Machine Learning and Artificial Neural Network Accelerated Computational Discoveries in Materials Science

Yang Hong[1], Bo Hou[2], Hengle Jiang[3], Jingchao Zhang[3*]

[1]Department of Chemistry, University of Nebraska-Lincoln, Lincoln, NE 68588, USA

[2]Department of Engineering, University of Cambridge, CB3-0FA, Cambridge, UK

[3]Holland Computing Center, University of Nebraska-Lincoln, Lincoln, NE 68588, USA

ABSTRACT

Artificial intelligence (AI) has been referred to as the "fourth paradigm of science", and as part of a coherent toolbox of data-driven approaches, machine learning (ML) dramatically accelerates the computational discoveries. As the machinery for ML algorithms matures, significant advances have been made not only by the mainstream AI researchers, but also those work in computational materials science. The number of ML and artificial neural network (ANN) applications in the computational materials science is growing at an astounding rate. This perspective briefly reviews the state-of-the-art progress in some supervised and unsupervised methods with their respective applications. The characteristics of primary ML and ANN algorithms are first described. Then, the most critical applications of AI in computational materials science such as empirical interatomic potential development, ML-based potential, property predictions, and molecular discoveries using generative adversarial networks (GAN) are comprehensively reviewed. The central ideas underlying these ML applications are discussed, and future directions for integrating ML with computational materials science are given. Finally, a discussion on the applicability and limitations of current ML techniques and the remaining challenges are summarized.

*Corresponding Author: Email: zhang@unl.edu; Tel.: +01-402-472-6400

1. The Rise of Machine Learning in the Era of Big Data

Machine learning (ML) has grown into the foundations of countless vital applications, such as speech recognition, image classification, web search, email anti-spam, and customized advertisement.[1-4] Learning in ML is categorically different from learning in traditional programming. In traditional programming, experts write a set of rules using which we deduce the result from a set of observations, which is a deductive process. However, in ML, learning happens via induction, which means general rules are derived from specific observations. In the past decade, the application of ML finally catches up to its early promise with the help of accelerated computing hardware, powerful algorithms, and the explosion of big data. The progress made in deep neural network (DNN) unraveled a series of advances in the field of artificial intelligence (AI).

The definitions of artificial intelligence, machine learning, and deep learning have changed over the years, and their correlations have also evolved. Conventionally, AI is a broader concept which aims to provide approximate solutions to computationally complex problems, while ML is the most common application of AI. Deep learning, such as an artificial neural network (ANN), is considered as a sub-category of machine learning. Nowadays, it has been argued that ML has outgrown its parent. The original and new relationships among these definitions are shown in Figs. 1(a) and (b), respectively. In the new diagram, AI represents a system that is "intelligent" through rules. Machine learning stands for self-learning algorithms that learn models from data, while deep learning is multilayered models that learn representations of data with multiple levels of abstraction.

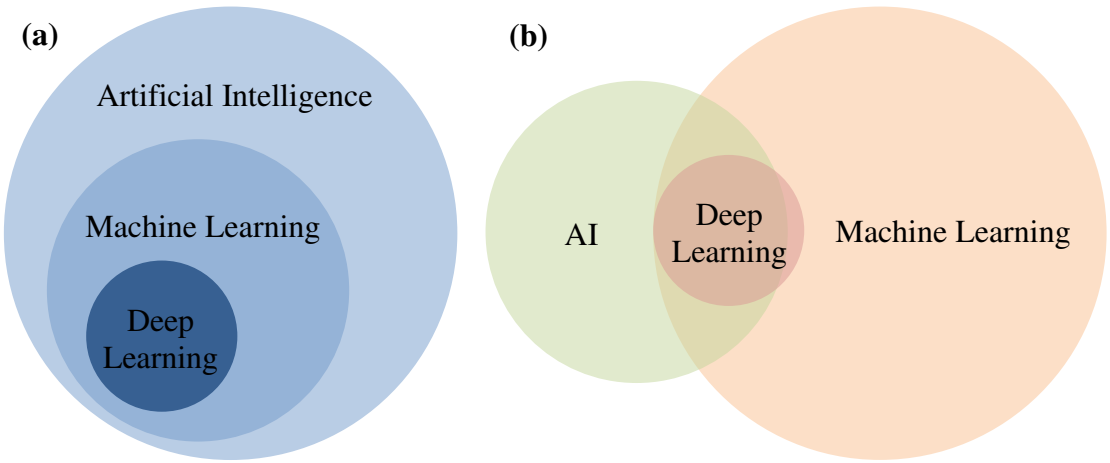

Figure 1. Schematics of (a) conventional and (b) new relationships among artificial intelligence, machine learning, and deep learning. In the new diagram, AI represents a system that is "intelligent" through rules. Deep learning is multilayered models that learn representations of data with multiple levels of abstraction, while machine learning stands for self-learning algorithms that learn models from "data".

Concerning computational materials science, the ML algorithms are applied to learn the rules from atomistic systems. For supervised machine learning algorithms, researchers are responsible for preparing data, creating a representation, and labeling data. The secret behind the versatility of ML is its ability to learn from experience. Given enough data generated by computational methods such as first principles calculations or classical molecular dynamics (MD) simulations, it can figure out the rules and therefore make new predictions. Unlike deterministic algorithms, the ML model learns to create new algorithms. Most ML tasks are about making predictions, which means after training on the given examples, the trained model needs to be able to generalize to cases it has never seen before. In this Perspective, recent applications of ML and ANN in computational materials science are briefly overviewed. In the following, the commonly used algorithms in supervised ML and ANN models are first summarized, followed by

the explanations of the GAN model. Then, a review on the employment of supervised ML algorithms in empirical interatomic potential (EIP) development and ML potential described MD are discussed in details. Predictive models that can directly make property predictions without repetitive experiments of simulations are also summarized. The usage of conventional ML algorithms, as well as ANN models, is covered with training data obtained from both experimental and computational studies. One of the most promising applications of AI in materials science, molecular discoveries using generative models, is elaborated with an emphasis on GAN. Finally, the paper is concluded with highlights on future directions for ML applications in computational materials studies.

2. Machine Learning Approaches in Computational Materials Science

The types of ML systems can be classified into several categories based on different standards. It can be categorized as either an online or a batch system based on whether it can train incrementally. If the ML system predicts results by comparing new data to existing data, it is categorized as an instance-based model. On the other hand, if a predictive model is created based on training data, then it is classified as a model-based system. Another way to categorize ML systems is whether it needs human supervision. Based on this standard, ML systems can be categorized into four different types, *i.e.,* supervised, unsupervised, semi-supervised, and reinforcement. A few notations used in ML and ANN include training instance, hypothesis, hyperparameter, cost function, feature, and target. Training instance stands for the input training data. The training dataset is composed of a collection of training instances. The hypothesis is the mathematical formula to model a particular problem. Hyperparameters are variables

that can be tuned to optimize the model performance, which is measured by the cost function. Features are the input values for the ML models, which is also known as descriptors or input neurons in ANN. Target is one or more values that the model is trained to predict, which is also the output neurons in ANN.

For best generalization, the complexity of the selected hypothesis should match the complexity of the underlying data. If the hypothesis is not powerful enough to describe the data, then there will be an issue of underfitting. On the other hand, if the selected hypothesis is over complicated, then the model will learn from not only the inherent trend of the data but also the noise, which end up with overfitting. The model should be carefully selected considering three factors: the complexity of the hypothesis, the complexity of the training data, and the generalization performance on new examples. A set of assumptions that work well in one domain may work poorly in another. Therefore, there is no universally best model.

2.1 Supervised Algorithms in ML and ANN

Supervised learning is the ML task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. Most material property predictions fall into this category. In supervised learning, the training data for the algorithm includes the desired solutions (labels) created by humans. Some popular supervised algorithms include linear regression, polynomial regression, logistic regression, *k*-nearest neighbors (kNN), support vector machine (SVM), decision trees (DT), random forests (RF), and ANN. The most basic ML model is linear regression, which is expressed as

$$\hat{y} = h_\theta(x) = \theta^T \cdot x\,, \tag{1}$$

where h_θ is the hypothesis function, x is the feature vector, and θ is the model's parameter vector with a bias term. Polynomial regression is very similar to linear regression, with powers of each feature as new features. Aside from training with the extended power features, it can also train on different combinations of feature values to find the best training result. Some more versatile ML algorithms include DT[5], RF[6], and SVM[7]. The decision tree method splits the dataset into binary tree structures, with each node optimized by a cost function. RF is an ensemble training method based on DT. Only a random subset of the DT features is considered in RF training.[8] An SVM regressor, or SVR, is to find a function that deviates from the target by value no more significant than a tolerance margin for each instance. The mathematical expression is to

$$\text{minimize} \qquad \frac{1}{2}\|w\|^2 + C\sum_{i}^{l}(\zeta_i + \zeta_i^*) \tag{2}$$

$$\text{subject to} \qquad \begin{aligned} & y_i - \langle w, x_i \rangle - b \le \varepsilon + \zeta_i \\ & \langle w, x_i \rangle + b - y_i \le \varepsilon + \zeta_i\,, \\ & \zeta_i, \zeta_i^* \ge 0 \end{aligned}$$

where i is the index of training instance, x_i is the input feature, y_i is the target value, l is the number of training data, w is the weight parameter, ζ_i is a slack variable. C stands for the regularization constraint which controls the imposed penalty. The value of C determines the distance of separating hyperplane.

