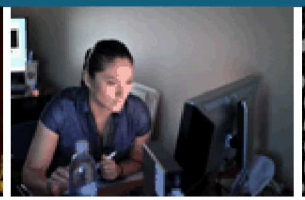
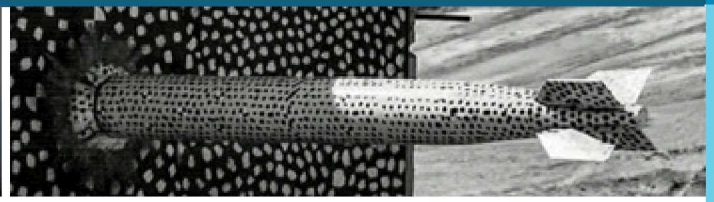
SAND2019-14692PE

Using Machine Learning to Predict Diffusion in Lennard Jones Fluids





Josh Allers

This work was funded through the Sandia LDRD program. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government. SAND...

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.

Problem and Motivation

Problem: Unlike gas phase, no universal kinetic theory for diffusion in multi-component mixtures (MCM) exists. The *complexity* of the problem only increases for MCM absorbed into porous materials!

Need: Need the ability to rapidly and accurately predict MCM diffusion for a wide range of materials and conditions. Diffusion impacts material performance, production and aging.

- **Experimental Methods:** (absorption, CT, QENS, NMR diffusometry etc.). Time consuming, difficult for MCM and temperature. Limited number of materials.
- **Molecular Dynamic (MD) Simulations**: Detailed information. Simulations for different loadings, temperatures, or materials becomes extensively time restrictive for engineering applications.

Solution: Develop machine learning (ML) methods for predicting diffusion.

- **Machine Learning:** Rapidly growing field in Materials Science (Future efforts needed at Sandia!)
  - Materials Genome
  - Computational Materials Engineering Initiative
  - Ionic liquid Genome
  - Polymer Genome
  - Electrolyte Genome
- **Surrogate Model Development:** Fundamental understanding of diffusion in MCM from ML.

Why LDRD?: From a materials scientist's perspective - Need rapid accurate models, applied for infinite number of MCM and materials … Need to develop fundamental and broadly applicable understanding of diffusion.

Current Theories for Diffusion in Mixtures

Maxwell Stephen (MS) Model
(dominant diffusion model)

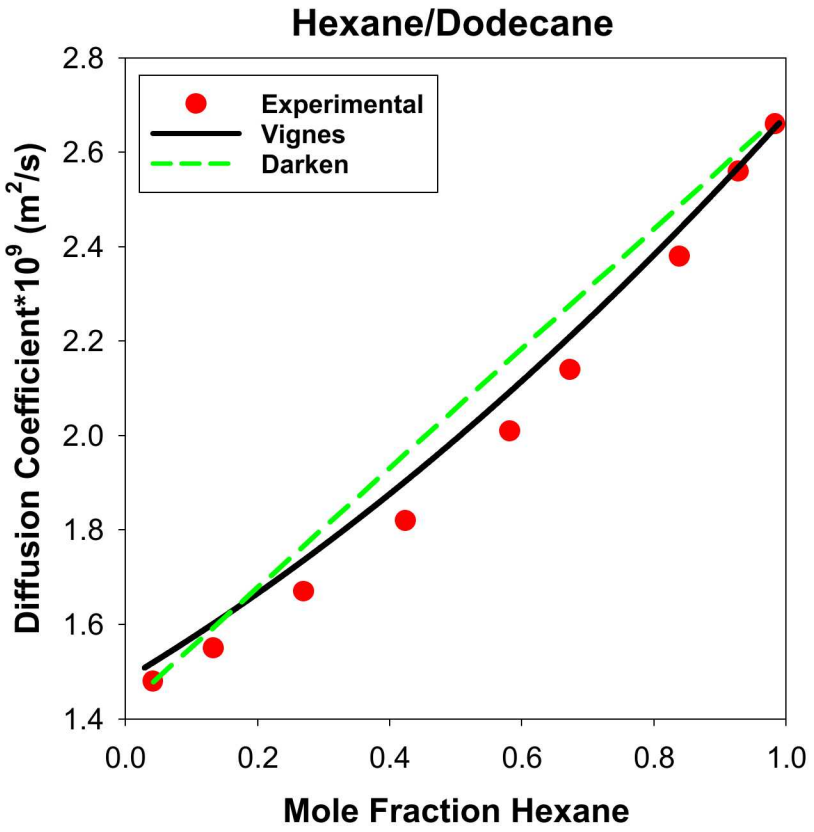
$$\frac{1}{D_{\mathrm{i,self,s}}} = \underbrace{\frac{1}{D_{i,s}}}_{\text{surface}} + \underbrace{\frac{x_i}{D_{ii}}}_{\text{self}} + \underbrace{\frac{x_j}{D_{ij}}}_{\text{exchange}}$$

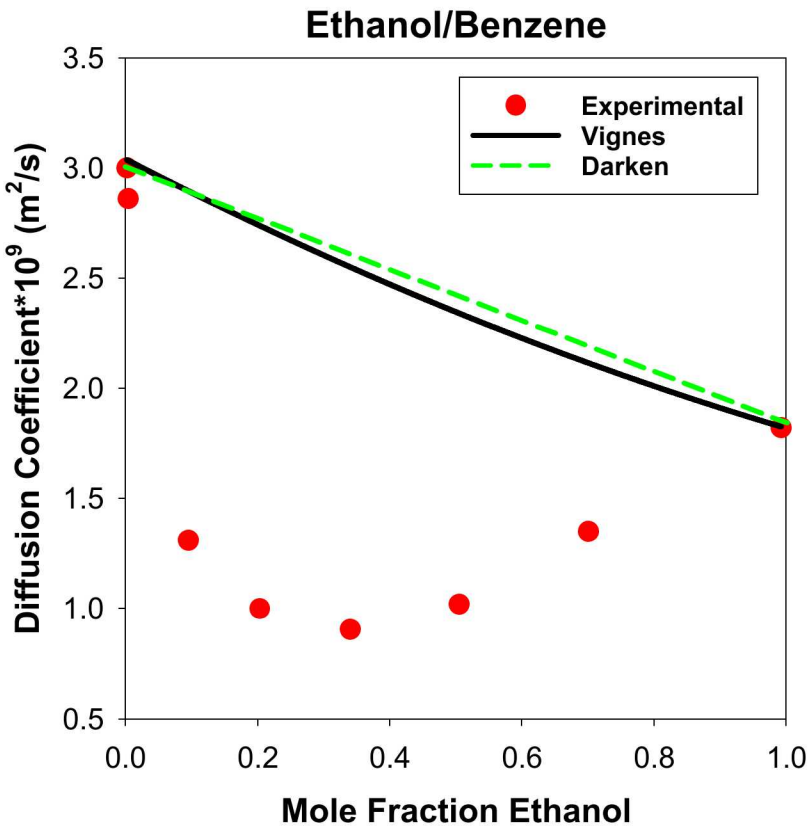
Darken Relationship
(semi-empirical - linear)

$$D_{ij} = D_i D_j \sum_{k=1}^{N} \frac{x_k}{D_k}$$

Vignes Model
(semi-empirical- power law)

$$D_{ij} = \left(D_{ij}^{x_j \to 1}\right)^{x_i} \left(D_{ij}^{x_j \to 1}\right)^{x_j} \prod_{k,k\neq i,j}^{i=N} \left(D_{ij}^{x_k \to 1}\right)^{x_k}$$

