ACS OMEGA

http://pubs.acs.org/journal/acsodf

Article

Uncertainty Prediction for Machine Learning Models of Material Properties

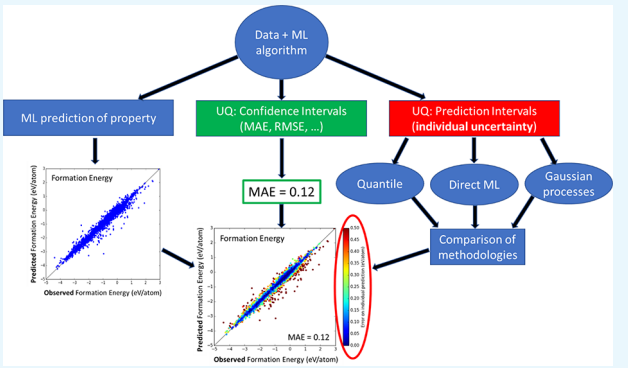
Francesca Tavazza, Brian DeCost, and Kamal Choudhary*

Cite This: *ACS Omega* 2021, 6, 32431−32440

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Uncertainty quantification in artificial intelligence (AI)-based predictions of material properties is of immense importance for the success and reliability of AI applications in materials science. While confidence intervals are commonly reported for machine learning (ML) models, prediction intervals, i.e., the evaluation of the uncertainty on each prediction, are not as frequently available. In this work, we compare three different approaches to obtain such individual uncertainty, testing them on 12 ML-physical properties. Specifically, we investigated using the quantile loss function, machine learning the prediction intervals directly, and using Gaussian processes. We identify each approach's advantages and disadvantages and end up slightly favoring the modeling of the individual uncertainties directly, as it is the easiest to fit and, in most of the cases, minimizes over- and underestimation of the predicted errors. All data for training and testing were taken from the publicly available JARVIS-DFT database, and the codes developed for computing the prediction intervals are available through the JARVIS-tools package.



1. INTRODUCTION

Artificial intelligence (AI) approaches for materials science have been explored for decades,[1−9] but only in the last few years, they have become consistently successful across a wide variety of materials science tasks. While materials science is still data-poor compared to many other application areas of AI modeling, modern instruments and powerful computational facilities are finally able to produce enough data for successful application of machine learning (ML) in this area. The Materials Genome Initiative (MGI) has also contributed to this flourishing, by supporting the creation of large, publicly accessible databases, with very consistently computed data. With respect to atomistic computational results, regression models were first successfully applied to the evaluation of energetics-related properties, like formation energy,[10−12] and then expanded to predict elastic,[13−15] electronic,[5,16,17] optical,[18] and many other physical properties. Computational parameters needed in simulations, not just physical quantities, can be predicted using regression models as well such as for *k*-points, as discussed in ref 19. JARVIS-ML[12] is an example of a publicly available suite of ML models for a variety of physical properties. Specifically, all its models apply to the prediction of properties for ideal, crystalline materials and have been trained on the density functional theory (DFT) data in the JARVIS-DFT database.[20,21]

ML methods are intrinsically statistical in nature, as the true underlying model form is not available to the ML model. However, quite surprisingly, during this period of explosion of AI applied to materials science, relatively little effort has been spent in evaluating its uncertainty, beyond assessing the average ability of the model. While there is a fair amount of discussion on how to evaluate uncertainty in ML/materials research,[22−24] on how to estimate prediction intervals,[25−30] and on how to collect data to improve ML models,[31] the uncertainty on the individual prediction is often not rigorously evaluated or reported when machine learning material properties. Uncertainty quantification (UQ) is an extremely important step in any experiment or computational assessments, as it determines how trustworthy the measured or computed data are. The same is true for the predictions of ML. Currently, the majority of ML/materials papers exclusively provide uncertainty evaluation on the average ability of ML models, providing quantities like the mean average error (MAE), the mean square error (MSE), or the root-mean-square error (RMSE). This approach fails to address the uncertainty/confidence of predictions for individual instances. For instance, when applying an ML model for formation energy to an unknown compound, we want to know if that

Received: July 14, 2021
Accepted: November 5, 2021
Published: November 23, 2021

specific material will be stable, and an evaluation of the average performance of the model over the entire dataset is not necessarily significant. Moreover, because the MSE-type stats make assumptions about the future data that we know we cannot satisfy (independent, identically distributed data) in materials discovery and design settings, only providing these quantities as quality evaluators may be deceiving about the real capability of the ML model. The reason behind ML-UQ focusing mostly on population variables is that determining prediction intervals, i.e., the uncertainty on each specific prediction, can be computationally fairly expensive, and usually, it requires an additional computational effort beyond the training of the model. Gaussian processes[32] are an exception to this rule, as the individual uncertainty is automatically determined when the model is fit. However, they have other limitations that often prevent them from being the approach of choice, as we will briefly discuss later in this work. Obviously, no amount of UQ can make up/reveal bad quality in the training data as systematic errors in the generation of the dataset will propagate through a machine learning model. For example, standard DFT is known to underestimate electronic bandgaps, so any ML model trained on such data will also predict underestimated bandgaps.

In this work, we estimate prediction intervals using three straightforward and easy-to-implement approaches, one that allows to choose which population interval to be covered by the UQ (quantile approach), the second, based on directly learning the error itself and applicable to any ML algorithm, and the third that automatically provides the uncertainty of each prediction when fitting the model (Gaussian processes). Each method has its advantages and disadvantages, and they are the focus of the Results/Discussion section. These methods are applied to energetic, mechanical, electronic, and optical properties, and their results across properties are compared to each other. All the ML models in this work are trained and validated on the same DFT data so that they can be compared in a completely consistent way. Specifically, for both training and validation, we use the DFT data contained in the JARVIS-DFT database.[20,21]

2. METHODS

All ML models discussed in this paper are regression models as implemented in LightGBM[33] in the case of gradient boosting decision tree (GBDT), or in scikit-learn,[34] for Gaussian processes (GP) and random forest (RF). We use JARVIS-ML-based classical force-field inspired descriptors (CFID)[12] when using GBDT, gradient boosting and random forest, and a subset of such descriptors when using GP. The CFID descriptors provide a complete set of structural and chemical features (1557 for each material), and CFID-based models have been successfully used to develop more than 25 highly accurate ML property prediction models.[12]

All the ML models in this work are trained and validated on the DFT data contained in the JARVIS-DFT database[20,21] (04.26.2020 version, https://doi.org/10.6084/m9.figshare.6815699). For each property, this allows for a consistent comparison between methodologies. However, the dataset size depended on the property under examination, as computing some properties is much more involved than computing others. The largest datasets were available for formation energy (35,885 materials), energy above the convex hull (34,880 materials), and bandgaps computed using the OPTB88vdW functional (OPT) (35,932 materials). The smallest dataset is for exfoliation energies, counting only 806 data. Table S3 gives the dataset size for each property that was examined in this work. In addition to the size of the dataset, its distribution is also an important characteristic and is shown for some key properties in Figure S1.

