PCCP

Artificial Neural Network Prediction of Self-Diffusion in Pure Compounds over Multiple Phase Regimes

Journal:	*Physical Chemistry Chemical Physics*
Manuscript ID	CP-ART-12-2020-006693.R1
Article Type:	Paper
Date Submitted by the Author:	09-Feb-2021

Complete List of Authors:	Allers, Joshua; Sandia National Laboratory, Department of Organic Materials Science Garzon, Fernando; Sandia National Laboratory, Department of Power Sources Research and Development; University of New Mexico, Center of Micro-Engineered Materials Alam, Todd; Sandia National Laboratory, Department of Organic Materials Science

ARTICLE

Artificial Neural Network Prediction of Self-Diffusion in Pure Compounds over Multiple Phase Regimes

Received 00th January 20xx,
Accepted 00th January 20xx

DOI: 10.1039/x0xx00000x

Joshua P. Allers, [a] Fernando H. Garzon [b,c] and Todd M. Alam *[a]

Artificial neural networks (ANNs) were developed to accurately predict the self-diffusion constants for pure components in liquid, gas and super critical phases. The ANNs were tested on an experimental database of 6625 self-diffusion constants for 118 different chemical compounds. The presence of multiple phases results in a heavy skew in the distribution of diffusion constants and multiple approaches were used to address this challenge. First, an ANN was developed with the raw diffusion values to assess what the main drawbacks of this direct method were. The first approach for improving the predictions involved taking the log10 of diffusion to provide a more uniform distribution and reduce the range of target output values used to develop the ANN. The second approach involved developing individual ANNs for each phase using the raw diffusion values. Results show that the log transformation leads to a model with the best self-diffusion constant predictions and an overall average absolute deviation (AAD) of 6.56%. The resultant ANN is a generalized model that can be used to predict diffusion across all three phases and over a diverse group of compounds. The importance of each input feature was ranked using a feature addition method revealing that the density of the compound has the largest impact on the ANN prediction of self-diffusion constants in pure compounds.

1 Introduction

Understanding diffusion of chemical compounds is important for the design and optimization of numerous chemical engineering and energy applications. With many processes operating over a range of conditions, there is interest in predicting how the self-diffusion constant will change relative to various input parameters. When it comes to developing new technologies or optimizing existing ones, experiments and simulations often lead to large expenses in time and effort, especially when screening multiple compounds and conditions. A generalized predictive model for diffusion would allow identification of more promising materials or conditions without the need for obtaining and isolating raw materials or incurring large computational expenses.

The kinetic theory of gases can accurately describe transport properties of low-density gases, but no equivalent, fully generalized model exists for dense gases, liquids, or supercritical fluids.[1] Many different approaches have been developed to predict the self-diffusion in dense fluids including molecular dynamics (MD) simulations[2-4] and theories based on free volume[5-7] or excess entropy.[8-10] MD approaches are the most common and utilize pair-wise potentials to describe the interactions between particles. From MD simulations, empirical equations can be developed to predict diffusion. One of the more accurate empirical models was developed by Silva *et. al.* who extended the Lennard Jones (LJ) diffusion into real systems.[11] The model successfully predicts diffusion for a large range of both hydrogen and non-hydrogen bonding compounds. Other successful models developed by Lee and Thodos and Zhu *et al.* are among the few able to predict self-diffusion over multiple phases.[12, 13] A brief review of existing empirical equations for predicting self-diffusion are presented in the supplementary information (Eqs. S1 – S26). These are mainly based on hard sphere (HS) and LJ interaction potentials. For a more comprehensive list and discussion of models, see the review by Suarez-Iglesias *et al.*[14]

Although these approaches produce reasonable predictions, issues can arise when attempting to predict the self-diffusion of a new compound not in the original data set used for parameter optimization. When introducing a new compound, any empirical model constants must be refit and, in the case of MD approaches, new simulations must be run to determine the compound parameters. An alternative approach is pursued in this work using machine learning (ML) methods, which allow models to be developed on existing experimental data. Rather than using the results of MD simulations to fit proposed equations for molecular properties, artificial neural networks (ANN) are trained using the experimental diffusion constants. The performance of ANN models is dependent on the data, both the quality and quantity. When it comes to self-diffusion in pure compounds, there is an abundance of experimental data available, making ML a viable option. The advantage of using an ANN is in the development process where separate training and test data sets are utilized. Using unseen data in

[a] Department of Organic Materials Science, Sandia National Laboratories, Albuquerque, NM 87185, USA. E-mail: tmalam@sandia.gov
[b] Advanced Materials Laboratory, Sandia National Laboratories, Albuquerque, NM, 87185, USA
[c] Center of Micro-Engineered Materials, University of New Mexico, Albuquerque, NM 87106, USA

Electronic Supplementary Information (ESI) available: Database 1 with density values and database 2 without density values. See DOI: 10.1039/x0xx00000x

the test set allows development of generalized models that will not need to be refit every time new data is presented.

ML has already been used to successfully model physical and transport properties in various systems.[15-17] The use of ML methods to predict diffusion is still limited with examples including the prediction of diffusion for organic compounds in air,[18] binary gas mixtures,[19, 20] organic compounds in water,[21-23] and mixtures of binary solvents and hydrocarbons.[24, 25] ML studies have also been reported for lithium diffusion through solid-state membranes,[26] and activation energies for atomic diffusion on metal surfaces.[27, 28] In previous work, we used ANNs and random forests to predict the self-diffusion of Lennard Jones fluids.[29] Focus was directed at understanding how different features would impact ML predictions and determining what methods produced the most accurate models. Our group has also reported the use of ANNs to correct finite-size effects in MD simulations of diffusion in binary LJ fluids, predicting a correction factor rather than the diffusion constant directly.[30]

The existing ML literature tends to focus on binary mixtures, so this work aims to fill the gap between model systems and binary systems by focusing on predicting the self-diffusion of pure chemical compounds. We take an existing database consisting of liquid, gas, and supercritical self-diffusion constants and use it to develop multiple ANN models. Different approaches are employed to handle the large range and heavy skew in the distribution of diffusion constants. We also build on our previous work by identifying the features that most impact the prediction of diffusion.

