Building robust machine learning models for small chemical science data: the case of shear viscosity of fluids

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 045032 https://doi.org/10.1088/2632-2153/acac01

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
2 September 2022

REVISED
2 December 2022

ACCEPTED FOR PUBLICATION
15 December 2022

PUBLISHED
29 December 2022

PAPER

Building robust machine learning models for small chemical science data: the case of shear viscosity of fluids

Nikhil V S Avula*, Shivanand Kumar Veesam, Sudarshan Behera and Sundaram Balasubramanian*

Chemistry and Physics of Materials Unit, Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore 560064, India
* Authors to whom any correspondence should be addressed.

E-mail: nikhil@jncasr.ac.in and bala@jncasr.ac.in

Keywords: small data, shear viscosity, model selection, performance estimation, uncertainty quantification, applicability domain

Supplementary material for this article is available online

Abstract

Shear viscosity, though being a fundamental property of all fluids, is computationally expensive to calculate from equilibrium molecular dynamics simulations. Recently, machine learning (ML) methods have been used to augment molecular simulations in many contexts, thus showing promise to estimate viscosity too in a relatively inexpensive manner. However, ML methods face significant challenges—such as overfitting, when the size of the data set is small, as is the case with viscosity. In this work, we train seven ML models to predict the shear viscosity of a Lennard–Jones fluid, with particular emphasis on addressing issues arising from a small data set. Specifically, the issues related to model selection, performance estimation and uncertainty quantification were investigated. First, we show that the widely used performance estimation procedure of using a single unseen data set shows a wide variability—in estimating the errors on—small data sets. In this context, the common practice of using cross validation (CV) to select the hyperparameters (model selection) can be adapted to estimate the generalization error (performance estimation) as well. We compare two simple CV procedures for their ability to do both model selection and performance estimation, and find that k-fold CV based procedure shows a lower variance of error estimates. Also, these CV procedures naturally lead to an ensemble of trained ML models. We discuss the role of performance metrics in training and evaluation and propose a method to rank the ML models based on multiple metrics. Finally, two methods for uncertainty quantification—Gaussian process regression (GPR) and ensemble method—were used to estimate the uncertainty on individual predictions. The uncertainty estimates from GPR were also used to construct an applicability domain using which the ML models provided even more reliable predictions on an independent viscosity data set generated in this work. Overall, the procedures prescribed in this work, together, lead to robust ML models for small data sets.

1. Introduction

Shear viscosity is a fundamental transport property of all fluids [1]. Understanding its molecular underpinnings would advance our scientific understanding of supercooled liquids [2], magma transport [3], mixing of fluids, etc. For example, a good estimate of shear viscosity is crucial to model the Earth's outer core which is believed to be a liquid form of iron based alloys [4, 5]. However, in the absence of direct measurements, its estimates from different methods differ by about fourteen orders of magnitude [6]. Hence, a better understanding of the behavior of viscosity from simulations can help address some of these issues. Much progress has been made in this area recently and we refer the interested readers to some excellent works [7–11]. Further, from the point of view of applications, predicting the viscosity of industrially relevant fluids (such as hydrocarbons and carbonates) *insilico* would accelerate the progress in energy storage, petroleum, lubricants, chemical processing, pharmaceutical, and many other sectors [12, 13].

1.1. Viscosity from molecular dynamics simulations

Atomistic molecular dynamics (MDs) simulations with *ab initio* or empirical force fields can be used to estimate the viscosity of any liquid, however complex, *in silico* [14–22]. While there exists many methods to estimate viscosity from MD simulations, they largely fall into two categories—equilibrium MD (EMD) [14, 23] and non-equilibrium MD (NEMD) based methods [24–26]. A comparison between them is beyond the scope of this work and the readers are directed to some excellent works in this area [14, 15].

Despite the progress in this area [14, 15, 19, 23, 27–34], the state-of-the-art methods to estimate viscosity accurately from MD simulations require huge computing time especially for viscous fluids [23, 35–37], as it is a collective quantity. This drawback precludes the use of MD simulations in viscosity-based high throughput screening processes in the industry [12]. Also, force field refinement strategies which use the experimental viscosity as a benchmark require significant effort for the same reason [38]. This is especially pertinent because many general purpose classical force fields cannot reproduce viscosity (and other transport properties) well [35, 39–41]. Another important difficulty in estimating viscosity from MD is its sensitive dependence on primary and ancillary simulation setup parameters. Hess showed that ancillary MD run parameters such as the number of independent replicas, numerical precision of the MD engine, neighbor list cutoff, and Ewald sum related parameters have a significant effect on the estimated viscosity value [15]. Hence, reporting meaningful confidence intervals of viscosity estimates is crucial and is a topic of ongoing research [14, 15, 28, 29, 42]. To address these problems related to estimating shear viscosity from MD simulations, we look at alternate approaches using machine learning (ML) methods.

1.2. ML methods

Recently, ML and deep learning (DL) models have started to augment various aspects of MD simulations [43–49]. More specifically, in the context of using ML methods to predict slowly converging properties of liquids—of which shear viscosity is one—some initial advances have been made. Allers *et al* used neural networks (NNs) and random forests (RFs) to predict the self diffusion coefficients of particles interacting via Lennard–Jones (LJ) interaction and showed them to be performing better than empirical models [50]. They also used ML methods to predict the finite size corrections to self diffusion coefficients in binary LJ fluids [51]. Also, several ML models were used to predict the experimental viscosity and ionic conductivity of ionic liquids using only molecular features [52–60]. However, to our knowledge, ML methods have not been used to predict shear viscosity derived purely from MD simulations.

The protocols for developing and comparing ML models—especially for chemical science applications—are not yet completely established [61]. There exist a plethora of supervised learning algorithms [62, 63], model selection rules [64–68], performance metrics [61, 69, 70], uncertainty quantification methods [71–75], and hyperparameter tuners [76] which are used to create ML models; yet, there are no clear protocols on which combination should be chosen. For example, among the many performance metrics that are used to train and compare ML models, the best choice is still a matter of debate [61, 77, 78]. Similar conclusions hold true for model selection [68], and uncertainty quantification [79] as well.

In this context, 'No-Free-Lunch' theorems by Wolpert and Mcready imply that there is no single algorithm that has the best performance across all possible optimization problems [80]. These theorems when applied to ML indicate that any single ML algorithm cannot be expected to perform well across all possible ML tasks. Though there is still a debate on the applicability of these theorems to practical ML problems, it is considered common knowledge that the ML algorithm should be tailored to the specific task at hand [63]. Exploiting the idiosyncrasies of the data set can lead to significant improvements in the performance of ML models. For example, Hansen *et al* were able to improve the performance (mean absolute error (MAE)) of ML models used to predict atomization energies from 10 kcal mol^{-1} to 3 kcal mol^{-1} (70% improvement) by exploring a number of ML techniques and molecular representations [81]. In this work, we use this approach to develop ML models tailored to viscosity data set generated from EMD simulations.

1.3. ML models for small data

As it is computationally costly to get reliable (including standard errors) estimates of viscosity from atomistic MD, generating large data sets (like GDB-17 etc with nearly 150 billion data points [82, 83]) is practically infeasible with the current computing resources. The largest MD-derived viscosity data set (with 1061 data points) we could find was the work of Jamali *et al* in which the MD computed viscosity was used to predict the box size corrections to diffusion coefficients of particles in binary LJ systems [84]. Such small data sets pose unique challenges to the ML methods [85]—(a) they are hard to generalize, (b) they are susceptible to overfitting, and (c) they tend to underestimate the generalization error [65, 86]. For example, NNs which generally outperform other ML models, struggle in the low data regime [81, 87, 88]. Also, Vabalas *et al*, on surveying over 50 articles on ML for autism, have shown that ML models tended to produce overoptimistic

results when the sample sizes are small [89]. All these issues exacerbate the reproducibility of the results which is already a fast growing challenge in all fields using ML methods [90–92].

In this work, we train seven ML models to predict the shear viscosity of binary LJ fluids, with particular emphasis on addressing issues arising from a small data set. Specifically, the issues related to model selection, performance estimation, performance metrics, uncertainty quantification and applicability domain (AD) were investigated. First, we show that the common practice of estimating the performance of the ML models on a single unseen data set shows wide variability for small data sets. The consequences of using individual unseen data set on the hyperparameter optimization landscape are demonstrated. Then, we compare two simple CV procedures for their ability to do both the model selection and performance estimation together. We discuss the role of performance metrics in selecting and evaluating ML models. We discuss some general principles for comparing different metrics and use them to choose a suitable set of metrics relevant to the viscosity data set. We propose a holistic ranking method based on multiple metrics to choose the best performing ML algorithms. To complement the traditional ML models, we train a probabilistic model to capture the inherent uncertainty in the data set and compare its performance with that of other models. The performance of the ML models developed here is shown to be better than empirical models for viscosity. Finally, the AD of ML models is also constructed (and tested) to assist the decision making of the end user. We believe that the techniques adopted herein to train the ML models, combined with the uncertainty quantification and AD can lead to accurate, reliable and *reproducible* ML models to predict shear viscosity of binary LJ mixtures and can help researchers to develop ML models for small data sets in general. Also, we hope that the detailed descriptions and the codes attached in the supporting information would help with the reproducibility of the results presented in this work.

The paper is organized as follows: the next section describes the background theory and empirical evidence that aids in the discussion on model selection and performance metrics. It is followed by sections on computational details and results. The final section presents our conclusions and suggestions for developing ML models for small data sets.

2. Background

2.1. The structure of the problem

We assume that there exists a joint probability density $p(\mathrm{x}, y)$ that generated the data set [62, 63]. Here, x is a vector of input features and y is the target variable, also called as the label. In the context of shear viscosity prediction, the feature vector can be constructed from quantities like x_1, σ_2, ϵ_2, k_{12}, ζ, ρ^* and the target variable is the shear viscosity η (see section 3.1). We focus on the regression task which aims to determine a function $f^*(\mathrm{x})$ that is an *optimal* representation of the data set. The sense in which the function $f^*(\mathrm{x})$ is *optimal* is often taken to be the one that minimizes the expected loss (also called as *risk*) $\mathbb{E}[L]$ (equation (1)). Most common ML models like kernel ridge regression (KRR), support vector regression (SVR), NN, etc fall under this category.

$$f^*(\mathrm{x}) = \arg\min_{f(\mathrm{x})} \mathbb{E}[L] = \arg\min_{f(\mathrm{x})} \iint L(y, f(\mathrm{x}))\, p(\mathrm{x}, y)\, \mathrm{d}\mathrm{x}\, \mathrm{d}y \tag{1}$$

where $L(y, f(\mathrm{x}))$ is the user-defined loss function. The choice of the loss function has a direct relation to the kind of function $f^*(\mathrm{x})$ obtained [77]. The most common loss function, the squared loss, where $L(y, f(\mathrm{x})) = (y - f(\mathrm{x}))^2$ yields the conditional mean $\mathbb{E}_y[y|\mathrm{x}]$ (equation (S2)) as the $f(\mathrm{x})$ [62]. Discussion on other loss functions and the consequent effect on the properties of $f^*(\mathrm{x})$ is presented in section S1.3.

However, to compute the expected loss/risk, the underlying joint probability density $p(\mathrm{x}, y)$ has to be known which is hard to do in practice. Hence, the expected loss is approximated by the empirical loss, $\mathbb{E}_{\mathrm{emp}}[L]$

$$f^*(\mathrm{x}) = \arg\min_{f(\mathrm{x})} \mathbb{E}_{\mathrm{emp}}[L] = \arg\min_{f(\mathrm{x})} \left(\frac{1}{N} \sum_{i=1}^{N} L(y_i^{\mathrm{true}}, f(\mathrm{x}_i)) \right) \tag{2}$$

where, N is the number of data points, y_i^{true} are the target values corresponding to the feature vector x_i. Generally, the empirical loss tends to be much lesser on the data set used to infer $f^*(\mathrm{x})$ (called the training set) than on new/unseen data set(s). This is because the minimization of empirical loss (*per se*) incentivizes the learning machine to learn the peculiarities (like noise) of the particular training data sample rather than the trends in the underlying model that generated that data set [62, 63, 93]. Hence, the goal of the learning protocol should be to minimize the error on new/unseen data set(s) called the generalization error. This phenomena of ML methods having significantly lesser training error than the generalization error is called overfitting and is especially relevant for models on small data sets [62, 63, 93].

