Journal of Applied Physics ARTICLE scitation.org/journal/jap

Small data materials design with machine learning: When the average model knows best

Cite as: J. Appl. Phys. **128**, 054901 (2020); doi: 10.1063/5.0012285
Submitted: 29 April 2020 · Accepted: 15 July 2020 ·
Published Online: 3 August 2020

View Online Export Citation CrossMark

Danny E. P. Vanpoucke,[1,2] Onno S. J. van Knippenberg,[3] Ko Hermans,[3] Katrien V. Bernaerts,[1,a)] and Siamak Mehrkanoon[4]

AFFILIATIONS

[1]Aachen-Maastricht Institute for Biobased Materials (AMIBM), Maastricht University, Brightlands Chemelot campus, Urmonderbaan 22, 6167 RD Geleen, The Netherlands
[2]Institute for Materials Research (IMO), Hasselt University, 3590 Diepenbeek, Belgium
[3]CCL Olympic B.V., Keizersveld 30, 5803 AN Venray, The Netherlands
[4]Department of Data Science and Knowledge Engineering, Maastricht University, 6226 GS Maastricht, The Netherlands

Note: This paper is part of the special collection on Machine Learning for Materials Design and Discovery
a)Author to whom correspondence should be addressed: katrien.bernaerts@maastrichtuniversity.nl

ABSTRACT

Machine learning is quickly becoming an important tool in modern materials design. Where many of its successes are rooted in huge datasets, the most common applications in academic and industrial materials design deal with datasets of at best a few tens of data points. Harnessing the power of machine learning in this context is, therefore, of considerable importance. In this work, we investigate the intricacies introduced by these small datasets. We show that individual data points introduce a significant chance factor in both model training and quality measurement. This chance factor can be mitigated by the introduction of an ensemble-averaged model. This model presents the highest accuracy, while at the same time, it is robust with regard to changing the dataset size. Furthermore, as only a single model instance needs to be stored and evaluated, it provides a highly efficient model for prediction purposes, ideally suited for the practical materials scientist.

I. INTRODUCTION

Modern day materials design is becoming ever more reliant on computational modeling and simulations.[1–4] Both of these are used for the elucidation of observations as well as the prediction of new material properties.[5,6] The associated computational methods such as molecular modeling and quantum-mechanical atomistic simulations are well-established[7,8] and have recently been expanded to include artificial intelligence (AI) and machine learning (ML).[3,4,9–16]

In the last decade, AI an ML have acquired a pervasive presence in all branches of science, going from natural languages to fluid dynamics.[17,18] With a steady pace, new achievements are being reported in the field of materials research.[3,9,19–25] In general, these achievements are rooted in the access to suitable large datasets, both theoretically and experimentally.[14–16,26–28]

However, even though such big datasets (and access to them) are becoming common place,[3,27,29–32] they do not represent the datasets most materials researchers work with on a day-to-day basis. Within general experimental material research projects, researchers generally produce no more than a hand full of data points (c.q. samples) when optimizing a production method, synthesizing a new material or tuning an existing one for a specific application. This stands in stark contrast to the phase-space spanned by the variable parameters of the experiment. The reason for the small number of samples generally originates in their cost; either in time for creating the samples or the monetary cost of the base materials and used machinery.[25] On the other hand, in computational materials science, one also finds small datasets to be more common, specifically when the research is aimed at corroborating and elucidating experimental results. There may also be a cost factor involved. For example, the computational cost of high-quality quantum-mechanical material simulations rises steeply with the system size and model accuracy.[33,34]

Given the success of ML in computational materials design within the context of large data sets,[28–31] there is a natural desire to also apply these methods on the small datasets produced in materials research and reap their benefits. The use of AI and ML in such cases is often aimed at the improvement of *design of experiments* for materials optimization, quite often in combination with robotic or other automation.[22,25,26,35–39]

In this context, *active learning* deserves mentioning.[26,39,40] Although it is not necessarily aimed at small datasets itself, active learning methods build a model by gradually learning the data. Starting from a small (randomly) selected dataset, the initial model is trained. Subsequently, this model is used to select—this is the active component—which additional data points should be added to the training set from the master pool. The model is then trained anew on this extended training set. The loop of selecting additional data points and retraining is repeated until a predefined target is reached or a calculation budget is spent. Two interesting examples are given by the work of Gubaev *et al.*,[26] who used active learning to predict the molecular properties in benchmark chemical datasets, and De Grave *et al.*,[39] who implemented an active k-optimization which searches for the k best performing instances instead of only the single best instance.

Within the context of materials physics and chemistry, several authors have applied ML to relatively small datasets. Ghafari *et al.* successfully predicted the properties of ultrahigh-performance concrete by applying a backpropagated neural network on a dataset of 53 points.[9] Houben *et al.* studied the co-polymerization of styrene and butyl acrylate within the context of a closed loop approach.[35] Using the MOAL framework,[41] which combines Gaussian processes (GPs) and evolutionary algorithms, they performed *in silico* experiments that gave rise to several successful theoretical recipes. The validation of these models showed not all of them to be as successful in the lab; however, it highlights the potential as a prescreening method. In contrast, working with lab experiments, they showed that starting from a mere five training experiments, a successful multi-target goal was reached within only 17 iterations.[35] Schweidtmann *et al.* used automated selection to extend an experimental dataset of only 20 points to a densely populated Pareto front with only 50 additional experiments.[42] They later successfully applied this method on the Sonogashira reaction and the Claisen–Schmidt condensation reaction.[37] Zhang and Ling proposed a strategy to improve ML on small datasets (∼100 data points), which they called "crude estimation of property."[38] In this strategy, a crude guess for the target property of interest, obtained from simple empirical models or computationally cheap(er) simulations, is added as an additional feature for the ML model to train on. This lead to a clear improvement for their presented examples. An automated polymer synthesis platform using ML was developed by Junkers and co-workers.[43] Combined with in-line characterization using a benchtop NMR, their machine-assisted synthesis setup allows for polymer synthesis with predefined conversion rates.[36]

Such a combination of AI and robotics is quickly gaining importance in the field of flow chemistry. However, it often results in a very data hungry setup. Coley *et al.* presented how a combination of neural networks, binary classifiers, and tree-searches can enhance a robotically controlled flow chemistry reactor.[44] This approach requires large amounts of human created and expert refined chemical recipe files to train the neural network in linking molecular fingerprints (e.g., SMILES) to reaction rules. Also, Wang *et al.* used a neural network trained on molecular fingerprints to predict the octanol–water partitioning coefficients while screening for green solvents.[45] Also here, training of the neural net required more than 10 000 data points. A behavior also observed by Cendagorta *et al.* in their performance study of ML models for high-dimensional free energy surfaces.[28] In contrast, Menon *et al.* presented a Hierarchical Machine Learning (HML) model, which provides useful predictions on the mechanical properties of elastomers or predict novel dispersants using only a handful of experiments.[46,47]

Although these examples show that, even in the context of small datasets, the use of AI and ML can be successful for materials research, the quality of the obtained models is often defined in an *ad hoc* fashion and their dependence on the used dataset is not discussed.

