



# **Surrogate Modeling Tools Benchmark Test Report**

**Connor Chai**

**Product Design Optimization Laboratory**

**Simon Fraser University**

**July 04, 2025**



## Table of Contents

Overview .....	3
Background .....	4
Surrogate Modeling Methods.....	4
Test Functions .....	6
Methodology.....	7
Results .....	9
Discussion.....	12
Conclusion .....	16
References .....	17
Appendix A. ....	18
Appendix B. ....	25
Appendix C .....	27



## Overview

The purpose of this report is to compare a variety of surrogate modeling methods in Dakota and Python software (SMT, GPy, and scikit-learn libraries) with OASIS.AI using accuracy metrics and visual comparisons. Thirteen benchmark test functions were chosen, which vary from 10 to 30 dimensions.

Surrogate modeling, specifically in an engineering context, is the term used to describe the process of constructing surrogate models, which are approximations of functions. Surrogate models may be built on functions that are computationally expensive to evaluate, for example, Finite Element Analysis (FEA) or Computational Fluid Dynamics (CFD) analysis. Performing functions such as uncertainty quantification, design space exploration, and optimization on these computationally expensive problems can be severely time-consuming. By creating a simpler “surrogate” model that is computationally cheaper while maintaining the vast majority of details from the original model, the desired processes can be carried out more efficiently.

This benchmark test is motivated by the desire to determine if there are any suitable open-source surrogate modeling alternatives to the commercial tool, OASIS.AI. OASIS.AI requires a commercial license, and finding free alternatives could provide benefits for teaching and research purposes. In this test, the performances of the proposed surrogate modeling alternatives from reputable toolkits are evaluated against the OASIS.AI benchmark values using three accuracy metrics:  $R^2$ , RMAE, and RMAE.

A total of ten surrogate models built in Python and seven methods built in Dakota were tested. The tests showed that a few methods came close to the performance of OASIS.AI on some of the given problems. When taking training time into consideration, the top contender was GPx from the SMT Python library. Additionally, the top three methods in both Dakota and Python were compared against one another, and overall, Python software showed superior performance to Dakota.

## Background

Not only do the individual implementations of surrogate modeling methods exist in abundance, but the world of research features a wide selection of differing methods. Probabilistic models like Gaussian Processes are the most common [1], which use a stochastic process to provide models with estimated mean values along with corresponding levels of uncertainty. Other models may be deterministic, meaning that their model predicts points without levels of uncertainty, such as linear regression or Radial Basis Function models.

It is important to note that different implementations of the same model can result in significantly varying values. This is largely due to internal differences in the way a package addresses parameter tuning and implements mathematical formulas. To avoid the messiness of hyperparameter optimization, the implementations of models applied in this test largely used their default hyperparameter settings that come with the code. It is possible that optimizing the hyperparameters of some methods may lead to better performance with certain functions. However, this is considered out of the scope of the study.

## Surrogate Modeling Methods

All surrogate model building methods chosen in this study do not utilize gradients and treat any function as a black box, which spills out a function value when given an input vector. The chosen methods for building models in Python are from the SMT, GPy, and scikit-learn libraries [2] – [4]. Although a plethora of similar libraries in Python exist, SMT has been referenced as the most popular open-source surrogate modeling library [2] and the GPy library has been proven to perform at similar or superior levels to libraries in R (DiceKriging, GPfit, laGP, mlegp) and C++ [5]. As alluded to earlier, commercial tools like SUMO and DACE in MATLAB were not considered.

The methods implemented in this test are shown below in Table 1. Other Python packages that were considered for the test include MVRSM and XGBoost, as discussed in [6], but their implementation required a deeper understanding of machine learning and hyperparameter tuning. Surrogate models such as RMTS (as implemented in the SMT library) that only work for problems with lower dimensionality were not considered in the comparison.

*Table 1. List of surrogate models used in the benchmark test*

<b>SMT</b>	<b>Other Python</b>	<b>Dakota</b>
Least-squares approximation (LS)	GPRegression (GPy package)	Gaussian Process
Second-order polynomial approximation (QP)	GaussianProcessRegressor or (scikit-learn package)	Multivariate adaptive regression splines (MARS)
Kriging (KRG)		Neural Network (NN)
Kriging with Partial Least Squares (KPLS)		Radial basis functions (RBF)
KPLSK (KPLS variant)		Linear polynomial
GPX (Kriging variant)		Quadratic polynomial
Radial basis functions (RBF)		Cubic polynomial
Inverse-distance weighting (IDW)		Moving Least Squares

From the GPy and scikit-learn packages, standard Gaussian Process models were feasibly implemented as the total number of sample points (sometimes they are called Design of Experiments (DoE) points) for each function is below or near 1000. A sample point refers to a point that was evaluated using the true function to train the surrogate models. Models that use more than 1000 sample points are advised to implement a Sparse Gaussian Process instead to boost performance.

The Python models implemented in this test largely use default settings, whereas Dakota models purely utilize their default settings. The modified changes for Python models are fully documented in the README file in the “surrogates\_python” folder for the Python methods in the benchmark source code. The changes made affected the GPy and scikit-learn GP models, as initial models built with their default settings yielded severely inaccurate metrics. For the final comparison, modified parameters were used. The accuracy metrics values derived from tests run using the models with default settings can be made available upon request.

All the non-experimental global surrogate building models in Dakota were tested. For background, Dakota is a state-of-the-art tool built specifically for engineers to more rapidly prototype and create new ideas. As listed on their website, it is a toolkit packed with a multitude of functions involving surrogate model building, among other operations including optimization and uncertainty quantification.

The code used in this test is discussed further in detail in the README files for both the Dakota and Python implementations.

