ChE data science special section

TEACHING ARTIFICIAL INTELLIGENCE TO CHEMICAL ENGINEERS: EXPERIENCE FROM A 35-YEAR-OLD COURSE

Venkat Venkatasubramanian
Columbia University • New York, NY 10027

ARTIFICIAL INTELLIGENCE AS A KNOWLEDGE MODELING PARADIGM

The dramatic success of machine learning (ML) in applications such as computer vision, game playing, and natural language processing has many chemical engineers excited about the potential opportunities in our domain. While it might seem that the prospect of developing AI-based solutions is brand new, the use of AI in chemical engineering is not that novel. It is, in fact, a 40-year-old ongoing effort with several thousand papers in the literature.[1]

To properly appreciate how and where AI fits into chemical engineering, we need to view it from the perspective of different knowledge modeling paradigms used in chemical engineering. Historically speaking, chemical engineering was primarily an empirical and heuristic discipline, lacking quantitative first principles-based modeling approaches for nearly a century. All that changed with the beginning of the Amundson era in the 1950s, when *applied mathematical* methods, particularly linear algebra, ordinary differential equations (ODEs), and partial differential equations (PDEs) were introduced to develop first-principles based models of *unit operations*.

Similarly, *decision-making* in process systems engineering was also largely empirical and heuristics based. That changed in the 1960s with the next critical turning point in modeling — the development of *mathematical programming* methods such as mixed integer linear programming (MILP) and mixed integer nonlinear programming (MINLP). Roger Sargent of Imperial College and his disciples played the leading role in this era.

The next significant development in this long arc of modeling paradigms, in my view, is the introduction of knowledge representation concepts and search techniques from *artificial intelligence*. This started in the early 1980s under the leadership of Westerberg, Stephanopoulos, and others from that era. After remaining in the background as a fringe activity for the past three decades, pursued by only a few researchers, this knowledge-modeling paradigm has now gone mainstream.

Broadly speaking, one might consider the Amundson era as the introduction of *formal* methods for modeling *process units*. The Sargent era and the AI era are about modeling the *process engineer* – that is, modeling and *automating* human information-processing and decision-making, *formally*, to solve problems in synthesis, design, control, scheduling, optimization, and risk analysis. Some of these could be addressed by the mathematical programming framework, i.e., the Sargent approach, but others, such as fault diagnosis and process hazards analysis, require causal model-based reasoning and are better addressed by AI concepts and techniques.

Role of *Symbolic AI* in Knowledge Modeling and Representation

When it comes to modeling knowledge, most chemical engineers immediately think of differential and algebraic equations (i.e., DAE models). These are suitable for certain classes of problems like those found in thermodynamics, transport phenomena, and reaction engineering.

Venkat Venkatasubramanian *is the Samuel Ruben-Peter G. Viele Professor of Engineering in the Department of Chemical Engineering at Columbia University. He considers himself an artist in science, whose natural tendency is to conduct curiosity-driven research in a style that might be considered impressionistic, emphasizing conceptual issues over mere techniques. Venkat's research interests are diverse, ranging from AI to systems engineering to theoretical physics to economics, but with a focus on understanding complexity and emergent behavior in different domains.*

However, there are other kinds of knowledge that do not lend themselves to such models. For example, reasoning about cause and effect in a process plant is central to fault diagnosis, risk analysis, alarm management, and supervisory control. Knowledge modeling for this problem class does not typically lend itself to the traditional DAE approach to modeling because it cannot deliver explicit relationships between cause(s) and effect(s). In some simple cases perhaps it can, but it is incapable of addressing real-life industrial process systems, which are often complex and nonlinear with incomplete and/or uncertain data. Further, even for simple systems, DAE-based models are not suitable for generating mechanistic explanations about causal behavior. This is where *symbolic AI* comes in.

We should not forget that the *conceptual breakthrough* of representing, and reasoning with, *symbolic* structures and relationships is an essential contribution of AI.[2-5] This is what I refer to as *symbolic* AI, the *classical* AI of 1960s-1980s, to differentiate it from data-driven machine learning, which I call *numeric* AI. Although the importance of symbolic AI has largely been missed in all the current excitement about data-driven machine learning, I expect it to resurface as we go beyond purely data-driven models towards more *comprehensive knowledge-based intelligent systems* that are *necessary* for many applications in chemical engineering.

The symbolic AI methodologies include models such as the following:

- Graph-theoretical models such as signed digraphs used extensively to perform causal reasoning in the identification of abnormal events, diagnosis, and risk analysis [6-9]
- Petri nets used for modeling discrete event systems [10-11]
- Rule-based production system models used in expert systems for automating higher-order reasoning [12-17]
- Semantic network models such as ontologies used in materials discovery and design, domain-specific compilers, etc. [12-22]
- Object-oriented models such as agent-based models used in simulating the behavior and decision-making choices of independent, interacting, entities endowed with complex attributes and decision-making powers [23, 24]

All these have far-reaching consequences as we begin to develop more comprehensive hybrid-AI systems such as the following in the near future:

- Combining first-principles with data-driven processing [25-27]
- Causal models-based explanatory systems [7, 16, 17, 28-31]
- Domain-specific knowledge engines [19, 32-36]

Thus, I do not view AI methods as merely useful tools to extract patterns from large amounts of data, even though that benefit is very much there, but as a *new knowledge modeling paradigm*, the next natural evolutionary stage in the long history of developing formal methods – first applied math (i.e., differential and algebraic equations), then operations research (i.e., math programming), and now artificial intelligence. Conceptually, applied math models *numerical* relationships between variables and parameters, mathematical programming models relationships between *constraints*, and AI models relationships between *symbolic* variables and *symbolic* structures. In the early years, logic was considered the discipline best suited to provide the formal foundations of AI, but recent developments suggest that probability, statistics, and network science are perhaps better suited. The truth might lie in some combination of both, depending on the application.

COURSE PHILOSOPHY

All these considerations motivated the development of this course to reflect such a view of AI and its role in chemical engineering. In my opinion, teaching AI appropriately is much more than teaching a potpourri of machine learning techniques and tools. In the current atmosphere, there is, unfortunately, a tendency to rush off and teach courses that are cookbook-style offerings in which students learn to mechanically apply different software packages without a deeper understanding of the fundamentals. I believe that we need a course that is properly founded on knowledge modeling philosophies, knowledge representation, search and inference, and knowledge extraction and management issues, all introduced in the context of chemical engineering problems. This is what I have strived to develop over the decades.

