Parallel Computing: Technology Trends
I. Foster et al. (Eds.)

doi:10.3233/APC200105

Replicating Machine Learning Experiments in Materials Science

Line POUCHARD [a,1], Yuewei LIN [a] Hubertus VAN DAM [a]
[a] *Brookhaven National Laboratory, Upton, NY 11973-5000*

Abstract. Transparency and reproducibility are important aspects of validation for Machine Learning (ML) models that are not fully understood and applies independently of the application domain. We offer a case study of reproducibility that highlights the challenges encountered when attempting to reproduce analyzes obtained with Machine Learning methods in materials informatics. Our study explores prediction results obtained with ML models and issues in training data serving as input. We discuss challenges related to theory-driven and numerical errors in training data, lack of reproducibility across platforms and versions, and effects of randomness when varying hyperparameters. In addition to model accuracy, a main metric of interest in the ML community, our results show that model sensitivity may be equally important for applying ML in domain applications such a materials science.

Keywords. reproducibility, machine learning, materials science, materials informatics

1. Introduction

The design of new materials depends upon the ability to combine precursor materials to make samples and test them using expensive characterization techniques to reconstruct molecular and atomic structures and deduce interesting properties. Discovery through empirical process is slow and largely depends for its success on the intuition of individual scientists. Materials informatics, seeking to elucidate structure-property relationships using complex, multi-scale information in a physically meaningful, statistically robust manner, is becoming more data-intensive due to the advent of high throughput detectors and more complex models [1], [2]. In this context, accelerating discovery requires reducing the combinatorial explosion of the number of potential candidates in the search space for making samples of interest. Machine Learning (ML) methods are increasingly used to predict the relationship between structures and properties and provide guidance to experimentalists for suggesting potentially useful combinations [3]–[5]. Neural Networks have witnessed a revival in the ML community thanks to new methods preventing overfitting, new training methods and the use of computer hardware with GPUs, so that their predictive power has superseded that of other methods, for instance in drug discovery [6], [7]. In order to harness the benefits of new generation ML algorithms in materials informatics and broaden the path of accelerated discovery it is necessary to understand their limits and establish transparency in how results are obtained. Better ways

[1] Corresponding Author: Line Pouchard; E-mail:pouchard@bnl.gov

of explaining results obtained with ML methods will lead towards better reproducibility of results in materials science, and thus better confidence in the ability of ML methods to predict new candidates for experimentation.

Transparency and reproducibility are important aspects of validation for Machine Learning models that are not fully understood and applies across the board independently of the application domain. The National Academy of Science defines reproducibility as obtaining consistent computational results using the same input data, computational steps, methods, code and conditions of analysis [8]. Replicability implies consistent results across studies aimed at answering the same scientific questions. Reproducibility and replicability of scientific results improve when researchers provide access to their codes, methods, data, and execution environments. A survey points to the fact that, of 400 Artificial Intelligence papers recently presented at major AI conferences, only 6% share their algorithms codes, and a third share their data [9].

Understanding the limits of computational reproducibility when dealing with complex mathematical models such as ML models goes beyond making accessible scientific code, training data, and hyperparameters. ML methods present specific challenges in reproducibility related to the building of models, the effects of random seeds, and the choice of platforms and execution environments [10]. ML models that are inherently non-deterministic also pose a problem for the validation of results. If the training sets are divided for cross-fold validation and testing, the partitioning of the sets will affect reproducibility. In materials science, the lack of open access and the heterogeneity of experimental data that describe only few aspects of a material have led to the use of computational structures calculated from physics first principles (ab initio) for the training of models. Experimental data can provide the ground truth to evaluate the accuracy of algorithms but these data are typically small, hard to come by, and not necessarily representative of the problem at hand.

In addition, there is another factor limiting the reproducibility of ML models in materials informatics. Data used as input, especially when they come from atomic and molecular structures computed from theory, can introduce biases due to theoretical approximations. These biases may or may not influence the outcomes for the ML models depending on where the sensitivity of these models lies. In diverse teams, the scientists who build the models are not the ones developing the scientific simulation codes that produce input data. They may not be aware of the presence of these biases and their potential for skewing models, as this requires in-depth knowledge of the parameters, theoretical methods and implementations used for calculating the input data. On the one hand the theoretical approximations made when calculating computational structures are known to those producing these structures, but not available to others. On the other, scientists who build ML models are presumably aware of the sensitivity of their models but not of the theoretical approximations in their input data. The disconnect introduced at the interface of these two groups of actors may result in poor performance prediction that remains unexplained by scrutinizing the ML models alone.