To determine prediction intervals, the first methodology employed in this work estimates the error through the evaluation of an upper and a lower prediction bound for the quantity under examination. In the rest of the paper, we will call this approach "quantile" as it is based on using the quantile loss function. Machine learning models are often fit through the minimization of a loss function[22−24] as such a function quantifies the errors that the model makes on the training data. There are many options for such loss functions, depending on factors like the distribution of the data, the choice of the machine learning algorithm, the quality of the data, etc. Some loss functions penalize low and high errors equally, and others do not. The least-square loss function (least square, aka MSE) is a commonly used regression loss among those that penalize low and high errors equally, meaning that a positive and a negative error of the same amount incur the same penalty. Obviously, errors of different amounts are penalized differently. Specifically, when using MSE,

$$\mathrm{MSE} = \frac{1}{N} \times \sum_{i=1}^{N} (y_i - y_i^{\mathrm{P}})^2 \tag{1}$$

optimal parameters are determined by minimizing the sum of squared residuals. Here, y_i^{P} are the values predicted for the target values y_i. In other words, MSE optimizes for the mean of the distribution.

Mean average error (MAE) is another example of loss function that penalizes low and high errors equally. MAE is less sensitive to outliers than MSE, so depending on the problem, it may be a better choice. Other loss functions penalize low and high errors unequally, therefore allowing optimization by percentiles. The quantile loss function (quantile) is part of this latter group, and what percentile it optimizes for depends on the choice of the quantile parameter α[22]

$$\text{quantile loss} = \begin{cases} \alpha \times |y_i - y_i^{\mathrm{P}}| \text{ if } (y_i - y_i^{\mathrm{P}}) \geq 0 \\ (\alpha - 1) \times |y_i - y_i^{\mathrm{P}}| \\ \text{if } (y_i - y_i^{\mathrm{P}}) < 0 \end{cases} \tag{2}$$

where α is between 0 and 1. Such a property is the base of the first method that we use to estimate the prediction intervals, therefore the name "quantile" for this approach. For each physical quantity that we are interested in, we will independently optimize three ML models, one for α_{upper}, one for α_{lower}, and one for α_{mid}, where $\alpha_{\mathrm{mid}} = 0.5$ (i.e., the median). In the rest of the paper, we will refer to these models as UPPER, LOWER, and MID, respectively. The prediction interval for each predicted data is given by

$$\mathrm{PredInt}(y_i^{\mathrm{P}}) = |y_i^{\mathrm{upper}} - y_i^{\mathrm{lower}}| \tag{3}$$

where y_i^{upper} (y_i^{lower}) are the values predicted for y_i using α_{upper} (α_{lower}) and y_i^{P} is the value predicted for y_i using α_{mid}. This will be referred to as the "predicted value" in the rest of the paper.

https://doi.org/10.1021/acsomega.1c03752
ACS Omega 2021, 6, 32431−32440

An advantage of this method is that the population interval to be covered is arbitrary. Most of the results discussed in this work have been obtained using $\alpha_{upper} = 0.84$ and $\alpha_{lower} = 0.14$ to construct the upper and lower bounds so that our prediction interval would ideally cover 68% of the population, corresponding to one standard deviation under the assumption of Gaussian error distribution. This makes for an easy comparison with the other methodologies examined in this paper. It is important to note, however, that the coverage of 68% of the population (in the case of $\alpha_{upper} = 0.84$ and $\alpha_{lower} = 0.14$) would only occur for perfect fits and that, in general, the direct count of how often the reference value falls within the UPPER−LOWER interval gives a better idea of the meaning of the uncertainty. We will refer to this quantity as "inbounds" in the rest of the paper.

Algorithm-wise, we utilize the gradient boosting decision tree (GBDT) regression for the quantile approach because it allows optimization of an arbitrary differentiable loss function so that the MID model could be run using either quantile with $\alpha_{mid} = 0.5$ (i.e., optimizing for the median) or MSE (i.e., optimizing for the mean). Another advantage of GBDT is that it provides some degree of interpretability through feature importance ranking, a property that we took advantage of when choosing descriptors for Gaussian processes. As a standard practice, we use a 90 to 10% train−test split.[12,15,17] The 10% independent test set is never used in the hyperparameter optimization or model training so that when the model is evaluated on it, we obtain unbiased performance metrics.

The second approach that we use to determine prediction intervals uses a machine learning model to predict the errors directly. This means that it requires the fitting of two ML models for each physical quantity under examination: one to predict the actual value of the property ("base" model) and one to determine the prediction intervals ("error" model). Consequently, training the models requires splitting the data into three groups: one to fit the base model, one to fit the error model, which can also be used to validate the base model, and the last one to validate the error model. Obviously, the third set can be used to validate the base model as well. Requiring splitting of the data into three sets may be a substantial disadvantage when the amount of available data is small to begin with. We chose to work with a 45−45−10 split so that the number of samples in the third set of data matches the number of data used to validate the other two methodologies. This makes the comparisons between approaches more consistent. In the rest of the paper, this approach will be referred to as "3split" because of its requirement of splitting the available data into three groups.

To model the error using ML, we explored two approaches: one where for each data point, we predicted the absolute value of the difference between exact and predicted values (Error_AD), and the other where we predicted the square of such a difference (Error_SD) instead

$$\text{Error\_AD}_i = |y_i^{\text{Observed}} - y_i^{\text{Predicted}}| \quad (4)$$

$$\text{Error\_SD}_i = (y_i^{\text{Observed}} - y_i^{\text{Predicted}})^2 \quad (5)$$

These two options were inspired by the comparison between mean absolute error (MAE) and mean square error (MSE) loss functions (L1 versus L2): using the square of the differences is easier to optimize, but using the absolute value of such differences is more robust to outliers. They will be indicated as "3split-L1" (eq 4) and "3split-L2" (eq 5) in the rest of the paper.

An important advantage of this second approach is that it does not require a specific loss function, which means that it can be used with any regression model. We used the GBDT regressor, for a direct comparison to the quantile approach, and the random forest regressor, as our preliminary tests showed it to be very effective even without hyperparameter optimization. As for the quantile approach, we computed the inbound percentage for each property model, to establish the meaning of the determined uncertainty.

The third approach that we investigated uses Gaussian processes so that individual uncertainties are automatically determined as the model is trained.[32] Fitting a Gaussian process (GP) starts with the choice of the covariance function, also called kernel, so that the prior distribution is defined. A prior is an infinite-dimensional multivariate Gaussian distribution completely specified by a mean function, $m(x)$, and a covariance function, $k(x, x')$

$$f(x) \sim \text{GP}(m(x), k(x, x')) \quad (6)$$

In other words, the prior does not depend on the training data, but it is a way to incorporate prior knowledge about the functions. The next step is determining the posterior distribution (predictive), which is obtained by limiting the functions going through the training data. Because we have a Gaussian process prior, the posterior is a Gaussian process as well: it is completely described by mean and covariance and automatically provides the UQ estimate. The training of a GP model involves the model selection (i.e., the choice of the kernel function) and the optimization of the kernel hyperparameters. In this work, all fitting parameters (λ_i for $i = 1,..N$, l, l', l'', p, α, noise, and c_i, where $i = 1$, 2, and 3) are optimized using the Broyden−Fletcher−Goldfarb−Shanno[35] (L-BGFS-B) algorithm, as implemented in scipy.optimize.minimize.

The main advantage of using GP is the fact that the uncertainty on each prediction is determined together with the prediction value itself, without requiring any extra effort. Gaussian processes also are extremely effective with a smaller set of data and allow easy incorporation of data into the model. However, GP have disadvantages as well, especially the fact that they lose efficiency in high-dimensional spaces, i.e., if the number of features exceeds a few dozens or for large datasets. They are computationally memory-intensive for large datasets as well.

Lastly, like in the cases of quantile and "3split", the meaning of the error on the individual prediction comes from knowing how often the observed value (the DFT data in this work) falls into the prediction interval. Because the posterior distribution of a Gaussian process is a Gaussian itself, the meaning of the uncertainty on each data should be straightforward. However, as in the case of quantile, the fitting is never perfect, so it is important to compute its "inbound" count on the validation set to establish its meaning for each fitted GP model.

