Research Report: Lasso and RFR

Dr. Greg Hunt,* Nick Reeder,* Benjamin Kim**

*William & Mary, ** University of Virginia

0. Abstract

Accurate Uncertainty Quantification of critical components is essential to the success of any technology. Uncertainty Quantification is, however, an extraordinary data intensive process. Previous literature has indicated that 5000 data points or more may be required. Our model, heat shielding on hypersonic vehicles, is incredibly complex. Simulation and physical testing are both time and financially expensive. Because of this, surrogate models are create d to ease the data creation process. This project seeks to explore multiple methods of surrogate analysis, such as Gaussian Process Regression, Lasso Regression, and Random Forest Regression, to create accurate models aimed at revealing the confidence in our target system.

1. Model description

Three models were tested initially, Gaussian Process Regression (GPR), Random Forest Regression (RFR), and Lasso Regression. GPR was selected as there is existing literature and is able to accurately predict upon very complex curves and distributions, however it can struggle at high dimensions. RFR was chosen as it is able handle high dimensions very well and is able to handle complex distributions; however, RFR models are highly data dependent and are sometimes not able reach the highest levels of accuracy. Lasso was selected as it is able to control for the unimportant variables and parse out dimensions easily, although Lasso can struggle when there are many important dimensions. Data for model creation was generated from

* William & Mary
** University of Virginia

a uniform distribution, with each dimension having a variance of ± 10% of its baseline value. Data from other distributions were also tested, but this was the primary data set. The most important credential for surrogates is accuracy, especially when being used for uncertainty analysis as the volume of data generated will be so large.

2. Model Performance

2.1 Methodology and overall results

To test model adequacy, three measures of accuracy were used: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (r2). Other considerations were whether the models were able to filter out noise effectively. These metrics were observed over various data volumes, always on a randomly sampled 80/20 train-test split over 30 repetitions. Overall results can be seen in the below in table 1:

Model	**RMSE**	**MAE**	**r2**
RFR	6.56	4.99	0.86
Lasso	3.42	2.72	0.95

Table 1: Results of RFR and Lasso Regression

2.2 Hyper parameter tuning and discussion of RFR Performance

Random Forest Regression was initially promising, but as Lasso and GPR improved it was outclassed. Three hyperparameters were adjusted in order to improve accuracy and reduce any potential overfitting: maximum depth, number of trees, and minimum leaf size.

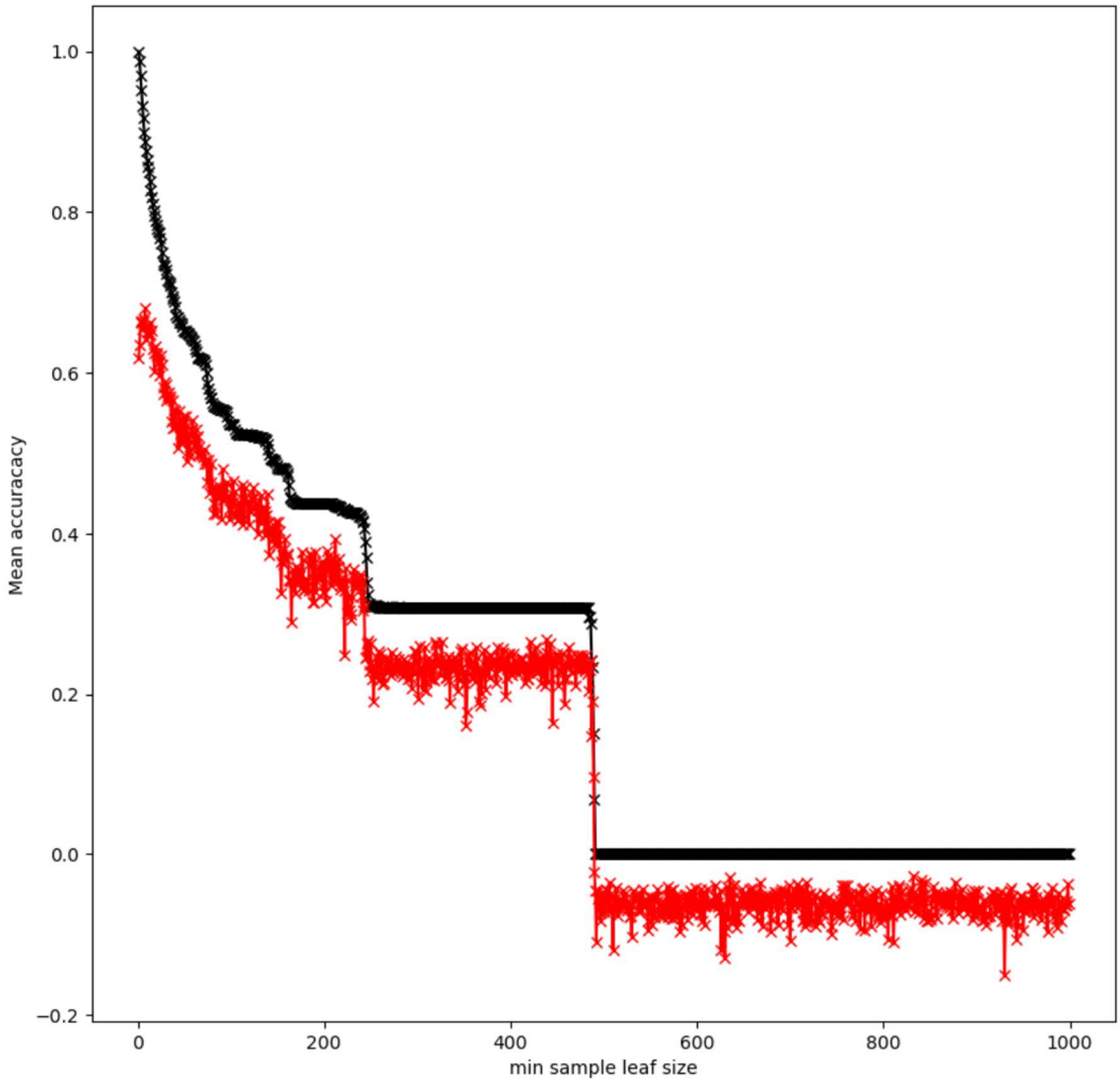

Figure 1: Train (black) and test (red) accuracies of a random forest as minimum leaf size increases.

The accuracy for both the train and test sets drop significantly, while the gulf between the two test does not appear to shrink [Figure 1]. This implies that limiting the leaf size does not provide much improvement to the model and should not be adjusted. We next adjusted the number of individual trees in the forest. This is often done not to improve accuracy, but efficiency. Figure 2 shows the increase in accuracy as the number of trees improves.

