ANNUAL REVIEWS

Annual Review of Materials Research

Opportunities and Challenges for Machine Learning in Materials Science

Dane Morgan and Ryan Jacobs

Department of Materials Science and Engineering, University of Wisconsin–Madison, Madison, Wisconsin 53706, USA; email: ddmorgan@wisc.edu, rjacobs3@wisc.edu

Annu. Rev. Mater. Res. 2020. 50:71–103

First published as a Review in Advance on May 6, 2020

The *Annual Review of Materials Research* is online at matsci.annualreviews.org

https://doi.org/10.1146/annurev-matsci-070218-010015

Keywords

machine learning, materials informatics, materials science, model assessment, applicability domain, model errors, materials discovery, materials design, artificial intelligence

Abstract

Advances in machine learning have impacted myriad areas of materials science, such as the discovery of novel materials and the improvement of molecular simulations, with likely many more important developments to come. Given the rapid changes in this field, it is challenging to understand both the breadth of opportunities and the best practices for their use. In this review, we address aspects of both problems by providing an overview of the areas in which machine learning has recently had significant impact in materials science, and then we provide a more detailed discussion on determining the accuracy and domain of applicability of some common types of machine learning models. Finally, we discuss some opportunities and challenges for the materials community to fully utilize the capabilities of machine learning.

1. INTRODUCTION

ML: machine learning

MS&E: materials science and engineering

MVLR: multivariate linear regression

GPU: graphics processing unit

NLP: natural language processing

NN: neural network

Machine learning (ML) is playing an increasing role in our society and, more specifically, in materials science and engineering (MS&E). This review seeks to provide a brief introduction to ML and its growing roles in an array of aspects of MS&E, as well as a more detailed discussion of some of the challenges and opportunities associated with using ML for predicting materials properties and accelerating the design of new materials. We hope this review will therefore be of value for both the novice and experienced user.

ML can be defined as the use of computer systems that do not require explicit programming to learn about the task they are completing. ML falls into two major categories: unsupervised and supervised learning. Unsupervised ML learns properties of data without any human guidance: for example, putting data into groups (clustering) or finding dominant directions of data variation in high-dimensional space (principal component analysis). These unsupervised methods have the advantage of being able to analyze data with no need for humans to explicitly label the data, which is often a time- and resource-intensive endeavor. In contrast, supervised ML uses labeled data to learn a relationship between an output Y and an input X and is supervised in the sense that it must be told the values of Y and the corresponding values of X. This type of learning includes traditional regression [e.g., multivariate linear regression (MVLR)] and more recent methods such as deep learning (discussed below) to find objects in an image. Supervised learning typically requires human input to label the data (e.g., labeling objects in an image), although sometimes the computer can generate labels itself (e.g., from a simulation).

Tools and applications of ML have undergone an extremely rapid growth in the past approximately 20 years, with a series of stunning achievements that have been widely reported, including ML algorithms exhibiting superhuman capability at chess (1), Go (2), poker (3, 4), Jeopardy! (5), and other computationally demanding tasks such as image recognition, autonomous driving, and real-time language translation. Many of these capabilities were until recently thought to be grand challenges likely to remain inaccessible for decades (6). A detailed discussion of the causes of this transformation is beyond the scope of this review but likely involves a confluence of ever-increasing computing power [e.g., graphics processing units (GPUs)], the exploding scale and accessibility of data (e.g., cloud resources), and multiple significant algorithmic advancements with quite general applicability [e.g., deep learning for images and natural language processing (NLP)]. These influences have now become self-reinforcing, with computing, data, and algorithms all taking advantage of, and driving innovations in, the other areas to enable increasingly impressive applications. For instance, in 2016, Google announced it had deployed tensor processing unit chips specifically designed to enable fast training of deep neural networks (NNs) and demonstrated improved performance by a factor of 15 to 30 compared to leading GPU technology (7). Another example is the development of neuromorphic chips (e.g., the TrueNorth chip from IBM and the Pohoiki Beach chip from Intel), which intrinsically have an NN-like functionality and, when compared to standard central processing unit and GPU chips, can be dramatically more efficient in terms of power consumption.

The potential impact of ML, particularly in critical and economically massive areas such as high-tech businesses, manufacturing, national defense, and health care, has led to resources being committed to develop the ML ecosystem (computing, data, algorithms, and software) on the scale of billions of dollars per year in the United States and other countries. These resources have created an extraordinary opportunity for MS&E researchers to benefit from this ecosystem with only modest investment, somewhat analogous to the way computational MS&E has been enabled by inexpensive commodity processors developed for other fields. In particular, open-source software implementing state-of-the-art ML algorithms is widely available—often developed by leading

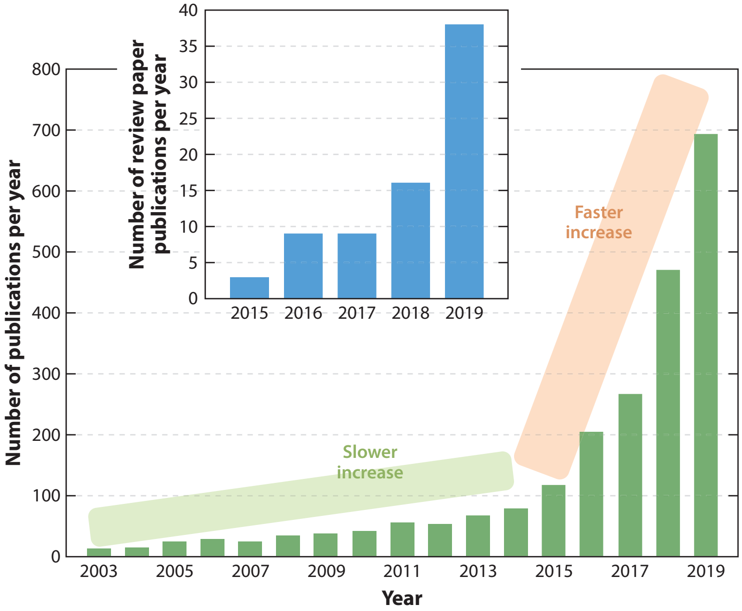


Figure 1

Growth of machine learning in materials science and engineering as seen in overall publications and review (*inset*) publications.

ML companies (e.g., Google, Facebook)—as are relatively inexpensive computing resources, including GPUs for deep learning. These techniques can be integrated with the rapidly growing world of materials data, which is being generated by new instruments and simulations and shared through new cloud-based resources. Worldwide growth of frameworks and initiatives such as Integrated Computational Materials Engineering (8–10), the Materials Genome Initiative (11, 12), the Novel Materials Discovery Laboratory, MAX (MAterials design at the eXascale), and the Materials Genome Engineering program have helped support a growing computation and data infrastructure in the MS&E community that is poised to take advantage of the new ML ecosystem.

The renaissance in computing power, data production and dissemination, and ML tools and their availability is creating very rapid growth in ML in MS&E, particularly since 2014 (**Figure 1**).[1] An examination of a logarithmic plot for the data in **Figure 1** suggests that since 2014, we have seen exponential growth of the form A(papers) × exp [*t*/B(years)], where A and B are constants and *t* represents time, suggesting a doubling about every 1.6 years. No single review can cover the broad range of areas and methods being pursued in detail, and in this review, we include both a high-level overview of areas and trends and a more detailed discussion of one specific central concern. In particular, in Section 3, we provide a brief discussion of major ML application areas in MS&E to help guide researchers attempting to understand the landscape and perhaps take first steps into a given area. In Section 4, we focus on the supervised learning models for property prediction, which is one of the most frequent uses of ML in MS&E, and describe some of the best practices for model development and assessment. Section 5 provides guidance

[1] Publications per year in materials informatics are from a Web of Science search for ("machine learning" or "artificial intelligence" or "materials informatics" or "data science") and ("materials"), scaled by 0.75 to correct for average rate of errors. Review publications per year are from a manual citation search in Google Scholar and Web of Science.

ONLINE AVAILABILITY OF DATA IN THIS REVIEW

The diffusion activation energy data set used in Section 4.4.4 is taken from the work of Lu et al. (35) and is available on Figshare (**https://doi.org/10.6084/m9.figshare.7418492**).

The data subsets used for training, validation, and testing used in this review; data used to make each original figure; and the most up-to-date supporting information document are available on Figshare (**https://doi.org/10.6084/m9.figshare.9546305**).

on our summaries of useful tools for ML in MS&E. Throughout this review, we focus on recent results and present opportunities and challenges, and then in Section 6, we offer some more speculative thoughts on longer-term future opportunities and challenges. All data associated with this review that are shared online are described, with appropriate links, in the sidebar titled Online Availability of Data in This Review. These data include catalogues of recent review papers and ML software tools shared via Figshare so they can be easily updated in the future. We include in the supplemental material a summary of useful infrastructure information for ML in MS&E (see the sidebar titled Summary of Supporting Information in the Supplemental Materials).

2. SOME NOTATION

To avoid repeating notation in multiple locations, we introduce it here and use it consistently in this review. We frequently consider supervised regression problems where we assume our data have the original form (X,Y), where X is a matrix of features and Y is a vector of target values. Commonly, each row of X corresponds to a system (e.g., a material structure and composition) to be modeled, and each element in that row is a value describing some feature of the system (e.g., amount of Cu); Y is a vector of target properties to be modeled (e.g., bandgap). X typically starts in the form of a human-relevant simple description [e.g., just composition and structure, or a simplified molecular-input line-entry system (SMILES) string], and corresponding features in a numerical form must be generated (this process is sometimes called featurization and is discussed in Section 4.2). The relationship between X and Y can be written as $Y = F(X) + \epsilon$, where ϵ is a noise term (with mean zero and variance σ^2), and we seek to use ML to construct a model for $F(X)$. We write this model as $\hat{F}(X)$ and its predictions as $\hat{Y}$. Given some new vector of descriptors, X^*, one can use the ML model to predict a corresponding single target value, $\hat{Y}^* = \hat{F}(X^*)$.

SMILES: simplified molecular-input line-entry system

In general, $\hat{F}$ can be specified by its model type, parameters, and hyperparameters. Model type refers to the overall functional forms used (e.g., linear regression or NNs). Model parameters are

SUMMARY OF SUPPORTING INFORMATION IN THE SUPPLEMENTAL MATERIALS

Supplemental material is available online from *Annual Reviews* and on Figshare (see the sidebar titled Online Availability of Data in This Review). It contains a list of recent ML in MS&E review papers (**Supplemental Section 1**), detailed lists of available ML-centric software packages useful for ML in MS&E (**Supplemental Section 2**), a short discussion of journals publishing ML in MS&E studies (**Supplemental Section 3**), a discussion of different types of commonly used ML models (**Supplemental Section 4**), a discussion on basic model checks (**Supplemental Section 5**), and a comparison of computed residuals versus model errors and the cumulative distribution of the ratio of residuals to model errors for the Gaussian process regression (GPR) and random forest model analysis presented in this review (Supplemental Section 6).

Supplemental Material >

the values that define the specific instantiation of the model and are fit during the training process (e.g., coefficients of linear terms or weights in an NN). Model parameters can generally be fit by some highly efficient method (e.g., matrix inversion for linear models or backpropagation for NNs). Model hyperparameters are similar to model parameters but cannot be easily optimized through an efficient method and are therefore typically treated separately from the model parameters and searched in a more restricted manner (e.g., with a simple grid search), with full optimization of model parameters for each evaluation of model hyperparameters. Examples of model hyperparameters include number of terms in a polynomial regression and number of layers in an NN.

KRR: kernel ridge regression

GPR: Gaussian process regression

RFDT: random forest decision tree

3. WHERE AND HOW IS MACHINE LEARNING IMPACTING MATERIALS SCIENCE AND ENGINEERING?

This section provides a high-level summary of some of the major areas in which ML is being applied in the field of MS&E. Some representative examples from recent studies are showcased in **Figure 2**.

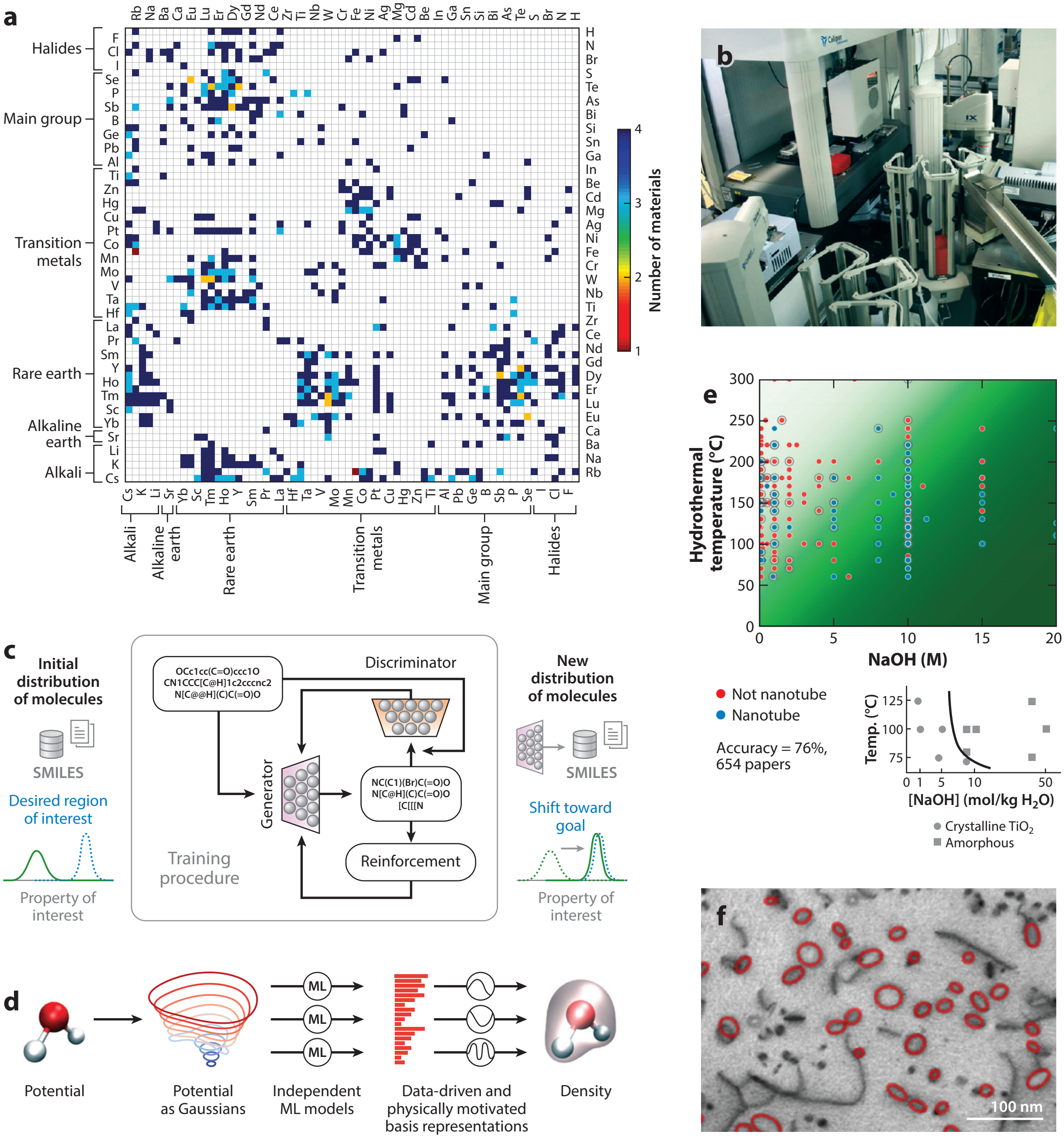
3.1. Property Prediction and Materials Discovery and Design

Two related and highly active research areas applying ML in MS&E are materials property prediction and materials discovery and design, each of which is discussed in this section.