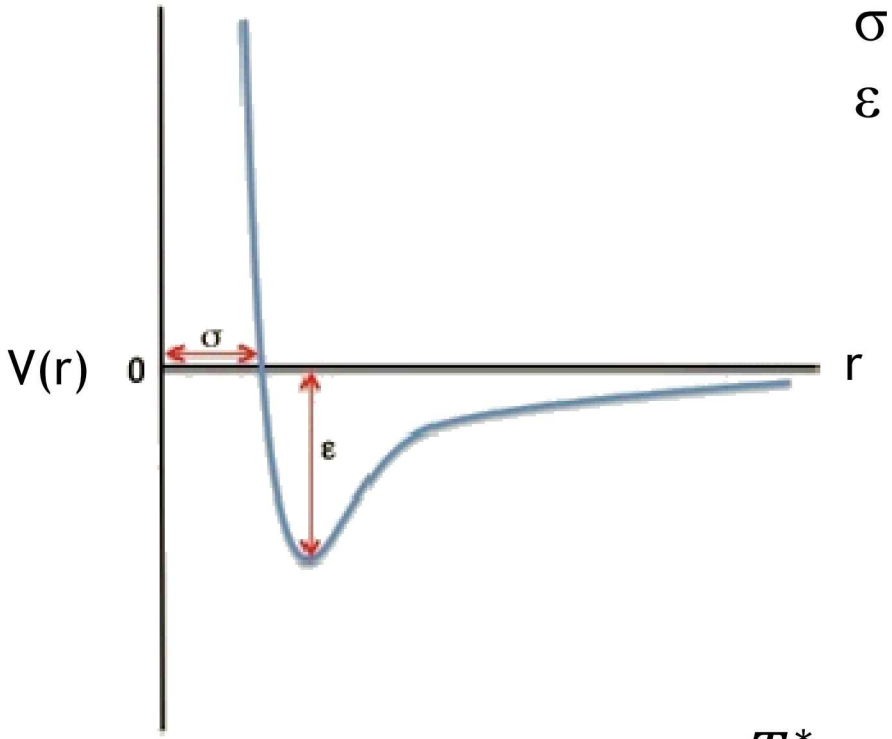




Need to start including things like:

- $\Delta E_{i,j\ \text{interaction}}$
- $\Delta E_{surface}$
- Pore Diamater

Start Small - Lennard Jones Fluids



σ – Distance where potential becomes zero

ε – Well depth and measure of attractive force

$$V(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$$

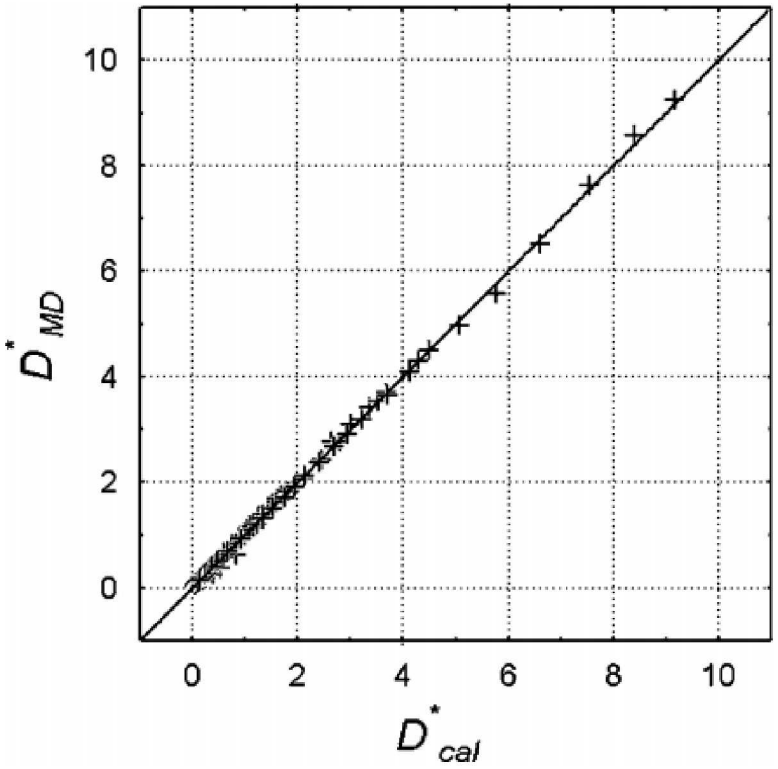
$$T^* = \frac{Tk}{\varepsilon}$$

$$D^* = D\frac{\sqrt{\frac{m}{\varepsilon}}}{\sigma}$$

$$\rho^* = \frac{N\sigma^3}{V}$$

Molecular Dynamics and Lennard Jones

- Zhu *et al*. collected multiple sets of MD data from other authors
- Fit the data to generate 9 parameters used in his equation



$$D^*_{cal} = \frac{3\sqrt{T^*}}{8\rho^*\sqrt{\pi}} A \times B$$

$$A = \left(1 - \frac{\rho^*}{a(T^*)^b}\right)\left[1 + (\rho^*)^c\left(\frac{P1(\rho^* - 1)}{P2(\rho^* - 1) + (T^*)^{(P3+P4\rho^*)}} + P5\right)\right]$$

$$B = exp\left(\frac{-\rho^*}{2T^*}\right)$$

Zhu *et al*. *Fluid Phase Equilibria*. **2002**.

The Database

Zhu's Dataset

550 data points from 8 sources

Parameters were fit using data from only one source (Rowley 1997)