In designing such a course, I have tried to differentiate between training and education. Training focuses on "know-how," i.e., how to execute a recipe to solve a problem, whereas education emphasizes "know-why," i.e., understanding why the problem exists in the first place from a first principles-based mechanistic perspective. There is an important difference, for example, between training someone on refrigerator repair versus teaching them thermodynamics. To be sure, the former is certainly useful and has a place, but our courses ought to be more than merely utilitarian. I am sure we would all wish to avoid the complaint that "the user-friendly approach of Leibniz made it easy for people *who didn't know* calculus to teach those *who will never know* calculus!" in our present context. The easy availability of user-friendly ML tools poses such a predicament.

It is certainly exciting to see that the barriers to entry for implementing AI have come down significantly due to the

emergence of relatively easy-to-use software environments such as R®, Python®, Julia®, Keras®, and TensorFlow™. However, practicing AI properly is much more than learning to run code in such environments. It requires more than a superficial understanding of AI. Imagine learning to run a MILP program in MATLAB® without understanding the theory of mixed-integer linear programming. In the past, since such easy-to-use environments were not available, researchers were forced to learn LISP, the main language of AI, in courses that stressed the fundamental concepts, tools, and techniques of AI comprehensively. A well-educated, applied-AI researcher from that era, for instance, would take four or five courses on artificial intelligence, machine learning, natural language processing, databases, and other relevant topics to get educated deeply in artificial intelligence methods. In contrast, the modern user-friendly AI software in machine learning, for instance, makes it quite easy for a newcomer to start building machine learning models quickly and easily, thereby lulling oneself into a false belief that he or she has achieved mastery in AI or ML. This is a dangerous trap. Just as statistical tools can be misused and abused if one is not careful (see, for example, Thiese et al.[37] and Jordan[38]), a similar fate can befall the users of ML tools.

Keeping all these in mind, the course was developed as a semester-length dual-level course, a junior/senior-1st-year graduate student offering, that teaches applied AI using chemical engineering examples. This course is, in spirit but not in content, like the typical Mathematical Methods in Chemical Engineering course that is required for all graduate students in most departments. No prior background in AI is required, but a reasonable familiarity with computer programming and MATLAB is necessary. In the earlier offerings, the archaic AI language, LISP, was taught as part of the course, but now the focus is on Python.

The course was first offered in the spring of 1986 (CHEN 4580, 3 credits) and was taught subsequently in 1987 and 1988 at Columbia University. A paper describing this course was published in *Chemical Engineering Education* in 1986 in the special issue on graduate education.[39] After moving to Purdue, I taught this course about a dozen times (ChE 557, 3 credits). Since my return to Columbia, I have been teaching it in its new format since 2019 (CHEN 4580, 3 credits).

COURSE ORGANIZATION

As AI evolved in the last forty years, this course also changed, adapted, and reflected the evolution in the field. In the 1980s, it was a course mainly on *symbolic* artificial intelligence (i.e., *symbolic* AI), utilizing the concepts and techniques of *knowledge-based expert systems*. As machine learning techniques, particularly neural networks, started succeeding in the 1990s, the course evolved to become about 50% symbolic AI and 50% data-driven machine learning (i.e., *numeric* AI). With the increasing prominence of data science in recent years, the current offering has about 30% symbolic AI and the rest numeric AI, as discussed below. However, the symbolic-numeric division is not surgical. Both concepts are interwoven throughout the course, and their roles are highlighted in the different case studies.

Course Organization in the 1980s

As noted, between 1986 – 1990, the course was entirely on symbolic AI, utilizing expert systems concepts and methods. The topics were as follows:

1. Introduction
   a. Historical overview of Al
   b. AI and Process Engineering
   c. Overview of LISP
2. Knowledge Representation
   a. Issues in knowledge representation
   b. Predicate calculus, propositional calculus, and semantic networks
3. Search Techniques
   a. Forward and backward chaining
   b. Depth-first, breadth-first, and best-first search algorithms
4. Expert Systems
   a. Stages of knowledge-based expert systems development
   b. Knowledge representation and search issues
   c. Knowledge acquisition and validation
5. Expert System Development in Process Engineering
   a. Detailed example using CONPHYDE
   b. Introduction to rule-based programming and OPS5
   c. Knowledge representation and control in OPS5
6. Dealing with Uncertainty
   a. Reasoning with incomplete and uncertain information
   b. Bayesian approach
   c. Dempster-Shafer theory and Fuzzy Logic
7. Representation of Engineering Objects and Processes
   a. Frames, objects, and hybrid representation techniques

8. Expert Systems for Process Synthesis
   a. Blackboard-based systems
9. Qualitative Reasoning of Physical Systems
   a. Introduction to qualitative physics and modeling
   b. Modeling of physical objects and processes
10. Expert Systems for Process Diagnosis and Safety
   a. Model-based reasoning for process diagnosis and process safety analysis
11. Expert Systems for Operator Training and Instruction
   a. Architectural issues, cooperating expert systems, and intelligent user interface
12. General Purpose Expert System Tools and Shells
   a. Critical evaluation of tools such as KEE, ART, LOOPS, KAS, and EMYCIN
13. Project Presentations by Students

Since there was no single book then (or now, for that matter) that appropriately addressed all the topics of this course, we relied heavily on lecture notes and papers in the literature. The five books listed below were used for general-purpose discussion on AI, and they were used for about 50% of the course:

- Rich, E. *Artificial Intelligence* [2]
- Winston, P. H. and B. K. P. Horn, *LISP* [40]
- Hayes-Roth, F., Waterman, D. A. and Lenat, D. B., *Building Expert Systems* [41]
- Brownston, L., et al, *Programming Expert Systems in OPS5* [42]
- Waterman, D., *A Guide to Expert Systems* [43]

The rest of the course material was covered using relevant papers and lecture notes.[39]

Expert System Projects by Students. The students were instructed to develop a rule-based expert system implemented in OPS5 (Official Production System, version 5, language) for a process engineering problem as a part of the course. There were no midterms, quizzes, or exams otherwise. The development of the expert system was organized into six phases where, in each phase, a student tried to achieve a small part of the complete system. The students had about thirteen weeks to perform this task. The phases were synchronous with the lecture material, so that in attempting to accomplish a particular phase, the student had been given the necessary background in the preceding lectures. Clearly, these were not hard and fast deadlines and requirements, and they were relaxed to some extent based on an individual's needs. However, we found them to be useful as they kept the efforts focused and allowed the system to be developed in an incremental and timely fashion, an important aspect of expert system development.

