JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

Mixture of Clustered Bayesian Neural Networks for Modeling Friction Processes at the Nanoscale

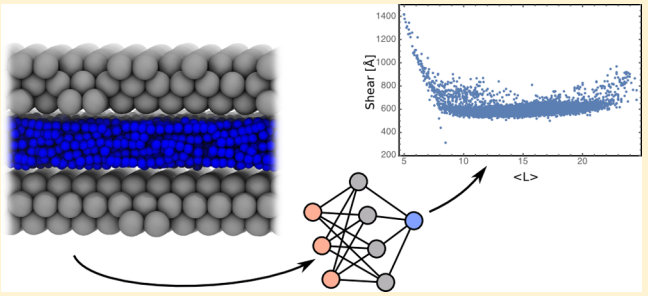
Martha A. Zaidan,*[,†,‡] Filippo F. Canova,[†,‡] Lasse Laurson,[‡] and Adam S. Foster[‡,§]

[†]Aalto Science Institute, Aalto University, P.O. Box 11100, 00076 Aalto, Espoo, Finland

[‡]COMP Centre of Excellence, Department of Applied Physics, Aalto University, P.O. Box 11100, 00076 Aalto, Espoo, Finland

[§]Division of Electrical Engineering and Computer Science, Kanazawa University, Kanazawa 920-1192, Japan

ABSTRACT: Friction and wear are the source of every mechanical device failure, and lubricants are essential for the operation of the devices. These physical phenomena have a complex nature so that no model capable of accurately predicting the behavior of lubricants exists. Thus, lubricants cannot be designed from scratch but have to be screened through expensive trial−error tests. In this study we propose a machine learning (ML) method that infers the relationship between chemical composition of lubricants and their performance from a database. Because no such database of desirable size and completeness is publicly available, we compiled one from molecular dynamics (MD) simulations of toy-model fluids nanoconfined between shearing surfaces. The fluid-friction relation is modeled by a Bayesian neural network (BNN), trained to reproduce the results for a training set of fluids. Due to the inhomogeneous data distribution it was necessary to carefully pick fluids for training and validation from the database with advanced clustering algorithms, rather than using the standard random selection. Different BNNs were then trained on the data clusters and their predictions combined into a mixture of experts. The model provides a prediction of lubricants performance as well as an error bar, at a fraction of the cost of MD. Because most values agree with the actual MD simulations within the estimated error σ, we conclude that the model is satisfactory. This method addresses the challenges brought by noisy, badly distributed, high-dimensional data that are likely to appear in reality as well, and it can be extended to real fluids, if a database could be provided.



1. INTRODUCTION

Friction and wear are the main cause of failure in every mechanical system, having quite a measurable economic impact.[1] Since ancient times, the problem has been partially solved with lubricants. However, as technology scales down toward the nanoscale, the effects of friction become more dramatic,[2] to the point where nanodevices cannot work at all. Specialized lubricants suitable to operate in such systems are then necessary. Theoretical models for friction and lubrication work in general at macroscopic scales, but none captures the dominant effects at the nanoscale.[3−6] Thus, we are left to explore a vast chemical space only through expensive trial−error laboratory tests, to search for optimal lubricants. A reliable theory for nanoconfined lubricants could provide the guidelines needed for lubricant design, speeding up the materials screening process. However, the underlying physics involves complex solid−liquid interactions that strongly depend on the atomic details of the materials, and formulating a general model seems an impractical task.

Researchers and practitioners have applied machine learning (ML) to discover patterns or predict outcomes from prior data,[7] traditionally in the fields of image recognition, medical diagnosis, robotics, and speech recognition among others.[8] In the last couple of years, ML has also gained attention from the materials community,[9] attracted by the promise of replacing expensive models and experiments with accurate and much faster ML models inferred from data,[10−13] with applications ranging from predicting materials properties[14] to identifying flow defects in disordered solids.[15] With ML, it is in principle possible to machine-learn away all the complexity of frictional processes and approximate the relationship between the chemical composition of a lubricant and its performance solely from prior measurements.