## Test Functions

13 of the 15 test functions found in [7] were used, and their formulations and bounds are as follows in Table 2, with the function numbers corresponding to the function numbers in the publication:

*Table 2. Functions used for benchmarking surrogate modeling methods*

$f1(x) = (x_1 - 1)^2 + (x_{10} - 1)^2 + 10 \sum_{i=1}^9 (10 - i)(x_i^2 - x_{i+1})^2$	$-3 \leq x_i \leq 2, i = 1, \dots, 10$
$f2(x) = [\sum_{i=1}^{10} i^3 (x_i - 1)^2]^3$	$-3 \leq x_i \leq 3, i = 1, \dots, 10$
$f3(x) = \sum_{i=1}^{10} [100(x_i - x_{i+10})^2 + (x_i - 1)^2]$	$-3 \leq x_i \leq 5, i = 1, \dots, 10$
$f4(x) = \sum_{i=1}^5 [100(x_i^2 + x_{i+5})^2 + (x_i - 1)^2 + 90(x_{i+10}^2 + x_{i+15})^2 + (x_{i+10} - 1)^2 + 10.1((x_{i+5} - 1)^2 + (x_{i+15} - 1)^2) + 19.8(x_{i+5} - 1)(x_{i+15} - 1)]$	$-3 \leq x_i \leq 5, i = 1, \dots, 5$
$f5(x) = \sum_{i=1}^5 [(x_i + 10 * x_{i+5})^2 + 5 * (x_{i+10} - x_{i+15})^2 + (x_{i+5} - 2 * x_{i+10})^4 + 10 * (x_i - x_{i+15})^4]$	$-2 \leq x_i \leq 5, i = 1, \dots, 5$
$f7(x) = (x^T A x)^2, A = \text{diag}(1, 2, 3, \dots, 30)$	$-2 \leq x_i \leq 3, i = 1, \dots, 30$
$f8(x) = \sum_{i=1}^{29} [100 * (x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$-2 \leq x_i \leq 2, i = 1, \dots, 30$
$f9(x) = x^T A x - 2x^1, A = \begin{bmatrix} 1 & 1 & & & 0 \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & \dots & \dots & \dots & 0 \\ & & & \dots & \dots & \dots \\ 0 & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}$	$0 \leq x_i \leq 25, i = 1, \dots, 20$
$f10(x) = \sum_{i=1}^{20} x_i^2 + \left[ \sum_{i=1}^{20} \frac{1}{2} i x_i \right]^2 + \left[ \sum_{i=1}^{20} \frac{1}{2} i x_i \right]^4$	$0 \leq x_i \leq 5, i = 1, \dots, 20$

$f11(x) = \sum_{i=1}^{65} [y_i - (x_1 e^{-x_5 t_i} + x_2 e^{-x_6(t_i-x_9)^2} + x_3 e^{-x_7(t_i-x_{10})^2} + x_4 e^{-x_8(t_i-x_{11})^2})]^2,$ $y = \begin{pmatrix} 1.366 & 1.191 & 1.112 & 1.013 & 0.991 & 0.885 & 0.831 & 0.847 & 0.786 & 0.725 & 0.746 \\ 0.679 & 0.608 & 0.655 & 0.616 & 0.606 & 0.602 & 0.626 & 0.651 & 0.724 & 0.649 & 0.649 \\ 0.694 & 0.644 & 0.624 & 0.661 & 0.612 & 0.558 & 0.533 & 0.495 & 0.500 & 0.423 & 0.395 \\ 0.375 & 0.372 & 0.391 & 0.396 & 0.405 & 0.428 & 0.429 & 0.523 & 0.562 & 0.607 & 0.653 \\ 0.672 & 0.708 & 0.633 & 0.668 & 0.645 & 0.632 & 0.591 & 0.559 & 0.597 & 0.625 & 0.739 \\ 0.710 & 0.729 & 0.720 & 0.636 & 0.581 & 0.428 & 0.292 & 0.162 & 0.098 & 0.069 & 0.054 \end{pmatrix}$	$0 \leq x_1 \leq 1.6; 0 \leq x_i \leq 2, i = 2, \dots, 5; 2 \leq x_i \leq 8, i = 6, \dots, 8; 1 \leq x_9 \leq 6, 4.5 \leq x_i \leq 6, i = 10, 11;$
$f12(x) = 100000 \prod_{i=1}^{11} x_i^{\alpha_i}, \alpha = (-0.00133, -0.00227, -0.00248, -4.67, \dots, -0.00814)$	$0.1 \leq x_i \leq 100; i = 1, \dots, 11$
$f13(x) = \sum_{i=1}^{14} \frac{a_i}{x_i}, a = (12842.275, 634.25, \dots, 940.838)$	$0.0001 \leq x_i, i = 1, \dots, 11; x_i \leq 0.04, i = 1, \dots, 5; x_i \leq 0.04, i = 1, \dots, 5; x_i \leq 0.03, i = 6, \dots, 14;$
$f14(x) = \sum_{i=1}^{30} \alpha_i(x), \alpha_i(x) = 420x_i + (i-15)^3 + \sum_{j=1}^{30} v_{ij} [(\sin(\log(v_{ij})))^5 + (\cos(\log(v_{ij})))^5], v_{ij} = \sqrt{x_j^2 + \frac{i}{j}}$	$-2 \leq x_i \leq 2, i = 1, \dots, 30$

## Methodology

The accuracy metrics used for measuring the accuracy of the surrogate model training methods were  $R^2$ , RAAE, and RMAE as shown below in Equations 1-3.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad \text{Equation 1}$$

$$\text{RAAE} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i - \bar{y}|} \quad \text{Equation 2}$$

$$\text{RMAE} = \frac{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|}{\frac{1}{n} \sum_{i=1}^n y_i} \quad \text{Equation 3}$$

Each of these gives a different perspective on the accuracy of the model. RAAE stands for Relative Average Absolute Error, indicating an average relative percentage error in the predictions. RMAE stands for Relative Maximum Absolute Error, which compares the ability of predictions to capture the outliers.  $R^2$  measures the global behavior of a surrogate model, which has a value between 0 and 1, if measured on the training points. The closer the  $R^2$  value to one, the more accurate the model. In this work, to test the extrapolation capability,  $R^2$  values are calculated on the testing points, as is done in literature. RAAE and RMAE are desired to be close to zero.

For each function and each model, 2000 prediction points were generated and compared to the benchmark value for that function. The number of training sample points for each function was determined by the OASIS.AI benchmark in order to get a fair comparison, as shown in Figure 1 for Function 4 surrogate models. This means, for a given computational budget, we would see which method leads to a more accurate surrogate model.