Different students developed different systems, and here is a sample of their projects from the 1986 offering:

- First principles model-based diagnostic expert system (Steven Rich)
  - Implemented in OPS5 and LISP
  - About 110 rules
  - This work eventually led to three papers and a PhD thesis (1988)
- Plastics selection and design expert system (C-F. Chen)
  - A blackboard architecture-based system
  - Implemented in OPS5 and Framesmith
  - 70 rules and 30 frames
  - Resulted in a paper (1988)
- FCCU diagnosis expert system (Karen Zilora)
  - Based on Exxon's FCCU operations guide to diagnose 27 faults
  - OPS5 – 105 rules
  - Led to a paper (1989)

Course Organization in the 1990s

By around 1991, neural networks were beginning to make inroads into process systems engineering applications.[44] So, the course was revised to include recent developments in machine learning, as seen below in the new syllabus. The course was roughly 50% on symbolic AI and 50% on machine learning. The symbolic AI part of the course retained the essential modeling concepts and techniques outlined previously.

1. Introduction to AI and LISP
2. Knowledge Representation
3. Search
4. Knowledge-based Expert Systems
5. Inexact Reasoning: Bayesian and Fuzzy Logic
6. Supervised and Unsupervised Learning
7. Neural Networks
8. Genetic Algorithms
9. Hybrid AI: Symbolic AI + Machine Learning
10. Industrial Case Studies in Process Fault Diagnosis, Control, and Safety

11. Industrial Case Studies in Process and Product Design
12. Industrial Case Studies in Scheduling and Planning of Process Operations
13. Project Presentations by Students

The course was still based on projects, but this time we had machine-learning based projects. There was only one textbook used, *LISP*, by Winston and Horn.[40] We relied even more heavily on lecture notes and papers in the literature for most of the course. The industrial case studies were based on papers from the literature as shown below. Only a few papers are cited here as examples:

- Process fault diagnosis and control [16, 17, 28, 29, 45-67]
- Operating procedures synthesis and process safety [10, 11, 68-76]
- Process design and product design [12-15, 25, 27, 32-34, 77-88]
- Scheduling and planning of process operations [24]

Course Organization in 2021-22

The course has been further revised to incorporate recent progress in data science. The revision is mainly to include three important ideas: deep or convolutional neural nets (DNNs or CNNs), reinforcement learning, and statistical machine learning. The course topics since 2019 have been as shown below. Once again, the essential elements of symbolic AI (as mentioned in the Course Organization in the 1980s section) are retained. Furthermore, their integration with data-driven ML techniques is also discussed throughout the course wherever appropriate. The topics listed below were taught in sequence in a 13-week semester (3 credits, meeting twice a week). The papers and books cited for each topic provide a good place to start. The reader might be amused, perhaps even annoyed, by the many citations to papers from our research group. This is so because for the past three decades, there were not that many researchers working on AI in chemical engineering, and hence the availability of papers suitable for our purposes was limited. Furthermore, since we had complete access to all the data, algorithms, and software associated with our research projects, it made it convenient to rely on own work for the lectures and projects. As more people engage in AI-based projects within chemical engineering, we can expect more material to become available in the coming years. Regarding the cited sources, interested readers can further customize them to develop their own lectures for teaching a similar course.

1. Introduction to Symbolic AI [3,4]
   a. Knowledge representation [2,5]
      - Propositional and predicate calculus
      - Production rules
      - Frames, objects, ontologies
   b. Search [2,5]
      - Game trees
      - Depth-first, breadth-first, best-first techniques
      - Forward and backward chaining
2. Python Tutorials
   a. There are several excellent sources on the web – Coursera® is a good place to start.
3. Knowledge-Based Expert Systems
   a. Introduction [3,4, 12-13, 40-44]
   b. Case studies in process fault diagnosis and control [16-17, 29, 45-47, 54]
   c. Case studies in operating procedures synthesis and process safety [7, 10-11, 30-31, 70-76]
   d. Case studies in Process Design [77-81]
   e. Case studies in Product Design [14-15, 82-88]
4. Review of Linear Algebra, Probability, and Statistics [89-90]
5. Clustering Techniques [91-92]
   a. Data visualization, k-means, k-medoids
   b. Density-based clustering, hierarchical clustering [51-53, 65-67]
6. Classification Techniques [91-92]
   a. PCA/PLS, Decision Trees, kNN, LDA [29, 50-53]
   b. SVM, kernel methods, RBN, neural networks, autoencoders [47-67, 93-98]
7. Regression Techniques [91-92]
   a. Linear regression, regularization [89-90, 99-100]
   b. Nonlinear regression [89-90]
8. Genetic Algorithms and Directed Evolution for Materials Design [25, 27, 32, 81-87, 101-103]
9. Ensemble Learning, Boosting, Random Forest [91-92]
10. Modeling with Deep Neural Nets and Recurrent Neural Nets [91-92, 104-105]
11. Reinforcement Learning and Graphical Models [91-92, 106]
12. Hybrid AI Models
    a. Combining Symbolic AI with Numeric AI [25, 27, 29, 32, 33, 61, 84, 85, 107, 108]
13. Domain-specific Ontologies, Languages, and Compilers [19-22, 34-36, 109-131]

Programming project. The course is based on weekly programming assignments in Python plus a six-week-long programming project, also in Python. Here is the description of the projects:

- **Modeling Diffusion Phenomena (Fall 2020)**

  Diffusion data for particles diffusing differently are provided. Each student is given a different dataset with locations of 300 independently moving particles over time. The data are given to you in the format of [particle ID, time, position (x), position (y)].

  From the data, characterize the motion of the particles by generating appropriate features to identify commonalities between the motion of different particles. How many different particle types are there in the dataset?

  Use data visualization, preprocessing, and appropriate regression and clustering methods to solve this problem. State all your assumptions and show your reasoning clearly.

- **Thermodynamic Property Prediction (Fall 2021)**

  The students were provided thermodynamic property data for different molecules. The property prediction problem could be modeled as a regression task where the students would try different regression methods such as linear regression, polynomial regression, support vector regression, decision trees, random forests, k-NN, and so on. As with any machine learning exercise, it was expected that the students would split the given data into training, testing, and validation sets, perform appropriate regularization, tune the model hyperparameters, and transform the data using kernels, if required. The students were expected to report the best performing final model along with all the performance metrics.

The students met weekly with the TAs who were monitoring their progress, answering questions, and providing assistance as and when needed. The pandemic imposed certain limitations on the extent of the projects; this should ease from next year onwards.

