Perspective

The Promise of Artificial Intelligence in Chemical Engineering: Is It Here, Finally?

Venkat Venkatasubramanian
Dept. of Chemical Engineering, Columbia University, New York, NY 10027

DOI 10.1002/aic.16489
Published online December 19, 2018 in Wiley Online Library (wileyonlinelibrary.com)

Keywords: AI, machine learning, data science, predictive analytics, materials science, design, control, optimization, diagnosis, safety

Artificial Intelligence in Chemical Engineering: Background

The current excitement about artificial intelligence (AI), particularly machine learning (ML), is palpable and contagious. The expectation that AI is poised to "revolutionize," perhaps even take over, humanity has elicited prophetic visions and concerns from some luminaries.[1-4] There is also a great deal of interest in the commercial potential of AI, which is attracting significant sums of venture capital and state-sponsored investment globally, particularly in China.[5] McKinsey, for instance, predicts the potential commercial impact of AI in several domains, envisioning markets worth trillions of dollars.[6] All this is driven by the sudden, explosive, and surprising advances AI has made in the last 10 years or so. AlphaGo, autonomous cars, Alexa, Watson, and other such systems, in game playing, robotics, computer vision, speech recognition, and natural language processing are indeed stunning advances. But, as with earlier AI breakthroughs, such as expert systems in the 1980s and neural networks in the 1990s, there is also considerable hype and a tendency to overestimate the promise of these advances, as market research firm Gartner and others have noted about emerging technology.[7]

It is quite understandable that many chemical engineers are excited about the potential applications of AI, and ML in particular,[8] for use in such applications as catalyst design.[9-11] It might seem that this prospect offers a novel approach to challenging, long-standing problems in chemical engineering using AI. However, the use of AI in chemical engineering is not new—it is, in fact, a 35-year-old ongoing program with some remarkable successes along the way.

This article is aimed broadly at chemical engineers who are interested in the prospects for AI in our domain, as well as at researchers new to this area. The objectives of this article are threefold. First, to review the progress we have made so far, highlighting past efforts that contain valuable lessons for the future. Second, drawing on these lessons, to identify promising current and future opportunities for AI in chemical engineering. To avoid getting caught up in the current excitement and to assess the prospects more carefully, it is important to take such a longer and broader view, as a "reality check." Third, since AI is going to play an increasingly dominant role in chemical engineering research and education, it is important to recount and record, however incomplete, certain early milestones for historical purposes.

It is apparent that chemical engineering is at an important crossroads. Our discipline is undergoing an unprecedented transition—one that presents significant challenges and opportunities in modeling and automated decision-making. This has been driven by the convergence of cheap and powerful computing and communications platforms, tremendous progress in molecular engineering, the ever-increasing automation of globally integrated operations, tightening environmental constraints, and business demands for speedier delivery of goods and services to market. One important outcome from this convergence is the generation, use, and management of massive amounts of diverse data, information, and knowledge, and this is where AI, particularly ML, would play an important role.

So, what is AI? The term was coined in 1956 at a math conference at Dartmouth College. Over the years, there have been many definitions of AI, but I have always found the following to be simple, visionary, and useful[12]: "Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better." Note that this definition does not say which "things." The implication is that AI could eventually end up doing all "things" that humans do, and do them much better—that is, achieve super-human performance as witnessed recently with AlphaGO[13] and AlphaGO Zero.[14] This implication is sometimes called the central dogma of AI. Historically, the term AI reflected collectively to the following branches:

- Game playing—for example, Chess, Go
- Symbolic reasoning and theorem-proving—for example, Logic Theorist, MACSYMA
- Robotics—for example, self-driving cars

Correspondence concerning this article should be addressed to V. Venkatasubramanian at venkat@columbia.edu

- Vision—for example, facial recognition
- Speech recognition, Natural language processing–for example, Siri
- Distributed & evolutionary AI—for example, drone swarms, agent-based models
- Hardware for AI—for example, Lisp machines
- Expert systems or knowledge-based systems—for example, MYCIN, CONPHYDE
- ML—for example, clustering, deep neural nets, Bayesian belief nets

Some of these are application-focused, such as game playing and vision. Others are methodological, such as expert systems and ML—the two branches that are most directly and immediately applicable to our domain, and hence the focus of this article. These are the ones that have been investigated the most in the last 35 years by AI researchers in chemical engineering. While the current "buzz" is mostly around ML, the expert system framework holds important symbolic knowledge representation concepts and inference techniques that could prove useful in the years ahead as we strive to develop more comprehensive solutions that go beyond the purely data-centric emphasis of ML.

Many tasks in these different branches of AI share certain common features. They all require pattern recognition, reasoning, and decision-making under complex conditions. And they often deal with ill-defined problems, noisy data, model uncertainties, combinatorially large search spaces, nonlinearities, and the need for speedy solutions. But such features are also found in many problems in process systems engineering (PSE)—in synthesis, design, control, scheduling, optimization, and risk management. So, some of us thought, in the early-1980s, that we should examine such problems from an AI perspective.[15-17] Just as it is today, the excitement about AI at that time was centered on expert systems. It was palpable and contagious, with high expectations for AI's near-term potential.[18-20] Hundreds of millions of dollars were invested in AI start-ups as well as within large companies. AI spurred the development of special purpose hardware, called Lisp machines (e.g., Symbolics Lisp machines). Promising proof-of-concept systems were demonstrated in many domains, including chemical engineering (see below). In this phase, it was expected that AI would have a significant impact in chemical engineering in the near future. However, unlike optimization and model predictive control, AI did not quite live up to its early promise.

So, what happened? Why was not AI as impactful? Before addressing this question, it is necessary to examine the different phases of AI, as I classify them, in chemical engineering.

Phases of AI in Chemical Engineering: Past Work

Phase 0: Early attempts

While major efforts to developing AI methods for chemical engineering problems started in the early 1980s, it is remarkable that some researchers (for instance, Gary Powers, Dale Rudd, and Jeff Siirola) were investigating AI in PSE in the late 1960s and early 1970s.[21] In particular, the Adaptive Initial DEsign Synthesizer system, developed by Siirola and Rudd[22] for process synthesis, represents a significant development. This was arguably the first system that employed AI methods such as means-and-ends analysis, symbolic manipulation, and linked data structures in chemical engineering.

Phase I: Expert systems era (~1983 to ~1995)

Phase I, the *Expert Systems Era* (from the early 1980s through the mid-1990s), saw the first broad effort to exploit AI in chemical engineering.

Expert systems, also called *knowledge-based systems*, *rule-based systems*, or *production systems*, are computer programs that mimic the problem-solving of humans with expertise in a given domain.[23,24] Expert problem-solving typically involves large amounts of specialized knowledge, called domain knowledge, often in the form of rules of thumb, called heuristics, typically learned and refined over years of problem-solving experience. The amount of knowledge manipulated is often vast, and the expert system rapidly narrows down the search by recognizing patterns and by using the appropriate heuristics.

The architecture of these systems was inspired by the stimulus–response model of cognition from psychology and pattern-matching-and-search model of symbolic computation, which originated in Emil Post's work in symbolic logic. Building on this work, Simon and Newell in the late 1960s and 1970s devised the production system framework, an important conceptual, representational, and architectural breakthrough, for developing expert systems.[25-27]

The crucial insight here was that one needs to, and one can, separate domain knowledge from its order of execution, that is, from search or inference, thereby achieving the necessary computational flexibility to address ill-structured problems. In contrast, conventional programs consist of a set of statements whose order of execution is predetermined. Therefore, if the execution order is not known or cannot be anticipated *a priori*, as in the case of medical diagnosis, for example, this approach will not work. Expert systems programming alleviated this problem by making a clear distinction between the knowledge base and the search or inference strategy. This not only allowed for flexible execution, it also facilitated the incremental addition of knowledge, without distorting the overall program structure. This rule-based knowledge representation and architecture are intuitive, and relatively easy to understand and generate explanations about the system's decisions.

This new approach facilitated the development of a number of impressive expert systems, starting with MYCIN, an expert system for diagnosing infectious diseases[28] developed at Stanford University during 1972–82. This led to other successful systems such as PROSPECTOR (for mineral prospecting[29]), R1 (configuring Vax computers[30]), and so on, in this era.

These systems inspired the first expert system application in chemical engineering, CONPHYDE, developed in 1983 by Bañares-Alcántara, Westerberg, and Rychner at Carnegie Mellon[16] for predicting thermophysical properties of complex fluid mixtures. CONPHYDE was implemented using Knowledge Acquisition System that was used for PROSPECTOR. This was quickly followed by DECADE, in 1985, again from the same CMU researchers,[17] for catalyst design.

There was other such remarkable early work in process synthesis, design, modeling, and diagnosis as well. In synthesis and in design, for instance, important conceptual advances were made by Stephanopoulos and his students, starting with Design-Kit,[31] and in modeling, MODELL.LA, a language for developing process models.[32] In process fault diagnosis, Davis[33] and Kramer,[34,35] and their groups, made important contributions in the same period. My group developed causal model-based diagnostic expert systems,[36] a departure from the heuristics-based approach, which was the dominant theme of the time. We also demonstrated the potential of learning expert systems, an unusual idea at that time as automated learning in expert systems was not in vogue.[37] The need for causal models in AI, a topic that has emerged as very important now,[38] was also recognized in those early years.[39] This period also saw expert system work commencing in Europe,[40] particularly for conceptual design support.