In this work, we select artificial neural networks (NNs) as ML models. NNs provide a robust approach to approximating real-valued (prediction) and discrete-valued (classification) target functions because they can mimic nonlinearity of the functions and their learning methods are well-developed. NNs have been a popular choice among ML methods for approximating complex functions[16] and have been adopted in a wide variety of problems in many fields.[17] NNs are structured combinations of nonlinear functions with many parameters (called weights). NN models are *trained* by adjusting their weights so that the predicted outputs match the known ones for all inputs in a database. The optimization can get stuck in

Received: August 22, 2016
Published: December 12, 2016

DOI: 10.1021/acs.jctc.6b00830
J. Chem. Theory Comput. 2017, 13, 3−8

local minima, or even overfit when the final model performs well on the training inputs but does not generalize to other cases. Ensemble learning[18] combines several NNs with random initial weights into a committee machine. This has been shown to yield a model that typically performs better than the individual NNs because it is less affected by local minima.[19] However, ensemble learning is not always suitable to produce a robust ML model, especially when the training data sets are complex, leading to overfitting or underfitting issues.[20,21] With the implementation of Bayesian inference into NNs, known as a Bayesian neural networks (BNN),[22,23] a regularisation term is added to the NN performance function. This steers the training toward simpler NNs, thus countering overfit. Moreover, BNNs automatically provide a degree of belief on the estimated output, which can be used to assess the quality of the predictions.

In this study we combine several ML methods into a novel approach to approximate the frictional behavior for a relatively simple set of model fluids. Because a consistent database of friction measurements of suitable size (>1000 samples) is not publicly available, we first compile one from molecular dynamics (MD) simulations of model liquids confined between simple shearing surfaces. The nontrivial relationship between the high-dimensional fluid descriptor and its shear rate is approximated by a modified committee machine of Bayesian neural networks, to avoid the issues of local minima and overfitting.

Because most fluids give low, similar shear rates despite being different, and only a few show high shear rate, the data distribution is not homogeneous, and conventional training schemes failed to produce reasonable models. We then partition the fluids into clusters according to their descriptors and train an *expert* BNN for each cluster, ensuring all categories are equally represented in the training and validation data sets. The final shear prediction is obtained by appropriately mixing all the BNNs with a nonlinear combiner, depending on the input descriptor.[24−26]

This paper is organized as follows. Section 2 explains the generation of our friction data set and the challenges on data modeling complexity. A novel ML strategy as well as the algorithms involved are described in the section 3. Section 4 presents and discusses the obtained results as a validation of the proposed approach. Data analysis was carried out with Wolfram's Mathematica 10.4.[27] Finally, the paper is concluded in section 5.

2. SIMULATED DATABASE

Because no database of sufficient size containing molecular descriptors for lubricants and their performance is publicly available, we calculated one using classical MD simulations of model fluids, nanoconfined between solid surfaces. Given the large amount of computation required, we can only consider simple toy models to represent the system. The confining surfaces are modeled by slabs of a FCC lattice, including 10 × 10 units cells in the *xy* plane and three atomic layers along *z*. Their atoms interact with each other via a Morse potential. The lubricant is composed of random chains of particles held together by harmonic bonds. A three-body harmonic potential keeps the chains linearly aligned. Particles of the liquid interact with each other via a Lennard-Jones potential (nonbonded), whereas Morse is used for liquid−surface interactions. The simulation is performed in reduced units; however, we made them consistent with standard ones. The unit cell size of the surfaces is 2.86 Å and atomic mass is 30 amu. Each bead of the liquid has a mass of 20 amu and the harmonic potential in the chain keeps them at 1.5 Å. We considered 8000 such systems, with different molecular compositions of the lubricant. All systems include 4000 liquid particles, connected in chains of random length, according to predesigned, random distributions. The chain length distribution is given by a sum of Gaussian functions; the number of these functions, their mean, and variance are randomized for each system, with the constraint of maximum chain length of 25 particles. A snapshot from a typical system is shown in the inset of 1a. A constant normal

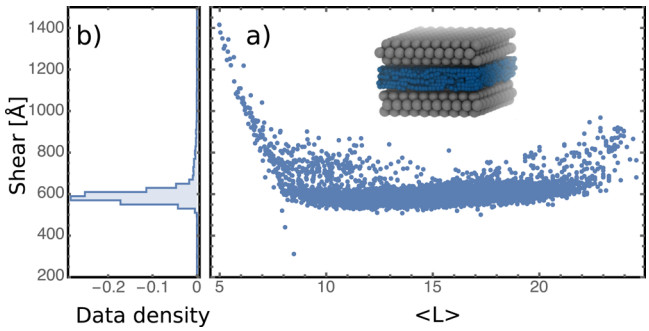


Figure 1. (a) Shear distance of all fluids as a function of their average chain length. The inset shows the snapshot of a system rendered with VMD:[28] blue particles belong to the liquid, and gray ones are in the shearing surfaces. (b) Density of the collected data points along the shear axis.

load of 0.1 kcal/(mol Å) is applied to both surfaces, keeping the system confined. Shear motion is enforced by applying a constant lateral force of 0.01 kcal/(mol Å) in opposite directions to each surface. The simulation temperature was kept constant at 350 K by a Langevin thermostat. After an initial relaxation time of 3 ns, the systems reach a steady state, and production runs of 6 ns start. The overall sliding distance of the two surfaces during the run is related to the average friction force exerted by the liquid, and thus it measures the performance of the lubricant. Even with this simple model and a simulation time step of 3 fs, the equilibration and production run of one lubricant takes up to few hours on a single GPU.