In this work, we critically investigate the role of the small dataset itself (< 25 data points), within the context of ML based regression analysis. An important limitation of ML on small datasets is highlighted: the strong dependence on the data points is considered for the training of the trained model instance and its quality [cf. Fig. 1(a)]. This behavior is demonstrated using both a

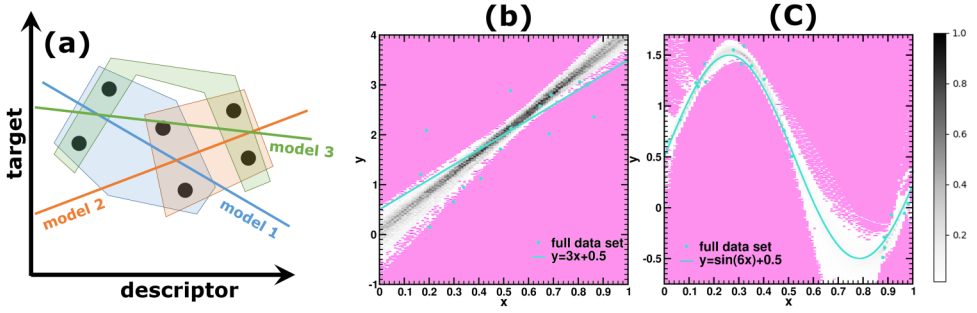

FIG. 1. (a) Schematic representation of model instances for different training sets created from a small dataset. (b) and (c) Heat map of 1000 model instances generated by 1000 random train-validation splits (80/20%) of an $n = 20$ point dataset. The datasets Lin^{20} and Sin^{20}, as well as the underlying theoretical model instance, are indicated (cf. Sec. III A). For the Lin^{20} dataset, the model instances are obtained using a linear regression, while a sixth order polynomial regression is used for the Sin^{20} dataset.

synthetic and an experimental dataset. We show that for these small data sets, model averaging (or an ensemble model) provides a simple and elegant solution that produces consistent models of the highest quality.

II. LIMITATION OF SMALL DATASETS FOR MACHINE LEARNING BASED REGRESSION

In regression analysis, the goal is to predict a continuous target value from a set of input descriptors (also called features). Within the context of ML, it is common practice not to fit or train the regression on the entire dataset but instead to split this dataset into a *training* and *test set*.[27,48] The training set is used to fit the regression, and the quality of the resulting model (i.e., generalization error) is then assessed by quantifying the performance of this model instance, using the mean-absolute-error (MAE) or root-mean-squared-error (RMSE), on the test set. Furthermore, in more complex (regression) models, one is required to have an additional "test set"—*the validation set*—at hand to fit the hyper parameters of the model. From this, it quickly becomes clear why ML approaches are considered data hungry, and why their rise coincides with that of big data.[4,12,15,16,28]

For a sufficiently large dataset, the performance measures will not be influenced much by the details of how the data were split, assuming a random splitting.[4938] However, for very small datasets, this is not the case. Figure 1 shows a schematic example of a dataset containing six data points. Three possible training sets of four data points are indicated as well as their associated hypothetical model realizations. The three model realizations are clearly different, and although they provide a good fit for their respective training set, they do not provide a good model realization for the full six-point dataset. This simple example highlights a significant caveat of ML models in the limit for small datasets, which goes beyond the deterioration of the quality with decreasing the system size.[28,38]

To avoid possible confusion, let us briefly define some specific terms and show how they are used in this work. We use the term "model" to refer to the abstract representation of a functional relation (e.g., $y = ax + b$), while a realization of such a model (e.g., $y = 5x + 3$) is indicated as a "model instance." The test set used to compare the quality of individual model instances is referred to as the "validation set" (as the role of the model coefficients is akin to that of hyperparameters in big data ML applications). Alternately, the term "test set" is used to refer to the test set used to compare the quality of different models (or specifically highlighted model instances).

III. RESULTS

Because the number of possible train-validation splits of a dataset is given by the binomial $nk = \frac{n!}{k!(n-k)!}$, with n the size of the full dataset and k the size of the validation set,[50] an exhaustive sampling becomes quickly impractical, even for small datasets of 20–40 points. Therefore, we choose to use a Monte Carlo approach to randomly generate samples of 1000 train-validation splits throughout this work.

A. Datasets

We consider two types of datasets: (a) synthetic datasets and (b) small experimental datasets.

1. Synthetic data

Because the small size of the experimental datasets limits the possibility of showing trends upon increasing the dataset size, we consider two simple synthetic datasets for this specific purpose. These sets are aimed at simplicity and clarity, therefore, they are limited to one-dimensional models.

The first type of n data point dataset (Lin^n) is generated by the linear model instance,

$$y = 3x + 0.5 + \mathcal{N}(0, 0.75) \quad x \in [0...1], \tag{1}$$

in which y represents the target property, x is the input descriptor, and a normal-distributed noise, $\mathcal{N}$, is added to emulate experimental spread of the data.

In addition to this linear data, also a non-linear synthetic dataset (Sin^n) is used, generated by the following model instance:

$$y = \sin 6x + 0.5 + \mathcal{N}(0, 0.10) \quad x \in [0..1]. \tag{2}$$

Using these two model instances, datasets of arbitrary size n are generated using a sample of n uniform randomly distributed x. For practical purposes, two $n = 1000$ sets are generated. One set is used exclusively for testing purposes, while the other is used to create the train-validation datasets. To emulate the expansion of an experimental dataset through the addition of new experiments, we grow the synthetic train-validation datasets of size m by taking incrementally larger fractions (c.q. the first m points) of the $n = 1000$ dataset (cf. Fig. 2).

2. Experimental dataset: A pressure-sensitive adhesive coating

The experimental dataset is gathered from a pressure-sensitive adhesive coating (Ref. 51). This specific coating is produced as a mixture of three components: 2-ethylhexyl acrylate (2EHA), acrylic

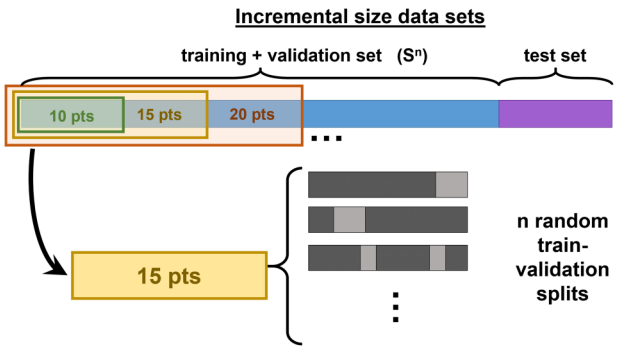

FIG. 2. Schematic representation of the partitioning of the experimental dataset.

acid (AA), and n-vinyl caprolactam (VCL). As such, the fractions of the compounds in the mixture are considered the initial descriptors. Although measurement data are provided for nine different target properties, we only consider two in the current work since not all target properties are provided for all data points. Only for a small subset of 25 data points, S^{25}, the set of target properties is complete.

The target property for which most data points are available (i.e., 45) is selected: the *90° peel strength on steel* (PSS). The second selected target property is the *elongation at break* (EB), which shows the highest correlation with the input parameters (cf. correlation matrix Fig. 3) and for which 31 data points are available. The PSS data are expressed in N/in., while the EB data are expressed in % of elongation.

B. Linear regression models

For the sake of simplicity, we first consider linear regression models.

1. Synthetic data

The synthetic datasets provide an ideal starting point for the investigation of the impact of the train-validation set splitting of a small dataset. In the following, we model the linear datasets, Lin^n—generated using Eq. (1)—using a linear regression model. The nonlinear datasets, Sin^n—generated using Eq. (2)—are modeled using a 6*th* order polynomial regression.[52]

The heat maps in Figs. 1(b) and 1(c) show a wide spread of model instances obtained for the synthetic datasets. More interestingly, this spread appears not to be uniform. Instead, it seems centered on an "average," which shows reasonable agreement with the (noiseless) underlying theoretical model instance of the synthetic data. Increasing the size of the dataset results in a narrowing of this distribution, as well as a convergence toward the underlying model, as is shown in Fig. 4.

2. Experimental data

In the case of the experimental datasets, the same behavior is observed. Let us start by considering a multiple linear regression model (MLRM), which predicts a property Y (either PSS or BE) from the three descriptors x_i =2EHA, VCL, AA. Training a MLRM on 1000 train-validation splits of the PSS data, using different dataset sizes $n = 10, 15, 20, 25, 40$, a distribution of the four fitting coefficients is obtained (cf. Fig. 5). In the case of $n =$ 10 data points, this distribution is very wide and narrows slowly with increasing dataset size n. More interestingly, the distributions also become much smoother with the increasing dataset size. The latter is a consequence of the growing number of possible combinations to select the training set from the dataset.[53] The same behavior is found for the EB dataset, as is shown in Fig. S.4 in the supplementary material.