OASIS BASELINE (Function 3)						
	R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions
	1	7.00E-16	5.80E-15	20	351	2000
Comments:						
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	0.00E+00	-2.39E-02	8.15E-01	3.00E+00	LS	
	5.01E-02	1.00E+00	1.02E-13	2.35E-13	QP	
	4.80E+02	1.00E+00	1.35E-04	5.93E-04	KRG	
	1.98E+00	1.29E-01	7.52E-01	2.88E+00	KPLS	
	5.79E+01	1.00E+00	2.08E-04	9.13E-04	KPLSK	
	5.98E+00	9.52E-01	7.10E-02	3.07E-01	GPX	
	1.05E-01	9.95E-01	5.83E-02	2.62E-01	IRBF	
	0.00E+00	1.11E-02	8.00E-01	3.06E+00	IDW	
Scikit-learn	2.47E+00	-6.71E-04	8.05E-01	3.07E+00	Gpy	
	1.17E+00	9.71E-01	1.36E-01	5.95E-01	scikit	
Dakota	1.02E+02	5.58E-01	5.20E-01	2.83E+00	gaussian_process dakota	
	9.40E+01	5.89E-02	7.70E-01	3.75E+00	mars	
	9.02E+01	-8.34E-02	8.12E-01	4.94E+00	neural_network	
	1.04E+02	-6.35E+01	1.37E+00	3.27E+02	radial_basis	
	1.15E+02	-5.77E-02	8.15E-01	4.89E+00	polynomial_linear	
	1.12E+02	1.00E+00	4.08E-10	1.89E-09	polynomial_quadratic	
	5.18E+02	1.00E+00	1.24E-08	9.07E-08	polynomial_cubic	
total time (m): 28.06580283						

Figure 1. Example display format for accuracy metric logging for individual functions

The benchmarking of models was done function by function, and the results can be found in Appendix A, from Figures A1 to A13. After collecting the data for all 17 methods across all 13 functions, the  $R^2$ , RMAE, and RAAE results were then grouped as shown in Appendix B1 to B3, and relative values were calculated and used to graph visual comparisons between the models. The raw values for RAAE and RMAE can be found in Appendix B, from Figures B4 to B5.

The  $R^2$  values were relative, as calculated using Equation 4. Negative values indicate that the model's  $R^2$  score is lower than the benchmark and thus has inferior performance. Before graphing, the relative  $R^2$  values smaller than -10 were deleted, as they indicated a model  $R^2$  of zero or less.

$$relative R^2 = - \frac{(OASIS R^2 - model R^2)}{OASIS R^2} \quad \text{Equation 4}$$



Similar adjustments were made for RMAE and RAAE scores, using Equations 5 and 6, respectively, on the raw values to indicate superior performance to the benchmark with positive values. Again, negative values show that the model has greater error and has inferior performance to the benchmark.

$$\text{relative RAAE} = \frac{(\text{OASIS RAAE} - \text{model RAAE})}{\text{OASIS RAAE}} \quad \text{Equation 5}$$

$$\text{relative RMAE} = \frac{(\text{OASIS RMAE} - \text{model RMAE})}{\text{OASIS RMAE}} \quad \text{Equation 6}$$

RAAE values vary drastically. For example, in Function 3, it is 7.85E-16, which is several orders of magnitude smaller than the average RAAE of the other functions. This results in outliers when graphing alongside the results of other functions. For better visual comparison, relative RAAE values less than -10 were filtered. Graphical boxplots with filtered RMAE and RAAE data are included in the “Results” section of the report, indicating the quartile ranges, with an inclusive mean and without outliers. In addition to accuracy metrics, the total time for each surrogate model construction per function was also documented. Before graphing, each value was transformed with a base-10 logarithm in order to better capture the non-linear relationships. For example, model construction times ranged from a tenth of a second to 15 minutes. Graphing this relationship in a linear format would not provide a helpful visual comparison.

Regarding hardware specifications, two laptops were used for running the benchmark tests to speed up the process. A G5 KB laptop (16GB memory, 2.5GHz) conducted all the tests in Python, along with Dakota tests on Functions 1 and 7-14. The rest were conducted on a UX425UA ASUS ZenBook (8GB memory, 2.10GHz).

## Results

As mentioned before, in all graphs presented below, values above zero indicate superior performance to the OASIS.AI benchmark.

After the extreme outliers are removed, the box plots of the RAAE and RMAE in Figures 2 and 3 were created. The “full scale” graphical comparison before this processing was done is listed in Appendix C to show the full scale of data points from this test, as it is an ineffective visual comparison. The box plots of the relative R<sup>2</sup> data are shown below in Figure 4.

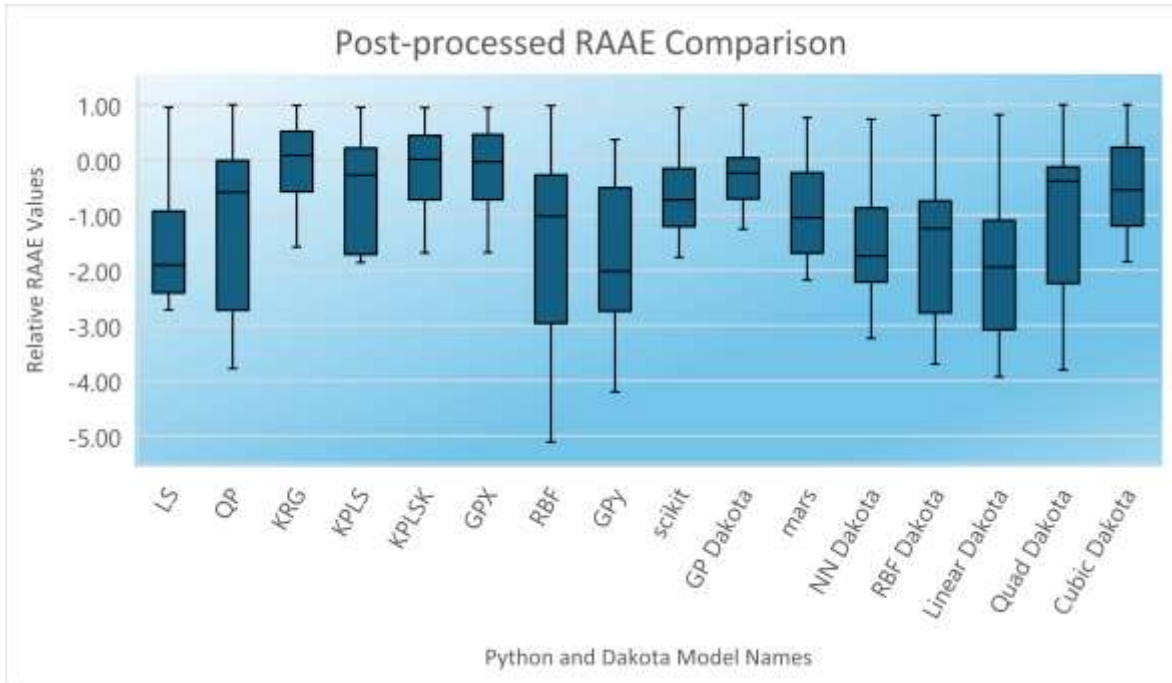


Figure 2. Post-processed relative RAAE scores of surrogate tests against OASIS.AI benchmark