The overall shear distance of each system is plotted against the average chain length $\langle L \rangle$ of the liquid in Figure 1. Despite the fact that $\langle L \rangle$ is a quite reductive descriptor, it is useful to show the trends in our results. Liquids made of light chains shear for as much as 1450 Å, and friction increases linearly up to $\langle L \rangle \approx 8$. The increased viscosity with chain length is simply explained because longer chains will give stronger intermolecular interactions that oppose shear motion. Similar behavior can be seen in hydrocarbons.[29,30] For $8 \leq \langle L \rangle \leq 22$ the shear is only about 600 Å and increases slightly with $\langle L \rangle$; most data points fall into this range. When $\langle L \rangle \geq 22$, the shear increases, reaching about 1000 Å. We found that fluids with mostly long chains solidify under nanoconfinement, and their molecules cease to shear against each other, thus dissipating less energy. Other high-shear mixtures can be seen around $\langle L \rangle = 10$. The chain-length distribution of these liquids resembles that of a base fluid of longer chains with an additive of short ones. The small chains contribute to lower the viscosity of the dense base fluid. Figure 2a shows the average thickness of the liquid and its relationship with $\langle L \rangle$. Mixtures of mostly short chains have

DOI: 10.1021/acs.jctc.6b00830
J. Chem. Theory Comput. 2017, 13, 3−8

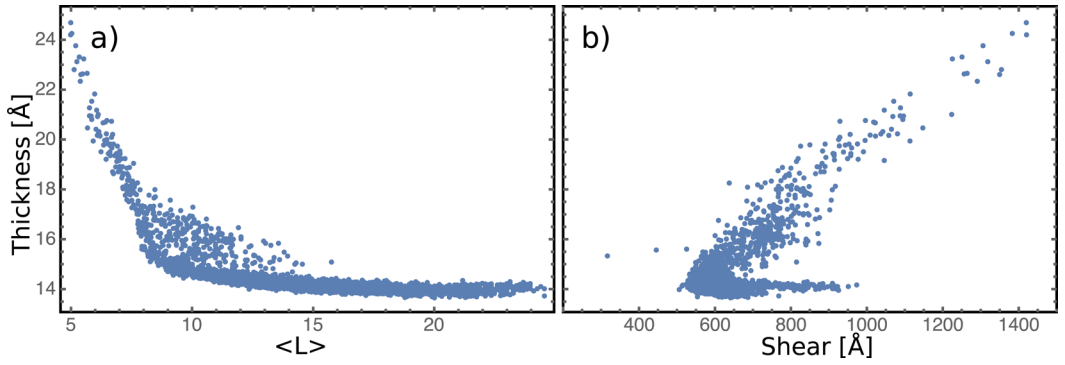


Figure 2. (a) Dependence of the mixture's thickness on the average chain length and (b) its correlation with the shear distance.