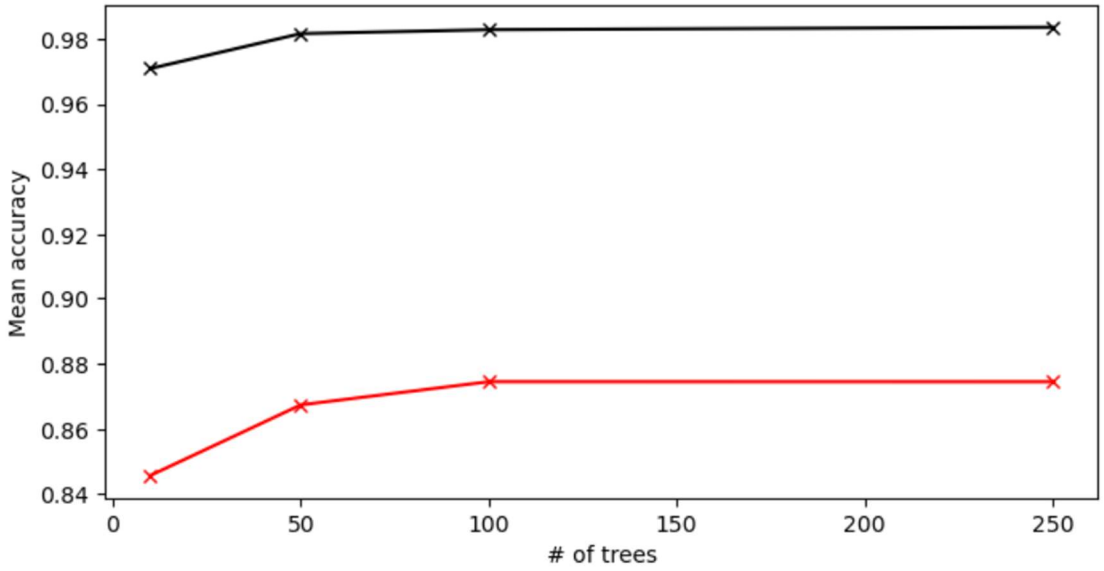

Figure 2: Increase in accuracy as the number of trees increases.

As accuracy does not continue in significantly increase as trees exceeds 100 and the difference of accuracies does not shrink, it appears that the optimal number of trees appears to be 100. After this a dual parameter optimization was conducted over both the number of trees and the optimal depth. Figure 3 shows a heat map of testing accuracy over both the number of trees and the maximum depth of the forest. As there are little overfitting concerns and limiting the depth of the trees does not appear to have a positive effect on accuracy, this hyper-parameter was not adjusted in the final model.

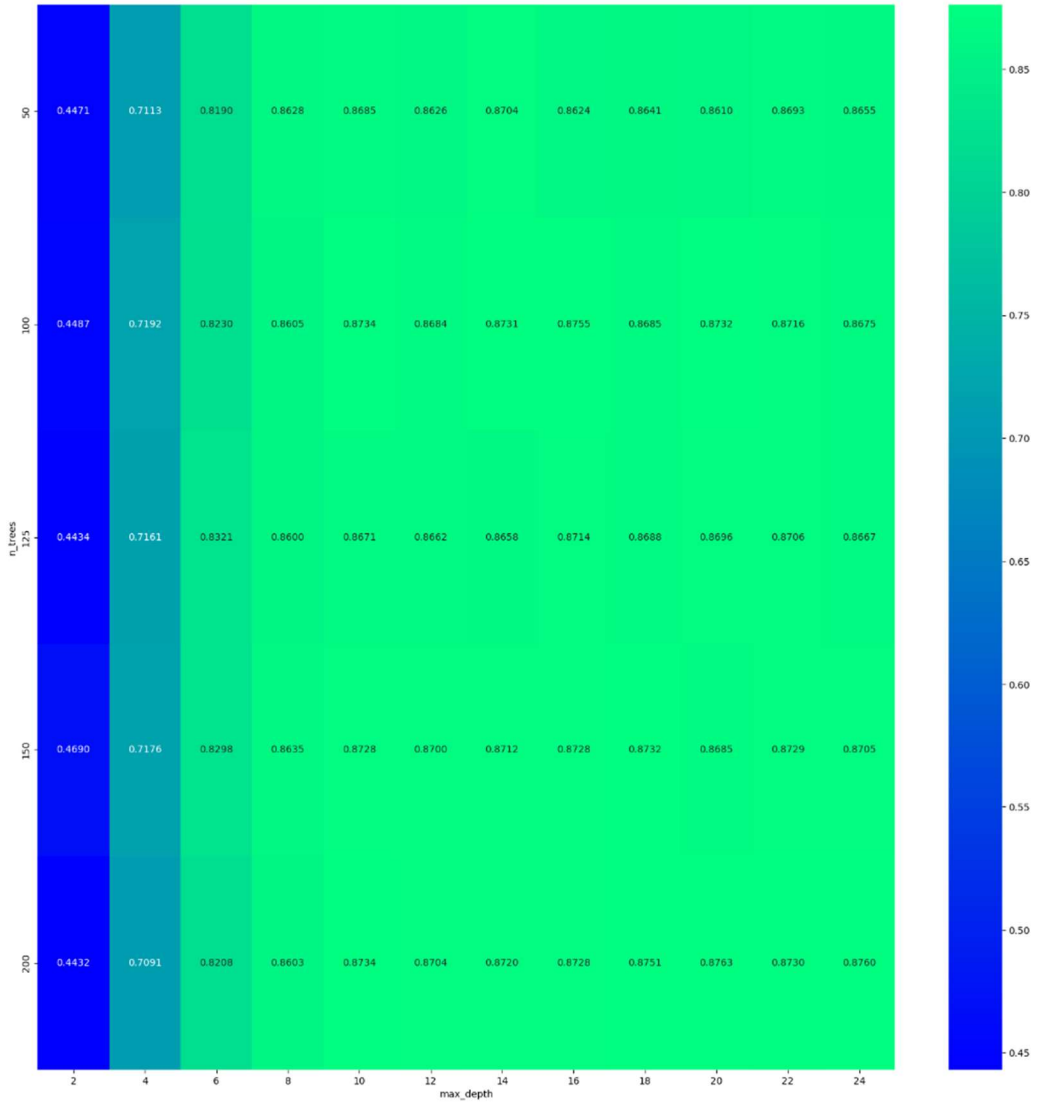

Figure 3: Heat map of Max Depth (x), Number of Trees (y), and testing accuracy (z)

Random Forest Regression has no built in variable selection. This is typically not a problem as the less important variables will often just not be used for very many questions; however, variable selection can be very useful in the context of Uncertainty Quantification as it will allow us to determine where the most (or least) uncertainty is coming from. This is essential in the contexts of development and engineering as it tells us what parts of the object will require the most failure tolerance. Despite the very low error tolerance in the manufacturing process, the

heat shield will warp and compress under the extreme conditions of hypersonic flight, this means that we will grow far more uncertain the longer the flight happens.

After all of the hyper parameter tuning, we determined that RFR simply was not able to achieve the accuracy required for surrogate analysis.

2.3 Discussion of Lasso Performance

After z standardization ($z = [x - u] / s$) of the x variables to control for differing units, we began hyper-parameter tuning for the Lasso model. The only hyper parameter to tune is alpha, which controls the strength of the penalty term. We began with a broad search [Figure 4] and then shrank the search radius for greater precision [Figure 5][1]. We also found that the model required much less data to reach approximately its best accuracy. The Lasso approach yielded much greater accuracy and was selected for further UQ analysis over the RFR.