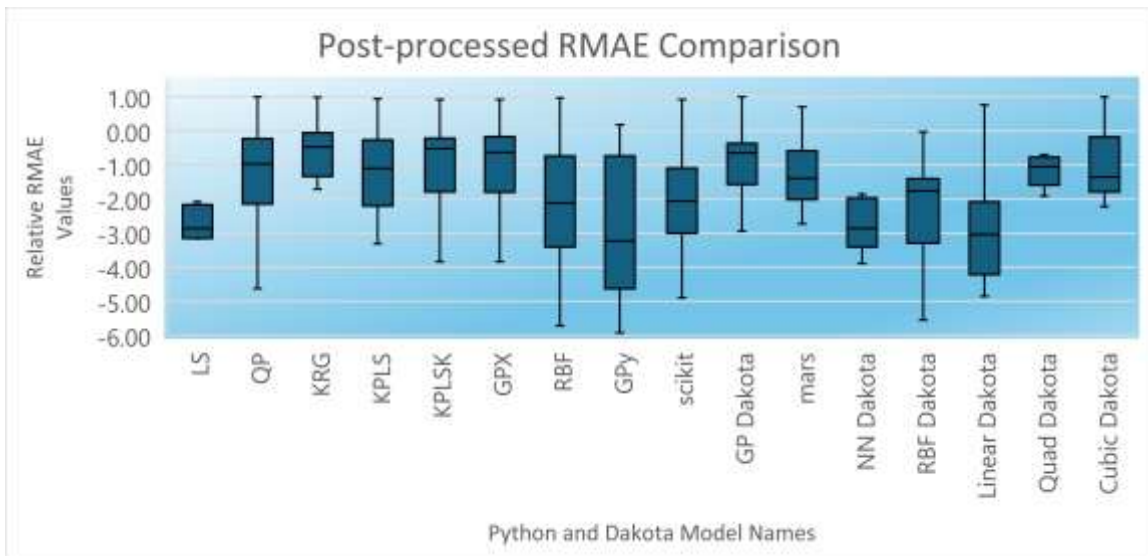


Figure 3. Post-processed relative RMAE scores of surrogate tests against OASIS.AI benchmark

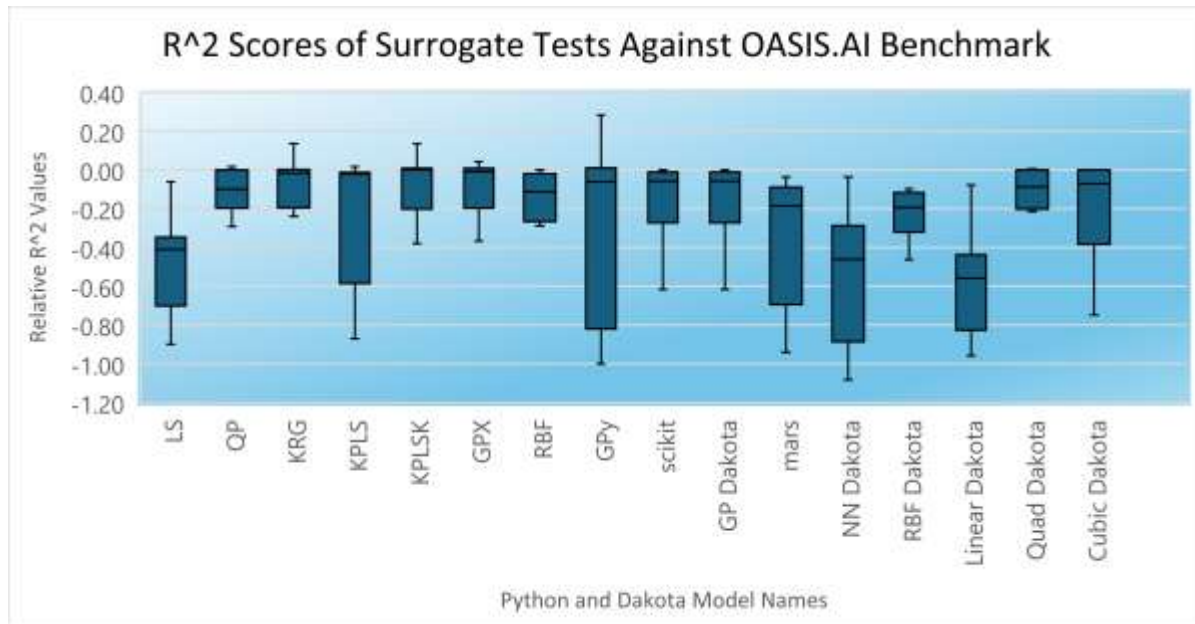


Figure 4. Relative  $R^2$  scores of surrogate tests against OASIS.AI benchmark

The SMT IDW model was removed from the comparison as it was consistently one of the worst performers, with its highest  $R^2$  score being 0.125 for all functions. Dakota's Moving Least Squares model was also removed as it completed less than half of the tests successfully. Additionally, in the event that it successfully ran, its results were extremely inaccurate.

The full data for the Relative RAAE, RMAE, and  $R^2$  are listed in Appendices B4 and B5. In the RAAE and RMAE tables, cells with slashes ("/") indicate N/A values because the associated model failed to build a surrogate for that function in under 20 minutes. In the  $R^2$  table, this extends to include deleted values based on the criteria discussed earlier, which include values less than -10. Each negative integer step away 0 indicates a linear deterioration of the model accuracy. For example, a relative RAAE value of -2 indicates that a model has an RAAE value three times as large as the benchmark.

As a quick note, data from the test on Function 12 was not included in the graphical comparison, as the methods performed equally poorly in characterizing the function.

It is also to be noted that many of the functions were cut off from the final graphical comparison. As stated earlier, none of the results from Function 12 were included in the graphical comparison. The cutoff for  $R^2$  ( $\leq -10$ ) removed values that performed very poorly, and the cutoff for the

other two metrics removed values that were ten times worse than the benchmark on a logarithmic scale. As a result, the number of remaining data points used to construct the boxplots in Figures 2-4 are listed below in Table 3.