The most common way to alleviate the problem of overfitting is to reduce the complexity/capacity of the learning machine, thereby reducing its ability to learn the noise associated with the training data sample. However, the complexity should not be reduced to such an extent that the general trends in the data are lost, resulting in underfitting. Hence, the ML model should choose an *optimal* complexity corresponding to the general trends in the data. A popular method to control the complexity of the models is called regularization in which a penalty term (called the regularizer, equation (3)) which penalizes complex models is added to the empirical loss [63, 93]. The common forms of the regularizer are based on the norm of the weights (w) of the model like—L^2 norm (called as ridge regression or Tikhonov regularization), L^1 norm (for example in least absolute shrinkage and selection operator (LASSO) model), or a combination of both [63]. We also note that there are many other regularization techniques that are specific to DL methods such as—early stopping, dropout, soft weight sharing, etc [94].

$$f^*(\mathrm{x}) = \arg\min_{f(\mathrm{x})} J = \arg\min_{f(\mathrm{x})} \left(\frac{1}{N} \sum_{i=1}^{N} L(y_i^{\mathrm{true}}, f(\mathrm{x_i})) + \sum_j \lambda_j \Omega_j(f) \right). \tag{3}$$

Where $\Omega_j(f)$ are the regularizers and λ_j are the parameters that control the amount of regularization. Now that the ML models have a mechanism to control the complexity through regularization, the natural next step would be to choose the values of regularization parameters. This task falls under the purview of model selection [63]. In the following section, we discuss various model selection criteria and a closely related topic of performance evaluation.

2.2. Model selection and performance evaluation

It is a common practice to distinguish the parameters of ML and DL models into model parameters and hyperparameters [62, 63]. The model parameters are learnt during the training phase on the training data. Examples of model parameters include—slope and intercept in linear regression, coefficients of kernel expansion in kernel methods (like KRR), weights of neurons in NNs, etc. The hyperparameters are generally the high level settings of ML algorithms which are either set by the user or inferred during the model selection procedure. Examples of hyperparameters include—regularization parameters, the degree of polynomial in polynomial regression, choice of the kernel in kernel methods, choice of activation function in NNs, number of neurons in NNs, etc.

The task of selecting the model with the *optimal* complexity is reduced to the estimation of values of hyperparameters; the criteria used for such selection are called model selection criteria. As stated earlier, the goal of ML models is to minimize the generalization error which is the average error over *all* unseen data. However, generalization error cannot be obtained in most practical situations and hence estimators on finite data sets are constructed to approximate it. The process of estimating the generalization error by using estimators on finite data sets is called performance evaluation and is a prerequisite for model selection. It is crucial to note that the error estimates are obtained over finite data sets and hence depend on the size of the data set, especially for small data sets. A simple example of such an estimator is the split sample estimator where the whole data set is split into two parts (generally unequal) and the error is computed on the split that was not used for training [65]. Split sample estimator is known to be unbiased i.e. the average split sample error over multiple independent realizations of unseen data asymptotically converges to the generalization error. Hence, minimizing the split sample error can in principle reduce the generalization error. However, it was recently shown that the unbiasedness *per se* is not as important as the variance of the estimator when it is used for model selection [65]. When an estimator has high variance (occurs with small data sets [65]) the value of the estimated error on any one particular unseen data sample can be very different from the generalization error; hence the hyperparameters that minimize the estimated error can be far off from the *optimal* ones. Cawley and Talbot showed (on a synthetic data set) that hyperparameters selected based on split sample estimators can severely overfit or underfit the data [65]. In practice, the users rarely have the capability of generating multiple independent realizations of the data and hence the variance of the estimator plays a major role. Therefore, for small data sets it is not considered a good practice to estimate the error on a single realization of the data set [65]. In order to mitigate this problem, various cross-validation (CV) schemes are generally used.

The core idea of k-fold CV is to split the entire data set into k equal disjoint sets, train the ML models on k-1 sets and estimate the error on the remaining one set. This process is repeated k times, each time with a different hold-out set [62, 63]. The average error over k-folds is used as the estimate for the generalization error. It is a common practice to use 5- or 10-folds during CV [63].

The error estimates from k-fold CV are often used for model selection by searching over the space of hyperparameters and choosing the one that yields minimum CV error. But once the k-fold CV error is used

to optimize the hyperparameters, it is no longer unbiased [63, 95, 96]. Typically another unseen data set (called the test set) is used to estimate the generalization error of the models with optimized hyperparameters [63]. Using a single realization of the test set, however, suffers from the high variance issue discussed above. Nested CV or double CV improves upon k-fold CV by doing performance evaluation and model selection in two nested loops [65, 81, 89, 95, 96]. The outer loop is used to estimate the generalization error and the inner loop is used to select the hyperparameters. Also, we note that there are many methods of splitting the data set into train/validation/test sets such as—Monte-Carlo CV, bootstrapping, Kennard–Stone splitting, and combinations thereof [68, 97]. Xu and Goodacre compared the performance (in terms of their ability to predict the generalization error) of various data splitting methods including k-fold CV, Monte-Carlo CV, bootstrapping, etc and found that a single best method could not be found *a priori* and suggest that the choice of the method should be tuned to the kind of data (No Free Lunch again) [68].

Finally, we note that model selection and performance evaluation are big and unsolved challenges on small data sets [63, 65, 86, 95–98]. Guyon *et al* organized a performance prediction challenge in which the participants (more than 100) were asked to predict the generalization error on finite data sets of real world importance like medical diagnosis, speech recognition, text categorization, etc [97]. They observed that most submissions were overconfident about their ML models i.e. their prediction of generalization error is less than the true generalization error. They also noted that the performance of the ML models truly improved in the first 45 days of the 180 day challenge after which overfitting set in. It is now a common belief that when a data set is worked upon repeatedly, even careful performance prediction protocols can result in optimistic performance predictions over time [63].

2.3. Performance metrics for regression

In this section, we summarize some of the principles that can be used to choose a relevant metric to the particular ML task at hand and also consider the particular case of viscosity data set. Performance metrics are generally used in two critical areas of ML model development workflow—model training and model comparison. Though the choice of the metric can significantly alter the *kind* of ML model developed and consequently its real-world performance, there is no clear consensus on this topic [61, 69, 77]. As is the case with model selection criterion, there is no single best metric *for model training* that can be used across all ML tasks [69]. Further discussion on this topic can be found in section S1.3.

Another area in which loss functions are used in ML workflow is model comparison, in which models are ranked based on their generalization performance. Ideally, the generalization performance of ML models should also be measured using the same metric used in their training phase [77]. For example, an ML model trained by minimizing mean squared error (MSE) should be compared to other models using MSE generalization error. However, in many cases, the choice of the loss functions cannot be controlled by the model developers and hence it is difficult to choose just one metric to compare such models. For example, Makridakis *et al* use a weighted average of symmetric mean absolute percentage error (sMAPE) and mean absolute scaled error (MASE) to compare the models in the M4 forecasting competition citing a lack of agreement on the advantages and drawbacks of various metrics [78]. Hence, it is generally recommended to report the estimates of generalization error using multiple metrics [61, 69, 81, 88]. Also, given the proliferation of various metrics, it is important to choose the set of metrics that are relevant to the ML task at hand and preferably containing complementary information to each other. Armstrong and Callopy compared six commonly used metrics and ranked them qualitatively (good, fair, poor) according to five characteristics—reliability, construct validity, sensitivity, outlier protection, and their relationship to decision making [69]. They conclude that there is no single metric *for model comparison* that can be considered the best in all situations and that they should be selected based on the kind of data set.

We use some of the arguments presented in their work to identify metrics suitable to the viscosity data set. See section 3.1 for a discussion about the characteristics of the viscosity data set used in this study. First, we look at the compatibility of metrics to a data set that spans many orders of magnitude. All metrics that have units i.e. are not scaled, tend to be dominated by the error from the highest order of magnitude and hence do not give information about the contributions of the errors from low orders of magnitude [69]. Metrics based on scaled error like mean absolute percentage error (MAPE) are more suited to such a situation. Next, we look at the level of outlier protection of various metrics. All metrics that take an average of individual errors suffer from outlier problem because the mean itself is sensitive to large outliers. Median based metrics like MedAE are better suited to such a situation. However median based metrics are not sensitive to small changes in the errors and also do not have clearly defined gradients with respect to model parameters. Finally, we look at metrics that can capture systematic biases (over or underestimation) in the ML models. Metrics based on error function with strictly positive range like squared error (SE), absolute error (AE), absolute percentage error (APE), etc cannot distinguish between systematic over or under prediction by the ML models. Metrics based on mean error (ME) or mean percentage error (MPE) can be used to gauge the bias in the models.

Therefore, we rank the ML models developed in this work based on the following metrics—MSE, MAE, MAPE, MedSE, MedAE, MedAPE, ME, MPE, MedE, MedPE, and R^2 (coefficient of determination).

3. Computational methods

3.1. Data

Viscosity is one of the few properties that can span many orders of magnitude (>10), depending on the complexity of the system and the thermodynamic conditions. In this work, we restrict ourselves to studying systems with simple interaction parameters (LJ only). This has the twin advantage that the liquid part of the phase diagram is well understood and also being a simple fluid, the viscosity computation is relatively easy. However, even for such simple systems, a *consistent* data set with a large number (several thousand) of systems is not yet available in the literature. In the absence of a coherent data set, smaller data from multiple sources is generally collated to build a larger data set [50, 99]. However, due to the sensitivity of viscosity (especially the confidence interval) to the ancillary MD parameters, this procedure can result in unreliable models.

3.1.1. Vlugt data set

Recently, Vlugt and co-workers simulated 250 binary LJ fluids to study the system size dependence of the self-diffusion coefficients [84]. In order to test the analytical expression for the system size corrections to the self-diffusion coefficients, they also computed the viscosity using the Einstein–Helfand equation [14, 100, 101]. We found that their work reported the largest consistent data set that used multiple long independent runs to compute viscosity and crucially, its confidence interval. In view of these attributes, our ML models were built using this data set. Here we note that there exists a significant amount of viscosity data of pure LJ systems [99, 102–108], and some sparse data of binary LJ systems [109–111] from other works. This collated data can, in principle, be used as a training, testing or constraining set in conjunction with the Vlugt data set. However, we do not use such a collated data in the current work, as it is not entirely clear how viscosity data sets from different sources with different MD run time parameters (potential cut off, system size, time step, numerical accuracy, trajectory length, etc) can be handled.

The data set contains a total of 1061 points, all of them at the same temperature and pressure of 0.65 and 0.05, respectively. All the quantities are reported in dimensionless units with interaction parameters of the first component as the base units i.e. $\sigma_1 = 1$, $\epsilon_1 = 1$, $\text{mass}_1 = 1$. The state space is spanned by varying three interaction parameters (σ_2, ϵ_2, k_{12}) and one compositional parameter (X_1). We call these parameters as 'preMD' features to be consistent with ML nomenclature where the independent variables are called as features. Further, each state point was studied at four different system sizes (quantified in terms of the total number of particles in the simulation box). In sum, about 250 state points were simulated, each at four different system sizes, giving a total of about 1000 data points. Unlike the self diffusion coefficient, shear viscosity does not have a strong dependence on system size [17, 20, 21, 42, 112–115]. Hence, we use only the data points at the system size of 2000 particles (273 out of 1000 data points) to develop the ML models in this work.