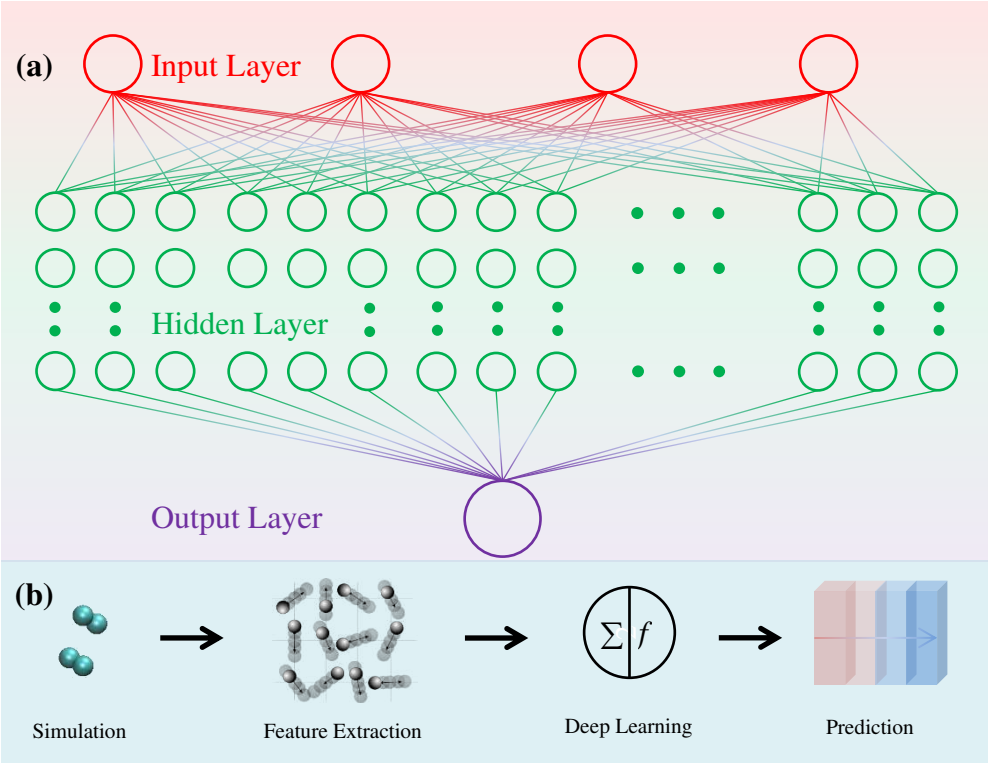

Figure 2. (a) Schematic of a multilayer feed-forward deep neural network. The red, green and purple circles represent input, hidden and output neurons, respectively. (b) A typical machine learning workflow of material property predictions. The structure of the atomistic system is first given. Specific mathematical representation is associated with the input structure information and feed into the ANN as training features. Than trained model can then make predictions on new instances.

Aside from the conventional ML algorithms mentioned above, the ANN models, which are based on the biological neural networks,[9] have also been widely used in computational studies. A neural network is composed of three parts: an input layer, one or more hidden layers, and an output layer. When an ANN has two or more hidden layers, it is called a DNN. Each layer consists of one or more neurons, which represent weights imposed on previous inputs. A typical structure of multilayer ANN is shown in Fig. 2(a). When a neuron helps to predict the correct results, the connections are

strengthened, meaning higher weight values are enforced. During the feed-forward training process, the algorithm first calculates the output of each neuron until the last layers. Afterward, the differences between the predicted and the target outputs are compared to determine how much each neuron contributes to the errors. The learning algorithm is represented as

$$w_{i,j}^{n+1} = w_{i,j}^{n} + \eta(y_j - \hat{y}_j)x_i, \tag{3}$$

where x_i is the ith input, y_j is the target value of the jth output, $\hat{y}_j$ is the predicted value, $w_{i,j}$ is the weight between ith input and jth output, n is the nth step, and η is the learning rate. The learning rate should be selected so that the model training can finish within a reasonable amount of time while eventually converge. The backpropagation training algorithm is used to update the weight values.[10] The errors are backpropagated to the input layer, and the weights are updated based on gradient descent. An activation function is used to calculate the weighted sum of the previous layer. The weighted sum is added with a bias to determine whether this neuron should be activated. Commonly used activation functions include linear function, sigmoid function, exponential linear unit (ELU), and rectified linear unit (ReLU).[11] In computational materials science, the feature values are obtained by numerical simulations, corresponding to one or more target values such as potential energy or mechanical properties. The ANN is trained against this dataset and eventually used to make new predictions.

2.2 Generative Modeling using Deep Neural Networks

Generative modeling includes several different algorithms such as variational autoencoder, reinforcement learning, recursive neural network, and generative adversarial network (GAN). The applications of GAN in accelerated molecular discoveries are overviewed in this Perspective. As opposed to the supervised ML algorithms, GAN is generally considered as unsupervised,[12] which are algorithms that can be used to solve problems like clustering and visualization.[13-15] The GAN needs to simultaneously train two models, a generator and a discriminator. In supervised learning, a prediction is made by learning from labeled input and output. The prediction process can also be referred to as discriminative modeling, which means the training data is used to find a discriminant function that maps each input to output. The trained model must discriminate input variables across the target values and determine the final class or value. From this perspective, the discriminator is trained in a supervised manner. On the other hand, the generator model is trained to learn from the distribution of input variables, which is then used to generate new examples following the same distribution. Therefore, these models are called generative models. In short, the generator is used to generate new examples, while the discriminator is used to classify examples as either real or generated. This approach can model the distribution of both inputs and outputs by generating synthetic data points in the input space. Once adequately trained, the GAN model can generate new examples that are indistinguishable from the original dataset.

For instance, if the input training data follow the Gaussian distribution, then a GAN can be trained to learn the distribution patterns from the input dataset and generate a new variable that plausibly fit into this distribution. The generator and discriminator are trained in a zero-sum game, which terminates at a saddle point where the

discriminator makes wrong decisions about half the time, which means the generator is producing plausible examples following the input patterns. The objective function is expressed as

$$\min_G \max_D V(D,G) = E_{x\in p_d(x)}[\log D(x)] + E_{z\in p_z(z)}[\log(1-D(G(z)))], \quad (4)$$

where G and D stand for generator and discriminator, respectively. And $p_d(x)$ is the data distribution. The input for the generator is a random vector with a fixed length, which is drawn from the Gaussian distribution. The training objective is to encode a representation of the data distribution in a multidimensional vector space, which is also referred to as the latent space. The latent space is a projection and compression of the original data distribution. The data drawn from the latent space is used by the generator as input for new example creations. The GAN models have been successfully employed to construct latent spaces representing images, sounds, and contexts, and generate new artworks by sampling from this space. On the other hand, the discriminator takes input values from either the generator or original data to predict whether it is real or fake. After training, the generator is preserved, and the discriminator will be discarded. GAN has the capability to generate realistic examples across different domains other than the mainstream AI fields. The GAN algorithm has been explored in several pioneer works in computational materials studies.[16-18]

2.3 ML Workflow in Materials Science

The first step in an ML project is to obtain the training data. In computational materials science, the dataset can be calculated from first-principles, classical MD or

lattice dynamics. The size of the test dataset should be large enough to provide high confidence in the overall performance of the system. In general, a popular heuristic is to use 20-30% of the overall dataset for testing purpose. However, in the era of big data with dataset sizes up to a billion, the fraction of data allocated to test dataset has been shrinking. One also needs to avoid allocating an excessive number of a dataset to the testing group, which may have adverse effects on the training results of the ML model. It is also vitally important to make sure that the training and testing dataset is drawn from the same distributions, meaning the computational setup should be consistent. For instance, to obtain the training data from classical MD simulations, the same EIP needs to be used for data collection, unless EIP itself is used as a feature value.