In this paper, we offer a case study of reproducibility that highlights the challenges encountered when attempting to reproduce analyzes and results obtained with Machine Learning methods in materials informatics. Section 2 presents the rationale for using these methods in materials discovery and highlights issues of reproducibility in the training data used by ML models. Section 3 presents several experiments reproducing results. The first experiment uses QM9, a publicly available dataset of computed small organic

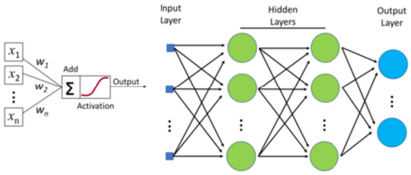

Figure 1. An illustration of a MLP structure.

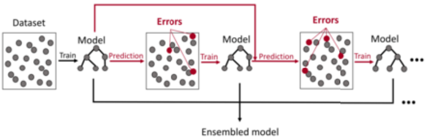

Figure 2. An illustration of a GBT structure.

molecules. It illustrates the fact that, even when data and methods are known, reproducibility is still a challenge, due to hidden theoretical assumptions and lack of transparency. The second set of experiments uses regression-based analysis, including a neural network (MLP) and a Gradient Boasted Tree (GBT), where computational structures are used for training models, and experimental structures for validation (Figures 1 and 2). Section 4 discusses the results and Section 5 concludes with some pointers to future work.

In this paper we make the following contributions:

- an inventory of issues related to reproducicility challenges in the use of ML in materials informatics
- an exploration of theoretical assumptions and uncertainties linked to training data
- several experiments reproducing results obtained with ML methods and computational data in materials informatics
- a comparison of results obtained with several common ML platforms (Tensorflow, PyTorch, lightGBM)
- an exploration of the information required for understanding discrepencies in results
- a discussion of reproducibility challenges when using ML in materials discovery

2. The use of ML in materials science

2.1. Motivation

Many technological advances depend on materials with properties of particular interest. Materials science is a field of research that takes the properties of interest and looks for or designs new materials and characterizes them in the pursuit of those with the desired properties. From a theoretical perspective it is understood that the structure of a material determines its properties. The structure of a material is given by the chemical composition and the positions of atoms as well as the length scales of material features (such as grain sizes, fiber thickness, surface roughness, etc.). At present, there are roughly two approaches to studying materials, an experimental approach, and a computational one. Use of ML algorithms represents a new third approach (Figure 3).

In experimental materials science the general workflow consists of making a sample of a candidate material and experimentally assess its properties (Figure 3a). There are a broad range of properties that might be of interest and an equally broad range of experiments to assess them. Examples are catalytic activity which may be measured in a flow cell, charging characteristics that may be measured in a battery cell, the color characteristics of LEDs, the thermopower of thermoelectric materials, critical temperature and critical current in superconductors, etc. The measured properties have to be understood in terms of the structure of the sample to be able to propose other candidate materials that may better match the set of desired properties. An additional set of experiments particularly targets the structure elucidation but in some cases, e.g. X-ray spectroscopy, experiments alone may not be sufficient to determine the structure.

In computational materials science theoretical models provide a way to compute properties of interest from a given material structure (Figure 3b). In particular ab-initio models based on Schrödinger's equation provide a path to calculating a broad range of properties. Resulting structures are validated against experimental results. A caveat is that these models are typically computationally very intensive and non-trivial approximations are needed to compute results with reasonable compute resources. Nevertheless, these computational models can be used to calculate materials properties. Comparison of the computationally obtained results with the measured properties can help determine the structure of the material sample from which the measurements were obtained.

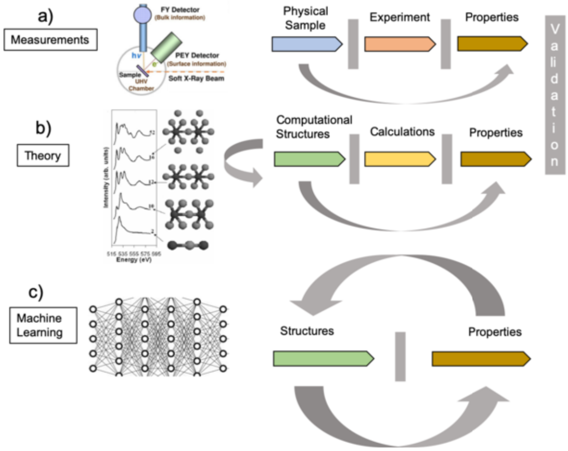

Figure 3.Discovering the structure-property relationship in materials informatics.