[1] Readers should note the difference in scale between figures 4 and 5

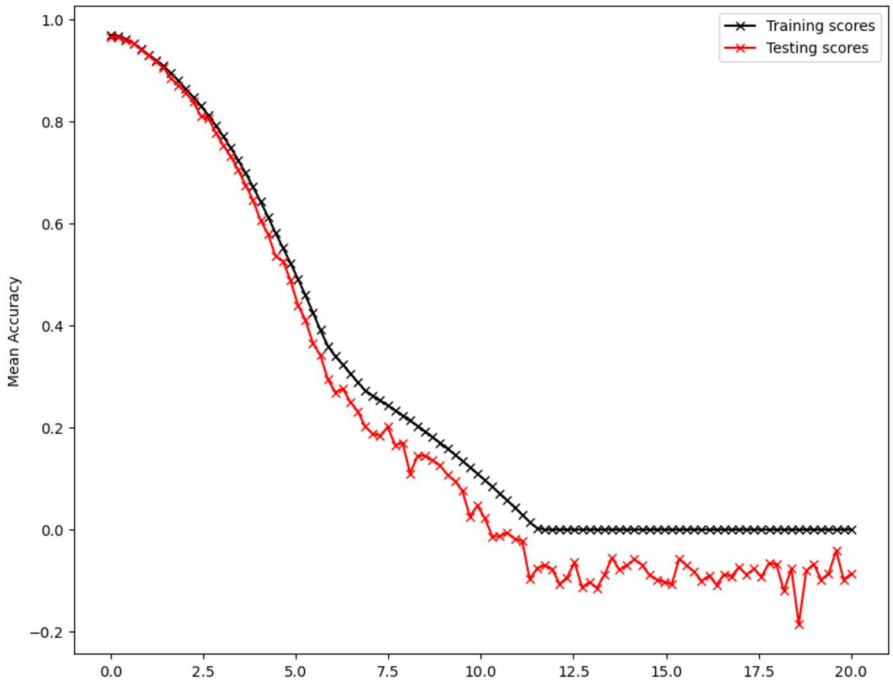

Figure 4: Broad search for optimal Alpha

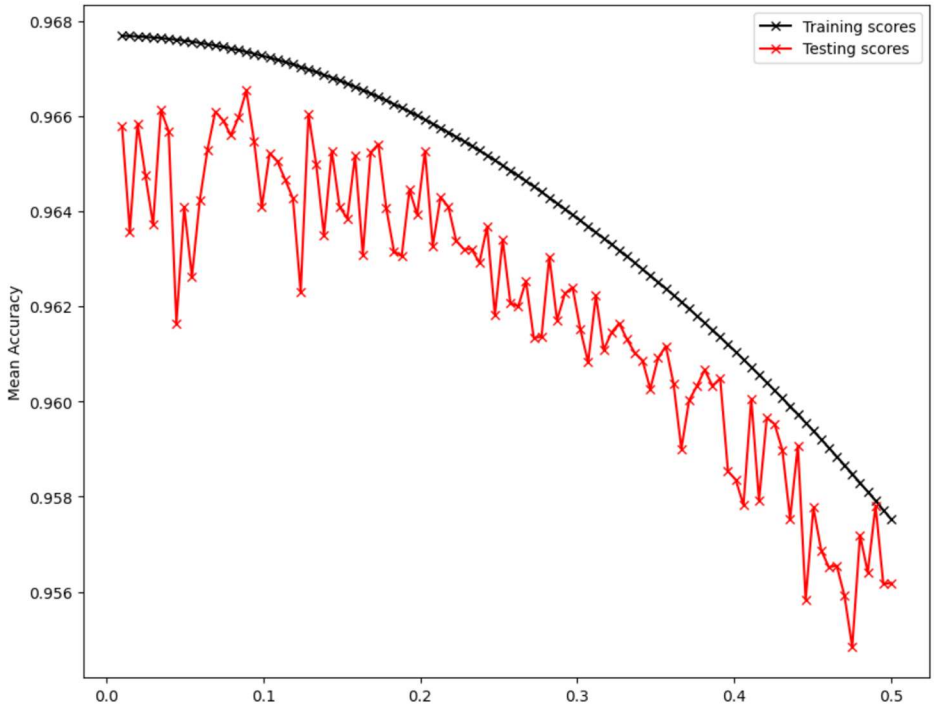

Figure 5: Narrow search for optimal Alpha

Next, we looked at how much data was required for the model to approach its maximum accuracy. We determined that the model only required about 450-500 data points, a massive improvement compared to RFR.

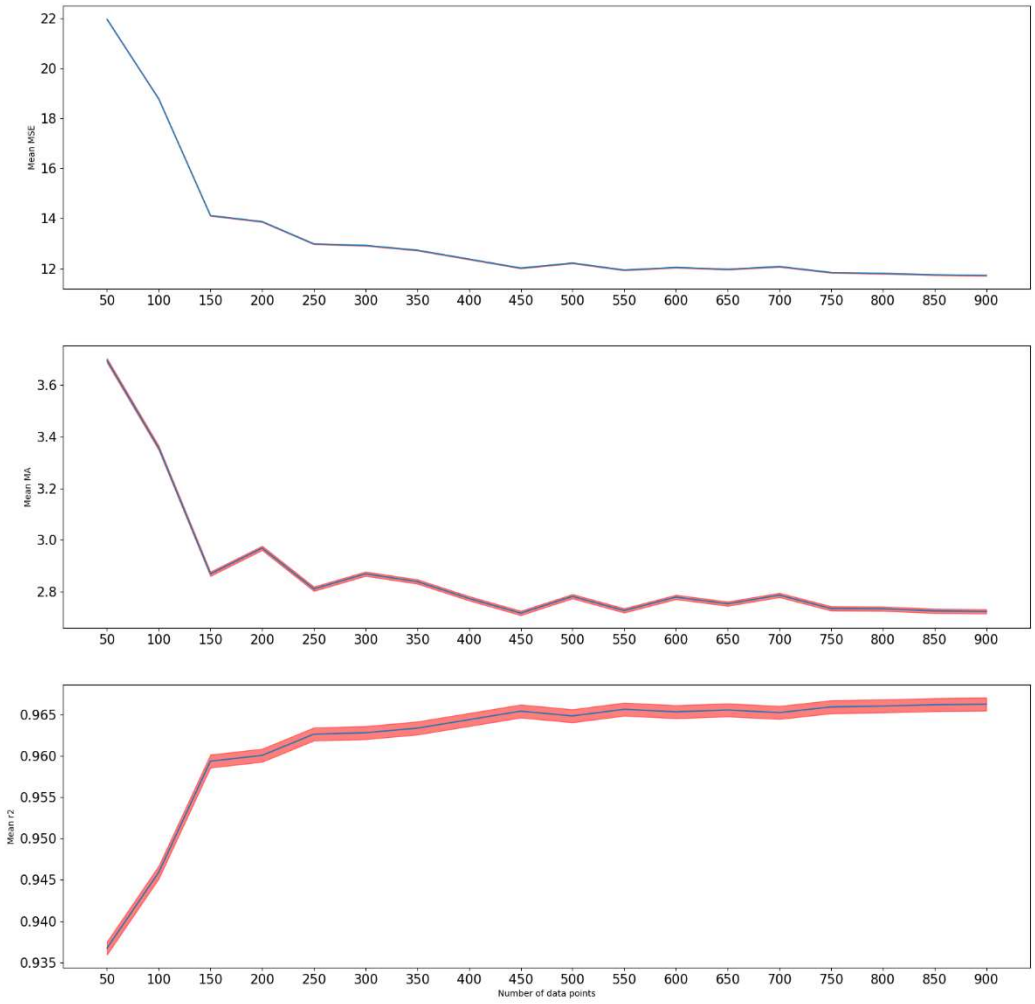

Figure 6: Model improvement as data volume increases

One of the advantages of the Lasso is that variable importance is very easy to determine. Because it is an expanded least squares regression, the importance of a variable can be derived from its coefficient. This coefficient can be driven to 0 by the penalty factor from lasso. To determine which variables were important and how fast we could learn that the coefficients were plotted as the volume of data increased [Figure 7]. We also plotted the average of the coefficients, weighted by the volume of data, as it is reasonable to assume that the coefficient will grow more accurate the more data is used [Figure 8].

