i.e. the number of layers and nodes within them, needs to be defined a priori. There are no exact guidelines for determining such a structure. However, it is obvious that more layers and nodes require more computing power. For a more advanced prediction, which can be the goal of artificial intelligence development, more layers and nodes (deep neural network, [125–127]) may be beneficial to the model performance. For filtration processes, shallow neural networks that consist of 2 – 3 hidden layers are reported to be sufficient [37,128–130].

Each node in a neural network has two connections: input and output. Both of them are related to functions (Fig. 9). The input from each node can come from multiple nodes in the previous layer, making the total input a cumulative sum of weighted values from those nodes. The values of these weights then need to be determined using a training dataset.

The activation functions, in principle, can be chosen from all simple empirical equations discussed above. However, in current practice, only a few functions are commonly used in neural networks (e.g. linear and sigmoid functions) [76]. The selected function is normally generalized for all nodes, i.e. all nodes have a similar activation function. One exception is usually the output layer. The activation function used here depends on the types of expected outputs. A huge number of neural network applications are used for classification [48,76,131], requiring a binary output. For this purpose, a sigmoid function will be suitable. For applications where continuous process parameters need to be predicted, a linear function is more suitable [84,129,130].

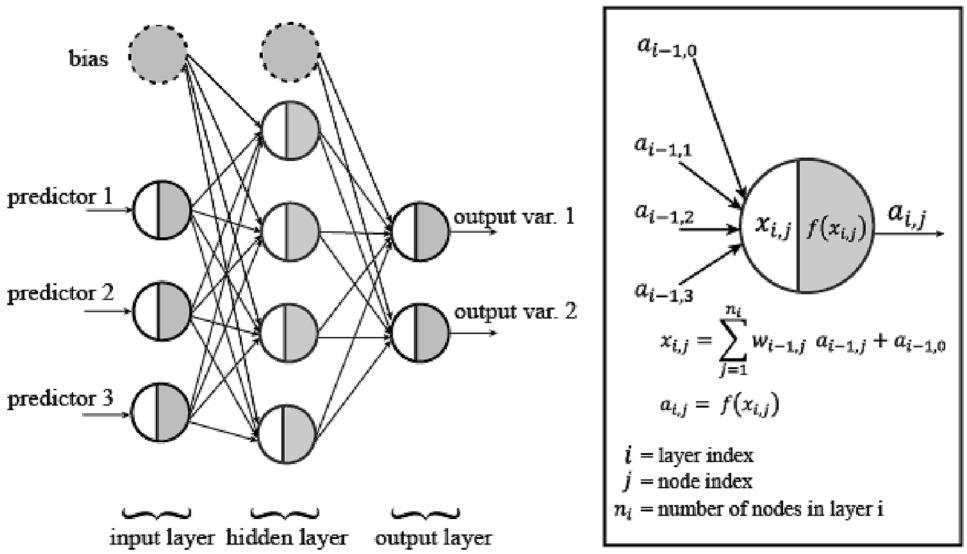


Fig. 9. Schematic illustration of a neural network structure (left) and model (right).

Developing empirical models, including neural networks, can be relatively straightforward and follow a similar framework as mechanistic models. Instead of developing physical equations, candidate models are proposed. These candidates can be selected by analyzing the shape of the input–output plot or through a heuristic approach based on similar data reported in literature. In the case of neural networks, the structure can be chosen through a heuristic approach or initially guessed and later optimized by adjusting the hyperparameters. These hyperparameters, which include the number of hidden layers, the number of nodes in each layer, and the type of activation function, determine the architecture of the neural network and must be decided before training the model [132].

The candidate models are then trained using experimental data. The training process is in principle a regression technique with the objective to minimize the error between the dataset and predicted values [74,75]. Several optimization algorithms are available to perform this regression, as discussed in the next section. From the training, a set of empirical parameters is obtained. For neural networks, these empirical parameters are replaced by a set of weights respective to the nodes in the structure. Once the parameters are determined, the models are complete and ready to be validated. The validation requires a new set of data coming from a different set than the one used for training, which is called the validation set. With these data, the model's ability to predict outputs is evaluated using metrics such as statistical error [124]. If the model's performance is not satisfactory, the second step is repeated by proposing another model or altering the neural network structure. This process can be repeated until the model's performance meets expectations.

Empirical models and neural networks can be a popular option due to their ease of development compared to meticulous calculations in mechanistic models. In addition, unlike mechanistic models, simple empirical models and shallow neural networks do not require extensive computational power to solve. However, empirical models and neural networks give no information about what is happening inside the system. The values of empirical parameters mostly do not represent anything. With this trade-off, one may opt for a hybrid approach, in which a mechanistic model is used towards a certain complexity, and some steps are replaced by an empirical model. The hybrid model can then still provide information about the behavior of the system without getting too complex to solve. We can also decide which part of the mechanism in the model is less important and replace that part with an empirical model.

While simple empirical models are not widely utilized, neural network models continue to garner attention for modeling filtration processes. Simple empirical models exhibit poor performance due to their inability to capture the non-linearities arising from multiple stages of non-linear relationships in each mechanism. In contrast, neural networks can capture such complex non-linearities due to their multiple nodes and layers. The following subsection examines how ML-based models are formulated for various applications in membrane filtration.

3.2.1. *Application of ML-based models for pressure-driven membranes*

Despite differences in the specific applications, the ultimate goal of ML-based models is similar to that of mechanistic models, which is to balance the mass of the output from the membrane processes. To achieve this, the models must predict the flow rates and concentrations of the outlet streams, using the input information and process parameters. However, this does not limit the formulation of the model outputs to flow rates and concentrations.

The output formulation may differ for each application, with a focus on critical parameters. In general, the permeate flow rate, m_p, is determined by multiplying the flux by the cross-sectional area of the membrane, while the retentate flow rate can be calculated from the permeate flow rate. The flux, J_v, is more complex to calculate, and typically formulated as the output of the ML-based models for membrane applications [133–135]. The models can predict the solute concentrations either directly or by predicting the rejection coefficient, $R_{o,i}$, [15,136,137], which relates the concentrations in the permeate and retentate streams ($C_{p,i}$ and $C_{r,i}$,). These concentrations are often expressed as other derivative quantities, such as turbidity, total dissolved solid (*TDS*) or total organic content (*TOC*). Alternatively, ML-based models can be used to predict only one mechanism in the filtration process, resulting in a hybrid model. This approach has been widely used in filtration cases involving fouling, where the ML models are formulated to predict the fouling resistance, R_f, or other derivative quantities leading to fouling characteristics, such as specific fouling growth, α [128,138]. It is easier to estimate membrane flux with a known fouling resistance than to estimate the resistance itself.

The TMP is a crucial factor in the separation processes described in this paper and therefore typically included as an input parameter in the ML-based models. In cases where the models only study isobaric systems, the TMP may be excluded from the input parameters. However, in systems where the aim is to maintain a controlled flux with variable TMPs, pressure buildup may occur to prevent fouling and to maintain the desired flux. In some instances, the TMP itself becomes the output of the ML-based models [139,140].

The input variables of ML-based models are more diverse in com-

parison to the output parameters, and tailored to specific applications. While they may resemble the variables utilized in mechanistic models, the inter-relationships between these variables are not necessarily considered. Feed flow rates and concentrations are frequently utilized as input variables. Other variables derived from the process setup, such as TMP, temperature, T, and cross-flow velocity, u_v, may also be incorporated. For filtration systems that involve charged solutes, further ionic information, such as ionic valence, z_i, zeta potential, ζ, ionic strength, I, and pH, is often included as relevant input variables [15,141]. In addition to the common process input data, studies have reported incorporating detailed analysis data, such as imaging [142], into ML-based models to predict fouling growth. This predicted fouling behavior can then be combined with other models to forecast permeate flux.

Applications of ML-based models have been reported across a broad spectrum of pressure-driven membrane processes, including micro-, ultra-, and nanofiltration, and reverse osmosis. These processes correspond to various product applications, such as cells, proteins, sugars, salts, and water purification. Representative cases of ML-based models in pressure-driven membranes are summarized in Table 1.

Fouling is a significant problem in micro- and ultrafiltration processes. The primary objective of modeling is to predict the decline in flux over time due to fouling. ML-based models are commonly formulated to directly predict the permeate flux [141,143,144] or fouling resistance [138,37], after which they are combined with flux models. The solute concentration is often not considered as a critical factor and is not directly formulated in the model. In micro- and ultrafiltration cases, solutes are either completely rejected or not rejected at all, making it easy to derive their concentration from flow rates if necessary.

ML-based models for micro- and ultrafiltration often incorporate common process parameters like TMP, temperature, and pH as input variables. Due to the dynamic nature of the system resulting from fouling phenomena, time is often included as an input in the model [128,143,145]. Additionally, specific material properties can be included, the choice of which varies between specific cases. They may include the ionic strength and zeta potential of colloids [143], indirect measures of concentration like total organic carbon [146], or turbidity [147].

The prediction of rejection coefficients is particularly important for nanofiltration cases as these membranes are used for more selective separations. As a result, ML-based models for nanofiltration often have the rejection coefficient as an output variable [15,148], in addition to permeate flux. This is also seen in reverse osmosis (RO) cases [149–151].

3.2.2. Current status of ML-based models for membrane separations

Despite the lack of a fundamental explanation, ML-based models can accurately predict critical parameters in pressure-driven membranes across a broad range of applications. These models are flexible enough to

Table 1
Summary of representative ML-based models for pressure-driven membrane applications, which include microfiltration (MF), ultrafiltration (UF), nanofiltration (NF), and reverse osmosis (RO).