However, the problem that remains is that Schrödinger's equations only provides a path to compute the properties from structures. In practical materials science problems the measured properties are given and the material structure needs to be solved. Schrödinger's equation does not provide a convenient formalism to solve such inverse problems. By contrast, machine learning can be used to train a model that correlates input data to output data (Figure 3c). In principle, machine learning does not care about the direction of the relationship. It can be used with material structure as inputs and properties as outputs, but it can also learn the inverse model with the properties as inputs and the structure as outputs. Based on this realization machine learning can be used in essentially two modes. First, it can used as an alternative to ab-initio calculations to predict materials properties from structures, but at a much lower computational cost. Second, it can be used to build models for inverse relationships for which there are few, if any, alternative models available. Both kinds of applications could prove very valuable to moving materials science forward.

2.2. *Theory reproducibility and artifacts in training data*

In the previous section the importance of machine learning to help solve materials science problems was explained. For machine learning to be successful a key ingredient is the availability of diverse sets of accurate training data. The accuracy of the training data is particularly important as ML models typically have no way of knowing the underlying physics they aim to "learn". Instead, the training data is supposed to be a representation of that physics in the form of a large number of individual examples. Systematic errors in the training data will lead to these errors becoming ingrained in the ML model. The availability of mature simulation codes, significant compute resources as well as software for automatically running and analyzing simulations make it possible to generate such data sets automatically. This is the approach taken in a few research projects already [11]–[14]. The outstanding problem then is to ensure that the data obtained in this way is sufficiently accurate. Solving Schrödinger's equation is only practical after making some approximations. These approximations lead to artifacts in the computed results and these artifacts have non-trivial relationships to the underlying approximations. For example, Density-Functional Theory (DFT) has been claimed to be in principle exact. But when using the currently available density functionals, the models of electron structure suffer from unphysical self-interaction errors and strong correlation effects that are poorly described. In another example, configuration interaction (CI) methods with fixed excitation levels, such as singles-doubles (SDCI), produce electron correlation energies that scale incorrectly with the number of electrons.

In fact one may state that the important expertise of practitioners in the field is related to these artifacts and therefore to knowing in which cases and how the results are affected by them. In cases where preferred methods leave doubt about the results it is common to try more advanced methods to reduce uncertainty. This expertise is only available to the ML experts who design models in well-functioning inter-disciplinary teams that share expertise and knowledge. When data obtained by such computations is made publicly available for re-use, the lack of transparency may lead to inaccurate predictions in ML results.

3. Study: Re-running ML models

We designed several experiments in reproducibility to illustrate the issues encountered when using Machine Learning. The first experiment illustrates the lack of transparency when calculating the computational structures and properties of small organic molecules that can be used as training data (section 3.1). The second experiment tests common platforms for training models under various versions of each platform (section 3.2). The third experiment tests the models themselves under various conditions (section 3.3).

3.1. QM9 experiment on the accuracy of training data

Small organic molecules are used in *de novo* drug design and a number of studies with large sets of computational molecules have been published [15] that can be used for training data or benchmarking existing codes. QM9, one of these data sets, was built using DFT methods that are supported by a wide range of quantum chemistry codes. The QM9 data set contains the subset of $C_7H_{10}O_2$ isomers consisting of 6,095 molecules. This subset of molecules is small enough in numbers and the molecules are small enough in size that calculations on this set can be repeated with relatively modest resources. In their paper [16], [17] the authors document two important issues that may affect the results of simulations performed to obtain computational structures for training data. Here these issues are accepted and the work focuses on reproducing the calculations within the reported limits:

- that the reliability and accuracy of DFT depends on the chemical composition and atomistic configurations in molecules and materials,
- and that most reported calculations are done on small molecules implying the existence of a selection bias.