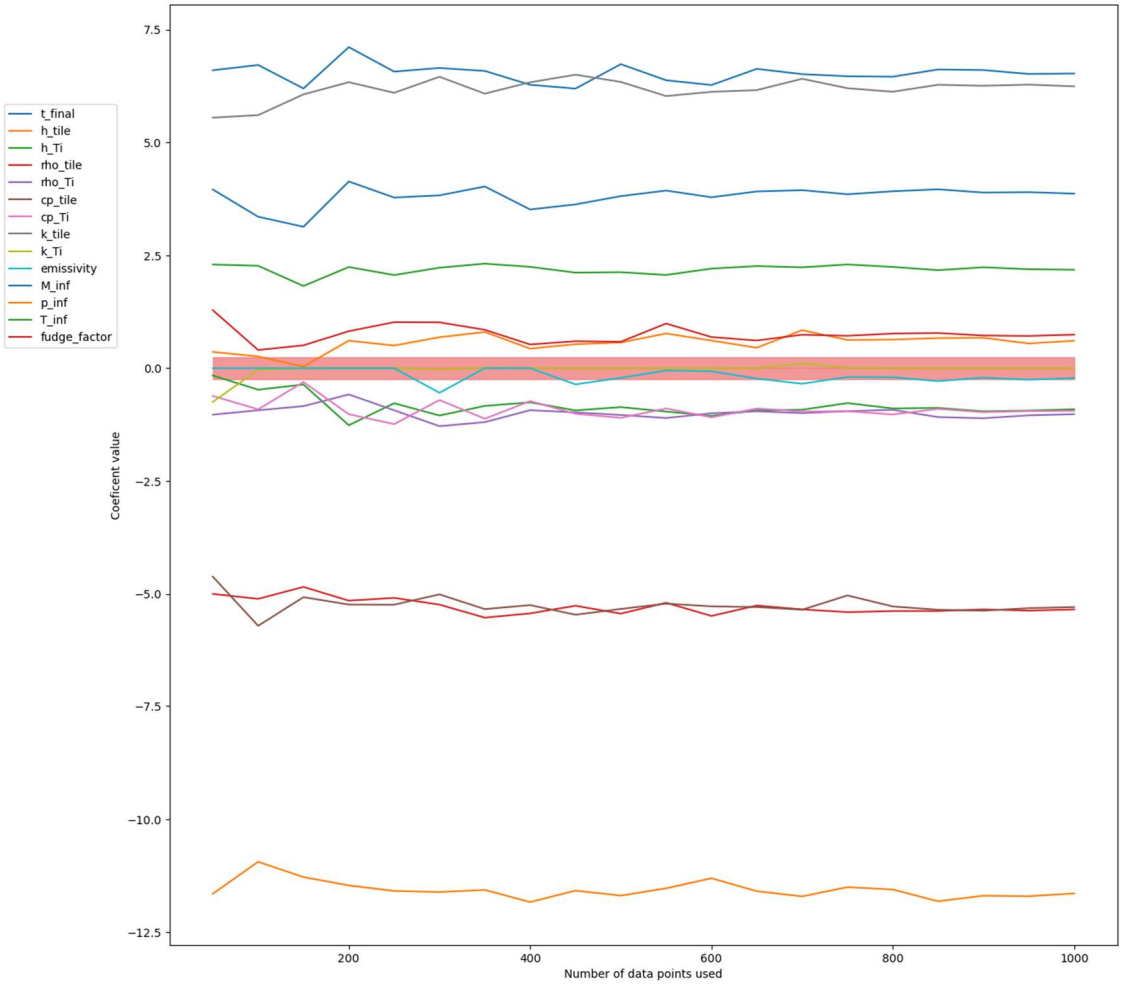

Figure 7: Value of each dimension's coefficient and data volume increases. Red zone is -0.5-0.5.

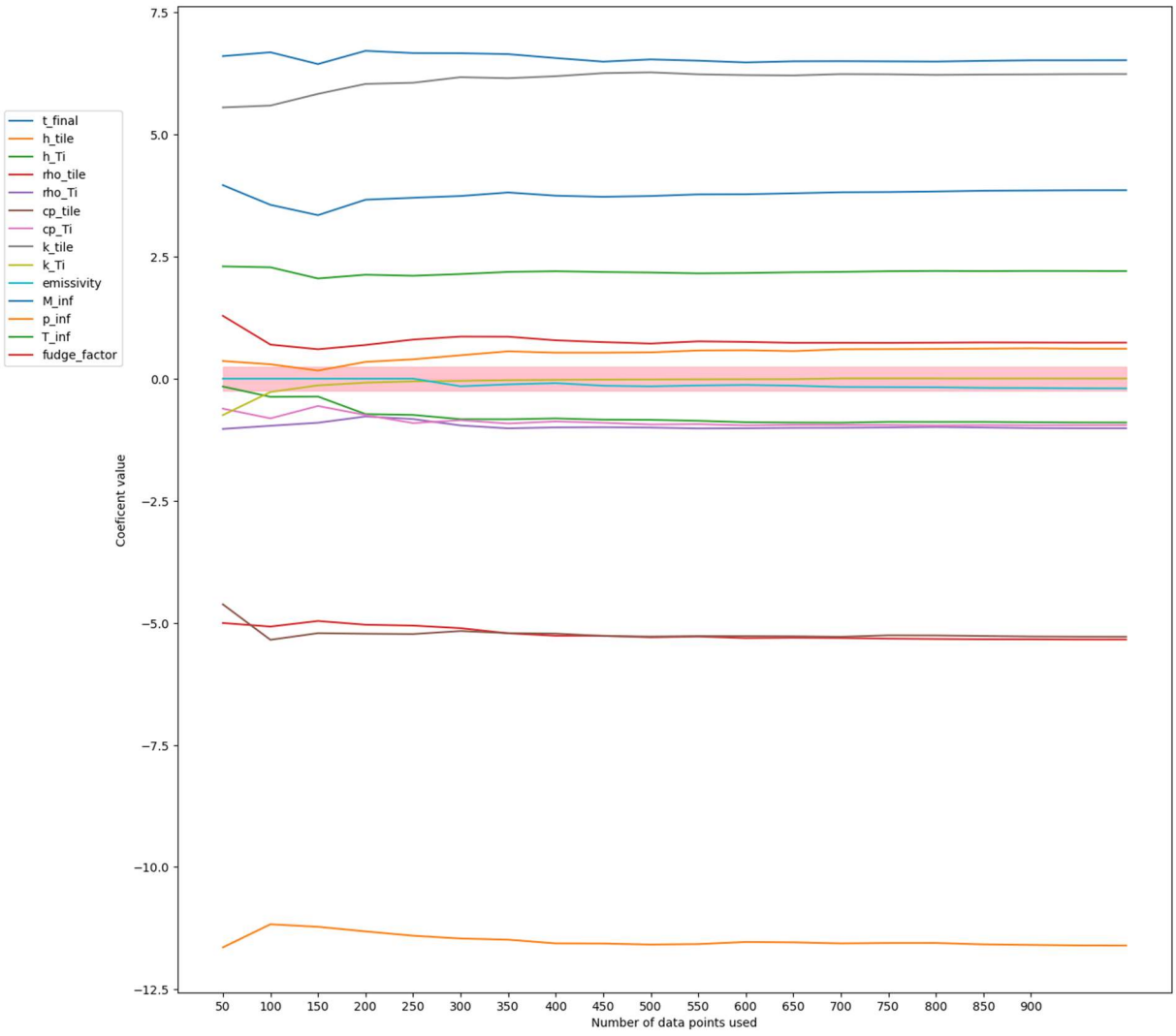

Figure 8: Weighted average of each dimension's coefficient and data volume increases. Red zone is -0.5-0.5.