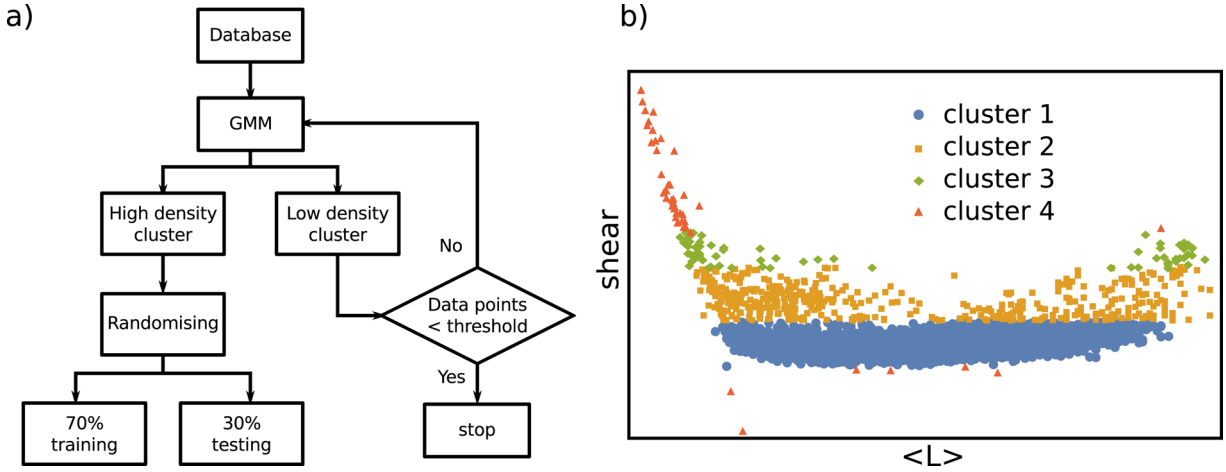


Figure 3. (a) Iterative Gaussian mixture model algorithm for selection of training and testing data. (b) Resulting clusters in our database.

weaker intermolecular interactions and thus tend to occupy more volume; a similar effect is observed for mixtures where short chains are added to a base fluid of longer chains. As $\langle L \rangle$ increases, the liquid film becomes denser. Shear and thickness follow two main trends, visible in Figure 2b. In most liquids, the two are linearly correlated; however, when $\langle L \rangle$ increases above 22, the fluid thickness and shear become uncorrelated.

3. MACHINE LEARNING MODEL

The main difficulty in modeling the lubrication performance is the statistical imbalance of available data, illustrated by the histogram in Figure 1b. Most of the generated shear are concentrated on the values between 550−650 Å, whereas there is only a small number of mixtures with shear values lying anywhere else. Unlike many other ML applications where more diverse data can be gathered with relatively small effort, it is not so easy to design a mixture with a specific shear performance and get a more uniform histogram. Moreover, the mixtures that include only long chains are only a small subset of all possible combinations. In this situation, the standard subdivision of the data set into 70−30% for training and testing is not effective because there is a large possibility that sparse high-shear systems will be overshadowed by the more abundant low-shear ones during training, and the final model will not be general. Another issue is the stochastic nature of friction, bringing uncertainty and outliers in the data sets. Capturing the uncertainty as well as dealing with the outliers are nontrivial modeling tasks.

3.1. Iterative Gaussian Mixture Model (iGMM). We use the Gaussian mixture model (GMM) clustering to address the imbalance in the MD data set. The GMM expresses the probability of measuring a shear value s given a set of conditions λ as

$$p(s|\lambda) = \sum_{j=1}^{T} \gamma_j p(s|\bar{s}_j, \Sigma_j) \quad (1)$$

where γ_j for $j = 1, ..., T$, are the mixture weights and $p(s|\bar{s}_j,\Sigma_j)$ is a multivariate Gaussian distribution with mean $\bar{s}_j$ and covariance matrix Σ_j. By setting $T = 2$, GMM effectively approximates the density of data points with a sum of two smooth Gaussian distributions: one with high and one with low density of shear measurements. The parameters $\lambda = \{\gamma_j, \bar{s}_j, \Sigma_j\}$ are estimated using the expectation-maximization algorithm.[23] The high-density cluster is ready to be partitioned into 70% training and 30% testing data sets through random selection. The low-density cluster goes through another GMM iteration as long as enough data points remain. By applying this procedure, summarized in Figure 3a, on our database, we obtained the four clusters illustrated in Figure 3b.

3.2. Bayesian Neural Network. We use a Bayesian neural network[23] (BNN) to model the unknown relationship $s = f(\mathbf{x})$ between the system $\mathbf{x}$ and its shear response s (Figure 4a). For our purpose, the input layer consists of a 25-dimensional array $\mathbf{x} = \{x_1, x_2, ..., x_{25}\}$, containing the concentration x_i of chains of length i. Because 25 particles is the longest chain allowed in our simulations, this descriptor is enough to characterize the fluid completely. The output is just one number, the shear distance s,