The structures published in QM9 were obtained by translating SMILES strings into structures, and performing subsequent geometry optimization. The final part of the geometry optimization was done with DFT using the Gaussian 09 code [18] but similar capabilities are available in most Gaussian basis set quantum chemistry codes. We used NWChem, an open source package that implements this capability [19]. As these codes have been stable for some time, and the final structures are minimal energy structures it should be straightforward to test the following hypothesis: a geometry optimization started at the published structure should converge at the first point with the same total energy if the same energy expression and basis set (input data) are used.

The basis sets are specifically formatted for the code itself and some codes make historically developed basis sets available with the codes. This is the case for Gaussian09, where the 6-31G(2df,p) basis set used in this experiment was historically developed by Pople and his collaborators on the Gaussian project [20]–[23]. The basis set made available in NWChem comes from the Basis Set Exchange [24]–[26], a community project collecting basis sets and making them publicly available in formats allowable for their respective codes. This project relies on publicly accessible data for the specification of the basis set. Some basis sets have been revisited and refined over time and so discrepancies between a built-in version of a basis set in one code versus that of another code are possible.

Inspite of well documented sources of uncertainty in DFT calculations, for atoms that are spherically symmetric, such as the ones we tested here, highly accurate results are generally achievable. However, our results were significantly different from the results published by the authors of the QM9 data set.

The difference between the two sets of results can largely be attributed to a technical detail related to the way the handling of the angular momentum of the basis functions may be chosen in different codes. Most codes allow one to choose between either Cartesian or spherical harmonic basis functions. Gaussian09 is different in that it allows one to choose between Cartesian and spherical harmonic basis functions independently for d-functions and other angular momentum functions. For example, in Gaussian one can choose to use Cartesian d-functions together with spherical harmonic functions for all higher angular momentum functions. The authors know of no other code that allows this kind of flexibility.

The basis set chosen for calculating the QM9 data set is one that exploits this particular Gaussian feature. It uses Cartesian d-functions and spherical harmonic f-functions. This means that the calculations reported for the QM9 data set can only be reproduced with the Gaussian code and no other code. As Gaussian09 is proprietary code, the authors of the QM9 paper are not at liberty to publish their implementation nor the input data made available with the code. This lack of transparency can affect the re-use of their data sets as training data for ML that may amplify the uncertainties.

3.2. Training models

In a previous experiment by co-author Lin, ML models were used to predict Coordination Numbers (CN) known to characterize size and 3D shape of nanoparticles [27], [28] (Figure 4). Training sets are built using computational data produced from ab initio methods. The model can then be used on experimental spectra to help determine the properties of experimental particles. In the experimental process, XANES spectra, a type of properties, are measured (4a). Coordination Numbers (CN) are calculated (not shown). The computational approach calculates XANES spectra and Coordination Numbers from computational structures (4b). After validation, computational XANES spectra and CNs are used to train the model. Predicted CN (box on the right) are compared to Calculated CN to validate the model. In the ML approach, the trained model can be used with large amounts of experimental spectra pouring out of high throughput detectors to predict expected CN (4c).

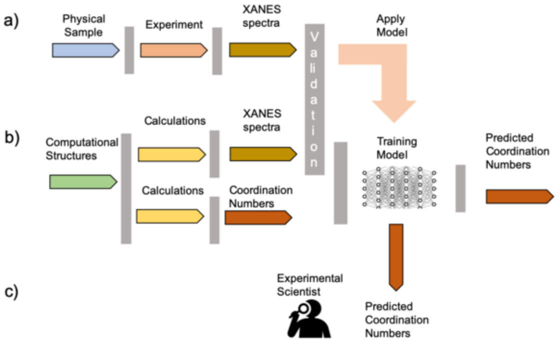

Figure 4.Application of ML for guiding high throughput experiments

Specifically, the machine learning models are defined as a nonlinear function which maps the spectra vector to the coordination number vector. It is a typical regression task. In this work, we evaluated three major powerful and widely used regression models, (1) Gradient Boosted Trees (GBT), an efficient machine learning model that ensembles a set of decision trees; (2) multilayer perceptron (MLP), a classic fully connected neural network; and (3) the most popular type of neural network, (one-dimensional) convolutional neural networks (1D-CNN). For the reproducibility experiment, we focus on GBT and MLP with a relatively deep structure of 5 layers with 400, 400, 200, 200, and 100 nodes respectively ([28].