A second interesting observation is that the mean value of the distributions converges to different values for the subsets of the S^{25} and S^{40} PSS datasets. This may seem strange as S^{25} is a subset of S^{40}.

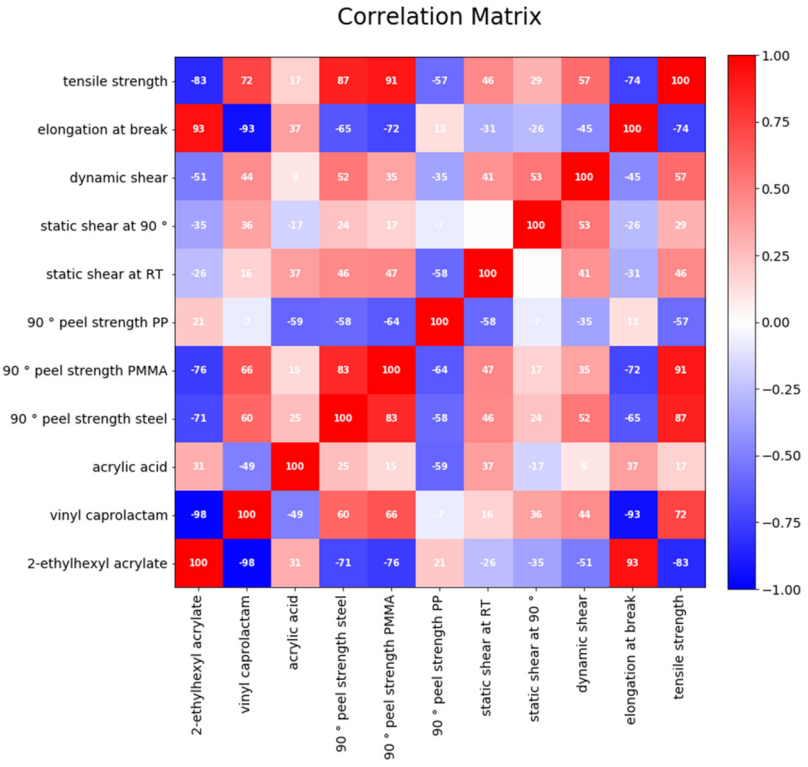

FIG. 3. Correlation matrix obtained for the S^{25} dataset.

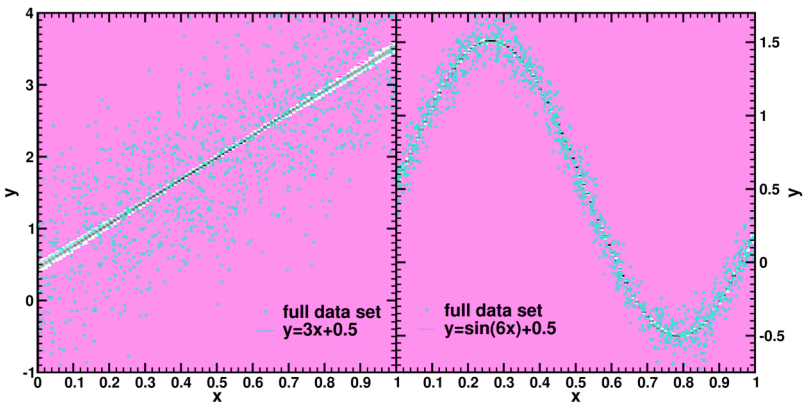

FIG. 4. Similar to Figs. 1(b) and 1(c), using an $n = 1000$ point dataset.

However, one needs to consider that the 20 point subsets (cf. green curves in Fig. 5) are randomly selected from the S^{25} and S^{40} sets. As such, it is possible for these 20 point sets not to have a single data point in common. The impact on the position of the distribution is furthermore amplified by the use of a standard scaler. Due to the small size of the datasets, the transformation performed by the standard scaler strongly depends on the data available. The resulting variations of the transformed data, in turn, significantly impacts the coefficients of the model instances (cf. the supplementary material for more details).

C. Different data–different results: Repetition of the computer experiments

The dependence of the model parametrization on the specifics of the data used (e.g., variances and errors of property data) is an important challenge in machine learning.[14] From the above, it is clear that this challenge is significantly amplified within the context of small datasets. To gain further insight into the impact of the dataset on the (mean) value of the coefficient distributions, we perform 100 repetitions for various dataset sizes n, each giving rise to a distribution of 1000 model instances.

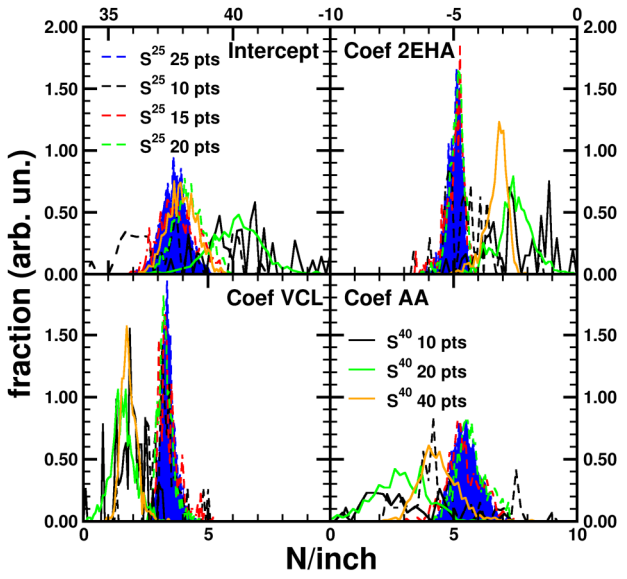

FIG. 5. Distribution of the intercept and coefficients of a linear regression performed on the PSS data.

1. Synthetic data

For the synthetic Lin^n datasets, each repetition made use of a new independently drawn set of random data points. For each repetition, we collect information on three model instances: (1) the instance with the *Best* RMSE validation score, (2) the instance with the *Worst* RMSE validation score, and (3) the model instance of which the model coefficients are constructed as the *Average* of the model coefficients of the ensemble of 1000 model instances. The quality of the three models is quantified by the MAE on a test set of 1000 (new) data points.

The resulting mean of the 100 repetitions is presented in Fig. 6. The 95% confidence intervals (CIs) are obtained using the bias-corrected and accelerated (BCa) bootstrap method, with the acceleration coefficient estimated from a jackknife resampling of 2000 bootstrap replications.[54,55]

As expected, for sufficiently large datasets (in this case $n > 100$), the linear model converges to the underlying theoretical model, with the CI becoming narrower. It is interesting to note that the results for the *Best* and *Average* model instances are nearly identical. Furthermore, they present a smooth evolution of the mean model coefficients. In contrast, the *Worst* model instance shows strong oscillations of the mean model coefficients and a much wider CI for small dataset sizes. For large dataset sizes (*c.q.*, several hundred data points), the three model instances show similar results, reminding the need for big data in successful ML approaches.