Type	Input	Output	Feed	Performance	Ref.
MF	TMP, u_v, t	R_t	cane sugar	R^2 = 0.94–0.99	[145]
MF	$TMP, C_{f,i}, u_v, t$	R_h	yeast	Error = 11 %	[152]
MF	m_f, J_v, α, μ, t	TMP	natural water	Error = 10 %	[146]
MF	$m_f, T, C_{f,i}(turbidity, TOC), t$	$J_v, R_f(foulingindex)$			
MF	Cf, pH, r_p	$Rf(\alpha)$	yeast suspension	MSE = 9.85E-8	[153]
UF	TMP, pH, I, ζ, t	$J_v, foulinggrowth$	silica coloids	Average error = 5.6 %	[135]
UF	TMP, u_v, T, pH	J_v, R_f	waste water	R^2 0.99, RMSE = 7.2 E-4	[37]
UF	pH, I, ζ	J_v	BSA	Error = 2.7 %	[154]
UF	$TMP, m_p, T, C_{f,i}(turbidity)$	R_t	natural water	Error < 5 %	[139]
UF	TMP, t	$J_v, R_t, R_{o,i}$	milk	Error < 1 %	[128]
UF	$pH, t, C_{f,i}(fatcontent)$	J_v, R_t	milk	Error < 3.6 %	[155]
UF	$TMP, C_{f,i}, t$	$J_v, C_{p,i}$	fruit juice	MAE = 0.076 lmh	[156]
UF	$Jv, t, C_{f,i}(turbidity)$	TMP	(model) natural water	SSE = 10^{-4}	[157]
UF	m_f, t	J_v	BSA	RE = 5 %	[147]
NF	$TMP, C_{f,i}, pH, D_i, z_i$	$R_{o,i}$	NaCl, Na_2SO_4, $MgCl_2$, $MgSO_4$	Error = 1.751 × 10^{-5}	[15]
NF	$TMP, C_{f,i}$	$J_v, R_{o,i}$	NaCl, $MgCl_2$	5 % deviation	[158]
NF	TMP, u_v	J_v	humic acid	RE = 0.1 %	[159]
NF/RO	$TMP, pH, \zeta, T, MW, othersaltproperties$	$R_{o,i}$	NaCl, $MgSO_4$	R^2 = 0.986	[160]
NF/RO	$Foulingimage, membranetype, J_{v,0}, t$	$J_v, R_f(foulinggrowth)$	humic acid	R^2 = 0.99 (fouling), R^2 = 0.99 (flux)	[142]
RO	TMP, T, C_f, u_v	$J_v, R_{o,i}$	ethanol / acetic acid	10 % accuracy	[149]
RO	TMP, T, C_f	J_v	NaCl	R^2 = 0. 998	[134]
RO	TMP, T, m_f, TDS, t	$m_p, Cp(TDS)$	sea water	R^2 = 0.96 (TDS)	[161]
RO	$TMP, T, pH, C_{f,i}(conductivity)$	$m_p, Cp(TDS)$	sea water	R^2 = 0.75 (mp) R^2 = 0.99	[162]

model both steady-state and dynamic systems. Studies utilizing ML-based models have shown their robustness in addressing various issues in membrane separations. It can be assumed that the models are useful for addressing unforeseen problems within pressure-driven membrane applications. The simplicity of their formulation may increase interest in industrial applications where historical data are readily available and the underlying mechanisms not yet fully understood.

One of the main challenges of ML-based models is their limited generalizability. The models are only effective within the range of data they were trained on and may not perform well when applied to new data outside of this range. The formulation approach and architecture of ML-based models are transferable, providing a valuable source of knowledge for future applications. This includes the definition of input and output variables, which can be applied across different cases. However, it is important to keep in mind that the model's parameters and performance are specific to the training data and may not be directly transferable to new cases unlike parameters defined for mechanistic models.

Current studies have shown that ML-based models can perform well within specific problems. However, the application of these models to more complex systems and to optimization is still an area of ongoing research. One study reported the integration of a neural network model with a genetic algorithm for process optimization [37], but this study was limited to single-stage processes. However, a similar approach to model development can be applied to each stage of a multi-stage system, with the possibility of retraining the models for each stage as they may need to operate within different parameter ranges.

Future efforts to improve the efficiency of ML-based models and reduce the need for repetitive training could include incorporating physical information into the training process. This might offer the potential to extend the validity range and lower the size of the training set [163], leading to fewer retraining needs. The training set can be designed to encompass the range of all stages involved in the system, thereby reducing the need for retraining. The field of physics-informed ML is currently growing, and has potential for application in multi-stage membrane systems.

4. Models for membrane systems

With models for single membranes available, efforts to model membrane systems focus on solving stream balances. The material balances between the inlet and outlet streams of a single membrane unit have been incorporated in the single-stage model. The membrane system model then looks at the interconnections between inlet and outlet streams for certain configurations: which streams become the inlet streams and which streams are mixed with other streams? It is then important to have a single-stage model that is valid through all stream conditions (e.g. different concentrations). This single-stage model will then become a building block for the membrane system structure with stream balance to be solved. Solving steady-state stream balance itself is not difficult, especially in a sequential order.

The move towards larger systems with more stages, and to a generalization of small system models has shifted the focus of modeling works towards system design and process synthesis, which is the main scope of process system engineering. Models of small membrane systems (with only a few stages) have been reported in the literature [22,26,91,92]. Most of these works focus on developing single-stage models without addressing the extension of these models to full membrane systems. Some design parameters are introduced in these works, such as stage cut, volume concentration factor, and separation factor. The models are then used to assess the overall performance of the system in relation to those design parameters, at which point the performance of the individual membrane is no longer important.

Membrane systems are often designed based on simplifications, which limit the design options to solve the calculations more easily. One common simplification is the use of identical membranes operated under similar process conditions [25,89,164]. The overall performance will then be a multiplication or power of the performance of a single membrane. The number of membranes can then be introduced as a variable, n, in the design. Another simplification is to replace the whole single-stage model with constants or ratios, such as separation factors that are found heuristically. This practice is even simpler than empirical models. While the accuracy of this approach is questionable when evaluating the performance of each stage, it can be satisfying enough for the initial design, for example, as a part of larger systems or a plant design.

The performance of inhomogeneous systems with variable membrane properties and process operations has been reported to be better than that of homogeneous ones with a similar number of stages [26,27,91]. However, the calculation is more rigorous with stage-by-stage evaluation instead of using one general model. This a trade-off that has to be considered.

Modeling membrane system with a pre-defined single-stage model is like building a structure with existing building blocks. Solving the system can be done by sequentially solving the building blocks. However, sequential solving only works for simple scenarios, namely a steady-state system without recycling. With current trends towards waste stream utilization and circular processing, recycled streams are more commonly found in practice. Some techniques already incorporate recycle streams in their models, applying some constraints for simplification.

Solving models with recycling cannot be done sequentially because of the interdependency between units that are connected by the recycle streams. The model should then be solved holistically. One well-known example of a simple recycling configuration was introduced in the work of Lightfoot [22] with three membranes. Generalizing this concept to more membranes, the large system still has three sections as in Lightfoot's cascaded configuration, i.e. feed stage, refining section, and concentration section. Such a configuration resembles the classical design of a distillation column, which is commonly solved using the McCabe-Thiele technique. A similar approach has been used for membrane systems [34,35,89].

Another approach to model multi-stage membrane systems is to utilize flow sheet design tools. These tools make it easy to design multi-stage membrane systems by simply drawing the flow configurations within the system. The conditions of each flow are then calculated automatically. These tools typically require custom-defined models for each unit, which can be accomplished using the single-stage models defined in the previous section. This approach has been demonstrated in the literature [165,166] and is widely used in industry.

5. Optimization models

In general, optimization can be mathematically formulated as follows [167]:

$$\text{Optimize } F = f(x)$$

$$\text{Subject to } h(x) = 0$$

$$k(x) \leq 0$$

Here, the general problem is to optimize objective function F with equality, h, and inequality, k, constraints. The optimization problem can be formulated to find a maximum, a minimum, or a combination of both, of the objective function. These optima can appear regionally (local optima) or globally (global optima). Finding global optima may require more extensive techniques [168,169] while local optimization can be more sensitive since it searches within a smaller domain.

Both objective and constraints must be formulated as functions of decision variable(s), x. How these functions are formulated is the major factor in classifying mathematical programming cases. Generally, the programming cases are classified based on their linearity and continuity.

The simplest form of programming is when the functions are linear with continuous variables, which is called linear programming (LP) [170]. Similar as with continuous variables, non-linear programming (NLP) falls in the second category of mathematical programming. This type of programming can involve any type of non-linearity in the objective, constraints, or both. This may range from quadratic functions (also known as quadratic programming, QP [171]) to complex functions as a composite of many non-linear relationships. When discrete variables are included in the problem formulation, the problem becomes integer programming [172,173]. The discrete variables are often included in combination with continuous variables, resulting in a more complex approach known as mixed-integer linear (MILP) or non-linear programming (MINLP) [39,174,175]. To the best of our knowledge, MINLP is the most complex optimization problem discussed in the literature.

Available optimization solvers or algorithms are often developed for specific programming types. Therefore, it is important to define which programming case is to be formulated. It is commonly found that a solver developed to solve a complex problem also performs well in solving a simpler problem. However, this does not work the other way around. Sometimes, simplifying the case by linearization or relaxation can be a strategic option in optimization modeling [81,176].