SUMMARY

I have described the motivation, philosophy, topics, and organization of a course on teaching artificial intelligence to undergraduate (juniors/seniors) and graduate students in chemical engineering. The purpose is to teach them how to incorporate AI-based modeling with first-principles based models derived from the understanding of the physics and chemistry (and biology) of our products, processes, and systems. To accomplish this, I combined the *classical symbolic* AI of the 1980s with the more recent *data-driven numeric AI*, i.e., machine learning. In this regard, this course is different from the standard machine learning course, which typically focuses only on the data-driven aspect and does not address the symbolic components.

This course was first taught at Columbia University in Spring 1986. This was the first course on AI in chemical engineering taught anywhere globally. As AI evolved, the course also evolved, and the topics changed in the last thirty-five years, as discussed above. Current topics include symbolic knowledge representation, symbolic reasoning and inference, knowledge-based expert systems, modern machine learning methods, hybrid-AI models, and domain-specific languages, compilers, and knowledge engines. All these are discussed using chemical engineering case studies in process monitoring, diagnosis, control, process/product design, scheduling, optimization, and process safety.

The course philosophy, topics, and organization reflect my view on the role of AI in chemical engineering. For over three decades, I have emphasized the importance of building in first-principles knowledge into our AI models.[132-133] I have argued the importance of such hybrid models, combining mechanistic understanding based on the underlying physics, chemistry, and/or biology of our systems and processes with data-driven techniques. The domain of chemical engineering – in fact, most of science and engineering – is governed by such fundamental principles. In this regard, it is different from areas such as game playing and computer vision, where there are no such conservation laws or constitutive equations to exploit. So it makes sense in those domains to be primarily data driven.

But our applications are quite different. Unlike the ML-driven recommendation systems like Yelp® and Rotten Tomatoes®, where a disappointing experience might only result in a couple of hours and a couple of hundred dollars wasted, the cost of a mistake can be potentially high in chemical engineering applications. When a purely ML-driven black-box-based control system fails in a reactor, the resulting accident could kill people. Therefore, in our domain, we should not blindly mimic strategies used in game-playing and natural language processing and rush into developing black-box models that are not constrained by the underlying physics and chemistry. Thus, the need to explain and justify the recommendations of an AI-agent is imperative in such applications.

Another point I have long argued is that many chemical engineering applications are not "big data." We certainly have access to more data now than we did, say, a decade ago. But, unlike vision or game-playing, we don't generate terabytes of data readily, except perhaps in computer simulations. So purely data-driven techniques mimicked from such domains are not appropriate and might not work. On the other hand, our first-principles knowledge can be lever-

aged and exploited to reduce the need for large amounts of data. Thus, building hybrid-AI models is more appropriate for many chemical engineering applications. Throughout the course, my lectures and assignments emphasize this view in different contexts and applications.

I classify AI applications into three categories [134] – "easy," "hard," and "harder" problems. The relatively "easy" ones are those where plenty of data is available, and many off-the-shelf machine learning software could be used to extract useful patterns from such data. These are not particularly difficult problems, as such applications were demonstrated even in the 1990s. These applications serve well as introductory examples in the course.

On the other hand, the "hard" problems require building hybrid-AI models and generating mechanism-based causal explanations. Many problems in materials science and materials design, such as the design of polymers, [82] fuel additives [25] and rubber compounds,[27] fall into this category. Building in the underlying physics and chemistry is very important here. While work in the 1990s, again, showed how these could be done, we are still quite some ways from developing such models systematically, reliably, and easily for large-scale and diverse applications. I think this might take another 5-10 years or so. The course topics and lectures stress this important need and point out the opportunities.

The third category, the "harder" problems, includes those problems where one would have to build domain-specific ontologies, compilers, languages, and knowledge engines, such as the ones reported for catalyst design [32-34] and pharmaceutical manufacturing. [19, 21-22, 35-36] These would require much more work, not the mere application of standard techniques and software in machine learning. They need careful integration of symbolic domain knowledge with data-driven methods. I estimate such systems might take ~10 years to emerge as routine implementations.

In my opinion, it is in the latter two categories where the most interesting and intellectually challenging problems lie. Therefore, the course lectures and case studies are designed to highlight these points and opportunities.

Having participated in the first two phases of the AI "revolution" – expert systems in the 1980s and neural networks in the 1990s – I am somewhat wary of the hype surrounding data science now. While there is much to be excited about in machine learning, we should not abandon first-principles based modeling. As noted, this is particularly important given that our domain is governed by physicochemical (and biological) mechanisms, unlike game playing and natural language processing. Therefore, we need to strike a proper balance between mechanistic models and data-driven techniques. Over the past three decades, I designed and taught a course that strived to achieve this balance by incorporating both symbolic and numeric AI.

ACKNOWLEDGMENTS

I am very grateful to my former and current PhD students who took this course over the years and subsequently served as teaching assistants. They suggested many improvements that have benefitted the course. I am also thankful to the several hundred students, both undergrad and grad, who have taken this course at Columbia and at Purdue for their helpful feedback. The constructive critiques of the three anonymous reviewers are also much appreciated. I also would like to thank Phil Westmoreland, Martha Grover, Lisa Bullard, Erin Stevic, and the CEE staff for their kind support and assistance with the review and copy editing process.