Lasso is able to quickly determine which variables are important or unimportant, especially when a weighted average is use. Future works could develop a sequentially designed surrogate and simulator using the variables that are deemed important.

3. Uncertainty Quantification and Analysis

3.1 Uncertainty Quantification Over One Dimension

To perform the UQ, the data was split into a training and validation sets. These set were individually standardized. The Lasso model was trained and then fed the x values of the validation set, predicting the final temperatures. Initially, only the exposure time was varied as it was one of the most important variables. To judge accuracy we used two metrics: the Wasserstein Distance and the Kolmogrov-Shmirnov test. The Wasserstein distance is not an absolute measure, but can be very useful for internal comparison.

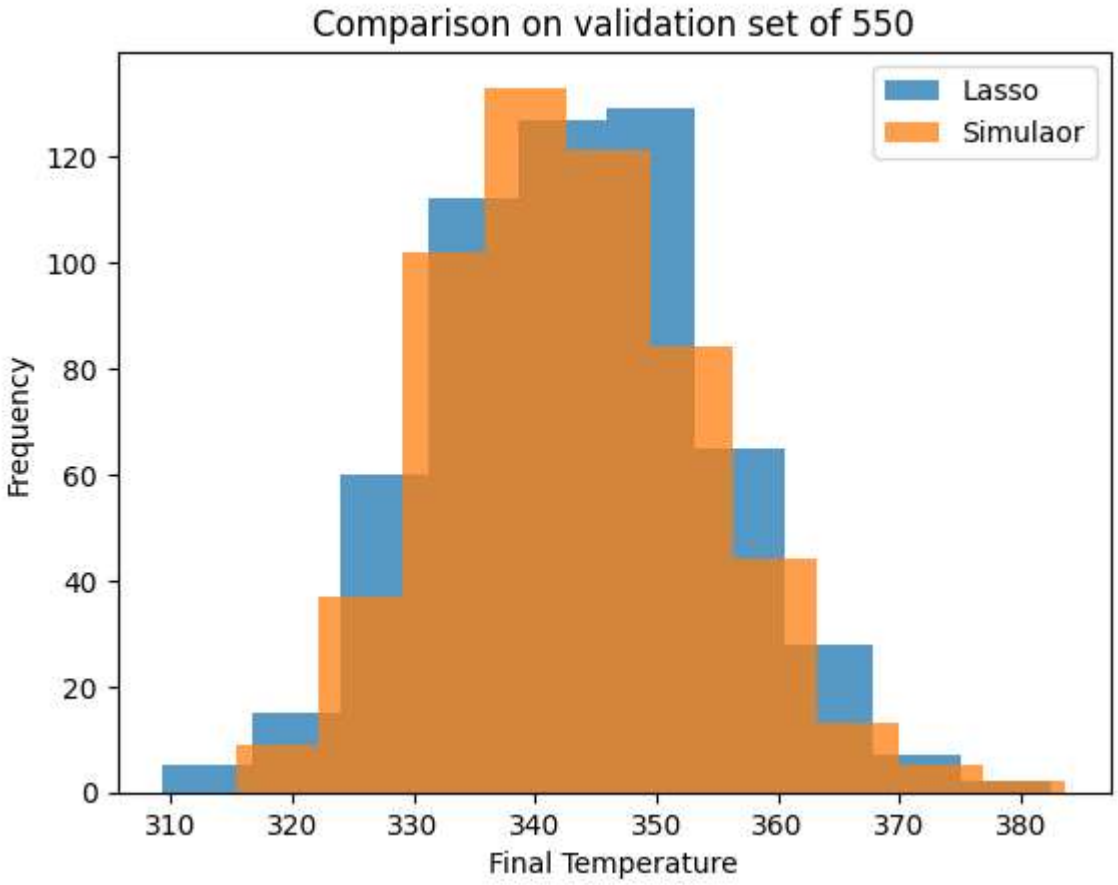

Figure 9: Distributions of final temperature of Lasso and Simulated data

These distributions produced a Wasserstein distance 0.671 and a KS Statistics of 0.045. This statistic gives a p-value of 0.621, failing to reject the null and concluding that the distributions are statistically the same.

3.2 Uncertainty Quantification for all variables

Two approaches were taken for this portion. The first, the data was randomly split 450/550 as had been done previously. This was done over 30 repetitions, producing a mean Wasserstein distance of 2.46. The second, all 1000 data points were used to train the model, then 5000 new data points without a final temperature were generated. These data points were then fed into the surrogate to get the predicted temperature. This was done to test that the surrogate would be able to create the correct distributions and uncertainty on brand new data, like it would be used for real applications. This second approach produced a Wasserstein distance of 2.33.