In the raw data, viscosity values span four decades, from 10^{-1} to 10^3, but only two data points had a value greater than 20. These two data points ($\eta_{\text{true}} > 20$) were identified as outliers and were not considered during the ML modeling. Figure 1 shows the distribution of data points by their viscosity values indicating that most of the data points are populated around the mean viscosity value of around 3 and the extremal decades are sparsely populated. Due to the uneven distribution of viscosity values across decades, the models trained subsequently can be biased towards values around the mean. The distribution of the standard error relative to the corresponding mean is shown in figure S2. The relative standard error seems to be uncorrelated to the viscosity value itself indicating that the data across decades is of similar quality. The standard error on the mean was used to calculate the irreducible minimum value of various loss functions [62] (also called as the Bayes error [63]). The irreducible errors are incurred by all non-probabilistic ML models because of their approximation of the conditional density $\mathbb{E}_y[y|\mathrm{x}]$ by point estimates [62]. The irreducible MSE is the average variance in the data. The irreducible MAE and MAPE were estimated by sampling from a Gaussian conditional density [62]. We also note that, in general, metric value obtained from the average standard error would be different from the irreducible loss. For example, the MAPE value obtained from the average standard error(%) of the data is about 2%, whereas the irreducible MAPE is 0.8%. Unless explicitly mentioned, the metric values obtained from the average standard errors are used to compare the corresponding metrics from the ML models. Hence, we consider ML models with MAPE metric lower than 2% to be successful models.

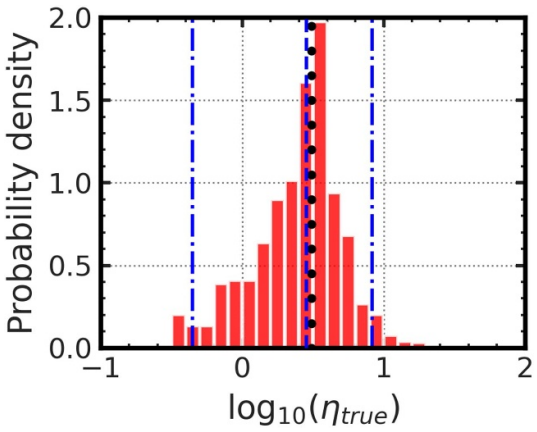

Figure 1. Data distribution: the distribution of the viscosity (η_{true}) from the Vlugt data set [84] across decades of viscosity. The blue vertical dashed line represents the median and the two blue vertical dash-dot lines represent the 95 percentile range around the median. The black dotted line represents the mean of the data.

Furthermore, to get a preliminary understanding about the underlying correlations in the data, viscosity is plotted against other features—X_1, σ_2, ϵ_2, k_{12}, box length, packing fraction (ζ), and self-diffusion coefficients (D1 and D2). These features can be divided into two sets—preMD and postMD features. As their names suggest, the preMD feature set consists of all those features that are fixed before running the MD simulation and postMD features are obtained only after the MD simulations. In this case, there are four preMD features—X_1, σ_2, ϵ_2, k_{12} and six postMD features—number density, ζ and the four preMD features. As expected, self-diffusion coefficients are inversely correlated to viscosity, consistent with the Stokes–Einstein relation. Apart from self-diffusion constant, only the packing fraction seems to be well correlated with viscosity, with higher packing fractions corresponding to higher viscosity. Rest of the plots show a wide spread of viscosity values at any given feature value indicating that no single feature can predict viscosity accurately. See section S3.1 for more details.

Two different sets of models, using postMD and preMD features, respectively, were developed for each ML algorithm. Unless otherwise mentioned, the results presented are from the models developed using postMD features. All the features were scaled using Min-Max scaler before training the ML models. Also, the logarithm of viscosity was used as the label. However, all the metrics presented in the subsequent sections were computed on the untransformed viscosity values.

The choice of the data transformations was made *before* the development of the ML models by using Exploratory Data Analysis techniques [116, 117]. The central idea is to transform the raw data to a distribution with known statistical properties (like normal distribution) [118]. Briefly, the label (shear viscosity) was transformed using a log-transform. This serves two purposes—(a) the transformed label is closer to a normal distribution (see figures 1 and S1) and (b) it enforces the non-negativity (of viscosity) constraint on the ML models. As the features were already sampled (by Vlugt *et al*) from a regular grid, only Min-Max scaling was applied to bring them into the interval [0,1].

3.1.2. Interpolation data set

In order to test the predictive performance of the ML models away from the Vlugt data grid, a complementary data set called the interpolation data set was created. As the name suggests, the data set is created in the interpolation region of the preMD feature space of the Vlugt data set. The interpolation set consists of a total of 17 points at several interpolation distances. We note that the interpolation space is not entirely in the equilibrium liquid region of the binary LJ phase diagram at the thermodynamic conditions studied by Jamali *et al* [84]. Hence, it is difficult to generate a 'representative sample' of the interpolation space, which is required to obtain quantitative estimates of predictive performance. In this context, the current data set of 17 points (though small) can be used to understand the predictive performance in a qualitative sense. Details of the MD simulation procedure used to create the interpolation data set are given in the supporting information.

3.1.3. Applicability domain (AD)

Given that the interpolation space is not entirely in the liquid region, the ML models cannot be expected to perform well over the entire interpolation space, especially far away from the Vlugt data grid. One way to

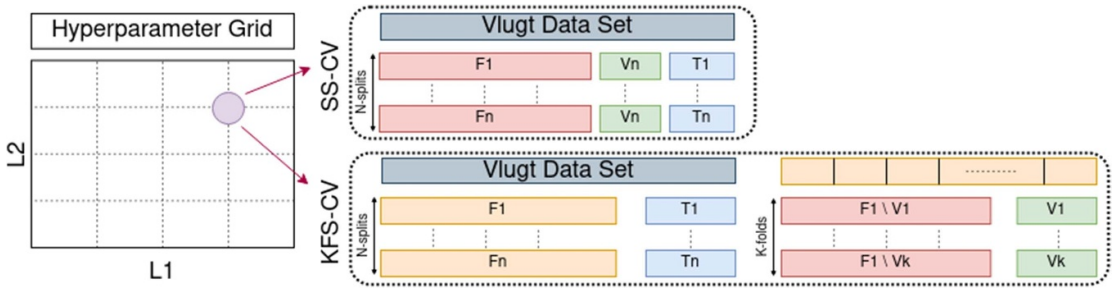

Figure 2. Model selection and performance estimation: a schematic representation of the two model selection procedures used in this study—shuffle split cross validation (SS-CV) and k-fold split cross validation (KFS-CV). L1, L2 represent the hyperparameters to be tuned in the model selection procedure. F_i, V_i, T_i represent the train, validation and test sets, respectively. See section S5.1 for more details.

tackle this issue is to define an AD within which the ML models are expected to perform well [119–123]. There are many methods to construct an AD and a detailed comparison is beyond the scope of this work [121–123]. The AD used in this work is described in section 4.3. The interpolation data set is divided into two parts called In-AD and Out-AD based on whether the points fall within or outside the AD, respectively.

3.2. ML models

A total of seven ML models were tested for their ability to predict shear viscosity—KRR, artificial NN (ANN), Gaussian process regression (GPR), SVR, RF, k-nearest neighbors (KNNs), and LASSO. In the current work, GPR is the probabilistic model (level 2 in section S1.1) and all others are non-probabilistic in nature (level 2 in section S1.1). Except ANN, all other the ML models used in this work were from the scikit-learn implementation [124]. The ANN models were built using the keras (part of tensorflow (v2.7.0)[125]) library in a python environment [126]. Many helper functions from numpy [127], scipy [128], pandas [129] and scikit-learn [124] were also used in the model construction, model selection and performance estimation steps.

3.3. Model selection and performance estimation

In this work, we compare two popular model selection and performance estimation methods called—shuffle split CV (SS-CV) and k-fold split CV (KFS-CV). A precise algorithmic description of these methods is given in section S4.1. Briefly, SS-CV splits the data set into three parts (named train/validation (val)/test) multiple times. Each time the data is shuffled and hence independent random realizations of the data can be obtained by SS-CV. KFS-CV is a two step procedure in which firstly entire data is split into two parts (named train/test) and later the train part is again split into k-folds of roughly same size. The k-folds (obtained in the second step) are used to obtain validation score and the test sets are used to do performance evaluation. Figure 2 summarizes the two procedures. Finally, we note that the procedures outlined above are referred to by slightly different names in the literature [52, 62, 65, 68] and the ML software packages [124]. Hence we recommend using the algorithmic description of these methods given in section S4.1.

3.4. Interpolation grid

The interpolation capabilities of the models can be qualitatively tested by plotting the predicted viscosity values at the grid of interpolated feature values. As the feature space is generally high dimensional (four in this case), only projections onto 1D/2D sub-spaces can be visualized. To keep the visualization uniform across the features, interpolation was done in the scaled feature space i.e. after the min-max scaler is applied. The Vlugt data set was generated at discrete values of each feature—X_1: (0.1, 0.3, 0.5, 0.7, 0.9), σ_2: (1.0, 1.2, 1.4, 1.6), ϵ_2: (1.0, 0.8, 0.6, 0.5), k_{12}: (0.05, 0.0, −0.3, −0.6). However, the final data set consists of only 250 unique combinations of preMD features as opposed to 320, had all the combinations been studied.

We have constructed four different interpolation grids to test the interpolation capability across each feature individually. The interpolation grid for a particular feature is generated at values uniformly spaced in the range of 0–1 while holding all other features at the values from the Vlugt data set. For example, in order to generate the X_1 interpolation grid, 19 uniformly distributed values between 0–1 were used for X_1, while the values for σ_2, ϵ_2 and k_{12} were taken from their scaled values in the Vlugt data set. We also define the interpolation distance of any interpolated point as its Euclidean distance from the nearest training data point.

4. Results and discussion

In this section, we first present evidence related to aspects of model selection and performance evaluation that are pertinent to ML models at the low data regime. We then compare different ML models based on their performance metrics and interpolation behavior. We finally compare the predicted uncertainties of ensemble models and that of a probabilistic ML model (GPR).

4.1. Model selection and performance estimation

4.1.1. Understanding the hyperparameter optimization landscapes

In order to understand the optimization landscape of the hyperparameters, we use LASSO and KRR models. They were chosen because they have less than three hyperparameters and are hence conducive for the visualization of the optimization landscape in 2D plots. Moreover, both models have analytic solutions and are hence much faster when trained on small data sets. Both these models were also studied in the context of model selection (albeit on synthetic data sets) and hence allow a close comparison wherever possible [65].

In this section, we elucidate the dependence of the performance of these ML models on the particulars of the data splitting procedure, which is an essential step for model selection and performance evaluation. The entire data set is randomly split into three parts—train, validation (val) and test sets with 60/20/20 ratio. This procedure is repeated N_{split} times thereby creating multiple random realizations of the train, validation and test splits. These N_{split} train sets are used to train N_{split} ML models at each hyperparameter value. The trained models are then evaluated on their corresponding validation and test sets.

Figure 3(a) shows the *average* test MSE of the LASSO model across a wide hyperparameter (α) range. Clearly, α values less than 10^{-3} are suited for the viscosity data set. However, no single optimal hyperparameter can be selected as there is no discernible change in the MSE values for α less than 10^{-3}. A similar 'flat-minima' hyperparameter landscape was also observed by Beckner *et al* on an ionic liquid experimental viscosity data set [52]. Hence, strictly applying the common model selection criteria of selecting the hyperparameter with the best performance on the validation set (in this case, minimizing MSE) belies the flat-minima nature of hyperparameter landscape. Also, the wide confidence interval around the average test MSE indicates that there is significant variability in the performance (MSE) of the ML models across different data splits.