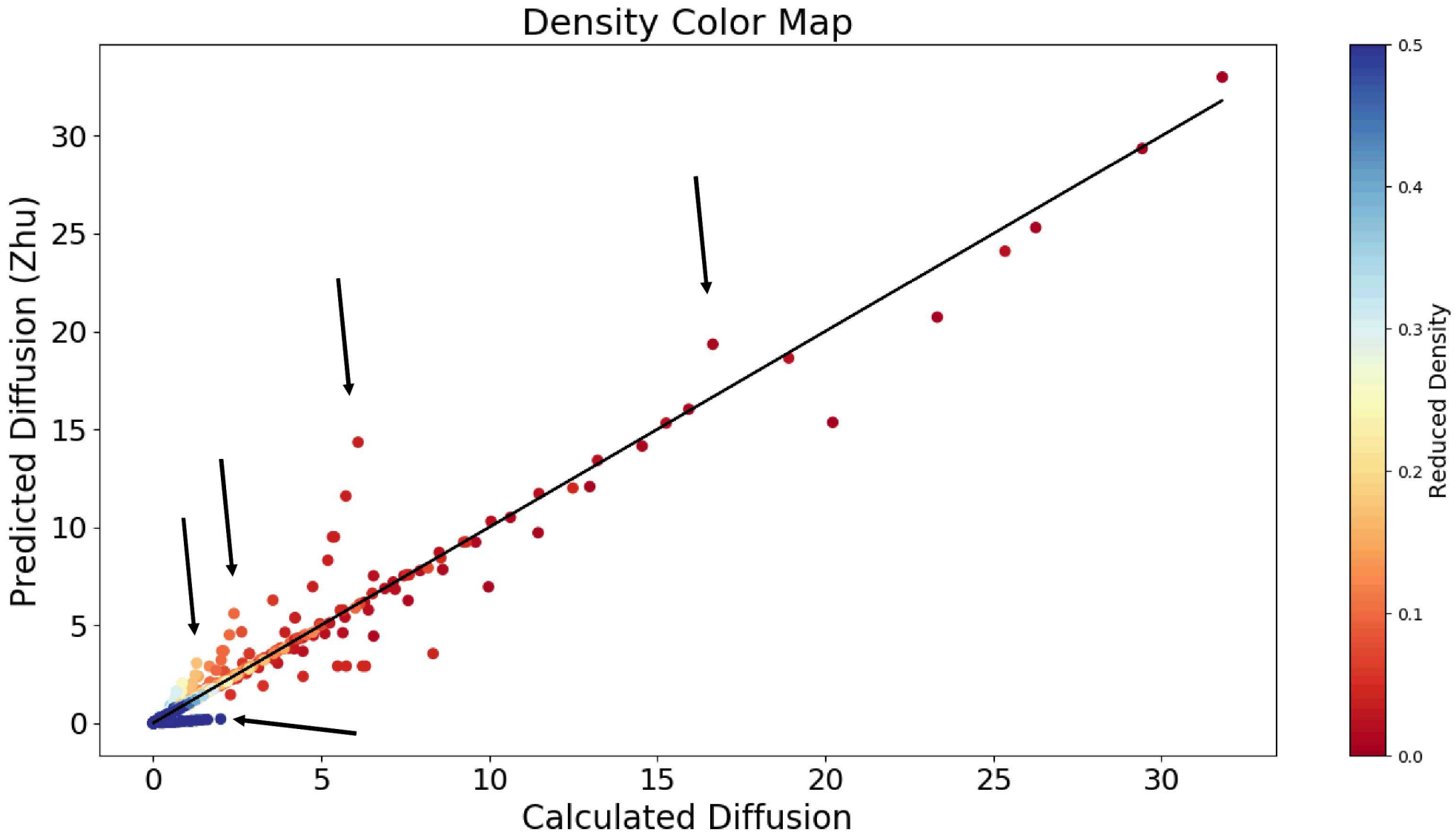
Total Dataset

Originally 1183 data points from 19 sources

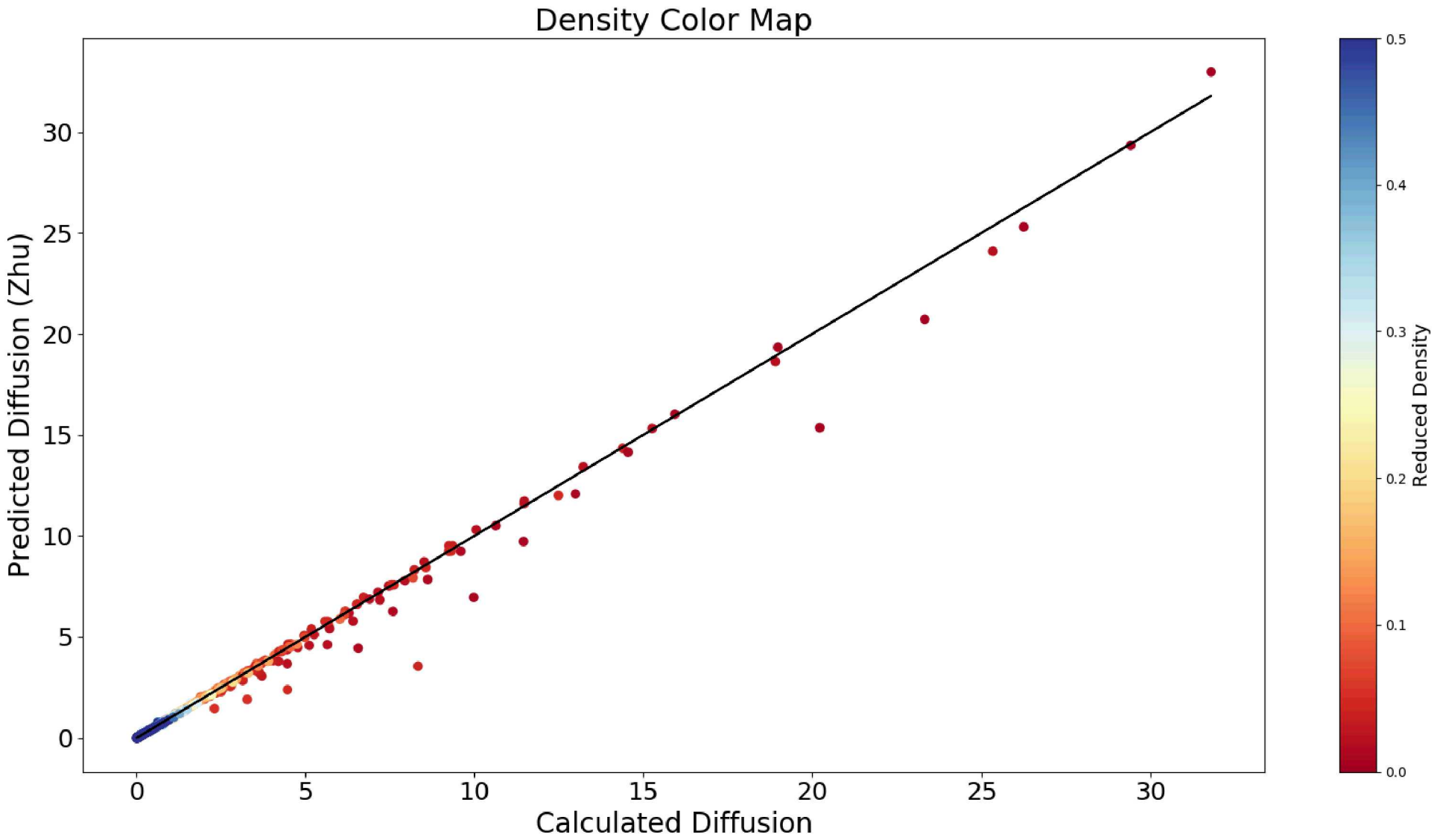
16 Argon studies, 1 Lithium, 1 unity, 1 unknown

