Systems & Control Transactions

Research Article
Foundations of Computer Aided Process Design (FOCAPD 2024)
Breckenridge, Colorado, USA. July 14-18, 2024
Peer Reviewed Conference Proceeding

PSE PRESS

From Then to Now and Beyond: Exploring How Machine Learning Shapes Process Design Problems

Burcu Beykal[a,b]*

[a] Department of Chemical & Biomolecular Engineering, University of Connecticut, Storrs, CT, USA
[b] Center for Clean Energy Engineering, University of Connecticut, Storrs, CT, USA
* Corresponding Author: beykal@uconn.edu

ABSTRACT

Following the discovery of the least squares method in 1805 by Legendre and later in 1809 by Gauss, surrogate modeling and machine learning have come a long way. From identifying patterns and trends in process data to predictive modeling, optimization, fault detection, reaction network discovery, and process operations, machine learning became an integral part of all aspects of process design and process systems engineering. This is enabled, at the same time necessitated, by the vast amounts of data that are readily available from processes, increased digitalization, automation, increasing computation power, and simulation software that can model complex phenomena that span over several temporal and spatial scales. Although this paper is not a comprehensive review, it gives an overview of the recent history of machine learning models that we use every day and how they shaped process design problems from the recent advances to the exploration of their prospects.

Keywords: Surrogate modeling, Artificial Intelligence, Historical view, Data-driven analysis, Process synthesis

A BRIEF HISTORY OF MACHINE LEARNING

The roots of machine learning (ML) can be traced back to the early 19th century when the method of least squares was first discovered by Legendre in 1805 and later by Gauss in 1809 [1]. However, the main concept of computers learning from experience without explicitly being programmed has roots tracing back to more recent history, the mid-20th century.

The foundational idea of neural networks emerged in the 1940s and 1950s when researchers began exploring mathematical models inspired by the structure and functioning of the human brain. Warren McCulloch and Walter Pitts' paper, "A Logical Calculus of Ideas Immanent in Nervous Activity," was published in 1943, where they proposed a mathematical model of an artificial neuron which was the first idea of using a computational model for neural networks [2]. This foundational paper laid the groundwork for subsequent developments in neural network theory. The term "neural network" itself gained prominence in the 1950s and 1960s as researchers like Frank Rosenblatt developed the perceptron, an early form of a neural network designed for pattern recognition tasks. While the perceptron had limitations, the idea of using computational models to simulate neural processes became a cornerstone in the evolution of artificial neural networks (ANNs) and ML. Around the same timeline, response surface methodology was introduced by Box and Wilson [3], and the term “machine learning” was coined by Arthur Samuel [4].

Throughout the following decades, various modeling approaches and algorithms, including Gaussian process (GP) regression, backpropagation algorithm, support vector machines (SVMs), and Random Forest (RF), emerged in the ML landscape. Especially, the establishment of the backpropagation algorithm was a pivotal moment in the resurgence and widespread adoption of neural works starting 1980s, enabling researchers to revisit the complex problems that were not possible to address before. This ultimately led to the application of neural networks in various scientific and engineering domains, including process design and operations, and to the development of more complex algorithms. These include reinforcement learning, deep learning, natural language processing, and generative artificial intelligence (AI)

which are emerging areas of research within process systems engineering [5,6]. Although NN models have been the primary modeling strategy in process design problems due to their ability to capture nonlinearities very accurately, we will also demonstrate that SVMs and tree-based ensemble models like RF and gradient boosted trees are also studied in depth.

PAST APPLICATIONS OF ML IN PROCESS DESIGN

With these developments underway, it was also imperative to revisit optimal design problems from the lens of ML models and process synthesis. Regression analysis and parameter estimation for kinetic or thermodynamic models were already performed for process design, dating back to the 1960s. However, with the increasing computation power and the development of process simulation software, optimal design problems recognized the need for surrogate ML models due to: (1) the "black-box" nature of the simulation software that lacks the derivative information that is imperative for optimization; (2) the computational expense associated with sample-based derivative-free optimization techniques; and (3) the high mathematical complexity of process synthesis problems (mixed-integer nonlinear program – MINLP) that become intractable with high number of variables and constraints. Hence, earlier introduction of ML techniques in process design focused on replacing highly complex and/or noisy simulations with relatively simpler representations, especially within optimization frameworks to alleviate the mathematical complexity.