*Table 3. Number of post-processed sample points remaining for each method*

	KRG	KPLSK	GPX	GP Dakota	Quad Dakota	Cubic Dakota
<b>R<sup>2</sup></b>	8	12	12	12	10	5
<b>RMAE / RAAE</b>	6	8	8	6	8	3

## Discussion

Overall, OASIS.AI demonstrates superior performance for surrogate modelling. OASIS.AI has moderately stronger relative RAAE and R<sup>2</sup> scores than the majority of other methods and definitively outperforms the rest in terms of relative RMAE scores.

Graphs of the error metrics from the top 3 methods from each software are shown in Figures 5-7, with accompanying median values for the metrics from each method shown in Table 4. In these figures, the first three boxes on the left correspond to metrics from the Python models, and the latter three correspond to metrics from the Dakota models. The dots shown on the graphs are outlier cases. All 3 Python methods beat the Dakota methods in all three error metrics. GPX performed slightly worse than the other two methods, in terms of R<sup>2</sup> and RAAE, but most notably in RMAE.

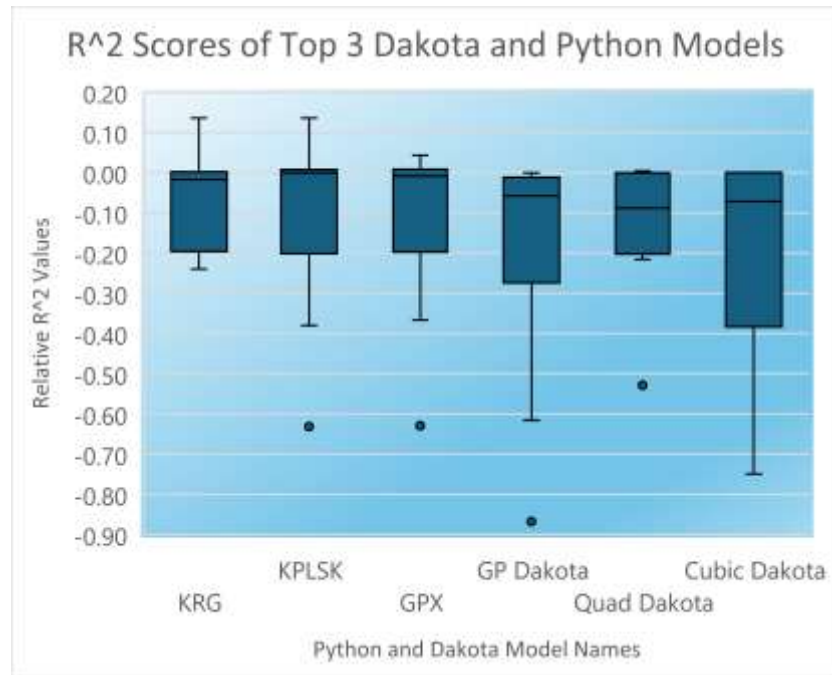


Figure 5. Top 6 methods comparison for  $R^2$

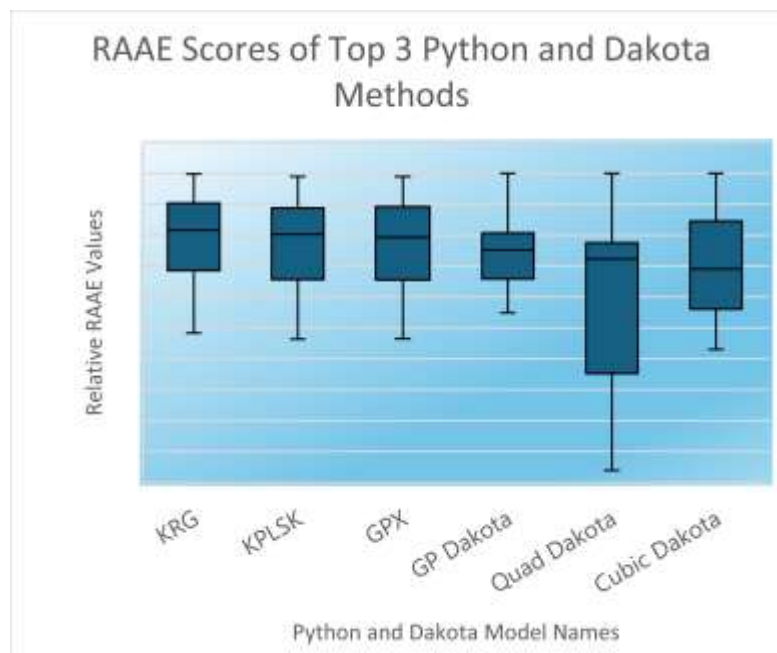


Figure 6. Top 6 methods comparison for RAAE

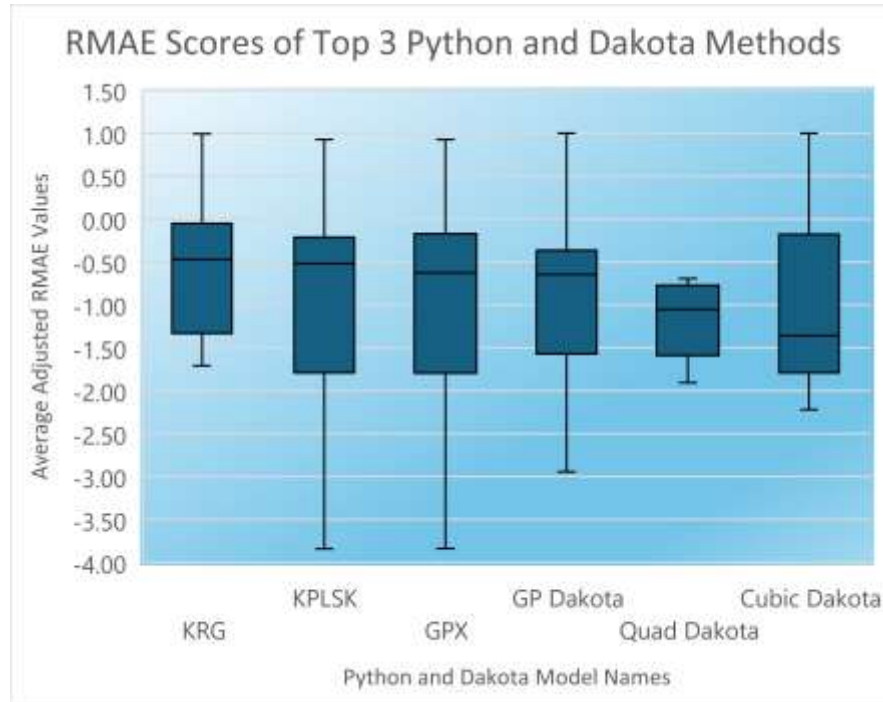


Figure 7. Top 6 methods comparison for RMAE

Table 4. Summary of the median of error metrics collected over the top 3 Python methods (first three columns after the header column) and top 3 Dakota methods (last three columns)