3.3. Reproducibility across several different platforms

We first evaluate the reproducibility of different machine learning platforms and their versions, and the results are shown in Table 1. Here we test platforms used for the MLP (with Tensorflow and PyTorch) and GBT (with lightGBM) models. Specifically we keep the same random seed for each model to avoid the randomness in model training, and only use different platforms (Tensorflow, PyTorch and lightGBM) with their different versions. We train models on the same training dataset. In Table 1 we define models as reproducible if predictions on the same testing dataset are exactly the same. Our experiment shows that different platforms cannot reproduce exactly the same model, while different versions of the same platform show good reproducibility for TensorFlow and PyTorch.

		Tensorflow		PyTorch		lightGBM		
		1.9.0	1.14.0	1.2.0	1.13.0	2.2.0	2.2.1	2.3.0
Tensorflow	1.9.0	Y	Y	N	N	-	-	-
	1.14.0	Y	Y	N	N	-	-	-
PyTorch	1.2.0	N	N	Y	Y	-	-	-
	1.3.0	N	N	Y	Y	-	-	-
lightGBM	2.2.0	-	-	-	-	Y	Y	N
	2.2.1	-	-	-	-	Y	Y	N
	2.3.0	-	-	-	-	N	N	Y

Table 1. The reproducibility across different deep learning platforms and versions.

3.4. Influence of some random factors in machine learning training

In this section, we investigate the influence of two random factors in two machine learning models that we applied to our Coordination Number prediction task, the multilayer perceptron (MLP) and gradient boosted trees (GBT). We measure the influence of one random factor at a time by fixing all the hyperparameters and other random factors (with the appropriate random seeds), and let free the factor under consideration. We train the model N ($i \in \{1, ...N\}$) times, and with each trained model, we obtain an accuracy x_i on the test data. If accuracy numbers $x_{i}{}_{1}^{N}$ are close to each other, we say the random factor has a small influence, in other words, the model is robust in terms of the random factor. Specifically, the metrics we used to measure the dispersion of accuracy numbers are: 1) the Coefficient of Variation (CV), also known as Relative Standard Deviation (RSD). CV is defined as standard deviation divided by mean; and 2) the Mean Absolute Difference (MAD), defined as $\frac{1}{N(N-1)}\Sigma_{i=1}^{N}\Sigma_{j=1}^{N}|x_i - x_j|$. In this work, we set $N = 5$.

3.4.1. Influence of random factors in MLP

In this section, we investigate the influence of two random factors in MLP, i.e., data order and weight initialization.

Influence of different data orders Most modern machine learning models use stochastic (or mini-batch) gradient descent (SGD) to iteratively optimize the loss function. As a result, data is fed into the model in a random order for training, and this data order adds randomness to the models. Theoretically, in gradients of different orders of (mini-batch) samples, the optimizer uses different paths to get local minimums, and those are usually different. To see how the orders affect the accuracy of the trained model, we keep all the other factors fixed and only leave the freedom for the data order. The results are: the CV of five models is 0.0933, and the MAD is 0.0464.

Influence of different weight initializations The optimizer of the modern machine learning models usually uses a random start point (weights initialization) to start the optimization process which also results in different local minimums. To see how the weight initialization affects the accuracy of the trained model, we keep all the other factors and only leave the freedom for weight initialization. In this task, as we did in last section, we also trained the model five times, and apply the mean of absolute z scores to measure the deviation of all five trained models. The results are: the CV of five models is 0.0349, and the MAD is 0.0171.

Model	MLP		GBT	
Random factor	Data order	Weight init.	Feature selection	Data selection
CV	0.0933	0.0349	0.0035	0.0063
MAD	0.0464	0.0171	0.0023	0.0043

Table 2. The influence of the random factors in model training.

3.4.2. Influence of random factors in GBT

In this section, we investigate the influence of two random factors in GBT, i.e., the feature random selections and data random selections.

Influence of random feature selections In each iteration (tree) of the training, GBT may only use a fraction of the randomly selected features to speed up the learning process while reducing overfitting. We set the features fraction at 0.5, i.e., the half of the features are randomly selected to trained in each iteration. The results are: the CV of five models is 0.0035, and the MAD is 0.0023.

Influence of random data selections In each iteration (tree) of the training, GBT may only also use a fraction of the randomly selected data to speed up the learning process. We set the data fraction is 0.5, i.e., half of the data are randomly selected to trained in each iteration. The results are: the CV of five models is 0.0063, and the MAD is 0.0043.