The calculated MAE shows a consistent improvement of all models with increasing dataset sizes, following a power law behavior and reaching the theoretical limit for dataset sizes approaches 1000 data points. For all dataset sizes, the *Best* and *Average* model instances appear to provide nearly identical quality. More importantly, for small dataset sizes, they clearly outperform the *Worst* model instance. This indicates that for a given small dataset, the

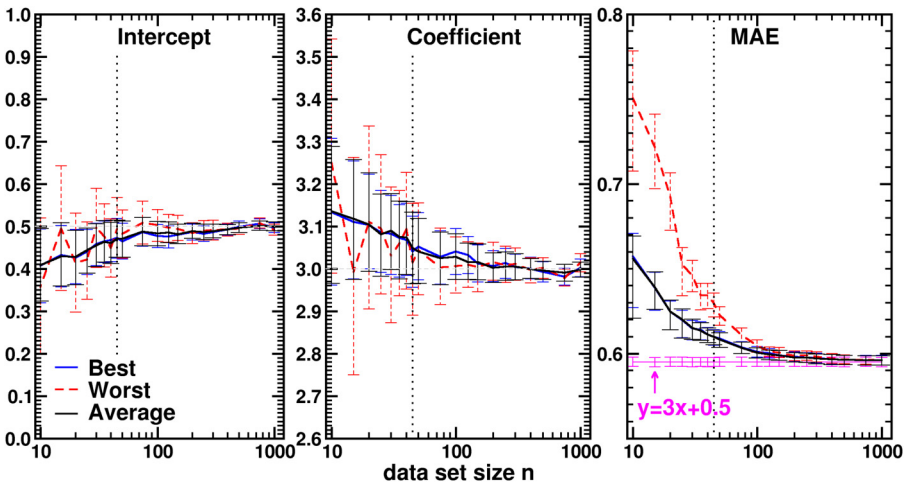

FIG. 6. Mean values and 95% confidence intervals, based on 100 repetitions, of the intercept (left) and coefficient (middle) predicted for the Lin^n synthetic data, using the *Best*, *Worst*, and *Average* model instances. The MAE (right) results of the true linear model instance $y = 3x + 0.5$ are shown for comparison. The vertical dotted line indicates the size of the largest experimental dataset considered in this work.

performance of a random model instance can vary quite strongly. However, it also shows that a significant improvement is obtained if an ensemble average is used.

2. An approximate test set for experimental data and work-flow of repetition computer experiments

In contrast to the synthetic data above, the experimental dataset available is limited to a small and finite number of data points (45 for PSS and 31 for EB). Selecting a small subset (e.g., 5–10 points) to serve as test set will significantly limit (a) the variability of repetition sets and (b) the range of dataset sizes to consider. Furthermore, such a small test set would give rise to a calculated RMSE or MAE, which is very sensitive to the specific data points in the test set (cf. Fig. S.1 in the supplementary material).

In contrast, for a test set of 50 points, the behavior of the RMSE and MAE is expected to be less volatile (cf. the vertical dotted line in Fig. S.1 in the supplementary material).

We, therefore, opted to use the full set of 45 and 31 experimental data points for PSS and EB, respectively, as test set. This approximation has some obvious limitations when it comes to predicting the generalization error and should, therefore, be avoided in practical ML studies. For example, calculated MAE and RMSE values for the larger dataset sizes are lower bounds as they converge to the quality of the training data. However, this approach has some very relevant benefits for our purposes: (1) the obtained RMSE/MAE is much better converged than what could be obtained with merely 5–10 data points, (2) the impact of this approximation is sufficiently small for small dataset sizes ($n < 20$), and (3) it allows for the observation of trends over larger dataset size intervals, making it easier to compare to the synthetic data.

The limited size of our pool of experimental data points also has an impact on the possible variation when performing repetition experiments. We, therefore, expect the repetition experiments on the experimental data to present narrower distributions, than would be the case if truly distinct datasets are used.

The work-flow of the repetition experiments is similar as for the synthetic data. A copy of the entire experimental dataset is set aside as a test set for the reasons we just discussed (this in contrast to the synthetic data, in which case the test set is an independently generated dataset). For each of the 100 repetitions, a copy of the entire experimental dataset is randomly shuffled and datasets of sizes ranging from 10 to 31 (EB) or 45 (PSS) are generated according to the incremental size dataset scheme shown in Fig. 2. This initial shuffling of the data points ensures the independence of the smaller dataset sizes between repetitions. For each dataset size in each repetition, an ensemble of 1000 train-validation splits (80/20) is generated. This gives rise to an ensemble of 1000 trained model instances for each dataset size in each repetition. For each ensemble, the *Best* and *Worst* model instance is selected based on the RMSE of the validation dataset associated with the model instance. In addition, an *Average* model instance is generated—as will be discussed later—for each ensemble. The predictive quality of these three model instances is then estimated by calculating their MAE on the test set. As such, for each dataset size, we obtain 100 values (1 per repetition). The mean of these values is presented, and the 95% CI is estimated using BCa bootstrapping as before.

3. Experimental data

The mean values and the 95% CI for the model coefficients of a linear regression model on the PSS and EB dataset are shown in Figs. 7 and S.5 in the supplementary material, respectively. In contrast to the synthetic data, there is a distinct difference between the model coefficients for the *Worst* model instance and the *Best* and *Average* model instances, the latter presenting very similar coefficient values. This indicates that the model-coefficient distributions are asymmetric, but more importantly, that better quality model instances are clustered around the average of the ensemble of models. Furthermore, note that the CIs are significantly larger in the case of the *Worst* model instance.

Similar to the synthetic data, the MAE of the *Best* and *Average* model instances roughly coincide (cf. Figs. 8 and S.6 in the

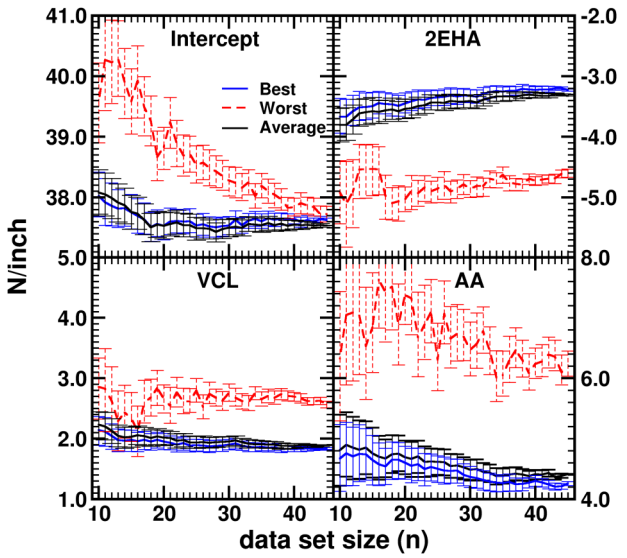

FIG. 7. Mean values and 95% confidence intervals, based on 100 repetitions, of the four fitting coefficients of a linear regression model of the PSS data, using the *Best*, *Worst*, and *Average* model instances.

supplementary material) following a power law behavior (cf. Table S.III in the supplementary material). The MAE of the *Worst* model instance, on the other hand, shows a significant deterioration of the model quality. This highlights the capricious nature of small data sets, while at the same time providing a simple escape route. By creating an ensemble of subsets and constructing an *Average* model instance, the predictive quality can be markedly improved and stabilized (*c.q.* no strong fluctuations upon addition of one or a few extra data points).

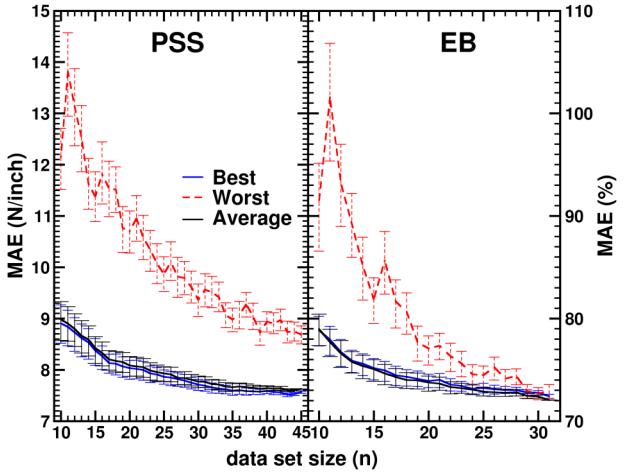

FIG. 8. The estimated MAE values for the *Best*, *Worst*, and *Average* model instances for the PSS and EB dataset.