REFERENCES

1. Venkatasubramanian V (2019) The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*. 65(2):466-478.
2. Rich E (1983) *Artificial Intelligence*. McGraw-Hill. New York, NY.
3. Venkatasubramanian V (1989) *Knowledge-Based Systems in Process Engineering: Case Studies in Heuristic Classification*. The CACHE Corporation. Austin, TX.
4. Ungar L and Venkatasubramanian V (1990) *Advance Reasoning Architectures for Expert Systems*. The CACHE Corporation. Austin, TX.
5. Russell S and Norvig P (2010) *Artificial Intelligence: A Modern Approach*. Prentice Hall. Fort Collins, CO.
6. Iri M, Aoki K, O'Shima E, and Matsuyama H (1979) An algorithm for diagnosis of system failures in the chemical process. *Computers & Chemical Engineering*. 3(1-4): 489-493.
7. Vaidhyanathan R and Venkatasubramanian V (1995). Digraph-based models for automated HAZOP analysis. *Reliability Engineering & System Safety*. 50(1):33-49.
8. Maurya MR, Rengaswamy R, and Venkatasubramanian V (2003) A systematic framework for the development and analysis of signed digraphs for chemical processes. 1. Algorithms and analysis. *Industrial & Engineering Chemistry Research*. 42(20):4789-4810.
9. Maurya MR, Rngaswamy R, and Venkatasubramanian V (2003) A systematic framework for the development and analysis of signed digraphs for chemical processes. 2. Control loops and flowsheet analysis. *Industrial & Engineering Chemistry Research*. 42(20):4811-4827.
10. Viswanathan S, Johnsson C, Srinivasan R, Venkatasubramanian V, and Ärzen KE (1998). Automating operating procedure synthesis for batch processes: Part I. Knowledge representation and planning framework. *Computers & Chemical Engineering*. 22(11):1673-1685.
11. Viswanathan S, Johnsson C, Srinivasan R, Venkatasubramanian V, and Ärzen K E (1998). Automating operating procedure synthesis for batch processes: Part II. Implementation and application. *Computers & Chemical Engineering*. 22(11):1687-1698.
12. Banares-Alcantara R, Sriram D, Venkatasubramanian V, Westerberg A, and Rychener M (1985) Knowledge-based expert systems for CAD. *Chemical Engineering Progress*. 81(9): 25-30.
13. Banares-Alcantara R, Westerberg AW, and Rychener MD (1985). Development of an expert system for physical property predictions. *Computers & Chemical Engineering*. 9(2):127-142.
14. Bañares-Alcántara R, Westerberg AW, Ko EI, and Rychener MD (1987) DECADE—A hybrid expert system for catalyst selection—I. Expert system consideration. *Computers & Chemical Engineering*. 11(3):265-277.
15. Banares-Alcantara R, Ko EI, Westerberg AW, and Rychener M D (1988) DECADE—a hybrid expert system for catalyst selection—II. Final architecture and results. *Computers & Chemical Engineering*. 12(9-10): 923-938.

16. Rich SH and Venkatasubramanian V (1987) Model-based reasoning in diagnostic expert systems for chemical process plants. *Computers & Chemical Engineering*. 11(2):111-122.
17. Venkatasubramanian V and Rich SH (1988) An object-oriented two-tier architecture for integrating compiled and deep-level knowledge for process diagnosis. *Computers & Chemical Engineering*. 12(9-10):903-921.
18. Aldea A, Bañares-Alcántara R, Bocio J, Gramajo J, Isern D, Kokossis A, Jiménez L, Moreno A, and Riaño D (2003) An ontology-based knowledge management platform. *Proceedings of the Workshop on Information Integration on the Web, 18th International Joint Conference on Artificial Intelligence*. 177—182.
19. Venkatasubramanian V, Zhao C, Joglekar G, Jain A, Hailemariam L, Suresh P, Akkisetty P, Morris K, and Reklaitis G V (2006) Ontological informatics infrastructure for pharmaceutical product development and manufacturing. *Computers & Chemical Engineering*. 30(10-12):1482-1496.
20. Marquardt W, Morbach J, Wiesner A, Yang A, and CAPE O (2010) *A Re-Usable Ontology for Chemical Process Engineering*. Springer. Berlin, Germany.
21. Hailemariam L and Venkatasubramanian V (2010) Purdue ontology for pharmaceutical engineering: Part I. Conceptual framework. *Journal of Pharmaceutical Innovation*. 5(3):88-99.
22. Hailemariam L and Venkatasubramanian V (2010) Purdue ontology for pharmaceutical engineering: Part II. Applications. *Journal of Pharmaceutical Innovation*. 5(4):139-146.
23. Katare S and Venkatasubramanian V (2001) An agent-based learning framework for modeling microbial growth. *Engineering Applications of Artificial Intelligence*. 14(6):715-726.
24. Julka N, Srinivasan R and Karimi I (2002) Agent-based supply chain management—1: Framework. *Computers & Chemical Engineering*. 26(12):1755-1769.
25. Sundaram A, Ghosh P, Caruthers JM, and Venkatasubramanian V (2001) Design of fuel additives using neural networks and evolutionary algorithms. *AIChE Journal*. 47(6):1387-1406.
26. Viswanathan S, Shah N, and Venkatasubramanian V (2002) Hybrid framework for hazard identification and assessment in batch processes. *AIChE Journal*. 48(8):1765-1774.
27. Ghosh P, Katare S, Patkar P, Caruthers JM, Venkatasubramanian V, and Walker, KA (2003) Sulfur vulcanization of natural rubber for benzothiazole accelerated formulations: From reaction mechanisms to a rational kinetic model. *Rubber Chemistry and Technology*. 76(3):592-693.