2 Methods

2.1 Pure solution database and features

The self-diffusion data used in this work was gathered and published by Suarez-Iglesias *et al*.[31] The database consists of roughly 15,000 data points from 360 unique compounds. Numerous classes of compounds including alkanes, cycloalkanes, alcohols, ketones, noble gases, fluorinated compounds, and other chemical functionalities are included in the database. The experimental data was provided in four main sections: tracer numbers, tracer graphics, NMR numbers and NMR graphics, where NMR refers to nuclear magnetic resonance. Tracer numbers and NMR numbers refer to diffusion constants that were explicitly reported by the authors whereas tracer graphics and NMR graphics refer to diffusion constants that were extracted from figures. The ML dataset used for the current study utilizes a combination of all four types of data and includes experiments at liquid, gas, and supercritical states. The range of self-diffusion constants spans from 10^{-11} to 10^{-3} m^2/s and are given in units of m^2/s as this is the form commonly reported by authors.

Table 1. List of features and abbreviations collected for each molecule. Units are in parenthesis where applicable.

M	**Molar Mass (g/mol)**	**H_d**	**# H-bond Donors**
T_m	**Melting Point (K)**	**F_C**	**Carbon Fraction**
T_b	**Boiling Point (K)**	**F_H**	**Hydrogen Fraction**
T_c	**Critical Temperature (K)**	**F_O**	**Oxygen Fraction**
P_c	**Critical Pressure (bar)**	**logP**	**Log of Partition Coefficient**
V_c	**Critical Volume (cm^3/mol)**	**PSA**	**Polar Surface Area ($Å^2$)**
T	**Experimental Temperature (K)**	**R**	**Eccentricity Radius (atoms)**
P	**Experimental Pressure (bar)**	**SA**	**Shape Attribute**
ρ	**Experimental Density (kg/m^3)**	**SC**	**Shape Coefficient**
H_a	**# H-bond Acceptors**	**Ph**	**Phase/State**

The features used to train the ANNs in this work are listed in **Table 1**. The original database tabulated 9 usable features (M, T_m, T_b, T_c, P_c, V_c, T, P, and ρ). An additional 11 features were added, including 10 (H_a, H_d, F_C, F_H, F_O, logP, PSA, R, SA, and SC) that were predicted using Chem3D from PerkinElmer Informatics. The phase/state (Ph) of each experimental point was also included as an input feature. The National Institute of Science and Technology (NIST) online database was used to fill in missing density and pressure values and the phase/state for each experimental diffusion value.[32] The phase state feature was broken into 5 categories, labelled 0 through 4. In the database, liquid points are assigned to 0, supercritical points are 1 and gas points are 2. Some authors explicitly state that experiments were performed at or near a phase change. Diffusion constants measured at liquid to gas phase changes are assigned a 3. Many experiments were performed to observe the diffusion behavior across a gas/supercritical phase change. In these sets of data, the point(s) closest to the critical pressure are assigned to 4. In general, the experimental pressure is less than a 1 bar difference from the critical pressure, but some cases are as high as 3 bar. For atomic fractions (F_C, F_H, F_O), if a compound did not contain one of the species it was marked as a zero (e.g., argon would have a zero for all three of these atomic fraction input features).

For many compounds, the density could not be determined at the given experimental conditions as the P-V-T data was not readily available. Any points where the density, or any other feature, could not be determined were left out, resulting in a final database containing 7221 points (DB1). Because density and pressure are highly correlated, a separate database was created that did not contain density as an input feature. The database without density (DB2) contains 11,537 points as authors more often reported the pressure. Both databases are provided in the supplementary information as Excel files.

2.2 Data Selection

The experimental self-diffusion data was inspected and cleaned before any ML models were developed. There are 5 clear outliers that are noticeable when the log of diffusion is plotted versus density (**Figure S1**). The outliers come from three different studies: Winn in 1950 studying methane and oxygen,[33] Paul and Watson in 1966 studying ammonia,[34] and Beatty in 1969 studying n-pentane.[35] Beatty published two values at the same conditions, which is why there are only 4 clear outliers on the plot. These points all exhibit relatively high densities and high diffusivity values. When compared to similar results in the database, the outlier diffusion values are much larger. We expect that these studies may have had a mixture of both liquid and gas phases present during their experimental measurements leading to the anomalously large diffusion constants. These were the only experimental points removed from the databases prior to ML model development.

Prior to ANN training, the databases were also checked for any duplicate values. Any instance where a compound was tested at the same experimental temperature and pressure was considered a duplicate. Duplicate points were combined, and the mean of the diffusion value was taken. All models use the mean diffusion values

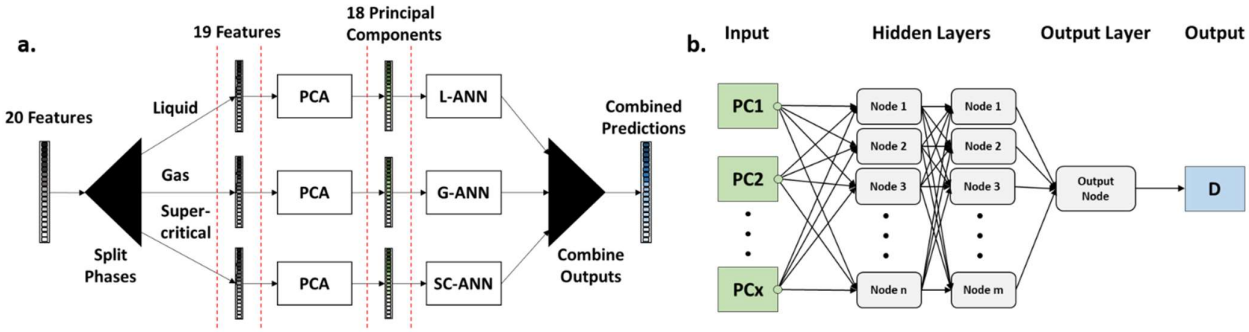

Figure 1. a) Process flow for the phase-specific ANNs and b) general form of an Artificial Neural Network showing two hidden layers with n and m nodes, respectively. PCx is either the 18th or 19th principal component depending on the model. When the log distribution is applied to self-diffusion, the output from the ANN model becomes log10(D).

where applicable. The final sizes of DB1 and DB2 were 6,625 and 10,569 points, respectively.