4. Discussion of challenges

There are many reproducibility issues in materials informatics workflows that can affect the accuracy of results when using ML algorithms. We discuss them below in sequential order:

- Theory-driven errors in the calculation of computational structures
- Numerical errors due to scalability
- Lack of reproducibility across platforms and versions
- Effects of randomness related to training the model itself
- Effects of randomness caused by domain shift when experimental results are used in the test set of a model trained on computational structures.

In cases where the fundamental approximations are not expected to be problematic it is critical to realize that important conclusions are drawn from comparisons of different results. If these results are obtained from calculations that are based on the same approximations then the comparison is likely to benefit from partial cancellations of errors. Therefore it is important to understand what errors are made in a given method and design the calculations to carefully control these errors. This in turn requires consistent choices of input data such as basis sets, energy expressions, convergence criteria, cutoffs, etc. In addition it may require comparisons of results between different codes, as well as validation of the results against more advanced methods.

In computational modeling progress in validating methods has been made through the design and development of a number of test sets. In practice the test sets are typically small (on the order of a hundred atomistic systems) but they usually include typical as well as known difficult systems for a particular property. Based on this diversity it is often assumed that methods that do well on these sets will work well in general. In practice this assumption has to be somewhat qualified. In order for these tests to be readily usable the test cases have to be reasonably small, so that the tests can be run quickly with many different methods, codes and parameter choices. However, with increasingly large atomistic systems can come increasingly severe numerical issues. Hence the performance of a method on a small system may not be indicative for the performance on a large system. These numerical problems are mainly discovered during applications that reach beyond the scale of prior applications, simply for the reason that it is too expensive to systematically test methods on large problems.

Training models is much less computationally expensive than calculating theoretical properties, thus making it an attractive solution. Once built, models can be used to parse "on-the-fly" large amounts of experimental spectra. This is advantageous during the course of an experiment at the beamline, where scientists typically only spend a few four-hour sessions. Models can enable them to guide the course of their experiments and correct it if needed by comparing their new data out of the detector with the predicted results. When this guidance becomes automated, and computationally driven methods steer the course of experiments, the experiment itself is on the way to become autonomous, a long-term goal of the materials science community. However, this will only become possible when criteria for defining and evaluating reproducibility in ML results are well established.

It is common knowledge that training a machine learning model multiple times, even with the same data set, does not usually produce the same model, as different training and testing errors are produced with each run. Several factors influence the randomness introduced at training as seen in the previous section. Model hyperparameters, such as network structure, layer number, neuron number, optimizer, learning rate, batch size, epoch number, activation function, can be treated as a configuration file and easily written into the trained model or an external file. Some platforms can now store hyperparameters and models [29] but a common data model enabling reproducibility does not exist in materials informatics.

In Table 1, we found that TensorFlow and PyTorch cannot reproduce exactly the same model, while different versions of the same platform showed good reproducibility. For lightGBM platform, different versions may or may not reproduce exactly the same model. We would like to emphasize that our evaluations are only based on the specific platforms and versions we test in this paper. Reproducibility across different platform versions really depends on implementation updates of related functions for specific versions, such as optimizers and I/O. In other words, the fact that specific version pairs can or cannot produce the same results may just be due to our accidental choice of version pairs using the same or different implementations of related functions.

We also evaluated the reproducibility of the MLP and GBT based on some random factors. Table 2 showed that GBT has better robustness than MLP, i.e., with certain randomness, the accuracy of GBT is more consistent than MLP. It suggests that the performance of GBT may be more stable than MLP in practice. Although this is generally agreed upon in the ML community, no theoretical proof for it exists. In addition, the best results in terms of accuracy are usually the ones reported, regardless of their robustness to randomness. When ML results are re-used for additional conclusions or guiding experiments, decisions based on both robustness and accuracy will be needed.

With ML models, there are two general classes of errors. The first one that we investigated here concerns how the models perform when trained on the same training dataset with the same hyper parameters. The second one (domain shift) refers to transfer learning or domain adaption and typically draws more attention from ML researchers [30]. Researchers study how models use a training dataset not necessarily representative of the custom data to which models are applied. In our case, models trained on computational structures are used to make predictions about experimental data. Our experiment shows that the first class of errors should not be ignored by practitioners interested in applying these models in their domain science.