Solving an optimization problem is in principle a means to find a maximum (or minimum) value of a function within certain constraints. This search is done iteratively by comparing points until the maximum is found. The strategy to find the next iterate is crucial, and distinguishes one algorithm from another. A good algorithm is able to find the real optima with only a few iterations. Based on the technique to find the next iterate, optimization algorithms can be classified into gradient-based and gradient-free algorithms.

Gradient-based algorithms search for an optimum based on the fact that the optimum has a gradient of zero ($\nabla f(x) = 0$). Thus, the algorithm looks for a new iterate with a lower derivative value than that of the starting point. Finding the new iterate from the initial point requires two major parameters: the search direction, $\boldsymbol{p_i}$, and the step length, $\alpha_i \boldsymbol{p_i}$ [177,178]. Thus, the computation of the new iterate can be formulated as follows:

$$x_{i+1} = x_i + \alpha_i \boldsymbol{p_i} \tag{23}$$

Within the gradient-based algorithm, the calculation of both α_i and $\boldsymbol{p_i}$ requires the function derivative. The iteration is then stopped when the updated iterate has a gradient value of (computationally) zero.

As the main goal of gradient-based algorithms is to find the new iterate with the lowest gradient, it is then logical to find the new iterate based on the steepest gradient descent. This idea led to the gradient descent (or steepest descent) algorithm [178], which calculates the direction vector, $\boldsymbol{p_i}$, based on its local derivative. Therefore, the step size gets smaller as the search is gets closer to the optimum, which makes the algorithm slow. Further improvements of this algorithm focus on a more straightforward search that finds the optimum without significantly slowing down close to the optimum. Among other algorithms in this cluster, Newton's method (Eq. 61) has gained a lot of attention and has been applied to many cases [179–181].

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)} \tag{24}$$

Gradient-based algorithms are not always practical to solve an optimization problem. One reason is that some cases simply do not have explicit derivatives or the derivatives are difficult to formulate. Alternative algorithms to find the optimum without employing the function gradients (gradient-free algorithms) have been developed and published in the literature. While all gradient-based algorithms have similar stopping criteria – to find the zero-gradient point – gradient-free algorithms have various stopping criteria that are unique to each algorithm. The comparison between iterates is not done via their gradient values, but via the values of the iterates themselves. The mechanism of gradient-free algorithms is similar to gradient-based algorithms, which is to find the next iterate coming from the initial point. The difference lies in the direction and step size of the search. Here, no derivatives are used to determine the next iterate.

The iteration in gradient-free algorithms may start from a group of starting points. The group can be selected randomly, initiated by the algorithm itself or predefined by the user. The points are then structurally evaluated. From that evaluation, the algorithms determine which points should be dropped and which ones should remain. From these remaining points (or a single point) new candidates are generated to find a new group for the next iteration. The evaluation and generation steps are then repeated. Algorithms can be distinguished based on the techniques they use to evaluate the group and to generate new candidates.

Many algorithms are available. Some advanced techniques may be difficult to access or require paid licenses. Published algorithms have been summarized in several reviews [182–184], from which a suitable algorithm can be selected.

6. Outlook and discussion

In practice, multi-stage membrane systems, as shown in Fig. 1, are more common than single-stage membranes. The need for multi-stage membrane models only emerges when different purposes are assigned to each stage. Multi-purpose membrane systems are used to handle multi-component mixtures, which are more prevalent than simplified binary mixtures. In these systems, several components are considered valuable, so that processes involving multiple products are desired. Such processes are also known as fractionation processes, where several fractions are extracted from the feed mixture. As each fraction may have different properties, multi-purpose separation processes are required. Here, each membrane stage can be tuned to effectively separate the targeted fraction.

For binary mixtures, multi-stage membrane models are only considered necessary in the presence of recycle streams that create interdependencies between stages. Recycle streams are usually introduced when the by-product stream still contains valuable compounds, which usually appears due to low selectivity between the main product and by-product compounds within one particular membrane. Another reason to introduce recycles is to reduce the fresh solvent requirement in diafiltration [30,185]. Diafiltration itself is commonly designed as a multi-stage process because of the limitation of permeated compounds. As a consequence, each diafiltration stage requires fresh solvent, which increases the production cost. While back-looping the permeate stream immediately to the same stage is a futile effort, using permeate streams coming from the subsequent diafiltration stage as diafiltrate significantly reduces the fresh solvent requirement [186].

Fractionation systems and systems with recycles are getting more attention due to the trend towards circular processes. Here, two principles apply: make use of as many compounds as possible and discharge as little waste as possible. This drives the evolution of multi-stage membrane systems from classical two-outlet-stream membrane processes.

Industrial and lab-scale examples of fractionation systems exist, usually in the context of multi-component feed streams. Such feed streams are more prevalent in food, bio- and pharmaceutical processes that involve natural sources. One well-known case is milk fractionation [2,5,187], the components of which are widely utilized within the dairy industry. Other examples come from extracting solutions from crops containing macromolecules such as proteins and carbohydrates, or functional micromolecules such as colorants, antioxidants, or other bioactive compounds. The fractionation processes can be used to fractionate mixtures of different proteins [5,7] or various fractions of carbohydrates [24], or to strip those functional components [188–190].

In the bio-pharmaceutical industry, the need for multistage membrane fractionation systems appears during down-stream processing. Here, mixtures of metabolites produced during fermentation are

handled [191–194]. These mixtures often contain multiple components of interest.

All the mentioned systems preferably perform at their optimum conditions. The optimization framework discussed in this review is a valuable tool for ensuring they do. With the proposed approach discussed in this paper, one can efficiently design an optimum process, preventing a time-consuming and costly trial-and-error approach.

7. Conclusion

This paper explores various methods for optimizing membrane systems using a four-phase modeling approach. Despite its numerous potential applications, the optimization of membrane systems through a comprehensive modeling approach is underrepresented in the literature. Most studies focus either on the development of membrane filtration models without proper optimization or on rigorous optimization with oversimplified filtration models. This paper presents an overview of both perspectives and provides an integrated guideline, including alternative pathways within this integration. The paper also discusses the advantages and drawbacks of different methods and highlights the use of both detailed mechanistic approaches and empirical or machine learning-based approaches, as well as hybrid optimization approaches.

CRediT authorship contribution statement

Zulhaj Rizki: Conceptualization, Writing – original draft, Writing – review & editing, Visualization. **Marcel Ottens:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This publication is part of a project called ‘Novel Process Routes’ and is financially supported by the Institute for Sustainable Process Technology (ISPT) and TKI-E&I with the supplementary grant ’TKI- Toeslag’ for Topconsortia for Knowledge and Innovation (TKI’s) of the Ministry of Economic Affairs and Climate Policy. More information can be found at https://ispt.eu/projects/novel-process-routes/.