An important large-scale program in this era was the Abnormal Situation Management (ASM) consortium, funded at $17 million by the National Institute of Standards and Technology's Advanced Technology Program and by the leading oil companies, under the leadership of Honeywell.[41] Three different academic groups, led by Davis (Ohio State), Vicente (University of Toronto), and myself at Purdue, were also involved in the consortium. This program is the forerunner to the current Clean Energy Smart Manufacturing Innovation Institute that was funded in 2016.[42]

The first course on AI in PSE was developed and taught at Columbia University in 1986, and it was subsequently offered at Purdue University for many years. The earlier offerings had an expert systems emphasis, but as ML advanced, in later years, the course evolved to include topics such as clustering, neural networks, statistical classifiers, graph-based models, and genetic algorithms. In 1986, Stephanopoulos published an article[43] titled, "Artificial Intelligence in Process Engineering", in which he discussed the potential of AI in process engineering and outlined a research program to realize it. Coincidentally, in the same issue, I had a article with the same title, which described the Columbia course.[44] In my article, I discussed topics from the course, and it mirrored what Stephanopoulos had outlined as the research program. (Curiously, we did not know each other at that time and had written our articles independently, yet with the same title, at the same time, with almost the same content, and had submitted to the same journal for the same issue!)

The first AIChE session on AI was organized by Gary Powers (CMU) at the annual meeting held in Chicago in 1985. The first national meeting on AI in process engineering was held in 1987 at Columbia University, co-organized by Venkatasubramanian, Stephanopoulos, and Davis, sponsored by the National Science Foundation, American Association for Artificial Intelligence, and Air Products. The first international conference, *Intelligent Systems in Process Engineering (ISPE'95)*, sponsored by the Computer Aids for Chemical Engineering (CACHE) Corporation, was co-organized by Stephanopoulos, Davis, and Venkatasubramanian, held at Snowmass, CO, in July 1995. The CACHE Corporation had also organized an Expert Systems Task Force in 1985, under the leadership of Stephanopoulos, to develop tools for the instruction of AI in chemical engineering.[45] The task force published a series of monographs on AI in process engineering during 1989–1993.

Despite impressive successes, the expert system approach did not quite take-off as it suffered from serious drawbacks. It took a lot of effort, time, and money to develop a credible expert system for industrial applications. Furthermore, it was also difficult and expensive to maintain and update the knowledge base as new information came in or the target application changed, such as in the retrofitting of a chemical plant. This approach did not scale well for practical applications (more on this in sections *Lack of impact of AI during Phases I and II* and *Are things different now for AI to have impact?*).

Phase II: Neural networks era (~1990 to ~2008)

As the excitement about expert systems waned in the 1990s due to these practical difficulties, interest in another AI technique was picking up greatly. This was the beginning of Phase II, the *Neural Networks Era*, roughly from 1990 onward. This was a crucial shift from the top-down design paradigm of expert systems to the bottom-up paradigm of neural nets that acquired knowledge automatically from large amounts of data, thus easing the maintenance and development of models.

It all started with the reinvention of the backpropagation algorithm by Rumelhart, Hinton, and Williams in 1986 for training *feedforward* neural networks to learn hidden patterns in input–output data. It had been proposed earlier, in 1974, by Paul Werbos as part of his Ph.D. thesis at Harvard. It is essentially an algorithm for implementing gradient descent search, using the chain rule in calculus, to propagate errors back through the network to adjust the strength (i.e., weights) of connections between nodes iteratively, to make the network learn the patterns. While the idea of neural networks had been around since 1943 from the work of McCulloch and Pitts, and was further developed by Rosenblatt, Minsky, and Papert in the 1960s, these earlier models were limited in scope as they could not handle problems with nonlinearity. The key breakthrough this time was the ability to solve *nonlinear function approximation* and *nonlinear classification* problems in an automated manner using the backpropagation learning algorithm.

The typical structure of a feedforward neural network from this era is shown in Figure 1, with its input, hidden, and output layers of neurons, and their associated signals, weights and biases. The figure also shows examples of nonlinear function approximation and nonlinear classification problems such networks were able to solve provided enough data were available.[46]

This novel automated nonlinear modeling ability spurred a tremendous amount of work in a variety of domains including chemical engineering.[47] Researchers made substantial progress on addressing challenging problems in modeling,[48,49] fault diagnosis,[50-55] control,[56,57] and product design.[58] In particular, the recognition of the connection between the autoencoder architecture and the nonlinear principal component analysis by Kramer,[48] and the recognition of the nature of the basis function approximation of neural networks through the WaveNet architecture by Bakshi and Stephanopoulos[49] are outstanding contributions. There were hundreds of articles in our domain during this phase and only some of the earliest and key articles are highlighted here.

While this phase was largely driven by neural networks, researchers also made progress on expert systems (such as the ASM consortium) and genetic algorithms at that time. For instance, we proposed[59] directed evolution of engineering polymers *in silico* using genetic algorithms. This led in subsequent years[60] to the multiscale model-based informatics framework called *Discovery Informatics*[61] for materials design. The discovery informatics framework led to the successful development of materials design systems using directed evolution in several industrial applications, such as gasoline additives,[62] formulated rubbers,[63] and catalyst design.[64] During this period, researchers were also beginning to realize the challenges and opportunities in multiscale modeling using informatics techniques.[65,66]

Other important advances not using neural networks included research into frameworks and architectures for building AI systems, such as blackboard architectures, integrated problem-solving-and-learning systems, and cognitive architectures. Architectures such as Prodigy and Soar are examples of this work.[67] Similarly, there was progress in process synthesis and in design,[68] domain-specific representations and languages,[32,69] domain-specific compilers,[70]

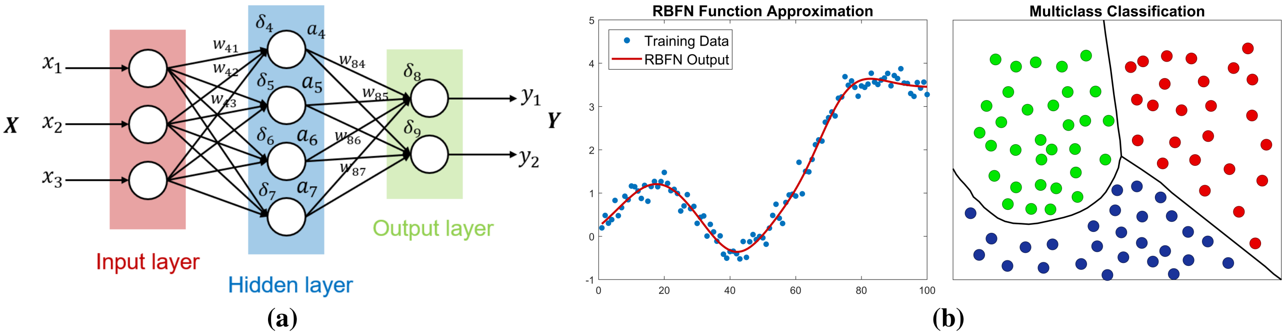

Figure 1. (a) Architecture of a feedforward neural network. (b) Examples of nonlinear function approximation and classification problems. Adapted from: https://medium.com/@curiously/tensorflow-for-hackers-part-iv-neural-network-from-scratch-1a4f504dfa8 https://neustan.wordpress.com/2015/09/05/neural-networks-vs-svm-where-when-and-above-all-why/ http://mccormickml.com/2015/08/26/rbfn-tutorial-part-ii-function-approximation/

ontologies,[71,72] modeling environments,[32,73] molecular structure search engines,[74] automatic reaction network generators,[64] and chemical entities extraction systems.[74] These references by no means constitute a comprehensive list. All this work, and others along similar lines, performed some two decades ago is still relevant and useful today in the modern era of data science. Building such systems using modern tools presents major opportunities.

Despite the surprising success of neural networks in many practical applications, some especially challenging problems in vision, natural language processing, and speech understanding remained beyond the capabilities of the neural nets of this era. Researchers suspected that one would need neural nets with many more hidden layers, not just one, but training these turned out to be nearly impossible. So, the field was more or less stuck for about a decade or so until a breakthrough arrived for training deep neural nets, thus launching the current phase which we discuss in section *Phases of AI in Chemical Engineering: Current*.

Lack of impact of AI during Phases I and II

In spite of all this effort over two decades, AI was not as transformative in chemical engineering as we had anticipated. In hindsight, it is clear why this was the case. First, the problems we were attacking are extremely challenging even today. Second, we were lacking the powerful computing, storage, communication, and programming environments required to address such challenging problems. Third, we were greatly limited by data. And, finally, whatever resource that was available was very expensive. There were three kinds of challenges in Phases I and II—conceptual, implementational, and organizational. While we made good progress on the conceptual issues such as knowledge representation and inference strategies for attacking problems in synthesis, design, diagnosis, and safety, we could not overcome the implementation challenges and organizational difficulties involved in practical applications. In short, there was no "technology push."

Further, as it turned out, there was no "market pull" either, in the sense that the low-hanging fruits in process engineering, in that period, could be picked more readily by optimization and by model-predictive control (MPC) technologies. Generally speaking, as algorithms and hardware improved over the years, these traditional approaches scaled well on problems for which we could build and solve first-principles-based models. On the contrary, problems for which such models are difficult to build (e.g., diagnosis, safety analysis, and materials design), or almost impossible to generate (e.g., speech recognition), required AI-based approaches, which required enormous computational power and voluminous data, both of which were not available during this period. This lack of practical success led to two "AI winters," one at the end of the Expert Systems era and the other at the end of the Neural Networks era, when funding for AI research greatly diminished both in computer science and in the application domains. This slowed progress even more.