High throughput computation plays an essential role in data generations. Depending on the problem complexity, thousands of calculations might be performed, which requires highly automated workflows using a job scheduler, the pre-processing script for input file preparations and post-processing for results collections. Most computational packages in materials science such as Quantum-ESPRESSO[19], VASP[20], LAMMPS[21] and GROMACS[22] support message passing interface (MPI), which allows the calculation to run across multiple nodes and dramatically shortens the computational time. For regression problems, it is also important to normalize the feature values to the same range, generally between 0 and 1. This is because the numerical values could have orders of magnitude of differences, which makes it extremely hard for optimization. Meanwhile, if the target values are significantly different from the feature values, it can also be normalized to help achieve better performance. Once the training data has been adequately prepared, they will be used as input for the ML algorithms. As

abovementioned, each algorithm may have several hyperparameters. To help find the best parameters for a particular model, the grid search method can be applied.[23]

In supervised regression problems, accessible performance measurements are the root mean square error (RMSE), mean square error (MSE), and mean absolute errors (MAEs). The MSE is expressed as

$$MSE(X,h) = \frac{1}{m}\sum_{i=1}^{m}(h(x^{(i)}) - y^{(i)})^2 \,, \tag{5}$$

where m is the number of instances, $x^{(i)}$ and $y^{(i)}$ are feature and label of instance i, h is the hypothesis, and X is a matrix containing all the instance features. The term performance measure is interchangeable with cost function. Once the calculated error has been optimized to meet the deployment standard, the ML model can then be used to make new predictions. A typical workflow of an ANN-based property prediction is shown in Fig. 2(b).

3. Empirical Potential Development and Machine Learning Driven Simulations

State-of-the-art experiments in the fields of materials science, chemistry and condensed mattered physics had made accelerated progress in recent years. In order to guide and support the experimental discoveries, simulations of realistically sized systems with high accuracy are urgently needed. Classical MD simulations play a vital role in such simulation, but the development of high fidelity EIP has been a bottleneck for such studies.[24-26] Despite the high accuracy provided by *ab initio* density functional theory (DFT) calculation, the tremendous computational cost hinders its applications in

large systems. Meanwhile, the calculations of the electronic structure require intensive computations using DFT, and thus the *ab initio* driven molecular dynamics simulations are restricted to tens of picoseconds and a few hundred atoms. Therefore, EIP remains the best option in many applications for fast energy and force access. Conventional development of a reliable EIP relies on fitting the parameters of certain function forms such as the pairwise interactions given by Lennard-Jones or Morse potentials, or many-body potentials such as Tersoff and Stillinger-Weber potentials. However, the underlying problem of this approach is that the assumed functional form is not appropriate in some scenarios. Instead of imposing an explicit functional form of EIP, ML potential obtained using the Gaussian approximation method provides a more flexible approach to make predictions based on pre-obtained training dataset.[27-30]

Machine learning has significantly facilitated the applications of MD simulation from two perspectives, *i.e.,* the accelerated development of conventional EIP and the direct representation of the MD system using ML potential. For conventional EIP, ML methods such as genetic algorithm have considerably shortened the lifecycle of parameters optimization and become an indispensable tool for potential function global minimization. On the other hand, the ML potential constructs a direct relationship between the atomic structures and the system energy. It does not make any physical approximations on the functional form, and only the electronic structure information is used to model the MD system.

The development of a conventional EIP is a tedious process which requires strong domain knowledge. The fitting parameters are obtained by fitting experimental or

theoretical data such as phonon dispersions, elastic constants, lattice parameters, surface energies, and cohesive energies.[31-36] A general bond-order potential (BOP) could involve dozens of parameters, and it is challenging to find the global minimum in such high dimensions. For instance, the Tersoff-Brenner potential takes the form

$$V_{ij} = f_C(r_{ij})[f_R(r_{ij}) + b_{ij} f_A(r_{ij})], \tag{6}$$

where V_{ij} represents the bond energy, and r_{ij} is the distance from atom i to atom j. The indices i and j run over the atoms of the system. The f_R, f_A, and f_C represent repulsive, attractive pair potentials and cutoff function, respectively. The exponential functions for f_R and f_A are expressed in the Morse potential form as

$$\begin{aligned} f_R(r) &= A\exp(-\lambda_1 r), \\ f_A(r) &= -B\exp(-\lambda_2 r). \end{aligned} \tag{7}$$

where A, B, λ_1, and λ_2 are free parameters related to Pauling constant and dimer strength, which control the overall strength and length scale of the repulsive and attractive potentials. The cutoff function is expressed as

$$f_C(r) = \begin{cases} 1, & r < R - D \\ \frac{1}{2} - \frac{1}{2}\sin(\frac{\pi}{2}(r-R)/D), & R - D < r < R = D, \\ 0, & r > R + D \end{cases} \tag{8}$$

where R and D are free parameters to include only the first-neighbor shell for most structures of interest. The bond-order between atoms i-j is described by the parameter b_{ij}. The angular interactions are described by

$$b_{ij} = (1+\beta^n \xi_{ij}^n)^{\frac{-1}{2n}}, \tag{9}$$

$$\xi_{ij} = \sum_{k \neq i,j} f_c(r_{ik}) g_{ik}(\theta_{ijk}) e^{[\lambda_3^m (r_{ij} - r_{ik})^m]}, \tag{10}$$

$$g(\theta_{ijk}) = \gamma_{ijk}(1 + \frac{c^2}{d^2} - \frac{c^2}{d^2 + (\cos\theta_{ijk} - h)^2}), \tag{11}$$

In total, the Tersoff BOP involves 12 parameters (R, D, A, B, λ_1, λ_2, β, n, λ_3, c, d, and h) which need to be simultaneously optimized. To tackle this optimization problem, a two-stage optimization process has been proposed involving a global minimization using ML generic algorithm[37] and a local minimization using simplex method[38]. Aside from the many-body BOPs, there are numerous pairwise potentials such as 12-6 Lennard-Jones (LJ), Morse, and Buckingham potentials, which are expressed as

$$E = 4\varepsilon[(\frac{\sigma}{r})^{12} - (\frac{\sigma}{r})^6] \qquad r < r_c, \tag{12}$$

$$E = D_0[e^{-2\alpha(r-r_0)} - 2e^{-\alpha(r-r_0)}] \qquad r < r_c, \tag{13}$$

$$E = Ae^{-r/\rho} - \frac{C}{r^6} \qquad r < r_c, \tag{14}$$

where ε and σ are the energy and distance parameters in the LJ potential, D_0 and α are the energy and distance parameters in the Morse potential, A/C and ρ are the energy and distance parameters in the Buckingham potential, and r_c is the cutoff distances.

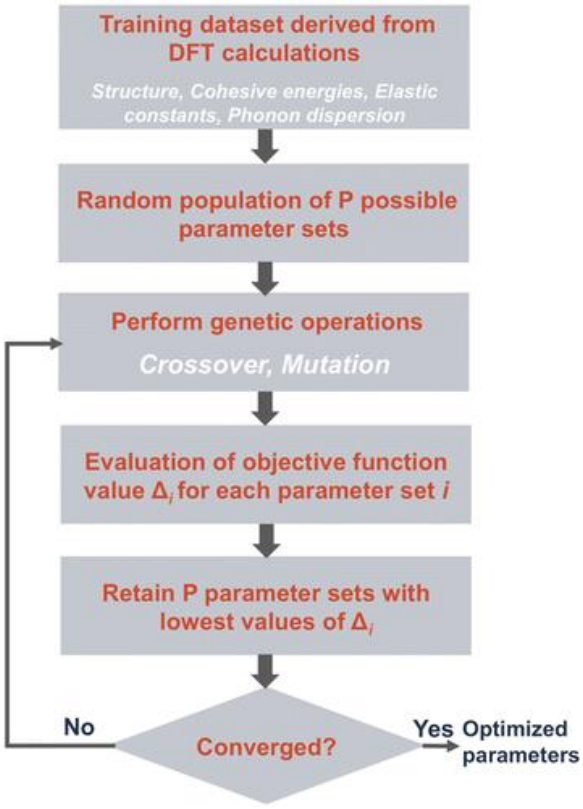

Figure 3. The general workflow of empirical interatomic potential development using ML algorithms. With permission to use from ACS Publications[39].

Using this approach, Cherukara *et al*.[39] successfully parameterized a Tersoff BOP to describe monolayer Stanene. The derived potential can accurately describe the phonon dispersion, cohesive energy, elastic constants, and crystal structure of Stanene with DFT accuracy. The genetic evolutionary framework has also been used to develop EIPs for the computational screening of molecular structure suitable for organic light-emitting diode (OLED) devices.[40] Nguyen *et al*.[41] developed two versions of EIP describing the iron-carbon system based on BOP formalism, which successfully describes the effect of carbon on the phase transition in iron cubic. A typical workflow of EIP development using ML genetic algorithm is shown in Fig. 3.

The EIPs in the abovementioned forms will always be fundamentally restricted by their functional forms. A high fidelity representation of a DFT potential energy surface (PES) can be obtained by the Gaussian approximation potential (GAP) model which facilitates accurate MD simulation approaching *ab initio* precisions. Meanwhile, the computational cost of GAP driven simulations is several orders of magnitude lower than that of comparable calculations involving electronic structure methods. The ML-based approaches use the *ab initio* data utterly different from the general optimization method in conventional EIP development. The ML potential is created by directly fitting the PES solved from electronic structures. The usage of PES for ML potential development was first proposed by Behler *et al*.[42], where the atomic positions in systems of arbitrary size are integrated into functions describing the system energy and forces. Several ML potentials based ANN have been successfully employed on several materials such as graphene[43], graphite-diamond[44], silicon[45], amorphous silicon[46], boron[47], tantalum[48], and amorphous carbon[49]. The GAP model proposed by Rowe *et al*.[43] can successfully reproduce the phonon dispersion, phonon spectra, and thermal expansion properties of graphene. The accuracy of this GAP potential outperforms popular EIPs such as REBO, AIREBO, and Tersoff. Comparison of phonon dispersion profiles generated by GAP and EIPs are shown in Fig. 4.