References

[1] F.P. Cuperus, H.H. Nijhuis, Applications of membrane technology to food processing, Trends Food Sci. Technol. 4 (9) (1993) 277–282, https://doi.org/10.1016/0924-2244(93)90070-q.
[2] V. Dhineshkumar, D. Ramasamy, Review on Membrane Technology Applications in Food and Dairy Processing, J. Appl. Biotechnol. Bioeng. 3 (5) (Jul. 2017) 399–407, https://doi.org/10.15406/jabb.2017.03.00077.
[3] A.W. Mohammad, C.Y. Ng, Y.P. Lim, G.H. Ng, Ultrafiltration in Food Processing Industry: Review on Application, Membrane Fouling, and Fouling Control, Food and Bioprocess Technology, vol. 5, no. 4. Springer, pp. 1143–1156, May 28, 2012. doi: 10.1007/s11947-012-0806-9.
[4] V. Calabrò, A. Basile, Economic analysis of membrane use in industrial applications, Adv. Membr. Sci. Technol. Sustain. Energy Environ. Appl. (Jan. 2011) 90–109, https://doi.org/10.1533/9780857093790.1.90.
[5] A. Zydney, Protein Separations Using Membrane Filtration: New Opportunities for Whey Fractionation, Int. Dairy J. 8 (98) (1998) 243, https://doi.org/10.1016/S0958-6946(98)00045-4.
[6] S. Iltchenco, et al., Whey protein concentration by ultrafiltration and study of functional properties, Ciência Rural 48 (5) (2018) 1–11, https://doi.org/10.1590/0103-8478cr20170807.
[7] N.v. Patil, A.E.M. Janssen, R.M. Boom, Separation of whey proteins using cascaded ultrafiltration, Sep Sci Technol, vol. 49, no. March 2015, p. 140620110053007, 2014, doi: 10.1080/01496395.2014.927488.
[8] M.T.C. Machado, S. Trevisan, J.D.R. Pimentel-Souza, G.M. Pastore, M. D. Hubinger, Clarification and concentration of oligosaccharides from artichoke extract by a sequential process with microfiltration and nanofiltration membranes, J Food Eng 180 (2016) 120–128, https://doi.org/10.1016/j.jfoodeng.2016.02.018.
[9] A.K. Goulas, P.G. Kapasakalidis, H.R. Sinclair, R.A. Rastall, A.S. Grandison, Purification of oligosaccharides by nanofiltration, J Memb Sci 209 (1) (2002) 321–335, https://doi.org/10.1016/S0376-7388(02)00362-9.
[10] Y.M. Feng, X.L. Chang, W.H. Wang, R.Y. Ma, Separation of galacto-oligosaccharides mixture by nanofiltration, J Taiwan Inst Chem Eng 40 (3) (2009) 326–332, https://doi.org/10.1016/j.jtice.2008.12.003.
[11] X. Li, S. Tan, J. Luo, M. Pinelo, “Nanofiltration for separation and purification of saccharides from biomass,” Frontiers of Chemical Science and Engineering, vol. 15, no. 4. Higher Education Press Limited Company, pp. 837–853, Aug. 01, 2021. doi: 10.1007/s11705-020-2020-z.
[12] A. Hinkova, Z. Bubník, P. Kadlec, J. Pridal, Potentials of separation membranes in the sugar industry, Sep Purif Technol 26 (2002) 101–110, https://doi.org/10.1016/S1383-5866(01)00121-6.
[13] Z. Rizki, A.E.M. Janssen, A. van der Padt, R.M. Boom, Separation of fructose and glucose via nanofiltration in presence of fructooligosaccharides, Membranes (Basel) 10 (10) (2020) pp, https://doi.org/10.3390/membranes10100298.
[14] K. Pan, Q. Song, L. Wang, B. Cao, A study of demineralization of whey by nanofiltration membrane, Desalination 267 (2–3) (2011) 217–221, https://doi.org/10.1016/j.desal.2010.09.029.
[15] W.R. Bowen, M.G. Jones, J.S. Welfoot, H.N.S. Yousef, Predicting salt rejections at nanofiltration membranes using artificial neural networks, Desalination 129 (2) (Jul. 2000) 147–162, https://doi.org/10.1016/S0011-9164(00)00057-6.
[16] K.B. Hicks, High-Performance Liquid Chromatography of Carbohydrates, Adv Carbohydr Chem Biochem vol. 46, no. C (Jan. 1988) 17–72, https://doi.org/10.1016/S0065-2318(08)60164-x.
[17] K. Robards, M. Whitelaw, Chromatography of monosaccharides and disaccharides, J Chromatogr A vol. 373, no. C (1986) 81–110, https://doi.org/10.1016/S0021-9673(00)80209-5.
[18] K.C. Khulbe, C. Feng, T. Matsuura, The Art of Surface Modification of Synthetic Polymeric Membranes, Polym. Polym. Compos. 21 (7) (2013) 449–456, https://doi.org/10.1002/app.31108.
[19] L. Upadhyaya, X. Qian, S. Ranil Wickramasinghe, Chemical modification of membrane surface — overview, Curr Opin Chem Eng 20 (2018) 13–18, https://doi.org/10.1016/j.coche.2018.01.002.
[20] X. Yang, R. Wang, A. G. Fane, C. Y. Tang, I. G. Wenten, “Membrane module design and dynamic shear-induced techniques to enhance liquid separation by hollow fiber modules: A review,” Desalination and Water Treatment, vol. 51, no. 16–18. Taylor and Francis Inc., pp. 3604–3627, 2013. doi: 10.1080/19443994.2012.751146.
[21] J.I. Marriott, E. Sørensen, I.D.L. Bogle, Detailed mathematical modelling of membrane modules, Comput Chem Eng 25 (4–6) (May 2001) 693–700, https://doi.org/10.1016/S0098-1354(01)00670-6.
[22] E.N. Lightfoot, T.W. Root, J.L. O’Dell, Emergence of ideal membrane cascades for downstream processing, Biotechnol Prog 24 (3) (2008) 599–605, https://doi.org/10.1021/bp070335l.
[23] A. Córdova, C. Astudillo, L. Santibañez, A. Cassano, R. Ruby-Figueroa, A. Illanes, Purification of galacto-oligosaccharides (GOS) by three-stage serial nanofiltration units under critical transmembrane pressure conditions, Chem. Eng.Research and Design 117 (2017) 488–499, https://doi.org/10.1016/j.cherd.2016.11.006.
[24] Z. Rizki, A.E.M. Janssen, R.M. Boom, A. van der Padt, Oligosaccharides fractionation cascades with 3 outlet streams, Sep Purif Technol 221 (2019), https://doi.org/10.1016/j.seppur.2019.03.086.
[25] S.S. Gunderson, W.S. Brower, J.L. O’Dell, E.N. Lightfoot, Design of membrane cascades, Sep Sci Technol 42 (10) (2007) 2121–2142, https://doi.org/10.1080/01496390701444121.
[26] V. Aguirre Montesdeoca, A. van der Padt, R.M. Boom, A.E.M. Janssen, Modelling of membrane cascades for the purification of oligosaccharides, J Memb Sci 520 (2016) 712–722, https://doi.org/10.1016/j.memsci.2016.08.031.
[27] Z. Rizki, A.E.M. Janssen, G.D.H. Claassen, R.M. Boom, A. van der Padt, Multi-criteria design of membrane cascades: Selection of configurations and process parameters, Sep Purif Technol 237 (2020), https://doi.org/10.1016/j.seppur.2019.116349.
[28] M. Ebrahimi, Z. Kovacs, M. Schneider, P. Mund, P. Bolduan, P. Czermak, Multistage filtration process for efficient treatment of oil-field produced water using ceramic membranes, Desalination Water Treat 42 (1–3) (2012) 17–23, https://doi.org/10.1080/19443994.2012.682964.
[29] G. Foley, Water usage in variable volume diafiltration: comparison with ultrafiltration and constant volume diafiltration, Desalination 196 (1–3) (2006) 160–163, https://doi.org/10.1016/j.desal.2005.12.011.
[30] W. Richard Bowen, A. Wahab Mohammad, Diafiltration by nanofiltration: Prediction and optimization, AIChE J 44 (8) (1998) 1799–1812, https://doi.org/10.1002/aic.690440811.
[31] W.R. Bowen, J.S. Welfoot, Modelling of membrane nanofiltration-pore size distribution effects, Chem Eng Sci 57 (8) (2002) 1393–1407, https://doi.org/10.1016/S0009-2509(01)00412-2.
[32] N.v. Patil, A.E.M. Janssen, R.M. Boom, The potential impact of membrane cascading on downstream processing of oligosaccharides, Chem Eng Sci 106 (2014) 86–98, https://doi.org/10.1016/j.ces.2013.11.007.