28. Rich SH and Venkatasubramanian V (1989) Causality-based failure-driven learning in diagnostic expert systems. *AIChE Journal*. 35(6):943-950.
29. Vedam H and Venkatasubramanian V (1999) PCA-SDG based process monitoring and fault diagnosis. *Control Engineering Practice*. 7(7):903-917.
30. Zhao C, Bhushan M, and Venkatasubramanian V (2005) PHASuite: An automated HAZOP analysis tool for chemical processes: Part I - Knowledge engineering framework. *Process Safety and Environmental Protection*. 83(6)509-532.
31. Zhao C, Bhushan M, and Venkatasubramanian V (2005) PHASuite: An automated HAZOP analysis tool for chemical processes: Part II - Implementation and Case Study. *Process Safety and Environmental Protection*. 83(6):533-548.
32. Caruthers JM, Lauterbach JA, Thomson KT, Venkatasubramanian V, Snively CM, Bhan A, Katare S, and Oskarsdottir G (2003) Catalyst design: Knowledge extraction from high-throughput experimentation. *Journal of Catalysis*. 216(1-2):98-109.
33. Katare S, Caruthers JM, Delgass WN, and Venkatasubramanian V (2004) An intelligent system for reaction kinetic modeling and catalyst design. *Industrial & Engineering Chemistry Research*. 43(14):3484-3512.
34. Hsu SH, Krishnamurthy B, Rao P, Zhao C, Jagannathan S, and Venkatasubramanian V (2008) A domain-specific compiler theory-based framework for automated reaction network generation. *Computers & Chemical Engineering*. 32(10):2455-2470.
35. Suresh P, Hsu S H, Akkisetty P, Reklaitis GV, and Venkatasubramanian V (2010) OntoMODEL: Ontological mathematical modeling knowledge management in pharmaceutical product development, 1: Conceptual framework. *Industrial & Engineering Chemistry Research*. 49(17):7758-7767.
36. Suresh P, Hsu S H, Akkisetty P, Reklaitis GV, and Venkatasubramanian V (2010) OntoMODEL: Ontological mathematical modeling knowledge management in pharmaceutical product development, 2: Applications. *Industrial & Engineering Chemistry Research*. 49(17):7768-7781.
37. These MS, Arnold ZC, and Walker SD (2015) The misuse and abuse of statistics in biomedical research. Biochemia Medica. 25(1):5-11.
38. Jordan MI (2019) Artificial intelligence—The revolution hasn't happened yet. *Harvard Data Science Review*. 1(1).
39. Venkatasubramanian V (1986) Artificial intelligence in process engineering: experiences from a graduate course. *Chemical Engineering Education*. 20(4):188-192.
40. Winston PH and Horn BK (1986) LISP. Addison-Wesley. Boston, MA.
41. Hayes-Roth F, Waterman DA, and Lenat DB (1983) *Building Expert Systems*. Addison-Wesley. Boston, MA.
42. Brownston L, Farrell R, Kant E, and Martin N (1985) *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming*. Addison-Wesley. Boston, MA.
43. Waterman D A (1985) *A Guide to Expert Systems*. Addison-Wesley. Boston, MA.
44. Venkatasubramanian V and McAvoy TJ (1992) Neural network applications in chemical engineering. *Computers & Chemical Engineering*. 16(4):v-vi.
45. Venkatasubramanian V (1988) Towards a specialized shell for diagnostic expert systems for chemical plant. *Journal of Loss Prevention in the Process Industries*. 1(2):84-91.
46. Rich SH, Venkatasubramanian V, Nasrallah M, and Matteo C (1989) Development of a diagnostic expert system for a whipped toppings process. *Journal of Loss Prevention in the Process Industries*. 2(3):145-154.
47. Venkatasubramanian V (1985) Inexact reasoning in expert systems: A stochastic parallel network approach. *Proceedings of the Second Conference on Artificial Intelligence Applications*.
48. Venkatasubramanian V and Chan K (1989) A neural network methodology for process fault diagnosis. *AIChE Journal*. 35(12):1993-2002.
49. Venkatasubramanian V, Vaidyanathan R, and Yamamoto Y (1990) Process fault detection and diagnosis using neural networks—I. Steady-state processes. *Computers & Chemical Engineering*. 14(7):699-712.
50. Vaidyanathan R and Venkatasubramanian V (1992) On the nature of fault space classification structure developed by neural networks. *Engineering Applications of Artificial Intelligence*. 5(4):289-297.
51. Kavuri SN and Venkatasubramanian V (1993) Representing bounded fault classes using neural networks with ellipsoidal activation functions. *Computers & Chemical Engineering*. 17(2):139-163.
52. Kavuri SN and Venkatasubramanian V (1993) Using fuzzy clustering with ellipsoidal units in neural networks for robust fault classification. *Computers & Chemical Engineering*. 17(8):765-784.
53. Kavuri SN and Venkatasubramanian V (1994) Neural network decomposition strategies for large-scale fault diagnosis. *International Journal of Control*. 59(3):767-792.
54. Kramer MA and Palowitch BL (1987) A rule-based approach to fault diagnosis using the signed directed graph. *AIChE Journal*. 33(7):1067-1078.
55. Kramer MA (1991) Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*. 37(2):233-243.
56. Ungar LH, Powell BA, and Kamens SN (1990) Adaptive networks for fault diagnosis and process control. *Computers & Chemical Engineering*. 14(4-5):561-572.
57. Hoskins JC, Kaliyur KM, and Himmelblau DM (1991) Fault diagnosis in complex chemical plants using artificial neural networks. *AIChE Journal*. 37(1):137-141.