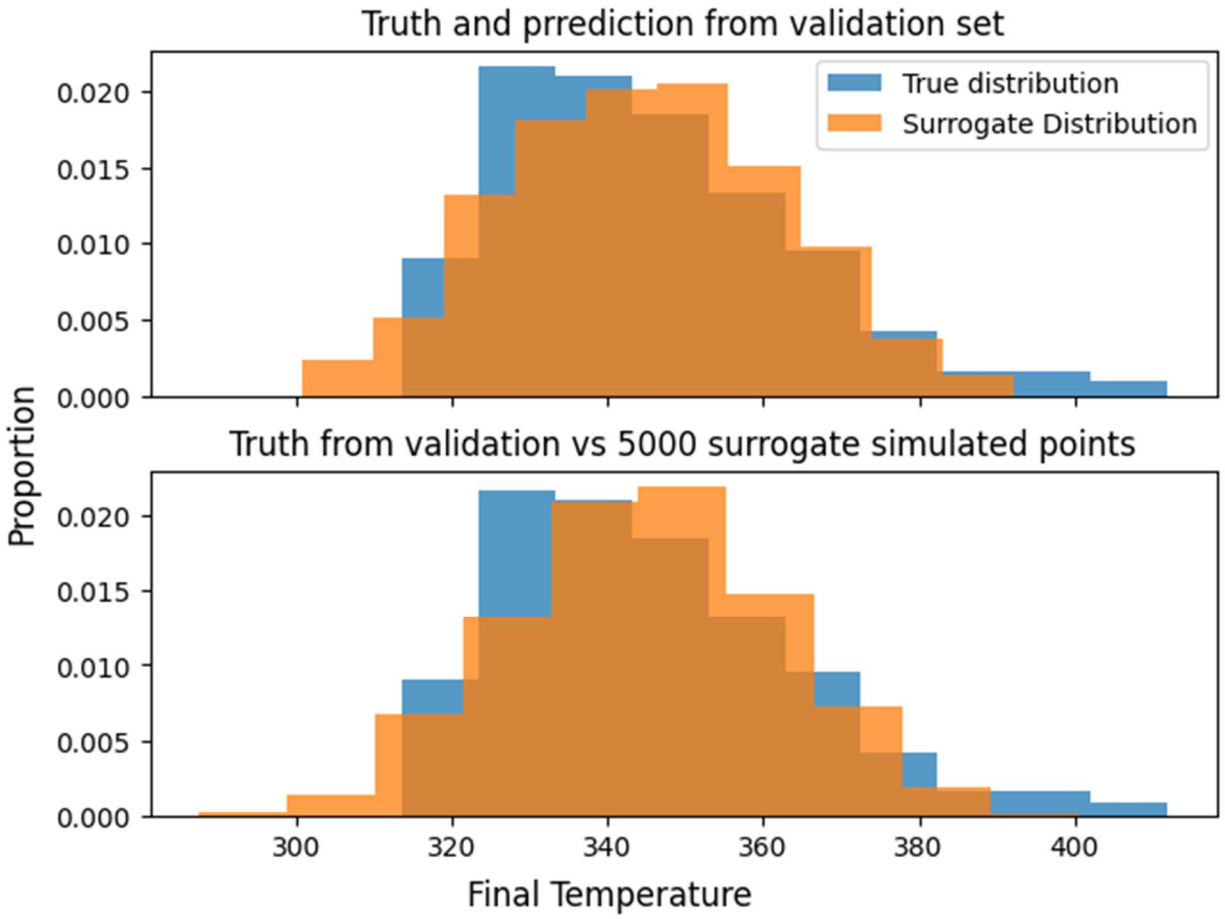

Figure 10: Comparison of surrogate and true distributions.

To determine how accurate approach two can be, this test was performed from 550 to 50,000, stepping up by 100. This was repeated 30 times at each step. The Wasserstein distance and time test time were recorded at each repetition. The time was not plotted as each repetition was trivially fast, the maximum time being 7500 microseconds (0.075 seconds).

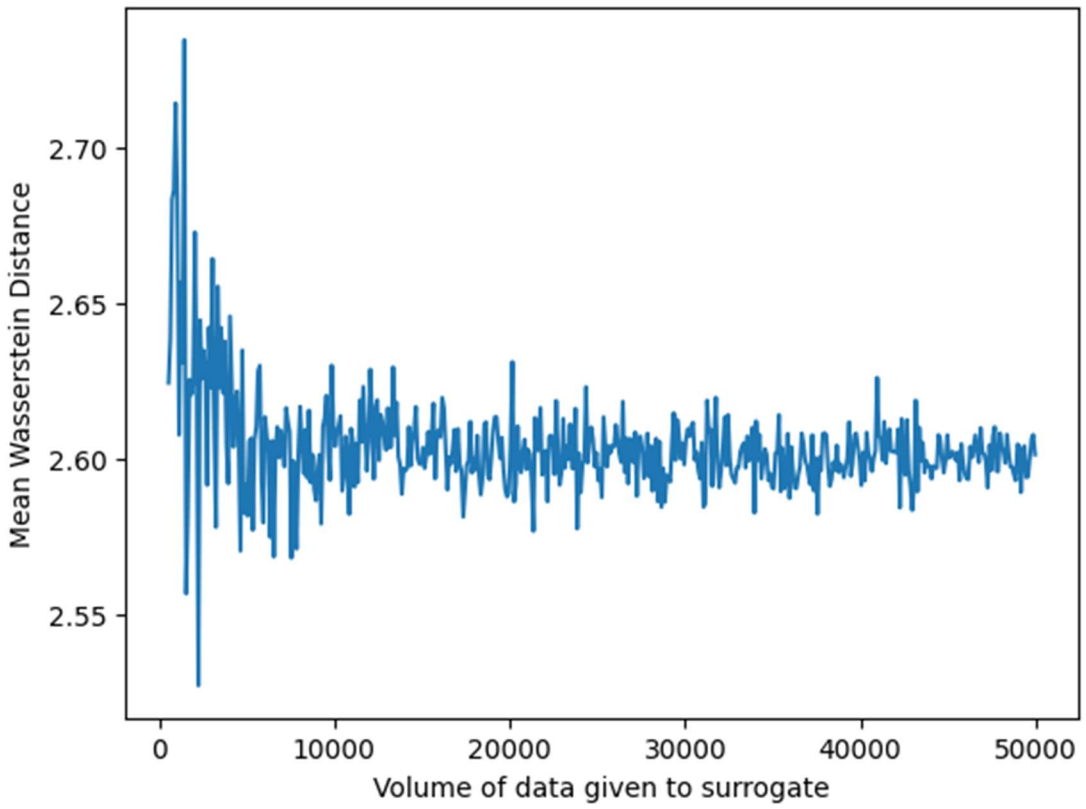

Figure 11: Average Wasserstein Distance from true distribution as data volume increases

3.3 Determining Variable Importance for Uncertainty

We then sought to determine which variables were the most important for creating uncertainty in the final temperature. To do this, all variables but one were kept constant. A Lasso model was then trained on that data and used to predict on a validation set. The standard

deviation of true final temperature, the predicted final temperature, and the standard deviation of that column [Figures 12, 13, 14].

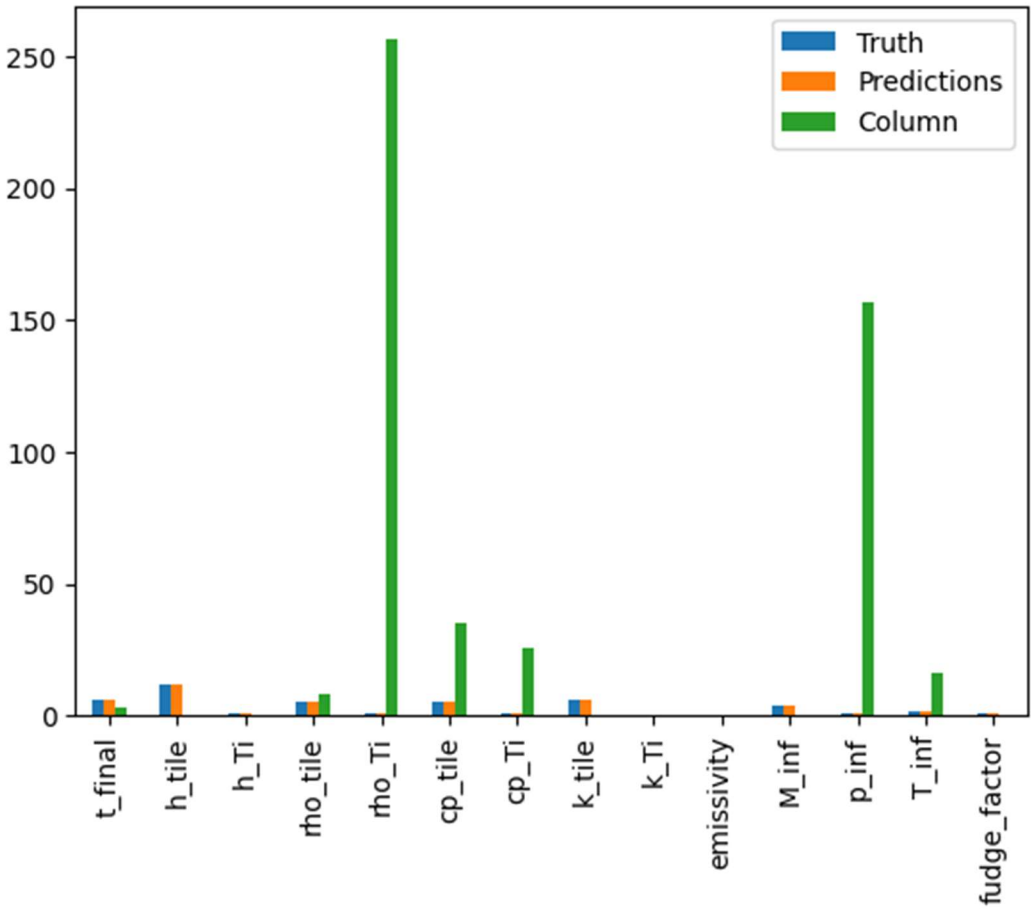

Figure 12: Standard Deviation of each dimension (column) compared to the standard deviation of final temperature.