[33] A. Caus, L. Braeken, K. Boussu, B. van der Bruggen, The use of integrated countercurrent nanofiltration cascades for advanced separations, J. Chem. Technol. Biotechnol. 84 (3) (2009) 391–398, https://doi.org/10.1002/jctb.2052.
[34] R. Abejón, A. Garea, A. Irabien, Integrated countercurrent reverse osmosis cascades for hydrogen peroxide ultrapurification, Comput Chem Eng 41 (2012) 67–76, https://doi.org/10.1016/j.compchemeng.2012.02.017.
[35] Z. Rizki, A.E.M. Janssen, A. van der Padt, R.M. Boom, Design of nanofiltration cascades for fructooligosaccharides using the McCabe-Thiele approach, Sep Purif Technol 259 (Mar. 2021), 118094, https://doi.org/10.1016/j.seppur.2020.118094.
[36] A. Aliaga-Vicente, J.A. Caballero, M.J. Fernández-Torres, Synthesis and optimization of membrane cascade for gas separation via mixed-integer nonlinear programming, AIChE J 63 (6) (Jun. 2017) 1989–2006, https://doi.org/10.1002/AIC.15631.
[37] R. Soleimani, N.A. Shoushtari, B. Mirza, A. Salahi, Experimental investigation, modeling and optimization of membrane separation using artificial neural network and multi-objective optimization using genetic algorithm, Chem. Eng. Res. Des. 91 (5) (May 2013) 883–903, https://doi.org/10.1016/j.cherd.2012.08.004.
[38] V.S.K. Adi, M. Cook, L.G. Peeva, A.G. Livingston, B. Chachuat, Optimization of OSN Membrane Cascades for Separating Organic Mixtures, in: Computer Aided Chemical Engineering, vol. 38, Elsevier B.V, 2016, pp. 379–384, https://doi.org/10.1016/B978-0-444-63428-3.50068-0.
[39] Z. Rizki, A.E.M. Janssen, E.M.T. Hendrix, A. van der Padt, R.M. Boom, G.D. H. Claassen, Design optimization of a 3-stage membrane cascade for oligosaccharides purification using mixed integer non-linear programming, Chem Eng Sci 231 (Feb. 2021), 116275, https://doi.org/10.1016/j.ces.2020.116275.
[40] K.M. (Katalin M.) Hangos, I.T. Cameron, A systematic approach to model building, Process Systems Engineering 4 (2001) 19–40. https://doi.org/10.1016/S1874-5970(01)80026-6.
[41] S. Brover, Y. Lester, A. Brenner, E. Sahar-Hadar, Optimization of ultrafiltration as pre-treatment for seawater RO desalination, Desalination 524 (Feb. 2022), 115478, https://doi.org/10.1016/J.DESAL.2021.115478.
[42] L. Mencarelli, Q. Chen, A. Pagot, I.E. Grossmann, A review on superstructure optimization approaches in process system engineering, Comput Chem Eng 136 (May 2020), https://doi.org/10.1016/j.compchemeng.2020.106808.
[43] H. Yeomans, I.E. Grossmann, A systematic modeling framework of superstructure optimization in process synthesis, Comput Chem Eng 23 (1999) 709–731.
[44] L. Mencarelli, Q. Chen, A. Pagot, I.E. Grossmann, A review on superstructure optimization approaches in process system engineering, Comput Chem Eng 136 (May 2020), 106808, https://doi.org/10.1016/J.COMPCHEMENG.2020.106808.
[45] D. Solle et al., "Between the Poles of Data-Driven and Mechanistic Modeling for Process Operation," Chemie-Ingenieur-Technik, vol. 89, no. 5. Wiley-VCH Verlag, pp. 542–561, May 01, 2017. doi: 10.1002/cite.201600175.
[46] D. Keulen, G. Geldhof, O. le Bussy, M. Pabst, M. Ottens, Recent advances to accelerate purification process development: A review with a focus on vaccines, J Chromatogr A 1676 (Aug. 2022), 463195, https://doi.org/10.1016/J.CHROMA.2022.463195.
[47] C.R. Bernau, M. Knödler, J. Emonts, R.C. Jäpel, J.F. Buyel, The use of predictive models to develop chromatography-based purification processes, Front Bioeng Biotechnol 10 (Oct. 2022) 1896, https://doi.org/10.3389/FBIOE.2022.1009102/BIBTEX.
[48] D. Dhall, R. Kaur, M. Juneja, Machine learning: A review of the algorithms and its applications, Lecture Notes in Electrical Engineering 597 (2020) 47–63, https://doi.org/10.1007/978-3-030-29407-6_5.
[49] C.F. Galinha, J.G. Crespo, From Black Box to Machine Learning: A Journey through Membrane Process Modelling, Membranes 2021, Vol. 11, Page 574, vol. 11, no. 8, p. 574, Jul. 2021, doi: 10.3390/membranes11080574.
[50] B. van der Bruggen, M. Mänttäri, M. Nyström, Drawbacks of applying nanofiltration and how to avoid them: A review, Sep Purif Technol 63 (2) (2008) 251–263, https://doi.org/10.1016/j.seppur.2008.05.010.
[51] B. van der Bruggen, C. Vandecasteele, T. van Gestel, W. Doyen, R. Leysen, A review of pressure-driven membrane processes in wastewater treatment and drinking water production, Environ. Prog. 22 (1) (2003) 46–56, https://doi.org/10.1002/ep.670220116.
[52] M. Mulder, Basic Principles of Membrane Technology, 1996 10.1007/978-94-009-1766-8.
[53] W.R. Bowen, J.S. Welfoot, Modelling the performance of membrane nanofiltration-critical assessment and model development, Chem Eng Sci 57 (7) (2002) 1121–1137, https://doi.org/10.1016/S0009-2509(01)00413-4.
[54] M. Krippl, I. Bofarull-Manzano, M. Duerkop, A. Dürauer, "Hybrid Modeling for Simultaneous Prediction of Flux, Rejection Factor and Concentration in Two-Component Crossflow Ultrafiltration," Processes 2020, Vol. 8, Page 1625, vol. 8, no. 12, p. 1625, Dec. 2020, doi: 10.3390/PR8121625.
[55] C.J.D. Fell, K.J. Kim, V. Chen, D.E. Wiley, A.G. Fane, Factors determining flux and rejection of ultrafiltration membranes, Chem. Eng. Process. 27 (3) (Jun. 1990) 165–173, https://doi.org/10.1016/0255-2701(90)87005-4.
[56] B. Wang, M. Ando, H. Yin, B. Hong, Y. Peng, "Mathematical Modeling of Flux in Ultrafiltration Membrane for Water Treatment," https://doi.org/10.1080/01496390600641587, vol. 29, no. 7, pp. 1179–1188, Jul. 2007, doi: 10.1080/01496390600641587.
[57] C. Quezada, H. Estay, A. Cassano, E. Troncoso, R. Ruby-Figueroa, "Prediction of Permeate Flux in Ultrafiltration Processes: A Review of Modeling Approaches," Membranes 2021, Vol. 11, Page 368, vol. 11, no. 5, p. 368, May 2021, doi: 10.3390/MEMBRANES11050368.
[58] V. Aguirre Montesdeoca, A.E.M. Janssen, R.M. Boom, A. van der Padt, Fine ultrafiltration of concentrated oligosaccharide solutions – Hydration and pore size distribution effects, J Memb Sci 580 (2019) 161–176, https://doi.org/10.1016/j.memsci.2019.03.019.
[59] E. Binabaji, J. Ma, S. Rao, A.L. Zydney, Ultrafiltration of highly concentrated antibody solutions: Experiments and modeling for the effects of module and buffer conditions, Biotechnol Prog 32 (3) (May 2016) 692–701, https://doi.org/10.1002/BTPR.2252.
[60] E. Binabaji, J. Ma, S. Rao, A.L. Zydney, Theoretical analysis of the ultrafiltration behavior of highly concentrated protein solutions, J Memb Sci 494 (Nov. 2015) 216–223, https://doi.org/10.1016/j.memsci.2015.07.068.
[61] S. Bandini, V. Morelli, Effect of temperature, pH and composition on nanofiltration of mono/disaccharides: Experiments and modeling assessment, J Memb Sci 533 (February) (2017) 57–74, https://doi.org/10.1016/j.memsci.2017.03.021.
[62] Z. Rizki, E. Suryawirawan, A.E.M. Janssen, A. van der Padt, R.M. Boom, Modelling temperature effects in a membrane cascade system for oligosaccharides, J Memb Sci 610 (2020), 118292, https://doi.org/10.1016/j.memsci.2020.118292.
[63] A. Suki, A.G. Fane, C.J.D. Fell, Modeling fouling mechanisms in protein ultrafiltration, J Memb Sci 27 (2) (Jun. 1986) 181–193, https://doi.org/10.1016/S0376-7388(00)82055-4.
[64] K.M. Persson, J.L. Nilsson, Fouling resistance models in MF and UF, Desalination 80 (2–3) (May 1991) 123–138, https://doi.org/10.1016/0011-9164(91)85153-L.
[65] G. Vidrich, et al., High Concentration Protein Ultrafiltration: a Comparative Fouling Assessment, IOP Conf Ser Mater Sci Eng 358 (1) (May 2018), 012056, https://doi.org/10.1088/1757-899X/358/1/012056.
[66] Y. Sano, S. Yamamoto, Mutual Diffusion Coefficient of Aqueous Sugar Solutions, J. Chem. Eng. Jpn. 26 (6) (1993) 633–636, https://doi.org/10.1252/jcej.26.633.
[67] V. Aguirre Montesdeoca, J. Bakker, R. M. Boom, A. E. M. Janssen, and A. van der Padt, "Ultrafiltration of non-spherical molecules," J Memb Sci, vol. 570–571, pp. 322–332, Jan. 2019, doi: 10.1016/j.memsci.2018.10.053.
[68] C. Cruickshank, "The Stokes-Einstein law for diffusion in solution," Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, vol. 106, no. 740, pp. 724–749, Dec. 1924, doi: 10.1098/RSPA.1924.0100.
[69] K. D. Schwenke, B. Raab, P. Plietz, G. Damaschun, "The structure of the 12 S globulin from rapeseed (Brassica napus L.)," 1983.
[70] J. Chirife, M.P. Buera, A simple model for predicting the viscosity of sugar and oligosaccharide solutions, J Food Eng 33 (3) (1997) 221–226, https://doi.org/10.1016/S0260-8774(97)00060-5.
[71] S. Miyamoto, K. Shimono, Molecular Modeling to Estimate the Diffusion Coefficients of Drugs and Other Small Molecules, Molecules 25 (22) (2020) Nov, https://doi.org/10.3390/molecules25225340.
[72] J.T. Bullerjahn, S. von Bülow, G. Hummer, Optimal estimates of self-diffusion coefficients from molecular dynamics simulations, J Chem Phys 153 (2) (Jul. 2020), 024116, https://doi.org/10.1063/5.0008312.
[73] N. Hilal, A. F. Ismail, T. Matsuura, D. Oatley-Radcliffe, "Preface," Membrane Characterization, pp. xix–xxi, Feb. 2017, doi: 10.1016/B978-0-444-63776-5.05001-4.
[74] "Least-squares regression and covariance," Understanding Regression Analysis, pp. 26–30, Nov. 2007, doi: 10.1007/978-0-585-25657-3_6.
[75] J.O. Rawlings, S.G. Pantula, D.A. Dickey, Applied regression analysis: a research tool, New York: Springer (1998), https://doi.org/10.1007/b98890.
[76] K. Gurney, N. York, An introduction to neural networks, 1997. https://doi.org/10.1201/9781315273570.
[77] D.J. Wilde, C.S. Beightler, "Foundations of Optimization" (1967).
[78] W. Stadler, Fundamentals of Multicriteria Optimization, Multicriteria Optimization in Engineering and in the Sciences, pp. 1–25, 1988, doi: 10.1007/978-1-4899-3734-6_1.
[79] J. Dodgson, M. Spackman, A. Pearman, L. Phillips, Multicriteria Analysis: A Manual, 2001.
[80] M. Zeleny, Multiple Criteria Decision Making (MCDM): From Paradigm Lost to Paradigm Regained? J. MultiCriteriaDecis. Anal. 110 (October) (2010) https://doi.org/10.1002/mcda.
[81] M. Asghari, A. M. Fathollahi-Fard, S. M. J. Mirzapour Al-E-Hashem, M. A. Dulebenets, "Transformation and Linearization Techniques in Optimization: A State-of-the-Art Survey," Mathematics, vol. 10, no. 2. MDPI, Jan. 01, 2022. doi: 10.3390/math10020283.
[82] Y. Saif, A. Elkamel, M. Pritzker, "Superstructure optimization for the synthesis of chemical process flowsheets: Application to optimal hybrid membrane systems,", vol. 41, no. 4, pp. 327–350, Apr. 2009, doi: 10.1080/03052150802488399.
[83] S.H. Kim, F. Boukouvala, Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques, Optim Lett 14 (4) (Jun. 2020) 989–1010, https://doi.org/10.1007/S11590-019-01428-7.
[84] S.M. Pirrung, L.A.M. van der Wielen, R.F.W.C. van Beckhoven, E.J.A.X. van de Sandt, M.H.M. Eppink, M. Ottens, Optimization of biopharmaceutical downstream processes supported by mechanistic models and artificial neural networks, Biotechnol Prog 33 (3) (2017) 696–707, https://doi.org/10.1002/btpr.2435.
[85] A.J. Booker, J.E. Dennis, P.D. Frank, D.B. Serafini, V. Torczon, M.W. Trosset, A rigorous framework for optimization of expensive functions by surrogates, Structural Optimization 17 (1) (1999) 1–13, https://doi.org/10.1007/BF01197708.