58. Qin SJ and McAvoy TJ (1992) Nonlinear PLS modeling using neural networks. *Computers & Chemical Engineering*. 16(4):379-391.
59. Leonard JA and Kramer MA (1993) Diagnosing dynamic faults using modular neural nets. *IEEE Expert*. 8(2):44-53.
60. Psichogios DC and Ungar LH (1992) A hybrid neural network-first principles approach to process modeling. *AIChE Journal*. 38(10):1499-1511.
61. Mylaraswamy D and Venkatasubramanian V (1997) A hybrid framework for large scale process fault diagnosis. *Computers & Chemical Engineering*. 21:S935-S940.
62. Aoyama A, Doyle III FJ, and Venkatasubramanian V (1995) Control-affine fuzzy neural network approach for nonlinear process control. *Journal of Process Control*. 5(6):375-386.
63. Aoyama A, Doyle III FJ, and Venkatasubramanian V (1995) A fuzzy neural-network approach for nonlinear process control. *Engineering Applications of Artificial Intelligence*. 8(5):483-498.
64. Aoyama A and Venkatasubramanian V (1995) Internal model control framework using neural networks for the modeling and control of a bioreactor. *Engineering Applications of Artificial Intelligence*. 8(6):689-701.
65. Venkatasubramanian V, Rengaswamy R, Yin K, and Kavuri SN (2003) A review of process fault detection and diagnosis: Part I - Quantitative model-based methods. *Computers & Chemical Engineering*. 27(3):293-311.
66. Venkatasubramanian V, Rengaswamy R, Yin K, and Kavuri SN (2003) A review of process fault detection and diagnosis: Part II - Qualitative models and search strategies. *Computers & Chemical Engineering*. 27(3), 313-326.
67. Venkatasubramanian V, Rengaswamy R, Yin K, and Kavuri SN (2003) A review of process fault detection and diagnosis: Part III - Process history based methods. *Computers & Chemical Engineering*. 27(3), 327-346.
68. Venkatasubramanian V and Vaidhyanathan R (1994) A knowledge-based framework for automating HAZOP analysis. *AIChE Journal*. 40(3):496-505.
69. Vaidhyanathan R and Venkatasubramanian V (1996) Experience with an expert system for automated HAZOP analysis. *Computers & Chemical Engineering*. 20: S1589-S1594.
70. Srinivasan R and Venkatasubramanian V (1996) Petri net-digraph models for automating HAZOP analysis of batch process plants. *Computers & Chemical Engineering*. 20:S719-S725.
71. Shin D and Venkatasubramanian V (1996) Intelligent tutoring system framework for operator training for diagnostic problem solving. *Computers & Chemical Engineering*. 20:S1365-S1370.
72. Srinivasan R and Venkatasubramanian V (1998) Automating HAZOP analysis of batch chemical plants: Part I. The knowledge representation framework. *Computers & Chemical Engineering*. 22(9):1345-1355.
73. Srinivasan R and Venkatasubramanian V (1998) Automating HAZOP analysis of batch chemical plants: Part II. Algorithms and application. *Computers & Chemical Engineering*. 22(9):1357-1370.
74. Venkatasubramanian V, Zhao J, and Viswanathan S (2000) Intelligent systems for HAZOP analysis of complex process plants. *Computers & Chemical Engineering*. 24(9-10):2291-2302.
75. Mu F and Venkatasubramanian V (2003) Online HAZOP analysis for abnormal event management of batch process. *Computer Aided Chemical Engineering*. 14:803-808). Elsevier.
76. Dash S and Venkatasubramanian V (2003) Integrated framework for abnormal event management and process hazards analysis. *AIChE Journal*. 49(1):124-139.
77. Venkatasubramanian V and Chen CF (1986) A blackboard architecture for plastics design. *Artificial Intelligence in Engineering*. 1(2):117-122.
78. Stephanopoulos G, Johnston J, Kriticos T, Lakshmanan R, Mavrovouniotis M, and Siletti C (1987) DESIGN-KIT: An object-oriented environment for process engineering. *Computers & Chemical Engineering*. 11(6):655-674.
79. Stephanopoulos G, Henning G, and Leone H (1990) MODEL. LA. A modeling language for process engineering—I. The formal framework. *Computers & Chemical Engineering*. 14(8):813-846.
80. Myers DR, Davis JF, and Herman DJ (1988). A task-oriented approach to knowledge-based systems for process engineering design. *Computers & Chemical Engineering*. 12(9-10):959-971.
81. Androulakis IP and Venkatasubramanian V (1991) A genetic algorithmic framework for process design and optimization. *Computers & Chemical Engineering*. 15(4):217-228.
82. Harper PM, Gani R, Kolar P, and Ishikawa T (1999) Computer-aided molecular design with combined molecular modeling and group contribution. *Fluid Phase Equilibria*. 158:337-347.
83. Venkatasubramanian V, Chan K, and Caruthers JM (1995) Evolutionary design of molecules with desired properties using the genetic algorithm. *Journal of Chemical Information and Computer Sciences*. 35(2):188-195.
84. Sundaram A and Venkatasubramanian V (1998) Parametric sensitivity and search-space characterization studies of genetic algorithms for computer-aided polymer design. *Journal of Chemical Information and Computer Sciences*. 38(6):1177-1191.
85. Katare S, Patkar P, Ghosh P, Caruthers JM, Delgass WN, and Venkatasubramanian V (2001) A systematic framework for rational materials formulation and design. *Proceedings of the Symposium on Materials Design Approaches and Experiences*.
86. Patkar P and Venkatasubramanian V (2003) Genetic Algorithms based Computer-Aided Molecular Design. *Computer Aided Molecular Design: Theory and Practice*, Achenie L, Gani R, and Venkatasubramanian V (Eds). Elsevier. Amsterdam.
87. Patkar PR and Venkatasubramanian V (2003) Polymer design case study. *Computer Aided Chemical Engineering*. 12:303-317. Elsevier, Amsterdam.
88. Syal S (2007) *Intelligent System Framework for the Rational Design of Rubber Components*. Ph. D. thesis, Purdue University.
89. Montgomery DC and Runger GC (2010) Applied Statistics and Probability for Engineers. John Wiley & Sons. New York, NY.
90. Strang G. (1993) *Introduction to Linear Algebra*. Wellesley-Cambridge. Wellesley, MA.
91. Hastie T, Tibshirani R, Friedman JH, and Friedman JH (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer. New York, NY.
92. Müller A C and Guido S (2016) Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly Media. Sebastopol, CA.
93. Rengaswamy R, Mylaraswamy D, Årzén KE, and Venkatasubramanian V (2001) A comparison of model-based and neural network-based diagnostic methods. *Engineering Applications of Artificial Intelligence*. 14(6):805-818.
94. Hsu SH, Stamatis SD, Caruthers JM, Delgass WN, Venkatasubramanian V, Blau GE, Lasinski M, and Orcun S (2009) Bayesian framework for building kinetic models of catalytic systems. *Industrial & Engineering Chemistry Research*. 48(10):4768-4790.
95. Maddipati S, Nandigam R, Kim S, and Venkatasubramanian V (2011) Learning patterns in combinatorial protein libraries by Support Vector Machines. *Computers & Chemical Engineering*. 35(6):1143-1151.
96. Li YF and Venkatasubramanian V (2016) Leveraging Bayesian approach to predict drug manufacturing performance. *Journal of Pharmaceutical Innovation*. 11(4):331-338.
97. Li YF and Venkatasubramanian V (2016) Integrating design of experiments and principal component analysis to reduce downstream cost of goods in monoclonal antibody production. *Journal of Pharmaceutical Innovation*. 11(4):352-361.
98. Li YF and Venkatasubramanian V (2018) Neural network to understand process capability and process intermediates acceptance criteria in monoclonal antibody production process. *Journal of Pharmaceutical Innovation*. 13(1):36-47.
99. Huang Y, Dash S, Reklaitis GV, and Venkatasubramanian V (2000) EKF based estimator for FDI in Model IV FCCU. *Proceedings of SAFEPROCESS*.