5. Future work

In addition to evaluate the influence of the random factors of models in high level, it is also interesting to explore the randomness in low level implementations, such cuDNN deterministic factor. In addition, and independently of the model itself, system level configurations, such as operation system, GPU drivers/library version, eg. CUDA, cuDNN versions also influence the reproducibility of results. The provenance of model execution needs to be extracted and made available.

6. Acknowledgements

This work was performed using resources of Brookhaven National Laboratory operated for the U.S. Department of Energy by Brookhaven Science Associates under contract number DE-SC0012704. The authors gratefully acknowledge some funding support from the Brookhaven National Laboratory under the Laboratory Directed Research and Development 18-05 FY 18-20.

References

[1] K. Rajan, “Materials informatics,” *Materials Today*, vol. 8, no. 10, pp. 38–45, 2005. DOI: 10.1016/S1369-7021(05)71123-8.

[2] J. Hill, G. Mulholland, K. Persson, R. Seshadri, C. Wolverton, and B. Meredig, “Materials science with large-scale data and informatics: Unlocking new opportunities,” *Mrs Bulletin*, vol. 41, no. 5, pp. 399–409, 2016. DOI: 10.1557/mrs.2016.93.

[3] E. O. Pyzer-Knapp, K. Li, and A. Aspuru-Guzik, “Learning from the harvard clean energy project: The use of neural networks to accelerate materials discovery,” *Advanced Functional Materials*, vol. 25, no. 41, pp. 6495–6502, 2015. DOI: 10.1002/adfm.201501919.

[4] N. Wagner and J. M. Rondinelli, “Theory-guided machine learning in materials science,” *Frontiers in Materials*, vol. 3, p. 28, 2016. DOI: 10.3389/fmats.2016.00028.

[5] P. Raccuglia, K. C. Elbert, P. D. Adler, C. Falk, M. B. Wenny, A. Mollo, M. Zeller, S. A. Friedler, J. Schrier, and A. J. Norquist, “Machine-learning-assisted materials discovery using failed experiments,” *Nature*, vol. 533, no. 7601, p. 73, 2016. DOI: 10.1038/nature17439.

[6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015. DOI: 10.1038/nature14539.

[7] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure–activity relationships," *Journal of chemical information and modeling*, vol. 55, no. 2, pp. 263–274, 2015. DOI: `10.1021/ci500747n`.

[8] E. National Academies of Sciences and Medicine, *Reproducibility and Replicability in Science*. May 2019. DOI: `10.17226/25303`.

[9] O. E. Gundersen and S. Kjensmo, "State of the art: Reproducibility in artificial intelligence," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, "The Materials Project: A materials genome approach to accelerating materials innovation," *APL Materials*, vol. 1, no. 1, p. 011 002, 2013, ISSN: 2166532X. DOI: `10.1063/1.4812323`. [Online]. Available: `http://doi.org/10.1063/1.4812323`.

[12] S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, and C. Wolverton, "The open quantum materials database (oqmd): Assessing the accuracy of dft formation energies," *npj Computational Materials*, vol. 1, p. 15 010, 2015. DOI: `10.1038/npjcompumats.2015.10`. [Online]. Available: `https://doi.org/10.1038/npjcompumats.2015.10`.

[13] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R. H. Taylor, L. J. Nelson, G. L. Hart, S. Sanvito, M. Buongiorno-Nardelli, N. Mingo, and O. Levy, "Aflowlib.org: A distributed materials properties repository from high-throughput ab initio calculations," *Computational Materials Science*, vol. 58, pp. 227–235, 2012, ISSN: 0927-0256. DOI: `10.1016/j.commatsci.2012.02.002`. [Online]. Available: `https://doi.org/10.1016/j.commatsci.2012.02.002`.

[14] Computational Atomic-scale Materials Design. (2014). Computational materials repository, [Online]. Available: `https://cmr.fysik.dtu.dk/` (visited on).

[15] (2013). Quantum-machine.org, [Online]. Available: `https://http://quantum-machine.org/` (visited on).

[16] L. Ruddigkeit, R. van Deursen, L. C. Blum, and J.-L. Reymond, "Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17," *Journal of Chemical Information and Modeling*, vol. 52, no. 11, pp. 2864–2875, 2012. DOI: `10.1021/ci300415d`. [Online]. Available: `https://doi.org/10.1021/ci300415d`.