D. Elastic net regression models

One might question if the behavior presented in Secs. III B–III C is merely an artifact of the simplicity of the used models. In this section, an Elastic Net Regression (ENR) model is considered an example of a more complex regression model.[56,57] An ENR is itself a linear combination of two regularized regression methods, indicated as LASSO or l_1 regression[58,59] and Ridge or l_2 regression.[60] The regularization of the ENR is performed using the cost term,[48]

$$C(\boldsymbol{\theta}) = r\alpha \sum_i |\theta_i| + (1-r)\frac{\alpha}{2}\sum_i \theta_i^2, \tag{3}$$

with θ_i being the descriptor weights and the hyperparameters α and r being the regularization strength and the mixing term, respectively. As a result, if $r = 1$, the ENR becomes a LASSO regression, while for $r = 0$, it becomes a Ridge regression.

Because the value of the two hyperparameters strongly influences the resulting model, it is necessary to tune them.[61]

1. Synthetic data

To investigate the behavior of the ENR, we focus on synthetic data generated by the non-linear model (Sin^n). For a small dataset ($n = 20$), we compare three ENR polynomial models of order 5, 6, and 10, respectively. These models are trained on 1000 random subsets (i.e., 80/20 train-validation splits) of the small dataset. The distribution of the two hyperparameters is presented in Fig. 9

Similar to the model coefficients, the hyperparameters do not give rise to a single solution but rather a distribution of solutions. In the case of the mixing parameter r, a large majority of the ENR model instances (93 and 85%) ends up with (almost) pure LASSO regularization for the fifth and tenth order polynomial. Alternately, for the sixth order polynomial, only 23% of the model instances

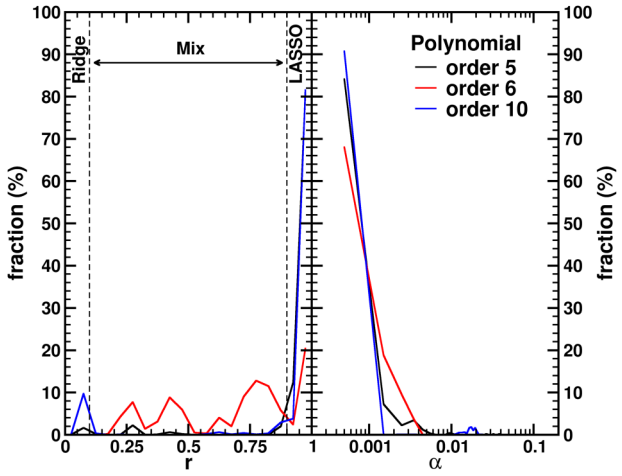

FIG. 9. Distribution of the two hyperparameters of the ENR model for a non-linear synthetic dataset of 20 data points.

have pure LASSO regularization, while the bulk has a regularization in between Ridge and LASSO. This shows that even though the same dataset is being considered, the subsets do not give rise to a universal regularization. This behavior is even more pronounced in the experimental data, as will be shown later (cf. Table S.I in the supplementary material). This highlights the sensitivity of the model parametrization with regard to the actual data used. It shows that rather different model instances are obtained using the same data set but with a different train-validation split of this data.

The regularization strength α presents a distribution with a clear preference for small regularization, even in the case of the tenth order polynomial. This regularization results in a modest reduction in the number of retained polynomial terms (cf. Fig. S.11 in the supplementary material). Within this context, it is interesting to note that for the different train-validation splits, the regularization prunes away a different number of polynomial terms. More importantly, the removed terms may differ between model instances (cf. Fig. S.11 in the supplementary material). In the case of the tenth order polynomial, about 30% of the model instances retain 7 or 8 terms, while 9 terms are retained by slightly less than 20%. Furthermore, the eighth and ninth power terms are removed in 79.1 and 75.0% of the model instances, while the quadratic term is always retained (cf. Fig. S.11 in the supplementary material). In this type of ensemble learning, akin to pasting,[62] this behavior can be used for dimensionality reduction. Terms are removed if a sufficiently high fraction of the coefficients in the model ensemble are (near) zero. Note, however, that the use of a standard scaler may significantly alter the proposed model instance outcome with little or no loss in model quality (cf. Sec. S.2 in the supplementary material).

2. Experimental data

A third order polynomial ENR model is trained on the S^{25} dataset of the experimental PSS and EB data. Because the experimental data has three descriptors, a third order polynomial model gives rise to 19 polynomial terms. With only 25 data points available, there is a serious risk of overfitting. As such, a significant regularization of the model is desired. As before, we create an ensemble of 1000 random train-validation splits, and for each subset, the hyperparameters of the ENR are optimized.

In this case, the mixing hyperparameter seems to be roughly independent of the standard scaler and more interestingly, also the distribution of the MAE's on the training and validation data shows no sensitivity with regard to how the standard scaler is used (cf. Fig. S.7 in the supplementary material). In contrast, using a denser r-grid during hyperparameter tuning seems to benefit convergence toward either LASSO or Ridge regularization (cf. Table S.I in the supplementary material). The distribution of the hyperparameters over the entire ensemble is shown in Fig. S.8 in the supplementary material and presents a strong dependence of the hyperparameters on the specific points included in the dataset. Furthermore, it also shows that hyperparameters are not transferable between different targets for the same descriptor sets.

As we are interested in finding a strongly regularized model, the ensemble of model instances is split in three sets: $r \leq 0.1$ (Ridge), $0.1 < r < 0.9$ (Mix), and $r \geq 0.9$ (LASSO). Highlighting these three sets in the MAE correlation plot in Fig. S.9 in the supplementary material shows that the Ridge and Mix sets present generally a slightly lower MAE on the training data than the LASSO set. Alternately, the LASSO set appears to present slightly higher presence in the lower MAE values for the validation set; however, upon closer investigation the three sets give comparable distributions. This shows that the three sets give rise to qualitatively similar accuracy.

This similar accuracy stands in stark contrast to the model regularization. Where the Ridge model instances retain all polynomial terms, the Lasso model instances reduce this number to 2–3 and 3–11 for PSS and EB, respectively. Comparison of Figs. S.12 and S.13 in the supplementary material, furthermore, shows that the EB model is much harder to regularize than the PSS model.

As expected, the polynomial models provide a significant improvement over the linear models (compare Figs. 8 and 10). More interestingly, the same typical behavior for the MAE of the *Best*, *Worst*, and *Average* models is observed. A broad spread in quality is seen for the model instances making up the ensembles, while the Average model brings forward a model instance competitive with the best available model instances.

Our choice of the ENR model is motivated by the need to regularize the polynomial model. Although some regularization is observed, it is also observed to decrease with increasing dataset size. Furthermore, the *Average* model instance is by construction even less regularized, as the mean of the coefficient ensembles rarely presents a zero value. We, therefore, compare these results to the results of a pure LASSO regression with the regularization strength $\alpha = 1.0$. In contrast to the ENR model instances, all LASSO model instances show a significant regularization. Even more, the regularization is sufficiently strong to lead to a regularized *Average* model instance in the case of the PSS data.

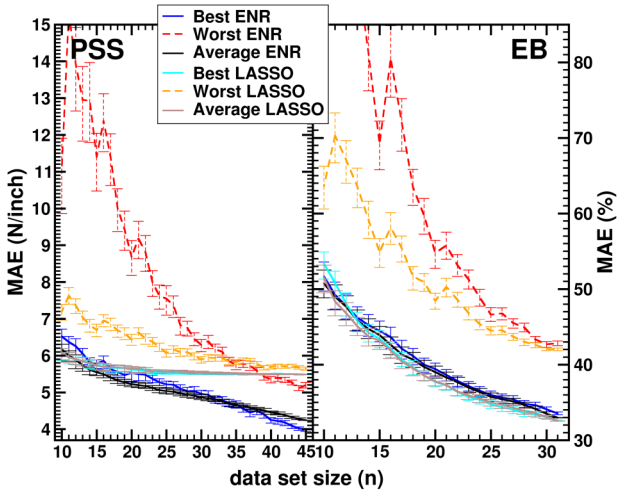

FIG. 10. The estimated MAE values for the *Best*, *Worst*, and *Average* polynomial model instances for the PSS and EB dataset. Elastic Net Regularization (ENR) is compared to LASSO Regularization.