3.1.1. Property prediction. One of the most common and easy-to-understand uses of ML in MS&E is predicting new materials data from existing databases through regressing Y on X followed by prediction of $\hat{Y}^* = \hat{F}(X^*)$ for new data (see Section 2 for notation). There is no unique approach to assigning feature vectors in X to represent a material, and this is a critical challenge we discuss in detail in Section 4.2. This overall approach can be used to extend almost any database to new systems, allowing prediction of new data, rapid exploration of large spaces, and iterative optimization to find new materials (sometimes called active learning). The use of ML in MS&E has been applied to predict myriad materials properties for many classes of materials. A representative but not exhaustive list of recent studies includes the prediction of bulk stability of perovskite oxides, garnet oxides, and elpasolites (13–16); formability of novel ternary compounds (17, 18); superconducting critical temperatures of complex oxides (19, 20); melting points of unary and binary solids (21); dielectric properties of perovskites and polymers (22, 23); formability of novel half- and full-Heusler intermetallic compounds (24, 25); casting size of metallic glass alloys (26); electronic bandgap of different classes of inorganic materials, such as oxides and covalent semiconductors (27–30); stability and bandgap of halide perovskites for solar cells (31–33); dilute metal element solute diffusion barriers in an array of metallic hosts (34, 35); electromigration of impurity elements in metals (36); scintillator materials (37); and piezoelectric materials with high electrostrains (38). **Figure 2*a*** shows an example heat map detailing the number of newly discovered ternary oxide materials across chemical space; predictions were obtained by using ML to inform the probability a ternary oxide will form (17). When too little data are available for regression, clustering can still provide a tool by grouping similar materials based on their features. To the extent that these groups share properties, such a clustering can provide powerful predictions, and some uses for finding phase diagrams and allotropes are summarized in the review from Ramprasad et al. (39). Some effective and widely used regression methods employed in the materials data studies listed above include MVLR (36), kernel ridge regression (KRR) (14, 32), Gaussian process regression (GPR) (27, 38), ensemble methods such as random forest decision trees (RFDTs) and gradient boosted

Supplemental Material >

regression (24, 31, 33), and both basic and deep learning NNs (13, 14, 34, 40). For readers less familiar with these different ML methods, we have included an introductory discussion of these different model types in **Supplemental Section 4**; this topic is also mentioned in Section 4.3. In addition, more detailed information on these general ML methods is covered in References 41–44.



(*Caption appears on following page*)

Figure 2 (*Figure appears on preceding page*)

(*a*) Heat map showing number of newly discovered ternary oxide materials across chemical space. (*b*) Photograph of an autonomous synthesis and characterization robot. (*c*) Overview of usage of the objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC) model for molecular design in a simplified molecular-input line-entry system (SMILES). (*d*) Scheme to represent the Hohenberg-Kohn map using machine learning (ML) models. (*e*) Machine-learned probability of nanotube synthesis compared with experimental outcomes from text mining; darker green indicates a higher probability of synthesizing nanotubes. (*f*) Model-labeled elliptical irradiation defects in a characterized steel micrograph. Panel *a* adapted with permission from Reference 17; copyright 2010 American Chemical Society. Panel *b* republished with permission from Reference 64. Panel *c* adapted with permission from Reference 70. Panel *d* adapted with permission from Reference 127. Panel *e* adapted with permission from Reference 97; copyright 2017 American Chemical Society. Panel *f* adapted with permission from Reference 74.

Many present ML approaches for predicting structure-property-performance relationships of materials in MS&E can be viewed as part of, or emerging from, the field of study known as quantitative structure-activity relationships (QSAR) [and the closely related field of quantitative structure-property relationships (QSPR)] (45, 46). QSAR and QSPR have used data science tools for over 100 years to correlate physical and molecular properties of chemical substances and their associated properties, from biological activity to boiling point, and therefore present well-established best practices and powerful techniques that can provide excellent guidance to the MS&E community.

3.1.2. Materials discovery and design.

ML has built on its strength in property prediction (see Section 3.1.1) to enable the discovery, design, and development of novel materials spanning an array of applications and materials classes by providing a new understanding of key chemical or physical relationships governing properties of interest. As a concrete example, in the field of halide perovskites for solar photovoltaics, the use of ML on data has resulted in assessment of chemical trends (e.g., halogen content and alkali versus organic species content) on properties such as the bandgap and stability and in the prediction of promising new halide perovskite materials such as $Cs_2Au^{1+}Au^{3+}I_6$ and $NH_3NH_2InBr_3$, the former of which has been investigated in detail as a promising solar material (31–33, 47). In addition, materials data predictions from ML on a large space of Br- and Cl-based elpasolite compounds led to the discovery of numerous new promising scintillator materials and reproduced more than 20 known well-performing scintillators. In this case, insights from ML provided rational material composition changes to realize a favorable placement of the Ce^{3+} 4f and 5d levels within the material bandgap, a necessary design criterion for scintillators (37).

In some cases, ML is used as an integral guide to the data collection effort, such as in active learning, where iterative design of experiments (or simulations) is performed using ML property models and carefully tuned optimization approaches (38, 48–53). More specifically, active learning is a method that seeks to balance exploitation of information contained in existing data in an ML model (i.e., data points with the best predictions) and exploration of less-sampled portions of the design space (i.e., data points likely to have high model uncertainties). Active learning is used to obtain a target outcome as efficiently as possible by first quickly sampling potential regions of interest to construct an initial ML model, followed by adaptive sampling of the exploitation-exploration trade-off to maximize the expected improvement of the ML model for finding the target, thus optimizing the experimental objective (e.g., finding a new material with the highest electronic bandgap) with the smallest number of measurements. Active learning methods have yielded numerous success stories, such as a new Pb-free piezoelectric material with the largest measured electrostrain in the $BaTiO_3$ family (38) and new polymers with high glass transition temperatures, the latter result being obtained by starting from a remarkably small training data set of just five materials (54).

QSAR: quantitative structure-activity relationships

QSPR: quantitative structure-property relationships

An exciting and fairly new area for materials discovery using ML is the integration of autonomous high-throughput experimentation conducted by robots with on-the-fly decision making guided by ML model predictions made using active learning techniques (55–64). This integration has the potential to perform guided exploration of large materials spaces with limited to no human intervention, greatly accelerating rates of materials discovery as well as potentially supporting work with materials or in environments that are inhospitable to humans (58) and reducing human biases in materials searches (58, 62). These approaches have had some notable recent successes. Duros et al. (62) explored new approaches for the synthesis and crystallization of a new polyoxometalate compound and demonstrated that the purely machine-based search covered a parameter space to realize crystallization about six times larger than that explored by humans, with an accuracy about 5% higher than that obtained by humans in predicting whether the compound will crystallize. Granda et al. (55) demonstrated an ML-guided organic synthesis robot that was able to predict the outcome of untested chemical reactions with greater than 80% accuracy and then was able to construct prioritized lists of new reactions to attempt based on their evaluated likelihood to produce the desired products. A further outcome of this work was the identification of unusual reaction mixes, which were later evaluated by human researchers, leading to the discovery of previously unknown chemical reactions. Finally, Nikolaev et al. (59) developed a robot scientist named the Autonomous Research System (ARES) that specialized in the autonomous growth and characterization of carbon nanotubes, a model problem due to its complex coupling of synthesis and processing to resulting structure-property relationships (e.g., example nanotube diameter, helicity, and the effects of these parameters on the nanotube electronic properties). **Figure 2*b*** contains a photograph showing the lab setup of the ARES instrument. ARES successfully optimized nanotube synthesis in a high-dimensional design space and determined the correct parameters to maintain accurate growth rate control, thus demonstrating the potential utility and possible disruptive potential of ML-guided robot scientists in MS&E.

GAN: generative adversarial network

A particularly interesting area is the development of new materials with generative models such as variational autoencoders and generative adversarial networks (GANs). These methods are particularly well-suited to execute the paradigm of inverse materials design, in which the desired material characteristics are first enumerated and candidate materials are suggested and evaluated on the fly (65–68). Inverse design creates the challenge of the exploration of an exceedingly large chemical space, which can be partly overcome by the use of GANs to automatically suggest and evaluate novel molecules and materials for a desired application (58). Concrete successes have already been demonstrated in this area; for example, the CrystalGAN (69) model was used to generate, screen, and subsequently discover new stable hydride compounds for solid-state hydrogen storage applications. The objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC) model was shown to be successful in predicting new high-melting-point organic molecules (70). **Figure 2*c*** shows a schematic of the ORGANIC model, which consists of separate generator (discriminator) NNs used to suggest new molecular structures (predict desired molecular properties), where the generation of new candidate molecules is informed by the discriminator and the reinforcement algorithm. Finally, the reinforced adversarial neural computer was found to outperform ORGANIC and function as a valuable tool for the discovery of novel molecules for drug design and development (71).

3.2. Materials Characterization

Materials characterization tools are increasingly producing data on scales of quantity and complexity that outstrip human ability to manage and interpret, and ML methods are being used to process and analyze these data. For example, Voyles (72) recently reviewed numerous applications in electron microscopy, pointing out uses of ML in image improvement (e.g., denoising, drift and distortion correction) and analysis (e.g., spectral demixing and clustering to identify features). Several studies have recently applied deep learning machine vision techniques to electron microscopy images [e.g., to cluster materials based on microstructure images (see references in 73) and to identify defects in images (74, 75)], in multiple cases with apparently human levels of accuracy. For example, **Figure 2*f*** shows dislocation loops in electron micrographs of a steel alloy identified by a deep learning model, the accuracy of which was as good as or better than that of domain-specific expert humans (74). ML has also been applied to X-ray diffraction data [e.g., using deep learning to accurately perform identification of space-group, extinction-group, and crystal-system from X-ray powder diffraction patterns (76)]. Other intriguing examples have shown how ML could replace more challenging measurements or calculations with simpler ones. As an experimental example, Stein et al. (77) demonstrated that a variable autoencoding approach could quite accurately reproduce UV-visible spectra from simple images of a thin film generated with a commercial scanner. In simulation, Combs et al. (78) recently demonstrated that an MVLR model could correlate low- and high-fidelity scanning tunneling electron microscopy image modeling, allowing approximations to full multislice simulations of nanoparticles millions of times faster than a full scanning tunneling electron microscopy image simulation. ML appears likely to provide many paths toward accelerated characterization through simplified experiments and computations and automated analysis, reducing time spent in traditional methods and enabling processing of the enormously large data streams coming from newer and next-generation characterization instruments.

AI: artificial intelligence

CNN: convolutional neural network

3.3. Knowledge Extraction via Text Mining

NLP tools are central to text and speech extraction and recognition, enabling artificial intelligence (AI)-related speech tools such as Apple's Siri and Amazon's Alexa and real-time language translation. Numerous open-source NLP tools currently exist, such as the Word2vec (79) and Global Vector (GloVe) (80) packages, as well as tools to conduct sentiment analysis using deep convolutional neural networks (CNNs) (81). NLP, text extraction, and sentiment analysis (i.e., the characterization of subjective information such as opinions, communicated through text) have seen widespread use—for instance, in computational biology and biomedical research (82, 83), genetics (84), health care (85), and social science (86)—but work has been much more limited in materials.

A basic NLP analysis in MS&E can be considered in three steps. First, one maps words to real-valued vectors, a process called embedding, which can be done with unsupervised learning and requires significant time and large data sets. However, once completed, such embeddings can be reused in many applications, and multiple MS&E-specific embeddings are already available (87–89). Given an embedding, the second step is to train an NLP model to recognize target information using embeddings, typically with supervised training on a set of expert-annotated sentences, some of which are now being made open-source to encourage democratization of the NLP ML model-training process (90). The second step treats the sentence as a sequence of words, which are converted into a sequence of vectors, and the model is trained to predict the correct annotated categories (e.g., identify text "Fe" as category "metal") on the training data. Models are typically recursive, convolutional, or transformer NNs (91–93). A common third step then applies grammar rules to understand connections between identified words in a dependency parse tree (e.g., allowing one to determine if a given value is describing a given property in a sentence) (94).