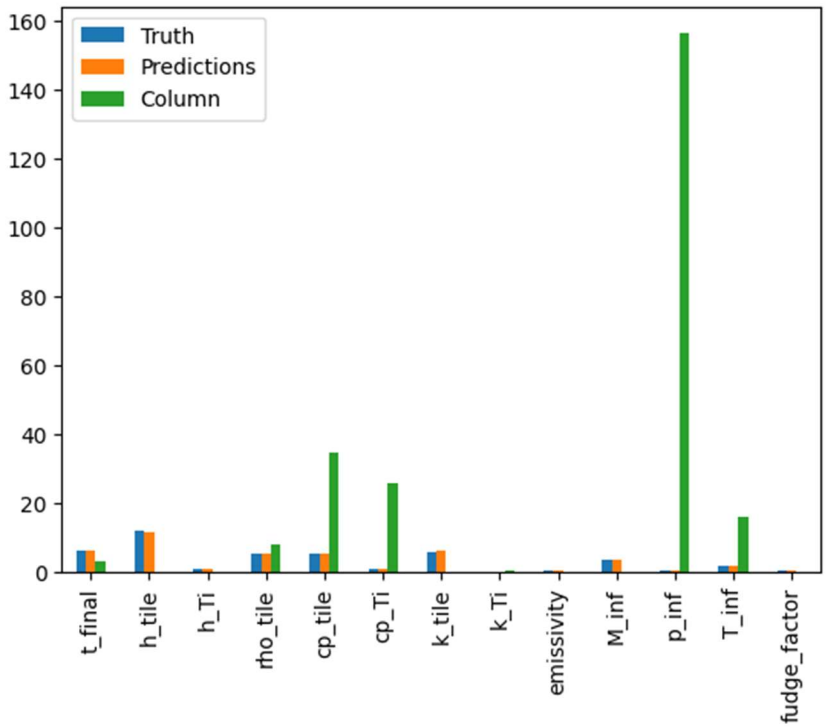

Figure 13: Standard Deviation of each dimension (column) compared to the standard deviation of final temperature with Rho Tile removed.

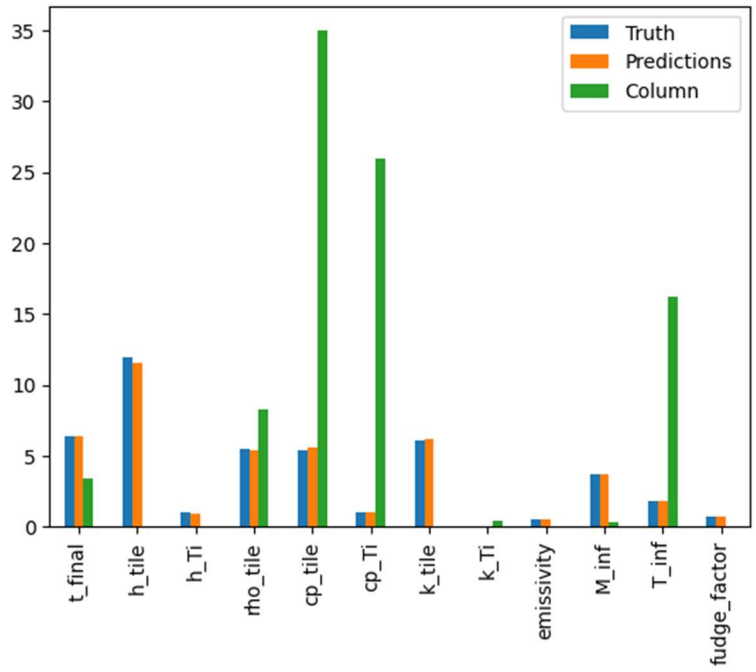

Figure 14: Standard Deviation of each dimension (column) compared to the standard deviation of final temperature with Rho Tile and P Inf removed.

3.4 Analyzing Effect of Increasing Uncertainty

We also sought to determine the accuracy of the model as uncertainty increased. It is not currently known how uncertain we are in any of the dimensions, and it is both possible and reasonable that the uncertainty will increase as flight time increases. Because of this we mapped the standard deviation of the true final temperatures and the predicted final temperatures as the proportional variance increased from 0% to 40% [Figure 15].

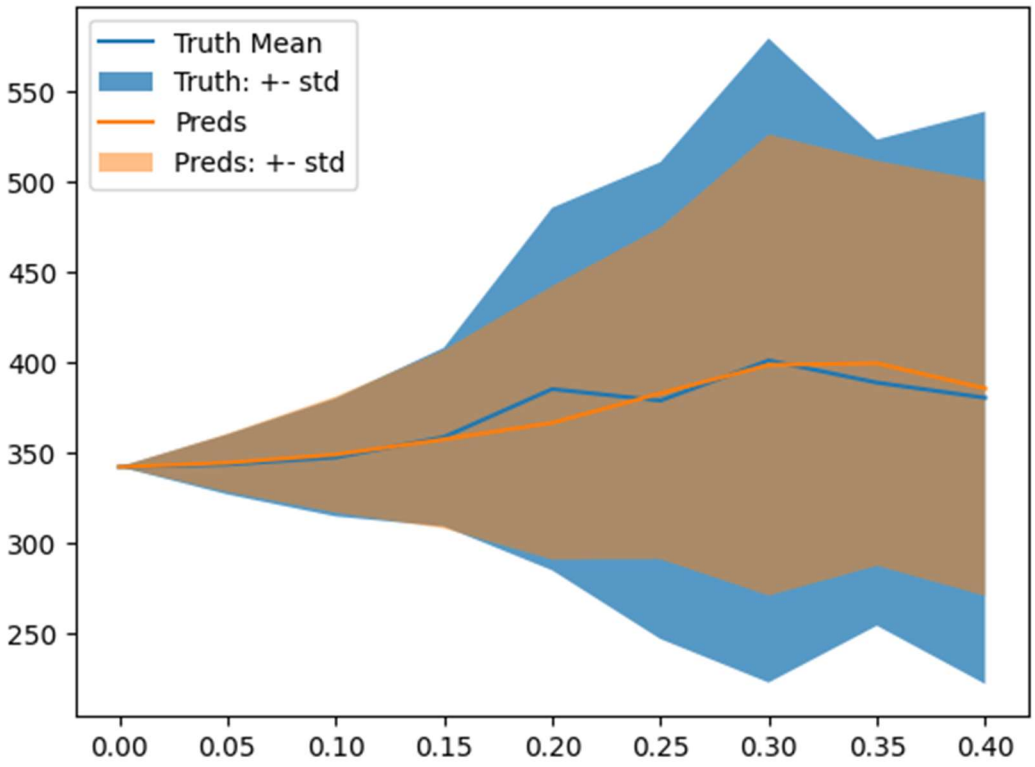

Figure 15: Fields of increasing standard deviations of both the true and predicted temperatures and error increases in all dimensions.