Comparison of the model quality shows the LASSO models to present a much narrower spread, with a notably better quality for the *Worst* model instance. The *Best* and *Average* model instances are of a similar quality as obtained for the ENR model.[63] Furthermore, the evolution of the model coefficients is much more well-behaved for the explicit LASSO model than it is for the ENR model.

IV. DISCUSSION

A. Model variation in small datasets

The impact of the (small) dataset size on the parametrization of the trained regression models clearly cannot be neglected. Even for datasets which are quite large—from the experimental perspective (e.g., 40 data points)—we show that within an ensemble of randomly drawn subsets the coefficients of the model instances show a wide spread (cf. Fig. 5 and S.4 in the supplementary material). It is important to understand that this spread is not due to the model trained—although under- and overfitting may contribute to it—but instead, it originates from the inherent uncertainty of experimental measurements: e.g., due to variation of environmental conditions, slight differences in actual preparation and synthesis. When modeling experimental systems, the goal is to discover the underlying abstract and perfect model, such as the noiseless versions of Eqs. (1) and (2) in the synthetic data examples. However, the uncertainty in the experimental results leads to a variation of the coefficients in a numerical regression when data points are added or if a different set of data points is used (cf. Fig. 1). Furthermore, adding a single data point to a set of ten gives rise to a stronger perturbation than adding it to a data set of 10 000 points. In the latter case, the low relative weight of a single point makes its contribution nearly negligible. In the former case, the high relative weight not only results in the observed distribution of the model coefficients, but it also leads to different distributions for different dataset drawings (*c.q.*, repeating the set of experiments to obtain a new data set). This latter aspect is visualized by the presence of a non-zero spread of the ensemble averages upon 100 repetitions (e.g., Figs. 6 and 7).

When considering an ensemble of subsets, the resulting ensemble of trained model instances shows a wide variation in quality as well. Therefore, if only a single model instance is trained on a single subset of the data, one may get lucky and end up with a well performing model instance, or not. This behavior is typical for small datasets and disappears with increasing dataset size, as the distribution of the quality (i.e., MAE or RMSE's of the ensemble instances) becomes narrower. As a result, one might consider using the entire data set for training and dispense with any validation or test sets. This, however, does not resolve the intrinsic problem. The observation above, with regard to the subsets, is also valid for the full dataset under consideration, as this dataset itself is a subset of any dataset produced by adding extra data points. This creates an interesting conundrum: How lucky is your available dataset?

For example, an 80/20 splitting of a dataset of 20 data points provides 16 points for training model instances. Investigation of the MAE data of 100 repetitions with ensembles of 1000 subsets shows something very intriguing in Figs. 8 and 10. The best case model instance trained on 16 points outperforms the worst case model instance trained on 20 points. This highlights the impact the individual data points have, as the 16 and 20 data point sets share 16 data points.

B. Measuring quality

The ability to associate a level of trustworthiness to different models (instances) is of high importance in model development.[64] The easiest way is via a numeric quality measure. The quality of a model can be measured by testing how well it predicts (preferably) unseen test data. For large datasets, this presents few problems as there is sufficient data to go around. In contrast, small datasets have only few data points and, therefore, even less to spare for quality control. Since quality measures like RMSE and MAE average a penalty over the data points tested, they present large variations if only very few points are used for testing (cf. Fig. S.1 in the supplementary material).

Given this knowledge, direct comparison of the quality of individual model instances becomes tenuous. Even though the difference between the best and worst model instances may be clear cut, defining the *Best* and *Worst* model instance in absolute terms is not.[65]

To sidestep the uncertainty of the validation quality of the model instances, we take a closer look at the *Average* model instance. The coefficients of this model instance are calculated as the mean of the specific coefficient in the ensemble of model instances trained on the subsets. (In the case of linear and polynomial models one can prove this model instance to present identical prediction results as taking the ensemble average but using only a fraction of the computational resources, cf. Sec. S.7 in the supplementary material.)

As can be seen in Figs. 8 and 10, the *Average* model instance shows a very good quality measure, comparable or even better than the best model instances. This behavior can be understood within the context of the Kullbach–Leibler divergence and the model information available in the dataset.[66] As some information in a dataset may be superfluous, any subset may contain up to the same amount of relevant model information. Accordingly, the best model instance may not have access to all model information available in the dataset. In contrast, the *Average* model instance has access to the information available to each of the contributing model instances and as such to at least as much information as the best model instance. Therefore, we expect the *Average* model instance to perform as well or better than the best model instance, as is seen in Figs. 8 and 10.

From the above, it becomes clear that the usual quality measures lose most of their potency due to the use of small datasets. There are, however, some ways to deal with this. As it may have become clear to the more expert reader; our *Average* model instance is equivalent to an ensemble average, and our construction used to investigate the model coefficient distributions strongly resembles ensemble learning, more specifically pasting.[62] The context of ensemble learning comes with its own nomenclature where the most important concept is in-bag and out-of-bag samples, which replace the concept of training and validation samples. Within the area of ensemble learning, the quality of the ensemble model (i.e., our *Average* model instance) is often taken as the mean of the out-of-bag quality measures (i.e., the mean of the

quality measures of all the validation sets of the ensemble). Thus, model averaging may provide a means to obtain the best possible model instance for a certain model, while the mean out-of-bag quality estimate provides a possible metric to compare different (ensemble) models. In addition, there are also other possible metrics, which do not require additional datasets, such as the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) that may be of interest.[67–69] Note that although the AIC quality estimate is only valid in the limit of large sample sizes, corrections can (and should) be made for small datasets.[70,71]

C. The *average* model

A somewhat surprising aspect of the *Average* model instance is the fact that the mean of the coefficients seems to coincide with the coefficient value of the *Best* model instances.

Figure 11 provides an elucidating example of the correlation between a coefficient value (here the intercept) and the RMSE of the model instance on the validation set. The spread of the coefficient value clearly narrows toward the mean coefficient value with increasing model instance quality. This exemplifies how by construction, the coefficients of the *Average* model instance will coincide with those of the best possible model instance, even if this model instance in not present in the generating ensemble. Our examples in Sec. III show this is the case for linear and polynomial regression. Furthermore, it is interesting to note that for an ensemble of M model instances with N fitting coefficients, the *Average* model instance only contains N fitted coefficients. This is significantly less than the equivalent ensemble model.

D. Comparison to other ensemble methods

The presented *Average* model is by construction equivalent to model averaging, i.e., an ensemble learning method. More specifically, it can be considered a special case of what is called "pasting" and related to "bagging."[62,72] In contrast to standard ensemble learning methods, our approach requires it to be possible to construct an average of the base model. This excludes some base models (e.g., decision trees) but in return provides a single powerful model instance for predictive purposes, instead of an ensemble of instances. This makes it computationally very efficient both in terms of memory storage and numerical evaluation, ideally suited as a generator in, for example, a Monte Carlo algorithm. It is, however, interesting to note at this point that the same ensemble benefits and behavior presented in this work are expected for base models that cannot be averaged, but then without the computational efficiency benefit.

A typical example is Random forests, for which the base model is a single decision tree.[73] In this method, an ensemble of decision trees is trained on an ensemble of data subsets, i.e., the decision trees take up the role of our linear or polynomial regressor function. In contrast, however, defining a single averaged model instance (i.e., a decision tree) for a random forest is not as straightforward, and as such ensemble averages have to be calculated upon prediction of new data.