Trimmed down to 884 data points after cleaning

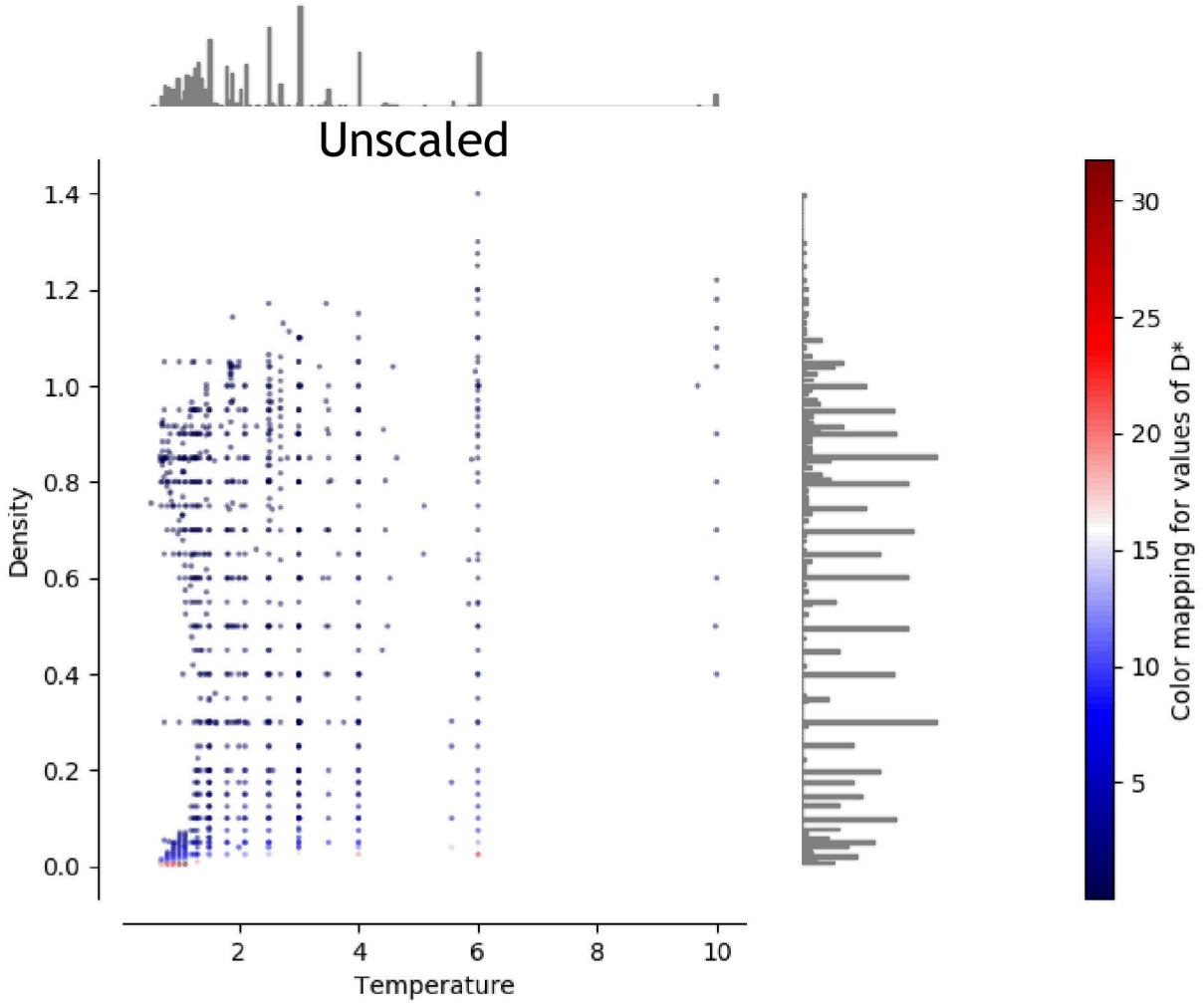
Raw Dataset


Density Color Map
Predicted Diffusion (Zhu)
Calculated Diffusion
Reduced Density
0
5
10
15
20
25
30
0.0
0.1
0.2
0.3
0.4
0.5