Due to some limitations in the simulator, as the variance increased there were less data points available.

3.5 Extrapolation

To test if the model, in this context, is able to extrapolate beyond the current level of error we generated a training set of 1000 data points at 10% error and then tested it on 500 data points at 20% error. Both sets of data were generated from a uniform distribution. After plotting the two distributions it was clear that the model was unable to capture the wider distribution, especially as temperatures increased. There is a limit in the simulator in that the minimum temperature will always be 300, so it is likely that in reality the Lasso predictions will miss the cooler predictions as well. It is also worth noting that extreme temperatures will only occur under a narrow set of conditions which are less likely to occur so there will be less training and testing data and those bounds.

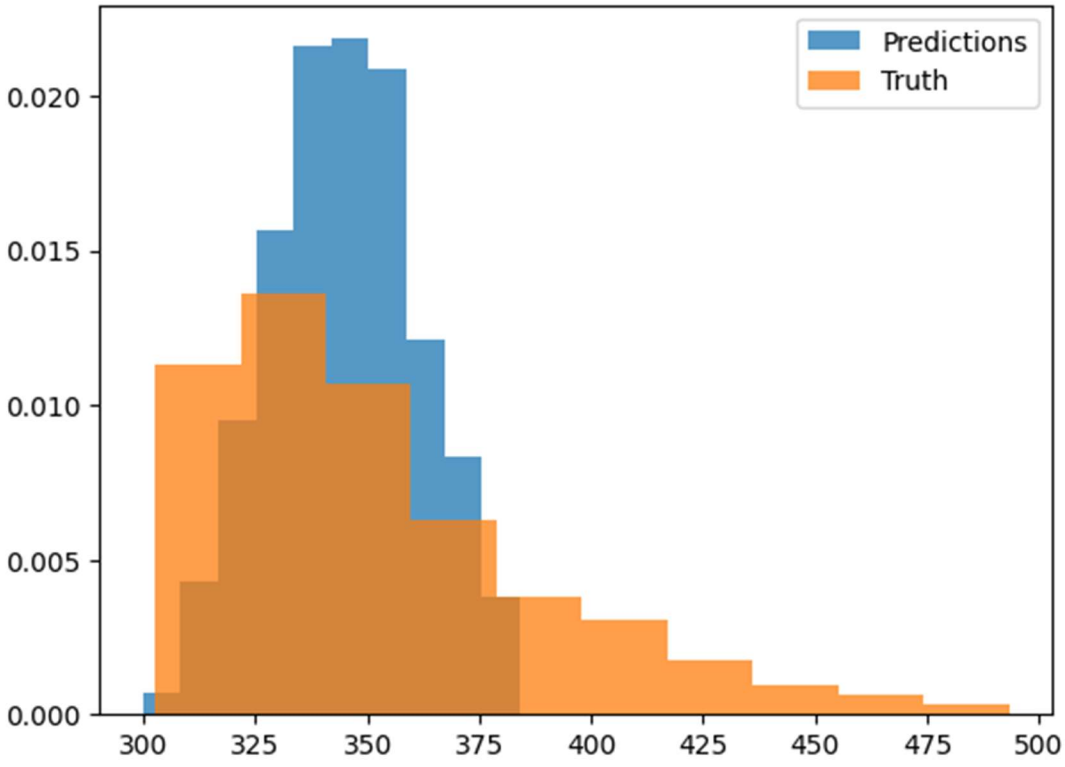

Figure 16: Predicted and true distributions of 20% error when model is only trained on 10% error.

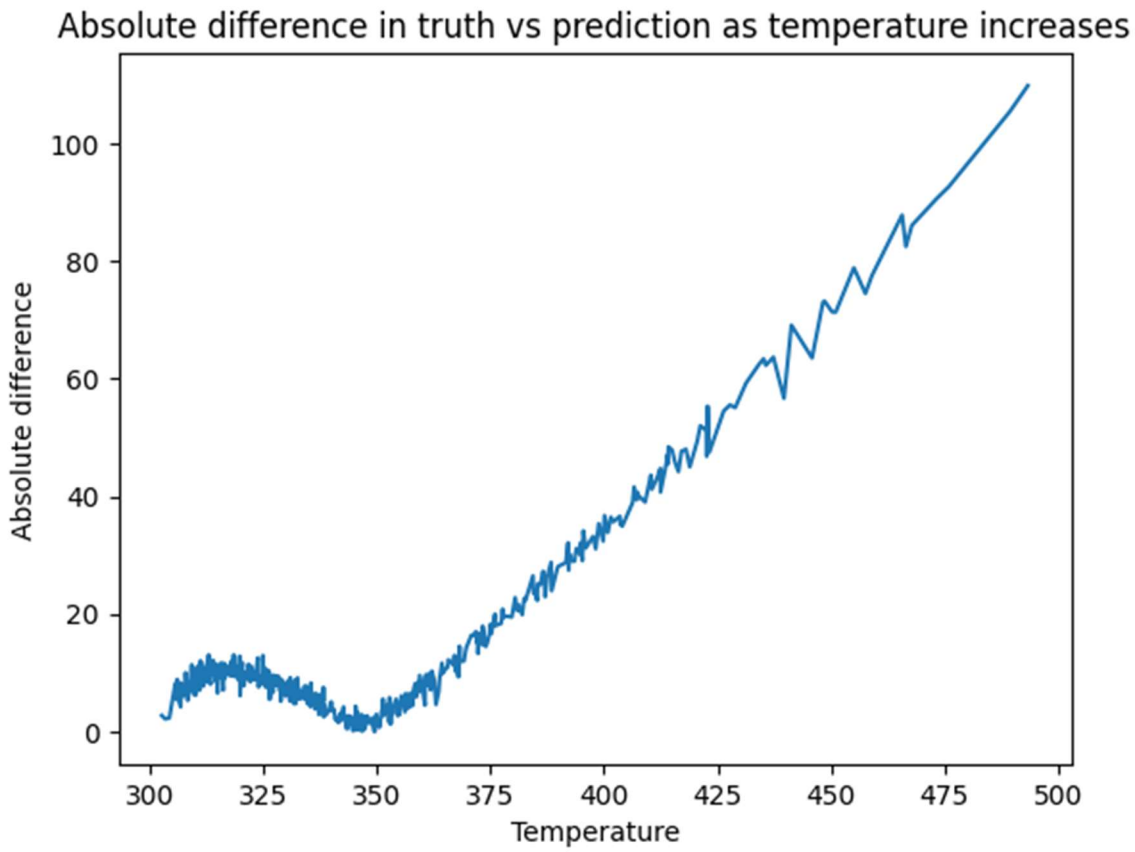

Figure 17: Absolute difference in the true and predicted final temperature when error is 20%.

4. Conclusions

It is clear that Lasso is a very effective approach for surrogate modeling. The accuracy it is able to achieve and the speed of the prediction can effectively fill in data points and provide the large swaths of data required for uncertainty quantification. However, there is a major weakness with the Lasso approach in that it is entirely unable to extrapolate. This requires it to be trained on the entire solution space before it can be used.