In addition, it typically seems to take about 50 years for a technology to mature, penetrate, and have widespread impact, from discovery to adoption. For instance, for the simulation technology such as Aspen Plus to achieve about 90% market penetration, it took about 50 years from the time computer simulation of chemical plants was first proposed in the 1950s.[75] A similar discovery-growth-and-penetration cycle occurred in optimization as well, for mixed-integer linear programming (MILP) and mixed-integer nonlinear programming technologies, and for MPC. In retrospect, during Phase I and II, AI as a tool was only about 10–15 years old. It was too early to expect widespread impact.

This analysis suggests that one could expect wider impact around 2030–35. While predicting technology penetration and impact is hardly an exact science, this estimate nevertheless seems reasonable given the current state of AI. As it turns out, for those of us who started working on AI in the early-1980s, we were much too early as far as impact is concerned, but it was intellectually challenging and exciting to attack these problems. Many of the intellectual challenges, such as developing hybrid AI methods and causal model-based AI systems, are still around, as I shall discuss.

Are things different now for AI to have impact?

The progress of AI over the last decade or so has been very exciting, and the resource limitations mentioned above are largely gone now. Implementational difficulties have been greatly diminished. Organizational and psychological barriers are also lower now, as people have started to trust and accept recommendations from AI-assisted systems, such as Google, Alexa, and Yelp, more readily and for a variety of tasks. Companies are beginning to embrace organizational and work-flow changes to accommodate AI-assisted work processes.

It is both interesting and instructive to make the following comparison. In 1985, arguably the most powerful computer was the CRAY-2 supercomputer. Its computational speed was 1.9 gigaflops and it consumed 150 kW of power. The

16 million dollar machine (about $32 million in today's dollars) was huge, and required a large, customized, air-conditioned environment to house it.

So, what would CRAY-2 look like now? Well, it would look like the Apple Watch (Series 1). In fact, the Apple Watch is more powerful than the CRAY-2 was. The Apple Watch performs at 3 gigaflops, while consuming just 1 W of power—and it costs $300! That is a 150,000-fold performance/unit-cost gain, just on the hardware side. There have been equally dramatic advances in software—in the performance of algorithms and in high-level programming environments such as MATLAB, Mathematica, Python, Hadoop, Julia, and TensorFlow. Gone are the days when we had to program in Lisp for weeks to achieve what can now be accomplished in a few minutes with a few lines of code. We have also seen great progress in wireless communication technologies. The other critical development is the availability of tremendous amounts of data, "big data", in many domains, which made possible the stunning advances in ML (more on this below). All this is game-changing.

What accounts for this progress? Basically, Moore's law continued to deliver without fail for the last 30 years, far outlasting its expected lifespan, making these stunning advances possible. As a result, the "technology push" is here. The "market pull," I believe, is also here because much of the efficiency gains that could be accomplished using optimization and MPC technologies have largely been realized. Hence, for further gains, for further automation, one must go up the value chain, and that means going after challenging decision-making problems that require AI-assisted solutions. So, now we have a "technology push-market pull" convergence.

Looking back some 30 years from now, history would recognize that there were three early milestones in AI. One is Deep Blue defeating Gary Kasparov in chess in 1997, the second Watson becoming *Jeopardy* champion in 2011, and the third is the surprising win by AlphaGO in 2016. The AI advances that made these amazing feats possible are now poised to have an impact that goes far beyond game playing.

Phases of AI in Chemical Engineering: Current

Phase III: Deep learning and the data science era (~2005 to present)

In my view, we entered Phase III around 2005, the era of *Data Science* or *Predictive Analytics.* This new phase was made possible by three important ideas: deep or convolutional neural nets (CNNs), reinforcement learning, and statistical ML. These are the technologies that are behind the recent AI success stories in game playing, natural language processing, robotics, and vision.

Unlike neural nets of the 1990s, which typically had only one hidden layer of neurons, deep neural nets have multiple hidden layers, as shown in Figure 2. Such an architecture has the potential to extract features hierarchically for complex pattern recognition. However, such *deep networks* were nearly impossible to train using the backpropagation or any gradient descent algorithm. The breakthrough came in 2006[76,77] by using a layer-by-layer training strategy coupled with considerable increase in processing speed, in the form of graphics processing units. In addition, a procedure called *convolution* in the training of the neural net[78] made such feature extraction feasible. Convolution is a filtering technique, well-known in the domain of signals processing, for extracting features from a noisy signal. After the initial specification of the network architecture and the filter parameters such as the size and number of filters, a CNN learns during training, from a very large data set—and this is a crucial requirement—the appropriate filters that lead to a successful performance by the network.[79-81]

Another architectural innovation was the *recurrent* neural network.[82] A feedforward neural network has no notion of temporal order, and the only input it considers is the current example it has been shown. This is not appropriate for problems which have sequential information, such as time series data, where what comes next typically depends on what has gone before. For instance, to predict the next word in a sentence one needs to know which words came before it. Recurrent networks address such problems by taking as their input not just the current input example, but also what they have seen previously. Since the output depends on what has occurred before, the network behaves as if it has "memory." This "memory" property was further enhanced by another architectural innovation called the *long short-term memory* (LSTM) unit. The typical LSTM unit consists of a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. LSTM networks are well suited for making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

While the key advances here are in the architecture and training of large-scale deep neural networks, the second important idea*, reinforcement learning*, can be thought of as a scheme for learning a sequence of actions to achieve a desired outcome, such as maximizing an objective function. It is a goal-oriented learning procedure in which an agent learns the desired behavior by suitably adapting its internal states based on the reward-punishment signals it receives iteratively in response to its dynamic interaction with the environment. A simple example is the strategy one uses to train a pet, say a dog, where one rewards the pet with a treat if it learns the desired behavior and punishes it if it does not. When this is repeated many times, one is essentially reinforcing the reward-punishment patterns to the pet until it adopts the desired behavior. This feedback control-based learning mechanism is essentially Bellman's dynamic programming in modern ML garb.[83]

For this approach to work well for complex problems, such as the game of Go, one literally needs millions of "training sessions." AlphaGo played millions of games against itself to learn the game from scratch, accumulating thousands of years' worth of human expertise and skill during a period of just a few days.[13,14] As stunning as this accomplishment is, one must note that the game playing domain has the envious property that it can provide almost unlimited training data over unlimited training runs with a great deal of accuracy. This is typically not the case in science and engineering, where one is data-limited even in this "big data" era. But this limitation might be overcome whenever the source of the data is a computer simulation, as in some materials science applications.

For the sake of completeness, it is important to point out that reinforcement learning differs from the other dominant learning paradigms–supervised and unsupervised learning. In supervised learning, the system learns the relationship between input (X) and output (Y) given a set of input–output (X-Y) pairs. On the other contrary, in unsupervised learning, only a set of X is given with no labels (i.e., no Y). The system is

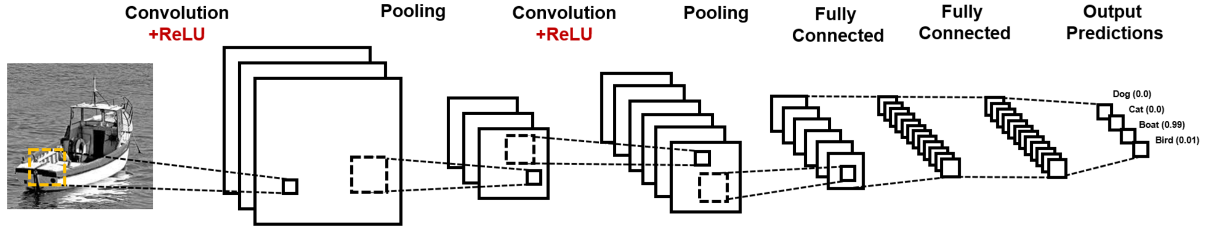

Figure 2. Typical convolutional neural network (CNN) architecture.
(Adapted from https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/)

supposed to discover the regularities in the data on its own, hence "unsupervised." One could say that unsupervised learning looks for similarities among entities so that it can cluster them into various groups of similar features, while supervised learning looks for differences between them. For instance, given a data set of animal features with no labels, unsupervised learning could lead a system to group horses and zebras together as a single class. Supervised learning, on the other hand, would have a data set with the labels of "horse" and "zebra," and so would look for features to aid in the differentiation between these two classes, such as the zebra's black and white stripes.

The third key idea, statistical ML, is the combination of mathematical methods from probability and statistics with ML techniques. This led to a number of useful techniques such as LASSO, Support Vector Machines, random forests, clustering, and Bayesian belief networks.

These three advances are the main drivers, enabled by the dramatic progress in hardware and communication capabilities, and the availability of copious amounts of data, behind the current AI "revolution" in game playing, image processing, and autonomous vehicles.

But what does this progress mean for chemical engineering? Is the promise of AI in chemical engineering here, finally, after three decades of valiant attempts and dashed hopes? What would it take to develop a "Watson"-like system for chemical engineering applications? Before addressing such questions, we first need to discuss issues concerning knowledge modeling in the AI era.

AI's Role in Modeling Knowledge: From Numeric to Symbolic Structures and Relationships

As Phase III advances, concepts and tools from AI become ubiquitous, we enter a transformative era in the acquisition, modeling, and use of knowledge. We have already seen a preview of this in Phases I and II,[84] but it has yet to become widespread.