For example, Caballero and Grossmann used kriging surrogate models to replace noisy unit operations in modular flowsheet optimization [7]. Likewise, Davis and Ierapetritou used kriging surrogates for tertbutyl methacrylate production design and process synthesis [8]. Henao and Maravelias used ANN surrogate models trained using the data collected from the process simulator to replace complex unit operations (*e.g.,* distillation column, expansion valves, heaters/coolers, flash vessels, absorption columns) and reformulated these ANNs to incorporate within their superstructure optimization framework [9]. Fahmi and Cremaschi also used ANNs to substitute for thermodynamics and mixing models, as well as unit operations for process synthesis of biodiesel production [10]. One of the key challenges using ANNs was also noted in this work, where these models were "data hungry" (*i.e.,* large amounts of data were required to train accurate ANN representations). The modeling complexity of the ANNs also made it challenging to incorporate them in large-scale optimization problems without any efficient reformulation strategies. Equation 1 shows the mathematical structure of a general feed-forward NN, represented by a repeated composition of functions,

$$y = \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}) = f_L \circ f_{L-1} \circ \ldots \circ f_2 \circ f_1(\boldsymbol{x};\boldsymbol{\theta}) \quad (1)$$

where y is the output of the network, $\boldsymbol{x}$ are the inputs to the network, f_i is a layer in the neural network with transformations applied by the activation functions, and $\boldsymbol{\theta}$ are the weights and biases for the entire network. In simpler terms, this mathematical structure generates highly nonlinear expressions (except for purely linear activation functions) that create additional complexities for optimization algorithms to handle (Equation 2).

$$y = f_L(f_{L-1}\,(f_{L-2}(\ldots . f_1(\boldsymbol{x})))) \quad (2)$$

Especially, within a global optimization framework, this nested functional form can be intractable as well. Motivated by this, most recent progress focused on using more simplified surrogate models for process design and synthesis problems, as well as developing novel reformulation strategies that exploit the mathematical properties of activation functions. Next, we discuss these developments and other key progress in this area.

CURRENT PROGRESS

Reformulation of ML Models

One of the most recent key breakthroughs in using ML in any optimization framework (*e.g.,* process design, synthesis, or operations) is the ability to reformulate deep NNs with rectified linear unit (ReLU) activation functions into a mixed-integer linear program (MILP) [11-13]. By recognizing ReLU activation functions as max-affine spline operators that are piecewise linear (Equation 3), ANNs can be exactly reformulated with big-M constraints to create a MILP that can be solved to global optimality with off-the-shelf solvers.

$$y = \max\{0, z\} = \begin{cases} 0 & if\ z < 0 \\ z & otherwise \end{cases} \quad (3)$$

This enabled MILP-reformulated ANNs to be embedded in a variety of problems, including optimizing the extractive distillation process [14], sustainable hydrogen production using sorption enhanced steam methane reforming [15], as well as for modeling flexibility index constraints in biorefinery design by superstructure optimization [16]. This technique is also extended to other activation functions that are nonlinear [17,18], and different modeling strategies such as tree-based ML models, as they also partition the modeling space with piecewise linear models (Figure 1).

Mišić [19] and Mistry *et al.* [20] encoded trained gradient boosted regression trees, which are ensemble decision tree models), to MILP models that are later embedded into optimization problems. This technique is further extended as a black-box optimization algorithm in the ENTMOOT framework [21] and implemented as an open-source software package named OMLT [22]. The applicability of the tree-based reformulation is also

demonstrated on an optimal layout design problem of an offshore windfarm [23].