3. RESULTS

All results discussed in this session refer to test data. In all cases, we performed a 90−10 split, so the test data were 10% of the whole dataset.

3.1. Prediction Intervals: Quantile Approach. When analyzing prediction intervals obtained using the quantile approach and eq 3, the first characteristic that appears evident

https://doi.org/10.1021/acsomega.1c03752
ACS Omega 2021, 6, 32431−32440

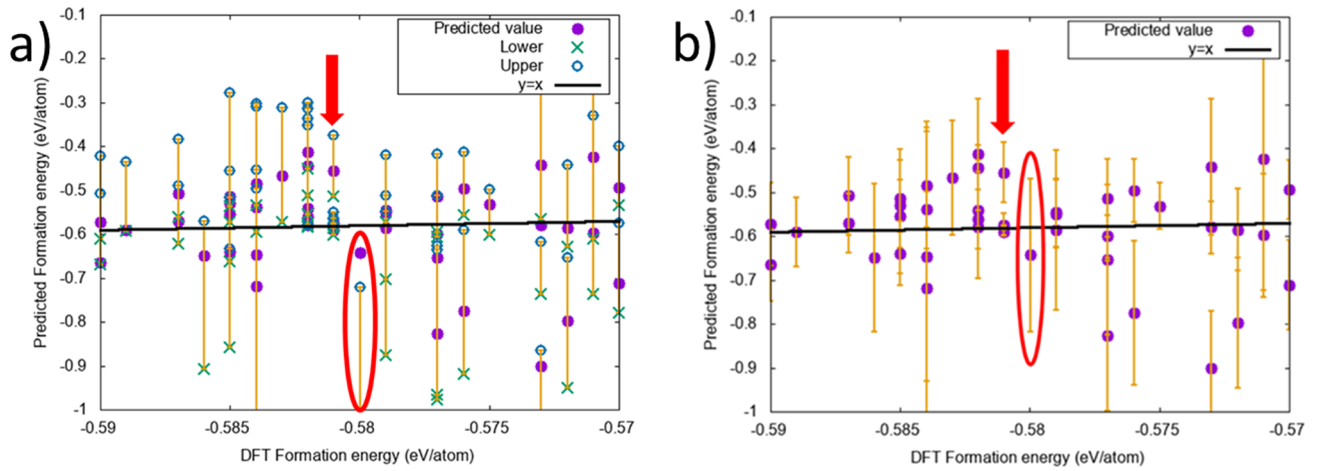


Figure 1. (a) Quantile prediction intervals for formation energy (enlargement). The upper and lower predictions determine the size of the prediction interval, while the "mid" prediction determines the predicted value y^P. The $y = x$ line shows where the predicted value should be to match the DFT data. The arrow points out a case where the predicted intervals are not large enough to include the observed (DFT) data. The elliptical enclosure shows an example of a predicted value outside the prediction interval. (b) Formation energy data (enlargement) with error bars given by $\text{err}_{\text{quantile}} = \text{PredInt}(y_i^P)/2$. The $y = x$ line indicates where the predicted values should be to match the DFT data.

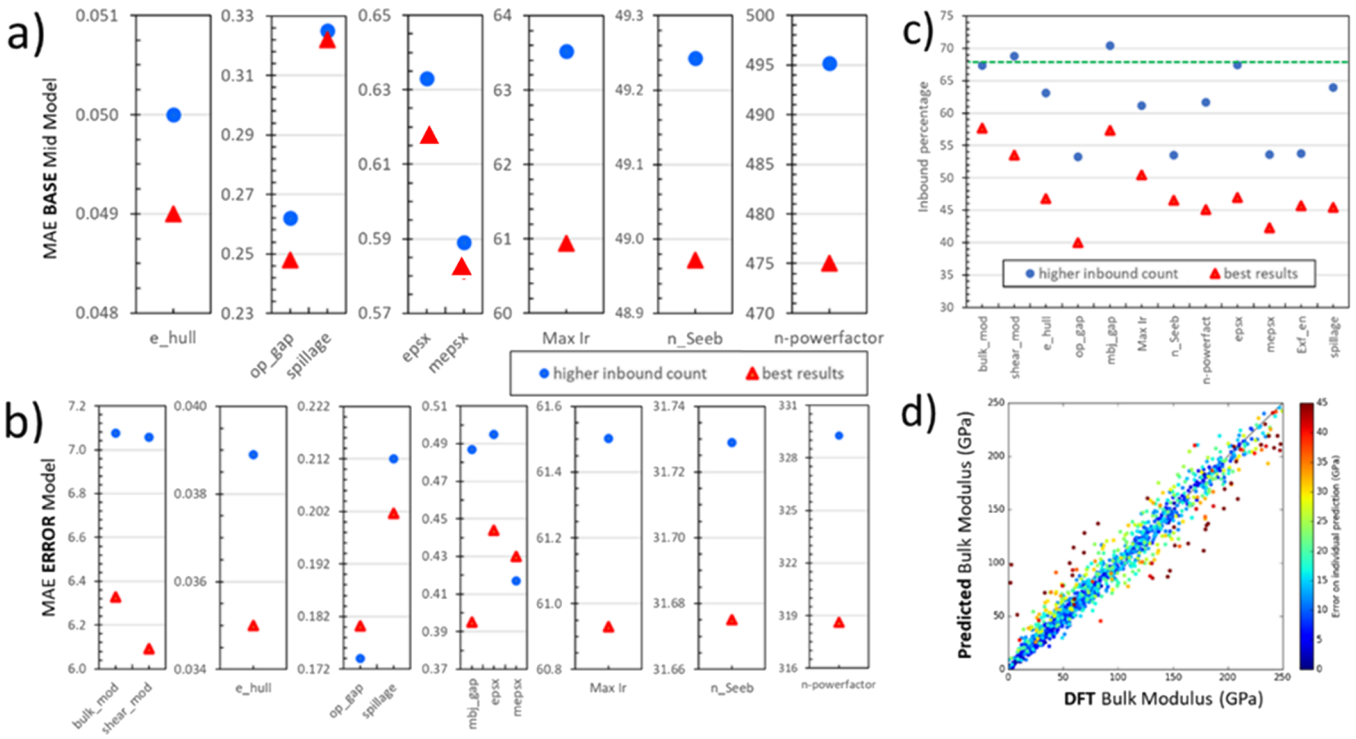


Figure 2. Comparison between the lowest MAE models ("best results") and models with an inbound count closer to 68% ("higher inbound count"), using quantile. MAEs for base mid models are shown in (a), corresponding MAEs for error models in (b), and inbound counts in (c). An example of the prediction distribution is given in (d), for the bulk modulus in the "best results" case. The color of each point represents the size of its prediction interval. The dashed line in (c) indicates the 68% inbound counts that would give easy interpretability to the error bars. Each MAE is in the units of the corresponding property (bulk_mod and shear_mod = bulk and shear modulus, respectively (GPa); e_hull = energy above the convex hull (eV); op_gap and mbj_gap = bandgaps using OPTB88vdW and MBJ data, respectively (eV); spillage, no units; epsx (mepsx) = refractive index along x using OPTB88vdW and MBJ data, respectively (no units); Max_ir = maximum infrared frequency (cm^{-1}); n_Seeb = Seebeck coefficient for electrons (mV/K)).

is their asymmetry with respect to the predicted value y_i^P, as shown in Figure 1a for a subset of formation energy data. However, for plotting purposes and to be able to directly compare quantile-given errors to prediction intervals obtained using the other methodologies, in the rest of the paper, we will define a quantile error as

$$\text{err}_{\text{quantile}} = \text{PredInt}(y_i^P)/2 \quad (7)$$

where $\text{PredInt}(y_i^P)$ is given by eq 3, and we will investigate $y_i^P \pm \text{err}_{\text{quantile}}$.