	KRG	KPLSK	GPX	GP Dakota	Quad Dakota	Cubic Dakota
Relative RAAE	0.08	0.01	-0.03	-0.24	-0.38	-0.55
Relative RMAE	-0.47	-0.52	-0.63	-0.65	-1.05	-1.36
Relative R <sup>2</sup>	-0.02	0.00	-0.01	-0.06	-0.09	-0.07

All three of Python methods demonstrate consistent performance as they show lower variance in prediction error across all three accuracy metrics when compared to the other methods in Figure 5 to Figure 7. However, Kriging failed to construct a surrogate model four separate times, and both Kriging and KPLSK displayed significantly longer runtimes than GPX, as shown in Figure 8. Also, the only other method that failed to successfully generate a surrogate model in under 20

minutes was GP Dakota, which failed only once. Note that an increase of 1 on the graph indicates a performance time of 10 times longer. Thus, in situations where the time budget is limited, GPX may be the ideal surrogate modeling implementation to use, despite its slightly weaker performance than the other two Python methods.

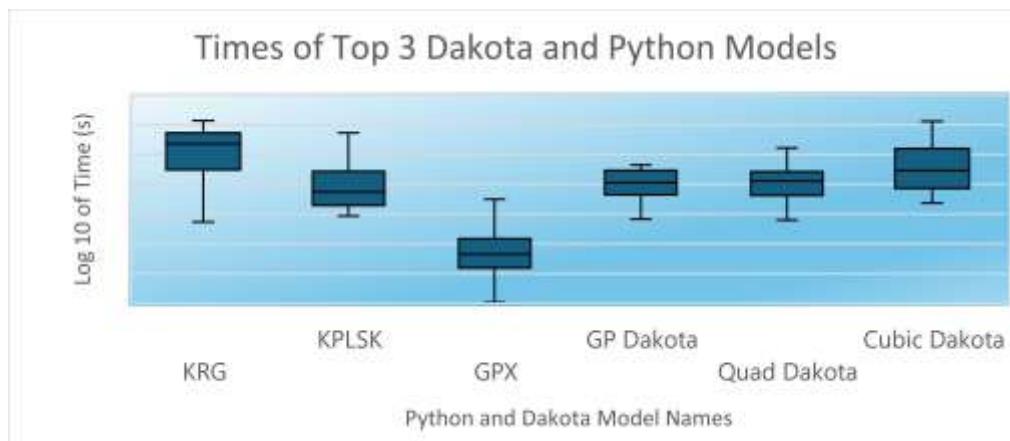


Figure 8. Time comparison between the top 3 methods from each group (note the non-linear scale of runtime for each function)

Zooming in on the box plots of its error metrics, we can better understand the performance of the GPX model, as shown in Figure 9. Analyzing the median value, there is nearly 0 difference between the  $R^2$  score of GPX and the benchmark value. Its accuracy in terms of RAAE is also nearly on par with OASIS.AI, with half of the tests yielding than 5% from the benchmark value and a quarter outperforming the benchmark value. However, the RMAE score is larger than the scores from the other two Python methods, -0.63 as compared with -0.47 and -0.52. Similar to the case for RAAE, each negative integer step away from 0 indicates a linear deterioration of the model accuracy. For example, a relative RAME value of -2 indicates that a model has an RMAE value three times as large as the benchmark.

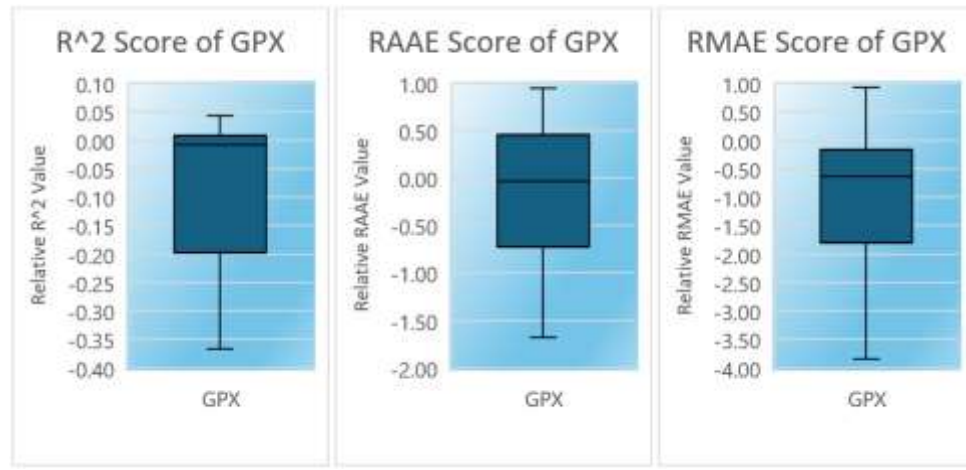


Figure 9. Zoomed in accuracy metrics score of the GPX model

## Conclusion

In this paper, several methods for surrogate modeling were tested on a group of complex black box functions with varying DoE and dimensionality. The accuracy metrics of these models were then compared with the metrics from OASIS.AI, provided by Empower Operations Corp. After analyzing the results, it is evident that OASIS.AI remains superior in performance. On average, from the chosen Python and Dakota libraries tested, open-source software does not outperform OASIS.AI. However, there are several promising methods that perform at nearly similar levels. This list includes the Python SMT library methods KRG, KPLSK, and GPX, with GPX demonstrating the quickest runtime with a small prediction accuracy tradeoff. From a comparison of the top three methods in Python software and Dakota, respectively, it is evident that Python methods slightly outperform Dakota.

Depending on the surrogate modeling requirements of the user, Python's SMT library may be suitable for their needs. If the user does not mind slightly lower performance, SMT's KRG, KPLSK, and GPX models will likely perform at the desired levels. The top three methods from Dakota's package may also be a suitable choice, given the low amount of code setup needed in order to run the built-in methods on user-defined problems.