2.3 Data Pre-processing

All ANN models used principal components (PCs) as inputs rather than the raw features. This was done to remove any correlations among the features. Principal component analysis (PCA) was performed in MATLAB R2018b (MathWorks, Natick, MA). The 20 input features selected (Table 1) were scaled prior to the PCA using the standard scaling method:

$$x' = \frac{(x - \mu)}{\sigma} \quad (1)$$

where x' is the scaled value, x is the raw value, μ and σ are the mean and standard deviation, respectively. For models with all phases combined, 19 PCs were generated and used as input features. The explained variance for each principal component when using DB1 can be seen in **Figure S2**. Similar behaviour is observed when the PCA is applied to DB2.

For the multi-ANN model, an individual PCA was performed for each phase-specific dataset. The phase/state feature was removed from these phase-specific models as it is constant, resulting in 18 PCs used as input (**Figure 1a**). There are instances where a subset of PCs is used as the input features to reduce the dimensionality of the problem. Here, we chose to retain all PCs for model training as it has been argued that even low-variance PCs can contribute greatly to predictions in regression problems.[36, 37] We did in fact see an improvement from including these low-variance PCs, shown in **Table S1**.

The target diffusion values are scaled using a minmax method:

$$y' = \frac{y - \min(y)}{\max(y) - \min(y)} \quad (2)$$

where y' is the scaled value, y is the raw value, $\min(y)$ and $\max(y)$ are the minimum and maximum diffusion values, respectively. In certain models, the (base 10) logarithm of the self-diffusion coefficient is taken before the minmax scaling. These models are identified in the results section.

The scaling was performed on the entire dataset prior to splitting into training, validation, and test datasets in ratios of 70%/15%/15%, respectively. The subsets are split randomly using the MATLAB 'randperm' function. To ensure reproducibility and allow comparison between models, a random seed of 50 is set before 'randperm'. The test dataset was not used during the development (i.e. training) of the ANN models and thus provides a measure the predictive performance.

2.4 Artificial Neural Network Models

ANN models were developed using the deep learning toolbox in MATLAB. An ANN is an ML algorithm that iteratively adjusts a set of weights and biases in an interconnected node structure to best fit a set of data (**Figure 1b**). The ANN inputs (i.e., PCs) are fed to a hidden layer with a specified number of nodes, each with its own weights and bias.

For the ANN models the hidden layers are fully connected, meaning each neuron (node) is connected to all neurons in the previous layer. The relationships between the neurons is linear:

$$y_j^{(k)} = F\left(b_j^{(k)} + \sum_{i=1}^{n} w_{ij}^{(k)} x_j \right) \quad (3)$$

where $y_j^{(k)}$ is the output for neuron j in layer k, and x_j are the i different inputs to that neuron, $w_{ij}^{(k)}$ are the weightings of input i for neuron j, and $b_j^{(k)}$ is the bias for that neuron (i.e., node) in that layer. The activation function F is applied to all the neurons in a layer and is used to introduce non-linearity to the ANN model. For the first hidden layer, the inputs (x_i) to the nodes are the PC inputs while for subsequent layers the inputs are the outputs from the neurons in the previous (k - 1) layer. The Levenberg-Marquardt (LM) optimization method was used to minimize the mean-squared-error (MSE).[38, 39]

Optimization of the ANNs was performed by varying model hyperparameters. The validation set is used to assess the model performance during optimization and the combination of hyperparameters that results in the lowest validation error is deemed "optimal". The starting setup for optimization was a single hidden layer, 10 hidden layer nodes, a hyperbolic tangent activation function, and a default learning rate of 0.001. The first parameter varied was the random seed, which controls the initialization of the weights and biases. The random seed was varied from 1-100 and the best value was kept constant throughout the remainder of the model development. The structure of the hidden layer nodes was then optimized by performing a grid search. For this work, models with two hidden layers were developed with the nodes in each layer being varied from 1-25. The 'tansig', 'logsig', and 'poslin' activation functions were tested for the hidden layers and the output layer uses a 'purelin' activation function. The default value for the learning rate was kept for all ANNs (0.001).

When assessing the model performance on the test set, the train and validation sets are combined to provide more training data. Because MATLAB automatically uses an early-stopping method, the number of epochs must be set when combining the train and validation sets. Once the where the early stopping occurs. This is used as a starting point when the validation set is re-incorporated back into the train set. Then, the performance curves are used to see whether the model is sufficiently trained. Parameters for the optimized models are listed in **Table S2** in the supporting optimal hyperparameters have been chosen, the model is run to see information.

2.6 Performance Metrics

Multiple metrics were used to assess the performance of the ANNs. The mean-squared-error (MSE) was used as the loss function when optimizing and training the models:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(D_{exp,i} - D_{pred,i}\right)^2 \quad (4)$$

where n is the number of observations, $D_{exp,i}$ is the experimental diffusion constant and $D_{pred,i}$ is the predicted diffusion constant. Reported MSE values are calculated on the minmax scaled diffusion values. For models using log transformed diffusion, the MSE is reported on the log10 and minmax scaled diffusion constants.

The correlation coefficient (R^2) is reported for all models and is calculated on the entire dataset:

$$R^2 = \left(\frac{\sum_{i=1}^{n}\left(D_{exp,i} - \overline{D}_{exp}\right)\left(D_{pred,i} - \overline{D}_{pred}\right)}{\sqrt{\sum_{i=1}^{n}\left(D_{exp,i} - \overline{D}_{exp}\right)^2\left(D_{pred,i} - \overline{D}_{pred}\right)^2}}\right)^2 \quad (5)$$

where $\overline{D}_{exp}$ and $\overline{D}_{pred}$ are the mean of the experimental diffusion and predicted diffusion, respectively. All R^2 values are calculated using the minmax scaled self-diffusion values.