This approach is exemplified in Figure 1b, where the same predicted data as in Figure 1a are displayed, and the sizes of

https://doi.org/10.1021/acsomega.1c03752
ACS Omega 2021, 6, 32431−32440

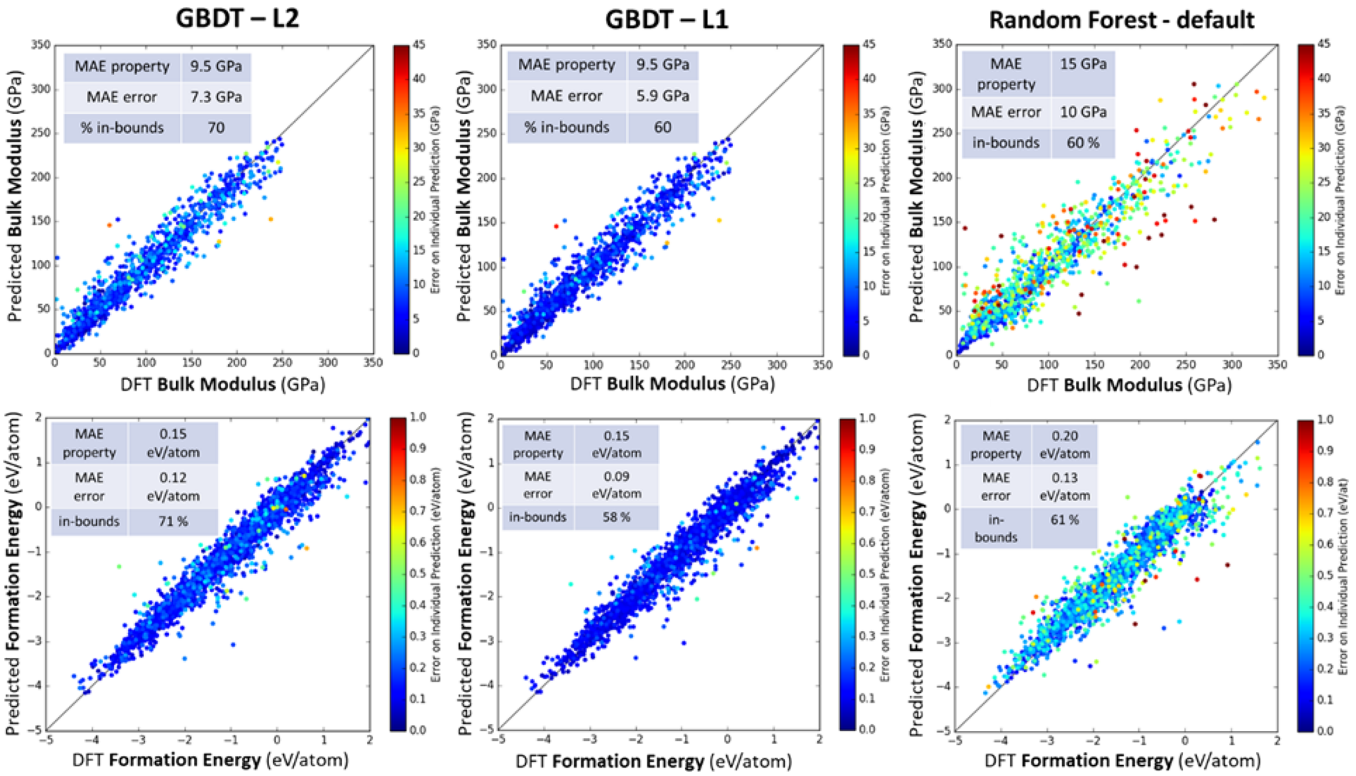


Figure 3. Property and error predictions using different algorithms: gradient boosting decision tree (GBDT) with optimized hyperparameters and errors defined either as the square or as the absolute value of the difference between observed and predicted values (L2 or L1, respectively) and random forest with default parameters. The error on each data point is indicated by its color. Results are given for the bulk modulus (top row) and formation energy (bottom row). For each case, MAEs for both the property and error are reported, as well as their inbound counts.

their predicted intervals are unchanged as well, but now, the error bar is given by $\mathrm{err}_{\mathrm{quantile}}$.

Figure 1a also showcases possible problems that come with using this approach: the arrow points out a case where the predicted intervals are not large enough to include the observed (DFT) data, while the elliptical enclosure shows an example of a predicted value outside the prediction interval. In general, y_i^{P} may not fall between y_i^{upper} and y_i^{lower} because all three values are predicted independently, using models with different optimized hyperparameters although identical training and validation sets. The same "problem" points are pointed out in Figure 1b, but now, because of the symmetric definition of the error, the second type of problem is eliminated by construction (the ellipse case, in the figure). The possibility of having the observed value (DFT data, indicated by the $y = x$ line in Figure 1) outside the prediction interval cannot be eliminated, and therefore, it is necessary to compute the inbound percentage for each model, using the validation set.

Another point worth making, when discussing how to use the quantile approach for uncertainty evaluation, is that the predicted value y_i^{P} can be determined using the least-square loss function (MSE), instead of the quantile loss with $\alpha_{\mathrm{mid}} = 0.5$. The use of MSE corresponds to driving the predictions toward the mean of the distribution instead of its median, and it is the approach followed in this work.

We are now ready to examine results obtained implementing the quantile approach. In Figure 2, we compare two sets of models for each physical quantity of interest. The first set was obtained independently optimizing the hyperparameters of LOWER, MID, and UPPER, to achieve the lowest possible MAE in each case, and it is indicated as the "best results" in the figure. Figure 2c shows that the inbound percentages for the "best results" are always lower than the wished-for 68% (dashed line) and as low as 40% in some cases. By choosing UPPER and LOWER models with slightly larger MAEs, it is possible to increase the inbound counts. The new results are labeled "higher inbound count", and while 68% is not always obtained, the inbound counts are closer to it than before. MAEs for base mid models are displayed in (a), but only for those quantities for which the property prediction changed between the two cases. Error MAEs for all quantities are reported in (b).