Current Dataset After Being Cleaned

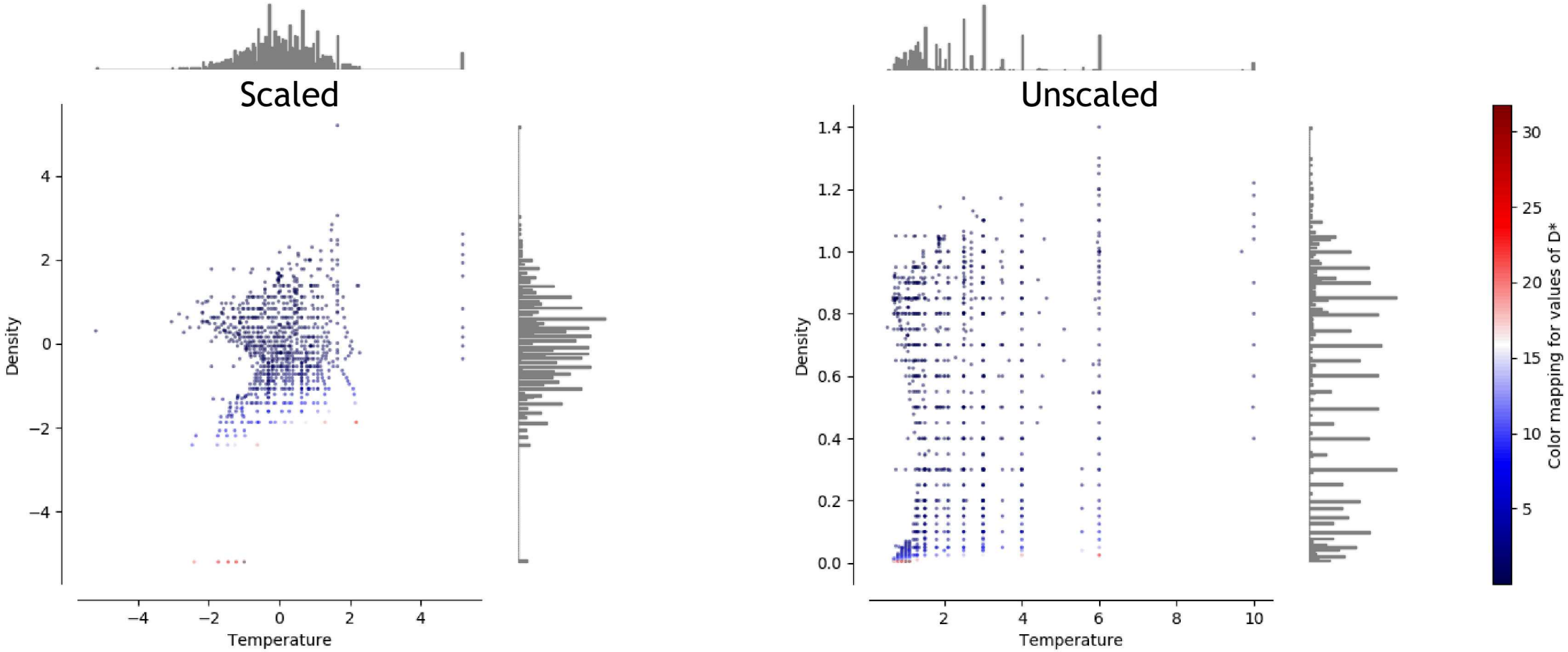


Distribution of Data



- Majority of T* values are below 5
- Distribution of rho* is more even

Scaling the Data - Normal Distribution

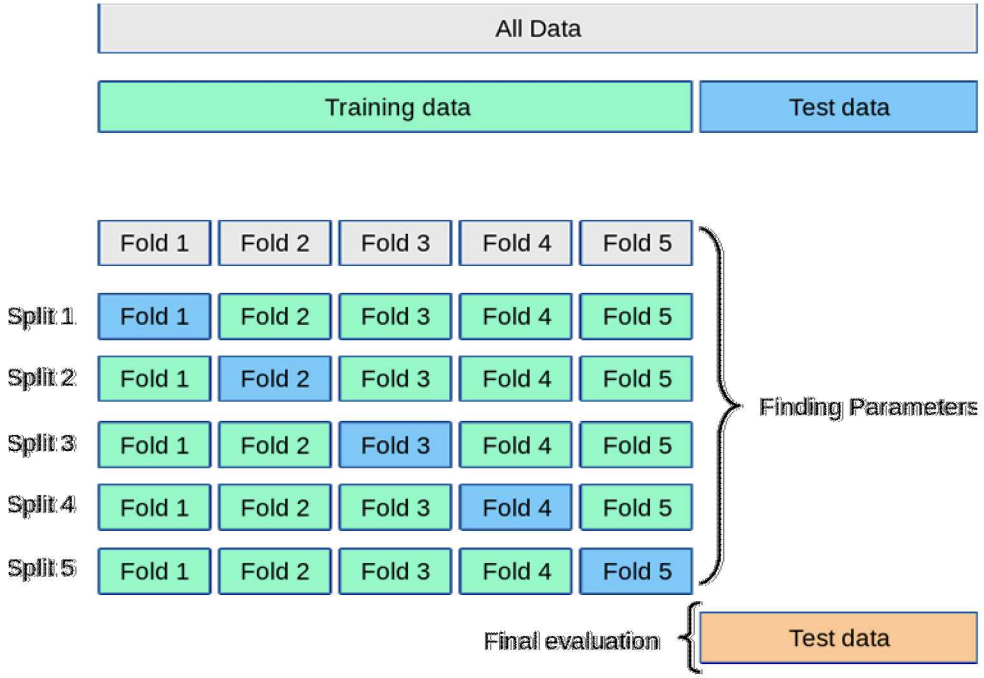


Using Kfold to Assess the Performance

Splits dataset in even “folds”

- 10% of the data points become test points
- Other 90% is used to train model
- 10 “folds” are created

Can “shuffle” the dataset before splitting to improve performance



Scaling Method	Forest (No Shuffle)	Forest (Shuffle)
Raw Data	0.55 ± 0.53	0.45 ± 0.21
Standard	0.55 ± 0.53	0.44 ± 0.18
Robust	0.55 ± 0.53	0.45 ± 0.29
Normalizer	0.60 ± 0.54	0.76 ± 0.25
MinMax	0.55 ± 0.53	0.46 ± 0.19
Box-Cox	0.55 ± 0.54	0.45 ± 0.15
Yeo-Johnson	0.55 ± 0.54	0.44 ± 0.18
Normal	0.58 ± 0.56	0.46 ± 0.19
Uniform	0.55 ± 0.54	0.46 ± 0.19

RMSE Calculated Over a 10-Fold CV

	SVR	Linear	Ridge	Lasso	SGD	Forest
Raw Data	2.44 ± 0.90	2.46 ± 0.76	2.47 ± 0.72	2.97 ± 0.97	2.49 ± 0.62	0.45 ± 0.21
Standard	2.09 ± 1.02	2.38 ± 0.99	2.45 ± 0.80	3.02 ± 0.78	2.40 ± 0.92	0.44 ± 0.18
Robust	2.27 ± 1.04	2.47 ± 0.71	2.43 ± 0.85	3.03 ± 0.74	2.46 ± 0.77	0.45 ± 0.29
Normalizer	2.75 ± 0.81	2.37 ± 0.79	2.42 ± 0.62	2.99 ± 0.91	2.68 ± 0.74	0.76 ± 0.25
MinMax	2.56 ± 0.79	2.44 ± 0.81	2.37 ± 1.02	3.00 ± 0.88	2.42 ± 0.90	0.46 ± 0.19
Box-Cox	1.97 ± 0.77	2.37 ± 0.65	2.25 ± 1.02	2.99 ± 0.91	2.28 ± 0.96	0.45 ± 0.15
Yeo-Johnson	2.03 ± 1.12	2.43 ± 0.91	2.43 ± 0.93	2.98 ± 0.92	2.45 ± 0.86	0.44 ± 0.18
Normal	1.73 ± 0.73	2.12 ± 0.65	2.13 ± 0.62	3.03 ± 0.75	2.14 ± 0.51	0.46 ± 0.19
Uniform	2.44 ± 0.76	2.41 ± 0.94	2.47 ± 0.72	3.05 ± 0.65	2.45 ± 0.81	0.46 ± 0.19

Random Forest is the best performer regardless of scaling method. Therefore, more time was put towards optimizing Random Forest parameters