To appreciate how and where AI fits into chemical engineering, one needs to view it from the perspective of different knowledge modeling paradigms. Historically, chemical engineering was largely an empirical and heuristic discipline, lacking quantitative, first-principles-based, modeling approaches. That all changed with the beginning of the Amundson era in the 1950s, when *applied mathematical* methods, particularly linear algebra, ordinary differential equation (ODE), and partial differential equation (PDE), were introduced to develop first-principles-based models of *unit operations.* Similarly, *decision-making* in PSE was also largely empirical and heuristics-based. That changed in the 1960s with the next important turning point in modeling—the development of *mathematical programming* methods such as MILP, in which Roger Sargent played the leading role. The next significant development in this long arc of modeling paradigms, in my opinion, is the introduction of knowledge representation concepts and search techniques from AI. This arguably started in the early 1980s, as noted, under the leadership of Westerberg, Stephanopoulos, and others. After remaining in the background as a fringe activity for the past three decades, pursued by only a small number of researchers, this knowledge modeling paradigm is now going mainstream.

Broadly speaking, one might consider the Amundson era as the introduction of formal methods for modeling the *process units.* The Sargent era and the AI era are about modeling the *process engineer*—that is, modeling 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, the Sargent approach, but others, such as fault diagnosis and process hazards analysis, require causal models-based reasoning, and are better addressed by AI concepts and techniques (more on this below).

We should not forget that the conceptual breakthrough of representing, and reasoning with, symbolic structures and relationships is an important contribution of AI. Even though this crucial aspect of AI has largely been missed in all the current excitement about ML, I expect it to resurface as we go beyond purely data-driven models toward more comprehensive symbolic-knowledge processing systems. This has far reaching consequences as we develop hybrid AI systems that combine first-principles with data-driven processing, causal models-based explanatory systems, domain-specific knowledge engines, and so forth.

Thus, I do not view AI methods as merely useful tools to extract patterns from large volumes 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 [DAEs]), then operations research (i.e., math programming), and now AI. 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 seem to suggest that probability, statistics, and network science are perhaps better suited. The truth might lie in some combination of both, depending on the application.

Chemical engineers have always prided themselves on their modeling capabilities, but, in this new era, modeling would go beyond DAEs, the staple of chemical engineering models—the Amundson legacy. Addressing the formidable modeling challenges we face in the automation of symbolic reasoning and higher-levels of decision-making would require a broader approach to modeling than chemical engineers were used to. There is a wide variety of knowledge representation concepts leading to other classes of models[85] that would play an important role in this new era. While it is not the purpose of this article to extensively discuss modeling concepts, it is, nevertheless, useful for our theme to outline and summarize the issues involved.

One may broadly classify models into (1) mechanism-driven models based on first-principles and (2) data-driven models. Each of these classes may be further categorized into (1) quantitative and (2) qualitative models. Combinations of these classes lead to hybrid models.

DAE models are suitable for a certain class of problems that are amenable to such a mathematical description—chemical engineering has abundant examples of this class in thermodynamics, transport phenomena, and reaction engineering. 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 problem often requires a hybrid model, such as a combination of a graph-theoretical model (e.g., signed digraphs) or a production system model (e.g., rule-based representations) and a data-driven model (e.g., principal component analysis [PCA] or neural networks).[86]

Thus, while we are quite familiar with ODE/PDE, statistical regression, and mathematical programming models, we are less so with other classes which are widely used in AI. These include: graph-theoretical models (as noted, used extensively to perform causal reasoning in the identification of abnormal events, diagnosis, and risk analysis.[87,88]), Petri nets (used for modeling discrete event systems),[89,90] rule-based production system models (used in expert systems for automating higher-order reasoning), semantic network models such as ontologies (used in materials discovery and design, domain-specific compilers, and so forth[71,91-94]), and 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[95,96]). In addition, there are the data-driven quantitative models such as pattern recognition-based models (e.g., neural nets, fuzzy logic), stochastic models (e.g., genetic algorithm, simulated annealing), and so forth. Recently, there has been interesting work on equation-free pattern-recognition models in the study of nonlinear dynamical systems.[97]

These AI-based modeling approaches are becoming an essential part of our modeling arsenal, in this new phase of AI. However, the number of academic researchers developing AI-based models in chemical engineering is small and woefully inadequate, considering the emerging challenges. This needs to be addressed both in our research and education agendas, as I observed a decade ago in another perspective article.[84]

It is encouraging 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, 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. This would be like learning to run a MILP program in MATLAB vs. 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 AI, ML, natural language processing, databases, and other relevant topics, to get educated deeply in AI methods. In contrast, the modern user-friendly AI software in ML, for instance, makes it quite easy for a newcomer to start building ML 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, e.g., Thiese et al.[98] and Jordan[99]), a similar fate can befall the users of ML tools.

Thus, developing AI methods is not just a matter of tracking new developments in computer science and merely applying them in chemical engineering. While there are some easy gains to be made, many intellectually challenging problems in our industry are not amenable to "open can, add water, and bring to boil" solutions. Nor can they be solved by computer scientists, as they lack our domain knowledge. This would be akin to thinking that since transport phenomena problems are solved by using differential equations, mathematicians would address these core chemical engineering problems. They can only provide us with generic concepts, tools, and techniques. It is up to us to suitably adapt and extend these with our domain knowledge to solve our problems. But to do this well one has to be well educated in AI fundamentals, going beyond merely running ML code.

It is becoming clear that our undergraduate and graduate students need to become familiar with applied AI techniques. We should develop a dual-level course, a junior/senior—1st year graduate student offering that teaches applied AI using chemical engineering examples. It would be like the applied mathematical methods core course required by chemical engineering graduate programs. We need an applied AI core, but one that goes beyond purely data-centric ML. But, teaching AI properly is much more than teaching a potpourri of ML algorithms and tools. There is a tendency to rush off and teach courses that are cookbook-style offerings, where students learn to mechanically apply different software packages without a deeper understanding of the fundamentals. The course needs to be properly founded on knowledge modeling philosophies, knowledge representation, search and inference, and knowledge extraction and management issues.

In designing such a course, we need to differentiate between training and education. Training focuses on "know-how," that is, how to execute a recipe to solve a problem, whereas education emphasizes "know-why," that is, understand 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 air-conditioner repair vs. 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 criticism, voiced in the formative years of calculus, that "how the user-friendly approach of Leibniz made it easy for people *who did not know* calculus to teach those *who will never know* calculus!" in our present context. The easy availability of user-friendly ML tools on the internet poses such a predicament.

AI in Chemical Engineering: Recent Trends and Future Outlook

The following summary is not meant to be a comprehensive review but only a representative survey that can serve as a useful starting point for interested researchers.

In Phase III (2005–Present), the emergence of largely bottom-up, data-driven strategy for knowledge acquisition and modeling using deep convolutional networks, reinforcement learning, and statistical learning has made it much easier to address problems such as image recognition and speech understanding. However, it is not entirely clear whether all this machinery is necessary to achieve results in chemical engineering. First, for these techniques to work well one needs tremendous amounts of data, which are available in game playing, vision, and speech problems, but might not be the case for several applications in chemical engineering (the exception might be those where computer simulations can generate such voluminous and reliable data). To be sure, we collect much more data now than we did before in many applications, but our domain is not a true "big data" domain like finance, vision, and speech. Second, our systems, unlike game playing, vision, and speech, are governed by fundamental laws and principles of physics and chemistry (and biology), which we ought to exploit to make up for the lack of "big data".

Thus, I would argue that many, but not all, of our needs can be met by using Phase I and/or II techniques, now updated with more powerful and easy-to-use software and hardware. Therefore, before one jumps into applying deep neural nets and/or reinforcement learning, one might examine the earlier approaches. What is truly needed is to figure out how to integrate first-principles knowledge with data-driven models to develop hybrid models more easily and reliably. Again, past work on hybrid model development is a place to start.[62,64,100,101]

While many topics in chemical engineering can benefit from the renewed interest in AI, a few stand out. They are materials design, process operations, and fault diagnosis. These areas abound in low-hanging fruit. We already have proof-of-concept solutions from Phases I and II, and the implementation challenges and organizational barriers have diminished greatly. There are similar opportunities in biomedical and biochemical engineering as well.

Industry is already using AI in process operations and diagnosis.[102] For instance, General Electric and British Petroleum (BP) use ML software to monitor oil-well performance in real time, allowing BP to maximize production and minimize downtime.[103] Uptake, the predictive analytics company, used ML to analyze dozens of failure modes, such as yaw misalignment and generator failure, in wind turbines and successfully predicted most of the failures in advance to schedule preventive maintenance.[104] Italian dairy producer Granarolo implemented ML to analyze data about sales estimates and planned promotions to forecast how much its dairy farms should produce and how to minimize wastage and maximize profits.[105] Chaffart and Ricardez-Sandoval[106] report on a hybrid multiscale thin film deposition approach that couples a neural network with a first-principles model for optimization and control. There are many more such examples. The reincarnation of the Honeywell ASM Consortium as the Clean Energy Smart Manufacturing Innovation Institute is likely to be more successful this time around as the implementational and organizational difficulties, the main roadblocks earlier, are considerably less now.