[86] J.E. Almazán, E.M. Romero-Dondiz, V.B. Rajal, E.F. Castro-Vidaurre, Nanofiltration of glucose: Analysis of parameters and membrane characterization, Chem. Eng. Res. Des. 94 (Feb. 2015) 485–493, https://doi.org/10.1016/J.CHERD.2014.09.005.
[87] R.B. Bird, W.E. Stewart, E.N. Lightfoot, Transport phenomena, R. B. Bird, W. E. Stewart, and E. N. Lightfoot, John Wiley and Sons Inc, New York (1960). 780 pages. $11.50, AIChE J 7 (2) (Jun. 1961) 5J–6J, https://doi.org/10.1002/aic.690070245.
[88] A. Giacobbo, E.V. Soares, A.M. Bernardes, M.J. Rosa, M.N. de Pinho, Atenolol removal by nanofiltration: a case-specific mass transfer correlation, Water Sci. Technol. 81 (2) (Jan. 2020) 210–216, https://doi.org/10.2166/WST.2020.073.
[89] A. Lejeune, M. Rabiller-Baudry, T. Renouard, Design of membrane cascades according to the method of McCabe-Thiele: An organic solvent nanofiltration case study for olefin hydroformylation in toluene, Sep Purif Technol 195 (December) (2018), https://doi.org/10.1016/j.seppur.2017.12.031.
[90] R. Agrawal, J. Xu, Gas-Separation Membrane Cascades Utilizing Limited Numbers of Compressors, AIChE J 42 (8) (1996) 2141–2154, https://doi.org/10.1002/aic.690420806.
[91] N.V. Patil X. Feng J.J.W. Sewalt R.M. Boom A.E.M. Janssen Separation of an inulin mixture using cascaded nanofiltration Sep Purif Technol 146 2015 261 267 10.1016/j.seppur.2015.03.061.
[92] N.V. Patil, et al., Continuous purification of galacto-oligosaccharide mixtures by using cascaded membrane filtration, J. Chem. Technol. Biotechnol. 91 (5) (2016) 1478–1484, https://doi.org/10.1002/jctb.4746.
[93] J. Shao, A.L. Zydney, Optimization of ultrafiltration/diafiltration processes for partially bound impurities, Biotechnol Bioeng 87 (3) (Aug. 2004) 286–292, https://doi.org/10.1002/BIT.20113.
[94] S. M. Seyed Shahabadi, A. Reyhani, Optimization of operating conditions in ultrafiltration process for produced water treatment via the full factorial design methodology," Sep Purif Technol, vol. 132, pp. 50–61, Aug. 2014, doi: 10.1016/J.SEPPUR.2014.04.051.
[95] V.S.K. Adi, M. Cook, L.G. Peeva, A.G. Livingston, B. Chachuat, Optimization of OSN Membrane Cascades for Separating Organic Mixtures, Comput. Aid. Chem. Eng. 38 (Jan. 2016) 379–384, https://doi.org/10.1016/B978-0-444-63428-3.50068-0.
[96] J. Adrian Chavez Velasco, R. T. Gooty, M. Tawarmalani, R. Agrawal, "Optimal Design of Membrane Cascades for Gaseous and Liquid Mixtures via MINLP," 2021.
[97] D. K. R. Ernest J. Henley, J. D. Seader, "Separation Process Principles, 3rd Edition International Student Version".
[98] S. Whitaker, "Flow in porous media I: A theoretical derivation of Darcy's law," Transport in Porous Media 1986 1:1, vol. 1, no. 1, pp. 3–25, Mar. 1986, doi: 10.1007/BF01036523.
[99] A.F. Derradji, S. Taha, G. Dorange, Application of the resistances in series model in ultrafiltration, Desalination 184 (1–3) (Nov. 2005) 377–384, https://doi.org/10.1016/j.desal.2005.02.059.
[100] J. Hermia, Blocking filtration: application to non-Newtonian fluids, NATO ASI Ser., Ser. E 88 (1985) 83–89, https://doi.org/10.1007/978-94-009-5091-7_5.
[101] W. L. Mccabe, J. C. Smith, and P. Harriott, Unit operations of chemical engineering, 1993.
[102] V. Starov, D. Lloyd, A. Filippov, S. Glaser, Sieve mechanism of microfiltration separation, Sep Purif Technol 26 (2002) 51–59.
[103] S. Ilias, R. Govind, A Study on Concentration Polarization in Ultrafiltration, Sep Sci Technol 28 (1993) 361–381, https://doi.org/10.1080/01496399308019495.
[104] L. Song, M. Elimelech, Theory of concentration polarization in crossflow filtration, J. Chem. Soc. Faraday Trans. 91 (19) (1995) 3389–3398, https://doi.org/10.1039/FT9959103389.
[105] G.A. Denisov, Theory of concentration polarization in cross-flow ultrafiltration: gel-layer model and osmotic-pressure model, J Memb Sci 91 (1994) 173–187, https://doi.org/10.1016/0376-7388(94)00035-2.
[106] Y.-C. Kim, A.S. Myerson, Diffusivity of protein in aqueous solution, Korean J. Chem. Eng. 13 (3) (1996) 288–293, https://doi.org/10.1007/BF02705952.
[107] D.M. Clarke, M. Dole, Diffusion in Sugar Solutions. III. Diffusion of Sucrose in Glucose Solutions as Solvent, J Am Chem Soc 76 (14) (1954) 3745–3751, https://doi.org/10.1021/ja01643a049.
[108] H.C. Price, J. Mattsson, B.J. Murray, Sucrose diffusion in aqueous solution, PCCP 18 (28) (Jul. 2016) 19207, https://doi.org/10.1039/C6CP03238A.
[109] Y. Chenyakin, D.A. Ullmann, E. Evoy, L. Renbaum-Wolff, S. Kamal, A.K. Bertram, Diffusion coefficients of organic molecules in sucrose-water solutions and comparison with Stokes-Einstein predictions, Atmos. Chem. Phys 17 (2017) 2423–2435, https://doi.org/10.5194/acp-17-2423-2017.
[110] W.M. Haynes, CRC Handbook of Chemistry and Physics, vol. 97th ed. CRC Press, 2016. 10.1201/9781315380476.
[111] D. Brune, S. Kim, Predicting protein diffusion coefficients (linear operator theory/Fredholm theorems), Proc. Natl. Acad. Sci. USA 90 (1993) 3835–3839.
[112] C.O. Bennett, J.E. Myers, Momentum, heat, and mass transfer, McGraw-Hill, 1982, p. 832. https://doi.org/10.1002/aic.690200538.
[113] V. Gekas, B. Hallstrom, Mass transfer in the membrane concentration polarization layer under turbulent cross flow. I. Critical literature review and adaptation of existing Sherwood correlations to membrane operations, J Memb Sci 30 (1987) 153–170, https://doi.org/10.1016/S0376-7388(00)81349-6.
[114] C. A. C. van de Lisdonk, B. M. Rietman, S. G. J. Heijman, G. R. Sterk, J. C. Schippers ~', "Prediction of supersaturation and monitoring of scaling in reverse osmosis and nanofiltration membrane systems," 2001. doi: 10.1016/S0011-9164(01)00272-7.