Interestingly, the wide scatter of points in figure 3(b) indicates that the *individual* test MSEs are not correlated to *individual* validation MSEs. Hence model selection criteria based on optimization of the performance of *individual* validation sets need not necessarily result in a good generalization performance. However, the average validation MSE (over N_{split} splits) is perfectly correlated to the average test MSE as shown in figure 3(c). Consequently, the data splitting procedures that reserve a single 'unseen' data set (often called as test set [56, 88]) for evaluating the generalization performance should be discouraged as they suffer from wide variability. We note that this problem is unique to small data sets and the variability decreases rapidly with increase in data set size [65].

The same procedure was applied to the KRR model to elucidate its hyperparameter optimization landscape. In addition to validation and test sets, the performance of the KRR models was also evaluated on a single Interpolation set containing 17 data points. Figure 4 shows the 2D contour plots of the average validation, test and interpolation MSE over a wide range of KRR hyperparameters α and γ. The landscapes of the average test and validation MSE are similar, consistent with the LASSO results (figure 4). The interpolation MSE landscape is slightly different from others near the minima while still retaining the overall features. Importantly, it is more rugged than the validation and test MSE landscapes, possibly because the same Interpolation set was used across all the N_{split} splits.

The wide scatter of points in figures 4(a)–(c) show the optimal hyperparameter selected by minimizing the *individual* validation, test and Interpolation MSE, respectively. This demonstrates that using a single realization of data split to model selection or performance evaluation can result in wide variability, again consistent with LASSO results. Cawley and Talbot used KRR on a small synthetic data set to demonstrate this issue of wide variability when single realizations of data are used [65]. Further, while the average MSE landscapes are smooth, those corresponding to individual realizations of the data splits are rugged as shown in figure 5. The validation and test landscapes of these individual splits also show significant differences. For example, figures 5(c) and (d) corresponds to validation and test landscapes of a randomly chosen realization and their optimal hyperparameters vary by more than seven orders of magnitude.

These results demonstrate that there is a wide variability in choosing the optimal hyperparameter values (model selection) and also in estimating the generalization performance (performance evaluation) of the ML models on small data sets. Hence, it is crucial to do both the model selection and performance evaluation tasks by training an ensemble of models on different random splits of the data set.

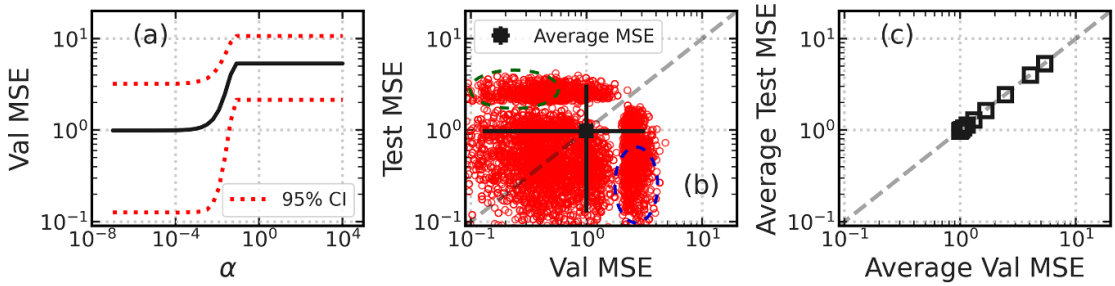

Figure 3. Model selection: (a) average validation MSE plotted against the α hyperparameter of the LASSO model. The red dotted lines show the 95% confidence interval of the validation MSE. (b) A scatter plot of the *individual* test MSE and validation MSE at $\alpha = 5 \times 10^{-6}$. The black square shows the test MSE vs validation MSE averaged over the 5000 splits. The error bars correspond to the 95 percentile about the median of the respective MSE. The green and the blue ellipses are marked to indicate the data splits in which test MSE can be much greater or lower than validation MSE, respectively. (c) Average test vs Average validation MSE at different values of the α hyperparameter of the LASSO model. The test and validation MSE were obtained using LASSO model and shuffle-split data sampling scheme (see section 3).

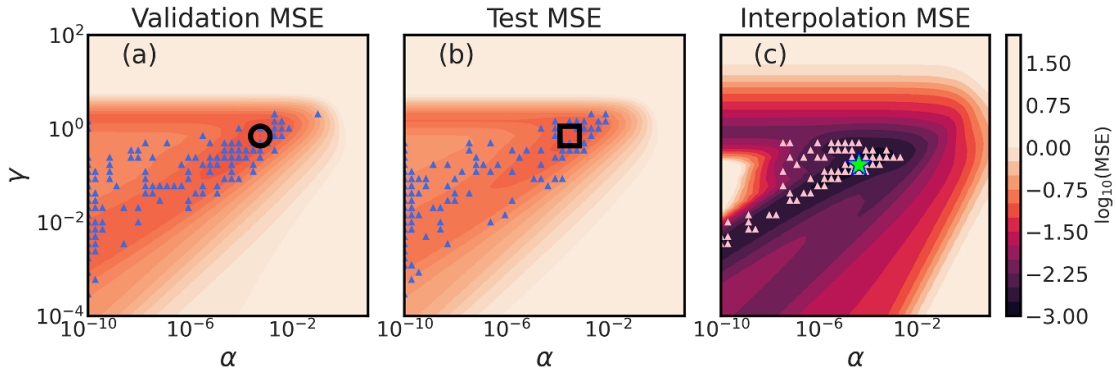

Figure 4. Model selection: 2D contour plots of (a) average validation MSE, (b) average test MSE and (c) average interpolation MSE of KRR model. The minima in the 2D contour plots are marked as follows—black-unfilled-circle: average validation MSE, black-unfilled-square: average test MSE, green-filled-star: average interpolation MSE. The blue triangles in (a) and (b) mark the optimal hyperparameters that minimize *individual* validation and test MSE, respectively. The pink triangles in (c) mark the optimal hyperparameters that minimize *individual* interpolation MSE. The validation, test, and interpolation MSE were obtained using KRR model and shuffle-split data sampling scheme (see section 3).

4.1.2. Comparison of CV procedures

The model selection criterion is based on the *average* validation score obtained from the CV procedures. We observe that the *average* validation score landscapes are almost identical in both SS-CV and KFS-CV irrespective of the kind of metric used to construct the landscape (figure S8). However, the two procedures differ in the variance of the validation landscape, with KFS-CV yielding a lower variance than SS-CV (figure S9). Cawley and Talbot show that the estimators with lower variance can do a better job of selecting the optimal hyperparameters [65]. Hence we use KFS-CV to do model selection and performance evaluation on rest of the ML models—SVR, RF, KNN, ANN.

Also, in both the CV procedures, the variance of the validation landscapes was strongly dependent on the error metric used to construct the landscapes. In general, we observe that MSE shows the highest variance followed by MAE, MAPE and R^2. The variance in MSE validation landscape is often so high (>100%), that unambiguous selection of optimal hyperparameters is difficult. Hence, the use of other metrics that have lesser variance such as MAE, MAPE, R^2 can help rectify the issue. In this work, we use the MAE validation landscape to choose the optimal hyperparameter values. Tables in section S5.1.2 list the optimal hyperparameters and the corresponding values of metrics for all the ML models studied in this work.

Finally, the performance estimation is done by evaluating the trained models (with the optimal hyperparameters) on the test sets. As both CV procedures result in nearly identical performance scores, we use the KFS-CV method to do the performance estimation for the other ML models—SVR, RF, KNN, ANN. In the following section, we compare the performance of various ML methods and rank them using multiple metrics.

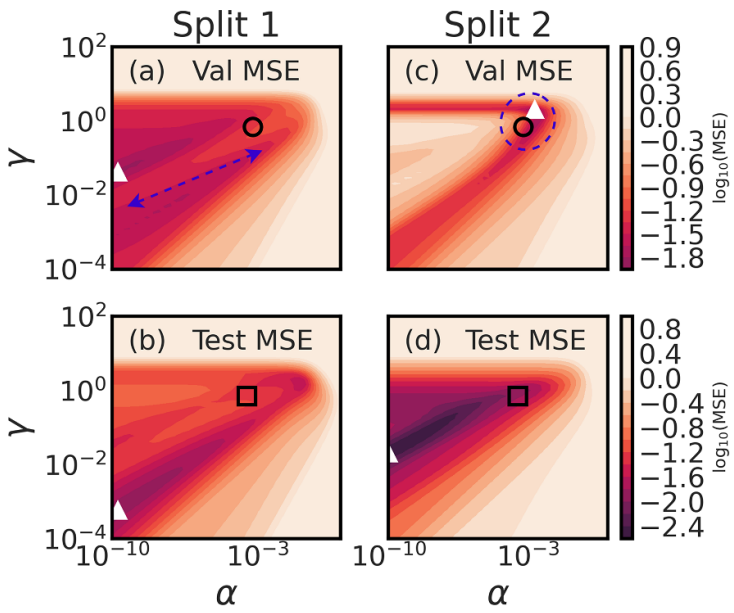

Figure 5. Model selection: (a) and (b) show 2D contour plots of the *individual* validation and test MSE, respectively, of a randomly chosen realization (called split 1) of the train-val-test split. (c) and (d) show the same for another random realization (called split 2) of the train-val-test split. In all panels, the minima of the average MSE are marked as follows: black-unfilled-circle: average validation MSE, black-unfilled-square: average test MSE. The minima of these particular realizations are marked with a white triangle symbol. The blue dashed arrow in (a) is drawn as a guide to the eye to highlight the large difference between the individual (white triangle) and the average (black circle) minima. The blue dashed ellipse in (c) is drawn as a guide to the eye to indicate that the individual and the average minima can be close by chance. The validation and test were obtained using KRR model and shuffle-split data sampling scheme (see section 3).

4.2. Model comparison and ranking

The predicted viscosity values from all the models, except KNN and LASSO, agree well with the true viscosity values both for test and interpolation data sets as shown in figure 6. The agreement is seen to be good across decades of viscosity values indicating, at least to the naked eye (figure 6(a)), that models do not bias any particular decade of viscosity values. A detailed discussion on the model bias is presented in the supporting information (section S5.2). Figure 6(c) compares the MAE and MAPE of all the models evaluated using the KFS-CV performance estimation procedure (section 3.3). The MAPE of KRR, GPR, ANN and SVR models are below the average standard error (%) of the data (called as threshold henceforth) and hence can be considered as successful models. On the other hand, KNN and LASSO models have both their test and train MAPE much above the threshold and can hence be considered as unsuccessful models. While the RF model performs well, it is still considered unsuccessful, due to its peculiar interpolation behavior (see section 4). Also, the successful models outperform the empirical model developed by Bell *et al* and Lautenschlaeger and Hasse for pure LJ fluids (see section S4.2) [99, 108].

The MAE and MAPE of the successful models—i.e. KRR, GPR, ANN, and SVR—are very close to each other as seen in figure 6(c) and hence more information is required to unambiguously rank them. In figure 7, we show the relative ranks of all the models based on seven (MAE, MSE, MAPE, MedAE, MedSE, MedAPE and R^2) performance based metrics and four (ME, MPE, MedE, MedPE) bias based metrics. The mean, median values of the AE, SE, and the APE along with the coefficient of determination (R^2) are the performance metrics, while mean, median error (E) and the percentage error (PE) constitute the bias metrics. An average rank (averaging done across the metrics with uniform weights) is also shown for each model. The average rank follows closely the MAE rank with four successful models—KRR, GPR, SVR, and ANN—having an average rank less than four and three unsuccessful models having a rank greater than four. According to the average rank, GPR is the best performing model, followed closely by SVR and KRR. These three models are followed by ANN, RF, KNN, and LASSO, respectively, with the last two consistently ranked sixth and seventh. Interestingly, there is considerable mixing of ranks based on MAE vs MSE, indicating that a holistic approach using a combination of metrics needs to be used to objectively evaluate the models. Another noticeable trend is the disconnect between the ranking based on performance and those based on bias metrics. These findings highlight the need to evaluate models based on metrics beyond the simple loss functions used to train the models themselves to get a complete picture of the models accuracy, bias and generalizability. Now that the models have been validated for performance and bias, we discuss their interpolation capabilities in the next section.