Materials design offers another opportunity, where we have the important and challenging problem of discovering and designing new materials and formulations with desired properties. This encompasses a wide variety of products such as catalysts, nano-structures, pharmaceuticals, additives, polymeric composites, rubber compounds, and alloys. The Holy Grail is to be able to design or discover materials with desired macroscopic properties rationally, systematically, and quickly, rather than by the slow and expensive trial-and-error Edisonian approach that has predominated for decades. To accomplish this, one has to solve two different but related problems: (1) the "forward" problem, where one predicts the material properties given the structure or formulation, and (2) the "inverse" problem, where one determines the appropriate material structure or formulation given the desired properties.[60]

The opportunities for AI have recently been recognized by the materials science community, which has enthusiastically advocated the "inverse design" informatics.[107-114] As noted, many elements of this framework had been anticipated and demonstrated in the chemical engineering community about two decades ago with successful industrial applications.[58,60,115,116] What is promising now is the ability to do all this more quickly and easily for more complicated materials due to the availability of powerful and user-friendly computing environments, and, of course, lots of data. This was highlighted in a recent workshop organized by The National Academy of Sciences.[112] A quick review of recent progress shows a variety of applications such as the design of crystalline alloys and organic photovoltaics,[112] nanoparticle packing and assembly,[117,118] estimates of physical properties of small organic molecules,[112] design of shape memory alloys,[119] determination of local structural features of ferritic and austenitic phases of bulk iron,[120] colloidal self-assembly,[121] simplifying model development in transition metal chemistry.[122]

There is also considerable excitement about ML for catalyst design and discovery. At a recent conference hosted by Carnegie Mellon several articles on this topic were presented.[123] Recent articles highlight some of the challenges and opportunities in catalyst design.[9-11,124] One challenge is the need for a systematic storage and access of catalytic reaction data in a curated form for common use (at least for certain classes of model reactions that can serve as testbeds to evaluate various AI-based approaches). For instance, the creation of such testbeds for computer vision was crucial to the dramatic advances accomplished with deep neural nets. The chemical engineering community also needs open-source ML software for catalyst design applications. In general, there is a sense that further progress requires such community-based efforts. Recent open-source efforts such as the Stanford Catalysis-Hub Database[125] and the Atomistic ML Package[126] are very encouraging.

Again, the relatively easy problems in materials design are those with plenty of data that can be analyzed using various ML algorithms to gain insight about process-structure–property or process-composition-property relationships. The data may come from high-throughput combinatorial experiments

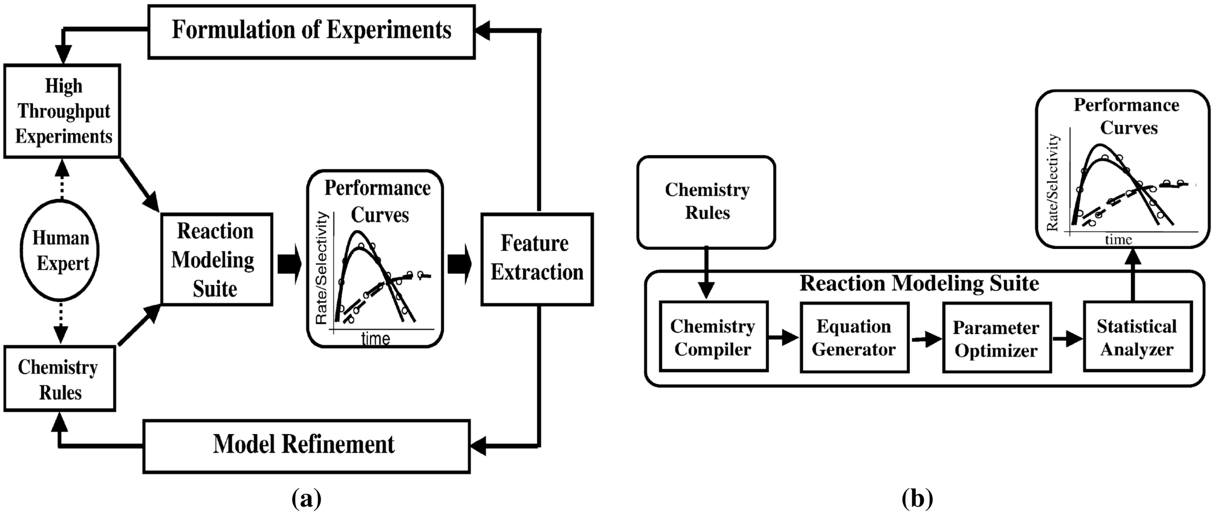

Figure 3. Automated knowledge extraction framework catalyst design. (a) Active learning framework and (b) Architecture of Reaction Modeling Suite.[61]

or high-fidelity computer simulations, or a combination of both. While there are opportunities for some quick successes, the next sought-after breakthrough—the design of a comprehensive materials discovery system using active learning, such as the one envisioned by Caruthers et al.[61] (see Figure 3, 3b is a part of 3a)—is much more intellectually challenging.

To accomplish this, one needs to develop domain-specific representations and languages, compilers, ontologies, molecular structure search engines, chemical entities extraction systems—that is, domain-specific "Watson-like" knowledge discovery engines. While this is more challenging, requiring more than a superficial familiarity with AI methods, prior proof-of-concept contributions (cited above) offer potential starting points. In this regard, the recent contributions on automatic reaction network generation and reaction synthesis planning[127-132] are promising developments.

The automation of higher-levels of decision-making highlights the need to model symbolic relationships (in addition to numeric ones) between concepts or entities, that is, an ontology. An ontology is the explicit description of domain concepts and relationships between them.[133] The entities may be materials, objects, properties, or variables. Modeling relationships is what graph and network-based models do. Most domain knowledge is in the form of a vast network of such interdependent relationships, which is what ontologies try to capture through both the syntax (i.e., structure) and semantics (i.e., meaning). Ontology is the most recent avatar in a long series of knowledge representation concepts, starting with predicate and propositional calculus in the 1940s, semantic networks in the 1960s, frames in the 1970s, and objects in the 1980s. While there has been some work recently on ontologies in process engineering,[93,94,134,135] much more remains to be done, particularly for materials design. In this regard, the material science repository, Novel Materials Discovery Laboratory,[136] and the use of text-mining to discover interesting relationships among material science concepts[137] are promising developments.

As important as data is, it is vital to remember that it is not raw data that we are after. What we desire are first-principles-based understanding and insight of the underlying phenomena that can be modeled to aid us in rational decision-making. As the oft-quoted example goes, one of the earliest success stories of data mining was the empirical discovery from point-of-sale data that people who bought diapers also bought beer in Osco Drug stores. Osco was happy to use this finding to rearrange the placement of these products and increase their sales. It was not particularly interested in finding out why this happened.

However, even in materials design, where it is perhaps adequate to discover the formulation or structure that would result in a particular property, one would still like to know *why* this works from a mechanistic perspective. A mechanistic understanding-based on fundamental principles of physics and chemistry has many benefits. There are no fundamental laws behind the diaper–beer relationship, but in physical/natural sciences and engineering there most certainly are. Science and engineering are different from marketing—for that matter, they are also different from vision and speech recognition. The same critique applies to the recognition and differentiation of a car from a cat by a deep neural network. Mechanism-based causal explanations are at the foundations of science and engineering. Thankfully, this explicability or interpretability problem is gaining more attention in mainstream AI, for experience with black-box AI is proving to be a concern because it undermines trust in the model (see articles in recent Explicable AI workshop[104]). This is an age-old concern that we encountered in the context of process diagnosis and control in the early years, and was addressed by hybrid AI approaches that combined first-principles understanding with data-driven techniques.[138] At present, we do not have a satisfactory way of embedding explanation in deep-learning systems. This could lead to disenchantment with deep-learning in favor of more transparent systems, such as Bayesian networks, in some applications.[139]

But there is an even more serious and fundamental issue at stake here. The glaring failure in purely data science models, such as deep-neural networks, is their lack of "understanding" of the underlying knowledge. For instance, a self-driving car can navigate impressively through traffic, but does it "know" and "understand" the concepts of mass, momentum, acceleration, force, and Newton's laws, as we do? It does not. Its behavior is like that of a cheetah chasing an antelope in the wild—both animals display great mastery of the dynamics of the chase, but do they "understand" these concepts? At best, the current AI systems have perhaps achieved animal-like mastery of their tasks, but they have not gained deeper "understanding" as many humans do. This naturally raises a series of very interesting and fundamental questions, starting with: What structural/functional difference between the animal and human brains makes such a deeper understanding possible for humans? Understanding this "cognition gap" is, in my opinion, one of the fundamental unsolved mysteries in AI and in cognitive science. This, of course, is not a chemical engineering problem, but it will have important consequences in our domain.

Beyond data science: Phase IV—Emergence in large-scale systems of self-organizing intelligent agents

This problem, however, has a significant systems engineering component. At the root of this problem is the following question: How do we predict the macroscopic properties and behavior of a system given the microscopic properties of its entities (i.e., going from the parts to the whole).[140] For example, a single neuron is not self-aware, but when 100 billion of them are organized in a particular configuration, the whole system is spontaneously self-aware. How does this happen? We know how to go from the parts to the whole in many cases, but not all, if the entities are non-rational or purpose-free (such as molecules). This is what statistical mechanics is all about. But, what about rational entities that exhibit purposeful, intelligent behavior?

Going from the parts to the whole is the opposite of the *reductionist* paradigm that was so successful in 20th century science. In the reductionist paradigm, one tries to understand and predict macroscopic properties by seeking deeper, more fundamental mechanisms and principles that cause such properties and behaviors. Reductionism is a top-down framework, starting at the macro-level and then going deeper and deeper, seeking explanations at the micro-level, nano-level, and at even deeper levels (i.e., from the whole to the parts). Physics and chemistry were spectacularly successful pursuing this paradigm in the last century, producing quantum mechanics, the general theory of relativity, quantum field theory, the Standard Model, and string theory. Even biology pursued this paradigm with phenomenal success and showed how heredity can be explained by molecular structures and phenomena such as the double helix and the central dogma.