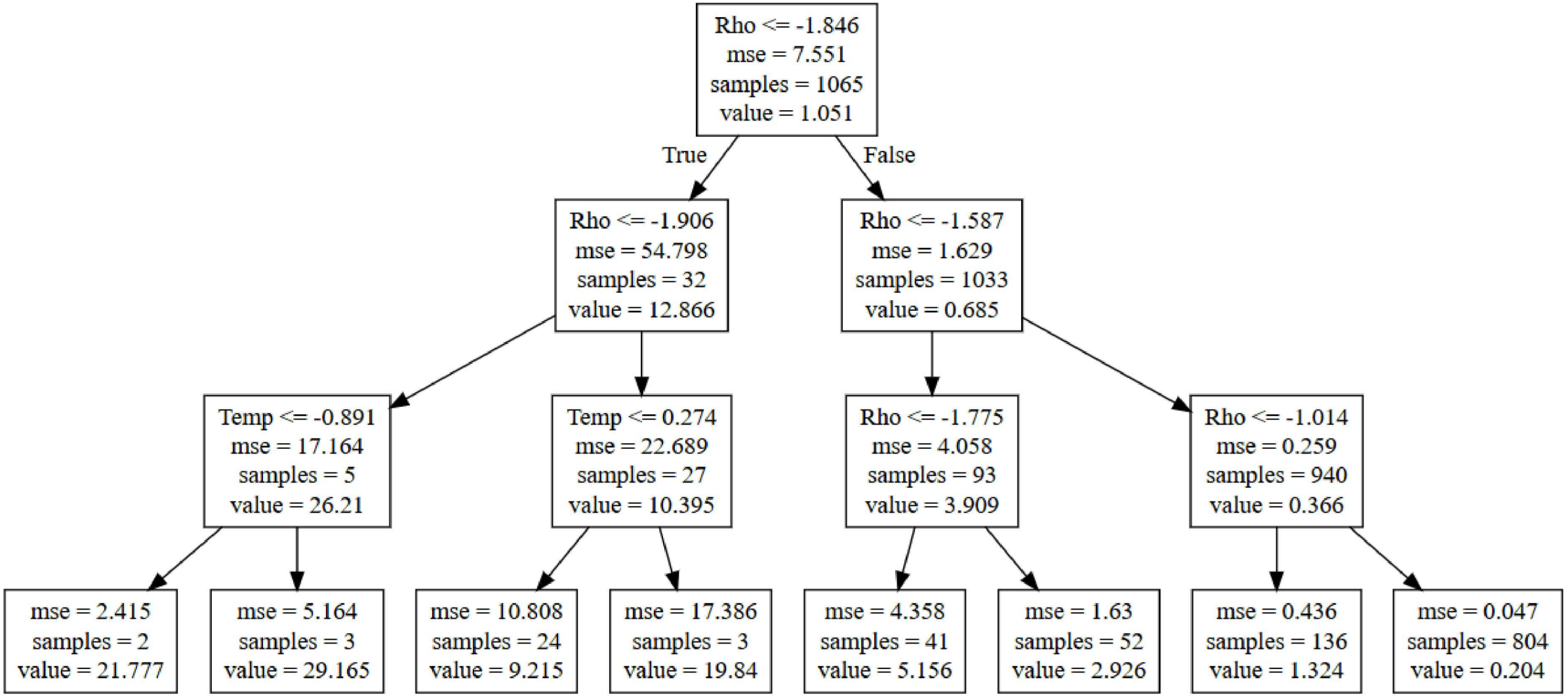
RMSE Calculated Over a 10-Fold CV

	SVR	Linear	Ridge	Lasso	SGD	Forest
Raw Data	2.44 ± 0.90	2.46 ± 0.76	2.47 ± 0.72	2.97 ± 0.97	2.49 ± 0.62	0.45 ± 0.21
Standard	2.09 ± 1.02	2.38 ± 0.99	2.45 ± 0.80	3.02 ± 0.78	2.40 ± 0.92	0.44 ± 0.18
Robust	2.27 ± 1.04	2.47 ± 0.71	2.43 ± 0.85	3.03 ± 0.74	2.46 ± 0.77	0.45 ± 0.29
Normalizer	2.75 ± 0.81	2.37 ± 0.79	2.42 ± 0.62	2.99 ± 0.91	2.68 ± 0.74	0.76 ± 0.25
MinMax	2.56 ± 0.79	2.44 ± 0.81	2.37 ± 1.02	3.00 ± 0.88	2.42 ± 0.90	0.46 ± 0.19
Box-Cox	1.97 ± 0.77	2.37 ± 0.65	2.25 ± 1.02	2.99 ± 0.91	2.28 ± 0.96	0.45 ± 0.15
Yeo-Johnson	2.03 ± 1.12	2.43 ± 0.91	2.43 ± 0.93	2.98 ± 0.92	2.45 ± 0.86	0.44 ± 0.18
Normal	1.73 ± 0.73	2.12 ± 0.65	2.13 ± 0.62	3.03 ± 0.75	2.14 ± 0.51	0.46 ± 0.19
Uniform	2.44 ± 0.76	2.41 ± 0.94	2.47 ± 0.72	3.05 ± 0.65	2.45 ± 0.81	0.46 ± 0.19

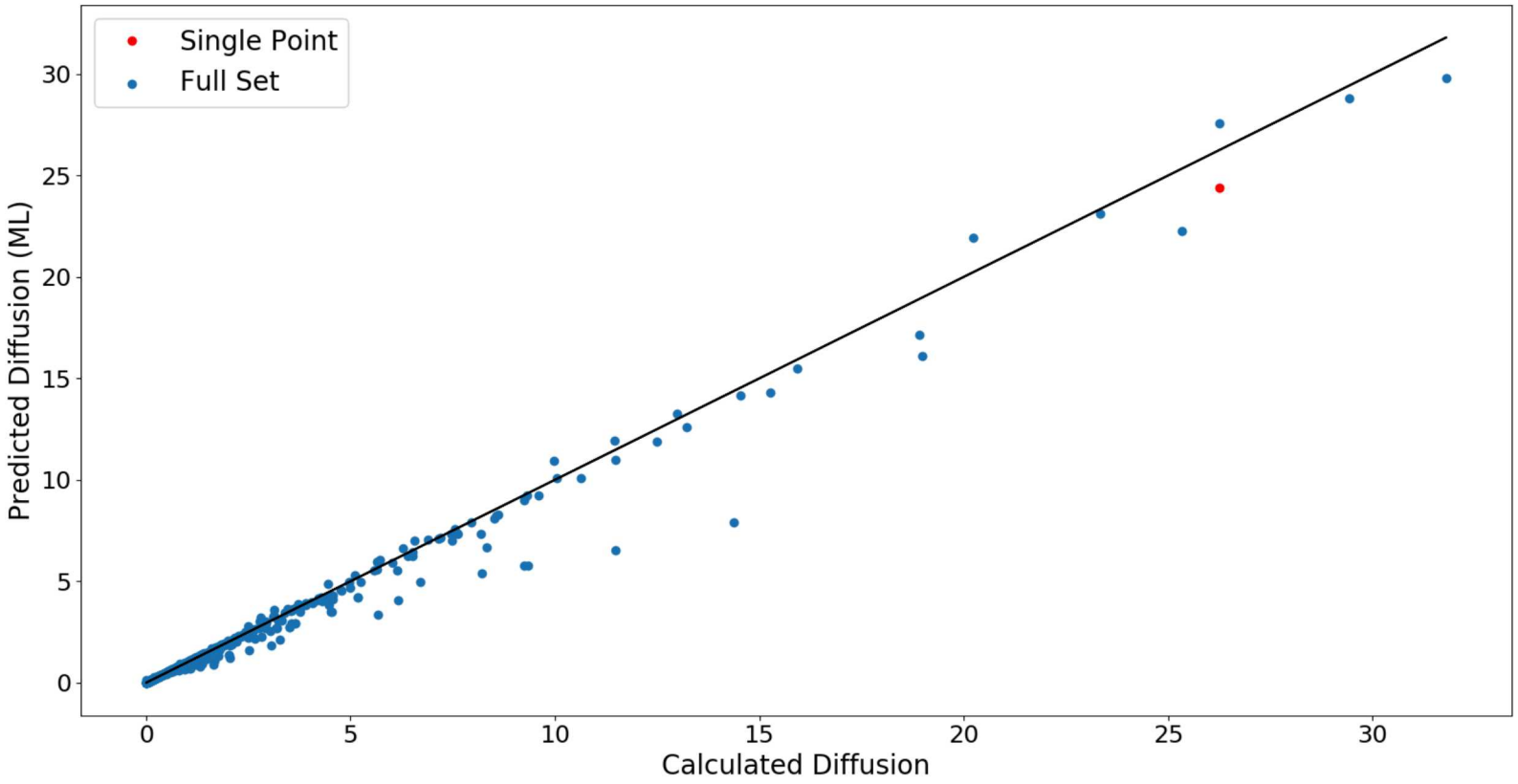
Chapman	Speedy	Speedy (ext)	Zhu
1.22 ± 0.60	1.17 ± 0.61	0.61 ± 0.18	0.26 ± 0.18

Zhu's model shows much better performance.