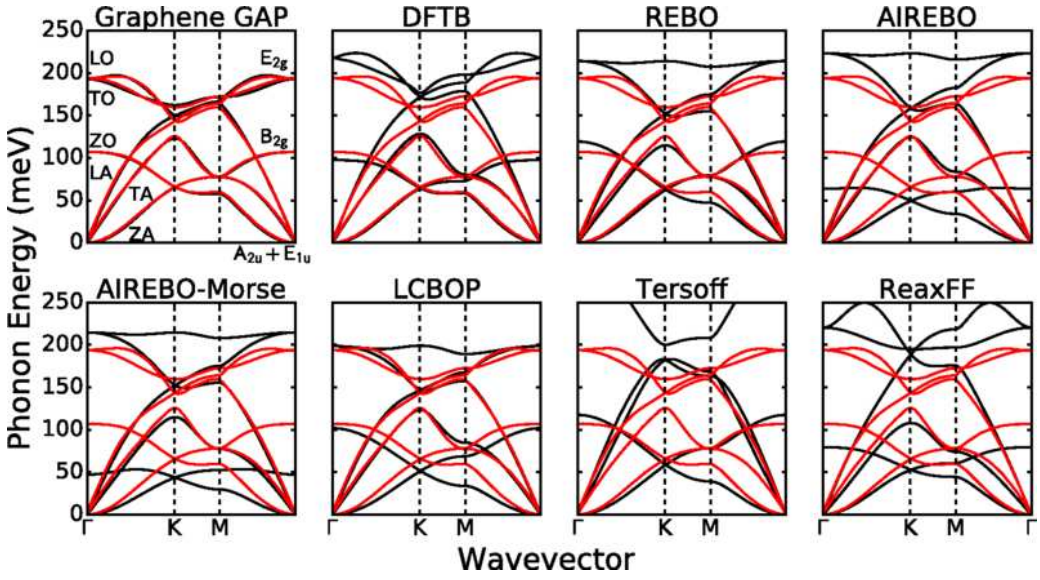

Figure 4. Comparison of the GAP model with other popular EIPs on the phonon spectrum of graphene. It can be observed that the GAP model has the highest accuracy reproducing the experimentally determined phonon spectrum overall the whole wave vector range. With permission to use from APS Physics.[43]

The most challenging step of developing a ML potential is the selection of appropriate structural features to feed into the ML algorithm. Handling of information such as atomic coordinate, bond angle and bond length is trivial in conventional EIPs. However, the atomic coordinates need to be transformed into suitable set of features to describe the system energy and atomic forces. The ANN structure takes vectors of numbers as input and the target value cannot be solely determined on the Cartesian coordinates of the system.[50] On the other hand, the Cartesian coordinate could have variations due to system rotations and translations, yet the ending output value could remain the same. The ML potential must take all these factors into consideration. Several attempts have also been made to directly develop empirical potentials using ML techniques. Machine-learned bond-order potential (ML-BOP) for coarse-grained water

models have been developed with on-the-fly dihedrals.[51] Tutorials on ML potential development have been summarized in other literatures.[52-55] A software package has also been developed by Rodríguez-Fernández *et al*.[56] for automatic PES fitting, which can be applied in a wide range of force-field parameterization problems.

4. Property Predictions using Supervised Algorithms

The pursuit of high accuracy and efficiency EIP is to facilitate the predictions of material properties. The search for materials with exceptional optical, electrical, thermal, and mechanical properties has been going on for decades. However, the traditional experimental or computational approaches are either capital intensive or time-consuming. Even with the help of ML potentials, the property predictions are under fixed conditions such as temperature, dimension, strain, and defect. By combining material science and ML techniques, the materials development and property predictions can be significantly accelerated using data collected from experiments and simulations. In recent years, computational approaches such as first principles and classical MD simulations have been successfully combined with high throughput computations to extract the bandgaps, atomization energies, thermal properties, mechanical properties, and nuclear chemical shifts.[57-61] Traditionally, the predictions of these properties heavily rely on computational approaches such as first principles calculations, molecular dynamics, and lattice dynamics.[62-71] Given enough data, proper ML and ANN models can be trained to directly predict the material properties with only the knowledge of initial conditions without repetitive experiments or simulations. The predictions using ML and ANN only take a fraction of second for a single case compared to hundreds to thousands of CPU hours using the traditional approach.[72] For proper

training of any model, a dataset on the scales of thousands of data points is desired. In this chapter, predictions of the thermal conductivity (κ), interfacial thermal resistance (R), and mechanical properties of various atomistic structures are overviewed.

From the perspective of engineering applications, low-dimensional materials with extremely low or high thermal conductivities have the potential to be used in thermal management devices.[73-77] Using HTC and automatic *ab initio* calculations, Carrete *et al*.[78] constructed an RF regression model to efficiently estimate the thermal conductivity for a large number of compounds. The training data was scanned from ~79,000 half-Heusler entries. It was reported that the thermal conductivity for compounds whose elements in equivalent positions have large atomic radii have the lowest thermal conductivities. The features used in their study are *priori* chemical information, which includes atomic number and weight, position in the periodic table, atomic radius, Pauling electronegativity, and Pettifor's chemical scale. The target value for the RF regression algorithm is thermal conductivity. Seko *et al*.[79] searched a library containing 54779 compounds using Bayesian optimization. The initial data for thermal conductivity were obtained from first-principles anharmonic lattice-dynamics calculations. The feature values used in their models are volume, density, and a set of newly introduced elemental descriptors, which are binary digits representing the presence of chemical elements. To address the problem of feature selection in physical property representations, a follow-up study was performed on 18000 compounds with their cohesive energies computed by DFT calculation.[80] It was reported that the bond-orientation order parameter could significantly enhance the accuracy of lattice thermal

conductivity predictions in ML models. Aside from the conventional ML algorithms, ANN has also been used in the predictions of thermal properties. Wei *et al*.[81] trained a convolution neural network (CNN) to predict the effective thermal conductivities of composite materials. A database using the quartet structure generation set was used to generate composite material structure, and the lattice Boltzmann method is employed to calculate the effective thermal conductivity as target values. The size of the image matrix is 100 × 112. There are two basic setups for the successful training of a CNN for property predictions, *i.e.*, the kernel size and the feature map. A schematic of the CNN used to predict effective thermal conductivity is shown in Fig. 5.

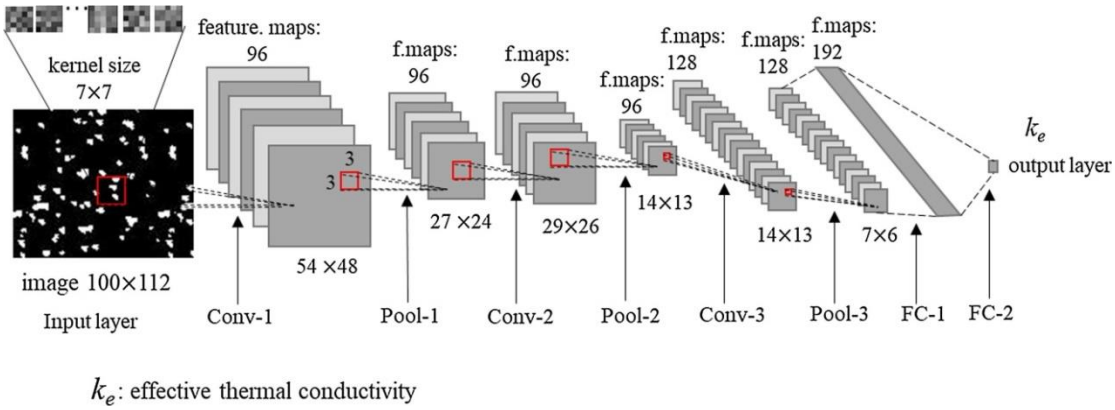

Figure 5. Schematic of the convolution neural network to predict the effective thermal conductivity of composite materials. With permission to use from ScienceDirect.[81]

Aside from the thermal conductivity predictions, several attempts have been made via supervised ML algorithms to predict *R* at materials junctions. Conventional methods for *R* calculations include non-equilibrium molecular dynamics (NEMD) and transient pump-probe methods.[82-84] Zhan *et al*.[85] collected the thermal boundary resistance data for various materials from 62 published journal papers. A series of impact factors, such as measurement temperature, film thickness, and heat capacity, has been considered.