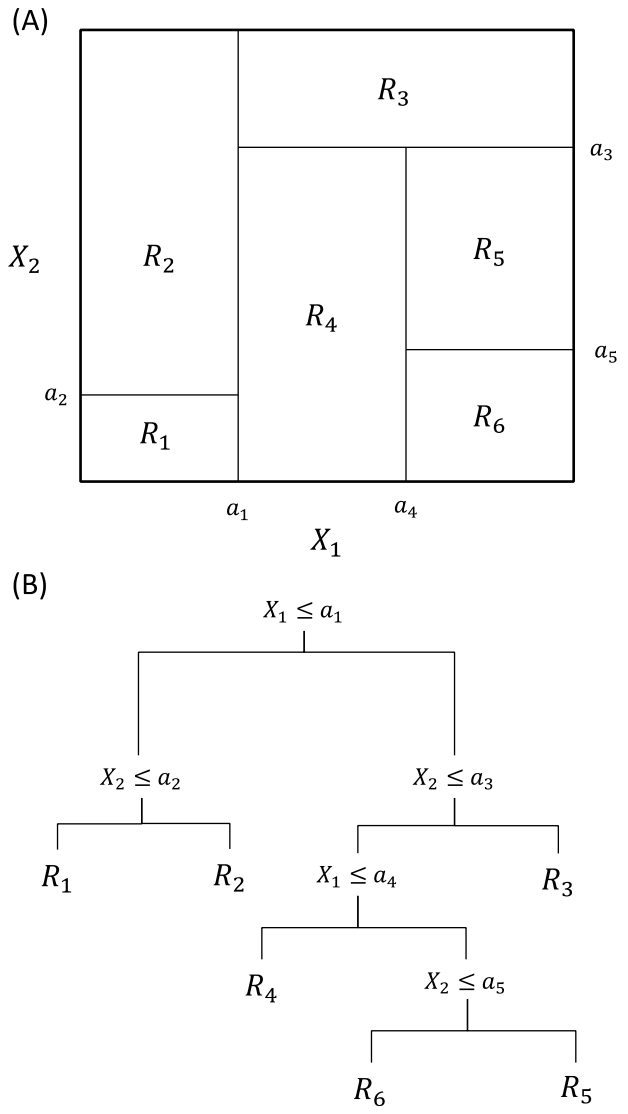

Figure 1. A demonstration of how decision tree models (A) partition the space with piecewise linear models; and (B) map this partitioning onto decision trees for a visual representation.

Despite reformulation strategies alleviating a portion of the nonlinearity issues in NN and tree-based ensemble models, we also observe that large-scale process synthesis problems still rely on more simplistic models. For instance, Demirhan *et al.* used linear surrogate models to model the conversion of a Haber-Bosch reactor within the renewable ammonia process synthesis problem that has 18,573 continuous, 38 binary variables, and 18,924 constraints [24]. Under such large-scale global optimization problems, reformulating MILP representations of NN or tree-based models of individual units will amplify the number of binary variables, which will further increase the complexity of the overall optimization model. Hence, the use of ML in large-scale synthesis problems is still contingent on the overall problem complexity, even when ML models offer highly accurate predictions.

ML Algorithms as Constraints

Nowadays, the use of ML is not limited to modeling individual unit operations or an entire flowsheet, but it can also serve as constraints to process design problems. Especially, process simulations are typically subject to black-box constraints that lack explicit analytical expressions relating the decision variables to constraint violations (*i.e.,* the constraint violations can only be obtained once the simulation run is completed). Constraint handling can be achieved in many ways, including augmented Lagrangian formulations, penalty, filter, or barrier methods [25].

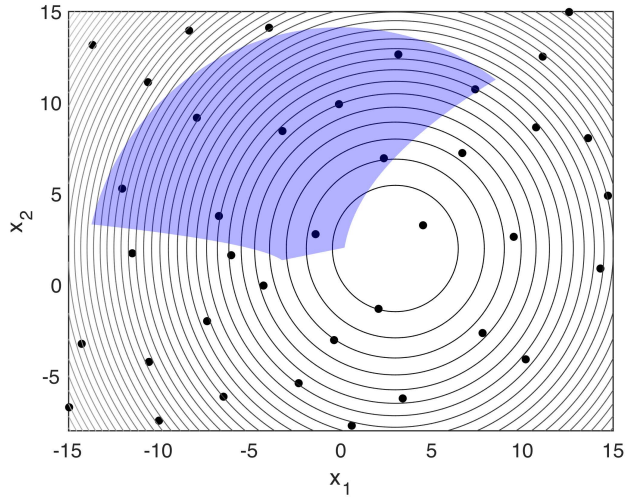

Figure 2. The feasible region derived by fitted surrogates using regression analysis (purple) is shown on a contour plot of the objective function. Black dots show the sampling points for the input space. The constraints are: $x^2 + y^2 - 200 \leq 0$; $x - 5y + 10 \leq 0$; $25x - 2y^2 + 4y - 5 \leq 0$; $-x - 2y^2 + 4y - 5 \leq 0$.

On the other hand, ML tools can be leveraged to handle constraints individually as a regression task [26,27], where constraint violations are modeled as less than or equal to constraints with surrogate models (Figure 2), or holistically as a classification task [28-30], where a separating model between feasible and infeasible solutions are established. A conceptual demonstration of a nonlinear constraint being modeled as a classifier using SVM with Gaussian radial basis function kernel is provided in Figure 3.