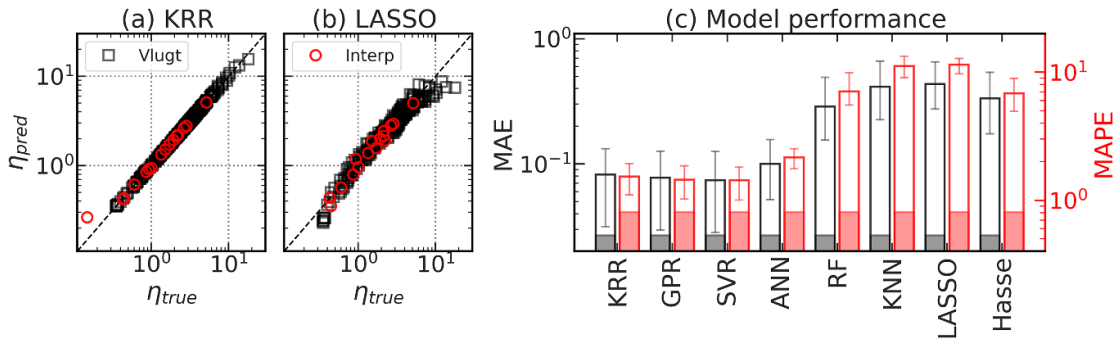

Figure 6. Model performance: (a), (b) predicted vs the true viscosity values from KRR and LASSO ensemble models, respectively. The black squares are for the Vlugt data set and the red circles are for the interpolation data generated in this work. The black dashed line represents the $y = x$ line. (c) Comparison of the mean absolute error (MAE) and mean absolute percentage error (MAPE) of all the seven ML models studied in this work. The performance of the empirical correlation by Lautenschlaeger and Hasse [108] is also shown for comparison. The black bars show the MAE and the red bars show the MAPE. The error bars indicate the 95% confidence interval region. The filled regions of the bars indicate the irreducible errors (0.027 for MAE and 0.8% for MAPE) in the Vlugt data set (section 3.1.2). See the section 3 for model description and performance estimation.

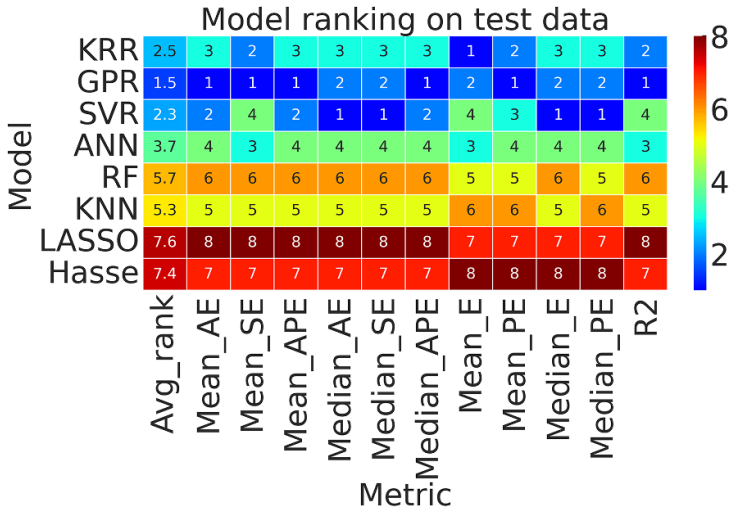

Figure 7. Model ranking: grid plots of model's ranking on the entire Vlugt data based on various metrics. The text and color of each square represent the relative ranks of models. Each row shows the ranks of the model labeled on the y-axis and each column corresponds to the metric based on which the ranking was done. The metric values are computed using the ensemble predictions of the ML models on the entire Jamali data [84]. Lower model rank is better.

4.2.1. Interpolation behavior

Figure 8 shows the predicted viscosity values from KRR, ANN, SVR, RF, KNN, and LASSO models at the interpolation points plotted against X_1 feature values. The color of the points (represented as dashes in the figure) indicates the distance from the nearest training data point that the models have seen, with darker shades being farther away. The viscosity values from the Vlugt data set are also shown (as red diamond symbols) for comparison. KRR, SVR, and ANN models show a smooth variation as the feature values move farther away from their corresponding values in the Vlugt data set. On the other hand, RF and KNN models show sudden discontinuities in the predicted viscosity values at some specific X_1 values. These discontinuities are probably due to the presence of decision boundaries in RF and a sudden change of nearest neighbors in the case of KNN. Such sudden discontinuities are incompatible with viscosity which is expected to be continuous (at least as long as there is no phase transition).

4.3. Uncertainty quantification

As demonstrated in the previous sections, the performance of ML models trained on small data sets can have wide variations. In this regard, models that can estimate the uncertainty on individual predictions can help alleviate this issue. The uncertainty estimates can be used as a guide to end-users about the reliability of a given prediction and thus of the models. More generally, uncertainty quantification has many other

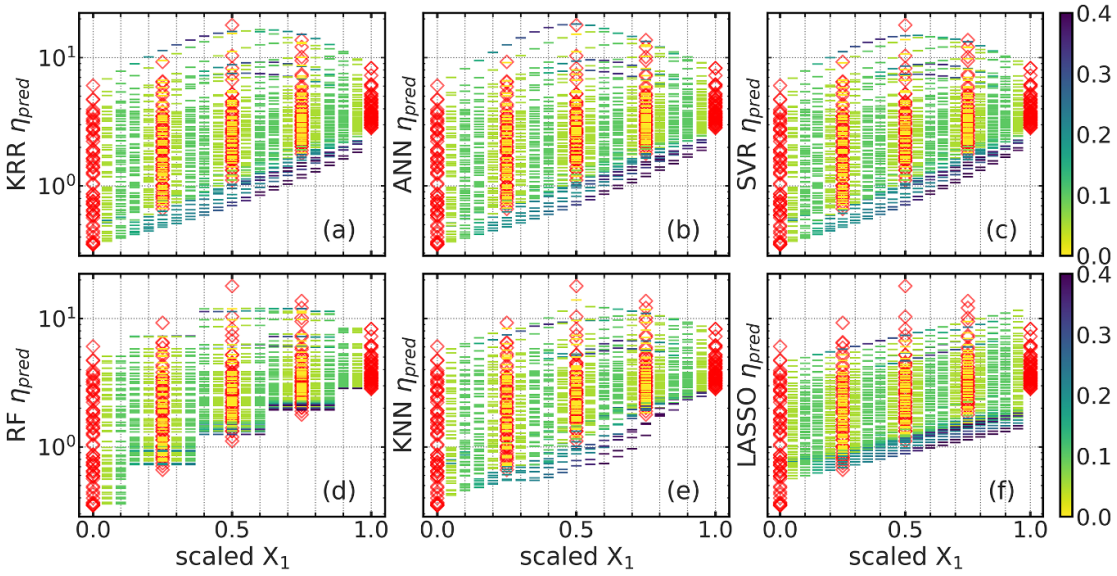

Figure 8. Model interpolation: (a)–(f) show the predicted viscosity values from the ensembles of KRR, ANN, SVR, RF, KNN, and LASSO, respectively, in the X_1 interpolation range. Red diamonds are viscosity values from the Vlugt data set. The color of the dashes indicate the distance from the nearest Vlugt data point in the scaled feature space (abscissa). See section 3 for details on feature scaling and the construction of the interpolation grid.

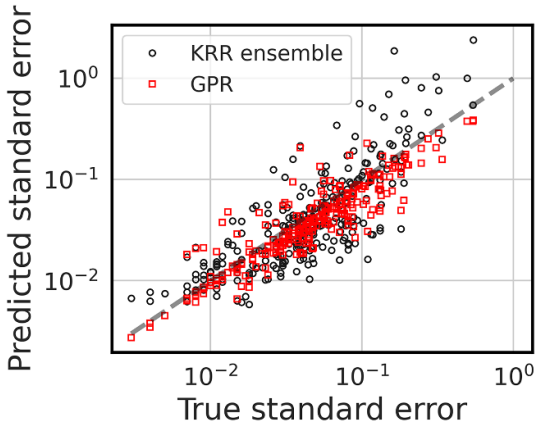

Figure 9. Uncertainty comparison: the standard errors of the predicted viscosity values plotted against their corresponding true standard errors. The standard errors from the ensemble of KRR models are shown as black circles and those from the GPR model are shown as red squares. The grey dashed line corresponds to $y = x$ and is drawn as a guide to the eye.

applications [71, 73–75, 130] such as—setting the AD of the ML models [121], active learning for generating data on the fly [131, 132], etc.

In this work, we used two approaches to estimate the uncertainty—probabilistic ML method and ensemble ML method. The probabilistic ML methods inherently capture the uncertainty through their model architecture, whereas the ensemble ML approach uses an ensemble of ML models on several random realizations of the data. GPR was the choice of probabilistic ML method due to its simplicity (few hyperparameters), wide applicability and near universality. KRR was used to test the ensemble approach again due to its simplicity (two hyperparameters), training speed (analytical solution) and wide applicability.

Figure 9 compares the standard error estimated from the ML models and the true standard error of the data. Both GPR and ensemble KRR methods show good agreement with the true data. The standard errors predicted by GPR show a slightly better agreement with that of the data than KRR ensemble does, especially at high error values. This is because the ensemble methods capture the uncertainty in the data indirectly by repeated sampling from the training data set. On the other hand, the standard error values of the data are directly fed into the GPR training. A similar observation was made by Schroeter *et al* on their comparison of GPR and ensemble methods to predict error bars on the solubility data [121].

Information about closeness of a new query point to the training data would be useful to decide whether or not to trust the values predicted by the model. This information can be naturally encoded into GPR in the

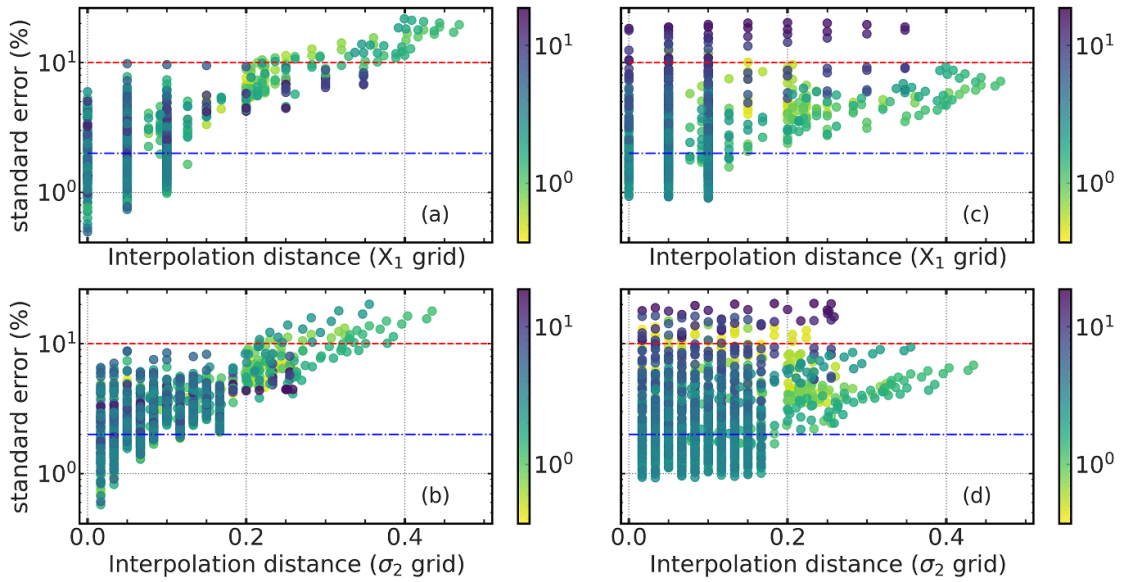

Figure 10. Interpolation uncertainty: (a) and (b) show the GPR predicted relative standard error (%) plotted against the interpolation distance in the X_1, σ_2 interpolation grids, respectively. (c) and (d) show the KRR ensemble predicted relative standard error (%) plotted against the interpolation distance in the X_1, σ_2 interpolation grids, respectively. The colors indicate the predicted mean viscosity value corresponding to the interpolation point. The blue horizontal dash-dot line indicates the average standard error (%) (2%) of the Vlugt data set and the red horizontal dashed line indicates 10% standard error (%) which is then used to define a qualitative cut-off for the applicability domain. The interpolation distance is computed in the scaled feature space.