Pearson's correlation among different features and the target value are shown in Fig. 6. Based on the results, the thermal conductivity has the largest positive correlation with elastic modulus. Overall, a total of 876 thermal resistance values for 368 interfaces are collected as a function of temperature and other feature values. Several supervised ML algorithms such as generalized linear regression with and without least-absolute shrinkage and selection operator regularization, Gaussian process regression, and support vector regression have been used to construct models for *R* predictions.

Figure 6. Pearson correlation coefficient map between different materials properties. htcp (heat capacity), thcd (thermal conductivity), debye (Debye temperature), melt (melting point), dens (density), spdl (speed of sound longitudinal), spdt (speed of sound transverse), elam (elastic modulus), blkm (bulk modulus), thex (thermal expansion coefficient), and unitc (unit cell volume). With permission to use from Nature Research.[85]

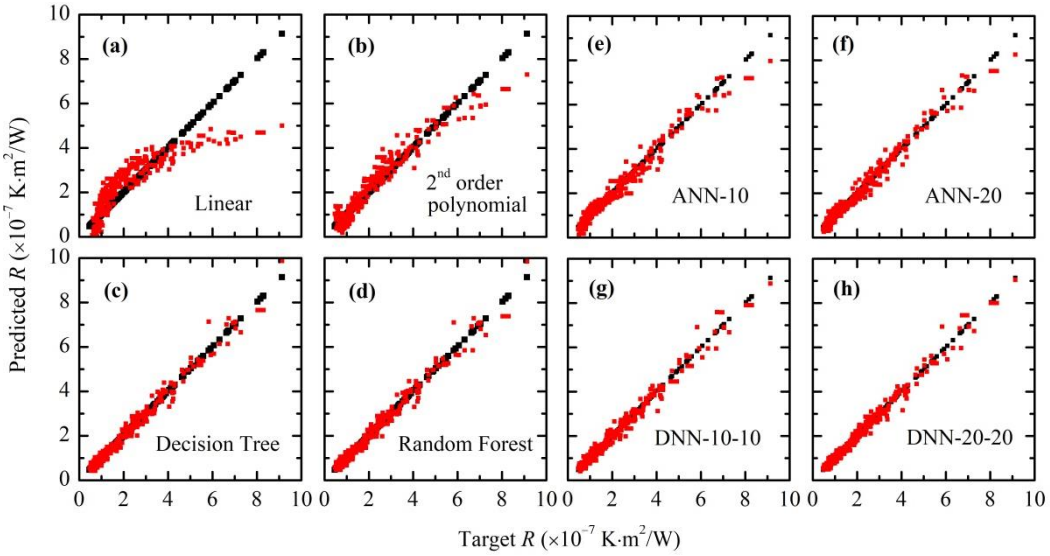

Figure 7. Machine learning results of (a) linear regression, (b) 2nd order polynomial regression, (c) DT, (d) RF, (e) 3-10-1, (f) 3-20-1, (g) 3-10-10-1, and (h) 3-20-20-1 ANN. The red and black square dots represent predicted and the target R values, respectively. Adapted with permission of The Royal Society of Chemistry.[86]

Hong *et al.*[86] trained several ML and ANN models to predict the interfacial thermal resistance between graphene and hexagonal boron nitride. The trained models can predict the *R* values given only the temperature, coupling strength, and in-plane tensile strains. The training dataset is obtained using MD and HTC. Several models, such as linear regression, polynomial regression, DT, and RF, are explored. Four different ANN structures of 3-10-1, 3-10-10-1, 3-20-1, 3-20-20-1 are used. It was reported that the linear regression model could not properly predict the *R* values with high MSE equals 0.854×10^{-7} K·m²/W. The 2nd order and higher order polynomial regressions performed better than linear regression but had worse performance compared to DT, RF, and ANN. Overall, the 3-20-20-1 ANN has the best performance on *R* predictions. A comparison of prediction results using different algorithms are

shown in Fig. 7. However, since the number of features is limited and the size of the dataset is on the scale of thousands, the ANN structures do not have an obvious advantage over traditional ML algorithms.

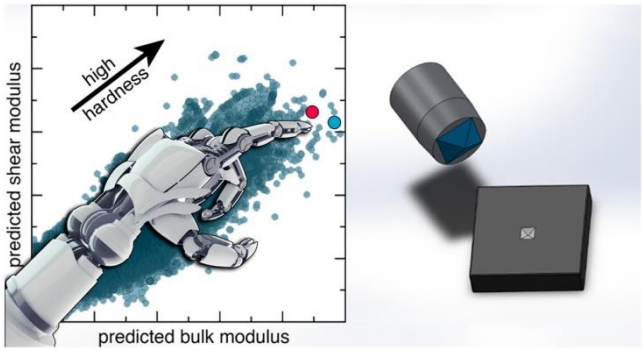

Figure 8. Machine learning directed search for ultra-incompressible, superhard materials. With permission to use from ACS Publications.[89]

Aside from various thermal properties mentioned above, different ML models have also been employed to predict the mechanical properties of several 2D materials such as graphene[72], $MoSe_2$[87], and WS_2[88] under impact factors of system temperature, strain rate, vacancy defect, and chirality. Unlike the thermal conductivity predictions, the mechanical property predictions have multiple outputs, such as fracture strain, fracture strength, and Young's modulus. It is worth noting that when using ANN to predict multiple outputs, the model is optimized to have a minimum MSE on all outputs, whereas the MSE for each output could be further improved if trained separately. For the mechanical property predictions in graphene, several algorithms such as stochastic gradient descent (SGD), kNN, SVM, DT, and ANN are explored. Aside from the SGD method, all models can provide high-accuracy predations on its mechanical properties. The simple models such as kNN even out-performed ANN in the fracture strain

predictions. Using the SVR method, Tehrani *et al*.[89] screened the mechanical properties of 118287 compounds in the crystal structure databases. The model was trained to predict the bulk and shear moduli and successfully predicted two ultra-incompressible and superhard materials, which are confirmed by experimental syntheses. An illustration of their ML approach is shown in Fig. 8.

5. Molecular Discoveries using Generative Adversarial Networks

The accelerated property predictions for several materials using ML techniques have given rise to another challenging task: how to search through the periodic table with more than a hundred elements and find the potential new materials with desired properties. For instance, the possible compounds eligible for drug design is between 10^{23} to 10^{60}.[94] Recent advances in DNN and specifically in generative adversarial networks (GAN) have enabled innovations in creating a new image or composing a symphony.[90-92] This discovery paradigm can be applied to various materials and provided thoughtful guidance to the synthesis of new materials. GAN is one of the non-parametric approaches for deep generative models initially proposed by Goodfellow *et al*.[12]. The generative models can be used to create plausible molecular structures for high-throughput screening, which is the first step in molecular discovery. Generative models such as GAN can illuminate property-structure correlations and use them to guide the molecular designs. With a properly trained generator, the compressed latent spaces can be used to optimize the molecular structures with desired properties. In recent years, the generative models have been used in several fields such as drug design, OLED, solar cell, metal-organic framework, and energetic materials.[95-97]

The first step to build a GAN model for molecular discovery is to define correct molecular representations. The information in each molecule needs to be converted into digital encodings that can be used as input features. Meanwhile, the representation must be able to capture the essential features of each molecule. A one-to-one mapping needs to be constructed between each structure and the corresponding representation. Common representations include 3D coordinates and 2D connectivity graph. While it is possible to use 3D coordinates, the variations with molecular movement and permutation pose a problem for feature extractions. One method is to use a 3D grid of voxels to create a consistent representation, which does not track the molecular reflection, translation, and rotation information. To remain consistent, different molecular structures need to be aligned along a principal axis as directed by principal component analysis.[98] Another popular representation method is molecular graphs, where the molecule is considered as an undirected graph with a set of edges and vertices.

A SMILES string representation method was proposed to convert the molecular graph into texts.[100] Although the features extracted by this approach are more consistent, it has lost some 3D information such as bond length, and it is non-unique. Fortunately, there are a few packages available to standardize this information.[101] Aside from the above methods, the molecule can also be represented by pixel-wise images.[102] This representation method has recently been used by Liu *et al.*[99] for inverse designs of metasurfaces. The workflow of their design method is shown in Fig. 9. The generator was trained on 6500 full-wave finite element simulations for meta-surfaces with different shapes. After training, the unit cell is represented as 64 by 64 binary images,

which are used as input for GAN models. The CNN models were used to train the generator and discriminator.