3.2. Prediction Intervals: ML Modeling of the Error Directly. As the main advantage of modeling the error directly is that any ML algorithm can be used, in Figure 3, we compare optimized gradient boosting decision tree (GBDT) results to random forest (RF) ones, where RF was run using default parameters only. The reason for such a comparison is the fact that fitting ML models and, especially, optimizing their hyperparameters are a computationally expensive endeavor. Adding UQ on individual predictions necessarily increases such computational cost, even significantly if ad hoc hyperparameter optimization is needed. A possible work-around is to avoid using hyperparameter fitting, at least for the UQ part. Such an approach is feasible only if an algorithm is available, which produces acceptable MAE using default hyperparameters. Our investigation identified random forest as such an algorithm, with default parameters as in its scikit-learn implementation. While default-RF results are clearly worse than those from optimized GBDT, Figure 3 shows that their inbound percentage is like the one from GBDT 3split-L1 and that the error modeling is at least mostly qualitatively correct:

https://doi.org/10.1021/acsomega.1c03752
ACS Omega 2021, 6, 32431−32440

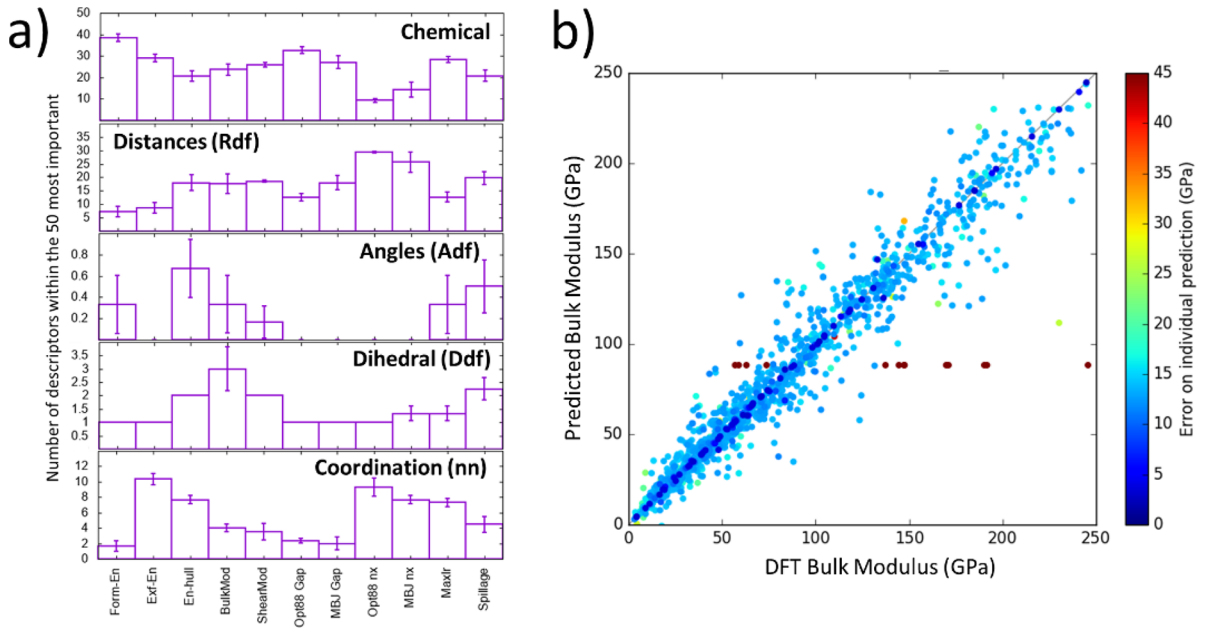


Figure 4. (a) Property-dependent descriptor importance analysis: for each property, the 50 most important descriptors are classified into their type (chemical, distance-related (Rdf = radial distribution function), angle-related (Adf = angular distribution function), dihedral angle-related (Ddf = dihedral distribution function), and coordination-related (number of neighbors)). (b) Prediction versus observed (DFT) data for the bulk modulus. The color gives the predicted error on each individual prediction, using the same scale as in Figures 2 and 3, to facilitate comparisons.

blue points are on or near the diagonal, and red and brown points are away from it. However, the presence of more light blue or green points on or very near the diagonal in RF results than in GBDT ones tells us that RF has more cases of overestimation than GBDT.

Figure 3 also provides a visual comparison between 3split-L1 and 3split-L2 results: in both the bulk modulus and the formation energy cases, 3split-L1 has a lower error MAE but also a lower inbound percentage than 3split-L2. Also, less light blue-colored points are seen in its plots, especially over or near the $y = x$ line that indicates perfect fit, possibly indicating not as much overestimation. Conversely, more dark-blue points are seen away from the diagonal for 3split-L1 than for 3split-L2, indicating occasional underestimation of the errors. Finding lower error MAEs and lower inbounds for 3split-L1 than for 3split-L2 is a trend common to all physical quantities that we examined. More on how 3split-L1 and 3split-L2 models compare to each other and to quantile and Gaussian process results will be covered in the Discussion section (Section 3.4). Lastly, it is well-known that prediction intervals are always wider than confidence intervals,[4] so it is no major concern that the errors are in many cases larger than the corresponding MAEs.

3.3. Prediction Intervals: Gaussian Processes. In this work, we limited the investigation of Gaussian processes (GP) to six quantities: the formation energy, exfoliation energy, bulk modulus, bandgap obtained using the Tran−Blaha-modified Becke−Johnson approach (MBJ bandgap), maximum infrared mode, and spillage. Such quantities were chosen to cover a variety of different physical properties, spanning from energetics to elastic, electronic, optical, and topological while only probing some of the smaller datasets available to us, as the GP becomes inefficient with larger sets. A major exception to this criterion is the inclusion of formation energy, as it comes with one of our largest datasets. It was important to add it, though, because formation energy is a quantity systematically investigated in ML works focused on nanoscale materials science, and therefore, analyzing its single prediction UQ is of widespread interest.

The choice of the kernel is crucial to successfully fit GP. The specific kernel used in this work depends on the property of interest, but they all were a combination of two or more of the following kernel functions[36]

$$K(x, x') = c_1 \times \mathrm{RBF}(\lambda_i, i = 1, \ldots, N) + c_2 \times \mathrm{RQ}(\alpha, l) + c_3 \times \mathrm{RBF}(l') \times \mathrm{ExpSineSquared}(l'', p) + c_4 \times \mathrm{WhiteNoise}(\mathrm{noise})$$

where

- RBF (radial-basis function kernel) is of the form $\exp\left(-\frac{d(x,x')}{2\lambda^2}\right)$, where $d(x,x')$ is the Euclidean distance between x and x′ and λ is a vector of length-scale parameters with as many dimensions as the number of descriptors (N).
- RQ (rational quadratic kernel) is of the form $\left(1 + \frac{d(x,x')^2}{2\alpha l^2}\right)^{-\alpha}$, and it is parameterized by a length-scale parameter l and a scale mixture parameter α.
- The ExpSineSquared kernel is of the form $\exp\left(-\frac{2\sin^2(\pi d(x,x')/p)}{l''^2}\right)$, where l'' is the length-scale parameter and p is a periodicity parameter.
- The WhiteNoise kernel is equal to the noise parameter "noise" for $x = x'$, and it is null otherwise.
- c_1, c_2, c_3, and c_4 are fitting constants and can be set to zero.

Through the RBF term, we include the long-term, smooth rising behavior, while the smaller, medium-term irregularities are made possible through the RQ component, in the case that