The average absolute deviation (AAD) is calculated to compare directly to previous empirical models and provide an alternative view of the model performance:

$$\mathrm{AAD}(\%) = \frac{100}{n}\sum_{i=1}^{n}\frac{\left|D_{exp,i} - D_{pred,i}\right|}{D_{exp,i}} \quad (6)$$

All AAD values are calculated using the raw, unscaled self-diffusion values. MSE and AAD are reported for the train and test sets individually, as well as the entire dataset to assess model performance.

3 Results and Discussion

3.1 Single ANN Baseline

Initially, a single ANN was developed on DB1 using the raw diffusion constants (distribution shown in **Figure S3a**). This will act as the baseline to compare all other models to and will be referred to as B-ANN (baseline-ANN). The optimized B-ANN has two hidden layers with 4 nodes in each layer. The correlation plot is shown in **Figure 2**.

Table 2. Performance metrics for each model developed on DB1.

		Model					
	Metric	**B-ANN**	**Log-ANN**	**L-ANN**	**SC-ANN**	**G-ANN**	**Multi-ANN**
Train	AAD(%)	1710.1	6.5	12.0	7.9	42.7	17.5
	MSE	8.6e-6	3.45e-5	3.8e-6	4.1e-4	1.5e-6	7.0e-5
Test	AAD(%)	1827.1	7.1	14.8	9.4	39.9	21.0
	MSE	2.6e-6	4.0e-5	7.7e-6	8.0e-4	1.6e-6	1.4e-4
All	AAD(%)	1727.6	6.6	12.5	8.1	42.3	18.0
	R^2	0.9836	0.9911	0.9977	0.9767	0.9997	0.9997

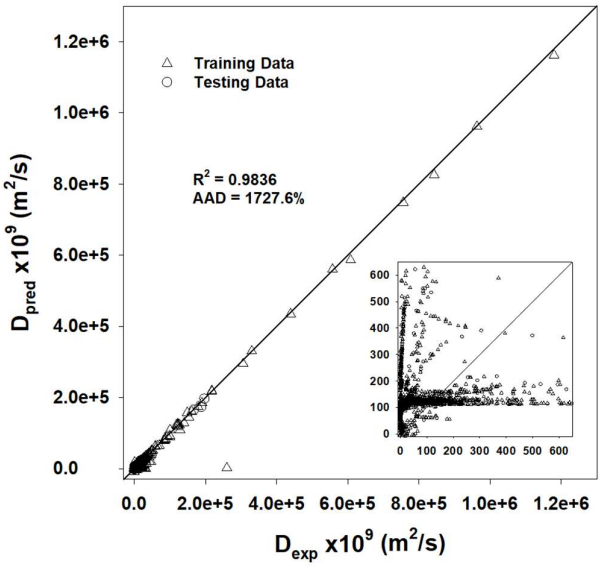

Figure 2. Correlation plot for the B-ANN with the experimental self-diffusion values on the x-axis and predicted self-diffusion values on the y-axis. The solid indicates the 1:1 line. The inset has the same units as the main figure.

The test set MSE for the B-ANN is $2.6\text{x}10^{-6}$, which is excellent (**Table 2**). This implies that the model is performing well and is consistent with the initial impression given by the full-scale correlation plot, but these results are misleading. The AAD provides a more representative value of model performance as it considers the percent deviation of all points. The AAD over all data is 1727.6%, revealing that the B-ANN is performing much worse than the MSE would indicate. The low MSE and high AAD can be understood by the subplot in **Figure 2**, which highlights the liquid and supercritical diffusion region. The B-ANN model struggles to capture the diffusion behaviour in this region, leading to an extremely high AAD value. This phenomenon can be attributed to the use of the MSE loss function (Eqn. 4) and the heavy skew in the distribution of the self-diffusion constants, which is applying very high importance to the low-density gas diffusion constants (i.e., large diffusion constants). The easiest way for the optimizer to reduce the MSE is to fit the larger diffusion values and neglect the smaller values. This is a problem as the resulting model is unusable for predicting diffusion constants over the entire liquid, supercritical, and dense gas regions. Alternative loss functions were considered, but the use of the LM optimizer restricted us to loss functions that are twice differentiable.

A similar B-ANN was developed on the larger database that did not include density (DB2). The AAD for the test set was 63,538%, which is significantly worse than the B-ANN with density (**Table S3**). The same problem arises in the low-diffusion region in which the MSE places high importance on the larger gas (low-density) diffusion values. The order-of-magnitude increase in the AAD between the

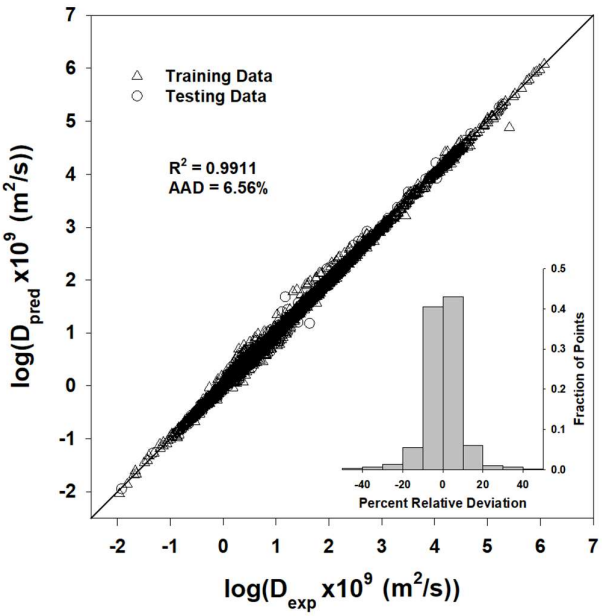

Figure 3. Correlation plot for the log-ANN with the experimental self-diffusion values on the x-axis and predicted self-diffusion values on the y-axis. The solid line indicates the 1:1 line. The inset shows the distribution of errors from -50 to 50 percent deviation. Additional errors are present beyond -50 but are left out for clarity.

two models also indicates that density is a more important feature than pressure and is pivotal for model development.