Software packages such as ChemDataExtractor (95) are being developed to enhance standard NLP approaches with the ability to parse unstructured text in complex scientific publications—for example, chemical formulas or domain-specific words or abbreviations (e.g., the meaning of the

DFT: density functional theory

MLP: machine learning potential

term UV-Vis spectroscopy). One area in which text mining is playing an increasingly large role in MS&E is synthesis (96). Recent text extraction studies have resulted in useful guidance regarding key experimental parameters needed for optimal materials synthesis, such as in creating TiO_2 nanotubes (97), as shown in **Figure 2*e***; synthesis of new perovskite materials (98); and aggregated synthesis parameters for 30 different oxide materials systems (89). These studies have also provided insights on best writing practices to facilitate efficient transfer of machine-readable knowledge (99) and understanding of best synthesis practices through graph representations (100). In contrast to the above examples, which conducted NLP using supervised methods with annotated studies as training data, Tshitoyan et al. (87) demonstrated the successful use of unsupervised techniques in extracting structure-property and chemical relationship information and showed that these tools can aid in future materials discovery by codifying knowledge contained in past publications. Another impressive example is the recent book *Lithium-Ion Batteries: A Machine-Generated Summary of Current Research*, a review extracted from over 150 papers on Li-ion batteries that was generated by an ML model (101). This work suggests a future in which information aggregation in topical reviews could be automatically delivered in a very human-understandable form, significantly accelerating the process of learning new areas.

3.4. Machine Learning for Molecular Simulation

Another area in which ML has had a large impact on MS&E is molecular simulation. Here, ML has been widely applied to develop interatomic potentials and to improve and accelerate ab initio simulations, each of which is discussed in this section.

3.4.1. Interatomic potentials. Atomistic-scale simulations of molecules and condensed phases typically find the interaction between classically treated nuclei through Hamiltonians that are based on either approximate solutions to the Schrödinger equation for electrons or coarse-grain quantum electronic effects to develop an effective interatomic potential. Interatomic potentials are typically about 10^3–10^6 times faster than common quantum methods [e.g., density functional theory (DFT)], but finding and parametrizing appropriate functional forms to treat systems with complex electronic behavior (e.g., with charge transfer, bond breaking, or multiple types of hybridization) is very challenging. Replacing interatomic potential functional forms and fitting procedures with those from ML offers the alluring possibility of both greatly reducing the time and expertise required for developing potentials and perhaps enhancing their accuracy. In the past decade, many researchers have used ML to generate interatomic potentials [referred to as machine learning potentials (MLPs)], which have enabled studies of larger size and longer time than what is accessible with direct DFT (102–108). Generating an MLP is fundamentally a complex regression problem to map the potential energy surface, its derivative the force field, or both by fitting an ML model to a large training database, typically containing thousands of DFT calculations (often derived from individual time steps of ab initio molecular dynamics simulations) (103, 109–112). Constructing input features for the MLP model (sometimes called atomic structure descriptors or fingerprints) is critical and has received a lot of attention over the past decade (39, 105, 113, 114) (see Section 4.2). As a concrete example of the success of these methods, Botu & Ramprasad (103) trained a KRR model to demonstrate a large-scale acceleration of ab initio molecular dynamics calculations for bulk and surface slabs of Al. More recently, Bartók et al. (109) have shown using GPR that an MLP can accurately capture the energetics of Si surface reconstructions. Finally, Artrith and colleagues (110, 111) demonstrated that modeling systems with up to 11 elemental components with NNs is not only computational feasible but highly accurate.

Despite these and many other notable successes, challenges remain, such as the difficulty in obtaining enough high-quality DFT data to fit an MLP for complex phenomena (such as grain boundaries, surfaces, cluster defects, or other extended defects) and multiple alloying elements. An additional challenge of using MLPs is similar to that encountered with the construction of empirical potentials: namely, how to assess the chemical and physical applicability domain of the MLP and understanding when the MLP may fail (102, 110, 115, 116). Finally, MLPs are often quite slow compared to many traditional interatomic potentials (e.g., about 1–2 orders of magnitude slower) (113), so approaches that can accelerate their evaluation would broaden their utility.

3.4.2. Improving and accelerating ab initio simulations. Ab initio methods (e.g., DFT and hybrid functionals) use approximate solutions to the quantum mechanical equations of electrons to model materials systems and have become some of the most widely used tools in materials and chemical science (DFT is today used in at least 30,000 new research publications every year) (117). However, these methods suffer from limitations of accuracy and speed that significantly inhibit their use, and there have been multiple strategies to apply ML to improve and accelerate the calculation of ab initio functionals, each showing significant notable advances. One strategy has focused on improving the accuracy of DFT methods. For example, the work of Nagai et al. (118) used an NN to numerically calculate the Hartree exchange-correlation functional in an effort to improve its accuracy. Bogojeski et al. (119) found that one can efficiently learn the energy differences from DFT and coupled cluster simulations and use ML to provide a promising avenue to have coupled-cluster-level accuracy and DFT-level speed for physical situations where standard DFT is inadequate. Another strategy is to use ML to learn the computationally expensive portions of solving the Kohn-Sham equations in a DFT calculation, namely contributions to the exchange-correlation energy (120–122). To this end, Snyder et al. (120) modeled the kinetic energy of a one-dimensional system of noninteracting electrons; this analysis was then extended to more general cases (121). Mills et al. (123) and Lei & Medford (124) showed that a CNN can learn the mapping between the potential energy landscape and the resulting one-electron ground state and kinetic energies.

The second strategy is to directly learn the electron charge density itself. This strategy has the advantage of allowing one to completely bypass solving the Kohn-Sham equations and instead rely on the Hohenberg-Kohn theorems, which allow one to obtain the total energy (and other properties) directly from the charge density (125–129). **Figure 2*d*** shows a representation of using ML and charge densities to bypass the calculation of the Kohn-Sham equations. Researchers have employed diverse methods to learn the charge density directly. For example, Kajita et al. (126) proposed a method of descriptor generation based on a three-dimensional voxel representation of the electron density for use in CNNs. In contrast, Brockherde et al. (127) and Bogojeski et al. (128) formulated KRR models to directly learn the charge density using a suite of training data, and Sinitskiy & Pande (129) showed that CNNs trained on low-fidelity charge density data can learn meaningful characteristics of the charge density for a variety of organic molecule chemical environments, enabling predictions with DFT-level accuracy but orders of magnitude faster. Once sufficiently mature, these methods may fundamentally alter the way researchers conduct ab initio calculations, wherein ML fundamentally provides quantum mechanical knowledge of complex systems without needing to solve the Schrödinger equation.

4. SOME CHALLENGES AND BEST PRACTICES FOR MACHINE LEARNING IN MATERIALS SCIENCE AND ENGINEERING

In this section, we discuss issues that occur in many ML modeling projects, focusing on supervised regression learning models for property prediction, although many of the issues are similar in other applications. The key steps in an ML workflow broadly include the following:

1. Data collection and cleaning
2. Feature generation and selection (featurization or feature engineering)
3. Model type selection, fitting, and hyperparameter optimization
4. Model uncertainty assessment (e.g., performance on test data) and domain applicability
5. Final model predictions

RMSE: root mean squared error

R^2: coefficient of dependence

The ML workflow has been discussed extensively in other reviews (39, 130), and a detailed discussion of all parts is not included here. However, we do wish to discuss some critical aspects associated with steps 2–4 that we feel are valuable to help the community move toward best practices in ML modeling.

4.1. Basic Statistics of Accuracy

In many steps of supervised regression learning, ML models are assessed by some statistic related to the differences between the predicted data $\hat{Y}$ and true data Y. The equations for these statistics are widely available and are not given here, but we briefly discuss their effective use. The root mean squared error (RMSE) is a commonly used error metric and is frequently the error metric that the ML model seeks to minimize. Mean absolute error is also useful to calculate and will typically trend with RMSE, but it is less sensitive to large errors from outlier predictions and, unlike RMSE, is not smoothly differentiable, making it harder to use in some optimizations. Mean absolute percentage error (often called by many different names, including average absolute relative error) is just the absolute error as a percentage of the true data point value and is also very helpful, as the importance of an error is often related to the size of the quantity being predicted. It is also important to give RMSE errors relative to the standard deviation of the data set, which is sometimes called the reduced RMSE, as this provides a reasonable representation of the scale of the ML errors with respect to which RMSE should be measured. In particular, the reduced RMSE value for a well-performing ML model should be significantly less than 1, as simply guessing the mean of the predicted data (typically not useful) would yield a reduced RMSE equal to 1.

Another widely used metric is the coefficient of dependence, R^2, which gives the fraction of variance in the true value that is predictable from the predicted values (the parity plot, which shows predicted versus actual data, is a very useful plot and gives a graphical feel for R^2). R^2 is ≤ 1, with 1 representing perfect prediction, and can be <0 for predictions that trend with the opposite sign slope as the true values (R^2 technically has no lower bound). Reduced R^2 (sometimes referred to as adjusted R^2) is given as $R^2_{\mathrm{red}} = 1 - [(1 - R^2)(n - 1)/(n - k - 1)]$, where n is the number of observations and k is the number of features. $R^2_{\mathrm{red}} \leq R^2$ and, since it adjusts for the complexity in the model, it decreases when terms that have no predictive ability are added. R^2 gives a useful overall assessment of model quality, and generally, values >0.7 are desired for a useful model. However, R^2 can be misleading; a few widely separated regions that are fit on average can give a high R^2 even when no predictive ability within each region is given by the model.

Overall, we suggest determining at least RMSE, reduced RMSE, mean absolute error, mean absolute percentage error, R^2, and R^2_{red}, generating a parity plot as standard practice, and using the metrics most relevant for your application. RMSE is typically used for choosing the best features and models during ML model development. The exact method of choosing data for fitting and assessing a model with RMSE (or any metric) can be complicated and is described in Section 4.4.1.

4.2. Feature Engineering

Feature engineering is a key component of developing useful supervised ML models. Features must be machine readable (i.e., vectors of numbers), be practical to obtain for the desired application (e.g., they should certainly be significantly easier to obtain than the target property values), capture as much of the relevant variables controlling behavior as possible, and ideally contain limited additional information that is not useful and that may lead to overfitting data and poor predictions. Generally, feature engineering consists of two steps: feature generation and feature selection, each of which is described here.

A common set of minimal descriptors may include composition and processing conditions (e.g., precursors, annealing temperature, or gas pressure), as these can completely specify the final material, although perhaps rather indirectly. Additional characterization information can also be included (e.g., infrared or X-ray diffraction spectral data). While composition specified by weight or atomic percent is useful, it cannot be used to extrapolate to any new elements, since the model will have no knowledge of how to predict effects of that element if it has not appeared in the training database. One solution to this limitation is to represent each element with a feature vector of elemental properties (e.g., melting temperature or electronegativity). These can then be used to generate features for alloys by taking arithmetic- or composition-averaged combinations of the constituent element features—for example, constructing the composition-averaged melting point of the elements in a compound. This approach has been codified by the materials agnostic platform for informatics and exploration (131), which gives a canonical set of elemental properties and arithmetic operations that have proved successful in predicting stable compounds (14), glass-forming ability (26), and diffusion coefficient, to name a few (34, 35).

For cases in which some level of atomic structure (by which we mean atom position and element type) information can be readily determined (e.g., in atomistic modeling or organic molecule descriptions), the atomic structure forms a powerful feature set, as it is likely to play a large or even totally controlling role in setting a property of a molecule or crystal. Direct use of the atomic coordinate vectors and atom types as a feature is inadvisable, as they do not satisfy the translational, rotational, and permutation (swapping atoms of same types) symmetries of the system under study and thus likely need a very large amount of data to be trained well enough to reflect these basic symmetries. An array of different feature-generation methods have therefore been developed that do satisfy these symmetry requirements. For molecules, these methods consider properties such as bond lengths, connectivity, and functional groups and can include relative atomic position and electronic structure data computed with quantum mechanical atomistic simulations. Such properties have been widely used in QSAR/QSPR analysis. Thousands of basic QSAR/QSPR features are now available and can be extracted automatically from basic molecular formulae (e.g., SMILES strings) (see 132 for a summary of recent automated tools for QSAR/QSPR). Numerous tools have also been developed for extended systems (i.e., not just molecules) in the context of constructing ML-based potentials (see Section 3.4). These features have been validated for particular materials systems and benchmarked against key standard databases (e.g., the QM9 molecule data set), including, but not limited to, atom-centered symmetry functions (133); the smooth overlap of atomic orbitals method (134); partial radial distribution functions (135); bag of bonds (136); bonds, angles, and machine learning (137); and the many-body tensor representation (138). The streamlined production of many of these features has been implemented in the matminer code package (see **Supplemental Section 2**) (139). Another approach to feature generation is graph-based deep learning methods, which first map atomic structure onto a vector of atom descriptors (e.g., type and simple properties, such as formal charge) and bond distances and connectivity (the graph) and then merge those descriptions with weighted averaging to ensure flexible joining of the atomic descriptions with the correct bonds (140–143). These methods work from very basic information and replace the step of invoking human intuition and analysis to generate features with a more automated deep learning generation of a feature map. Finally, we note that one can work from unsymmetrized data if the method itself performs the symmetrization. For example, Nie et al. (112) recently generalized kernel regression approaches to include permutation symmetry and showed it could generate effective energy fitting directly from atomic pair distances.

Supplemental Material >

Once a set of features to represent a data set have been generated, it is common to select a representative set of features that is large enough to result in low model errors and avoid model underfitting, yet not so large as to incur penalties to overall model accuracy and extrapolative ability due to overfitting. Certain ML models such as polynomial regression and KRR can easily become confused or overfit if too many features are used, but other models, such as random forest methods (see Section 4.3; **Supplemental Section 4**), intrinsically function as a form of feature selector, as more important features carry heavier weights in the final ensemble of trees compared to less pertinent features. A simple approach to feature selection is to enumerate all possible feature subsets and select the one minimizing some model error score (e.g., RMSE of a particular cross-validation routine) (see Section 4.4). For testing up to *M* features out of *N* possible features, this approach requires *N* choose *M* model score evaluations, which is computationally prohibitive for large feature sets. Similar spirited approaches iteratively test one descriptor at a time and then add it to a growing list (forward feature selection) or remove it from a shrinking list (reverse feature selection) based on whether it results in the greatest reduction (or least increase) in the model error score. Forward (reverse) feature selection methods take $N! / (N - M)!$ model score evaluations to find *M* (remove *M*) features, which is generally tractable for models that are computationally fast to evaluate.