https://doi.org/10.1021/acsomega.1c03752
ACS Omega 2021, 6, 32431−32440

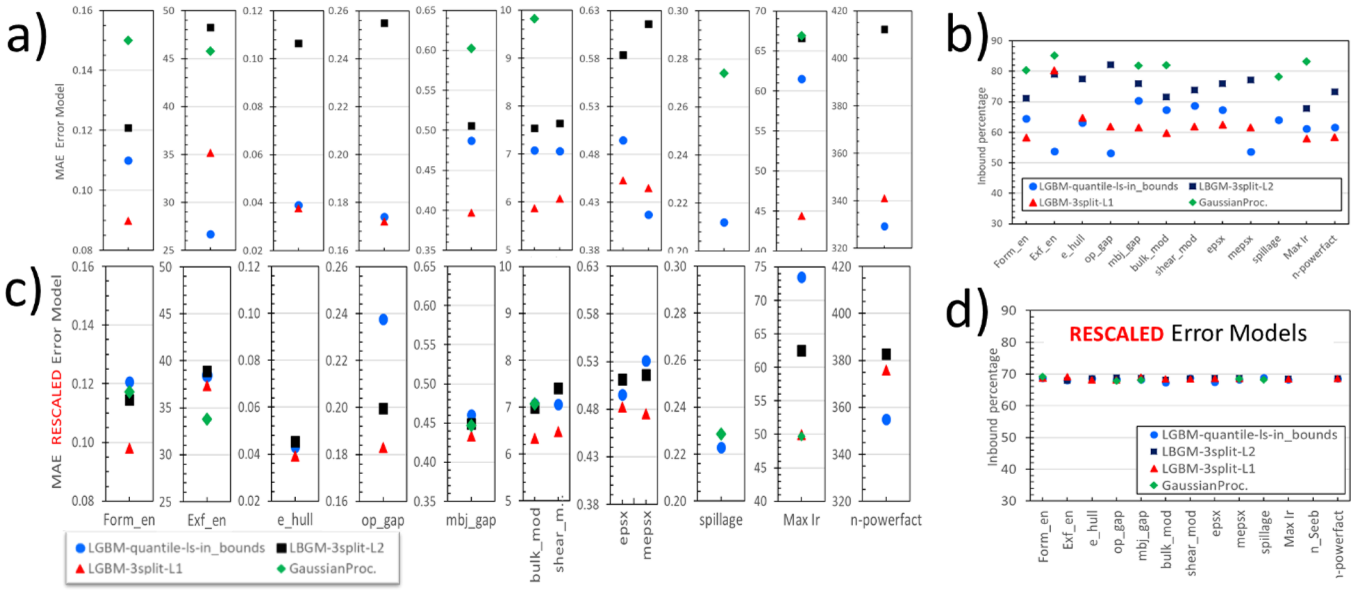


Figure 5. Population variables for the four error models investigated in this work: MAEs in (a) and inbound percentages in (b). The same quantities for rescaled error models in (c) and (d). As the inbound percentages become very similar among models, the corresponding MAEs get closer as well, with 3split-L1 being the lowest in most of the properties. To facilitate comparison, we kept the *y*-axis scale the same between the error model as-fitted cases (a) and the rescaled ones (c). Each MAE is in the units of the corresponding property, and details on the properties and their units are given in the caption of Figure 2.

periodic trends need to be considered. As the name hints, the WhiteNoise component of the kernel accounts for white noise on the data. As our descriptors are very different in nature, going from chemical ones to structural and coordination-related ones, we found the importance to allow different length-scale parameters for each of them, at least in the RDF term. All fitting parameters (λ_i for i = 1,..N, l, l', l'', p, α, noise, and c_i, where i = 1, 2, and 3) are optimized using the Broyden–Fletcher–Goldfarb–Shanno (L-BGFS-B) algorithm.[37]

CFID descriptors provide a complete set of structural and chemical features through 1557 descriptors for each material. Although 1557 is not an unreasonable number of descriptors/features for most of the ML models, they are too many to be effectively dealt with using Gaussian processes. We used the "feature importance" capability of gradient boosting methodologies to reduce their number to something manageable, between 15 and 30, depending on the property. In Figure 4a, data leading to our choice of which descriptors to use for each property are shown. Out of the 50 most important descriptors, the majority are chemical descriptors, for all properties but optical. After the chemical ones, either those distance-related or coordination-related are the most common ones. The error bars come from averaging importance results obtained in the three most successful GBDT models, for each property. There is some variability, among runs, on which the descriptor, in each category, is among the 50 most important. Therefore, the intersection of the 50 most important descriptors between the three best models, for each property, gave us a manageable number of descriptors to use when fitting Gaussian processes. Using a combination of the features, instead of a down selection of them, is an alternative approach that loses some of the direct interpretability of each feature but that may possibly lead to tighter inbound predictions.

Figure 4b shows our results in the case of the bulk modulus. Compared to quantile (Figure 2) and "3split" (Figure 3) results, the GP predicts less "very accurate" errors (dark blue, less than 2 GPa error), many more "accurate" ones (medium blue, 2 to 10 GPa error), and almost none with error larger than 17 GPa (green or above). This behavior, a reduced dispersion in the predicted error, occurs consistently in all six investigated properties and always corresponds to a very high inbound percentage (usually around 80%).

3.4. Prediction Intervals: Discussion. After having explored individual results from each of our modeling approaches, we are finally ready to quantitatively compare them across all the properties. Figure 5a shows error MAE results, for all four approaches, having chosen models with low MAEs and inbound percentages closest to 68%. Having a comparable inbound count is necessary to get an accurate MAE comparison. However, as displayed in Figure 5c, no matter how much time and focus we put in the fits, we could not obtain models with exactly 68% inbound percentages (one standard deviation). Again, fitting for lower MAEs and higher inbound counts at the same time is not straightforward, as the hyperparameter search is driven only by the minimization of the loss function. Figure 5c shows that the GP inbound count is systematically around 80%, 3split-L2 between 70 and 80%, while 3split-L1 is consistently around 60%. The only exception is exfoliation energy, which is the quantity with an incredibly small dataset (only about 600 data points), so its results are often unpredictable. Quantile has the largest spread in its inbound percentages, from just above 50 to 70%. In terms of MAEs for the error models, quantile and 3split-L1 are consistently lower than 3split-L2 and GP. The systematic difference between 3split-L1 and 3split-L2 hints to the importance of how the error is modeled (in 3split-L1, the absolute value of the error is machine-learned directly, while in 3split-L2, it is the absolute value of the error to be machine-learned directly). All MAEs and inbound counts discussed here (error models) are given in Table S1 in the Supporting Information (SI), together with the corresponding MAEs for