3.2 Single ANN with log transformation

The heavy skew in the range of self-diffusion rates for both the DB1 and DB2 datasets is analogous to using an imbalanced dataset in classification problems. Multiple groups have studied the use of imbalanced datasets and suggested solutions,[40-42] but little analysis has been done concerning heavy skew in regression datasets. One common theme is to perform a log transformation of the target values to produce a more uniform distribution. This allows the MSE loss function to improve optimization for the low-diffusion values during the ANN training. A (base 10) log transformation was applied to the diffusion values in DB1 and the distribution before and after can be seen in **Figure S3**. The heavy skew is successfully transformed into a more suitable distribution for the MSE loss function with a smaller range. The transformed diffusion values were then used as targets to develop an ANN (log-ANN). The log10 transformation is done only to improve model performance. All output values from the log-ANN are returned to their original scale before comparing to experiment and calculating the AAD.

The optimized log ANN has two hidden layers with 18 and 15 nodes, respectively. The correlation plot is shown with the log transformation applied (**Figure 3**). With an R^2 value of 0.9911 the correlation between the predicted and experimental values is in good agreement. The test set MSE and AAD are $4.0x10^{-5}$ and 7.1%, respectively. The MSE for the log-ANN is an order of magnitude larger than that of the B-ANN, which is likely due to the transformation into log space and reduced range.

It is clear from the AAD values alone that the log transformation greatly improved the accuracy of predictions, resulting in an overall AAD that is 3 times smaller than that of the B-ANN. The modified distribution allowed the model to train more effectively on the smaller diffusion values while maintaining accuracy at high values. The improvement is visible when comparing the results visually (**Figures S4 and S5**). When the B-ANN predictions are plotted on a log scale it becomes clear that the smaller diffusion values are poorly fit.

The distribution of relative errors is shown for the log-ANN in the subplot of **Figure 3**. The relative errors calculated after reverting the log transformation show 83.4% of the predictions are within 10% of the experimental values and 97.3% are within 30% of the experimental values. The model tends to over-predict slightly with about 51% of the points being in the negative. The largest deviation is -220 % and comes from a supercritical methane point reported by Jeffries and Drickamer in 1953.[43]

The log transformation was also assessed using the larger database that did not include density (DB2). The log-ANN without density had an overall AAD of 14.1% compared to the 63,538% using the B-ANN (**Table S3**). Again, we see that the log transformation significantly improves the self-diffusion predictions, again by 3 orders of magnitude. The log-ANN performs worse when density is not present in the database, with only 34.8% of points being within 10% of the experimental value and 95.8% of points being within 30%. The importance of density as a feature is again emphasized by these results and is consistent with previous efforts to develop empirical relationships (supplemental materials Eqs. S1-26)

3.3 Phase-Specific Multi-ANN

An alternative approach to taking the logarithm of the self-diffusion constant is to split the data and develop multiple ANNs on smaller subsets of raw diffusion values. By splitting the data, the range of diffusion values used in each ANN can be minimized, and some accuracy should be gained by training on the diffusion values directly rather than the log transformed values. We chose to split the data by phase and have separate ANNs for liquids (L), supercritical fluids (SC), and gases (G). The phases/states were identified using the NIST online database, but like density, many compounds do not have available phase data. Therefore, this approach was only assessed on the density database (DB1), which contains compounds with available P-V-T and phase data. Points that were labelled as 3 or 4 (near a phase transition) were not included in the phase-specific models.

The liquid-only ANN (L-ANN) contained 70.9% of the original diffusion data points with 4698 out of the 6625. Even after removing the gas and supercritical points, the distribution of diffusion remains skewed, but the range is significantly reduced. In the full dataset, the largest diffusion value is on the order of 10^{-3} m^2/s and in the liquid-only subset, the largest diffusion value is on the order of 10^{-7} m^2/s.

The optimized L-ANN has two hidden layers with 19 and 17 nodes, respectively. The MSE and AAD for the test set were $7.7x10^{-6}$ and 14.8%, respectively (Table 2). The overall AAD for the liquid data is 12.5%. The model predictions showed extremely good correlation with experimental values, with an R^2 value of 0.9976 (**Figure 4a**). The improvement over the B-ANN is immediately noticeable when comparing to the subplot in Figure 2 (approximately same D range). The separation of the liquid points from the gas and supercritical points allows the L-ANN to optimize on the smaller diffusion values.

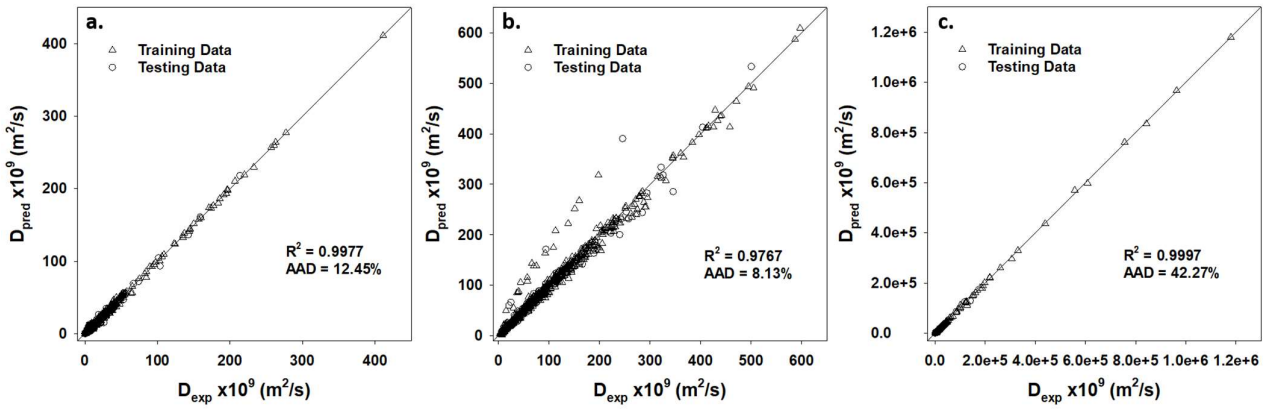

Figure 4. Correlation plots of experimental and predicted self-diffusion for the a) L-ANN (liquids), b) SC-ANN (super critical), and c) G-ANN (gas). The solid lines indicate the 1:1 line.