But many grand challenges that we face in 21st century science are bottom-up phenomena, going from the parts to the whole. Examples include predicting a phenotype from a genotype, predicting the effects of human behavior on the global climate, predicting the emergence of economic inequality, and predicting consciousness and self-awareness, quantitatively and analytically. Reductionism, by its very nature, cannot help us solve these problems because addressing this challenge requires the opposite approach, going from the parts to the whole. In addition, reductionism, by the very nature of its inquiry, typically does not concern itself with *teleology* or purposeful behavior. Modeling bottom-up phenomena, in contrast, requires addressing this important feature because teleology-like properties often emerge at the macroscopic levels, either explicitly or implicitly, even in the case of purpose-free entities. Therefore, we need a new paradigm, a bottom-up analytical framework, a *constructionist* or *emergentist* approach, the opposite of the reductionist perspective, to go from the parts to the whole.

By a bottom-up constructionist framework, I do not mean the mere discovery of hidden patterns, such as when a deep-learning neural network discerns complex statistical correlations in huge amounts of data. I mean the need for a comprehensive mathematical framework that can explain and predict macroscopic behavior and phenomena from a set of fundamental principles and mechanisms, a theory that not only can predict the important qualitative and quantitative emergent macroscopic features but can also explain why and how these features emerge (and not some other outcomes). The current deep-learning AI systems lack this ability. Developing such a theoretical framework ought to be the ultimate goal of AI. It is a long way off, but we need systems that can reason using first-principles-based mechanisms as the first step. I consider self-organization and emergence as the next crucial phase in AI.

Summary

Having participated in the first two phases of the AI "revolution"—expert systems in the 1980s and neural networks in the 1990s—I am a bit wary of the hype surrounding the new phase. However, for the reasons discussed above, it does seem different this time. Perhaps the conditions finally exist for AI to play a greater role in chemical engineering. We are witnessing the dawn of a new transformative era in the acquisition, modeling, utilization, and management of different kinds of knowledge.

As we develop AI-based models, it is important to recognize, as Jordan[99] and Rahimi[141] caution, that the current status of ML is more like alchemy, a collection of *ad hoc* methods. Just as alchemy became more rigorous and formal over time, and evolved into chemistry and chemical engineering, ML needs to evolve. This limitation could be offset by the use of first-principles knowledge, wherever possible, which can impose some rigor and discipline on purely data-driven models.

There are many applications, as noted, that are ready to yield quick successes in this new data science phase of AI. However, the really interesting and intellectually challenging problems lie in developing such conceptual frameworks as hybrid models, mechanism-based causal explanations, domain-specific knowledge discovery engines, and analytical theories of emergence. These breakthroughs would require going beyond purely data-centric ML, despite all the current excitement, and leveraging other knowledge representation and reasoning methods from the earlier phases of AI. They would require a proper integration of symbolic reasoning with data-driven processing. This is a long, adventurous, and intellectually exciting journey, one that we have only barely begun. The progress will revolutionize all aspects of chemical engineering.

Acknowledgments

In preparing this perspective article, I have benefited greatly from the suggestions of René Bañares-Alcántara, Kyle Bishop, Ted Bowen, Bryan Goldsmith, Ignacio Grossmann, John Hooker, Jawahar Krishnan, Yu Luo, Andrew Medford, Matthew Realff, Luis Ricardez-Sandoval, Abhishek Sivaram, George Stephanopoulos, Resmi Suresh, Lyle Ungar, Shekhar Viswanath, Richard West, and Phil Westmoreland, and of the five anonymous referees, all of whom I gratefully acknowledge. I am also grateful to my former and current students, who worked on various aspects of my AI work cited in this perspective. I am thankful to my former Ph.D. student, Yu Luo, for the cover art.

This article is based on my Roger Sargent Lecture at Imperial College of Science, Technology, and Medicine (2016), the George Stephanopoulos Retirement Symposium Lecture at MIT (2017), and the Park and Veva Reilly Distinguished Lecture at the University of Waterloo (2018). I deeply appreciate and thank the respective organizers and their institutions for the opportunities that led to this article. This article was written while I was on sabbatical leave at the John A. Paulson School of Engineering and

Applied Science at Harvard University. I greatly appreciate the warm hospitality of Frank Doyle, Rob Howe, and their colleagues in Harvard SEAS. Finally, the financial support from the Center for the Management of Systemic Risk, Columbia University, is also gratefully acknowledged.

Literature Cited

1. Markoff J. Scientists worry machines may outsmart man. The New York Times, 2009.
2. Russell S, Norvig P. *Artificial Intelligence: A Modern Approach*. Prentice Hall; 2009.
3. Hawking S, Russel S, Tegmark M, Wilczek F. Transcendence looks at the implications of artificial intelligence - but are we taking AI seriously enough?. The Independent, 2014.
4. Bostrum, N. *Superintelligence: Paths, Dangers, Strategies*, Oxford, U.K.: Oxford University Press; 2014.
5. Gardels N. The great artificial intelligence duopoly, The Washington Post, 2018.
6. Chui M. Manyika J, Miremadi M, Henke N, Chung R, Nel P, Malhotra S, Chui M. Notes from the AI frontier: insights from hundreds of use cases. McKinsey Global Institute Discussion Paper. Available at www.mckinsey.com/mgi. Accessed on December 5, 2018, 2018.
7. Panetta K. 5 trends emerge in the Gartner hype cycle for emerging technologies, 2018.
8. Lee JH, Shin J, Realff MJ. Machine learning: overview of the recent progresses and implications for the process systems engineering field. *Comput Chem Eng*. 2018;114:111-121.
9. Goldsmith BR, Esterhuizen J, Liu J-X, Bartel CJ, Sutton C. Machine learning for heterogeneous catalyst design and discovery. *AIChE J.* 2018;64:2311-2323.
10. Kitchin JR. Machine learning in catalysis. *Nat Catal.* 2018;1: 230-232.
11. Medford AJ, Kunz MR, Ewing SM, Borders T, Fushimi R. Extracting knowledge from data through catalysis informatics. *ACS Catal.* 2018;8:7403-7429.
12. Rich E. *Artificial Intelligence*. McGraw Hill; 1983.
13. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Driessche GV, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D. Mastering the game of Go with deep neural networks and tree search. *Nature*. 2016;529:484-489.
14. Silver D. Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, Lillicrap T, Simonyan K, Hassabis D. Mastering chess and shogi by self-play with a general reinforcement learning algorithm, 2017; arXiv:1712.01815v1.
15. Bañares-Alcántara R, Sriram D, Venkatasubramanian V, Westerberg AW, Rychener MD. Knowledge-based expert systems: an emerging technology for CAD in chemical engineering. *Chem Eng Prog*. 1985;81:25-30.
16. Bañares-Alcántara R, Westerberg AW, Rychener MD. Development of an expert system for physical property predictions. *Comput Chem Eng*. 1985;9:127-142.
17. Bañares-Alcántara R, Westerberg AW, Ko EI, Rychener MD. DECADE—A hybrid expert system for catalyst selection—I. expert system consideration. *Comput Chem Eng*. 1987;11:265-277.
18. Barr A, Tessler S. Expert systems: a technology before its time. *AI Expert*. 1995.
19. Gill TG. Early expert systems: where are they now? *Manage Inf Syst Q*. 1995;19:51.
20. Haskin D. Years after hype, ‘expert systems’ paying off for some, Datamation, 2003.
21. Rudd DF, Powers GJ, Siirola JJ. *Process Synthesis*. Englewood Cliffs, NJ: Prentice-Hall; 1973.
22. Siirola JJ, Rudd DF. Computer-aided synthesis of chemical process designs. From reaction path data to the process task network. *Ind Eng Chem Fundam*. 1971;10:353-362.
23. Hayes-Roth F, Waterman DA, Lenat DB. *Building Expert System*. Addison-Wesley Pub. Co; 1983.
24. Waterman DA. *A Guide to Expert Systems*. New York: Addison-Wesley; 1986.
25. Simon HA. *The Sciences of the Artificial*. MIT Press; 1968.
26. Simon HA, Newell A. Human problem solving: the state of the theory in 1970. *Am Psychol*. 1971;26:145-159.
27. Simon HA. Information processing models of cognition. *Annu Rev Psychol*. 1979;30:363-396.
28. Shortliffe EH. *Computer-Based Medical Consultations, MYCIN*. Elsevier; 1976.
29. Hart PE, Duda RO, Einaudp MT. PROSPECTOR—a computer-based consultation system for mineral exploration. *Math Geol*. 1978;10.
30. McDermott J. R1: a rule-based configurer of computer systems. *Artif Intell*. 1982;19:39-88.
31. Stephanopoulos G, Johnston J, Kriticos T, Lakshmanan R, Mavrovouniotis M, Siletti C. Design-kit: an object-oriented environment for process engineering. *Comput Chem Eng*. 1987;11:655-674.
32. Stephanopoulos G, Henning G, Leone H. MODEL.LA. A modeling language for process engineering—I. the formal framework. *Comput Chem Eng*. 1990;14:813-846.
33. Myers DR, Davis JF, Hurley CE. An expert system for diagnosis of a sequential, PLC-controlled operation. *Artif Intell Process Eng*. 1990; 81-122. https://doi.org/10.1016/B978-0-12-480575-0.50007-1.
34. Kramer MA. Malfunction diagnosis using quantitative models with non-Boolean reasoning in expert systems. *AIChE J*. 1987;33:130-140.
35. Kramer MA, Palowitch BL. A rule-based approach to fault diagnosis using the signed directed graph. *AIChE J.* 1987;33:1067-1078.
36. Rich SH, Venkatasubramanian V. Model-based reasoning in diagnostic expert systems for chemical process plants. *Comput Chem Eng*. 1987;11:111-122.
37. Rich SH, Venkatasubramanian V. Causality-based failure-driven learning in diagnostic expert systems. *AIChE J.* 1989;35:943-950.
38. Pearl J, Mackenzie D. *The Book of Why: The New Science of Cause and Effect*. Basic Books; 2018.
39. Venkatasubramanian V, Suryanarasimham KV. Causal modeling for process safety analysis. In: *AIChE National Meeting,* 1987.
40. Bañares-Alcántara R, King JMP, Ballinger GH. Égide: a design support system for conceptual chemical process design. *AI System Support for Conceptual Design*. London: Springer; 1996:138-152.
41. Nimmo L. Adequately address abnormal operations. *Chem Eng Prog*. 1995;91:36-45.
42. CESMII, Clean Energy Smart Manufacturing Innovation Institute. Available at: https://www.cesmii.org/. Accessed on December 5, 2018.
43. Stephanopoulos G. Artificial intelligence in process engineering: a research program. *Chem Eng Educ.* 1986;182-185.
44. Venkatasubramanian V. Artificial intelligence in process engineering: experiences from a graduate course. *Chem Eng Educ*. 1986;188-192.
45. Kantor JC, Edgar TF. Computing skills in the chemical engineering curriculum. In: Carnahan B, ed. *Computers in Chemical Engineering Education*. Austin, TX: CACHE; 1996.
46. Gupta A. Introduction to deep learning: part 1. *Chem Eng Prog*. 2018;22-29.
47. Bulsari AB. *Neural Networks for Chemical Engineers*. Elsevier Science Inc.; 1995.
48. Kramer MA. Nonlinear principal component analysis using autoassociative neural networks. *AIChE J.* 1991;37:233-243.
49. Bakshi BR, Stephanopoulos G. Wave-net: a multiresolution, hierarchical neural network with localized learning. *AIChE J.* 1993;39:57-81.
50. Venkatasubramanian V. Inexact reasoning in expert systems: a stochastic parallel network approach. In: *Proceedings of the Second Conference on Artificial Applications*, pp. 191–195, 1985.
51. Venkatasubramanian V, Chan K. A neural network methodology for process fault diagnosis. *AIChE J.* 1989;35:1993-2002.
52. Ungar LH, Powell BA, Kamens SN. Adaptive networks for fault diagnosis and process control. *Comput Chem Eng*. 1990;14:561-572.
53. Hoskins JC, Kaliyur KM, Himmelblau DM. Fault diagnosis in complex chemical plants using artificial neural networks. *AIChE J.* 1991; 37:137-141.
54. Qin SJ, McAvoy TJ. Nonlinear PLS modeling using neural networks. *Comput Chem Eng*. 1992;16:379-391.
55. Leonard JA, Kramer MA. Diagnosing dynamic faults using modular neural nets. *IEEE Expert*. 1993;8:44-53.
56. Basila MR, Stefanek G, Cinar A. A model—object based supervisory expert system for fault tolerant chemical reactor control. *Comput Chem Eng*. 1990;14:551-560.
57. Hernández E, Arkun Y. Study of the control-relevant properties of backpropagation neural network models of nonlinear dynamical systems. *Comput Chem Eng*. 1992;16:227-240.