## References

- [1] Erickson, Collin B., Bruce E. Ankenman, and Susan M. Sanchez. "Comparison of Gaussian Process Modeling Software." *European Journal of Operational Research*, vol. 266, no. 1, 2018, pp. 179–192. *ScienceDirect*, <https://doi.org/10.1016/j.ejor.2017.10.002>.
- [2] Saves, P., Lafage, R., Bartoli, N., Diouane, Y., Bussemaker, J., Lefebvre, T., Hwang, J. T., Morlier, J., & Martins, J. R. R. A. (2024). *SMT 2.0: A Surrogate Modeling Toolbox with a focus on hierarchical and mixed variables Gaussian processes*. *Advances in Engineering Software*, 188, 103571. <https://doi.org/10.1016/j.advengsoft.2023.103571>
- [3] GPy. (2012). *GPy: A Gaussian process framework in Python*. Retrieved from <http://github.com/SheffieldML/GPy>
- [4] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (pp. 108–122).
- [5] ResiBots. *Regression benchmarks for Gaussian Processes in Limbo*. Retrieved June 10, 2025, from [https://resibots.eu/limbo/reg\\_benchmarks.html](https://resibots.eu/limbo/reg_benchmarks.html)
- [6] Bliet, L., Guijt, A., Verwer, S., & de Weerd, M. (2021). *Black-box mixed-variable optimisation using a surrogate model that satisfies integer constraints*. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '21)* (pp. 1851–1859). Association for Computing Machinery. <https://doi.org/10.1145/3449726.3463136>
- [7] Shan, S., & Wang, G. G. (2011). *Turning black-box functions into white functions* [Article 031003]. *Journal of Mechanical Design*, 133(3). <https://doi.org/10.1115/1.4002978>

## Appendix A.

### Surrogate model test results per function

OASIS BASELINE (Function 1)						
R square	RAAE	RMAE	Dimensionality		Number of Training Samples	Number of predictions
0.933	0.1989	0.9692	10		150	2000
Comments:						
	Python/Dakota					
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	3.74E-03	5.99E-01	5.03E-01	2.99E+00	LS	
	0.00E+00	8.54E-01	3.04E-01	1.52E+00	QP	
	2.33E+01	9.01E-01	2.46E-01	1.26E+00	KRG	
	6.29E-01	9.09E-01	2.35E-01	1.68E+00	KPLS	
	4.28E+00	9.01E-01	2.46E-01	1.26E+00	KPLSK	
	1.85E-01	8.97E-01	2.50E-01	1.27E+00	GPX	
	2.14E-02	8.82E-01	2.71E-01	1.43E+00	RBF	
	0.00E+00	1.50E-01	7.23E-01	4.64E+00	IDW	
Scikit-learn	1.26E+01	8.87E-01	1.04E+00	6.72E+00	GPy	
	3.97E-01	7.21E-01	4.05E-01	2.27E+00	scikit	
Dakota	2.60E+01	9.17E-01	2.24E-01	1.32E+00	gaussian_process dakota	
	2.53E+01	8.83E-01	2.67E-01	1.71E+00	mars	
	2.50E+01	5.73E-01	4.90E-01	3.74E+00	neural_network	
	2.49E+01	7.59E-01	3.89E-01	2.23E+00	radial_basis	
	2.48E+01	5.28E-01	5.40E-01	3.92E+00	polynomial linear	
	2.48E+01	8.63E-01	2.91E-01	1.64E+00	polynomial quadratic	
	4.84E+01	5.99E+01	5.74E+00	3.65E+01	polynomial cubic	
total time (m): 4.00889						

Figure A1. Surrogate model results for function 1

OASIS BASELINE (Function 2)							
R square		RAAE	RMAE	Dimensionality		Number of Training Samples	Number of predictions
0.9896		0.0675	0.7571	10		632	2000
Comments:							

Figure A2. Surrogate model results for function 2

OASIS BASELINE (Function 3)						
R square		RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions
1		7.00E-16	5.80E-15	20	351	2000
Comments:						
	Python/Dakota					
SMT	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
	0.00E+00	-2.39E-02	8.15E-01	3.00E+00	LS	
	5.01E-02	1.00E+00	1.02E-13	2.35E-13	QP	
	4.80E+02	1.00E+00	1.35E-04	5.93E-04	KRG	
	1.98E+00	1.29E-01	7.52E-01	2.88E+00	KPLS	
	5.79E+01	1.00E+00	2.08E-04	9.13E-04	KPLSK	
	5.98E+00	9.92E-01	7.10E-02	3.07E-01	GPX	
	1.05E-01	9.95E-01	5.83E-02	2.62E-01	RBF	
	0.00E+00	1.11E-02	8.00E-01	3.06E+00	IDW	
	Scikit-learn	2.47E+00	-6.71E-04	8.05E-01	3.07E+00	GPy
1.17E+00		9.71E-01	1.36E-01	5.95E-01	scikit	
Dakota	1.02E+02	5.58E-01	5.20E-01	2.83E+00	gaussian_process dakota	
	9.40E+01	5.89E-02	7.70E-01	3.75E+00	mars	
	9.02E+01	-8.34E-02	8.12E-01	4.94E+00	neural_network	
	1.04E+02	-6.35E+01	1.57E+00	3.27E+02	radial_basis	
	1.15E+02	-5.77E-02	8.15E-01	4.89E+00	polynomial_linear	
	1.12E+02	1.00E+00	4.08E-10	1.89E-09	polynomial_quadratic	
	5.18E+02	1.00E+00	1.24E-08	9.07E-08	polynomial_cubic	
total time (m): 28.06580283						

Figure A3. Surrogate model results for function 3

OASIS BASELINE (Function 4)						
	R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions
	1	7.85E-16	5.80E-15	20	361	2000
Comments:						
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	0.00E+00	5.91E-01	5.12E-01	2.31E+00	LS	
	1.58E-02	7.07E-01	4.31E-01	2.15E+00	QP	
	5.21E+02	8.12E-01	3.41E-01	1.96E+00	KRG	
	2.66E+00	9.18E-01	2.26E-01	1.15E+00	KPLS	
	6.03E+01	8.12E-01	3.41E-01	1.96E+00	KPLSK	
	6.45E+00	8.17E-01	3.37E-01	1.98E+00	GPX	
	1.27E-01	7.11E-01	4.29E-01	2.05E+00	RBF	
	0.00E+00	6.74E-02	7.69E-01	4.19E+00	IDW	
	6.04E+00	5.56E-01	9.66E-01	6.28E+00	GPy	
Scikit-learn	1.68E+00	7.03E-01	4.28E-01	2.64E+00	scikit	
Dakota	1.05E+02	8.02E-01	2.72E-01	1.31E+00	gaussian_process dakota	
	9.73E+01	8.88E-01	2.62E-01	1.51E+00	mars	
	9.90E+01	4.64E-01	5.68E-01	3.44E+00	neural_network	
	9.91E+01	4.73E+00	9.54E-01	5.41E+01	radial_basis	
	1.02E+02	4.51E-01	5.88E-01	2.96E+00	polynomial linear	
	1.13E+02	7.84E-01	3.66E-01	2.00E+00	polynomial quadratic	
	5.71E+02	5.22E+03	5.57E+01	1.13E+02	polynomial cubic	
total time (m): 29.73384967						