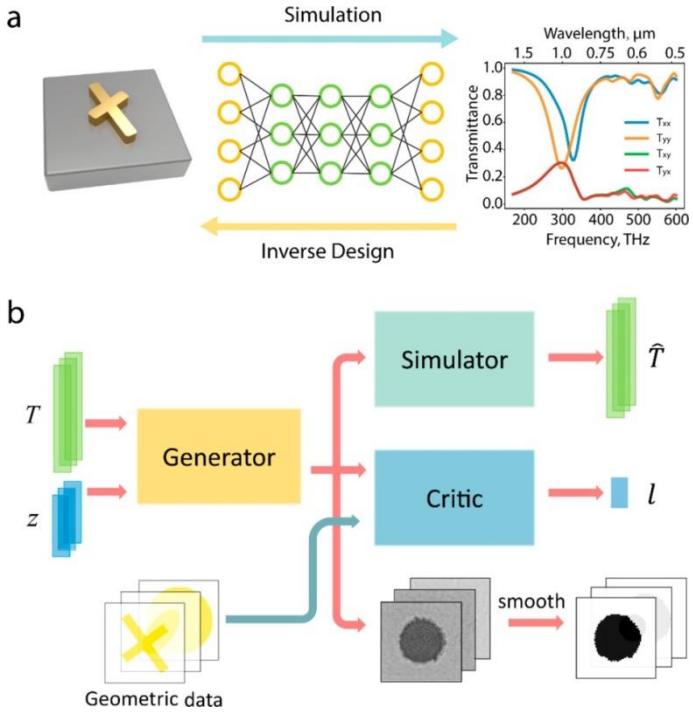

Figure 9. GAN enabled transitioning metasurface design. (a) Illustration of conventional methods. (b) The architecture of the proposed GAN model. With permission to use from ACS Publications.[99]

A few other pioneer works have employed GANs to molecular generations. Lately, a seven layer GAN was trained by Kadurin *et al.*[93] to screen 72 million compounds in the PubChem database, and the structures of the molecules with potential anti-cancer properties are successfully selected. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC) framework was proposed to produce a molecular structure with desirable properties.[103] The constructed model has been successfully used in drug discovery and organic photovoltaic material design. Cao *et al.*[104] proposed an implicit generative model for small sized molecular graphs. After

training, the GAN model can generate high validity and novelty molecular graphs. Besides, the image-based method is ~5 times faster to train than SMILES-based text method. Using GAN combined with reinforcement learning, Putin *et al.*[105] performed *de novo* molecular design using SMILES string-based feature extractions. The underlying distributions of the chemical features are taken into considerations during the GAN training process. Explanations of their neural work and data collection procedures are detailed in another work.[106] A set of machine-learned coarse-grained models have been trained to describe the structure and thermodynamic anomalies of both water and ice at mesoscopic scales.[51] The computational efficiency is two orders of magnitude higher than traditional atomistic models such as TIP4P models and TIP5P. Although the training of a GAN is non-trivial, some open-source software packages have been made available to help facilitate this process.[107]

6. Concluding Remarks and Future Directions

In summary, ML techniques have significantly accelerated the discoveries in computational materials science from several perspectives. The traditional EIP development has benefited from the usage of ML genetic algorithm, and the ML potentials possess significant advantages over conventional EIPs since they directly fit the PES obtained from the electronic structure. For instance, the GAP models are more accurate than traditional EIPs when describing the phonon properties of graphene.[108] On the other hand, the computational efficiency of ML potentials is orders of magnitude higher than those directly solving electronic structures. As a result, the classical MD simulations can be employed on large systems that are more close to practical applications. To build an ML potential, one needs to perform intensive *ab initio*

calculations to generate the training dataset, and perform comprehensive measurements against DFT or experimental results.

The motivations behind the employment of either conventional or ML potentials are to predict the material properties. Given enough data, supervised ML algorithms can directly construct models that can make property predictions without repetitive experiment or calculation. The material property predictions are generally regression problems which require careful selections of feature values and a large volume of training data. Feature extraction is a vital step in constructing a high-performance ML model. The data features used to train the model to have a massive influence on the training results. The first step in feature selections is to apply domain knowledge to extract the most critical impact factors.

On the other hand, one needs to avoid adding irrelevant features which may have a negative impact on the model performance. Several attempts have been made to construct models that can predict the thermal and mechanical properties. Compared to numerical studies, the dataset from experimental studies is more limited. Therefore, the usage of ANN in creating prediction models may not be necessary. In many cases, traditional algorithms such as kNN, DT, RF, and SVM can provide desirable prediction accuracies with fewer hyperparameters to tune. The kNN model even outperformed the ANN model in the prediction of graphene's mechanical properties.[72] On the other hand, in those cases where ANNs are used to construct predictions Nevertheless, neural networks such as CNN still have the edge over traditional algorithms on image-based regressions.[81] As to GAN models, the text-based representations are being replaced by

molecular graphs and 3D chemical structures. There is also a growing interest of directly generating 3D equilibrium structures which are essential in drug designs.

Performance of the ML models depends strictly on the training dataset. The training dataset should be large enough to detect the differences between ML models. If the performance difference between the two algorithms is minimal, for instance, 0.1%, then a small dataset of 100 examples would not be able to detect it. In order to achieve optimal training results, datasets with sizes from 10^3 to 10^4 are standard. Since data is scarce or fragmented, there will be much uncertainty in the prediction. That is why we will rely on the probability theory. Also, we will use linear algebra to manage the large array of data better and do some magic with less effort. All optimizations tricks are solely dependent on calculus. One needs to know the complexity of algorithms in order to achieve optimal model performances.

ACKNOWLEDGEMENT

The authors appreciate the insightful comments and useful discussions from several colleagues on early versions of this article, including Dr. Chen He, Dr. Zhe Zhang, Prof. Stephen Scott, and Prof. Mohammad Hasan. We thank the support from Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative.

CONFLICT OF INTEREST

The authors have declared no conflicts of interest for this article.

REFERENCES

1. Castelvecchi D. Machine learning comes up against unsolvable problem. *Nature* 2019, 565:277-277.
2. Kathuria V. Greed for data and exclusionary conduct in data-driven markets. *Computer Law & Security Review* 2019, 35:89-102.
3. Pangallo M, Loberto M. Home is where the ad is: online interest proxies housing demand. *Epj Data Science* 2018, 7:47.
4. Boselli R, Cesarini M, Mercorio F, Mezzanzanica M. Classifying online Job Advertisements through Machine Learning. *Future Generation Computer Systems-the International Journal of Escience* 2018, 86:319-328.
5. Quinlan JR. Simplifying Decision Trees. *International Journal of Man-Machine Studies* 1987, 27:221-234.
6. Ho TK. The random subspace method for constructing decision forests. *Ieee Transactions on Pattern Analysis and Machine Intelligence* 1998, 20:832-844.
7. Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory* 1992, Pages 144-152.
8. Breiman L. Bagging predictors. *Machine Learning* 1996, 24:123-140.
9. Freeman B, Lowel S, Singer W. Deoxyglucose Mapping in the Cat Visual-Cortex Following Carotid-Artery Injection and Cortical Flat-Mounting. *Journal of Neuroscience Methods* 1987, 20:115-129.
10. Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: David ER, James LM, Group CPR, eds. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*: MIT Press; 1986, 318-362.
11. Zhang C, Woodland PC. Parameterised Sigmoid and ReLU Hidden Activation Functions for DNN Acoustic Modelling. *16th Annual Conference of the International Speech Communication Association (Interspeech 2015), Vols 1-5* 2015:3224.

12. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative Adversarial Nets. *Advances in Neural Information Processing Systems 27 (Nips 2014)* 2014, 27.
13. Chen D, Goya G, Go RS, Parikh SA, Ngufor CG. Improved Interpretability of Machine Learning Model Using Unsupervised Clustering: Predicting Time to First Treatment in Chronic Lymphocytic Leukemia. *Jco Clinical Cancer Informatics* 2019, 3:1-11.
14. Omar AMS, Ramirez R, Haddadin F, Sabharwal B, Khandaker M, Patel Y, Argulian E. Unsupervised Machine Learning Clustering for Stratification of Cardiac Risk in Patients with Exercise Echocardiography Negative for Ischemia. *Journal of the American College of Cardiology* 2019, 73:110-110.
15. Guan C, Yuen KKF, Coenen F. Particle swarm Optimized Density-based Clustering and Classification: Supervised and unsupervised learning approaches. *Swarm and Evolutionary Computation* 2019, 44:876-896.
16. Papadopoulos S, Drosou A, Tzovaras D. Modelling of Material Ageing with Generative Adversarial Networks. *Proceedings 2018 Ieee 13th Image, Video, and Multidimensional Signal Processing Workshop (Ivmsp)* 2018.
17. Li XL, Yang ZJ, Brinson LC, Choudhary A, Agrawal A, Chen W. A Deep Adversarial Learning Methodology for Designing Microstructural Material Systems. *Proceedings of the Asme International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2018, Vol 2b* 2018.
18. Mosser L, Dubrule O, Blunt MJ. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Physical Review E* 2017, 96:043309.
19. Giannozzi P, Baroni S, Bonini N, Calandra M, Car R, Cavazzoni C, Ceresoli D, Chiarotti GL, Cococcioni M, Dabo I, et al. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter* 2009, 21:395502.