The supercritical ANN (SC-ANN) contains 16.1% of the original data points with 1068 out of the 6625. The supercritical subset spans an even smaller range than the liquid-only subset. The smallest diffusion value in the liquid-only subset was on the order of 10^{-11} m^2/s while the smallest in the supercritical subset was on the order of 10^{-9} m^2/s. Regardless, a heavy right skew was still present in the range of diffusion.

The SC-ANN has two hidden layers with 20 and 7 nodes in layers 1 and 2, respectively. The MSE for the test set was 8.0×10^{-4}, significantly higher than any of the other ANNs. The majority of supercritical points are well-predicted, but there is a distinct branch of diffusion points that breaks away from the 1:1 line (**Figure 4b**). The points in this branch all originate from the same 1953 study performed by Jeffries and Drickamer,[43] but were not identified or removed during the initial data cleaning. The same branch of points can be seen when the predictions from the log-ANN are reverted to the original scale (subplot in **Figure S5**). Considering these points are outliers in both the log-ANN and SC-ANN, it is likely there are large inaccuracies in the experimental data or a scaling factor for that study has not been identified. Even with those outliers, the performance of the ANN model is good with a test AAD of 9.4% and an overall AAD of 8.1%. The AAD ends up being smaller than that of the L-ANN due to a smaller range of diffusion. With the liquid subset having a lower minimum diffusion, the effects of the MSE loss function become more pronounced, leading to the increased AAD (**Figure S6a and b**).

The gas ANN (G-ANN) had the smallest subset of data, containing only 770 of the total 6625 data points. Although the liquid and supercritical diffusion values had been removed, the smallest diffusion values are still on the order of 10^{-8} m^2/s. The resulting range of diffusion is equivalent to the B-ANN model, which used all of the experimental self-diffusion data.

The optimized G-ANN has two hidden layers with 8 and 3 nodes, respectively. The correlation plot for the G-ANN is similar to the B-ANN and suffers from the same issues in the low-diffusion region (**Figure 4c**). The large range causes the MSE loss function to neglect the smaller values and focus on the larger diffusion constants (see **Figure S6c**). This is likely why the test AAD for the G-ANN is significantly larger than the liquid and supercritical ANNs at 39.9%. A log transformation would likely improve these results, as shown with the previous log-ANN.

Although the performance of each individual ANN still suffers from skew in the distribution of diffusion, the approach was successful in improving the accuracy of the predictions. With all three models combined, the overall AAD was improved by 2 orders of magnitude, from 1727.6% to 18.03%. The multi-ANN approach also corrected a large xenon outlier that is present in both the B-ANN and log-ANN. Between the two approaches, the log-ANN was more successful than the phase-specific ANNs at handling the skew of the experimental data and resulted in a lower AAD and MSE.

3.4 Feature Importance

The 20 features collected in this work (**Table 1**) were chosen based on accessibility and potential impact based on previously published correlations (see discussion in Supplemental Information). We wanted to use features that could easily be obtained for new, unknown chemical compounds allowing efficient incorporation into the model. It is likely that all 20 features do not have equivalent impact for the prediction of diffusion. To assess which features had the largest impact, a feature addition method is employed for the ANN models. The method sequentially adds the original features to an ANN (not the PCs), creating all possible combinations to find the best performing features. Because the log-ANN performed the best, it was used to assess the feature importance. To begin the evaluation, the log-ANN is trained with each individual feature alone and the validation MSEs are recorded. The feature that produces the

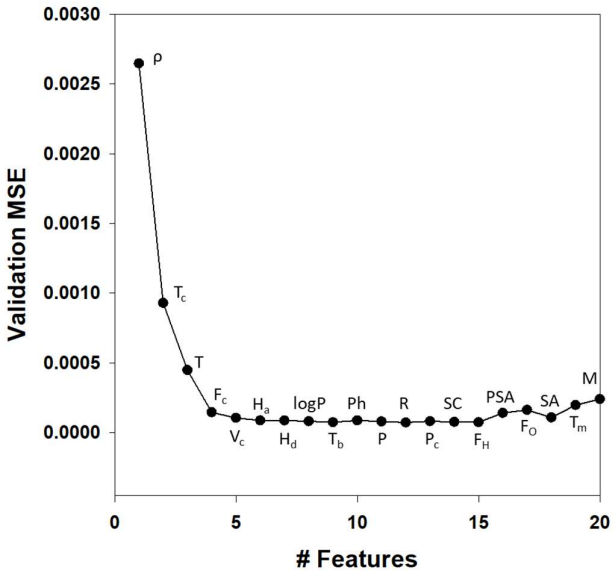

Figure 5. Validation MSE versus number of features during the feature addition process. The text refers to the feature retained at each iteration. See Table 1 for symbol definition.

lowest error is retained and subsequent models are developed with 2, 3 … n features. For example, if temperature produced the best model with one input, two-input models would then be made with temperature and every remaining feature. The process is continued until no features remain.

The results of the feature addition process are shown in **Figure 5**. The single-input ANN model had the lowest validation error when using ρ (density). The relationship between density and diffusion is intuitive and well-documented with the first successful correlation being the kinetic theory of gases.[1] Correlations for dense fluids commonly incorporate density as an input and few successful models can be found that do not consider the density of the system (See Background in SI). Because of the large range of diffusion values, the ANN model is likely picking up on the inverse relationship between density and diffusion.