Gaussian processes (GPs) are another ensemble type method in ML which presents similarities to the current work.[74] The difference between the two is located in both the construction of the ensemble and the resulting predictor. Where the current approach is limited to a single model (e.g., linear or third order polynomial), the model instances comprising the ensemble of a GP are drawn from an infinite set of possible models. In addition, the resulting predictor of the current approach only contains the same number of fitting parameters as a single model instance, which is quite different from the case of a GP. Although GP provide a very powerful modeling tool, their interpretability is significantly more complex—as the "model" has become a black-box superposition of infinite possible models. This latter aspect can be partially mitigated by the smart selection of the employed kernel.[75] In contrast, our approach draws all model instances from the same base model, leading to a general similarity between instances which simplifies interpretation. This gives access to an explicit relation between the experimental inputs (descriptors) and goals (targets). For ML models which are harder to interpret one can always turn to explanatory methods such as Shapley values or an implementation of the SHAP method.[76,77] In these, a game-theoretical approach is used to decompose the difference from the mean into contributions from the data features.

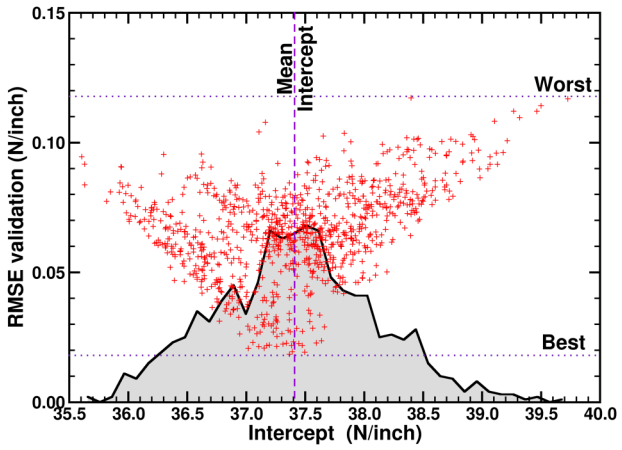

FIG. 11. Correlation plot of the RMSE of the validation set and the intercept value for linear model instances trained on 1000 subsets of a 25 point PSS dataset. The distribution of the correlation data is indicated by the black curve.

V. CONCLUSION

We elucidate the intricacies of using ML within the context of small experimental datasets. Using both synthetic and experimental data, we highlight the trends and illustrate how small datasets influence individual model instances. It is shown that the quality of model instances can vary wildly for the same dataset, highlighting the presence of a significant chance factor. This behavior is presented explicitly for linear and elastic net regularized regression models and is expected to be present for other ML regression models as well. We show that this far-reaching impact of the specific data points in a small dataset can be mitigated through the construction and use of ensemble-averaged model instances. Such

model instances are constructed by averaging the model coefficients over an ensemble of subset trained model instances. The resulting *Average* model instances are shown to be of very good quality as well as consistent over dataset sizes. Furthermore, we show that the model coefficient values converge toward those of the best model instances. Predictions by the *Average* model instance are equivalent to taking an ensemble average but more efficient by construction.

The use of small datasets also restricts quality measures of individual model instances. We discuss why their values should not be considered in terms of absolutes, but rather as indications (i.e., "comparable quality" instead of "quality A better than B").

As a result of our findings, we, therefore, propose the preferential use of ensemble-averaged model instances (or—if the base model does not allow for averaging—ensemble models) for modeling very small datasets. Such model instances reduce the detrimental chance factor and are robust with regard to varying dataset size, while presenting the best quality predictions. Although the results reported in this work are applied to linear and polynomial regression models, we expect them to extend to other regression models as well. Moving beyond the realm of small datasets, these insights may also be beneficial for ML schemes, which partition the training data into small datasets such as active learning approaches, leading to faster convergence and less data hungry ML.

SUPPLEMENTARY MATERIAL

See the supplementary material for the implementation and used hardware, discussion on the impact of the standard scaler, distribution of the model coefficients for the EB model, selected powers by the ENR model, coefficients and quality obtained by the LASSO method, and derivation of the equivalence between the average model and ensemble average.

ACKNOWLEDGMENTS

D.E.V.P. and K.V.B. acknowledge the project D-NL-HIT carried out in the framework of INTERREG-Program Deutschland-Nederland, which is co-financed by the European Union, the MWIDE NRW, the Ministerie van Economische Zaken en Klimaat, and the provinces of Limburg, Gelderland, Noord-Brabant, and Overijssel. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center) and funded by the Research Foundation Flanders (FWO) and the Flemish Government—Department EWI.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]O. K. Farha, A. O. Yazaydin, I. Eryazici, C. D. Malliakas, B. G. Hauser, M. G. Kanatzidis, S. T. Nguyen, R. Q. Snurr, and J. T. Hupp, Nat. Chem. **2**, 944 (2010).
[2]S. Curtarolo, G. L. W. Hart, M. B. Nardelli, N. Mingo, S. Sanvito, and O. Levy, Nat. Mater. **12**, 191 (2013).
[3]N. Nosengo, Nature **533**, 22 (2016).
[4]G. R. Schleder, A. C. M. Padilha, C. M. Acosta, M. Costa, and A. Fazzio, J. Phys. Mater. **2**, 032001 (2019).
[5]D. E. P. Vanpoucke, J. Phys. Condens. Matter **26**, 133001 (2014).
[6]D. E. P. Vanpoucke, S. S. Nicley, J. Raymakers, W. Maes, and K. Haenen, Diam. Relat. Mater. **94**, 233 (2019).
[7]R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules*, International Series of Monographs on Chemistry Vol. 16 (Oxford Science Publications, Oxford, 1989).
[8]K. Lejaeghere, G. Bihlmayer, T. Bjoerkman, P. Blaha, S. Bluegel, V. Blum, D. Caliste, I. E. Castelli, S. J. Clark, A. Dal Corso *et al.*, Science **351**, aad3000 (2016).
[9]E. Ghafari, M. Bandarabadi, H. Costa, and E. Júlio, J. Mater. Civ. Eng. **27**, 04015017 (2015).
[10]Y. Liu, T. Zhao, W. Ju, and S. Shi, J. Mater. **3**, 159 (2017).
[11]K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, Nature **559**, 547 (2018).
[12]M. Rupp, O. A. von Lilienfeld, and K. Burke, J. Chem. Phys. **148**, 241401 (2018).
[13]M. Haghighatlari and J. Hachmann, Curr. Opin. Chem. Eng. **23**, 51 (2019).
[14]Y. Zhang, X. He, Z. Chen, Q. Bai, A. M. Nolan, C. A. Roberts, D. Banerjee, T. Matsunaga, Y. Mo, and C. Ling, Nat. Commun. **10**, 5260 (2019).
[15]G. R. Schleder, A. C. M. Padilha, A. Reily Rocha, G. M. Dalpian, and A. Fazzio, J. Chem. Inf. Model. **60**, 452 (2020).
[16]H. Willems, S. De Cesco, and F. Svensson, J. Med. Chem. (published online 2020).
[17]Y. Goldberg, J. Artif. Intell. Res. **57**, 345 (2016).
[18]J. N. Kutz, J. Fluid. Mech. **814**, 1 (2017).
[19]S. Mehrkanoon, Y. A. Shardt, J. A. Suykens, and S. X. Ding, Eng. Appl. Artif. Intell. **55**, 219 (2016).
[20]S. Chmiela, H. E. Sauceda, K.-R. Müller, and A. Tkatchenko, Nat. Commun. **9**, 3887 (2018).
[21]A. Kamath, R. A. Vargas-Hernández, R. V. Krems, T. Carrington, and S. Manzhos, J. Chem. Phys. **148**, 241702 (2018).
[22]Y. A. W. Shardt, S. Mehrkanoon, K. Zhang, X. Yang, J. Suykens, S. X. Ding, and K. Peng, Can. J. Chem. Eng. **96**, 171 (2018).
[23]K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer, Nat. Comm. **10**, 5024 (2019).
[24]P. Z. Moghadam, S. M. J. Rogge, A. Li, C.-M. Chow, J. Wieme, N. Moharrami, M. Aragones-Anglada, G. Conduit, D. A. Gomez-Gualdron, V. Van Speybroeck *et al.*, Matter **1**, 219 (2019).
[25]W. Yang, T. T. Fidelis, and W.-H. Sun, ACS Omega **5**, 83 (2020).
[26]K. Gubaev, E. V. Podryabinkin, and A. V. Shapeev, J. Chem. Phys. **148**, 241727 (2018).
[27]A. Y.-T. Wang, R. J. Murdock, S. K. Kauwe, A. O. Oliynyk, A. Gurlo, J. Brgoch, K. A. Persson, and T. D. Sparks, Chem. Mater. **32**, 4954–4965 (2020).
[28]J. R. Cendagorta, J. Tolpin, E. Schneider, R. Q. Topper, and M. E. Tuckerman, J. Phys. Chem. B **124**, 3647 (2020).
[29]S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy *et al.*, Comput. Mater. Sci. **58**, 218 (2012).
[30]A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder *et al.*, APL Mater. **1**, 011002 (2013).
[31]S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl, and C. Wolverton, npj Comput. Mater. **1**, 15010 (2015).
[32]N. Carson, Chem. Eur. J. **26**, 3194 (2020).
[33]J. P. Perdew and K. Schmidt, AIP Conf. Proc. **577**, 1 (2001).
[34]D. E. P. Vanpoucke and K. Haenen, Diam. Relat. Mater. **79**, 60 (2017).
[35]C. Houben, N. Peremezhney, A. Zubov, J. Kosek, and A. A. Lapkin, Org. Process Res. Dev. **19**, 1049 (2015).
[36]M. Rubens, J. Van Herck, and T. Junkers, ACS Macro Lett. **8**, 1437 (2019).
[37]A. D. Clayton, A. M. Schweidtmann, G. Clemens, J. A. Manson, C. J. Taylor, C. G. Niño, T. W. Chamberlain, N. Kapur, A. J. Blacker, A. A. Lapkin *et al.*, Chem. Eng. **384**, 123340 (2020).
[38]Y. Zhang and C. Ling, npj Comput. Mater. **4**, 25 (2018).