[115] S. Bandini, V. Morelli, Mass transfer in 1812 spiral wound modules: Experimental study in dextrose-water nanofiltration, Sep Purif Technol 199 (Jun. 2018) 84–96, https://doi.org/10.1016/j.seppur.2018.01.044.
[116] I.M. Krieger, T.J. Dougherty, "A Mechanism for Non-Newtonian Flow in Suspensions of Rigid Spheres", *Citation*, Trans. Soc. Rheol. 3 (1959) 137, https://doi.org/10.1122/1.548848.
[117] R. Simha, Effect of Concentration on the Viscosity of Dilute Solutions, J Res Natl Bur Stand 42 (1934) 1949.
[118] R. Dandekar, A.M. Ardekani, New Model to Predict the Concentration-Dependent Viscosity of Monoclonal Antibody Solutions, Mol Pharm 18 (12) (Dec. 2021) 4385–4392, https://doi.org/10.1021/acs.molpharmaceut.1c00561.
[119] W.M. Deen, Hindered transport of large molecules in liquid-filled pores, AIChE J 33 (9) (1987) 1409–1425, https://doi.org/10.1002/aic.690330902.
[120] P. Dechadilok, W.M. Deen, Hindrance factors for diffusion and convection in pores, Ind Eng Chem Res 45 (21) (2006) 6953–6959, https://doi.org/10.1021/ie051387n.
[121] V.S. Kumar, K.S. Hariharan, K.S. Mayya, S. Han, Volume averaged reduced order Donnan Steric Pore Model for nano fi ltration membranes, DES 322 (2013) 21–28, https://doi.org/10.1016/j.desal.2013.04.030.
[122] A.L. Zydney, Concentration effects on membrane sieving: development of a stagnant film model incorporating the effects of solute-solute interactions, J Memb Sci 68 (1992) 183–190.
[123] W.K. Kim, S. Milster, R. Roa, M. Kanduč, J. Dzubiella, Permeability of Polymer Membranes beyond Linear Response, Macromolecules (Jun. 2022), https://doi.org/10.1021/acs.macromol.2C00605.
[124] D.A. Jones, Statistical Analysis of Empirical Models Fitted by Optimization, Biometrika 70 (1) (1983) 67–88, https://doi.org/10.2307/2335944.
[125] G. Montavon, W. Samek, K.R. Müller, Methods for interpreting and understanding deep neural networks, Digit Signal Process 73 (Feb. 2018) 1–15, https://doi.org/10.1016/J.DSP.2017.10.011.
[126] H. Yi, S. Shiyu, X. Duan, Z. Chen, "A study on Deep Neural Networks framework," Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2016, pp. 1519–1522, Feb. 2017, doi: 10.1109/IMCEC.2016.7867471.
[127] H. il Suk, "An Introduction to Neural Networks and Deep Learning," Deep Learning for Medical Image Analysis, pp. 3–24, Jan. 2017, doi: 10.1016/B978-0-12-810408-8.00002-X.
[128] M.A. Razavi, A. Mortazavi, M. Mousavi, Dynamic modelling of milk ultrafiltration by artificial neural network, J Memb Sci 220 (1–2) (Aug. 2003) 47–58, https://doi.org/10.1016/S0376-7388(03)00211-4.
[129] V. Gökmen, Ö.E. Aar, A. Serpen, I. Süüt, Modeling dead-end ultrafiltration of apple juice using artificial neural network, J Food Process Eng 32 (2) (Apr. 2009) 248–264, https://doi.org/10.1111/J.1745-4530.2007.00214.X.
[130] F. Schmitt, K.-U. Do, "Prediction of membrane fouling using artificial neural networks for wastewater treated by membrane bioreactor technologies: bottlenecks and possibilities," Environmental Science and Pollution Research 2017 24:29, vol. 24, no. 29, pp. 22885–22913, Sep. 2017, doi: 10.1007/S11356-017-0046-7.
[131] M. Mowbray, et al., Industrial data science-a review of machine learning applications for chemical and process industries, React Chem Eng 7 (2022) 1471, https://doi.org/10.1039/d1re00541c.
[132] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, Neurocomputing 415 (Nov. 2020) 295–316, https://doi.org/10.1016/J.NEUCOM.2020.07.061.
[133] M.K. Purkait, V.D. Kumar, D. Maity, Treatment of leather plant effluent using NF followed by RO and permeate flux prediction using artificial neural network, Chem. Eng. J. 151 (1–3) (Aug. 2009) 275–285, https://doi.org/10.1016/J.CEJ.2009.03.023.
[134] A. Abbas, N. Al-Bastaki, Modeling of an RO water desalination unit using neural networks, Chem. Eng. J. 114 (1–3) (Nov. 2005) 139–143, https://doi.org/10.1016/J.CEJ.2005.07.016.
[135] W.R. Bowen, M.G. Jones, H.N.S. Yousef, Prediction of the rate of crossflow membrane ultrafiltration of colloids: A neural network approach, Chem Eng Sci 53 (22) (Nov. 1998) 3793–3802, https://doi.org/10.1016/S0009-2509(98)00183-3.
[136] V. Yangali-Quintanilla, A. Verliefde, T.U. Kim, A. Sadmani, M. Kennedy, G. Amy, Artificial neural network models based on QSAR for predicting rejection of neutral organic compounds by polyamide nanofiltration and reverse osmosis membranes, J Memb Sci 342 (1–2) (Oct. 2009) 251–262, https://doi.org/10.1016/J.MEMSCI.2009.06.048.
[137] Y. Zhao, J.S. Taylor, S. Chellam, Predicting RO/NF water quality by modified solution diffusion model and artificial neural networks, J Memb Sci 263 (1–2) (Oct. 2005) 38–46, https://doi.org/10.1016/J.MEMSCI.2005.04.004.
[138] N. Delgrange, C. Cabassud, M. Cabassud, L. Durand-Bourlier, J.M. Lainé, Modelling of ultrafiltration fouling by neural network, Desalination 118 (1–3) (Sep. 1998) 213–227, https://doi.org/10.1016/S0011-9164(98)00132-5.
[139] N. Delgrange, C. Cabassud, M. Cabassud, L. Durand-Bourlier, J.M. Lainé, Neural networks for prediction of ultrafiltration transmembrane pressure – application to drinking water production, J Memb Sci 150 (1) (Nov. 1998) 111–123, https://doi.org/10.1016/S0376-7388(98)00217-8.
[140] Q.F. Liu, S.H. Kim, S. Lee, Prediction of microfiltration membrane fouling using artificial neural network models, Sep Purif Technol 70 (1) (Nov. 2009) 96–102, https://doi.org/10.1016/J.SEPPUR.2009.08.017.
[141] H. Chen, A.S. Kim, Prediction of permeate flux decline in crossflow membrane filtration of colloidal suspension: a radial basis function neural network approach, Desalination 192 (2006) 21–26, https://doi.org/10.1016/j.desal.2005.07.045.