58. Gani R. Chemical product design: challenges and opportunities. *Comput Chem Eng*. 2004;28:2441-2457.
59. Venkatasubramanian V, Chan K, Caruthers J. Designing engineering polymers: a case study in product design. In: *AIChE Annual Meeting, Miami, FL*, 1992.
60. Venkatasubramanian V, Chan K, Caruthers JM. Computer-aided molecular design using genetic algorithms. *Comput Chem Eng*. 1994; 18:833-844.
61. Caruthers JM, Lauterbach JA, Thomson KT, Venkatasubramanian V, Snively CM, Bhan A, Katare S, Oskarsdottir G. Catalyst Design: Knowledge Extraction from High Throughput Experimentation. In *Understanding Catalysis from a Fundamental Perspective: Past, Present, and Future*. Bell, A., Che, M. and Delgass, W.N., Eds., Journal of Catalysis, vol. 216/1-2, 2003; pp. 98-109.
62. Sundaram A, Ghosh P, Caruthers JM, Venkatasubramanian V. Design of fuel additives using neural networks and evolutionary algorithms. *AIChE J*. 2001;47:1387-1406.
63. Ghosh P, Katare S, Patkar P, Caruthers JM, Venkatasubramanian V, Walker KA. Sulfur vulcanization of natural rubber for benzothiazole accelerated formulations: from reaction mechanisms to a rational kinetic model. *Rubber Chem Technol*. 2003;76:592-693.
64. Katare S, Caruthers JM, Delgass WN, Venkatasubramanian V. An intelligent system for reaction kinetic modeling and catalyst design. *Ind Eng Chem Res*. 2004;43:3484-3512.
65. de Pablo JJ. Molecular and multiscale modeling in chemical engineering - current view and future perspectives. *AIChE J*. 2005;51: 2371-2376.
66. Prasad V, Vlachos DG. Multiscale model and informatics-based optimal design of experiments: application to the catalytic decomposition of ammonia on ruthenium. *Ind Eng Chem Res*. 2008;47:6555-6567.
67. Modi AK, Newell A, Steier DM, Westerberg AW. Building a chemical process design system within soar—1. Design issues. *Comput Chem Eng*. 1995;19:75-89.
68. Linninger AA, Stephanopoulos E, Ali SA, Han C, Stephanopoulos G. Generation and assessment of batch processes with ecological considerations. *Comput Chem Eng*. 1995;19:7-13.
69. Prickett SE, Mavrovouniotis ML. Construction of complex reaction systems—I. reaction description language. *Comput Chem Eng*. 1997; 21:1219-1235.
70. Hsu S-H, Krishnamurthy B, Rao P, Zhao C, Jagannathan S, Venkatasubramanian V. A domain-specific compiler theory based framework for automated reaction network generation. *Comput Chem Eng*. 2008;32:2455-2470.
71. Venkatasubramanian V, Zhao C, Joglekar G, Jain A, Hailemariam L, Sureshbabu P, Akkisetti P, Morris K, and Reklaitis GV. Ontological informatics infrastructure for chemical product design and process development. *Comput Chem Eng*. 2006;30(10-12):1482-1496.
72. Morbach J, Yang A, Marquardt W. OntoCAPE—A large-scale ontology for chemical process engineering. *Eng Appl Artif Intel*. 2007;20: 147-161.
73. Nagl M, Marquardt W. *Collaborative and Distributed Chemical Engineering*, Berlin: Springer; (2008).
74. Krishnamurthy B. *Information retrieval and knowledge management in catalyst chemistry discovery environments* [thesis], Purdue University, West Lafayette, IN, USA; 2008.
75. Evans L. The evolution of computing in chemical engineering: some perspectives and future direction. In: *CACHE Trustees 40th Anniversary Meeting*, 2009.
76. Hinton GE, Osindero S, Teh Y-W. A fast learning algorithm for deep belief nets. *Neural Comput*. 2006;18:1527-1554.
77. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521: 436-444.
78. Lecun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD. Handwritten digit recognition with a back-propagation network. In: Touretzky D, ed. *Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO*. Denver, CO: Kaufmann, Morgan; 1990.
79. Karn U. An intuitive explanation of convolutional. *Neural Netw*. 2016.
80. CS231n Convolutional neural networks for visual recognition. Available at: http://cs231n.github.io/convolutional-networks/. Accessed on December 5 2018.
81. Gupta A. Introduction to deep learning: part 2. *Chem Eng Prog*. 2018;39-46.
82. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw*. 2015;61:85-117.
83. A beginner's guide to deep reinforcement learning. *Skymind*. Available at: https://skymind.ai/wiki/deep-reinforcement-learning#define.
84. Venkatasubramanian V. Drowning in data: informatics and modeling challenges in a data-rich networked world. *AIChE J*. 2009;55:2-8.
85. Ungar L, Venkatasubramanian V. *Advance reasoning architectures for expert systems*. *CACHE Corp*; 1990:110.
86. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: part I: quantitative model-based methods. *Comput Chem Eng*. 2003;27:293-311.
87. Iri M, Aoki K, O'Shima E, Matsuyama H. An algorithm for diagnosis of system failures in the chemical process. *Comput Chem Eng*. 1979; 3:489-493.
88. Vaidhyanathan R, Venkatasubramanian V. Digraph-based models for automated HAZOP analysis. *Reliab Eng Syst Saf*. 1995;50:33-49.
89. Johnsson C, Årzén K-E. Grafchart and its relations to Grafcet and Petri Nets. *IFAC Proc*. 1998;31:95-100.
90. Viswanathan S, Johnsson C, Srinivasan R, Venkatasubramanian V, Ärzen KE. Automating operating procedure synthesis for batch processes: part I. Knowledge representation and planning framework. *Comput Chem Eng*. 1998;22:1673-1685.
91. Aldea A. Bañares-Alcántara R, Bocio J, Gramajo J, Isern D, Kokossis A, Jiménez L, Moreno A, Riaño D. An ontology-based knowledge management platform. In: *Proceedings of the Workshop on Information Integration on the Web (IIWeb-03)*, pp. 177–182, 2003.
92. Marquardt W, Morbach J, Wiesner A, Yang A. *OntoCAPE: A Reusable Ontology for Chemical Process Engineering*. Springer Science and Business Media; 2009.
93. Hailemariam L, Venkatasubramanian V. Purdue ontology for pharmaceutical engineering: part I. Conceptual framework. *J Pharm Innov*. 2010;5:88-99.
94. Muñoz E, Espuña A, Puigjaner L. Towards an ontological infrastructure for chemical batch process management. *Comput Chem Eng*. 2010;34:668-682.
95. Katare S, Venkatasubramanian V. An agent-based learning framework for modeling microbial growth. *Eng Appl Artif Intel*. 2001;14:715-726.
96. Julka N, Srinivasan R, Karimi I. Agent-based supply chain management—1: framework. *Comput Chem Eng*. 2002;26:1755-1769.
97. Yair O, Talmon R, Coifman RR, Kevrekidis IG. Reconstruction of normal forms by learning informed observation geometries from data. *Proc Natl Acad Sci U S A*. 2017;114:E7865-E7874.
98. Thiese MS, Arnold ZC, Walker SD. The misuse and abuse of statistics in biomedical research. *Biochem Med*. 2015;25:5-11.
99. Jordan M. *Artificial Intelligence—The Revolution Hasn't Happened Yet*. Medium; 2018.
100. Psichogios DC, Ungar LH. A hybrid neural network-first principles approach to process modeling. *AIChE J*. 1992;38:1499-1511.
101. Thompson ML, Kramer MA. Modeling chemical processes using prior knowledge and neural networks. *AIChE J*. 1994;40:1328-1340.
102. How cognitive anomaly detection and prediction works - Progress DataRPM. Available at: https://www.progress.com/datarpm/how-it-works. Accessed on December 5, 2018.
103. Kellner T. Deep machine learning: GE and BP will connect thousands of subsea oil wells to the industrial internet. In: *GE Reports*, 2015.
104. Winick E, Uptake is putting the world's data to work. In: *MIT Technology Review*, 2018.
105. Justin T. In the world of perishables, forecast accuracy is key, In: *SupplyChainBrain*, 2014.
106. Chaffart D, Ricardez-Sandoval LA. Optimization and control of a thin film growth process: a hybrid first principles/artificial neural network based multiscale modelling approach. *Comput Chem Eng*. 2018;119:465-479.
107. Eber K. Reinventing material science, In: *Continuum Magazine| NREL*, 2012.
108. Dima A, Bhaskarla S, Becker C, Brady M, Campbell C, Dessauw P, Hanisch R, Kattner U, Kroenlein K, Newrock M, Peskin A, Plante R, Li SY, Rigodiat PF, Amaral G, Trautt Z, Schmitt X, Warren J, Youssef S. Informatics infrastructure for the materials genome initiative. *JOM*. 2016;68:2053-2064.
109. Hill J, Mulholland G, Persson K, Seshadri R, Wolverton C, Meredig B. Materials science with large-scale data and informatics: unlocking new opportunities. *MRS Bull*. 2016;41:399-409.