This is achieved by using a dataset of simulated samples with their binary outcome (feasible/infeasible) to train one or more classification models instead of

individually modeling actual violation values as done in the regression analysis. The trained classifier can then be used with a data-driven optimizer or implemented in a superstructure model using the aforementioned reformulation strategies. In that respect, SVM classifiers are shown to effectively model the implicit constraints of numerically infeasible differential algebraic equations of a steam cracker reactor design problem [28,29]. SVM classifiers are also used for modeling feasibility constraints in the vertex formulation of modular design problems [30].

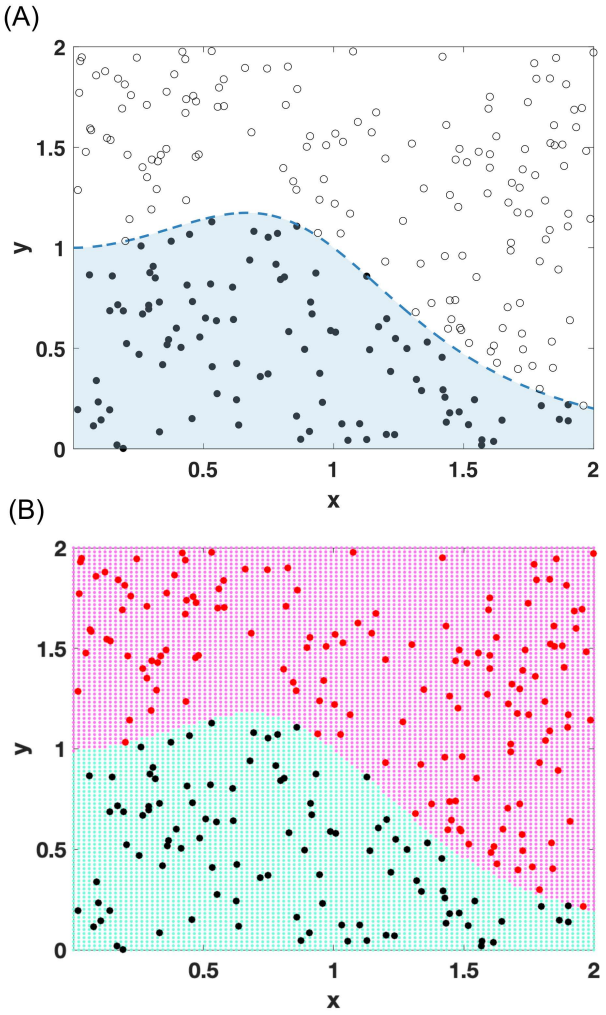

Figure 3. An SVM-based classifier trained to mimic the nonlinear constraint, $y < 1/(x^3 - x^2 + 1)$: (A) The original constraint within the bounded space; (B) The map of the feasible region captured by SVM. The predictive performance of the classifier on a blind testing set: Accuracy = 100%; Sensitivity = 100%; Specificity = 100%; F_1 score = 100%.

While these studies show promise for using classifiers, the offline model training is still time-consuming (*i.e.,* several thousand samples are collected from the simulator to create the model) with no efficient way to recycle or integrate the already collected data into the decision-making process. Also, understanding the uncertainties surrounding these models as well as their misclassification rate is of utmost importance for constraint modeling, as misclassifications can lead to infeasible solutions or designs, whereas in regression-based models, such misviolations are less likely to happen.

Large Language Models & Generative AI for Process Design

Choosing the right sequence of unit operations and connections to create a flowsheet is a fundamental practice in process design, whether it is done heuristically or through superstructure optimization. With the launch of ChatGPT, the key question becomes whether natural language processing or generative techniques can be utilized for process design and discovery. As a language model, ChatGPT is designed to understand and generate human-like text based on the input it receives. However, chemical engineering problems, such as process flowsheet generation, are based on recognizing the sequence of unit operations. This comes with the caveat of lack of suitable data to be able to train the large language models that can generate a flowsheet automatically [31]. For instance, there could be 20+ different representations for the same unit [32] or process flowsheets are most likely proprietary or unavailable to extract the information necessary for the AI model development [33].