20. Kresse G, Furthmüller J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science* 1996, 6:15-50.
21. Plimpton S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics* 1995, 117:1-19.
22. Van der Spoel D, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJC. GROMACS: Fast, flexible, and free. *Journal of Computational Chemistry* 2005, 26:1701-1718.
23. Kangasraasio A, Jokinen JPP, Oulasvirta A, Howes A, Kaski S. Parameter Inference for Computational Cognitive Models with Approximate Bayesian Computation. *Cognitive Science* 2019, 43:e12738.
24. Zhou Y, Smith R, Kenny SD, Lloyd AL. Development of an empirical interatomic potential for the Ag–Ti system. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 2017, 393:122-125.
25. Ito T, Akiyama T, Nakamura K. Systematic approach to developing empirical interatomic potentials for III–N semiconductors. *Japanese Journal of Applied Physics* 2016, 55:05FM02.
26. Mendelev MI, Kramer MJ, Becker CA, Asta M. Analysis of semi-empirical interatomic potentials appropriate for simulation of crystalline and liquid Al and Cu. *Philosophical Magazine* 2008, 88:1723-1750.
27. Polyak I, Richings GW, Habershon S, Knowles PJ. Direct quantum dynamics using variational Gaussian wavepackets and Gaussian process regression. *Journal of Chemical Physics* 2019, 150:041101.
28. Bartok AP, Kermode J, Bernstein N, Csanyi G. Machine Learning a General-Purpose Interatomic Potential for Silicon. *Physical Review X* 2018, 8:041048.
29. John ST, Csanyi G. Many-Body Coarse-Grained Interactions Using Gaussian Approximation Potentials. *Journal of Physical Chemistry B* 2017, 121:10934-10949.

30. Glielmo A, Sollich P, De Vita A. Accurate interatomic force fields via machine learning with covariant kernels. *Physical Review B* 2017, 95:214302.
31. Laio A, Bernard S, Chiarotti GL, Scandolo S, Tosatti E. Physics of Iron at Earth's Core Conditions. *Science* 2000, 287:1027-1030.
32. Ercolessi F, Adams JB. Interatomic Potentials from First-Principles Calculations: The Force-Matching Method. *Europhysics Letters (EPL)* 1994, 26:583-588.
33. Kowalski K, Piecuch P. New coupled-cluster methods with singles, doubles, and noniterative triples for high accuracy calculations of excited electronic states. *The Journal of Chemical Physics* 2004, 120:1715-1738.
34. Yue Y, Zhang J, Tang X, Xu S, Wang X. Thermal transport across atomic-layer material interfaces. *Nanotechnology Reviews* 2015. Vol. 4, Page 533.
35. Li C, Zhang J, Wang X. Phase change and stress wave in picosecond laser–material interaction with shock wave formation. *Applied Physics A* 2013, 112:677-687.
36. Hong Y, Zhu C, Ju M, Zhang J, Zeng XC. Lateral and flexural phonon thermal transport in graphene and stanene bilayers. *Physical Chemistry Chemical Physics* 2017, 19:6554-6562.
37. Jaramillo-Botero A, Naserifar S, Goddard WA. General Multiobjective Force Field Optimization Framework, with Application to Reactive Force Fields for Silicon Carbide. *Journal of Chemical Theory and Computation* 2014, 10:1426-1439.
38. Nelder JA, Mead R. A Simplex Method for Function Minimization. *The Computer Journal* 1965, 7:308-313.
39. Cherukara MJ, Narayanan B, Kinaci A, Sasikumar K, Gray SK, Chan MKY, Sankaranarayanan SKRS. Ab Initio-Based Bond Order Potential to Investigate Low Thermal Conductivity of Stanene Nanostructures. *The Journal of Physical Chemistry Letters* 2016, 7:3752-3759.
40. Halls MD, Giesen DJ, Hughes TF, Goldberg A, Cao YX, Kwak HS, Mustard TJ, Gavartin J. Accelerated Discovery of OLED Materials through Atomic-scale

Simulation. *Organic Light Emitting Materials and Devices Xx* 2016, 9941:99411C.

41. Nguyen TQ, Sato K, Shibutani Y. Development of Fe-C interatomic potential for carbon impurities in α-iron. *Computational Materials Science* 2018, 150:510-516.
42. Behler J, Parrinello M. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical Review Letters* 2007, 98:146401.
43. Rowe P, Csányi G, Alfè D, Michaelides A. Development of a machine learning potential for graphene. *Physical Review B* 2018, 97:054303.
44. Khaliullin RZ, Eshet H, Kühne TD, Behler J, Parrinello M. Graphite-diamond phase coexistence study employing a neural-network mapping of the ab initio potential energy surface. *Physical Review B* 2010, 81:100103.
45. Bartók AP, Kermode J, Bernstein N, Csányi G. Machine Learning a General-Purpose Interatomic Potential for Silicon. *Physical Review X* 2018, 8:041048.
46. Deringer VL, Bernstein N, Bartók AP, Cliffe MJ, Kerber RN, Marbella LE, Grey CP, Elliott SR, Csányi G. Realistic Atomistic Structure of Amorphous Silicon from Machine-Learning-Driven Molecular Dynamics. *The Journal of Physical Chemistry Letters* 2018, 9:2879-2885.
47. Deringer VL, Pickard CJ, Csányi G. Data-Driven Learning of Total and Local Energies in Elemental Boron. *Physical Review Letters* 2018, 120:156001.
48. Thompson AP, Swiler LP, Trott CR, Foiles SM, Tucker GJ. Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials. *Journal of Computational Physics* 2015, 285:316-330.
49. Deringer VL, Csányi G. Machine learning based interatomic potential for amorphous carbon. *Physical Review B* 2017, 95:094203.
50. Blank TB, Brown SD, Calhoun AW, Doren DJ. Neural network models of potential energy surfaces. *The Journal of Chemical Physics* 1995, 103:4129-4137.
51. Chan H, Cherukara MJ, Narayanan B, Loeffler TD, Benmore C, Gray SK, Sankaranarayanan SKRS. Machine learning coarse grained models for water. *Nature Communications* 2019, 10:379.

52. Behler J. Constructing high-dimensional neural network potentials: A tutorial review. *International Journal of Quantum Chemistry* 2015, 115:1032-1050.
53. Behler J. Perspective: Machine learning potentials for atomistic simulations. *The Journal of Chemical Physics* 2016, 145:170901.
54. Bartók AP, Kondor R, Csányi G. On representing chemical environments. *Physical Review B* 2013, 87:184115.
55. Bartók AP, Csányi G. Gaussian approximation potentials: A brief tutorial introduction. *International Journal of Quantum Chemistry* 2015, 115:1051-1057.
56. Rodríguez-Fernández R, Pereira FB, Marques JMC, Martínez-Núñez E, Vázquez SA. GAFit: A general-purpose, user-friendly program for fitting potential energy surfaces. *Computer Physics Communications* 2017, 217:89-98.
57. Montavon G, Rupp M, Gobre V, Vazquez-Mayagoitia A, Hansen K, Tkatchenko A, Müller K-R, Anatole von Lilienfeld O. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics* 2013, 15:095003.
58. Hansen K, Montavon G, Biegler F, Fazli S, Rupp M, Scheffler M, von Lilienfeld OA, Tkatchenko A, Müller K-R. Assessment and Validation of Machine Learning Methods for Predicting Molecular Atomization Energies. *Journal of Chemical Theory and Computation* 2013, 9:3404-3419.
59. Rupp M, Ramakrishnan R, von Lilienfeld OA. Machine Learning for Quantum Mechanical Properties of Atoms in Molecules. *The Journal of Physical Chemistry Letters* 2015, 6:3309-3313.
60. Lopez-Bezanilla A, von Lilienfeld OA. Modeling electronic quantum transport with machine learning. *Physical Review B* 2014, 89:235411.
61. Hansen K, Biegler F, Ramakrishnan R, Pronobis W, von Lilienfeld OA, Müller K-R, Tkatchenko A. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* 2015, 6:2326-2331.
62. Zhang J, Huang X, Yue Y, Wang J, Wang X. Dynamic response of graphene to thermal impulse. *Physical Review B* 2011, 84:235416.