Supplemental Material >

Feature selection usually benefits from a consideration of the physical reasonableness of the features, and features that make no physical sense are obviously a concern (e.g., cost of elements correlating with bandgap). Such correlations are likely created by the feature correlating with some other more physical feature or features but the model not having enough data to select the correct features. Better models can generally be generated by intentionally replacing such features with physically motivated handpicked features that perform equivalently well (or better) as automatically selecting features [note that forward (reverse) feature selection is not a global optimization method and can miss optimal feature sets]. For example, Liu et al. (36) and Lu et al. (35) found that starting forward selection with an initial physically meaningful feature chosen by human intuition (and known physics) resulted in improved model performance compared to using purely automated forward selection. One can consider iterative exploration of two or more features for addition or subtraction from the feature list, although we are not aware of any examples in which this yielded significantly better results, and it greatly increases computational cost. In addition to these feature selection methods, other popular dimensional reduction methods take linear combinations of the features to best explain their behavior with fewer variables, often called latent variables (e.g., principal component analysis, linear discriminant analysis, and factor analysis). Including just the most important latent variables generated by these methods can improve some fits, although the interpretation of the latent variables can be difficult.

An important trend to be aware of in ML is the use of deep learning to obtain better results from features without extensive human guidance in both feature construction and feature selection (for more details, see Section 4.3; **Supplemental Section 4**). Deep learning methods can effectively generate their own feature set (generally called a feature map), often doing so starting from an initially large and rather unstructured set of features (e.g., a vector of pixel intensities or graph-based matrix of atom and bond properties) that are not effective with traditional ML methods. The comparison between more human-crafted versus machine-learned features has largely established the latter as superior in machine vision applications, leading to a revolution in the accuracy of ML in this field (144, 145). While the outcome of the comparison of human-crafted and machine-learned features in MS&E problems is not yet clear, there is increasing evidence that deep learning will provide significant improvements. For example, the deep convolutional individual residual network (40) was shown to achieve better performance from a long list of initial features than traditional methods such as RFDTs and ridge regression. Some graph-based deep learning methods, which build feature maps from a very basic initial feature list, have shown comparable or better performance in organic molecule studies than human-crafted traditional features in QSAR/QSPR comparisons (e.g., message passing NN frameworks) (146, 174). Similarly, for inorganic materials, the graph-based MatErials Graph Network (142), SchNet (147) and SchNetPack (148), and crystal-graph CNN (149, 150) have shown performance comparable to or better than non-deep-learning approaches. Given the success of deep learning in machine vision and language translation and its already impressive performance compared to more human-crafted features used in traditional methods after just a few years, it seems likely that deep learning–based feature maps will play a major if not dominant role in the future of feature development in ML for MS&E.

CV: cross validation

4.3. Types of Machine Learning Models

The large number of ML models and their many technical details are well covered in many texts and reviews (41–43, 151), and their discussion would require more space than is available here, so we do not attempt any type of general review of ML models. However, in **Supplemental Section 4**, we provide a short discussion of some of the most commonly used models (with a focus on tools for supervised regression) in MS&E with a goal of highlighting the most salient features for an MS&E researcher.

Supplemental Material >

4.4. Model Development and Model Assessment

ML modeling typically has two closely connected but distinct major stages: model development and model assessment. In model development (Section 4.4.1), we determine model type, parameters, hyperparameters, and features (see Section 2 for definitions of these terms). In model assessment, we determine the accuracy of the model for expected use cases, which typically includes assessing the model performance with sampling methods such as cross validation (CV) (Section 4.4.2), understanding the domain of applicability where the model is expected to be accurate, and quantifying error bars in model-predicted values to understand expected model uncertainties (Section 4.4.3). To help illustrate these important concepts of model applicability domain and assessment of model errors more concretely, we provide and discuss an in-depth practical example using ML models trained on data of calculated migration energies for solute elements in metallic hosts (Section 4.4.4).

4.4.1. Best practices for managing data in model development and assessment.

The same model scoring approaches are often used in both model development and model assessment, which can lead to overfitting and overestimation of the model accuracy if one is not careful. This danger is increased when model development involves many degrees of freedom (e.g., many hyperparameters) and there are limited data to constrain those degrees of freedom. The simple rule to avoid model assessment errors from overfitting is that any data used for model development should not be used for model assessment. To understand how to apply this rule practically, it is useful to define three types of data points (here, a data point means a vector of corresponding features X_i and target property value or values Y_i):

1. Training data are data used to determine the optimal model parameters for a given model type, hyperparameters, and feature set.
2. Validation data are data not used in training. They are instead used to assess the error in the model with optimal model parameters determined from fitting the training data. This error is frequently used to determine the optimal model type, hyperparameters, and feature set.

3. Testing data are completely left out and are not used in training or validation. Instead, they are used to assess the error in the final optimized model.

LO: leave out

First consider the process of model development. We start by dividing the data into training, validation, and test data in some way (we discuss how to do this most effectively in the practical example in Section 4.4.4). A basic fit of the model, with fixed model type, hyperparameters, and feature set, uses training data to find the optimal model parameters that give the lowest possible value of some scoring metric (typically measured with RMSE, so we use that here) on the training data. The RMSE obtained from these training data shows how well the model fits the training data, but this error is usually not a good estimate of how the model will fit data outside the training data. This limitation arises because the model often adjusts its many degrees of freedom to properties of the training data that cannot be correctly represented by the model (a process called overfitting), either due to limitations of the model form and features or noise in the data that cannot be modeled. To obtain a reasonable estimate of the model errors on new data not used in training, we can look at how well the model predicts the validation data—for example, the validation data RMSE. We can now optimize the model type, hyperparameters, and feature set to minimize the validation data RMSE. The optimal model type, hyperparameters, and feature set can then be used to refit the parameters of this model to the combined training and validation data to get the best possible fitted model without using the test data. The use of any information in model development from the test data, or more generally from a source that would not be available in a corresponding manner during model use, is sometimes called data leakage and can lead to overestimating the quality of your model.

4.4.2. Model development and assessment with cross validation. Perhaps the most common way to split data into sets for model development and assessment, typically called training and validation sets (defined in Section 4.4.1), is CV. Splits can be done in many ways, and common approaches include leaving out (LO) one data point or some randomly chosen X% fraction (typically called LO one CV or LO X% CV, respectively), splitting the whole data set into *k* separate equal-sized groups called folds, iteratively LO each fold once (*k*-fold CV), LO targeted groups with certain characteristics (LO group CV, sometimes called LO class CV) (e.g., all data with a specific chemical composition), and time-split CV (152), which leaves out select data based on the time of their inclusion in the data set. For LO X% CV and *k*-fold CV, one typically chooses which data are in each fold randomly, and this can be done multiple times with different random permutations to ensure good sampling. As discussed in Section 4.4.1, the errors in prediction for validation data from models trained on training data, which we call CV errors, are typically a much better way to assess a model than are errors in the training data predictions, as the latter typically show overfitting. CV errors are a common method of model assessment and can be used to develop a model (e.g., RMSE for all folds in fivefold CV is a common scoring metric used in feature selection, as discussed in Section 4.1) and estimate its predictive error, as discussed in more detail in Section 4.4.3.

Once an optimized model has been developed, we would like to assess its errors and domain. This model error and domain assessment ideally should not be done with CV scores already obtained using the validation data, since these CV scores can be subject to overfitting based on the optimization done in model development. Thus, we need to consider yet another left-out data set, the test data, to quantify the model error. Specifically, we take the optimal model type, hyperparameters, and feature set obtained from optimizing the validation data RMSE and refit the parameters of this model to the combined training and validation data to get the best possible fitted model, and we then predict the test data to get the test data RMSE. Because the test data have not been used in any step of the optimization process, the test data RMSE is a good quantification of errors in the final model.

The above approach is often not practical, as it is difficult to simply separate out test data and never look at them until the model is finalized. In addition, use of just one training, validation, and test data set may introduce large biases associated with the specific data that end up in those splits, leading to suboptimal models and error estimates, particularly for smaller data sets. These problems can be avoided by effectively simulating the above steps multiple times with different splits in a method called nested CV. First, you must settle on at least a general model development approach, which includes the types of models you will consider, hyperparameters you will optimize for each model, and features you will explore. Then, you perform CV on all the data, considering each excluded set as test data (level-1 CV), and an additional nested CV (level-2 CV) on the included training and validation data to determine the best model. Each level-1 left-out test set then can be considered a true test set in the sense that it was not used in any part of the model development. Many authors effectively perform a level-1 CV just once with approximately 10–20% of the data left out at level 1 and perhaps also level 2, as multiple folds at level 1 and level 2 can lead to a lot of computation. If the splits are done many times (e.g., with fivefold CVs for levels 1 and 2), they provide a strong sampling across all the data, which is recommended for smaller data sets where it may also be most practical. The nested CV approach is typically not totally rigorous, as researchers will almost inevitably modify aspects of their approach in light of the final results, thereby introducing some level of data leakage and potential overfitting, but nested CV is a practical approach to quantifying model errors that avoids most effects of overfitting.

One subtlety of the nested CV approach is that while one uses the level-2 CV to optimize model type, hyperparameters, and feature set, one is potentially overfitting to the level-2 CV score with multiple variables, which could lead to some of them actually being incorrectly optimized for truly best performance. A practical example of this, documented in Reference 153, is that if you optimize hyperparameters for two different model types and then choose between them based on the level-2 CV score, you might choose the less-optimal model type simply because it is more overfit. Ideally, one would use many nesting levels and optimize just one property at each level, but this can quickly become impractical, and one nesting level is all that is typically used. Such nesting should be adapted to best meet the specific optimization and assessment needs of your problem.

4.4.3. Model domain of applicability and assessing uncertainties in model predictions. Perhaps the most important question one can ask of an ML model is this: How accurate is the model for the potential applications I have in mind? Answering this important question typically has two coupled components, which are (*a*) an estimate of the domain where the model can be accurately used and (*b*) an estimate of the uncertainty in the model-predicted values (e.g., a standard deviation in prediction accuracy). Regarding (*a*), the model domain of applicability is a region of feature space outside of which we simply cannot reliably use the model (e.g., using a model trained only on yield strengths of metal alloys to predict yield strengths of polymers). Regarding (*b*), error estimates provide some form of uncertainty quantification on each value predicted by the model, thus providing more information on the uncertainty of a prediction than simply assessing the average predicted RMSE of the model. In this section, we provide a general introduction to understanding model errors. In Section 4.4.4, we illustrate how one may assess the errors and applicability domain of real ML models using GPR and RFDT models fit to computed databases of DFT-calculated dilute impurity diffusion activation energies in a range of metal hosts.

One can determine a domain of applicability based on some criterion of maximum acceptable errors if the errors are accurate everywhere in the feature space, but it may be necessary to understand the domain of the model to ascertain where error estimates can be trusted. Many methods can be used to assess domain of applicability based on some measure of distance of the features of a potential data point from those in the model training data (e.g., within the convex hull) (Reference 154 summarizes many of these methods). However, these methods all rely on distance metrics of uncertain validity for the specific problem being studied and require somewhat arbitrary cutoffs, so they are difficult to apply for more than a qualitative guide on where you might consider the model to be at risk of being not applicable. We believe some combination of distances in feature space from training data and predicted error values is likely to provide the best guidance on domain and error estimates. However, predicted error values are more immediately and obviously useful for assessing models, and therefore, we discuss here in more detail common methods used to establish some type of error bar on the predictions and their use in establishing model domain, each of which has certain strengths and limitations.

To better understand model prediction errors, it is useful to start with the well-known bias-variance-noise decomposition of the error. Following the definitions in Section 2, one can rigorously decompose the expected squared error for prediction on a new point X^* as

$$E\left[\left(F(X^*)+\epsilon-\hat{F}(X^*)\right)^2\right]=\left(E\left[\hat{F}(X^*)\right]-F(X^*)\right)^2+E\left[\left(\hat{F}(X^*)-E\left[\hat{F}(X^*)\right]\right)^2\right]+\sigma^2 \qquad 1.$$

Here, the expectation is the average over all possible training data sets of size n, which we can imagine to be randomly sampled from the total possible space of (X_i, Y_i) pairs. The three right-hand side terms from left to right are the bias squared, variance, and noise variance, respectively. The bias is the difference between the expected value of our model averaged over all training set samplings $E[\hat{F}(X)]$ and the underlying true function $F(X)$. The variance is the squared spread in $\hat{F}(X)$ relative to its average, again taken over all training set samplings. Intuitively, models with few parameters that underfit but are very well constrained will minimize variance but have large bias, and models with many parameters that overfit will minimize bias but have large variance. The lowest overall errors are typically found with a balance between optimizing both the bias and the variance. Equation 1 formally requires exploring every training data set of size n, and we typically have a problem with a single data set of size n, so it is not straightforward how to estimate the expected squared error in Equation 1.

4.4.4. Example of assessing model errors and domain of applicability using Gaussian process regression and random forest decision tree models on real data. In this section, we consider the errors and domains of some widely used modeling approaches on a realistic data set. There are two very common approaches to estimating a distribution on model prediction values. The first approach is ensemble methods, where one fits an ensemble of models, which can then yield a distribution of predictions for any new data point. The ensembles can be generated by resampling data (e.g., bootstrap and CV) or by refitting models (e.g., retraining NNs from different starting weights or with different dropouts), or a combination of both (as is done in RFDTs), as is described further below. The second approach is to use Bayesian methods to modify a prior distribution and produce a posterior distribution (e.g., as done in GPR). Ensemble methods are very flexible and can be applied to many models. For example, resampling can be used to get a predicted distribution for essentially any model if it is computationally feasible. Bayesian methods tend to require more specialized methods adapted to use a Bayesian approach but can potentially avoid many iterations and include key information through priors.