[39]K. De Grave, J. Ramon, and L. De Raedt, in *11th International Conference on Discovery Science, Budapest, Hungary, October 13–16th, 2008*, Lecture Notes in Computer Science Vol. 5255, edited by J. F. Boulicaut, M. R. Berthold, and T. Horvath (Springer, 2008), pp. 185–196, ISBN isbn978-3-540-88410-1.
[40]D. A. Cohn, Z. Ghahramani, and M. I. Jordan, J. Artif. Intell. Res. **4**, 129 (1996).
[41]N. Peremezhney, E. Hines, A. Lapkin, and C. Connaughton, Eng. Optim. **46**, 1593 (2014).
[42]A. M. Schweidtmann, A. D. Clayton, N. Holmes, E. Bradford, R. A. Bourne, and A. A. Lapkin, Chem. Eng. **352**, 277 (2018).
[43]M. Rubens, J. H. Vrijsen, J. Laun, and T. Junkers, Angew. Chem. Int. Ed. **58**, 3183 (2019).
[44]C. W. Coley, D. A. Thomas, J. A. M. Lummiss, J. N. Jaworski, C. P. Breen, V. Schultz, T. Hart, J. S. Fishman, L. Rogers, H. Gao *et al.*, Science **365**, 6453 (2019).
[45]Z. Wang, Y. Su, W. Shen, S. Jin, J. H. Clark, J. Ren, and X. Zhang, Green Chem. **21**, 4555 (2019).
[46]A. Menon, C. Gupta, K. M. Perkins, B. L. DeCost, N. Budwal, R. T. Rios, K. Zhang, B. Póczos, and N. R. Washburn, Mol. Syst. Des. Eng. **2**, 263 (2017).
[47]A. Menon, J. A. Thompson-Colón, and N. R. Washburn, Frontiers Mater. **6**, 87 (2019).
[48]A. Géron, *Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (O'Reilly Media, Sebastopol, CA, 2017), ISBN isbn978-1491962299.
[49]We are assuming non-pathological datasets and data splittings. Furthermore, if the data requires it, we also assume stratification of the data is considered during the splitting of the dataset.
[50]Within the context of this work, we assume that a (possible) test set is split of initially. This test set can be used to compare the quality of different models. The remaining dataset is used to create an ensemble of train-validation splits. For simplicities sake, we refer to this combination of the validation and training set as the full set.
[51]K. Hermans, B. Kranz, and O. van Knippenberg, "Pressure sensitive adhesive tape," patent WO2018/002055A1 (1 April 2018).
[52]For practical purposes, a polynomial regression is equivalent to a multiple linear regression where each of the linear descriptors is one of the polynomial terms.
[53]For $n = 10$, only 45 distinct 80/20 splits are possible, while for $n = 20$, this number has already grown to 4845. In this work, we have used 1000 (random) splits regradless of the dataset size. Since the splits are drawn randomly from the collection of all possible splits, this means that for a dataset of size $n = 10$, the ensemble of 1000 splits contains multiple copies of the same split realization. Each split appears on average 22 times. For such cases, a computationally more efficient approach is to limit the ensemble of splits to the exhaustive sampling of all splits (which is to be implemented in our framework in the future). However, no significant differences are expected between this and the current implementation.
[54]B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Monographs on Statistics and Applied Probability Vol. 57 (Chapman & Hall/CRC, Boca Raton, FL, 1993).
[55]B. Efron and T. Hastie, *Computer Age Statistical Inference: Algorithms, Evidence, and Data Science*, 1st ed (Cambridge University Press, 2016).
[56]J. Friedman, T. Hastie, and R. Tibshirani, J. Stat. Softw. **33**, 1 (2010).
[57]S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, IEEE J. Sel. Top. Signal. Process. **1**, 606 (2007).
[58]F. Santosa and W. W. Symes, SIAM J. Sci. Comput. **7**, 1307 (1986).
[59]R. Tibshirani, J. Royal Stat. Soc. B **58**, 267 (1996).
[60]A. E. Hoerl and R. W. Kennard, Technometrics **12**, 55 (1970).
[61]As we are using the scikit learn framework, the hyperparameter tuning is performed using the ElasticNetCV model using at least 100α's.
[62]L. Breiman, Mach. Learn. **36**, 85 (1999).
[63]The better quality observed for the polynomial ENR model in case of the larger PSS datasets may be due to an overfitting by the ENR model in those instances.
[64]S. Wenmackers and D. E. P. Vanpoucke, Stat. Neerl. **66**, 339 (2012).
[65]In this context, the *Best* and *Worst* model instances in the figures should not be considered as absolute, but rather as representative.
[66]S. Kullback and R. A. Leibler, Ann. Math. Statist. **22**, 79 (1951).
[67]H. Akaike, IEEE Trans. Automat. Contr. **19**, 716 (1974).
[68]G. Schwarz, Ann. Statist. **6**, 461 (1978).
[69]E. Wit, E. v. d. Heuvel, and J.-W. Romeijn, Stat. Neer. **66**, 217 (2012).
[70]J. E. Cavanaugh, Stat. Probab. Lett. **33**, 201 (1997).
[71]S. Konishi and G. Kitagawa, *Information Criteria and Statistical Modeling*, 1st ed. (Springer Publishing Company, Incorporated, 2007), ISBN isbn0387718869.
[72]L. Breiman, Mach. Learn. **24**, 123 (1996).
[73]T. K. Ho, IEEE Trans. Pattern Anal. **20**, 832 (1998).
[74]C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)* (The MIT Press, 2005), ISBN isbn026218253X.
[75]D. K. Duvenaud, Ph.D. thesis, University of Cambridge, 2014.
[76]L. S. Shapley, *Notes on the n-Person Game—II: The Value of an n-Person Game* (RAND Corporation, Sante Monica, CA, 1951).
[77]S. M. Lundberg and S.-I. Lee, in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Advances in Neural Information Processing Systems Vol. 30, edited by Guyon, I and Luxburg, UV and Bengio, S and Wallach, H and Fergus, R and Vishwanathan, S and Garnett, R (2017), , ISSN issn1049-5258.