110. Ramprasad R, Batra R, Pilania G, Mannodi-Kanakkithodi A, Kim C. Machine learning in materials informatics: recent applications and prospects. *npj Comput Mater*. 2017;3:54.

111. Hutchinson M, Antono E, Gibbons B, Paradiso S, Ling J, Meredig B. Solving industrial materials problems by using machine learning across diverse computational and experimental data. *Bull Am Phys Soc*. 2018.

112. National Academies of Sciences, Engineering, and Medicine. Data science: opportunities to transform chemical sciences and engineering: proceedings of a workshop—in brief. Washington, DC: The National Academies Press. 2018; https://doi.org/10.17226/25191.

113. Ward L, Aykol M, Blaiszik B, Foster I, Meredig B, Saal J, Suram S. Strategies for accelerating the adoption of materials informatics. *MRS Bull.* 2018;43(9):683-689. doi:10.1557/mrs.2018.204.

114. Alberi K, Nardelli MB, Zakutayev A, Mitas L, Curtarolo S, Jain A, Fornari M, Marzari N, Takeuchi I, Green ML, Kanatzidis M, Toney MF, Butenko S, Meredig B, Lany S. The 2019 materials by design roadmap. *J Phys D Appl Phys*. 2019;52:13001.

115. Camarda KV, Maranas CD. Optimization in polymer design using connectivity indices. *Ind Eng Chem Res.* 1999;38:1884-1892.

116. Katare S, Patkar P, Ghosh P, Caruthers JM, Delgass WN. A systematic framework for rational materials formulation and design. In: *Proceedings of the Symposium on Materials Design Approaches and Experiences*, pp. 321–332, 2001.

117. Srinivasan B, Vo T, Zhang Y, Gang O, Kumar S, Venkatasubramanian V. Designing DNA-grafted particles that self-assemble into desired crystalline structures using the genetic algorithm. *Proc Natl Acad Sci U S A*. 2013; 110:18431-18435.

118. Miskin MZ, Khaira G, de Pablo JJ, Jaeger HM. Turning statistical physics models into materials design engines. *Proc Natl Acad Sci U S A*. 2016;113:34-39.

119. Xue D, Balachandran PV, Hogden J, Theiler J, Xue D, Lookman T. Accelerated search for materials with targeted properties by adaptive design. *Nat Commun*. 2016;7:11241.

120. Timoshenko J, Anspoks A, Cintins A, Kuzmin A, Purans J, Frenkel AI. Neural network approach for characterizing structural transformations by X-ray absorption fine structure spectroscopy. *Phys Rev Lett*. 2018; 120(225502).

121. Spellings M, Glotzer SC. Machine learning for crystal identification and discovery. *AIChE J.* 2018;64:2198-2206.

122. Nandy A, Duan C, Janet JP, Gugler S, Kulik HJ. Strategies and software for machine learning accelerated discovery in transition metal chemistry. *Ind Eng Chem Res*. 2018;57:13973-13986.

123. MLSE, In: *Machine Learning in Science and Engineering Conference.* Carnegie Mellon University, June 6–9, 2018. Available at https://events.mcs.cmu.edu/mlse/program-at-a-glance/.

124. Tran K, Ulissi ZW. Active learning across intermetallics to guide discovery of electrocatalysts for CO2 reduction and H2 evolution. *Nat Catal*. 2018;1:696-703.

125. Stanford Catalysis-Hub. Available at: https://www.catalysis-hub.org/. Accessed on Dec 1, 2018.

126. Khorshidi A, Peterson AA. Amp : a modular approach to machine learning in atomistic simulations. *Comput Phys Commun*. 2016;207:310-324.

127. Salciccioli M, Stamatakis M, Caratzoulas S, Vlachos DG. A review of multiscale modeling of metal-catalyzed reactions: mechanism development for complexity and emergent behavior. *Chem Eng Sci*. 2011;66:4319-4355.

128. Rangarajan S, Bhan A, Daoutidis P. Language-oriented rule-based reaction network generation and analysis: description of RING. *Comput Chem Eng*. 2012;45:114-123.

129. Gao CW, Allen JW, Green WH, West RH. Reaction mechanism generator: automatic construction of chemical kinetic mechanisms. *Comput Phys Commun*. 2016;203:212-225.

130. Goldsmith CF, West RH. Automatic generation of microkinetic mechanisms for heterogeneous catalysis. *J Phys Chem C*. 2017;121: 9970-9981.

131. Coley CW, Barzilay R, Jaakkola TS, Green WH, Jensen KF. Prediction of organic reaction outcomes using machine learning. *ACS Cent Sci*. 2017;3:434-443.

132. Coley CW, Green WH, Jensen KF. Machine learning in computer-aided synthesis planning. *Acc Chem Res*. 2018;51:1281-1289.

133. Gruber TR. A translation approach to portable ontology specifications. *Knowl Acquis*. 1993;5:199-220.

134. Suresh P, Hsu S-H, Akkisetty P, Reklaitis GV, Venkatasubramanian V. OntoMODEL: ontological mathematical modeling knowledge management in pharmaceutical product development, 1: conceptual framework. *Ind Eng Chem Res.* 2010;49:7758-7767.

135. Remolona MM, Conway MF, Balasubramanian S, Fan L, Feng Z, Gu T, Kim H, Nirantar PM, Panda S, Ranabothu NR, Rastogi N, Venkatasubramanian V. Hybrid ontology-learning materials engineering system for pharmaceutical products: Multi-label entity recognition and concept detection. *Comp. and Chem. Engg.* 2017;107(5):49-60.

136. Novel Materials Discovery Laboratory (Nomad). Available at: https://metainfo.nomad-coe.eu/nomadmetainfo_public/archive.html. Accessed on Dec 1, 2018.

137. Kim E, Huang K, Saunders A, McCallum A, Ceder G, Olivetti E. Materials synthesis insights from scientific literature via text extraction and machine learning. *Chem Mater*. 2017;29:9436-9444.

138. Vedam H, Venkatasubramanian V. PCA-SDG based process monitoring and fault diagnosis. *Control Eng Pract*. 1999;7:903-917.

139. Bartlett M, Cussens J. Integer linear programming for the Bayesian network structure learning problem. *Artif Intell.* 2017;244:258-271.

140. Venkatasubramanian V. Statistical teleodynamics: toward a theory of emergence. *Langmuir*. 2017;33:11703-11718.

141. Hutson M. AI researchers allege that machine learning is alchemy. *Science*. 2018. https://doi.org/10.1126/science.aau0577.