We also see temperature (T) being the third most important input feature. In contrast to density, the model will observe a direct relationship between T and self-diffusion. It is important to distinguish self- and mutual diffusion, as the temperature would have less of an impact if the model were trying to predict mutual diffusion. The self-diffusion is directly related to the thermal energy of the molecules and therefore the temperature has a large impact on how fast each molecule diffuses. In models where mutual diffusion is the focus, differences in chemical potential may have more of an impact.

In the top 5 features, we see that two are critical properties, T_c and V_c. Although the relationship between the critical properties and self-diffusion is less clear, critical properties have been used in empirical and theoretical models to predict the self-diffusion. For example, Zhu *et. al.* used T_c and V_c to calculate the updated LJ force constants for their pure solution model (Eqs. S22 – S26).[12] The carbon fraction, F_c, is also contributing significantly to the predictions. It is likely that the F_c is providing an idea of the size or length of each molecule, similar to the parameter N in the LJ chain models (Eqs. S5-S7 and S17-S21). We see from the correlation plot in **Figure S7**, that F_c has strong correlation with both the radius (R) and shape attribute (SA). It can be assumed that the relationship between the self-diffusion and F_c is inversely proportional as the larger/longer molecules will experience more hinderance and frictional forces.

Given that the ANN model performance does not improve with more than 5 features, this implies that several features could be removed for ANN development. The features to be incorporated last were F_O, SA, T_m and M, with the latter being the "least important". The F_O (oxygen fraction) is highly correlated to H-bond donors so it may be superfluous information (**Figure S7**). Similarly, M is highly correlated with F_C and T_m is highly correlated with T_c and T_b. The features could be removed to save computational resources, but the PCA feature employed here allows us to retain any additional information they may have while removing the correlations. Also, computational resources were not an issue as all models are able run quickly on a single processor (see Supplemental Information).

3.5 Comparison to Empirical Equations

To compare our log-ANN model to those previously proposed, we used MATLAB to implement the models of Silva *et al.* and Zhu *et al.* (Eqs. S15-16 and S22-26).[11, 12] The LJ4 model proposed by Silva *et al.* is tested against only liquid points while the Zhu *et al.* model is tested against all points in DB1, including the gas and supercritical.

The LJ4 model from Silva takes 4 variables as input: A_D, a shape factor, E_D, an energy parameter, T_D, a temperature parameter and σ, the molecular diameter. The values for these parameters were taken as reported by the authors (Table 3 in Ref. 11). Each of the 41 molecules tested by Silva are present in our database except for deuterated methanol, which did not have available data for its critical properties. A total of 3690 liquid diffusion values are present for the remaining 40 molecules in DB1, compared to the 2471 points used in the original paper. The performance on unseen data is poor with an overall AAD of 92.85% (**Figure S8a**). It is likely that some of the experimental conditions in the new data were not present when Silva *et al.* calculated the molecular parameters. The log-ANN performs much better on this subset of points, with an AAD of 6.69%.

The model developed by Zhu *et al.* was designed and tested on experimental diffusion at liquid, gas, and supercritical states. We were able to test Zhu's model on all points in DB1, to get a direct comparison to the log-ANN. The Zhu model performed well with an overall AAD of 36.37%. The largest deviations come from the low-density gas diffusion values (**Figure S8b**). As reported earlier, the log-ANN achieved an overall AAD of 6.56% on this data, performing better than the Zhu model across the three phases.

The ML approach not only resulted in better prediction of the self-diffusion constants, but also has the advantage of being easily updated when new data is presented. The expectation is that any new compounds or conditions could simply be predicted without any re-training or re-optimization of the ANN. This prediction behaviour of ANNs is also distinct from MD approaches, which require additional simulations to extract molecule and model parameters.

4 Conclusions

An experimental dataset of self-diffusion constants for pure compounds was used to develop artificial neural networks covering liquid, supercritical and gas states. The major challenge of using ANNs to predict over multiple phases is the large range of diffusion constants present across different phases. Using a database with heavy skew in the distribution of the diffusion rates in combination with an MSE loss function leads to the small diffusion values (primarily liquid and supercritical points) being insufficiently weighted and therefore poorly modelled. Two different approaches were explored in this work to address the large range of diffusion values and both proved to be successful in improving the ANN predictions. When individual ANNs are developed for each phase (multi-ANN), the overall AAD is reduced by two orders of magnitude when compared to the original baseline ANN. Performing a log transformation of the diffusion values improved the overall AAD by 3 orders of magnitude and lead to a generalized model that can predict well over multiple compounds and phases. The ANN models that were developed with density as an input feature had superior performance to those without density in all cases and emphasizes the importance of density in modelling self-diffusion.

This work presents generalized ML models for predicting self-diffusion in pure solutions over multiple phases. In future work, these types of ML models could also be developed to predict diffusion in mixture and porous materials. Recent work has shown that scaling relationships exist between the self-diffusion of bulk fluids and the self-diffusion of those fluids in porous materials.[44-47] For large pores and materials with a small interaction energy with the absorbed fluid, the diffusivity is a direct function of local pore density, excess entropy, or filling fraction and is approximately equal to the diffusivity of the bulk liquid. By utilizing these scaling relationships, the ANN models developed here, for pure solutions, provide very good initial guesses for the diffusivity. With increasing surface interactions and decreased pore size this relationship becomes more

complicated but highlights an area for future ML model development.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This research was supported entirely through the Sandia Laboratory Directed Research Development (LDRD) program. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