Here, we explore the behavior of error predictions from the very common approaches of CV, GPR, and RFDTs to better see how these errors behave and might be used. For simplicity, we usually consider just the mean and standard deviation (or RMSE, or just error) of predicted distributions, as these represent the prediction and a simple error bar, respectively, but the methods discussed here actually give a full distribution for predicted values. All these methods for estimating the error of a model result in model-predicted errors on every data point, although in the case of CV, the errors are often averaged over all predictions to obtain a single CV RMSE, as we do here. For each case below, we illustrate the accuracy of the estimated standard deviations by comparing them to actual observed standard deviations on validation and test data sets. We make use of models fitted to a database of DFT-calculated dilute impurity diffusion activation energies in a range of metal hosts. The data contain 408 activation energies for 15 different hosts and are described in detail in Reference 35 (see the sidebar titled Online Availability of Data in This Review for data availability on Figshare). All the models were evaluated using the routines available in the scikit-learn package (155), and the model fits and analysis were automated using the Materials Simulation Toolkit for Machine Learning (MAST-ML) (**https://github.com/uw-cmg/MAST-ML**) (156).

To help assess the model domain of applicability, we explore a chemistry test in which we consider Pd-X systems, where Pd is the host element and X is a dilute impurity taken from three sets (set 1 comprises 3d and 4d transition metals, set 2 comprises Col VIA elements except O, and set 3 comprises elements from the first two rows on the periodic table). In this test, we train the model with no Pd host data and then predict the errors for the three sets. While we have DFT data for only some of these predictions, they represent data that are very similar to those in our database (set 1, which has many 3d and 4d metals) and from quite to extremely different (sets 2 and 3, respectively), with set 2 sharing related chemistry due to being in the same column of the periodic table and set 3 having many dramatically distinct chemistries (e.g., Pd-O). Thus, we expect errors to be small in group 1, larger and similar in set 2, and larger and often outside the model domain in set 3.

Perhaps the most widely used approach for estimating model errors is the use of resampling methods, which estimate the uncertainty of predictions by sampling a subset of the available data (training data) and predicting remaining data left out of the subset (validation data). The errors on the left-out validation data are then used to estimate a typical error bar for the model. These approaches have the advantage of being relatively simple and applicable to any model being used. The most common resampling method for error prediction is probably CV (Section 4.4.2). Another common resampling method is bootstrapping, which differs from CV primarily by resampling with replacement. We do not discuss bootstrap in detail here owing to space limitations and the fact that CV appears to have some advantages over bootstrap for resampling (157). However, bootstrapping is used in the random forest method described below. In addition, basic *k*-fold CV has been shown to give relatively good estimates of errors (43) and is a recommended standard test for any model. Note that for *k*-fold CV, this error will generally increase with decreasing *k* (equivalently, increasing X% LO), particularly for smaller data sets, as the smaller and more independent training sets will lead to larger bias and variance. Values of *k* in the range of 3 to 10 are generally found to be a good compromise and yield good results. We illustrate this behavior in **Figure 3**, which shows a clearly increasing average CV error with *k* that matches the test data error best for *k* near 10.

In general, all resampling methods suffer from some significant limitations that are not always appreciated. The most severe and difficult to treat is that these methods give an estimate of the error for the data you have in your analysis (i.e., data in the training and validation sets), which can only be expected to be accurate for data in some way similar to your database. Therefore, resampling does not provide any guide on how similar new data are to those in the database. A related issue is that when you assess an error from an LO validation data point, you typically do not know how similar that point is to data in the subset used as training data. While duplicate

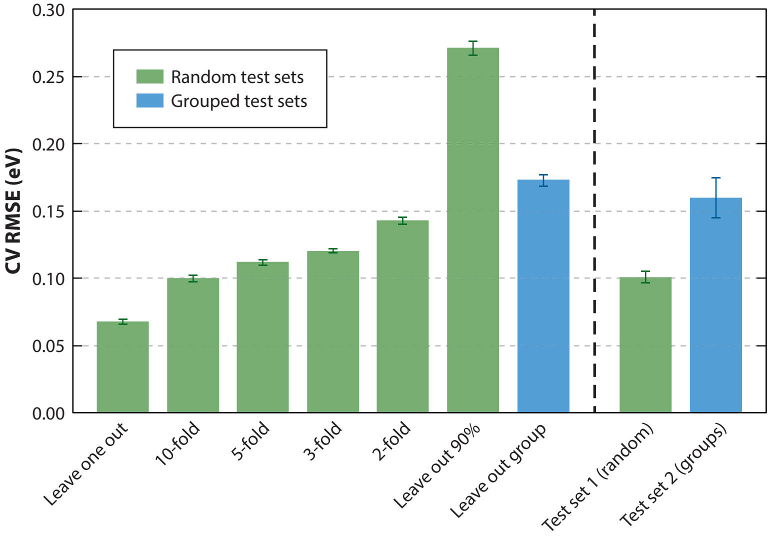


Figure 3

Plot of cross validation root mean squared error (CV RMSE) for various leave-out nested CV tests. The top level-1 split was into training and validation (*left of vertical dashed line*) and test data sets (*right of vertical dashed line*). These level-1 splits were done in two ways. Green bars signify tests done with five test sets chosen randomly with replacement, where each test set had 20% of the data and each training and validation set had 80% of the data. Blue bars signify tests with 15 test sets (one for each host), where each test set had one host and all impurities in that host and each training and validation set had all other data in the database. Predictions of test data were done with training on the full training and validation sets. The nested level-2 splits were done within the training and validation sets (*left of vertical dashed line*). The level-2 splits included leave one out, *k*-fold CV, and leave out 90% (randomly sampled five times, with replacement) (*green bars*) and leave out each host (14 splits) (*blue bar*). Error bars denote standard errors in the mean CV RMSE over level-2 splits (*left of vertical dashed line*) and level-1 splits (*right of vertical dashed line*). Error bars on the CV RMSE values indicate one standard error in the mean of the CV RMSE calculated with all values from level-2 or level-1 fits, as appropriate. All fits were done with Gaussian process regression using features optimized for this method taken from Lu et al. (35).

data can be easily removed, the validation data can be very similar to one or more elements of the training data, which will typically yield errors much lower than for a prediction on a data point less similar to the validation data (this is sometimes called the twin problem, as your validation data point has one or more nearly identical twins in the training data). Both of these issues are closely related and arise from the fact that resampling yields error estimates potentially closely tied to the specific characteristics of the data sampled and predicted and may not represent the errors one will obtain for the future predictions to be made by the model.

An excellent example of this problem can be found in a recent study of superconducting temperatures (20), in which models fit to just low- or just high-temperature superconductors both showed good CV scores within each group but essentially no ability to predict the other group. This result is easy to understand in terms of the known large qualitative differences in the physics governing low- and high-temperature superconductivity, but one cannot rely on such robust physical guidance in general. These issues can be somewhat alleviated by careful LO group error bar assessments, in which one attempts to mimic the types of prediction challenges the model will face in real applications (14, 20, 35, 158). For materials systems, good LO group tests might typically include leaving out certain elements, alloys, or composition ranges. For example, the LO host error on test data shown in **Figure 3** is significantly larger than that obtained from the *k*-fold CV for typical *k* values of 3 or 5, demonstrating that the latter is unreliable for predicting new hosts, but it is well estimated by the LO host error determined from the training and validation data. A more direct way to avoid the twin problem might be to remove all compositions within some hypersphere around any point in the validation data, thereby ensuring the predictions are always being made from significantly different compositions. A particularly elegant way to select LO groups that mimic how your model will be used is to explicitly test new data based on data from earlier times [time-split CV (152)], although this is not always practical or appropriate. Sheridan (152) used QSAR data to show that time-split CV was quite accurate, while random LO CV tests tended to result in an overly optimistic assessment of a model and LO clusters CV (i.e., a variant of LO group CV) tended to result in an overly pessimistic assessment of a model. In general, we recommend that all model development and error quantification done with resampling [e.g., nested CV (Section 4.4.2)] at least use a CV error determined by combining LO random folds and LO physically motivated groups that assess your planned uses for the model and remove twin effects.

As mentioned above, Bayesian methods can provide an error bar without resampling. Perhaps the most widely used Bayesian method in MS&E is GPR, discussed in **Supplemental Section 4**. GPR distributions for a new point are entirely determined by the feature matrix of the training data and the model kernel and do not depend on the specific values of the training data (except for a fit scale factor, σ_{max}, that typically closely matches the training data standard deviation), making GPR distributions effectively a measure of how similar a new data point to be predicted is to the database being used to train the GPR model. Data points very similar to those in the training database will have small errors, while those less similar will have larger errors. These error bars do have the limitation that they are estimates from a modified prior and are therefore expected to become less accurate for data points far from the training data. In fact, GPR error bars approach the constant value σ_{max} for points far from (i.e., having quite different feature values than) the original data set. Thus, for a good model and predicted errors significantly less than σ_{max}, the error estimates can potentially be taken as reliable, but for predicted errors near to σ_{max}, the errors cannot be taken as quantitative, although they do suggest that the model is not robust for those data. In this way, for any prediction, GPR potentially provides either potentially reasonable error bar estimates or a clear warning that a particular data point is outside the domain of the model. GPR error estimates can also be used to assess where the GPR model is least constrained, suggesting where a new data point might be added to best improve the model, making it a powerful guide for iterative optimization with active learning (see Section 3.1.2).

Supplemental Material >

Figure 4*a* shows the standard deviations predicted for the three chemistry groups discussed above in this section, and the results are astonishingly close to what we would expect from chemical intuition. These results suggest that, at least in this case, the GPR errors are both accurate on average in the domain of the model and capable of establishing set 1 (or set 3) as inside (or outside) the model domain, with set 2 at the border of the model domain. However, **Figure 4*c*** shows, at least in this case, that the root mean of the squared residuals and GPR-predicted errors show very limited correlation, suggesting that while in the model domain the GPR errors are of the correct average size, they do not appear to be varying by data point in a physically meaningful way. The results of **Figure 4*c*** suggest that GPR can predict large errors for systems well predicted by the model, so GPR may give a fairly conservative estimate for the model domain.

One of the most widely used ensemble ML approaches (in addition to CV) in MS&E is RFDTs, which are formulated to provide an intrinsically powerful tool for estimating uncertainties. RFDTs train an ensemble of models and thereby predict a distribution of values for new data points, generally providing both good estimates from mean values and uncertainties from the spread of the distribution (see **Supplemental Section 4** for more information). The ensemble of

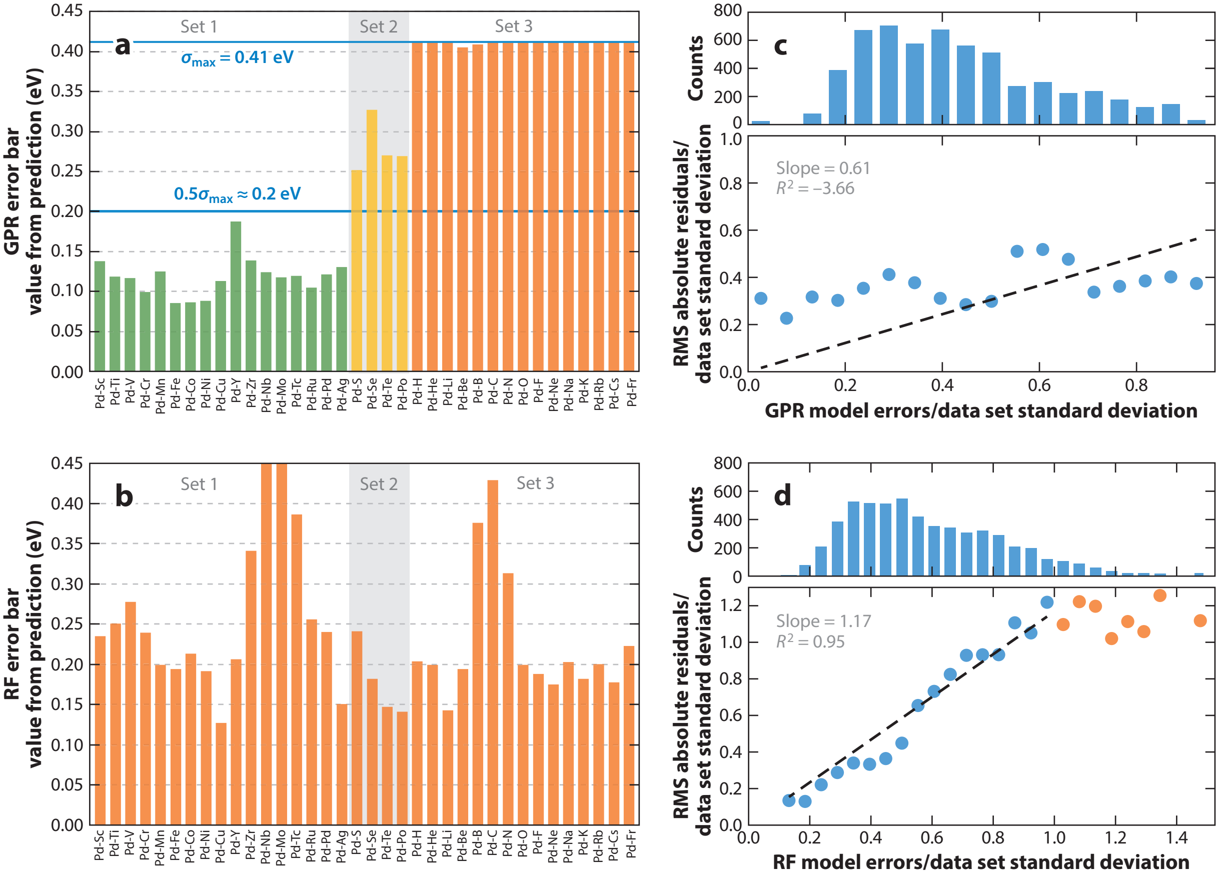


Figure 4

Summary of Gaussian process regression (GPR) and random forest decision tree (RFDT) chemistry tests. (*a*,*b*) Chemistry tests showing model errors on predicted values for various solutes in a Pd host using the GPR model (*a*) and RFDT model (*b*). (*c*,*d*) Comparison of root mean square (RMS) absolute value of the residuals versus the binned model error values for GPR (*c*) and RFDT (*d*) tests. In panels *a* and *b*, the models were trained on all data except Pd. In panels *c* and *d*, both the *x* and *y* axes values are normalized by the data set standard deviation, which is 0.4738 eV. The linear fits have intercepts that are forced to equal 0. In panel *d*, the linear fit is done only on the blue data points, which have normalized binned RFDT errors <1. The histograms in panels *c* and *d* show the counts of the number of mean squared residuals used to obtain the RMS residual for a given model error bin. The fits in panels *c* and *d* were performed using the same 15 grouped data sets as described in the caption of **Figure 3**. These data sets are equivalent to leave-out-two-hosts cross validation, where each training data set excludes two hosts and the predictions are done on the two excluded hosts. This resampling corresponds to 15 × 14 = 210 training and validation splits, and each data point is predicted 14 times for a total of 408 × 14 = 5,712 total predictions.

models is traditionally generated by fitting to different data samplings (e.g., bootstrap aggregation or bagging being perhaps the best known approach) or iteratively reweighting the fitted data to harder cases (boosting), but it can also be generated from varying the model used in fitting (e.g., changing dropouts in NNs or possible split criteria in decision trees). A detailed discussion of these approaches across all methods is outside the scope of this review, but it is useful to be aware of a few important examples.