To overcome these limitations, Vogel *et al.* developed SFILES 2.0 [33] to represent flowsheets using a graph notation, analogically similar to the text-based SMILES notation for representing chemical structures. This was primarily developed to topologically describe a flowsheet with the disadvantage of not storing any information about the sizing or the operating conditions of the units. Along the idea of ChatGPT, Vogel *et al.* also investigated the automatic completion of flowsheets using causal language modeling [34]. Their results show that the generative AI model can learn the topological patterns in flowsheet data and can automatically complete flowsheets. However, like the issues faced in ChatGPT, the generated flowsheet may not make practical sense, as the developed model does not intake contextual information about the process. Using the SFILES notation, Hirtreiter *et al.* also investigated the automatic generation of control structures for flowsheets [35]. While the predictive accuracy of the trained models was relatively high, significant limitations are also noted by the authors. Especially, concerns regarding safety indicators, understanding the process dynamics and operational objectives, and lack of information on the equipment sizing and operating conditions for the units pose major questions. These promising developments show that natural language processing and large language models can provide a "warm start" for flowsheet generation and facilitate

some of the time-consuming tasks. However, more research needs to be done in this area to be able to improve confidence in model predictions while providing a holistic view of process design beyond just the topological investigation.

CONCLUSIONS

Artificial Intelligence and machine learning (ML) models are now an essential component of process design with efficient model integration and reformulation strategies paving the way. Constraint handling with ML models to flowsheet generation using large language models, we see new and innovative ways of how ML is used for process design problems. While the main motivation for using ML models is to alleviate the model complexities and such models have proven to be successful over the course of decades, domain knowledge and model interpretability will still play the most important role despite the promise these models hold. The ability to understand why a model makes a particular prediction and to reason if predicted results are physically sound or whether a generative model-derived process flow diagram is safe to implement becomes critically important. This judgment requires a deep fundamental understanding of the process, engineering expertise, and other relevant domain knowledge. Incorporating safety and risk measures, and combining ML models with first-principles information to create hybrid models are a few avenues that researchers are currently investigating. Nevertheless, the importance of interpretability and understanding the process relevance of the predictions will persist in this field.

ACKNOWLEDGEMENTS

This work was supported by the National Institutes of Health [NIH P42 ES027704] and University of Connecticut. Portions of this research were conducted with the advanced computing resources provided by the University of Connecticut Storrs High Performance Computing facility. The manuscript contents are solely the responsibility of the grantee and do not necessarily represent the official views of the NIH. Further, NIH does not endorse the purchase of any commercial products or services mentioned in the publication.

REFERENCES

1. Stigler SM. Gauss and the invention of least squares. *Ann Stat* 9(3):465-474 (1981)
2. McCulloch WS, Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115-133 (1943)
3. Box GEP, Wilson KB. On the Experimental Attainment of Optimum Conditions. *J R Stat Soc Series B Stat Methodol* 13(1):1-45 (1951)
4. Samuel AL. Some studies in machine learning using the game of checkers. *IBM J Res Dev* 3(3):210-229 (1959)
5. Daoutidis P, Lee JH, Rangarajan S, Chiang L, Gopaluni B, Schweidtmann AM, Harjunkoski I, Mercangöz M, Mesbah A, Boukouvala F, Lima FV. Machine learning in process systems engineering: Challenges and opportunities. *Comput Chem Eng* 181:108523 (2024)
6. Schweidtmann AM, Esche E, Fischer A, Kloft M, Repke JU, Sager S, Mitsos A. Machine learning in chemical engineering: A perspective. *Chem Ing Tech* 93(12):2029-2039 (2021)
7. Caballero JA, Grossmann IE. An algorithm for the use of surrogate models in modular flowsheet optimization. *AIChE J* 54(10):2633-2650 (2008)
8. Davis E, Ierapetritou M. A kriging-based approach to MINLP containing black-box models and noise. *Ind Eng Chem Res* 47(16):6101-6125 (2008)
9. Henao CA, Maravelias CT. Surrogate-based superstructure optimization framework. *AIChE J* 57(5):1216-1232 (2011)
10. Fahmi I, Cremaschi S. Process synthesis of biodiesel production plant using artificial neural networks as the surrogate models. *Comp Chem Eng* 46:105-123 (2012)
11. Lomuscio A, Maganti L. An approach to reachability analysis for feed-forward relu neural networks. *arXiv preprint* arXiv:1706.07351 (2017)
12. Fischetti M, Jo J. Deep neural networks and mixed integer linear optimization. *Constr* 23(3):296-309 (2018)
13. Grimstad B, Andersson H. ReLU networks as surrogate models in mixed-integer linear programs. *Comp Chem Eng* 131:10658 (2019)
14. Ma K, Sahinidis NV, Bindlish R, Bury SJ, Haghpanah R, Rajagopalan S. Data-driven strategies for extractive distillation unit optimization. *Comp Chem Eng* 167:107970 (2022)
15. Arora A, Zantye MS, Hasan MF. Sustainable hydrogen manufacturing via renewable-integrated intensified process for refueling stations. *Appl Energy* 311:118667 (2022).
16. Luo Y, Ierapetritou M. Multifeedstock and multiproduct process design using neural network surrogate flexibility constraints. *Ind Eng Chem Res* 62(5):2067-2079 (2023)
17. Schweidtmann AM, Mitsos A. Deterministic global optimization with artificial neural networks embedded. *J Optim Theory Appl* 180(3):925-948 (2019)
18. Wilhelm ME, Wang C, Stuber MD. Convex and concave envelopes of artificial neural network