https://doi.org/10.1021/acsomega.1c03752
ACS Omega 2021, 6, 32431−32440

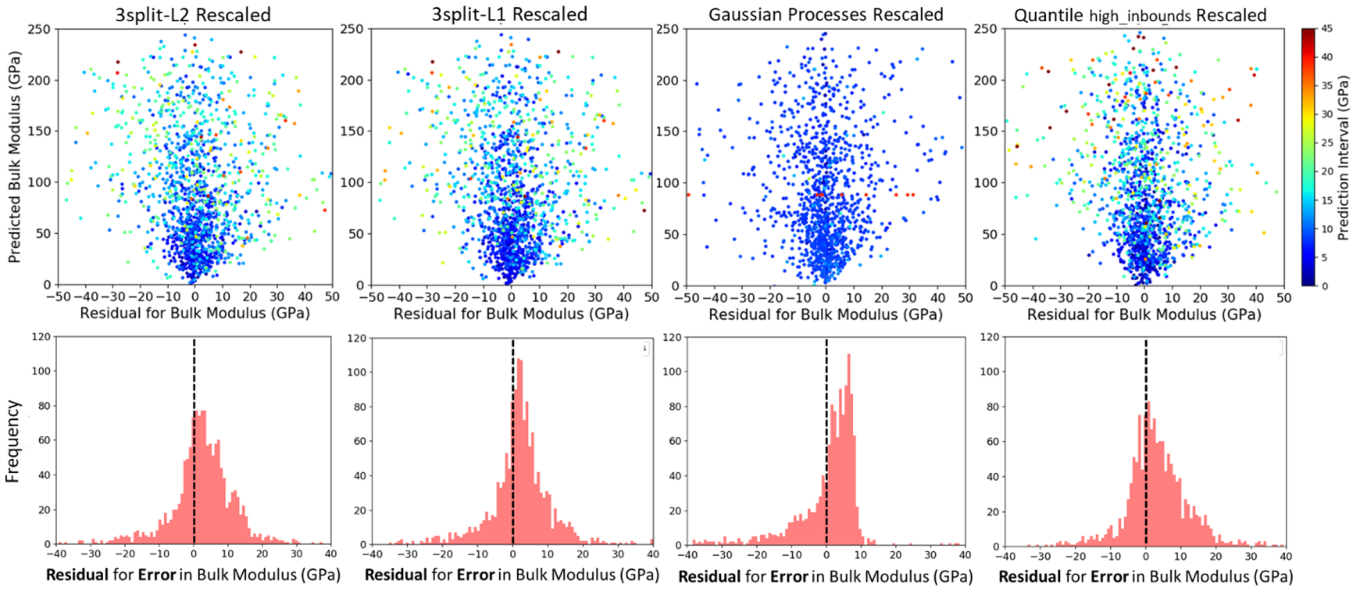


Figure 6. Top row: residual for the base model (predicted property − DFT value) versus the predicted property, in the case of the bulk modulus, for the four examined approaches. The data point color indicates the predicted error (prediction interval). The distribution of predicted intervals depends on the modeling approach. Bottom row: histograms for the error model residual (predicted error − exact error). An even stronger dependence on the modeling approach is evident here, as the height, range, and symmetry of the residual distribution vary significantly among error models.

the base models (i.e., the models predicting the physical quantities themselves).

As an accurate comparison of MAEs for the four methodologies is only possible for very similar inbound counts, we rescaled the prediction intervals, to obtain an inbound percentage of 68%. This was accomplished by multiplying all validation prediction intervals, for each physical property and error modeling approach, by a fitted factor, which is chosen so that the inbound percentage is 68% or as close to it as possible. Such factors are approach- and property-dependent, and they are listed in Table S2 in the SI. The error model MAEs are then recomputed using the scaled validation sample values. Results are given in Figure 5. By construction, all rescaled inbound counts are very close to 68% (Figure 5d). The corresponding MAEs (Figure 5c) are closer together than for the raw (not rescaled) case (Figure 5a), highlighting how important it is to compare prediction interval-determining methodologies only when the inbound counts are similar. For certain properties, all models give essentially the same MAE (MBJ bandgap, for instance), while in other cases, like for the maximum predicted infrared mode (Max IR), there is still a substantial difference. Overall, it is 3split-L1 to achieve the lowest error MAEs across most of the properties followed by GP. As the "3split" models are the easiest/fastest to fit, this is a good finding. Quantile does not always work as well as it could have been expected: in the case of two properties (OPTB88vdW(OPT)-bandgap and Max IR), it gives the highest MAEs for the error model.

MAEs give important information on how a model behaves but not completely. In Figure 6, the residuals for the base property (top row) and for the error prediction (bottom row) are displayed, in the case of the bulk modulus. Similar results are found for all the other properties investigated in this work. In the top row images, the data point color indicates the predicted error (i.e., prediction interval). These results show that while all the approaches are reasonably accurate both as a base (most of the data fall between +10 and −10 GPa) and as an error model (most of the blue data points fall between +10 and −10GPa, and nonblue data fall outside this range), the dispersion of the error distribution depends on the modeling approach. In other words, how many blue points actually fall outside the ±10 GPa range and, conversely, how many nonblue points fall inside such a range depend on how the prediction interval is determined. To directly investigate the uncertainty in the error prediction, we plot the residual histogram for the prediction interval models (Figure 6, bottom row). A strong dependence on the modeling approach is evident from these plots, larger than what the similarities among MAEs could have led us to expect, as the height, range, and symmetry of the residual distribution vary significantly among error models. Once again, 3split-L1 produces the best results, as its error−residual histogram is the most symmetric, is centered very close to zero, and is the tallest and the one with the smallest full-width-half-maximum. As the base model is the same among 3split-L1 and 3split-L2, all the differences between these two cases are to be attributed to the different definitions of "error". As already observed when commenting on predicted values versus DFT ones for GP (Figure 4b), the GP is characterized by getting a smaller spread in the predicted uncertainties than the other models, which leads to over-estimating the error of well-predicted data and, in a few cases, to underestimate the error of badly predicted ones. This is evidenced by its histogram peak not being centered around zero, as very few prediction intervals are predicted to be almost zero, even when the corresponding property values are very close to the exact ones. Conversely, not many large overpredictions occur, as made evident by the sharp dropping off of the histogram after 10 GPa.

4. CONCLUSIONS

In this work, we compare three different approaches to determine the uncertainty on individual machine learning predictions (prediction intervals). Specifically, we probe the

https://doi.org/10.1021/acsomega.1c03752
ACS Omega 2021, 6, 32431−32440

quantile loss function approach, used machine learning of the error directly, where the error is defined either as the absolute difference (3split-L1) or the square of the difference (3split-L2) between the predicted and the observed values, and used Gaussian processes. All approaches are applied to the modeling of 12 physical properties, ranging from energetics-related ones to elastic, optical, electronic, and more. This investigation is necessary because quantities like MAE only evaluate ML models in a statistical manner, and while users of such models need to know how reliable the specific prediction, they are interested in this. We identify each approach's advantages and disadvantages, and we learned that the descriptor choice matters. With good descriptors (like the CFID), even ML algorithms with default hyperparameters (random forest, in our investigation) give acceptable results. The choice of hyperparameters is particularly crucial when developing Gaussian process models. We find that Gaussian processes give a good estimate for prediction intervals, although a bit overestimated, but are more time-consuming to fit than the other approaches. Using the quantile approach requires fitting three models and, in general, gives lower inbound results than Gaussian processes or 3split-L2. However, among the three models, it is the easiest approach to fit for specific inbound results. Machine learning the error directly has the enormous advantage of allowing the use of any loss function. However, it requires splitting the dataset into three parts, which could be a problem if the dataset is small to begin with. How the error is modeled also makes a difference, and using the absolute difference between predicted and expected (3split-L1) values was found to be, overall, the best approach. All data for training and testing were taken from the publicly available JARVIS-DFT database, and the codes developed for computing the prediction intervals are available through JARVIS-tools github (https://github.com/usnistgov/jarvis).

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsomega.1c03752.

MAEs for the base model, MAEs for error models, and inbound percentages for data in Figure 5a,b; scaling factors; dataset information: size and distribution; list of acronyms/terminology used in the text (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Kamal Choudhary − *Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, United States;* orcid.org/0000-0001-9737-8074; Email: kamal.choudhary@nist.gov

Authors

Francesca Tavazza − *Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, United States*

Brian DeCost − *Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, United States;* orcid.org/0000-0002-3459-5888

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.1c03752

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We gratefully acknowledge the National Institute of Standards and Technology and the Materials Genome Initiative (MGI) for funding and support.

■ REFERENCES

(1) Agrawal, A.; Choudhary, A. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *APL Mater.* **2016**, *4*, No. 053208.

(2) Hkdh, B. Neural networks in materials science. *ISIJ Int.* **1999**, *39*, 966−979.

(3) Pun, G. P. P.; Batra, R.; Ramprasad, R.; Mishin, Y. Physically informed artificial neural networks for atomistic modeling of materials. *Nat. Commun.* **2019**, *10*, 2339−2310.

(4) Kusne, A. G.; Gao, T.; Mehta, A.; Ke, L.; Nguyen, M. C.; Ho, K.-M.; Antropov, V.; Wang, C.-Z.; Kramer, M. J.; Long, C.; Takeuchi, I. On-the-fly machine-learning for high-throughput experiments: search for rare-earth-free permanent magnets. *Sci. Rep.* **2015**, *4*, 6367−6367.