A particularly rigorous formulation of RFDT error estimates (which includes correction for the sampling and limited ensemble size as well as for missing bias and noise contributions) and an assessment showing their accuracy on materials properties are given in Reference 159, although here we simply use the standard deviation of the distribution of predicted values to obtain errors. Similar to GPR, these estimates are expected to become less accurate for data far from the original training data. For RFDTs that use the mean of the individual decision tree estimators for regression, this value is bounded at half the range of the training data (since maximally varying predictions will match the lowest and highest values half the time each), although it is unlikely to reach that value, and we here assume that any value approaching the standard deviation of the training data, σ_{train}, is likely to signify the data are outside the domain of the model. Unlike the GPR case, the RFDT error predictions are likely to be sensitive to both the X and Y values in the training data. **Figure 4*b*** shows the analogous chemistry plot for RFDTs as was shown for GPR in **Figure 4*a***. However, unlike GPR, the root mean of the squared residuals and RFDT-predicted standard deviations show strong correlation, as shown in **Figure 4*d***, for predicted standard deviations up to about the standard deviation of the total data set and then show a clear transition to almost no correlation. Also, unlike GPR, the Pd-X predictions show no ability to distinguish chemistries. These studies suggest that, for the data studied here, GPR errors are good for determining a conservative estimate for the model domain and good on average in that domain, but they are not reliable for distinguishing trends between data points in the domain, while RFDT errors are good on average and for distinguishing trends between data points in the domain but not so good at estimating true errors when they approach the standard deviation of the data set and not very good at determining the model domain itself. We reiterate that these studies were done on just one fairly small data set and absolutely cannot be used to make robust broad conclusions, but the results suggest some of the opportunities and challenges of using error estimates and show the need for further studies to establish how they can be best applied to problems in MS&E.

Finally, we note that NNs can also provide their own uncertainty estimates through an ensemble of networks approach. This can include simply starting from random weight initializations multiple times (which can be time consuming) (160), using snapshots taken during a typical optimization run (160), and exploring multiple fits done with different dropouts (dropouts in NNs are removing output of a random and changing subset of nodes) (161).

Despite one's best efforts using methods such as those described above, it can be difficult to be sure one has a meaningful model for the data for small data sets. A few checks against simple naive references are recommended to ensure that the model is adding significant value. These are described in **Supplemental Section 5**.

Supplemental Material >

5. MACHINE LEARNING TOOLS AND SOFTWARE FOR MATERIALS

Recently, there has been intense development of open-source software packages aimed at streamlining and accelerating the adoption of ML in general and in MS&E in particular. Effective software tools are becoming increasingly important to maintain community best practices and ease of use, especially given the rapidly evolving field of ML and its application to MS&E more specifically, and especially for users new to the field (67, 73, 162). We have provided a detailed list of these packages with a brief explanation of the types of ML-related analysis enabled by each package in **Supplemental Section 2** and posted via Figshare (see second link in the sidebar titled Online Availability of Data in This Review) to enable updates to this evolving list in the future.

6. FUTURE OPPORTUNITIES AND ONGOING CHALLENGES OF MACHINE LEARNING IN MATERIALS SCIENCE AND ENGINEERING

MS&E is still just beginning to utilize informatics on large databases (162, 163), but the increasing data generation rates from both experiments and simulation increasingly create opportunities for—and sometimes necessitate—using ML for analysis. This trend, along with the rapid evolution of ML algorithms, supporting hardware, cloud data, and computing resources, suggests that opportunities for ML in MS&E are still far from being fully realized. Here, we highlight what we see as three (of no doubt many) key opportunities and associated challenges for ML in MS&E to address in the coming years.

The first opportunity revolves around the creation of a codified, living materials informatics ecosystem that unifies materials data, MS&E-centric ML tools, and the generation, analysis, and dissemination of ML models in a democratized fashion. The development and dissemination of models in a robust innovation infrastructure is still missing and would dramatically increase the use and impact of ML on MS&E. As a testament to the importance of seizing this opportunity, the potential impact of and need for additional developments in ML across many fields of MS&E has been recognized in reports and at workshops hosted by many organizations—for example, the US Department of Energy (164), National Institute of Standards and Technology, American Society of Mechanical Engineers (165), and the National Science Foundation (165)—and has been reviewed in various articles (39, 67, 73, 162, 163, 166–168). As ML tools become ubiquitous in MS&E, we envision this new infrastructure could enable materials researchers, particularly the many who are not ML specialists, to construct multistep, automated workflows for complex analysis and to experiment with various algorithms and approaches to solve a particular problem, all within a consistent interface and nomenclature that implements best practices for materials-specific data, and without repeated human intervention for data formatting and translation. Such infrastructure is also necessary to allow ML models to be disseminated effectively in the broad materials innovation ecosystem, which includes ensuring they are discoverable, reproducible, reusable, and machine and human accessible, including access via an application programming interface for incorporation into more complex workflows.

In addition to this new infrastructure centered on ML models, there is also a need for open data that are curated and hosted, which will prevent data siloing and improve ease of access and sharing (67). Consistent materials metadata—for example, as implemented by the Citrination platform (169)—will also enable more informed comparisons between similar data sets, such as when comparing materials property data obtained from DFT calculations of different levels of fidelity. A long-standing challenge is related to the tradition in the scientific community of rarely reporting failed or null results in the literature. However, such results still constitute valuable information, particularly for training ML models, which can be leveraged to facilitate new materials advances. For example, recently the exploration of vanadium selenide materials synthesis was informed by failed synthesis approaches (170). Information that is often not deemed publishable in traditional peer-reviewed scientific studies can improve ML approaches by reducing the biases toward particular outcomes of data typically reported in the literature (e.g., data on solid-state Li electrolytes may be biased toward systems that are fast Li conductors) and thus should still be made publicly available, perhaps by way of the codified infrastructure described above or through new incentives encouraged by journal publishers.

In the coming years, the advancements made in the ML and broader field of AI will likely change how humans conduct scientific research. Indeed, the advent of autonomous robot scientists has already begun to shift the role of human scientists in the lab from actively conducting individual experiments to instead analyzing vast amounts of automatically produced data. These advancements create a large opportunity for more efficient and less error-prone scientific investigation but also create challenges related to how human researchers use and interact with ML/AI tools in a manner that results in improved outcomes compared to purely human- or machine-driven analysis. Human–ML collaboration (also referred to as centaur approaches, interactive ML, or human-in-the-loop ML) will likely evolve substantially in the near future and play a key role in many domains. For example, while computers equipped with ML/AI tools are better on average than humans at many tasks (e.g., image recognition), edge cases can still occur that result in incorrect model predictions; these cases could be quickly checked and fixed based on human intuition. Thus, human-in-the-loop ML approaches will remain useful for error minimization and sanity checks, particularly for situations in which data are sparse or the edge of the domain of applicability is being reached, and will be of particular importance in situations such as the health care field, in which decisions reached using ML tools can result in life or death (171). As a concrete example of the power of human-in-the-loop methods in MS&E, the work of Duros et al. (172) showed that active learning approaches incorporating a machine–human hybrid team outperformed both the pure human– and pure ML–based prediction of performing the chemical reaction of the self-assembly and crystallization of polyoxometalate clusters. As another example, the work of Gómez-Bombarelli et al. (173) found thousands of promising, organic, light-emitting diode molecules, in part by leveraging domain expert opinion on which molecules were most worth investigating experimentally using an online voting process. While it is currently the case that human–machine hybrid teams tend to result in better outcomes than what either humans or machines could produce in isolation, we speculate that it is very likely in the future (it is unclear when, but perhaps in the coming few decades) that ML/AI approaches will always outperform humans at numerous computationally intensive tasks integral to the scientific enterprise. It is also possible that how we perceive human-in-the-loop ML may change dramatically in the near future. Instead of the human and ML algorithm being used collectively but existing separately, it is possible that linkage of human and machine via brain-to-machine interfacing—for example, as is being developed by companies such as Neuralink—will fundamentally alter how human researchers interact with and use ML/AI approaches to advance the scientific enterprise.

To conclude, we see many ways in which ML (and AI) is already changing MS&E, but we believe their interactions are still in the nascent stages, with the full power of their merging still far from being fully realized. The impact of their coupling is also expected to evolve quickly through building on the rapid evolution of the broader ML ecosystem, providing the opportunity for transformative advances to the discovery, design, and deployment of new materials impacting myriad technologies central to today's society.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

The authors gratefully acknowledge support of this research by the US National Science Foundation through the Software Infrastructure for Sustained Innovation (SI^2) award number 1148011, which supported aspects of the machine learning tools and data used in this work; Designing Materials to Revolutionize and Engineer our Future (DMREF) award number 1728933, which supported aspects of the understanding of model development and fitting and relevant collaborations; Cyberinfrastructure for Sustained Scientific Innovation (CSSI) award number 1931298, which supported exploration of band gap models and machine learning infrastructure; and the University of Wisconsin–Madison Materials Research Science and Engineering Center (DMR-1720415), which supported projects related to QSAR/QSPR that informed this work.

LITERATURE CITED

1. Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv:1712.01815 [cs.AI]

2. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529(7587):484–89
3. Morav M, Schmid M, Burch N, Lisý V, Morrill D, et al. 2017. DeepStack: expert-level artificial intelligence in heads-up no-limit poker. *Science* 356(6337):508–13
4. Brown N, Sandholm T. 2018. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science* 359(6374):418–24
5. Ferrucci D, Brown E, Chu-carroll J, Fan J, Gondek D, et al. 2010. Building Watson: an overview of the DeepQA project. *AI Mag.* 31(3):59–79
6. Jordan MI, Mitchell TM. 2015. Machine learning: trends, perspectives, and prospects. *Science* 349(6245):255–60
7. Jouppi NP, Young C, Patil N, Patterson D, Agrawal G, et al. 2017. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, pp. 1–12. New York: Assoc. Comput. Mach.
8. Olson GB. 2000. Designing a new material world. *Science* 288(5468):993–98
9. Panchal JH, Kalidindi SR, McDowell DL. 2013. Key computational modeling issues in integrated computational materials engineering. *Comput.-Aided Des.* 45(1):4–25
10. McDowell DL, Kalidindi SR. 2016. The materials innovation ecosystem: a key enabler for the Materials Genome Initiative. *MRS Bull.* 41(4):326–35
11. Kailil T, Wadia C. 2011. *Materials Genome Initiative for global competitiveness*. White Pap., Nat. Sci. Technol. Counc., Washington, DC. **https://www.mgi.gov/sites/default/files/documents/materials_genome_initiative-final.pdf**
12. de Pablo JJ, Jackson NE, Webb MA, Chen LQ, Moore JE, et al. 2019. New frontiers for the Materials Genome Initiative. *npj Comput. Mater.* 5:41
13. Ye W, Chen C, Wang Z, Chu I-H, Ong SP. 2018. Deep neural networks for accurate predictions of crystal stability. *Nat. Commun.* 9:3800
14. Li W, Jacobs R, Morgan D. 2018. Predicting the thermodynamic stability of perovskite oxides using machine learning models. *Comput. Mater. Sci.* 150:454–63
15. Balachandran PV, Emery AA, Gubernatis JE, Lookman T, Wolverton C, Zunger A. 2018. Predictions of new $AB\mathrm{O}_3$ perovskite compounds by combining machine learning and density functional theory. *Phys. Rev. Mater.* 2(4):043802
16. Faber FA, Lindmaa A, von Lilienfeld OA, Armiento R. 2016. Machine learning energies of 2 million elpasolite (ABC_2D_6) crystals. *Phys. Rev. Lett.* 117(13):135502
17. Hautier G, Fischer CC, Jain A, Mueller T, Ceder G. 2010. Finding nature's missing ternary oxide compounds using machine learning and density functional theory. *Chem. Mater.* 22(12):3762–67
18. Meredig B, Agrawal A, Kirklin S, Saal JE, Doak JW, et al. 2014. Combinatorial screening for new materials in unconstrained composition space with machine learning. *Phys. Rev. B* 89(9):94104
19. Stanev V, Oses C, Kusne AG, Rodriguez E, Takeuchi I, et al. 2018. Machine learning modeling of superconducting critical temperature. *npj Comput. Mater.* 4:29
20. Meredig B, Antono E, Church C, Hutchinson M, Ling J, et al. 2018. Can machine learning identify the next high-temperature superconductor? Examining extrapolation performance for materials discovery. *Mol. Syst. Des. Eng.* 3(5):819–25
21. Seko A, Maekawa T, Tsuda K, Tanaka I. 2014. Machine learning with systematic density-functional theory calculations: application to melting temperatures of single- and binary-component solids. *Phys. Rev. B* 89(5):54303
22. Mannodi-Kanakkithodi A, Pilania G, Huan TD, Lookman T, Ramprasad R. 2016. Machine learning strategy for accelerated design of polymer dielectrics. *Sci. Rep.* 6:20952
23. Kim C, Pilania G, Ramprasad R. 2016. Machine learning assisted predictions of intrinsic dielectric breakdown strength of ABX_3 perovskites. *J. Phys. Chem. C* 120:14575–80
24. Kim K, Ward L, He J, Krishna A, Agrawal A, Wolverton C. 2018. Machine-learning-accelerated high-throughput materials screening: discovery of novel quaternary Heusler compounds. *Phys. Rev. Mater.* 2(12):123801