activation functions for deterministic global optimization. *J Glob Optim* 85(3):569-594 (2023)
19. Mišić VV. Optimization of tree ensembles. *Oper Res* 68(5):1605-1624 (2020)
20. Mistry M, Letsios D, Krennrich G, Lee RM, Misener R. Mixed-integer convex nonlinear optimization with gradient-boosted trees embedded. *INFORMS J Comput* 33(3):1103-1119 (2021)
21. Thebelt A, Kronqvist J, Mistry M, Lee RM, Sudermann-Merx N, Misener R. ENTMOOT: a framework for optimization over ensemble tree models. *Comp Chem Eng* 151:107343 (2021)
22. Ceccon F, Jalving J, Haddad J, Thebelt A, Tsay C, Laird CD, Misener R. OMLT: Optimization & machine learning toolkit. *J Mach Learn Res* 23(1):15829-15836 (2022)
23. Thebelt A, Tsay C, Lee RM, Sudermann-Merx N, Walz D, Tranter T, Misener R. Multi-objective constrained optimization for energy applications via tree ensembles. *Appl Energy* 306:118061 (2022)
24. Demirhan CD, Tso WW, Powell JB, Pistikopoulos EN. Sustainable ammonia production through process synthesis and global optimization. *AIChE J* 65(7):e16498 (2019)
25. Beykal B, Pistikopoulos EN. Data-Driven Optimization Algorithms. In: Artificial Intelligence in Manufacturing. Ed: Soroush M, Braatz RD. Elsevier (2024)
26. Beykal B, Boukouvala F, Floudas CA, Sorek N, Zalavadia H, Gildin E. Global optimization of grey-box computational systems using surrogate functions and application to highly constrained oil-field operations. *Comp Chem Eng* 114:99-110 (2018)
27. Beykal B, Boukouvala F, Floudas CA, Pistikopoulos EN. Optimal design of energy systems using constrained grey-box multi-objective optimization. *Comp Chem Eng* 116:488-502 (2018)
28. Beykal B, Onel M, Onel O, Pistikopoulos EN. A data-driven optimization algorithm for differential algebraic equations with numerical infeasibilities. *AIChE J* 66(10):e16657 (2020)
29. Beykal B, Aghayev Z, Onel O, Onel M, Pistikopoulos EN. Data-driven stochastic optimization of numerically infeasible differential algebraic equations: an application to the steam cracking process. *Comput Aided Chem Eng* 49:1579-1584 (2022)
30. Bhosekar A, Ierapetritou M. Modular design optimization using machine learning-based flexibility analysis. *J Process Control*, 90:18-34 (2020)
31. Xuan J, Daniel T. The Future of Chemical Engineering in the Era of Generative AI. The Chemical Engineer. https://www.thechemicalengineer.com/features/the-future-of-chemical-engineering-in-the-era-of-generative-ai/
32. Theisen MF, Flores KN, Balhorn LS, Schweidtmann AM. Digitization of chemical process flow diagrams using deep convolutional neural networks. *Digit Chem Eng* 6:100072 (2023)
33. Vogel G, Hirtreiter E, Balhorn LS, Schweidtmann AM. SFILES 2.0: an extended text-based flowsheet representation. *Optim Eng* 1-23 (2023)
34. Vogel G, Balhorn LS, Schweidtmann AM. Learning from flowsheets: A generative transformer model for autocompletion of flowsheets. *Comp Chem Eng* 171:108162 (2023)
35. Hirtreiter E, Schulze Balhorn L, Schweidtmann AM. Toward automatic generation of control structures for process flow diagrams with large language models. *AIChE J* e18259 (2023)