(5) Liu, Y.; Zhao, T.; Ju, W.; Shi, S. Materials discovery and design using machine learning. *J. Materiomics* **2017**, *3*, 159−177.

(6) Mueller, T.; Kusne, A. G.; Ramprasad, R. Machine learning in materials science: Recent progress and emerging applications. *Rev. in Comput. Chem.* **2016**, *29*, 186−273.

(7) Wei, J.; Chu, X.; Sun, X. Y.; Xu, K.; Deng, H. X.; Chen, J.; Wei, Z.; Lei, M. Machine learning in materials science. *InfoMat* **2019**, *1*, 338−358.

(8) Schmidt, J.; Marques, M. R. G.; Botti, S.; Marques, M. A. L. Recent advances and applications of machine learning in solid-state materials science. *npj Comput. Mater.* **2019**, *5*, 83−36.

(9) Vasudevan, R. K.; Choudhary, K.; Mehta, A.; Smith, R.; Kusne, G.; Tavazza, F.; Vlcek, L.; Ziatdinov, M.; Kalinin, S. V.; Hattrick-Simpers, J. Materials science in the artificial intelligence age: high-throughput library generation, machine learning, and a pathway from correlations to the underpinning physics. *MRS Commun.* **2019**, *9*, 821−838.

(10) Bartel, C. J.; Trewartha, A.; Wang, Q.; Dunn, A.; Jain, A.; Ceder, G. A critical examination of compound stability predictions from machine-learned formation energies. *npj Comput. Mater.* **2020**, *6*, 1−11.

(11) Ye, W.; Chen, C.; Wang, Z.; Chu, I.-H.; Ong, S. P. Deep neural networks for accurate predictions of crystal stability. *Nat. Commun.* **2018**, *9*, 3800−3806.

(12) Choudhary, K.; DeCost, B.; Tavazza, F. Machine learning with force-field-inspired descriptors for materials: Fast screening and mapping energy landscape. *Phys. Rev. Mater.* **2018**, *2*, No. 083801.

(13) Zeng, S.; Li, G.; Zhao, Y.; Wang, R.; Ni, J. Machine learning-aided design of materials with target elastic properties. *J. Phys. Chem. C* **2019**, *123*, 5042−5047.

(14) Choudhary, K.; Cheon, G.; Reed, E.; Tavazza, F. Elastic properties of bulk and low-dimensional materials using van der Waals density functional. *Phys. Rev. B* **2018**, *98*, No. 014107.

(15) De Jong, M.; Chen, W.; Notestine, R.; Persson, K.; Ceder, G.; Jain, A.; Asta, M.; Gamst, A. A statistical learning framework for materials science: application to elastic moduli of k-nary inorganic polycrystalline compounds. *Sci. Rep.* **2016**, *6*, 34256−34211.

(16) Chandrasekaran, A.; Kamal, D.; Batra, R.; Kim, C.; Chen, L.; Ramprasad, R. Solving the electronic structure problem with machine learning. *npj Comput. Mater.* **2019**, *5*, 22−27.

(17) Pilania, G.; Mannodi-Kanakkithodi, A.; Uberuaga, B. P.; Ramprasad, R.; Gubernatis, J. E.; Lookman, T. Machine learning bandgaps of double perovskites. *Sci. Rep.* **2016**, *6*, 19375−19310.

(18) Stein, H. S.; Guevarra, D.; Newhouse, P. F.; Soedarmadji, E.; Gregoire, J. M. Machine learning of optical properties of materials−predicting spectra from images and images from spectra. *Chem. Sci.* **2019**, *10*, 47−55.

(19) Choudhary, K.; Tavazza, F. Convergence and machine learning predictions of Monkhorst-Pack k-points and plane-wave cut-off in high-throughput DFT calculations. *Comput. Mater. Sci.* **2019**, *161*, 300−308.
(20) Choudhary, K.; Kalish, I.; Beams, R.; Tavazza, F. High-throughput identification and characterization of two-dimensional materials using density functional theory. *Sci. Rep.* **2017**, *7*, 5179−5116.
(21) Choudhary, K.; Garrity, K. F.; Reid, A. C. E.; DeCost, B.; Biacchi, A. J.; Walker, A. R. H.; Trautt, Z.; Hattrick-Simpers, J.; Kusne, A. G.; Centrone, A.; Davydov, A.; Jiang, J.; Pachter, R.; Cheon, G.; Reed, E.; Agrawal, A.; Qian, X.; Sharma, V.; Zhuang, H.; Kalinin, S. V.; Sumpter, B. G.; Pilania, G.; Acar, P.; Mandal, S.; Haule, K.; Vanderbilt, D.; Rabe, K.; Tavazza, F. The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. *npj Comput.l Mater.* **2020**, *6*, 173−113.
(22) Koenker, R.; Hallock, K. F. Quantile regression. *JEP* **2001**, *15*, 143−156.
(23) Hastie, T.; Tibshirani, R.; Friedman, J., *The elements of statistical learning: data mining, inference, and prediction*; Springer Science & Business Media: 2009.
(24) Berger, J. O., *Statistical decision theory and Bayesian analysis*; Springer Science & Business Media: 2013.
(25) Shrestha, D. L.; Solomatine, D. P. Machine learning approaches for estimation of prediction interval for the model output. *Neural Netw.* **2006**, *19*, 225−235.
(26) Khosravi, A.; Nahavandi, S.; Creighton, D. Quantifying uncertainties of neural network-based electricity price forecasts. *Appl. energy* **2013**, *112*, 120−129.
(27) Quan, H.; Srinivasan, D.; Khosravi, A. Particle swarm optimization for construction of neural network-based prediction intervals. *Neurocomputing* **2014**, *127*, 172−180.
(28) Ak, R.; Li, Y.; Vitelli, V.; Zio, E.; Droguett, E. L.; Jacinto, C. M. C. NSGA-II-trained neural network approach to the estimation of prediction intervals of scale deposition rate in oil & gas equipment. *Expert Syst. Appl.* **2013**, *40*, 1205−1212.
(29) Khosravi, A.; Nahavandi, S.; Creighton, D. A neural network-GARCH-based method for construction of Prediction Intervals. *Electr. Power Syst Res.* **2013**, *96*, 185−193.
(30) Khosravi, A.; Nahavandi, S.; Creighton, D.; Atiya, A. F. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE Trans Neural Netw Learn Syst* **2011**, *22*, 337−346.
(31) Ling, J.; Hutchinson, M.; Antono, E.; Paradiso, S.; Meredig, B. High-dimensional materials and process optimization using data-driven experimental design with well-calibrated uncertainty estimates. *Integr. Mater.* **2017**, *6*, 207−217.
(32) Rasmussen, C. E. In *Gaussian processes in machine learning, Summer school on machine learning*; Springer: 2003, pp. 63−71.
(33) Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146−3154.
(34) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn Res.* **2011**, *12*, 2825−2830.
(35) Head, J. D.; Zerner, M. C. A Broyden—Fletcher—Goldfarb—Shanno optimization procedure for molecular geometries. *Chem. Phys. Lett.* **1985**, *122*, 264−270.
(36) Duvenaud, D.. Automatic model construction with Gaussian processes; PhD diss., University of Cambridge: 2014.
(37) Fletcher, R., *Practical methods of optimization*; John Wiley & Sons: 2013.

https://doi.org/10.1021/acsomega.1c03752
ACS Omega 2021, 6, 32431−32440