25. Legrain F, Carrete J, Van Roekeghem A, Madsen GKH, Mingo N. 2018. Materials screening for the discovery of new half-Heuslers: machine learning versus ab initio methods. *J. Phys. Chem. B* 122(2):625–32
26. Ward L, O'Keeffe SC, Stevick J, Jelbert GR, Aykol M, Wolverton C. 2018. A machine learning approach for engineering bulk metallic glass alloys. *Acta Mater*. 159:102–11
27. Pilania G, Gubernatis JE, Lookman T. 2017. Multi-fidelity machine learning models for accurate bandgap predictions of solids. *Comput. Mater. Sci.* 129:156–63
28. Ramprasad R, Mannodi-Kanakkithodi A, Lookman T, Pilania G, Uberuaga BP, Gubernatis JE. 2016. Machine learning bandgaps of double perovskites. *Sci. Rep.* 6:19375
29. Lee J, Seko A, Shitara K, Nakayama K, Tanaka I. 2016. Prediction model of band gap for inorganic compounds by combination of density functional theory calculations and machine learning techniques. *Phys. Rev. B* 93(11):115104
30. Zhuo Y, Tehrani AM, Brgoch J. 2018. Predicting the band gaps of inorganic solids by machine learning. *J. Phys. Chem. Lett.* 9(7):1668–73
31. Lu S, Zhou Q, Ouyang Y, Guo Y, Li Q, Wang J. 2018. Accelerated discovery of stable lead-free hybrid organic-inorganic perovskites via machine learning. *Nat. Commun.* 9:3405
32. Li Z, Xu Q, Sun Q, Hou Z, Yin W-J. 2019. Thermodynamic stability landscape of halide double perovskites via high-throughput computing and machine learning. *Adv. Funct. Mater.* 29(9):1807280
33. Im J, Lee S, Ko T-W, Kim HW, Hyon Y, Chang H. 2019. Identifying Pb-free perovskites for solar cells by machine learning. *npj Comput. Mater.* 5:37
34. Wu H, Lorenson A, Anderson B, Witteman L, Wu H, et al. 2017. Robust FCC solute diffusion predictions from ab-initio machine learning methods. *Comput. Mater. Sci.* 134:160–65
35. Lu H-J, Zou N, Jacobs R, Afflerbach B, Lu X-G, Morgan D. 2019. Error assessment and optimal cross-validation approaches in machine learning applied to impurity diffusion. *Comput. Mater. Sci.* 169:109075
36. Liu Y, Jacobs R, Lin S, Morgan D. 2019. Exploring effective charge in electromigration using machine learning. *MRS Commun*. 9:567–75
37. Pilania G, McClellan KJ, Stanek CR, Uberuaga BP. 2018. Physics-informed machine learning for inorganic scintillator discovery. *J. Chem. Phys.* 148(24):241729
38. Yuan R, Liu Z, Balachandran PV, Xue D, Zhou Y, et al. 2018. Accelerated discovery of large electrostrains in $BaTiO_3$-based piezoelectrics using active learning. *Adv. Mater.* 30(7):1702884
39. Ramprasad R, Batra R, Pilania G, Mannodi-Kanakkithodi A, Kim C. 2017. Machine learning and materials informatics: recent applications and prospects. *npj Comput. Mater.* 3:54
40. Jha D, Ward L, Yang Z, Wolverton C, Foster I, et al. 2019. IRNet: a general purpose deep residual regression framework for materials discovery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2385–93. New York: Assoc. Comput. Mach.
41. Mueller T, Kusne AG, Ramprasad R. 2016. Machine learning in materials science: recent progress and emerging applications. In *Reviews in Computational Chemistry*, Vol. 29, ed. AL Parrill, KB Lipkowitz, pp. 186–273. Hoboken, NJ: Wiley
42. Raschka S, Mirjalili V. 2017. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. Birmingham, UK: Packt. 2nd ed.
43. Alpaydin E. 2014. *Introduction to Machine Learning*. Boston: MIT Press
44. Goodfellow I, Bengio Y, Courville A. 2016. *Deep Learning*. Cambridge, MA: MIT Press
45. Nantasenamat C, Isarankura-Na-Ayudhya C, Naenna T, Prachayasittikul V. 2009. A practical overview of quantitative structure-activity relationship. *EXCLI J.* 8:74–88
46. Karelson M, Lobanov VS, Katritzky AR. 1996. Quantum-chemical descriptors in QSAR/QSPR studies. *Chem. Rev.* 96(3):1027–44
47. Debbichi L, Lee S, Cho H, Rappe AM, Hong KH, et al. 2018. Mixed valence perovskite $Cs_2Au_2I_6$: a potential material for thin-film Pb-free photovoltaic cells with ultrahigh efficiency. *Adv. Mater.* 30(12):1707001
48. Rouet-Leduc B, Barros K, Lookman T, Humphreys CJ. 2016. Optimisation of GaN LEDs and the reduction of efficiency droop using active machine learning. *Sci. Rep.* 6:24862

49. Rouet-Leduc B, Hulbert C, Barros K, Lookman T, Humphreys CJ. 2017. Automatized convergence of optoelectronic simulations using active machine learning. *Appl. Phys. Lett.* 111(4):43506

50. Bassman L, Rajak P, Kalia RK, Nakano A, Sha F, et al. 2018. Active learning for accelerated design of layered materials. *npj Comput. Mater.* 4:74

51. Smith JS, Nebgen B, Lubbers N, Isayev O, Roitberg AE. 2018. Less is more: sampling chemical space with active learning. *J. Chem. Phys.* 148(24):241733

52. Lookman T, Balachandran PV, Xue D, Yuan R. 2019. Active learning in materials science with emphasis on adaptive sampling using uncertainties for targeted design. *npj Comput. Mater.* 5:21

53. Lookman T, Balachandran PV, Xue D, Hogden J, Theiler J. 2017. Statistical inference and adaptive design for materials discovery. *Curr. Opin. Solid State Mater. Sci.* 21(3):121–28

54. Kim C, Chandrasekaran A, Jha A, Ramprasad R. 2019. Active-learning and materials design: the example of high glass transition temperature polymers. *MRS Commun*. 9(3):860–66

55. Granda JM, Donina L, Dragone V, Long DL, Cronin L. 2018. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature* 559(7714):377–81

56. Soldatova LN, Clare A, Sparkes A, King RD. 2006. An ontology for a robot scientist. *Bioinformatics* 22(14):464–71

57. Talapatra A, Boluki S, Duong T, Qian X, Dougherty E, Arróyave R. 2018. Autonomous efficient experiment design for materials discovery with Bayesian model averaging. *Phys. Rev. Mater.* 2(11):113803

58. Tabor DP, Roch LM, Saikin SK, Kreisbeck C, Sheberla D, et al. 2018. Accelerating the discovery of materials for clean energy in the era of smart automation. *Nat. Rev. Mater.* 3(5):5–20

59. Nikolaev P, Hooper D, Webber F, Rao R, Decker K, et al. 2016. Autonomy in materials research: a case study in carbon nanotube growth. *npj Comput. Mater.* 2:16031

60. Häse F, Roch LM, Aspuru-Guzik A. 2019. Next-generation experimentation with self-driving laboratories. *Trends Chem*. 1(3):282–91

61. MacLeod BP, Parlane FGL, Morrissey TD, Häse F, Roch LM, et al. 2019. Self-driving laboratory for accelerated discovery of thin-film materials. arXiv:1906.05398 [physics.app-ph]

62. Duros V, Grizou J, Xuan W, Hosni Z, Long DL, et al. 2017. Human versus robots in the discovery and crystallization of gigantic polyoxometalates. *Angew. Chemie Int. Ed.* 56(36):10815–20

63. King RD, Oliver SG, Rowland J, Soldatova LN, Whelan KE, et al. 2009. The automation of science. *Science* 324(5923):85–89

64. Sparkes A, Aubrey W, Byrne E, Clare A, Khan MN, et al. 2010. Towards robot scientists for autonomous scientific discovery. *Autom. Exp.* 2:1

65. Zunger A. 2018. Inverse design in search of materials with target functionalities. *Nat. Rev. Chem.* 2:0121

66. Arróyave R, McDowell DL. 2019. Systems approaches to materials design: past, present, and future. *Annu. Rev. Mater. Res.* 49:103–26

67. Alberi K, Nardelli MB, Zakutayev A, Mitas L, Curtarolo S, et al. 2019. The 2019 materials by design roadmap. *J. Phys. D Appl. Phys.* 52(1):13001

68. Sanchez-Lengeling B, Aspuru-Guzik A. 2018. Inverse molecular design using machine learning: generative models for matter engineering. *Science* 361(6400):360–65

69. Nouira A, Crivello J-C, Sokolovska N. 2019. CrystalGAN: learning to discover crystallographic structures with generative adversarial networks. arXiv:1810.11203 [cs.LG]

70. Sanchez-Lengeling B, Outeiral C, Guimaraes GL, Aspuru-Guzik A. 2017. Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC). ChemRxiv. **https://doi.org/10.26434/chemrxiv.5309668.v3**

71. Putin E, Asadulaev A, Ivanenkov Y, Aladinskiy V, Sanchez-Lengeling B, et al. 2018. Reinforced adversarial neural computer for de novo molecular design. *J. Chem. Inf. Model.* 58(6):1194–204

72. Voyles PM. 2017. Informatics and data science in materials microscopy. *Curr. Opin. Solid State Mater. Sci.* 21(3):141–58

73. Dimiduk DM, Holm EA, Niezgoda SR. 2018. Perspectives on the impact of machine learning, deep learning, and artificial intelligence on materials, processes, and structures engineering. *Integr. Mater. Manuf. Innov.* 7:157–72

74. Li W, Field KG, Morgan D. 2018. Automated defect analysis in electron microscopic images. *npj Comput. Mater.* 4:36
75. Ziatdinov M, Dyck O, Maksov A, Li X, Sang X, et al. 2017. Deep learning of atomically resolved scanning transmission electron microscopy images: chemical identification and tracking local transformations. *ACS Nano* 11(12):12742–52
76. Park WB, Chung J, Jung J, Sohn K, Singh SP, et al. 2017. Classification of crystal structure using a convolutional neural network. *IUCrJ* 4:486–94
77. Stein HS, Guevarra D, Newhouse PF, Soedarmadji E, Gregoire JM. 2019. Machine learning of optical properties of materials—predicting spectra from images and images from spectra. *Chem. Sci.* 10(1):47–55
78. Combs A, Maldonis JJ, Feng J, Xu Z, Voyles PM, Morgan D. 2019. Fast approximate STEM image simulations from a machine learning model. *Adv. Struct. Chem. Imaging* 5:2
79. Mikolov T, Chen K, Corrado G, Dean J. 2013. Efficient estimation of word representations in vector space. arXiv:1301.3781 [cs.CL]
80. Pennington J, Socher R, Manning CD. 2014. GloVe: global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, ed. A Moschitti, B Pang, W Daelemans, pp. 1532–43. Doha, Qatar: Assoc. Comput. Ling.
81. dos Santos CN, Gatti M, dos Santos CN, Gatti M. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, ed. J Tsujii, J Hajic, pp. 69–78. Dublin: Assoc. Comput. Ling.
82. Westergaard D, Stærfeldt HH, Tønsberg C, Jensen LJ, Brunak S. 2018. A comprehensive and quantitative comparison of text-mining in 15 million full-text articles versus their corresponding abstracts. *PLOS Comput. Biol.* 14(2):e1005962
83. Rebholz-Schuhmann D, Oellrich A, Hoehndorf R. 2012. Text-mining solutions for biomedical research: enabling integrative biology. *Nat. Rev. Genet.* 13(12):829–39
84. Yandell MD, Majoros WH. 2002. Genomics and natural language processing. *Nat. Rev. Genet.* 3(8):601–10
85. Meystre SM, Savova GK, Kipper-Schuler KC, Hurdle JF. 2008. Extracting information from textual documents in the electronic health record: a review of recent research. *Yearb. Med. Inform.* 17(1):128–44
86. Evans JA, Aceves P. 2016. Machine translation: mining text for social theory. *Annu. Rev. Sociol.* 42:21–50
87. Tshitoyan V, Dagdelen J, Weston L, Dunn A, Rong Z, et al. 2019. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* 571(7763):95–98
88. Tshitoyan V, Dagdelen J, Weston L, Dunn A, Rong Z, et al. 2019. Supplementary materials for "Unsupervised word embeddings capture latent knowledge from materials science literature." *Nature* 571:95–98. **https://github.com/materialsintelligence/mat2vec**
89. Kim E, Huang K, Tomala A, Matthews S, Strubell E, et al. 2017. Machine-learned and codified synthesis parameters of oxide materials. *Sci. Data* 4:170127
90. Mysore S, Jensen Z, Kim E, Huang K, Chang H-S, et al. 2019. The materials science procedural text corpus: annotating materials synthesis procedures with shallow semantic structures. arXiv:1905.06939 [cs.CL]
91. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, et al. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, ed. I Guyon, UV Luxburg, S Bengio, H Wallach, R Fergus, et al., pp. 5999–6009. Long Beach, CA: Neural Inf. Process. Syst. Found.
92. Devlin J, Chang M-W, Lee K, Toutanova K. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, ed. J Burstein, C Doran, T Solorio, pp. 4171–86. Minneapolis, MN: Assoc. Comput. Ling.
93. Strubell E, Verga P, Belanger D, McCallum A. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, ed. M Palmer, R Hwa, S Riedel, pp. 2670–80. Copenhagen, Den.: Assoc. Comput. Ling.
94. Honnibal M, Johnson M. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, ed. L Màrquez, C Callison-Burch, J Su, pp. 1373–78. Lisbon, Port.: Assoc. Comput. Ling.