63. Zhang J, Wang Y, Wang X. Rough contact is not always bad for interfacial energy coupling. *Nanoscale* 2013, 5:11598-11603.
64. Zhang J, Wang X. Thermal transport in bent graphene nanoribbons. *Nanoscale* 2013, 5:734-743.
65. Zhang J, Wang X, Xie H. Phonon energy inversion in graphene during transient thermal transport. *Physics Letters A* 2013, 377:721-726.
66. Zhang J, Wang X, Xie H. Co-existing heat currents in opposite directions in graphene nanoribbons. *Physics Letters A* 2013, 377:2970-2978.
67. Hong Y, Li L, Zeng XC, Zhang J. Tuning thermal contact conductance at graphene-copper interface via surface nanoengineering. *Nanoscale* 2015, 7:6286-6294.
68. Zhang J, Hong Y, Yue Y. Thermal transport across graphene and single layer hexagonal boron nitride. *Journal of Applied Physics* 2015, 117:134307.
69. Zhang J, Hong Y, Tong Z, Xiao Z, Bao H, Yue Y. Molecular dynamics study of interfacial thermal transport between silicene and substrates. *Physical Chemistry Chemical Physics* 2015, 17:23704-23710.
70. Wang X, Hong Y, Ma D, Zhang J. Molecular dynamics study of thermal transport in a nitrogenated holey graphene bilayer. *Journal of Materials Chemistry C* 2017, 5:5119-5127.
71. Hong Y, Zhang J, Huang X, Zeng XC. Thermal conductivity of a two-dimensional phosphorene sheet: a comparative study with graphene. *Nanoscale* 2015, 7:18716-18724.
72. Zhang Z, Hong Y, Hou B, Zhang Z, Negahban M, Zhang J. Accelerated discoveries of mechanical properties of graphene using machine learning and high-throughput computation. *Carbon* 2019, 148:115-123.
73. Zobeiri H, Wang R, Zhang Q, Zhu G, Wang X. Hot carrier transfer and phonon transport in suspended nm WS2 films. *Acta Materialia* 2019, 175:222-237.
74. Xie Y, Han M, Wang R, Zobeiri H, Deng X, Zhang P, Wang X. Graphene Aerogel Based Bolometer for Ultrasensitive Sensing from Ultraviolet to Far-Infrared. *ACS Nano* 2019, 13:5385-5396.

75. Wang R, Zobeiri H, Lin H, Qu W, Bai X, Deng C, Wang X. Anisotropic thermal conductivities and structure in lignin-based microscale carbon fibers. *Carbon* 2019, 147:58-69.
76. Zobeiri H, Wang R, Wang T, Lin H, Deng C, Wang X. Frequency-domain energy transport state-resolved Raman for measuring the thermal conductivity of suspended nm-thick MoSe2. *International Journal of Heat and Mass Transfer* 2019, 133:1074-1085.
77. Wang R, Wang T, Zobeiri H, Yuan P, Deng C, Yue Y, Xu S, Wang X. Measurement of the thermal conductivities of suspended MoS2 and MoSe2 by nanosecond ET-Raman without temperature calibration and laser absorption evaluation. *Nanoscale* 2018, 10:23087-23102.
78. Carrete J, Li W, Mingo N, Wang S, Curtarolo S. Finding Unprecedentedly Low-Thermal-Conductivity Half-Heusler Semiconductors via High-Throughput Materials Modeling. *Physical Review X* 2014, 4:011019.
79. Seko A, Togo A, Hayashi H, Tsuda K, Chaput L, Tanaka I. Prediction of Low-Thermal-Conductivity Compounds with First-Principles Anharmonic Lattice-Dynamics Calculations and Bayesian Optimization. *Physical Review Letters* 2015, 115:205901.
80. Seko A, Hayashi H, Nakayama K, Takahashi A, Tanaka I. Representation of compounds for machine-learning prediction of physical properties. *Physical Review B* 2017, 95:144110.
81. Wei H, Zhao S, Rong Q, Bao H. Predicting the effective thermal conductivities of composite materials and porous media by machine learning methods. *International Journal of Heat and Mass Transfer* 2018, 127:908-916.
82. Zhang J, Hong Y, Liu M, Yue Y, Xiong Q, Lorenzini G. Molecular dynamics simulation of the interfacial thermal resistance between phosphorene and silicon substrate. *International Journal of Heat and Mass Transfer* 2017, 104:871-877.
83. Chen W, Zhang J, Yue Y. Molecular dynamics study on thermal transport at carbon nanotube interface junctions: Effects of mechanical force and chemical

functionalization. *International Journal of Heat and Mass Transfer* 2016, 103:1058-1064.

84. Zhang J, Wang X, Hong Y, Xiong Q, Jiang J, Yue Y. Understanding thermal transport in asymmetric layer hexagonal boron nitride heterostructure. *Nanotechnology* 2017, 28:035404.
85. Zhan T, Fang L, Xu Y. Prediction of thermal boundary resistance by the machine learning method. *Scientific Reports* 2017, 7:7109.
86. Yang H, Zhang Z, Zhang J, Zeng XC. Machine learning and artificial neural network prediction of interfacial thermal resistance between graphene and hexagonal boron nitride. *Nanoscale* 2018, 10:19092-19099.
87. Wang X, Hong Y, Wang M, Xin G, Yue Y, Zhang J. Mechanical properties of molybdenum diselenide revealed by molecular dynamics simulation and support vector machine. *Physical Chemistry Chemical Physics* 2019, 21:9159-9167.
88. Wang X, Han D, Hong Y, Sun H, Zhang J, Zhang J. Machine Learning Enabled Prediction of Mechanical Properties of Tungsten Disulfide Monolayer. *ACS Omega* 2019, 4:10121-10128.
89. Mansouri Tehrani A, Oliynyk AO, Parry M, Rizvi Z, Couper S, Lin F, Miyagi L, Sparks TD, Brgoch J. Machine Learning Directed Search for Ultraincompressible, Superhard Materials. *Journal of the American Chemical Society* 2018, 140:9844-9853.
90. Moruzzi C. Creative AI: Music Composition Programs as an Extension of the Composer's Mind. *Philosophy and Theory of Artificial Intelligence 2017* 2018, 44:69-72.
91. Bontrager P, Lin WD, Togelius J, Risi S. Deep Interactive Evolution. *Computational Intelligence in Music, Sound, Art and Design, Evomusart 2018* 2018, 10783:267-282.
92. Chen ZQ, Wu CW, Lu YC, Lerch A, Lu CT. Learning to Fuse Music Genres with Generative Adversarial Dual Learning. *2017 17th Ieee International Conference on Data Mining (Icdm)* 2017:817-822.

93. Kadurin A, Aliper A, Kazennov A, Mamoshina P, Vanhaelen Q, Khrabrov K, Zhavoronkov A. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget* 2017, 8:10883-10890.
94. Polishchuk PG, Madzhidov TI, Varnek A. Estimation of the size of drug-like chemical space based on GDB-17 data. *Journal of Computer-Aided Molecular Design* 2013, 27:675-679.
95. Brown N, Fiscato M, Segler MHS, Vaucher AC. GuacaMol: Benchmarking Models for de Novo Molecular Design. *Journal of Chemical Information and Modeling* 2019, 59:1096-1108.
96. Janicke M, Tomforde S, Sick B. Towards Self-Improving Activity Recognition Systems based on Probabilistic, Generative Models. *2016 Ieee International Conference on Autonomic Computing (Icac)* 2016:285-291.
97. Parrotta L, Faleri C, Del Duca S, Cai G. Depletion of sucrose induces changes in the tip growth mechanism of tobacco pollen tubes. *Annals of Botany* 2018, 122:23-43.
98. Kuzminykh D, Polykovskiy D, Kadurin A, Zhebrak A, Baskov I, Nikolenko S, Shayakhmetov R, Zhavoronkov A. 3D Molecular Representations Based on the Wave Transform for Convolutional Neural Networks. *Molecular Pharmaceutics* 2018, 15:4378-4385.
99. Liu Z, Zhu D, Rodrigues SP, Lee K-T, Cai W. Generative Model for the Inverse Design of Metasurfaces. *Nano Letters* 2018, 18:6570-6576.
100. Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* 1988, 28:31-36.
101. G.Landrum. RDKit: Open-source cheminformatics.http://www.rdkit.org.
102. Goh; GB, Siegel; C, Vishnu; A, Hodas; NO, Baker; N. Chemception: A Deep Neural Network with Minimal Chemistry Knowledge Matches the Performance of Expert-developed QSAR/QSPR Models. *arXiv.org* 2017.

103. Benjamin S-L, Carlos O, Gabriel L. G, Alan A-G. *Optimizing distributions over molecular space. An Objective-Reinforced Generative Adversarial Network for Inverse-design Chemistry (ORGANIC)*; 2017.

104. Cao; ND, Kipf; T. MolGAN: An implicit generative model for small molecular graphs. *arXiv.org* 2018:1805.11973

105. Putin E, Asadulaev A, Ivanenkov Y, Aladinskiy V, Sanchez-Lengeling B, Aspuru-Guzik A, Zhavoronkov A. Reinforced Adversarial Neural Computer for de Novo Molecular Design. *Journal of Chemical Information and Modeling* 2018, 58:1194-1204.

106. Putin E, Asadulaev A, Vanhaelen Q, Ivanenkov Y, Aladinskaya AV, Aliper A, Zhavoronkov A. Adversarial Threshold Neural Computer for Molecular de Novo Design. *Molecular Pharmaceutics* 2018, 15:4386-4397.

107. Maziarka; Ł, Pocha; A, Kaczmarczyk; J, Rata; K, Warchoł; M. Mol-CycleGAN - a generative model for molecular optimization. *arXiv.org* 2019.

108. Rowe P, Csanyi G, Alfe D, Michaelides A. Development of a machine learning potential for graphene. *Physical Review B* 2018, 97:054303.