How Random Forest Works



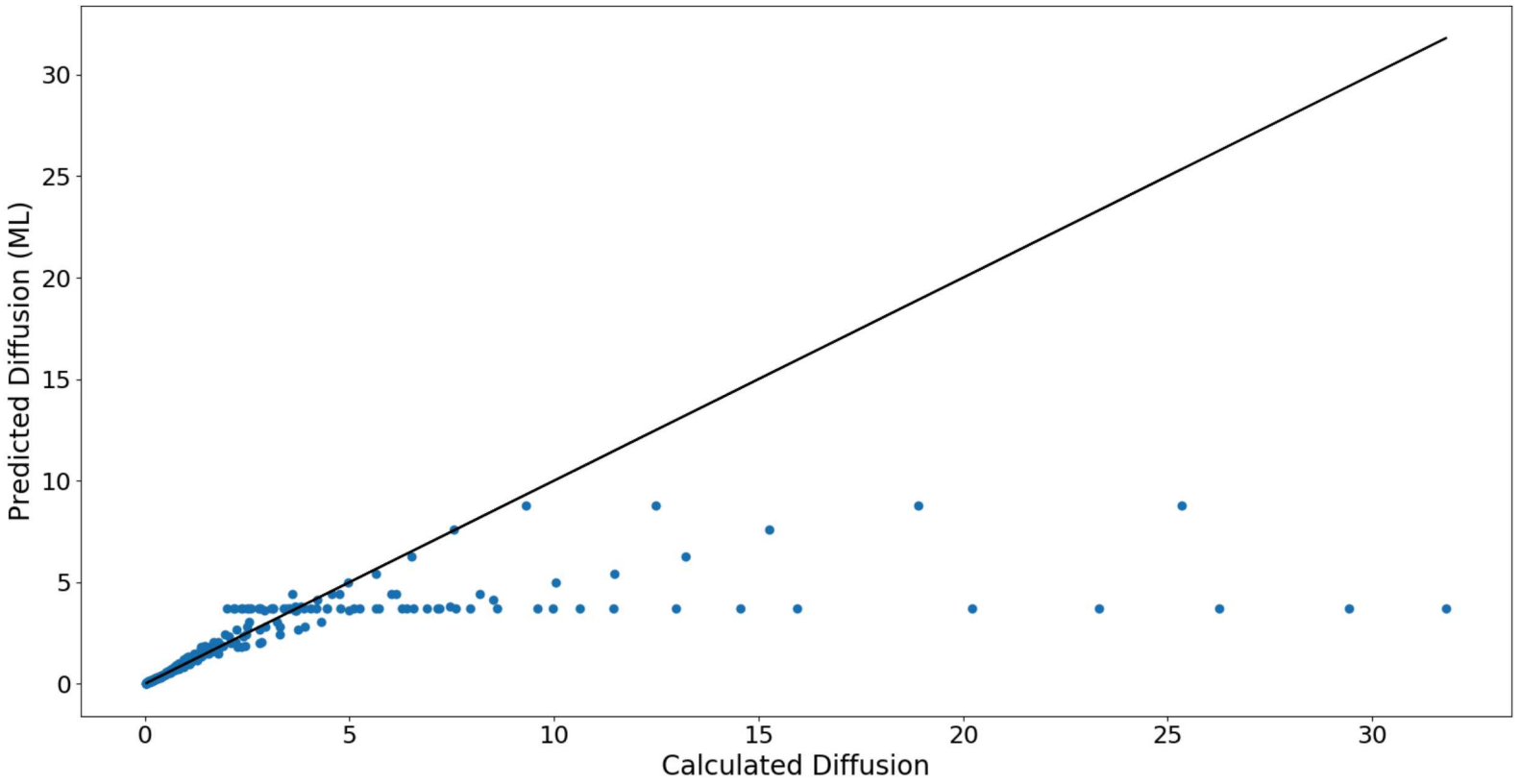
Interpolation With Random Forest



- Trained all but one data point
  D* = 26.26
  Rho* = 0.005
  T* = 0.900599
- Tested on full dataset and single point
  Prediction = 24.4
- **Data point closest to the test point**
  D* = 12.991
  Rho* = 0.01
  T* = 0.900501

Random Forest has the ability to interpolate!

Extrapolation With Random Forest



- Trained on Zhu's dataset
  - Largest D* is **9.2**
- Tested on Meier's data
  - Largest D* is **31.8**

Random Forest has no ability to extrapolate!

Feature Engineering - Algorithm

Environmental Science Nano

PAPER

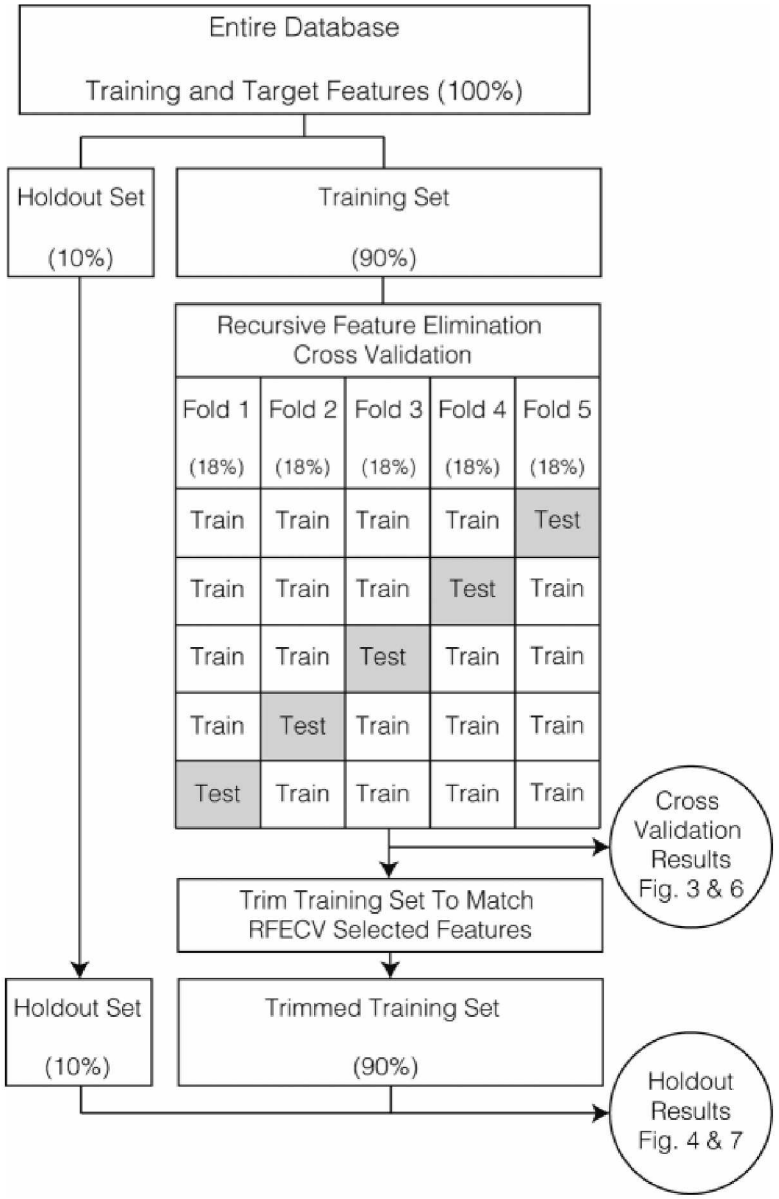
View Article Online
View Journal | View Issue

CrossMark click for updates

Cite this: *Environ. Sci.: Nano*, 2015, 2, 352

Prediction of nanoparticle transport behavior from physicochemical properties: machine learning provides insights to guide the next generation of transport models†

Eli Goldberg,[a] Martin Scheringer,*[ab] Thomas D. Bucheli[c] and Konrad Hungerbühler[a]



Feature Engineering - Feature Importance

0th iteration (107 features)