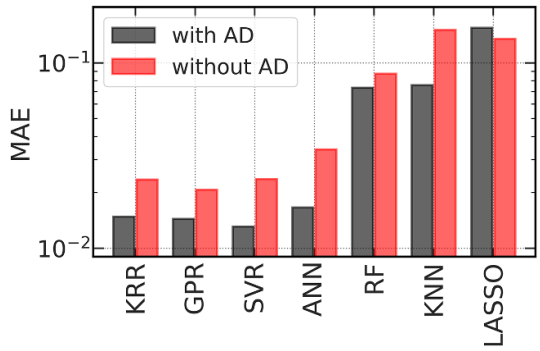

Figure 11. Applicability domain (AD): MAE values of the ensemble ML models estimated on the interpolation data set. The red bars indicate the MAE values on the entire interpolation data (without applicability domain) and the black bars indicate the MAE values on the part of the interpolation data that falls within the applicability domain (In-AD). Due to the small size of interpolation data set, these MAE values should be compared in a qualitative sense.

form of epistemic uncertainty [131]. Ideally, the predicted standard error should increase beyond the natural uncertainty in the data as the query point moves away from the training data set. Figures 10(a) and (b) show the predicted relative standard error from the GPR model plotted against the interpolation distance with the colors indicating the predicted mean viscosity value. The predicted relative standard errors clearly increase with increasing interpolation distance. However, this behavior could not be seen in the case of relative standard errors estimated by the ensemble method as seen in figures 10(c) and (d). Hence, the uncertainties estimated by GPR represent the true uncertainties better and also systematically increase when the query points move far from the training set. Additionally, GPR needs to be trained only once when compared to ensemble methods which need to be trained over multiple realizations of the training set.

Finally, the standard errors predicted by the GPR can be used to construct AD of the ML models [121]. For example, for queries which have a distance less than 0.2 (in the scaled feature space) from the nearest training point, a relative error of less than 10% can be expected (figure 10). Hence, the scaled distance of 0.2 can be set as a limit for the AD, beyond which the predictions from the ML models need to be treated with caution. Though such a distance based approach is simple, it can be applied to any ML model thereby justifying its use [121]. Also, such distance based AD methods have been successfully implemented for QSAR [119] and ML models [121].

The interpolation data set was split into In-AD and Out-AD based on whether the points fell within the AD or otherwise. The performance of all the ML models on the In-AD data set is much better than that on Out-AD set, demonstrating that the application of AD can be used to detect and remove the outliers (figure 11). Also, the improvements from the AD (though constructed from GPR only) were observed across all the ML models and performance metrics.

5. Conclusions

In this work, we trained and evaluated several successful ML models to predict the shear viscosity of binary LJ fluids. Being a collective property, shear viscosity is expensive to predict from equilibrium atomistic MD simulations and hence only small data sets can be found in the literature. The major challenges posed by such small data sets to build ML methods on are discussed. Specifically, we focus on—(a) model selection and performance estimation, (b) performance metrics, and (c) uncertainty quantification.

ML models are prone to overfitting on small data sets at both the model parameters and hyperparameter levels. We discuss various model selection methods—k-fold CV, nested k-fold CV, Monte Carlo CV, etc—that are generally used to address this issue. While these methods are commonly used for selecting hyperparameters, the generalization error (performance evaluation) is estimated on a single unseen data set (test set). We demonstrate that such estimates are prone to wide variability because of the small data set size. The hyperparameter optimization landscapes of LASSO, KRR models are shown to have 'flat-minima' thereby making it difficult to unambiguously select the optimal hyperparameters. We compared two simple CV procedures—SS-CV and KFS-CV and found that while their ME estimates were almost identical, KFS-CV showed a lower variance. Hence, it was chosen to both the model selection and performance estimation tasks *simultaneously*.

We discuss the role of performance metrics in model training and model evaluation, both from theoretical and empirical standpoints. We compare several commonly used metrics like MSE, MAE, MAPE and R^2 and discuss their relevance to the viscosity data set. We propose a holistic model ranking procedure based on inputs from multiple complementary metrics. The interpolation behavior of the ML models are compared qualitatively. While the KRR, ANN and SVR models showed smooth interpolation behavior, RF and KNN models showed sudden discontinuities and are hence considered unsuccessful. The successful models are also shown to outperform the best-in-class empirical model of Lautenschlaeger and Hasse [108] in the prediction of shear viscosity of binary LJ fluids.

We present two methods to estimate uncertainty in individual predictions from the ML models—(a) GPR and (b) ensemble of KRR ML models. The uncertainty (in terms of standard error) estimated by the methods showed overall agreement with the true uncertainty of the data, with GPR faring slightly better. The behavior of the estimated uncertainty by both the methods in the interpolation feature range is also compared. The GPR's uncertainty steadily increased as the query data points moved away from the training data, while no discernible pattern could be identified in the uncertainty from the ensemble method. The relative standard error estimated from the GPR model can be used to set distance limits for the query points, thereby defining the AD in which the results from the ML models are reliable. We found that the points of the interpolation set that fell within the AD were better estimated by the ML models than the ones that fell outside the AD, thereby demonstrating the utility of AD. Finally, the principles discussed in this work can be applied to develop ML models of viscosity for complex fluids. However, in such fluids, the identification of the features that are most relevant to shear viscosity would also be non-trivial and constitutes a part of our future work [133]. Also, incorporation of physical constraints (for example asymptotic behavior of viscosity in the zero density regime [99, 134]) into the architecture of the ML models can improve the robustness of the models [135] and will be explored in our future works.

Data availability statement

The data that supports the findings of this study are available within the article and its supplementary material.

The data that support the findings of this study will be openly available following an embargo at the following URL/DOI: 10.5281/zenodo.7043243. Data will be available from 1 December 2022.

Acknowledgments

The authors thank the Department of Science and Technology, India, for support. This work is a part of National Supercomputing Mission (NSM) project entitled 'Molecular Materials and Complex Fluids of Societal Relevance: HPC and AI to the Rescue' (Grant No. DST/NSM/ R&D_HPC_Applications/2021/05).

The support and the resources provided by 'PARAM Yukti Facility' under the National Supercomputing Mission, Government of India, at the Jawaharlal Nehru Centre For Advanced Scientific Research are gratefully acknowledged.

Conflict of interest

The authors have no conflicts to disclose.

Author contributions

Nikhil V S Avula Formal Analysis (lead); Methodology (lead); Software (lead); Visualization (lead); Writing/Original Draft Preparation (lead); Writing/Review & Editing (equal); Validation (equal); Data Curation (equal); Conceptualization (supporting); Funding Acquisition (supporting); **Shivanand K Veesam** Data Curation (equal); Software (supporting); Validation (equal); **Sudarshan Behera** Formal Analysis (supporting); Methodology (supporting); Software (supporting); Writing/Review & Editing (supporting); **Sundaram Balasubramanian** Conceptualization (lead); Funding Acquisition (lead); Project Administration (lead); Resources (lead); Supervision (lead); Writing/Review & Editing (equal).

Appendix. Performance metrics

$$e_i = y_i^{\text{pred}} - y_i^{\text{true}} \tag{A1}$$

$$r_i = (y_i^{\text{pred}} - y_i^{\text{true}})/y_i^{\text{true}} \tag{A2}$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i^{\text{true}} \tag{A3}$$

$$\text{ME} = \frac{1}{N}\sum_{i=1}^{N} e_i \tag{A4}$$

$$\text{MPE} = \frac{1}{N}\sum_{i=1}^{N} 100 \times r_i \tag{A5}$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} e_i^2 \tag{A6}$$

$$\text{MedSE} = \text{median}(e_i^2) \tag{A7}$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} |e_i| \tag{A8}$$

$$\text{MedAE} = \text{median}(|e_i|) \tag{A9}$$

$$\text{MAPE} = \frac{1}{N}\sum_{i=1}^{N} 100 \times |r_i| \tag{A10}$$

$$\text{MedAPE} = 100 \times \text{median}(|r_i|) \tag{A11}$$

$$R^2 = 1 - \sum_{i=1}^{N} \frac{|e_i|^2}{|y_i^{\text{true}} - \bar{y}|^2} \tag{A12}$$

ORCID iDs

Nikhil V S Avula https://orcid.org/0000-0001-6467-6040
Shivanand Kumar Veesam https://orcid.org/0000-0003-4671-9646
Sudarshan Behera https://orcid.org/0000-0003-1025-0639
Sundaram Balasubramanian https://orcid.org/0000-0002-3355-6764