[17] R. Ramakrishnan, P. O. Dral, M. Rupp, and O. A. von Lilienfeld, "Quantum chemistry structures and properties of 134 kilo molecules," *Scientific Data*, vol. 1, 2014. DOI: `10.1038/sdata.2014.22`. [Online]. Available: `https://doi.org/10.1038/sdata.2014.22`.

[18] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, T. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox, *Gaussian 09*, 2016.

[19] M. Valiev, E. Bylaska, N. Govind, K. Kowalski, T. Straatsma, H. V. Dam, D. Wang, J. Nieplocha, E. Apra, T. Windus, and W. de Jong, "Nwchem: A comprehensive and scalable

open-source solution for large scale molecular simulations," *Computer Physics Communications*, vol. 181, no. 9, pp. 1477–1489, 2010, ISSN: 0010-4655. DOI: `10.1016/j.cpc.2010.04.018`. [Online]. Available: `https://doi.org/10.1016/j.cpc.2010.04.018`.

[20] R. Ditchfield, W. J. Hehre, and J. A. Pople, "Self-consistent molecular-orbital methods. ix. an extended gaussian-type basis for molecular-orbital studies of organic molecules," *J. Chem. Phys.*, vol. 54, 1971. DOI: `10.1063/1.1674902`.

[21] M. J. Frisch, J. A. Pople, and J. S. Binkley, "Self-consistent molecular orbital methods 25. supplementary functions for gaussian basis sets," *J. Chem. Phys.*, vol. 80, 1984. DOI: `10.1063/1.447079`.

[22] W. J. Hehre, R. Ditchfield, and J. A. Pople, "Self-consistent molecular orbital methods. xii. further extensions of gaussian-type basis sets for use in molecular orbital studies of organic molecules," *J. Chem. Phys.*, vol. 56, 1972. DOI: `10.1063/1.1677527`.

[23] R. Krishnan, J. S. Binkley, R. Seeger, and J. A. Pople, "Self-consistent molecular orbital methods. xx. a basis set for correlated wave functions," *J. Chem. Phys.*, vol. 72, 1980. DOI: `10.1063/1.438955`.

[24] B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibson, and T. L. Windus, "New basis set exchange: An open, up-to-date resource for the molecular sciences community," *Journal of Chemical Information and Modeling*, vol. 0, no. 0, null, 2019. DOI: `10.1021/acs.jcim.9b00725`. [Online]. Available: `https://doi.org/10.1021/acs.jcim.9b00725`.

[25] D. Feller, "The role of databases in support of computational chemistry calculations," *J. Comput. Chem.*, vol. 17, 1996. DOI: `10.1002/(SICI)1096-987X(199610)17:13<1571::AID-JCC9>3.0.CO;2-P`.

[26] K. L. Schuchardt, B. T. Didier, T. Elsethagen, L. Sun, V. Gurumoorthi, J. Chase, J. Li, and T. L. Windus, "Basis set exchange: A community database for computational sciences," *J. Chem. Inf. Model.*, vol. 47, 2007. DOI: `10.1021/ci600510j`.

[27] J. Timoshenko, D. Lu, Y. Lin, and A. I. Frenkel, "Supervised machine-learning-based determination of three-dimensional structure of metallic nanoparticles," *The Journal of Physical Chemistry Letters*, vol. 8, no. 20, pp. 5091–5098, 2017. DOI: `10.1021/acs.jpclett.7b02364`.

[28] Y. Lin, M. Topsakal, J. Timoshenko, L. Deyu, S. Yoo, and A. I. Frenkel, "Machine Learning Assisted Structure Determination of Metallic Nanoparticles," in *Handbook on Big Data and Machine Learning in the Physical Sciences. Vol 2: Advanced Analysis Solutions for Leading Experimental Techniques*, K. Kleese Van Dam, S. Campbell, K. Yager, and R. Farnsworth, Eds., World Scientific, Nov. 2019.

[29] R. Chard, Z. Li, K. Chard, L. Ward, Y. Babuji, A. Woodard, S. Tuecke, B. Blaiszik, M. Franklin, and I. Foster, "Dlhub: Model and data serving for science," in *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2019, pp. 283–292. DOI: `10.1109/IPDPS.2019.00038`.

[30] Y. Lin, J. Chen, Y. Cao, Y. Zhou, L. Zhang, Y. Y. Tang, and S. Wang, "Cross-domain recognition by identifying joint subspaces of source domain and target domain," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1090–1101, 2016.