95. Swain MC, Cole JM. 2016. ChemDataExtractor: a toolkit for automated extraction of chemical information from the scientific literature. *J. Chem. Inf. Model.* 56(10):1894–904
96. Weston L, Tshitoyan V, Dagdelen J, Kononova O, Persson KA, et al. 2019. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. ChemRxiv. **https://doi.org/10.26434/chemrxiv.8226068.v1**
97. Kim E, Huang K, Saunders A, McCallum A, Ceder G, Olivetti E. 2017. Materials synthesis insights from scientific literature via text extraction and machine learning. *Chem. Mater.* 29(21):9436–44
98. Kim E, Jensen Z, van Grootel A, Huang K, Staib M, et al. 2018. Inorganic materials synthesis planning with literature-trained neural networks. arXiv:1901.00032 [cond-mat.mtrl-sci]
99. Kim E, Huang K, Kononova O, Ceder G, Olivetti E. 2019. Distilling a materials synthesis ontology. *Matter* 1(1):8–12
100. Mysore S, Kim E, Strubell E, Liu A, Chang H-S, et al. 2017. Automatically extracting action graphs from materials science synthesis procedures. arXiv:1711.06872 [cs.CL]
101. Beta Writ. 2019. *Lithium-Ion Batteries: A Machine-Generated Summary of Current Research*. Cham, Switz.: Springer Nat. Switz. AG
102. Botu V, Batra R, Chapman J, Ramprasad R. 2017. Machine learning force fields: construction, validation, and outlook. *J. Phys. Chem. C* 121(1):511–22
103. Botu V, Ramprasad R. 2015. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *Int. J. Quantum Chem.* 115(16):1074–83
104. Li Z, Kermode JR, De Vita A. 2015. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Phys. Rev. Lett.* 114(9):96405
105. Huan TD, Batra R, Chapman J, Krishnan S, Chen L, Ramprasad R. 2017. A universal strategy for the creation of machine learning-based atomistic force fields. *npj Comput. Mater.* 3:37
106. Behler J. 2017. First principles neural network potentials for reactive simulations of large molecular and condensed systems. *Angew. Chemie Int. Ed.* 56:2–15
107. Rupp M. 2015. Machine learning for quantum mechanics in a nutshell. *Int. J. Quantum Chem.* 115(16):1058–73
108. Chan H, Narayanan B, Cherukara MJ, Sen FG, Sasikumar K, et al. 2019. Machine learning classical interatomic potentials for molecular dynamics from first-principles training data. *J. Phys. Chem. C* 123(12):6941–57
109. Bartók AP, De S, Poelking C, Bernstein N, Kermode JR, et al. 2017. Machine learning unifies the modeling of materials and molecules. *Sci. Adv.* 3(12):e1701816
110. Artrith N, Morawietz T. 2011. High-dimensional neural-network potentials for multicomponent systems: applications to zinc oxide. *Phys. Rev. B* 83(15):153101
111. Artrith N, Urban A, Ceder G. 2017. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Phys. Rev. B* 96(1):14112
112. Nie X, Chien P, Morgan D, Kaczmarowski A. 2019. A statistical method for emulation of computer models with invariance-preserving properties, with application to structural energy prediction. *J. Am. Stat. Assoc.* **https://doi.org/10.1080/01621459.2019.1654876**
113. Behler J. 2016. Perspective: machine learning potentials for atomistic simulations. *J. Chem. Phys.* 145(17):170901
114. Ward L, Wolverton C. 2017. Atomistic calculations and materials informatics: a review. *Curr. Opin. Solid State Mater. Sci.* 21(3):167–76
115. Behler J, Parrinello M. 2007. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.* 98(14):146401
116. Peterson AA, Christensen R, Khorshidi A. 2017. Addressing uncertainty in atomistic machine learning. *Phys. Chem. Chem. Phys.* 19(18):10978–85
117. Maurer RJ, Freysoldt C, Reilly AM, Brandenburg JG, Hofmann OT, et al. 2019. Advances in density-functional calculations for materials modeling. *Annu. Rev. Mater. Res.* 49:1–30
118. Nagai R, Akashi R, Sasaki S, Tsuneyuki S. 2018. Neural-network Kohn-Sham exchange-correlation potential and its out-of-training transferability. *J. Chem. Phys.* 148(24):241737

119. Bogojeski M, Vogt-Maranto L, Tuckerman ME, Müller KR, Burke K. 2019. Density functionals with quantum chemical accuracy: from machine learning to molecular dynamics. ChemRxiv. **https://doi.org/10.26434/chemrxiv.8079917.v1**

120. Snyder JC, Rupp M, Hansen K, Müller KR, Burke K. 2012. Finding density functionals with machine learning. *Phys. Rev. Lett.* 108(25):253002

121. Li L, Snyder JC, Pelaschier IM, Huang J, Niranjan UN, et al. 2016. Understanding machine-learned density functionals. *Int. J. Quantum Chem.* 116(11):819–33

122. Nelson J, Tiwari R, Sanvito S. 2019. Machine learning density functional theory for the Hubbard model. *Phys. Rev. B* 99(7):075132

123. Mills K, Spanner M, Tamblyn I. 2017. Deep learning and the Schrödinger equation. *Phys. Rev. A* 96(4):42113

124. Lei X, Medford AJ. 2019. Design and analysis of machine learning exchange-correlation functionals via rotationally invariant convolutional descriptors. *Phys. Rev. Mater.* 3(6):63801

125. Ryczko K, Strubbe D, Tamblyn I. 2018. Deep learning and density functional theory. arXiv:1811.08928 [cond-mat.mtrl-sci]

126. Kajita S, Ohba N, Jinnouchi R, Asahi R. 2017. A universal 3D voxel descriptor for solid-state material informatics with deep convolutional neural networks. *Sci. Rep.* 7:16911

127. Brockherde F, Vogt L, Li L, Tuckerman ME, Burke K, Müller KR. 2017. Bypassing the Kohn-Sham equations with machine learning. *Nat. Commun.* 8:872

128. Bogojeski M, Brockherde F, Vogt-Maranto L, Li L, Tuckerman ME, et al. 2018. Efficient prediction of 3D electron densities using machine learning. arXiv:1811.06255 [physics.comp-ph]

129. Sinitskiy AV, Pande VS. 2018. Deep neural network computes electron densities and energies of a large set of organic molecules faster than density functional theory (DFT). arXiv:1809.02723 [physics.chem-ph]

130. Schmidt J, Marques MRG, Botti S, Marques MAL. 2019. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.* 5:83

131. Ward L, Agrawal A, Choudhary A, Wolverton C. 2016. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Comput. Mater.* 2:16028

132. Kausar S, Falcao AO. 2018. An automated framework for QSAR model building. *J. Cheminform.* 10(1):1

133. Behler J. 2011. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.* 134(7):74106

134. Barok AP, Kondor R, Csanyi G. 2013. On representing chemical environments. *Phys. Rev. B* 87(16):184115

135. Schütt KT, Glawe H, Brockherde F, Sanna A, Müller KR, Gross EKU. 2014. How to represent crystal structures for machine learning: towards fast prediction of electronic properties. *Phys. Rev. B* 89(20):205118

136. Hansen K, Biegler F, Ramakrishnan R, Pronobis W, von Lilienfeld OA, et al. 2015. Machine learning predictions of molecular properties: accurate many-body potentials and nonlocality in chemical space. *J. Phys. Chem. Lett.* 6(12):2326–31

137. Huang B, von Lilienfeld OA. 2016. Communication: understanding molecular representations in machine learning: the role of uniqueness and target similarity. *J. Chem. Phys.* 145(16):161102

138. Huo H, Rupp M. 2017. Unified representation of molecules and crystals for machine learning. arXiv:1704.06439 [physics.chem-ph]

139. Ward L, Dunn A, Faghaninia A, Zimmermann NER, Bajaj S, et al. 2018. Matminer: an open source toolkit for materials data mining. *Comput. Mater. Sci.* 152:60–69

140. Park CW, Wolverton C. 2019. Developing an improved Crystal Graph Convolutional Neural Network framework for accelerated materials discovery. arXiv:1906.05267 [physics.comp-ph]

141. Korolev V, Mitrofanov A, Korotcov A, Tkachenko V. 2019. Graph convolutional neural networks as "general-purpose" property predictors: the universality and limits of applicability. arXiv:1906.06256 [physics.comp-ph]

142. Chen C, Ye W, Zuo Y, Zheng C, Ong SP. 2019. Graph networks as a universal machine learning framework for molecules and crystals. *Chem. Mater.* 31(9):3564–72

143. DeepChem. 2017. *DeepChem*. **https://deepchem.io/**
144. He K, Zhang X, Ren S, Sun J. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–78. Las Vegas, NV: IEEE
145. Ren S, He K, Girshick R, Sun J. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39(6):1137–49
146. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, pp. 1263–72. Sydney: JMLR
147. Schütt KT, Sauceda HE, Kindermans PJ, Tkatchenko A, Müller KR. 2018. SchNet – a deep learning architecture for molecules and materials. *J. Chem. Phys.* 148(24):241722
148. Schütt KT, Kessel P, Gastegger M, Nicoli KA, Tkatchenko A, Müller KR. 2019. SchNetPack: a deep learning toolbox for atomistic systems. *J. Chem. Theory Comput.* 15(1):448–55
149. Xie T, Grossman JC. 2018. Hierarchical visualization of materials space with graph convolutional neural networks. *J. Chem. Phys.* 149(17):174111
150. Xie T, Grossman JC. 2018. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.* 120(14):145301
151. Lecun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature*. 521(7553):436–44
152. Sheridan RP. 2013. Time-split cross-validation as a method for estimating the goodness of prospective prediction. *J. Chem. Inf. Model.* 53(4):783–90
153. Cawley GC, Talbot NLC. 2010. On over-fitting in model selection and subsequent selection bias in performance evaluation. *J. Mach. Learn. Res.* 11:2079–2107
154. Schwaighofer A, Schroeter T, Mika S, Blanchard G. 2009. How wrong can we get? A review of machine learning approaches and error bars. *Comb. Chem. High Throughput Screen.* 12(5):453–68
155. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. 2011. Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* 12:2825–30
156. Jacobs R, Mayeshiba T, Afflerbach B, Miles L, Williams M, et al. 2019. The Materials Simulation Toolkit for Machine Learning (MAST-ML): an automated open source toolkit to accelerate data-driven materials research. *Comput. Mater. Sci.* 176:109544
157. Molinaro AM, Simon R, Pfeiffer RM. 2005. Prediction error estimation: a comparison of resampling methods. *Bioinformatics* 21(15):3301–7
158. Ren F, Ward L, Williams T, Laws KJ, Wolverton C, et al. 2018. Accelerated discovery of metallic glasses through iteration of machine learning and high-throughput experiments. *Sci. Adv.* 4(4):eaaq1566
159. Ling J, Hutchinson M, Antono E, Paradiso S, Meredig B. 2017. High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates. *Integr. Mater. Manuf. Innov.* 6(3):207–17
160. Cortés-Ciriano I, Bender A. 2019. Deep confidence: a computationally efficient framework for calculating reliable prediction errors for deep neural networks. *J. Chem. Inf. Model.* 59(3):1269–81
161. Gal Y, Ghahramani Z. 2016. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, Vol. 48, pp. 1050–59. New York: JMLR
162. Hill J, Mulholland G, Persson K, Seshadri R, Wolverton C, Meredig B. 2016. Materials science with large-scale data and informatics: unlocking new opportunities. *MRS Bull.* 41(5):399–409
163. Jain A, Persson KA, Ceder G. 2016. Research update: The materials genome initiative: data sharing and the impact of collaborative ab initio databases. *APL Mater*. 4(5):53102
164. Hall E, Stemmer S, Zheng H, Zhu Y, eds. 2014. *Future of electron scattering and diffraction: next-generation instrumentation and beyond*. Rep., Basic Energy Sci. Workshop Future Electron Scatt. Diffr., US Dep. Energy Off. Sci., Washington, DC
165. Henry S, Berardinis L. 2015. *Materials data analytics: a path-finding workshop: workshop results*. Rep., ASM Int., Columbus, OH
166. Belianinov A, Vasudevan R, Strelcov E, Steed C, Yang SM, et al. 2015. Big data and deep data in scanning and electron microscopies: deriving functionality from multidimensional data sets. *Adv. Struct. Chem. Imaging* 1:6

167. Agrawal A, Choudhary A. 2016. Perspective: materials informatics and big data: realization of the "fourth paradigm" of science in materials science. *APL Mater*. 4(5):53208

168. Liu Y, Zhao T, Ju W, Shi S. 2017. Materials discovery and design using machine learning. *J. Mater.* 3(3):159–77

169. O'Mara J, Meredig B, Michel K. 2016. Materials data infrastructure: a case study of the Citrination platform to examine data import, storage, and access. *JOM* 68(8):2031–34

170. Raccuglia P, Elbert KC, Adler PDF, Falk C, Wenny MB, et al. 2016. Machine-learning-assisted materials discovery using failed experiments. *Nature* 533(7601):73–76

171. Holzinger A. 2016. Interactive machine learning for health informatics: When do we need the human-in-the-loop? *Brain Inform*. 3(2):119–31

172. Duros V, Grizou J, Sharma A, Mehr SHM, Bubliauskas A, et al. 2019. Intuition-enabled machine learning beats the competition when joint human-robot teams perform inorganic chemical experiments. *J. Chem. Inf. Model.* 59(6):2664–71

173. Gómez-Bombarelli R, Aguilera-Iparraguirre J, Hirzel TD, Duvenaud D, Maclaurin D, et al. 2016. Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nat. Mater.* 15(10):1120–27

174. Sun X, Krakauer NJ, Politowicz A, Chen W-T, Li Q, et al. 2020. Assessing graph-based deep learning models for predicting flash point. *Mol. Inform.* **https://doi.org/10.1002/minf.201900101**