References

[1] March N and Tosi M 2002 *Introduction to Liquid State Physics* (Singapore: World Scientific)
[2] Levashov V A, Morris J R and Egami T 2011 Viscosity, shear waves and atomic-level stress-stress correlations *Phys. Rev. Lett.* **106** 115703
[3] Giordano D, Russell J K and Dingwell D B 2008 Viscosity of magmatic liquids: a model *Earth Planet. Sci. Lett.* **271** 123–34
[4] de Wijs G A, Kresse G, Vočadlo L, Dobson D, Alfè D, Gillan M J and Price G D 1998 The viscosity of liquid iron at the physical conditions of the Earth's core *Nature* **392** 805–7
[5] Vočadlo L 2007 2.05 mineralogy of the Earth—the Earth's core: iron and iron alloys *Treatise on Geophysics* ed G Schubert (Amsterdam: Elsevier) pp 91–120
[6] Secco R A 1995 Viscosity of the outer core *Mineral Physics & Crystallography* (Washington, DC: American Geophysical Union) pp 218–26
[7] Kawasaki T and Kim K 2017 Identifying time scales for violation/preservation of Stokes–Einstein relation in supercooled water *Sci. Adv.* **3** e1700399
[8] Bell I H 2019 Probing the link between residual entropy and viscosity of molecular fluids and model potentials *Proc. Natl Acad. Sci.* **116** 4070–9
[9] Bell I H, Dyre J C and Ingebrigtsen T S 2020 Excess-entropy scaling in supercooled binary mixtures *Nat. Commun.* **11** 4300
[10] Bell I H, Delage-Santacreu S, Hoang H and Galliero G 2021 Dynamic crossover in fluids: from hard spheres to molecules *J. Phys. Chem. Lett.* **12** 6411–7
[11] Rizk F, Gelin S, Biance A-L and Joly L 2022 Microscopic origins of the viscosity of a Lennard-Jones liquid *Phys. Rev. Lett.* **129** 074503
[12] Baled H O, Gamwo I K, Enick R M and McHugh M A 2018 Viscosity models for pure hydrocarbons at extreme conditions: a review and comparative study *Fuel* **218** 89–111
[13] Kontogeorgis G M, Dohrn R, Economou I G, de Hemptinne J-C, ten Kate A, Kuitunen S, Mooijer M, Žilnik L F and Vesovic V 2021 Industrial requirements for thermodynamic and transport properties: 2020 *Ind. Eng. Chem. Res.* **60** 4987–5013
[14] Maginn E J, Messerly R A, Carlson D J, Roe D R and Elliott J R 2019 Best practices for computing transport properties 1. Self-diffusivity and viscosity from equilibrium molecular dynamics *Living J. Comput. Mol. Sci.* **1** 6324
[15] Hess B 2002 Determining the shear viscosity of model liquids from molecular dynamics simulations *J. Chem. Phys.* **116** 209–17
[16] Alfè D and Gillan M J 1998 First-principles calculation of transport coefficients *Phys. Rev. Lett.* **81** 5161–4
[17] Jamali S H, Hartkamp R, Bardas C, Söhl J, Vlugt T J H and Moultos O A 2018 Shear viscosity computed from the finite-size effects of self-diffusivity in equilibrium molecular dynamics *J. Chem. Theory Comput.* **14** 5959–68
[18] Li Q, Sun T, Zhang Y-G, Xian J-W and Vočadlo L 2021 Atomic transport properties of liquid iron at conditions of planetary cores *J. Chem. Phys.* **155** 194505
[19] Malosso C, Zhang L, Car R, Baroni S and Tisi D 2022 Viscosity in water from first-principles and deep-neural-network simulations *npj Comput. Mater.* **8** 139
[20] Tazi S, Boţan A, Salanne M, Marry V, Turq P and Rotenberg B 2012 Diffusion coefficient and shear viscosity of rigid water models *J. Phys.: Condens. Matter* **24** 284117
[21] Wang H, DeFever R S, Zhang Y, Wu F, Roy S, Bryantsev V S, Margulis C J and Maginn E J 2020 Comparison of fixed charge and polarizable models for predicting the structural, thermodynamic and transport properties of molten alkali chlorides *J. Chem. Phys.* **153** 214502
[22] Fedosov D A, Pan W, Caswell B, Gompper G and Karniadakis G E 2011 Predicting human blood viscosity in silico *Proc. Natl Acad. Sci.* **108** 11772–7
[23] Zhang Y, Otani A and Maginn E J 2015 Reliable viscosity calculation from equilibrium molecular dynamics simulations: a time decomposition method *J. Chem. Theory Comput.* **11** 3537–46
[24] Müller-Plathe F 1999 Reversing the perturbation in nonequilibrium molecular dynamics: an easy way to calculate the shear viscosity of fluids *Phys. Rev.* E **59** 4894–8
[25] Ewen J P, Heyes D M and Dini D 2018 Advances in nonequilibrium molecular dynamics simulations of lubricants and additives *Friction* **6** 349–86
[26] Heyes D M, Dini D and Smith E R 2018 Incremental viscosity by non-equilibrium molecular dynamics and the Eyring model *J. Chem. Phys.* **148** 194506
[27] Stillinger F H and Debenedetti P G 2005 Alternative view of self-diffusion and shear viscosity *J. Phys. Chem.* B **109** 6604–9
[28] Jones R E and Mandadapu K K 2012 Adaptive Green-Kubo estimates of transport coefficients from molecular dynamics based on robust error analysis *J. Chem. Phys.* **136** 154102

[29] Kim C, Borodin O and Karniadakis G E 2015 Quantification of sampling uncertainty for molecular dynamics simulation: time-dependent diffusion coefficient in simple fluids *J. Comput. Phys.* **302** 485–508
[30] Oliveira L de S and Greaney P A 2017 Method to manage integration error in the Green-Kubo method *Phys. Rev.* E **95** 023308
[31] Heyes D M, Smith E R and Dini D 2019 Shear stress relaxation and diffusion in simple liquids by molecular dynamics simulations: analytic expressions and paths to viscosity *J. Chem. Phys.* **150** 174504
[32] Heyes D M, Dini D and Smith E R 2020 Single trajectory transport coefficients and the energy landscape by molecular dynamics simulations *J. Chem. Phys.* **152** 194504
[33] Heyes D M, Dini D and Smith E R 2021 Viscuit and the fluctuation theorem investigation of shear viscosity by molecular dynamics simulations: the information and the noise *J. Chem. Phys.* **154** 074503
[34] Heyes D M and Dini D 2022 Intrinsic viscuit probability distribution functions for transport coefficients of liquids and solids *J. Chem. Phys.* **156** 124501
[35] Avula N V S, Karmakar A, Kumar R and Balasubramanian S 2021 Efficient parametrization of force field for the quantitative prediction of the physical properties of ionic liquid electrolytes *J. Chem. Theory Comput.* **17** 4274–90
[36] Kondratyuk N D and Pisarev V V 2021 Predicting shear viscosity of 1,1-diphenylethane at high pressures by molecular dynamics methods *Fluid Phase Equilib.* **544–545** 113100
[37] Goloviznina K, Gong Z, Costa Gomes M F and Pádua Aílio A H 2021 Extension of the CL&Pol polarizable force field to electrolytes, protic ionic liquids and deep eutectic solvents *J. Chem. Theory Comput.* **17** 1606–17
[38] Gong Z and Sun H 2019 Extension of team force-field database to ionic liquids *J. Chem. Eng. Data* **64** 3718–30
[39] Nieto-Draghi C, Ungerer P and Rousseau B 2006 Optimization of the anisotropic united atoms intermolecular potential for *n*-alkanes: improvement of transport properties *J. Chem. Phys.* **125** 044517
[40] Kondratyuk N D 2019 Comparing different force fields by viscosity prediction for branched alkane at 0.1 and 400 MPA *J. Phys.: Conf. Ser.* **1385** 012048
[41] Hamani A W S, Bazile J-P, Hoang H, Luc H T, Daridon J-L and Galliero G 2020 Thermophysical properties of simple molecular liquid mixtures: on the limitations of some force fields *J. Mol. Liq.* **303** 112663
[42] Kim K-S, Han M H, Kim C, Li Z, Karniadakis G E and Lee E K 2018 Nature of intrinsic uncertainties in equilibrium molecular dynamics estimation of shear viscosity for simple and complex fluids *J. Chem. Phys.* **149** 044510
[43] Wang Y, Lamim Ribeiro J M and Tiwary P 2020 Machine learning approaches for analyzing and enhancing molecular dynamics simulations *Curr. Opin. Struct. Biol.* **61** 139–45
[44] Noé F, Tkatchenko A, Müller K-R and Clementi C 2020 Machine learning for molecular simulation *Annu. Rev. Phys. Chem.* **71** 361–90
[45] Miksch A M, Morawietz T, Kästner J, Urban A and Artrith N 2021 Strategies for the construction of machine-learning potentials for accurate and efficient atomic-scale simulations *Mach. Learn.: Sci. Technol.* **2** 031001
[46] Karthikeyan A and Priyakumar U D 2021 Artificial intelligence: machine learning for chemical sciences *J. Chem. Sci.* **134** 2
[47] Bonati L, Piccini G and Parrinello M 2021 Deep learning the slow modes for rare events sampling *Proc. Natl Acad. Sci.* **118** e2113533118
[48] Doerr S, Majewski M, Pérez A, Krämer A, Clementi C, Noe F, Giorgino T and De Fabritiis G 2021 Torchmd: a deep learning framework for molecular simulations *J. Chem. Theory Comput.* **17** 2355–63
[49] Winkler L, Müller K-R and Sauceda H E 2022 High-fidelity molecular dynamics trajectory reconstruction with bi-directional neural networks *Mach. Learn.: Sci. Technol.* **3** 025011
[50] Allers J P, Harvey J A, Garzon F H and Alam T M 2020 Machine learning prediction of self-diffusion in Lennard-Jones fluids *J. Chem. Phys.* **153** 034102
[51] Leverant C J, Harvey J A and Alam T M 2020 Machine learning-based upscaling of finite-size molecular dynamics diffusion simulations for binary fluids *J. Phys. Chem. Lett.* **11** 10375–81
[52] Beckner W, Mao C M and Pfaendtner J 2018 Statistical models are able to predict ionic liquid viscosity across a wide range of chemical functionalities and experimental conditions *Mol. Syst. Des. Eng.* **3** 253–63
[53] Koutsoukos S, Philippi F, Malaret F and Welton T 2021 A review on machine learning algorithms for the ionic liquid chemical space *Chem. Sci.* **12** 6820–43
[54] Valderrama J O, Muñoz J M and Rojas R E 2011 Viscosity of ionic liquids using the concept of mass connectivity and artificial neural networks *Korean J. Chem. Eng.* **28** 1451–7
[55] Dutt N V K, Ravikumar Y V L and Rani K Y 2013 Representation of ionic liquid viscosity-temperature data by generalized correlations and an artificial neural network (ANN) model *Chem. Eng. Commun.* **200** 1600–22
[56] Paduszyński K and Domańska U 2014 Viscosity of ionic liquids: an extensive database and a new group contribution model based on a feed-forward artificial neural network *J. Chem. Inf. Model.* **54** 1311–24
[57] Fatehi M-R, Raeissi S and Mowla D 2017 Estimation of viscosities of pure ionic liquids using an artificial neural network based on only structural characteristics *J. Mol. Liq.* **227** 309–17
[58] Baghban A, Kardani M N and Habibzadeh S 2017 Prediction viscosity of ionic liquids using a hybrid LSSVM and group contribution method *J. Mol. Liq.* **236** 452–64
[59] Datta R, Ramprasad R and Venkatram S 2022 Conductivity prediction model for ionic liquids using machine learning *J. Chem. Phys.* **156** 214505
[60] Duong D V, Tran H-V, Pathirannahalage S K, Brown S J, Hassett M, Yalcin D, Meftahi N, Christofferson A J, Greaves T L and Le T C 2022 Machine learning investigation of viscosity and ionic conductivity of protic ionic liquids in water mixtures *J. Chem. Phys.* **156** 154503
[61] Vishwakarma G, Sonpal A and Hachmann J 2021 Metrics for benchmarking and uncertainty quantification: quality, applicability and best practices for machine learning in chemistry *Trends Chem.* **3** 146–56
[62] Bishop C M 2006 *Pattern Recognition and Machine Learning* (New York: Springer)
[63] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
[64] Arlot S and Celisse A 2010 A survey of cross-validation procedures for model selection *Stat. Surv.* **4** 40–79
[65] Cawley G C and Talbot N L C 2010 On over-fitting in model selection and subsequent selection bias in performance evaluation *J. Mach. Learn. Res.* **11** 2079–107
[66] Zhang Y and Yang Y 2015 Cross-validation for selecting a model selection procedure *J. Econ.* **187** 95–112
[67] Burnham K and Anderson D 2003 *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach* (New York: Springer) (https://doi.org/10.1007/b97636)