DOI: 10.1021/acs.jctc.6b00830
J. Chem. Theory Comput. 2017, 13, 3−8

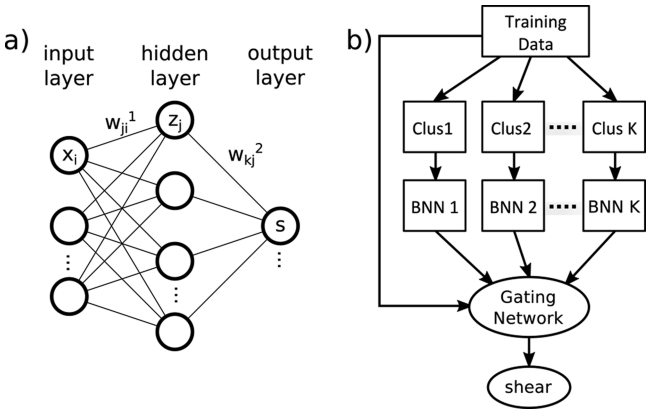


Figure 4. (a) Schematic representation of a NN with one hidden layer. (b) Mixture of expert model.

normalized by the maximum so that the resulting values are between 0 and 1. Using cross-validation methods, we determined that the optimal amount of hidden layer neurons is 15. The jth neuron in the Lth layer computes its output z_j^L as

$$z_j^L = \sigma\left(\sum_i w_{ji}^L x_i + b_j^L\right) \tag{2}$$

where w_{ji}^L is the weight of connection between the computing neuron and its ith input in the preceding layer, and b_j^L is an additional bias parameter. The activation function is $\sigma(x) = \tanh(x)$ in the hidden layer and $\sigma(x) = x$ for output layer neurons. Once a training data set $\{\mathbf{x}, \tilde{\mathbf{y}}\}$ with reference inputs $\mathbf{x}$ and their corresponding outputs $\tilde{\mathbf{y}}$ is given, it becomes possible to find a suitable set of weights $\mathbf{w}$ by minimizing the cost function

$$E = \frac{\beta}{2}\sum_n (f(\mathbf{x}_n,\mathbf{w}) - \tilde{y}_n)^2 + \frac{\alpha}{2}\sum_i w_i^2 \tag{3}$$

where $f(\mathbf{x}_n,\mathbf{w})$ is the output of the BNN from training inputs $\mathbf{x}_n$. The first term is the prediction error of the model on the training data, and its minimization leads to a model that fits the data. The second term comes from the application of Bayesian inference in the training and effectively gives a penalty to complex models with larger weights, thus impeding overfit. The two contributions are weighted by hyper-parameters α and β, iteratively updated during the training.

The accuracy of single BNN in our preliminary studies was not satisfactory, and for this reason we tried with a committee machine of BNNs.[18] The main idea, as illustrated in Figure 4b, is to divide the fluids in the training set into separate groups depending only on their descriptors, train *expert* networks separately for each group, and combine their outputs with a gating network to obtain the shear prediction.[24,25,31] The training set is subdivided with k-means clustering algorithm,[32] applied to the 25-dimensional input vectors. The total number of clusters k was determined by calculating the Davies−Bouldin index[33] after running the clustering with k = 1, 2, ..., 16: we found that k = 11 gives the lowest index. The most populated cluster contains 876 training samples, whereas the smallest has 178. Two other small clusters include about 300 samples, and all others have more than 500. The gating network is a GMM,[23] whose parameters are optimized with the expectation-maximization algorithm, once the BNN experts were trained.

4. RESULTS

Figure 5a shows the regression plot for the test data. The black line represents the perfect fit, where predicted shear and the

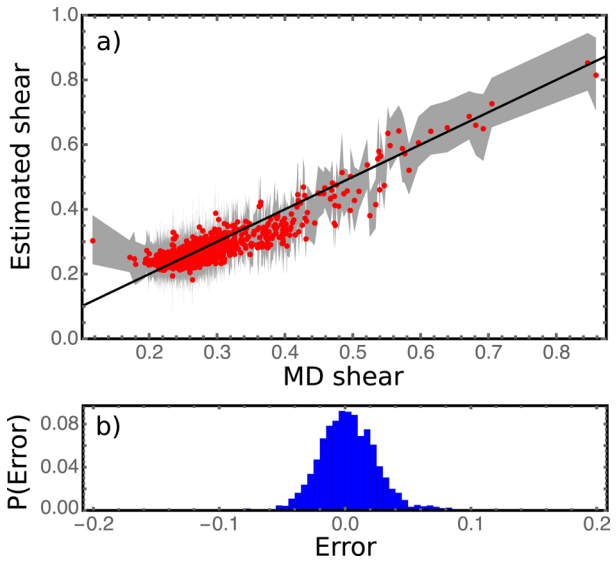


Figure 5. (a) Performance of the model on the test data set. The gray area marks the $\pm 2\sigma$ range. (b) Error histogram.