1. Feature[rho^3 / T^2.5] (0.202618)
2. Feature[T^2.5 / rho^3] (0.099727)
3. Feature[T^3.5 / rho^4] (0.096488)
4. Feature[rho^2 / T^1.5] (0.093950)
5. Feature[T^3.0 / rho^4] (0.052531)
6. Feature[T^1.5 / rho^2] (0.051877)
7. Feature[T^3.5 / rho^5] (0.049922)
8. Feature[rho^3 / T^2.0] (0.034661)
9. Feature[rho^1 / T^1.0] (0.033637)
10. Feature[rho^2 / T^2.0] (0.029279)

…

51. Feature[rho*] (0.001054)

…

84. Feature[T*] (0.000106)

…

107. Feature[rho^5 / T^1.0] (0.000001)

4th iteration (67 features)

1. Feature[T^2.5 / rho^3] (0.213646)
2. Feature[T^3.5 / rho^4] (0.123838)
3. Feature[T^2.5 / rho^4] (0.119322)
4. Feature[rho^3 / T^2.5] (0.068148)
5. Feature[T^3.0 / rho^4] (0.063451)
6. Feature[T^3.5 / rho^5] (0.056052)
7. Feature[T^2.0 / rho^3] (0.055414)
8. Feature[T^1.5 / rho^2] (0.055036)
9. Feature[rho^2 / T^1.5] (0.034017)
10. Feature[rho^3 / T^2.0] (0.028228)

…

49. Feature[rho*] (0.000476)

…

67. Feature[rho^3 * T^1.5] (0.000063)

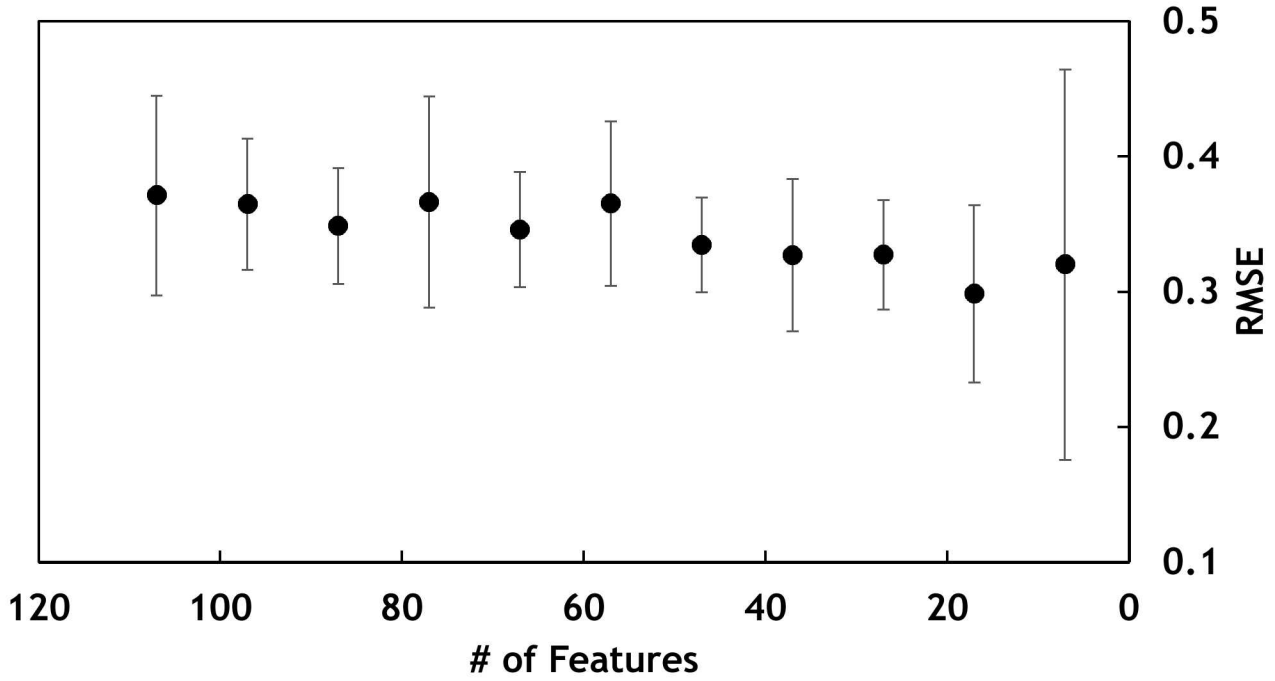
Feature Engineering - Feature Importance

Final Iteration (17 features)

1. Feature[T^2.5 / rho^3] (0.147572)
2. Feature[T^3.5 / rho^4] (0.115308)
3. Feature[T^2.5 / rho^4] (0.108927)
4. Feature[rho^3 / T^2.5] (0.093794)
5. Feature[T^3.5 / rho^5] (0.085812)
6. Feature[T^1.5 / rho^2] (0.075866)
7. Feature[T^3.0 / rho^4] (0.065934)
8. Feature[rho^2 / T^1.5] (0.062274)
9. Feature[T^2.0 / rho^3] (0.052163)
10. Feature[rho^3 / T^2.0] (0.049755)
11. Feature[T^3.0 / rho^3] (0.025409)
12. Feature[T^2.0 / rho^2] (0.021161)
13. Feature[rho^1 / T^1.0] (0.021112)
14. Feature[rho^2 / T^2.0] (0.020165)
15. Feature[T^1.0 / rho^1] (0.019573)
16. Feature[rho^4 / T^3.0] (0.018454)
17. Feature[rho^3 / T^3.0] (0.016722)

T* and rho* never make it to the final iterations

RMSE at Each Iteration



- Not a significant difference in the error
- Interested in how the model performs with less than 7 features

Random Forest RMSE Comparison

Features	T* & rho*	T* & rho*	$Rho^{2.0}/T^{1.5}$ & $T^{2.5}/rho^{3.0}$
Kfold Type	No Shuffle	Shuffle	Shuffle
Raw Data	0.55 ± 0.53	0.45 ± 0.21	0.31 ± 0.15
Standard	0.55 ± 0.53	0.44 ± 0.18	0.28 ± 0.16
Robust	0.55 ± 0.53	0.45 ± 0.29	0.30 ± 0.17
Normalizer	0.60 ± 0.54	0.76 ± 0.25	1.98 ± 0.68
MinMax	0.55 ± 0.53	0.46 ± 0.19	0.31 ± 0.14
Box-Cox	0.55 ± 0.54	0.45 ± 0.15	0.31 ± 0.14
Yeo-Johnson	0.55 ± 0.54	0.44 ± 0.18	0.30 ± 0.12
Normal	0.58 ± 0.56	0.46 ± 0.19	0.31 ± 0.14
Uniform	0.55 ± 0.54	0.46 ± 0.19	0.28 ± 0.14

Chapman	Speedy	Speedy (ext)	Zhu
1.22 ± 0.60	1.17 ± 0.61	0.61 ± 0.18	0.26 ± 0.18

Feature Engineering has generated promising results.

What's Next

- Is Random Forest the method we want to move forward with?
  - Extrapolation is important for what we are trying to do
  - Attempt to create a Neural Net
- What features to try next and how many?
  - Possibly add exponentials and logarithms
- Begin transitioning to pure solutions
  - Decide what features are important in real systems