[142] S. Park, S.S. Baek, J.C. Pyo, Y. Pachepsky, J. Park, K.H. Cho, Deep neural networks for modeling fouling growth and flux decline during NF/RO membrane filtration, J Memb Sci 587 (Oct. 2019), 117164, https://doi.org/10.1016/J.MEMSCI.2019.06.004.
[143] W.R. Bowen, P.M. Williams, Dynamic ultrafiltration model for proteins: A colloidal interaction approach, Biotechnol Bioeng 50 (2) (1996) 125–135.
[144] C. Niu, X. Li, R. Dai, Z. Wang, Artificial intelligence-incorporated membrane fouling prediction for membrane-based processes in the past 20 years: A critical review, Water Res 216 (Jun. 2022), 118299, https://doi.org/10.1016/J.WATRES.2022.118299.
[145] M. Dornier, M. Decloux, G. Trystram, A. Lebert, Dynamic modeling of crossflow microfiltration using neural networks, J Memb Sci 98 (3) (Jan. 1995) 263–273, https://doi.org/10.1016/0376-7388(94)00195-5.
[146] T.M. Hwang, H. Oh, Y.J. Choi, S.H. Nam, S. Lee, Y.K. Choung, Development of a statistical and mathematical hybrid model to predict membrane fouling and performance, Desalination 247 (1–3) (Oct. 2009) 210–221, https://doi.org/10.1016/J.DESAL.2008.12.025.
[147] S. Curcio, V. Calabrò, G. Iorio, Reduction and control of flux decline in cross-flow membrane processes modeled by artificial neural networks, J Memb Sci 286 (2006) 125–132, https://doi.org/10.1016/j.memsci.2006.09.024.
[148] H. Al-Zoubi, N. Hilal, N.A. Darwish, A.W. Mohammad, Rejection and modelling of sulphate and potassium salts by nanofiltration membranes: neural network and Spiegler-Kedem model, Desalination 206 (1–3) (Feb. 2007) 42–60, https://doi.org/10.1016/J.DESAL.2006.02.060.
[149] H. Niemi, A. Bulsari, S. Palosaari, Simulation of membrane separation by neural networks, J Memb Sci vol. 102, no. C (Jun. 1995) 185–191, https://doi.org/10.1016/0376-7388(94)00314-O.
[150] A. Moradi, V. Mojarradi, M. Sarcheshmehpour, Prediction of RO membrane performances by use of artificial neural network and using the parameters of a complex mathematical model, Res. Chem. Intermed. 39 (7) (Sep. 2013) 3235–3249, https://doi.org/10.1007/S11164-012-0835-Z/TABLES/1.
[151] M. Khayet, C. Cojocaru, M. Essalhi, Artificial neural network modeling and response surface methodology of desalination by reverse osmosis, J Memb Sci 368 (1–2) (Feb. 2011) 202–214, https://doi.org/10.1016/J.MEMSCI.2010.11.030.
[152] E. Piron, E. Latrille, F. René, Application of artificial neural networks for crossflow microfiltration modelling: 'black-box' and semi-physical approaches, Comput Chem Eng 21 (9) (Jun. 1997) 1021–1030, https://doi.org/10.1016/S0098-1354(96)00332-8.
[153] J. Ní Mhurchú, G. Foley, Dead-end filtration of yeast suspensions: Correlating specific resistance and flux data using artificial neural networks, J Memb Sci 281 (1–2) (Sep. 2006) 325–333, https://doi.org/10.1016/J.MEMSCI.2006.03.043.
[154] W.R. Bowen, M.G. Jones, H.N.S. Yousef, Dynamic ultrafiltration of proteins – A neural network approach, J Memb Sci 146 (2) (Aug. 1998) 225–235, https://doi.org/10.1016/S0376-7388(98)00108-2.
[155] M.A. Razavi, A. Mortazavi, M. Mousavi, Application of neural networks for crossflow milk ultrafiltration simulation, Int Dairy J 14 (1) (Jan. 2004) 69–80, https://doi.org/10.1016/S0958-6946(03)00150-X.
[156] P. Rai, G.C. Majumdar, S. DasGupta, S. De, Modeling the performance of batch ultrafiltration of synthetic fruit juice and mosambi juice using artificial neural network, J Food Eng 71 (3) (Dec. 2005) 273–281, https://doi.org/10.1016/J.JFOODENG.2005.02.003.
[157] Q.F. Liu, S.H. Kim, Evaluation of membrane fouling models based on bench-scale experiments: A comparison between constant flowrate blocking laws and artificial neural network (ANNs) model, J Memb Sci 310 (1–2) (Mar. 2008) 393–401, https://doi.org/10.1016/J.MEMSCI.2007.11.020.
[158] N.A. Darwish, N. Hilal, H. Al-Zoubi, A.W. Mohammad, Neural Networks Simulation of the Filtration of Sodium Chloride and Magnesium Chloride Solutions Using Nanofiltration Membranes, Chem. Eng. Res. Des. 85 (4) (Jan. 2007) 417–430, https://doi.org/10.1205/CHERD06037.
[159] J.C. Chen, A. Seidel, Cost Optimization of Nanofiltration with Fouling by Natural Organic Matter, J. Environ. Eng. 128 (10) (Oct. 2002) 967–973, https://doi.org/10.1061/(ASCE)0733-9372(2002)128:10(967).
[160] L. Khaouane, Y. Ammi, S. Hanini, Modeling the Retention of Organic Compounds by Nanofiltration and Reverse Osmosis Membranes Using Bootstrap Aggregated Neural Networks, Arab J Sci Eng 42 (4) (Apr. 2017) 1443–1453, https://doi.org/10.1007/S13369-016-2320-2/METRICS.
[161] Y.G. Lee, et al., Artificial neural network model for optimizing operation of a seawater reverse osmosis desalination plant, Desalination 247 (1–3) (Oct. 2009) 180–189, https://doi.org/10.1016/J.DESAL.2008.12.023.
[162] A.M. Aish, H.A. Zaqoot, S.M. Abdeljawad, Artificial neural network approach for predicting reverse osmosis desalination plants performance in the Gaza Strip, Desalination 367 (Jul. 2015) 240–247, https://doi.org/10.1016/J.DESAL.2015.04.008.
[163] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, "Physics-informed machine learning," Nature Reviews Physics, vol. 3, no. 6. Springer Nature, pp. 422–440, Jun. 01, 2021. doi: 10.1038/s42254-021-00314-5.
[164] W.E. Siew, A.G. Livingston, C. Ates, A. Merschaert, Continuous solute fractionation with membrane cascades - A high productivity alternative to diafiltration, Sep Purif Technol 102 (2013) 1–14, https://doi.org/10.1016/j.seppur.2012.09.017.
[165] D. Bocciardo, M.C. Ferrari, S. Brandani, Modelling and Multi-stage Design of Membrane Processes Applied to Carbon Capture in Coal-fired Power Plants, Energy Procedia 37 (Jan. 2013) 932–940, https://doi.org/10.1016/J.EGYPRO.2013.05.188.
[166] D. Bocciardo, M.-C. Ferrari, S. Brandani, Modelling and multi-stage design of membrane processes applied to carbon capture in coal-fired power plants Selection and/or peer-review under responsibility of GHGT, Energy Procedia 37 (2013) 932–940, https://doi.org/10.1016/j.egypro.2013.05.188.
[167] U. Diwekar, Introduction to Applied Optimization, Second Edi. (1984), https://doi.org/10.1007/978-3-662-00719-8.
[168] D. Gao, N. Ruan, and W. Xing, *Advances in Global Optimization*, vol. 95. 2015. doi: 10.1007/978-3-319-08377-3_41.
[169] C.A. Floudas, C.E. Gounaris, A review of recent advances in global optimization, J. Glob. Optim. 45 (1) (2009) 3–38, https://doi.org/10.1007/s10898-008-9332-8.
[170] D.G. Luenberger, J. Zhu, Linear and nonlinear programming 116 (2008), https://doi.org/10.1201/b17264-17.
[171] G.H. Golub, U. von Matt, Quadratically constrained least squares and quadratic problems, Numer Math (Heidelb) 59 (1) (1991) 561–580, https://doi.org/10.1007/BF01385796.
[172] P. Feautrier, Parametric integer programming, RAIRO - Operations Research 22 (3) (1988) 243–268, https://doi.org/10.1051/ro/1988220302431.
[173] L. Connectives, I.P. Constraints, Logic and Integer Programming 130 (2009) [Online]. Available:.
[174] M. Tawarmalani and N. v. Sahinidis, "Global optimization of mixed-integer nonlinear programs: A theoretical and computational study," Math Program, vol. 99, no. 3, pp. 563–591, 2004, doi: 10.1007/s10107-003-0467-6.
[175] D. Vázquez, R. Ruiz-Femenia, J.A. Caballero, Mixed integer non-linear programming model for reliable and safer design at an early stage, Comput Chem Eng 147 (Apr. 2021), 107256, https://doi.org/10.1016/j.compchemeng.2021.107256.
[176] B.N. Pshenichny, The linearization method, Optimization 18 (2) (Jan. 2007) 179–196, https://doi.org/10.1080/02331938708843231.
[177] Y. Bengio, Gradient-Based Optimization of Hyperparameters, Neural Comput 12 (8) (Aug. 2000) 1889–1900, https://doi.org/10.1162/089976600300015187.
[178] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, Sep. 2016, doi: 10.48550/arXiv.1609.04747.
[179] J. Fliege, L.M.G. Drummond, B.F. Svaiter, Newton's method for multiobjective optimization, SIAM J. Optim. 20 (2) (May 2009) 602–626, https://doi.org/10.1137/08071692X.
[180] A. Fischer, A special newton-type optimization method, Optimization 24 (3–4) (Jan. 1992) 269–284, https://doi.org/10.1080/02331939208843795.
[181] P.E. Gill, W. Murray, Quasi-Newton Methods for Unconstrained Optimization, IMA J Appl Math 9 (1) (Feb. 1972) 91–108, https://doi.org/10.1093/imamat/9.1.91.
[182] F. Boukouvala, R. Misener, C. A. Floudas, "Global optimization advances in Mixed-Integer Nonlinear Programming, MINLP, and Constrained Derivative-Free Optimization, CDFO," European Journal of Operational Research, vol. 252, no. 3. Elsevier B.V., pp. 701–727, Aug. 01, 2016. doi: 10.1016/j.ejor.2015.12.018.
[183] D. Fouskakis, D. Draper, Stochastic optimization: A review, Int. Stat. Rev. 70 (3) (2002) 315–349, https://doi.org/10.1111/j.1751-5823.2002.tb00174.x.
[184] L.M. Rios, N.v. Sahinidis, Derivative-free optimization: A review of algorithms and comparison of software implementations, J. Global Optimization 56 (3) (Jul. 2013) 1247–1293, https://doi.org/10.1007/s10898-012-9951-y.
[185] F. Lipnizki, J. Boelsmand, R.F. Madsen, Concepts of industrial-scale diafiltration systems, Desalination 144 (1–3) (Sep. 2002) 179–184, https://doi.org/10.1016/S0011-9164(02)00309-0.
[186] M.G. Jabra, C.J. Yehl, A.L. Zydney, Multistage continuous countercurrent diafiltration for formulation of monoclonal antibodies, Biotechnol Prog 35 (4) (Jul. 2019) e2810.
[187] G. Daufin, J.P. Escudier, H. Carrère, S. Bérot, L. Fillaudeau, M. Decloux, Recent and emerging applications of membrane processes in the food and dairy industry, Food Bioprod. Proces: Trans Instit. Chem. Eng., Part C 79 (2) (2001) 89–102, https://doi.org/10.1205/096030801750286131.
[188] G. Paun, E. Neagu, A. Tache, G. L. Radu, V. Purvulescu, "Application of the Nanofiltration Process for Concentration of Polyphenolic Compounds from Geranium robertianum and Salvia officinalis Extracts," 2011, [Online]. Available: https://www.researchgate.net/publication/267406907.
[189] R. Tundis, et al., Concentration of Bioactive Compounds from Elderberry (Sambucus nigra L.) Juice by Nanofiltration Membranes, Plant Foods Hum. Nutr. 73 (4) (Dec. 2018) 336–343, https://doi.org/10.1007/S11130-018-0686-X/TABLES/4.
[190] R. Castro-Muñoz, E. Díaz-Montes, A. Cassano, E. Gontarek, Membrane separation processes for the extraction and purification of steviol glycosides: an overview 61 (13) (2020) 2152–2174, https://doi.org/10.1080/10408398.2020.1772717.
[191] P. Blanc, G. Goma, Propionic acid and biomass production using continuous ultrafiltration fermentation of whey, Biotechnology Letters 1989 11:3, vol. 11, no. 3, pp. 189–194, Mar. 1989, doi: 10.1007/BF01026057.
[192] W. Tomczak, M. Gryta, The application of ultrafiltration for separation of glycerol solution fermented by bacteria, Pol. J. Chem. Technol. 15 (3) (Sep. 2013) 115–120, https://doi.org/10.2478/PJCT-2013-0057.
[193] S. Overmans, et al., Continuous extraction and concentration of secreted metabolites from engineered microbes using membrane technology, Green Chem. 24 (14) (Jul. 2022) 5479–5489, https://doi.org/10.1039/D2GC00938B.
[194] Y. Li, A. Shahbazi, C.T. Kadzere, Separation of cells and proteins from fermentation broth using ultrafiltration, J Food Eng 75 (4) (Aug. 2006) 574–580, https://doi.org/10.1016/J.JFOODENG.2005.04.045.