MD calculations coincide, whereas the gray area shows the range between the prediction and $\pm 2\sigma$, where σ is the estimated error. Even though some estimated shear values deviate considerably from the perfect line, the error histogram in Figure 5b shows that almost all estimation errors are below 0.05, corresponding to about 55 Å.

Figure 6 shows a more detailed bar chart of the errors distribution. About 83.3% of the predictions are less than σ off,

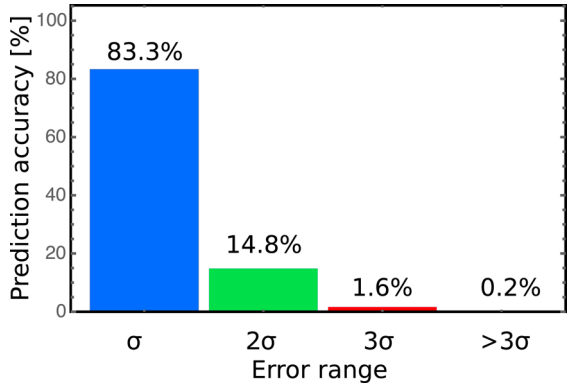


Figure 6. Accuracy of the model on the test data set.

and there are approximately 14.8% as bad as 2σ. It is noted that there is only 0.2% estimated shears that lie outside of 3σ. This bar chart suggests that the developed ML model is very accurate because 99.75% of estimates lie within the 3σ region.

Table 1 lists the performance of progressively more complex models in terms of root-mean-square error (RMSE) and Pearson product-moment correlation coefficient R^2. The first entry corresponds to the simplest linear model. This model does not give satisfactory performance because it is not flexible enough to cope with the nonlinearity of the data. The second entry is the average performance of individual NNs trained

DOI: 10.1021/acs.jctc.6b00830
J. Chem. Theory Comput. 2017, 13, 3−8

Table 1. Performance Metric Comparisons of Different ML Strategies Using RMSE and $1 - R^2$

	Method	RMSE	$1 - R^2$
1	linear model	0.0381	0.2997
2	NN	0.0262	0.1297
3	NN committee	0.0260	0.1284
4	KRR	0.0258	0.1254
5	BNN	0.0251	0.1182
6	BNN committee	0.0250	0.1173
7	BNN mixture	0.0250	0.1175
8	BNN mixture of experts	0.0240	0.1144
9	KRR mixture of experts	0.0320	0.1585

from different random initial weights. The prediction error is slightly reduced by averaging the output over all the NNs in a committee machine (third entry). We also tested kernel ridge regression (KRR), because it has been successfully applied to quantum chemistry.[12] The performance metrics suggest that KRR (fourth line) is better than the standard NN or NN committee. Switching to BNNs gives a more significant performance improvement. The NN weights can get stuck in local minima during training, depending on their initial values: this limits the efficacy of training. The problem is partly solved in committee machines, leading to better performance in general. The BNN mixture model implements a gating network to mix the BNN outputs; however, all BNN are trained with the same database and only differ in their random initialization. After training, the gating network of the mixture performs effectively the same operation as the averaging in the committee, thus giving similar performances. In our final model, BNNs are separately trained on the different clusters, and combined by a gating network into a mixture of experts. This brings an additional small performance improvement. We also trained different KRRs on the clustered data and combined them in the same fashion; however, the results are even worse than a single KRR (last entry). To check whether the different performance values from the metrics are statistically meaningful, we calculated the Wilcoxon signed rank tests for all pairs of models. The resulting p-values are listed in Table 2. The

Table 2. Wilcoxon Signed Rank Test Results (p-Value)

	1	2	3	4	5	6	7	8	9
1		0.03	0.05	0.05	0.02	0	0	0	0
2			0.84	0.86	0.74	0.76	0.76	0.0005	0
3				0.98	0.59	0.62	0.62	0.0002	0
4					0.61	0.64	0.64	0.0002	0
5						0.97	0.97	0.0015	0
6							0.99	0.0012	0
7								0.0012	0
8									0

worst performance of the linear model is a solid result, as $p \leq 0.05$ when compared to all other models. This was expected given the simple nature of the model and the high complexity of the data. The rank test, however, suggests that most models, from NN (2) to BNN mixture (7) could effectively be equivalent, because the high p-value indicates a significant probability that the observed differences are random. The poor performance of a KRR mixture of experts can be caused by underfitting for small clusters of data: all of the KRR experts gave larger RMSE than the corresponding BNN experts. It is possible that the kernel function we chose (L_1 norm) is not as flexible as a NN.