[68] Xu Y and Goodacre R 2018 On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning *J. Anal. Test.* **2** 249–62
[69] Armstrong J and Collopy F 1992 Error measures for generalizing about forecasting methods: empirical comparisons *Int. J. Forecast.* **8** 69–80
[70] Gneiting T 2011 Making and evaluating point forecasts *J. Am. Stat. Assoc.* **106** 746–62
[71] Schwaighofer A, Schroeter T, Mika S, Laub J, ter Laak A, Sülzle D, Ganzer U, Heinrich N and Müller K-R 2007 Accurate solubility prediction with error bars for electrolytes: a machine learning approach *J. Chem. Inf. Model.* **47** 407–24
[72] Tran K, Neiswanger W, Yoon J, Zhang Q, Xing E and Ulissi Z W 2020 Methods for comparing uncertainty quantifications for material property predictions *Mach. Learn.: Sci. Technol.* **1** 025006
[73] Scalia G, Grambow C A, Pernici B, Li Y-P and Green W H 2020 Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction *J. Chem. Inf. Model.* **60** 2697–717
[74] Imbalzano G, Zhuang Y, Kapil V, Rossi K, Engel E A, Grasselli F and Ceriotti M 2021 Uncertainty estimation for molecular dynamics and sampling *J. Chem. Phys.* **154** 074102
[75] Tavazza F, DeCost B and Choudhary K 2021 Uncertainty prediction for machine learning models of material properties *ACS Omega* **6** 32431–40
[76] Stuke A, Rinke P and Todorović M 2021 Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization *Mach. Learn.: Sci. Technol.* **2** 035022
[77] Kolassa S 2020 Why the "best" point forecast depends on the error or accuracy measure *Int. J. Forecast.* **36** 208–11
[78] Makridakis S, Spiliotis E and Assimakopoulos V 2020 The M4 competition: 100,000 time series and 61 forecasting methods *Int. J. Forecast.* **36** 54–74
[79] Hirschfeld L, Swanson K, Yang K, Barzilay R and Coley C W 2020 Uncertainty quantification using neural networks for molecular property prediction *J. Chem. Inf. Model.* **60** 3770–80
[80] Wolpert D and Macready W 1997 No free lunch theorems for optimization *IEEE Trans. Evol. Comput.* **1** 67–82
[81] Hansen K, Montavon G, Biegler F, Fazli S, Rupp M, Scheffler M, von Lilienfeld O A, Tkatchenko A and Müller K-R 2013 Assessment and validation of machine learning methods for predicting molecular atomization energies *J. Chem. Theory Comput.* **9** 3404–19
[82] Ruddigkeit L, van Deursen R, Blum L C and Reymond J-L 2012 Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17 *J. Chem. Inf. Model.* **52** 2864–75
[83] Gupta A, Chakraborty S and Ramakrishnan R 2021 Revving up ^{13}C NMR shielding predictions across chemical space: benchmarks for atoms-in-molecules kernel machine learning with new data for 134 kilo molecules *Mach. Learn.: Sci. Technol.* **2** 035010
[84] Jamali S H, Wolff L, Becker T M, Bardow A, Vlugt T J H and Moultos O A 2018 Finite-size effects of binary mutual diffusion coefficients from molecular dynamics *J. Chem. Theory Comput.* **14** 2667–77
[85] Schleinitz J, Langevin M, Smail Y, Wehnert B, Grimaud L and Vuilleumier R 2022 Machine learning yield prediction from NiCOlit, a small-size literature data set of nickel catalyzed C–O couplings *J. Am. Chem. Soc.* **144** 14722–30
[86] Varoquaux G 2018 Cross-validation failure: small sample sizes lead to large error bars *NeuroImage* **180** 68–77
[87] Pinheiro M, Ge F, Ferré N, Dral P O and Barbatti M 2021 Choosing the right molecular machine learning potential *Chem. Sci.* **12** 14396–413
[88] Allers J P, Priest C W, Greathouse J A and Alam T M 2021 Using computationally-determined properties for machine learning prediction of self-diffusion coefficients in pure liquids *J. Phys. Chem.* B **125** 12990–3002
[89] Vabalas A, Gowen E, Poliakoff E and Casson A J 2019 Machine learning algorithm validation with a limited sample size *PLoS One* **14** 1–20
[90] Walters W P 2013 Modeling, informatics and the quest for reproducibility *J. Chem. Inf. Model.* **53** 1529–30
[91] Heil B J, Hoffman M M, Markowetz F, Lee S-I, Greene C S and Hicks S C 2021 Reproducibility standards for machine learning in the life sciences *Nat. Methods* **18** 1132–5
[92] Kapoor S and Narayanan A 2022 Leakage and the reproducibility crisis in ML-based science (arXiv:2207.07048)
[93] Muller K-R, Mika S, Ratsch G, Tsuda K and Scholkopf B 2001 An introduction to kernel-based learning algorithms *IEEE Trans. Neural Netw.* **12** 181–201
[94] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 Dropout: a simple way to prevent neural networks from overfitting *J. Mach. Learn. Res.* **15** 1929–58
[95] Varma S and Simon R 2006 Bias in error estimation when using cross-validation for model selection *BMC Bioinform.* **7** 91
[96] Krstajic D, Buturovic L J, Leahy D E and Thomas S 2014 Cross-validation pitfalls when selecting and assessing regression and classification models *J. Cheminform.* **6** 10
[97] Guyon I, Reza A, Alamdari S A, Dror G and Buhmann J M 2006 Performance prediction challenge *Proc. Int. Joint Conf. on Neural Networks (IJCNN 2006)* pp 2958–65
[98] Robinson M C, Glen R C and Lee A A 2020 Validating the validation: reanalyzing a large-scale comparison of deep learning and machine learning models for bioactivity prediction *J. Comput. Aided Mol. Des.* **34** 717–30
[99] Bell I H, Messerly R, Thol M, Costigliola L and Dyre J C 2019 Modified entropy scaling of the transport properties of the Lennard-Jones fluid *J. Phys. Chem.* B **123** 6345–63
[100] Allen M P and Tildesley D J 2017 *Computer Simulation of Liquids* (Oxford: Oxford University Press) (https://doi.org/10.1093/oso/9780198803195.001.0001)
[101] Mondello M and Grest G S 1997 Viscosity calculations of *n*-alkanes by equilibrium molecular dynamics *J. Chem. Phys.* **106** 9327–36
[102] Heyes D M 1988 Transport coefficients of Lennard-Jones fluids: a molecular-dynamics and effective-hard-sphere treatment *Phys. Rev.* B **37** 5677–96
[103] Rowley R L and Painter M M 1997 Diffusion and viscosity equations of state for a Lennard-Jones fluid obtained from molecular dynamics simulations *Int. J. Thermophys.* **18** 1109–21
[104] Meier K, Laesecke A and Kabelac S 2004 Transport coefficients of the Lennard-Jones model fluid. I. Viscosity *J. Chem. Phys.* **121** 3671–87
[105] Oderji H Y, Ding H and Behnejad H 2011 Calculation of the second self-diffusion and viscosity virial coefficients of Lennard-Jones fluid by equilibrium molecular dynamics simulations *Phys. Rev.* E **83** 061202
[106] Baidakov V G, Protsenko S P and Kozlova Z R 2012 Metastable Lennard-Jones fluids. I. Shear viscosity *J. Chem. Phys.* **137** 164507

[107] Costigliola L, Pedersen U R, Heyes D M, Schrøder T B and Dyre J C 2018 Communication: simple liquids' high-density viscosity *J. Chem. Phys.* **148** 081101
[108] Lautenschlaeger M P and Hasse H 2019 Transport properties of the Lennard-Jones truncated and shifted fluid from non-equilibrium molecular dynamics simulations *Fluid Phase Equilib.* **482** 38–47
[109] Galliéro G, Boned C, Baylaucq A and Montel F 2005 Influence of the mass ratio on viscosity in Lennard–Jones mixtures: the one-fluid model revisited using nonequilibrium molecular dynamics *Fluid Phase Equilib.* **234** 56–63
[110] Meyer N, Wax J-F and Xu H 2018 Viscosity of Lennard-Jones mixtures: a systematic study and empirical law *J. Chem. Phys.* **148** 234506
[111] Viet T Q Q, Khennache S, Galliero G, Alapati S, Nguyen P T and Hoang H 2022 Mass effect on viscosity of mixtures in entropy scaling framework: application to Lennard-Jones mixtures *Fluid Phase Equilib.* **558** 113459
[112] Kim K-S, Kim C, Karniadakis G E, Lee E K and Kozak J J 2019 Density-dependent finite system-size effects in equilibrium molecular dynamics estimation of shear viscosity: hydrodynamic and configurational study *J. Chem. Phys.* **151** 104101
[113] Yeh I-C and Hummer G 2004 System-size dependence of diffusion coefficients and viscosities from molecular dynamics simulations with periodic boundary conditions *J. Phys. Chem.* B **108** 15873–9
[114] Gabl S, Schröder C and Steinhauser O 2012 Computational studies of ionic liquids: size does matter and time too *J. Chem. Phys.* **137** 094501
[115] Petravic J 2004 Cooperative effects, transport and entropy in simple liquids *J. Chem. Phys.* **121** 11202–7
[116] Tukey J 1977 *Exploratory Data Analysis* (*Addison-Wesley Series in Behavioral Science*) (Reading, MA: Addison-Wesley)
[117] Brillinger D R 2011 *International encyclopedia of political science* (*data Analysis, Exploratory*) Badie B, Berg-Schlosser D and Morlino L (Thousand Oaks, CA: SAGE Publications)
[118] Bland J M and Altman D G 1996 Statistics notes: transforming data *BMJ* **312** 770
[119] Sheridan R P, Feuston B P, Maiorov V N and Kearsley S K 2004 Similarity to molecules in the training set is a good discriminator for prediction accuracy in QSAR *J. Chem. Inf. Comput. Sci.* **44** 1912–28
[120] Dimitrov S, Dimitrova G, Pavlov T, Dimitrova N, Patlewicz G, Niemela J and Mekenyan O 2005 A stepwise approach for defining the applicability domain of SAR and QSAR models *J. Chem. Inf. Model.* **45** 839–49
[121] Schroeter T S, Schwaighofer A, Mika S, Ter Laak A, Suelzle D, Ganzer U, Heinrich N and Müller K-R 2007 Estimating the domain of applicability for machine learning QSAR models: a study on aqueous solubility of drug discovery molecules *J. Comput. Aided Mol. Des.* **21** 651–64
[122] Fechner N, Jahn A, Hinselmann G and Zell A 2010 Estimation of the applicability domain of kernel-based machine learning models for virtual screening *J. Cheminform.* **2** 2
[123] Rakhimbekova A, Madzhidov T I, Nugmanov R I, Gimadiev T R, Baskin I I and Varnek A 2020 Comprehensive analysis of applicability domains of QSPR models for chemical reactions *Int. J. Mol. Sci.* **21** 5542
[124] Pedregosa F *et al* 2011 Scikit-learn: machine learning in Python *J. Mach. Learn. Res.* **12** 2825–30
[125] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems (available at: https://tensorflow.org)
[126] Chollet F *et al* 2015 Keras (available at: https://keras.io)
[127] Harris C R *et al* 2020 Array programming with NumPy *Nature* **585** 357–62
[128] Virtanen P *et al* SciPy 1.0 Contributors 2020 SciPy 1.0: fundamental algorithms for scientific computing in Python *Nat. Methods* **17** 261–72
[129] The Pandas Development Team 2021 pandas-dev/pandas: Pandas 1.3.4 (available at: https://doi.org/10.5281/zenodo.5574486)
[130] Peterson A A, Christensen R and Khorshidi A 2017 Addressing uncertainty in atomistic machine learning *Phys. Chem. Chem. Phys.* **19** 10978–85
[131] Vandermause J, Torrisi S B, Batzner S, Xie Y, Sun L, Kolpak A M and Kozinsky B 2020 On-the-fly active learning of interpretable Bayesian force fields for atomistic rare events *npj Comput. Mater.* **6** 20
[132] Xie Y, Vandermause J, Ramakers S, Protik N H, Johansson A and Kozinsky B 2022 Uncertainty-aware molecular dynamics from Bayesian active learning: phase transformations and thermal transport in SIC (arXiv:2203.03824)
[133] Hoffmann M *et al* 2021 Deeptime: a python library for machine learning dynamical models from time series data *Mach. Learn.: Sci. Technol.* **3** 015009
[134] Bell I H, Messerly R, Thol M, Costigliola L and Dyre J C 2022 Correction to "modified entropy scaling of the transport properties of the Lennard-Jones fluid" *J. Phys. Chem.* B **126** 5595–6
[135] Karniadakis G E, Kevrekidis I G, Lu L, Perdikaris P, Wang S and Yang L 2021 Physics-informed machine learning *Nat. Rev. Phys.* **3** 422–40