1. S. Chapman, T. G. Cowling, D. Burnett and C. Cercignani, *The Mathematical Theory of Non-uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*, Cambridge University Press, 1990.
2. H. Liu and E. Ruckenstein, *Ind. Eng. Chem. Res.*, 1997, **36**, 888-895.
3. H. Liu, C. M. Silva and E. A. Macedo, *Chem. Eng. Sci.*, 1998, **53**, 2403-2422.
4. P. H. Salim and M. A. Trebble, *J. Chem. Soc., Faraday Trans.*, 1995, **91**, 245-250.
5. Q. Zhong, L. Yang, Y. Tao, C. Luo, Z. Xu and T. Xi, *Int. J. Thermophys.*, 2015, **36**, 1405-1415.
6. A. E. Nasrabad, R. Laghaei and B. C. Eu, *The Journal of Physical Chemistry B*, 2005, **109**, 8171-8179.
7. J. S. Vrentas, J. L. Duda, H.-C. Ling and A.-C. Hou, *J. Polym. Sci., Part B: Polym. Phys.*, 1985, **23**, 289-304.
8. M. Hopp, J. Mele and J. Gross, *Ind. Eng. Chem. Res.*, 2018, **57**, 12942-12950.
9. Y. Liu, J. Fu and J. Wu, *Langmuir*, 2013, **29**, 12997-13002.
10. R. V. Vaz, A. L. Magalhães, D. L. A. Fernandes and C. M. Silva, *Chem. Eng. Sci.*, 2012, **79**, 153-162.
11. C. M. Silva, H. Liu and E. A. Macedo, *Chem. Eng. Sci.*, 1998, **53**, 2423-2429.
12. Y. Zhu, X. Lu, J. Zhou, Y. Wang and J. Shi, *Fluid Phase Equilib.*, 2002, **194-197**, 1141-1159.
13. H. Lee and G. Thodos, *Ind. Eng. Chem. Res.*, 1988, **27**, 992-997.
14. O. Suárez-Iglesias, I. Medina, C. Pizarro and J. L. Bueno, *Chem. Eng. Sci.*, 2007, **62**, 6499-6515.
15. L. Chen, H. Tran, R. Batra, C. Kim and R. Ramprasad, *Comput. Mater. Sci.*, 2019, **170**, 109155.
16. X. Zhang, J. Cui, K. Zhang, J. Wu and Y. Lee, *J. Chem. Inf. Model.*, 2019, **59**, 4636-4644.
17. T. Varol, A. Canakci and S. Ozsahin, *Composites, Part B*, 2013, **54**, 224-233.
18. S. A. Mirkhani, F. Gharagheizi and M. Sattari, *Chemosphere*, 2012, **86**, 959-966.
19. R. Eslamloueyan and M. H. Khademi, *Chemom. Intell. Lab. Syst.*, 2010, **104**, 195-204.
20. M. Naima, L. Khaouane, Y. Ammi, S. Hanini, M. Laidi and H. Zentou, *Kem. Ind.*, 2019, **68**.
21. F. Gharagheizi, *Ind. Eng. Chem. Res.*, 2012, **51**, 2797-2803.
22. F. Gharagheizi, A. Eslamimanesh, A. H. Mohammadi and D. Richon, *J. Chem. Eng. Data*, 2011, **56**, 1741-1750.
23. A. Khajeh and M. R. Rasaei, *Struct. Chem.*, 2012, **23**, 399-406.
24. R. Beigzadeh, M. Rahimi and S. R. Shabanian, *Fluid Phase Equilib.*, 2012, **331**, 48-57.
25. A. Abbasi and R. Eslamloueyan, *Chemom. Intell. Lab. Syst.*, 2014, **132**, 39-51.
26. K. K. Rao, Y. Yao and L. C. Grabow, *Adv. Theory Simul.*, 2020, **3**, 2000097.
27. H. Wu, A. Lorenson, B. Anderson, L. Witteman, H. Wu, B. Meredig and D. Morgan, *Comput. Mater. Sci.*, 2017, **134**, 160-165.
28. Y. Zeng, Q. Li and K. Bai, *Comput. Mater. Sci.*, 2018, **144**, 232-247.
29. J. P. Allers, J. A. Harvey, F. H. Garzon and T. M. Alam, *J. Chem. Phys.*, 2020, **153**, 034102.
30. C. J. Leverant, J. A. Harvey and T. M. Alam, *J. Phys. Chem. Lett.*, 2020, DOI: 10.1021/acs.jpclett.0c03108, 10375-10381.
31. O. Suárez-Iglesias, I. Medina, M. d. l. Á. Sanz, C. Pizarro and J. L. Bueno, *J. Chem. Eng. Data*, 2015, **60**, 2757-2817.
32. P. J. L. a. W. G. Mallard, NIST Chemistry WebBook, NIST Standard Reference Database Number 69, https://doi.org/10.18434/T4D303, (accessed November 12, 2020).
33. E. B. Winn, *Phys. Rev.*, 1950, **80**, 1024-1027.
34. R. Paul and W. W. Watson, *J. Chem. Phys.*, 1966, **45**, 2675-2677.
35. J. W. Beatty, *J. Chem. Phys.*, 1969, **51**, 4673-4674.
36. I. T. Jolliffe, *J. Royal Stat. Soc.*, 1982, **31**, 300-303.
37. A. S. Hadi and R. F. Ling, *Am. Stat.*, 1998, **52**, 15-19.
38. D. W. Marquardt, *J. Soc. Ind. Appl. Math.*, 1963, **11**, 431-441.
39. K. Levenberg, *Q. Appl. Math.*, 1944, **2**, 164-168.
40. S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y. Lu, S. Chen and M. Shyu, presented in part at the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, Fl, 2018.
41. S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng and P. Kennedy, presented in part at the 2016 International Joint Conference on Neural Networks (IJCNN), 2016.
42. T. Vandal, E. Kodra, J. Dy, S. Ganguly, R. Nemani and A. R. Ganguly, presented in part at the Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, United Kingdom, 2018.
43. Q. R. Jeffries and H. G. Drickamer, *J. Chem. Phys.*, 1953, **21**, 1358-1358.
44. W. P. Krekelberg, D. W. Siderius, V. K. Shen, T. M. Truskett and J. R. Errington, *Langmuir*, 2013, **29**, 14527-14535.

45. W. P. Krekelberg, D. W. Siderius, V. K. Shen, T. M. Truskett and J. R. Errington, *J. Phys. Chem. C*, 2017, **121**, 16316-16327.
46. J. Mittal, J. R. Errington and T. M. Truskett, *Phys. Rev. Lett.*, 2006, **96**, 177804.
47. J. Mittal, J. R. Errington and T. M. Truskett, *J. Phys. Chem. B*, 2007, **111**, 10054-10063.