Based on the information given in Table 1 and the small p-values from the rank test, it can be concluded that the mixture of all BNNs outperforms all other tested strategies. To confirm whether our strategy is robust and there is no overfitting, we trained the model using only half of the data in each cluster and used the other half for testing. The training and testing halves were also switched, and the model trained again. The two models obtained this way have similar performances, giving RMSE of 0.0248 and 0.0252, which indicates that the method is stable.

5. CONCLUSIONS

We demonstrated how it is possible to approximate complex physical phenomena such as friction using ML models. The proposed method tackles the challenges posed by the nature of the system and the availability/quality of data. Iterative GMM ensures that the sparse data is partitioned between training and test sets in a balanced way. Without this step, the trained models could only make the most statistically obvious prediction that all fluids would give the same shear of about 60 nm. BNNs are chosen as the main modeling algorithm due to their ability to deal with uncertainty in the data and to prevent overfitting. Fluids in the training set are divided into clusters with the k-means algorithm, and a BNN expert is trained on each one. The final shear prediction is obtained by weighting the response of the experts appropriately depending on the fluid descriptor. The model not only predicts the shear but also provides standard deviation around its mean prediction, which can point out poorly sampled regions of the chemical space in the training set. As expected, the largest uncertainty as well as fail rate of the model occur for outlier systems with very low or high shear. However, most predictions are within the estimated standard deviation, making the model predictive.

Even if the method might seem complicated, the performance results indicate that the proposed ML strategy outperforms all the simple methods we tested. The ML model takes only a fraction of a second to calculate the answer—even the whole training process does not exceed 5 min on a conventional desktop computer with our Matlab implementation.[34]

On the contrary, MD simulations of one fluid takes more than 2 h, and a real experiment may take even longer, so the real bottleneck is the generation of the database. Although the accuracy of a ML model is mathematically bound to be lower than that of the MD simulation or experiments used to train it, and it is unable to provide physical insight into the nature of frictional processes, it enables much faster lubricant screening.

The toy model we employed to simulate the shear response of the fluids is not realistic; however, it can be argued that many lubricants consist of a mixture of polymers of different lengths and types, and they might show similar behaviors. It should be noted that simulating our database with more accurate atomistic models for hydrocarbon chain mixtures would require an exceedingly long time. Nevertheless, if such a database was calculated or, even better, compiled from experimental measurements, the ML method presented here would still be applicable and be a powerful tool for lubricant optimization.

The drawback of this method is in the use of k-means clustering to group different training data. Due to high-dimensional data, k-means may not be very optimal to obtain

DOI: 10.1021/acs.jctc.6b00830
J. Chem. Theory Comput. 2017, 13, 3−8

robust and accurate clusters that really reflect the lubricant characteristics. Our proposed ML method also requires more training data. As shown in Figure 1, most shear data points (i.e., obtained data from MD) concentrates on 600 Å shear. As a result, the model is better in the low-shear regime whereas on the high-shear regime is not as good (see regression plot in Figure 5). The future direction of this work is to apply different clustering strategies and evaluate more data points using MD, where the optimal chain-length distributions are suggested by our developed ML models.

AUTHOR INFORMATION

Corresponding Author

*M. A. Zaidan. E-mail: martha.a.zaidan@aalto.fi. Phone: +358 (0) 4692 20480. Fax: +358 (0) 9855 4019.

ORCID

Martha A. Zaidan: 0000-0002-6348-1230

Adam S. Foster: 0000-0001-5371-5905

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors have been supported by the Academy of Finland through its Centres of Excellence Program project No. 915804 and acknowledge use of the CSC, Helsinki, for computational resources. L.L. is supported by the Academy of Finland through an Academy Research Fellowship (Project No. 268302). A.S.F. received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 676580 NOMAD, a European Center of Excellence and no. 686053 CRITCAT. This work is supported in part by COST Action MP1303.