100. Villez K, Srinivasan B, Rengaswamy R, Narasimhan S, and Venkatasubramanian V (2011) Kalman-based strategies for fault detection and identification (FDI): Extensions and critical evaluation for a buffer tank system. *Computers & Chemical Engineering*. 35(5):806-816.
101. Srinivasan B, Vo T, Zhang Y, Gang O, Kumar S, and Venkatasubramanian V (2013) Designing DNA-grafted particles that self-assemble into desired crystalline structures using the genetic algorithm. *Proceedings of the National Academy of Sciences*. 110(46):18431-18435.
102. Chakraborty A, Sivaram A, Samavedham L, and Venkatasubramanian V (2020) Mechanism discovery and model identification using genetic feature extraction and statistical testing. *Computers & Chemical Engineering*. 140:106900.
103. Chakraborty A, Sivaram A, and Venkatasubramanian V (2021) AI-DARWIN: A first principles-based model discovery engine using machine learning. *Computers & Chemical Engineering*. 154:107470.
104. Sivaram A, Das L, and Venkatasubramanian V (2020) Hidden representations in deep neural networks: Part 1. Classification problems. *Computers & Chemical Engineering*. 134:106669.
105. Das L, Sivaram A, and Venkatasubramanian V (2020) Hidden representations in deep neural networks: Part 2. Regression problems. *Computers & Chemical Engineering*. 139:106895.
106. Venkatasubramanian V (2021) Editorial. *Computers & Chemical Engineering*. 145:107176.
107. Akkisetty P, Lee U, Reklaitis GV, and Venkatasubramanian V (2010) Population balance model-based hybrid neural network for a pharmaceutical milling process. *Journal of Pharmaceutical Innovation*. 5(4):161-168.
108. Sivaram A and Venkatasubramanian V (2022) XAI-MEG: Combining symbolic AI and machine learning to generate first-principles models and causal explanations. *AIChE Journal*. 68(6):e17687.
109. Sesen M B, Suresh P, Banares-Alcantara R, and Venkatasubramanian V (2010) An ontological framework for automated regulatory compliance in pharmaceutical manufacturing. *Computers & Chemical Engineering*. 34(7):1155-1169.
110. Hamdan IM, Reklaitis GV, and Venkatasubramanian V (2012) Real-time exceptional events management for a partial continuous dry granulation line. *Journal of Pharmaceutical Innovation*. 7(3):95-118.
111. Gupta A, Giridhar A, Venkatasubramanian V, and Reklaitis GV (2013) Intelligent alarm management applied to continuous pharmaceutical tablet manufacturing: an integrated approach. *Industrial & Engineering Chemistry Research*. 52(35):12357-12368.
112. Remolona MM, Conway MF, Balasubramanian S, Fan L, Feng Z, Gu T, Kim H, Nirantar PM, Panda S, Ranabothu NR, Rastogi N, and Venkatasubramanian V (2017) Hybrid ontology-learning materials engineering system for pharmaceutical products: Multi-label entity recognition and concept detection. *Computers & Chemical Engineering*. 107:49-60.
113. Venkatasubramanian V and Zhang Z (2016) TeCSMART: A hierarchical framework for modeling and analyzing systemic risk in sociotechnical systems. *AIChE Journal*. 62(9):3065-3084.
114. Zhang Z, Gonzalez MC, Morse SS, and Venkatasubramanian V (2017) Knowledge management framework for emerging infectious diseases preparedness and response: design and development of public health document ontology. *JMIR Research Protocols*. 6(10):e7904.
115. Suresh R, Sivaram A, and Venkatasubramanian V (2019) A hierarchical approach for causal modeling of process systems. *Computers & Chemical Engineering*. 123:170-183.
116. Mann V and Venkatasubramanian V (2021) Retrosynthesis prediction using grammar-based neural machine translation: An information-theoretic approach. *Computers & Chemical Engineering*. 155:107533.
117. Venkatasubramanian V and Mann V (2022) Artificial intelligence in reaction prediction and chemical synthesis. *Current Opinion in Chemical Engineering*. 36:100749.
118. Zhao C, Jain A, Hailemariam L, Suresh P, Akkisetty P, Joglekar G, Venkatasubramanian V, Reklaitis GV, Morris KR, and Basu P (2006) Toward intelligent decision support for pharmaceutical product development. *Journal of Pharmaceutical Innovation*. 1(1):23-35.
119. Jain A, Kumar P, Joglekar G, Hailemariam L, Suresh P, Zhao C, Morris K, Reklaitis G, and Venkatasubramanian V (2008) Integrated Decision Support Tool for Pharmaceutical Product Development. *Proceedings of the 18th European Symposium on Computer Aided Process Engineering*. Lyon, France.
120. Krishnamurthy B, Malik T, Stamatis S, Venkatasubramanian V, and Caruthers JM (2008) Rule-based classification systems for informatics. *Proceedings of the IEEE Fourth International Conference on eScience*.
121. Shukla A, Lalit VA, and Venkatasubramanian V (2011) Optimizing efficiency-robustness trade-offs in supply chain design under uncertainty due to disruptions. *International Journal of Physical Distribution & Logistics Management*. 41(6):623-647.
122. Luo Y, Iyengar G, and Venkatasubramanian V (2016) Soft regulation with crowd recommendation: Coordinating self-interested agents in sociotechnical systems under imperfect information. *PLOS One*. 11(3):e0150343.
123. Spellings M and Glotzer SC (2018) Machine learning for crystal identification and discovery. *AIChE Journal*, 64(6):2198-2206.
124. Nandy A, Duan C, Janet J P, Gugler S and Kulik HJ (2018) Strategies and software for machine learning accelerated discovery in transition metal chemistry. *Industrial & Engineering Chemistry Research*. 57(42):13973-13986.
125. Tran K and Ulissi ZW (2018) Active learning across intermetallics to guide discovery of electrocatalysts for CO_2 reduction and H2 evolution. *Nature Catalysis*. 1(9):696-703.
126. Salciccioli M, Stamatakis M, Caratzoulas S, and Vlachos DG (2011) A review of multiscale modeling of metal-catalyzed reactions: Mechanism development for complexity and emergent behavior. *Chemical Engineering Science*. 66(19):4319-4355.
127. Rangarajan S, Bhan A, and Daoutidis P (2012) Language-oriented rule-based reaction network generation and analysis: Description of RING. *Computers & Chemical Engineering*. 45:114-123.
128. Gao CW, Allen JW, Green WH, and West RH (2016) Reaction mechanism generator: Automatic construction of chemical kinetic mechanisms. *Computer Physics Communications*. 203:212-225.
129. Goldsmith CF and West RH (2017) Automatic generation of microkinetic mechanisms for heterogeneous catalysis. *The Journal of Physical Chemistry C*. 121(18):9970-9981.
130. Coley CW, Barzilay R, Jaakkola TS, Green WH, and Jensen KF (2017) Prediction of organic reaction outcomes using machine learning. *ACS Central Science*. 3(5):434-443.
131. Coley CW, Green WH, and Jensen KF (2018) Machine learning in computer-aided synthesis planning. *Accounts of Chemical Research*. 51(5):1281-1289.
132. Venkatasubramanian V (2019) Artificial Intelligence in Chemical Engineering: Quo Vadis? Leo Chiang (Dow) interview. AIChE Spring Meeting. New Orleans, LA. https://www.youtube.com/watch?v=npnJP7PBCQw. Accessed August 16, 2022.
133. Venkatasubramanian V (2020) Artificial Intelligence in Chemical Engineering: Past, Present, and Future. Seminar at Carnegie Mellon University. Pittsburgh, PA. http://egon.cheme.cmu.edu/ewo/audio/EWO_Seminar_03_20_20_AI_in_ChE.mp4. Accessed August 16, 2022.
134. Venkatasubramanian V (2019) Artificial intelligence in materials science: The good, the bad, and the ugly. *Proceedings of the AIChE Annual Meeting*. ❐