REFERENCES

(1) Carpick, R. W. *Science* **2006**, *313*, 184−185.
(2) Liechti, K. M. *Science* **2015**, *348*, 632−633.
(3) Mo, Y.; Turner, K. T.; Szlufarska, I. *Nature* **2009**, *457*, 1116.
(4) Urbakh, M.; Meyer, E. *Nat. Mater.* **2010**, *9*, 8−10.
(5) Boukhvalov, D. W.; Dobrovitski, V. V.; Kögerler, P.; Al-Saqer, M.; Katsnelson, M. I.; Lichtenstein, A. I.; Harmon, B. N. *Inorg. Chem.* **2010**, *49*, 10902−10906.
(6) Matsubara, H.; Pichierri, F.; Kurihara, K. *Phys. Rev. Lett.* **2012**, *109*, 197801.
(7) Krishnapuram, B.; Yu, S.; Rao, R. B. *Cost-sensitive Machine Learning*; CRC Press: Boca Raton, FL, 2011; pp xiii-xiv.
(8) Michalski, R. S.; Carbonell, J. G.; Mitchell, T. M. *Machine learning An artificial intelligence approach*; Springer Science & Business Media: Berlin, 2013; p 13.
(9) Schuld, M.; Sinayskiy, I.; Petruccione, F. *Contemp. Phys.* **2015**, *56*, 172−185.
(10) Rupp, M.; Tkatchenko, A.; Müller, K.-R.; Von Lilienfeld, O. A. *Phys. Rev. Lett.* **2012**, *108*, 058301.
(11) Montavon, G.; Rupp, M.; Gobre, V.; Vazquez-Mayagoitia, A.; Hansen, K.; Tkatchenko, A.; Müller, K.-R.; von Lilienfeld, O. A. *New J. Phys.* **2013**, *15*, 095003.
(12) Rupp, M. *Int. J. Quantum Chem.* **2015**, *115*, 1058−1073.
(13) Behler, J. *J. Chem. Phys.* **2016**, *145*, 170901.
(14) Pilania, G.; Wang, C.; Jiang, X.; Rajasekaran, S.; Ramprasad, R. *Sci. Rep.* **2013**, *3*.10.1038/srep02810
(15) Cubuk, E. D.; Schoenholz, S. S.; Rieser, J. M.; Malone, B. D.; Rottler, J.; Durian, D. J.; Kaxiras, E.; Liu, A. J. *Phys. Rev. Lett.* **2015**, *114*, 108001.
(16) Maren, A. J.; Harston, C. T.; Pap, R. M. *Handbook of neural computing applications*; Academic Press: New York, 2014; pp 1−6.
(17) Hagan, M. T.; Demuth, H. B.; Beale, M. H.; De Jesús, O. *Neural network design*; PWS Publishing Company: Boston, 2014; pp 1.5−1.7.
(18) Naftaly, U.; Intrator, N.; Horn, D. *Network-Comp.Neural* **1997**, *8*, 283−296.
(19) Krogh, A.; Vedelsby, J. *Adv. Neural Inf. Process Syst.* **1995**, *7*, 231−238.
(20) Bishop, C. M. *Neural networks for pattern recognition*; Oxford University Press: Oxford, U.K., 1995; pp 364−369.
(21) Hansen, K.; Montavon, G.; Biegler, F.; Fazli, S.; Rupp, M.; Scheffler, M.; von Lilienfeld, O. A.; Tkatchenko, A.; Müller, K.-R. *J. Chem. Theory Comput.* **2013**, *9*, 3404−3419.
(22) Titterington, D. *Stat. Sci.* **2004**, *19*, 128−139.
(23) Bishop, C. M. *Pattern recognition and machine learning*; Springer: Berlin, 2006; pp 277−284.
(24) Mengersen, K.; Robert, C.; Titterington, M. *Mixtures: estimation and applications*; John Wiley & Sons: New York, 2011; Vol. *896*, pp 103−112.
(25) Yuksel, S. E.; Wilson, J. N.; Gader, P. D. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1177−1193.
(26) Masoudnia, S.; Ebrahimpour, R. *Artif. Intell. Rev.* **2014**, *42*, 275−293.
(27) Wolfram Research, Inc. *Mathematica* 10.4; https://www.wolfram.com (accessed: July 30, 2016).
(28) Humphrey, W.; Dalke, A.; Schulten, K. *J. Mol. Graphics* **1996**, *14*, 33−38.
(29) Speight, J. G. *Handbook of Industrial Hydrocarbon Processes*; Gulf Professional Publishing: Amsterdam, 2010; p 511.
(30) Dymond, J. H.; Øye, H. A. J. *J. Phys. Chem. Ref. Data* **1994**, *23*, 41−53.
(31) Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; Hinton, G. E. *Neural Comput.* **1991**, *3*, 79−87.
(32) Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881−892.
(33) Davies, D. L.; Bouldin, D. W. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *PAMI-1*, 224−227.
(34) The MathWorks, Inc. *Matlab* R2016a; www.mathworks.com (accessed: July 30, 2016).

DOI: 10.1021/acs.jctc.6b00830
J. Chem. Theory Comput. 2017, 13, 3−8