Figure A4. Surrogate model results for function 4

OASIS BASELINE (Function 5)						
	R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions
	0.8909	0.2626	1.0231	20	523	2000
Comments: KRG had the same result as KPLSK, taking 10x longer to run						
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	1.12E-02	8.96E-02	7.63E-01	4.24E+00	LS	
	0.00E+00	7.16E-01	4.17E-01	2.02E+00	QP	
	1.17E+03	6.77E-01	4.43E-01	2.77E+00	KRG	
	5.27E+00	1.23E-01	7.51E-01	4.41E+00	KPLS	
	1.44E+02	6.77E-01	4.43E-01	2.77E+00	KPLSK	
	1.68E+01	6.76E-01	4.44E-01	2.79E+00	GPX	
	3.52E-01	7.16E-01	4.18E-01	2.04E+00	RBF	
	0.00E+00	1.94E-02	7.95E-01	4.11E+00	IDW	
	7.53E+00	3.19E-01	8.53E-01	5.14E+00	GPy	
Scikit-learn	4.88E+00	6.54E-01	4.59E-01	2.94E+00	scikit	
Dakota	1.71E+02	6.10E-01	4.79E-01	2.85E+00	gaussian_process dakota	
	1.60E+02	3.89E-01	6.10E-01	3.54E+00	mars	
	1.65E+02	1.35E-01	7.22E-01	4.01E+00	neural_network	
	1.66E+02	1.02E+00	8.50E-01	1.85E+01	radial_basis	
	1.64E+02	4.68E-02	7.72E-01	4.68E+00	polynomial linear	
	1.63E+02	8.01E-01	3.43E-01	1.89E+00	polynomial quadratic	
	5.56E+02	2.10E+04	1.11E+02	7.62E+02	polynomial cubic	
total time (m): 48.2573695						

Figure A5. Surrogate model results for function 5

OASIS BASELINE (Function 7)						
	R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions
	9.75E-01	1.27E-01	6.08E-01	30	2397	2000
Comments:						
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	1.57E-02	6.46E-01	4.70E-01	2.35E+00	LS	
	3.22E-02	9.91E-01	6.89E-02	5.39E-01	QP	
	\	\	\	\	KRG	took over 20 minutes
	1.63E+02	9.91E-01	7.13E-02	4.83E-01	KPLS	
	6.96E+02	9.83E-01	9.33E-02	8.48E-01	KPLSK	
	4.56E+00	9.92E-01	6.75E-02	5.30E-01	RBF	
	0.00E+00	4.35E-02	7.73E-01	4.09E+00	IDW	
	1.53E+02	-1.25E-08	7.92E-01	4.17E+00	GPy	
Scikit-learn	2.43E+02	9.89E-01	8.04E-02	4.97E-01	scikit	
	7.74E+00	9.60E-01	1.57E-01	8.17E-01	scikit	
Dakota	\	\	\	\	gaussian_process dakota	stalled at 2397 evals, computer shut down multiple times midrun
	7.53E+02	7.54E-01	3.86E-01	2.27E+00	mars	
	9.21E+02	6.24E-01	4.66E-01	2.98E+00	neural_network	
	7.57E+02	8.54E-01	2.84E-01	2.92E+00	radial_basis	
	7.34E+02	3.58E-01	6.24E-01	3.56E+00	polynomial linear	
	4.07E+02	9.79E-01	9.91E-02	1.25E+00	polynomial quadratic	
	9.79E+02	-1.78E+01	3.36E+00	1.92E+01	polynomial cubic	
total time (m): 96.952532						

Figure A6. Surrogate model results for function

OASIS BASELINE (Function 8)						
	R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions
	1	2.14E-15	1.11E-14	30	790	2000
Comments:						
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	1.57E-02	4.23E-01	6.07E-01	2.72E+00	LS	
	1.81E-01	1.18E-02	7.88E-01	3.86E+00	QP	
	\	\	\	\	KRG	took over 20 minutes
	1.03E+01	4.99E-01	5.57E-01	2.44E+00	KPLS	
	6.51E+02	3.68E-01	6.31E-01	2.79E+00	KPLSK	
	9.34E+01	3.71E-01	6.29E-01	2.89E+00	GPX	
	6.12E-01	5.18E-03	7.91E-01	3.86E+00	RBF	
	1.59E-02	3.18E-02	7.89E-01	3.65E+00	IDW	
Scikit-learn	1.34E+01	-9.77E-09	8.02E-01	3.73E+00	GPy	
	8.07E+00	3.84E-01	6.21E-01	2.92E+00	scikit	
Dakota	1.71E+02	7.77E-01	3.78E-01	1.71E+00	gaussian_process dakota	
	1.30E+02	1.70E-01	7.26E-01	3.82E+00	mars	
	1.35E+02	1.13E-01	7.46E-01	3.93E+00	neural_network	
	1.33E+02	7.27E-01	4.15E-01	2.49E+00	radial_basis	
	1.31E+02	1.84E-01	7.25E-01	3.27E+00	polynomial linear	
	1.32E+02	4.72E-01	5.73E-01	2.90E+00	polynomial quadratic	
	9.72E+02	-4.44E+03	5.12E+00	2.88E+01	polynomial cubic	
total time (m): 43.025347						

Figure A7. Surrogate model results for function 8



OASIS BASELINE (Function 9)						
	R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions
	1	1.16E-15	2.14E-15	20	385	2000
Comments:						
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	7.60E-03	1.12E-01	7.54E-01	3.96E+00	LS	
	0.00E+00	1.00E+00	1.22E-14	5.23E-14	QP	
	\	\	\	\	KRG	took over 20 minutes
	2.39E+00	1.60E-01	7.34E-01	3.47E+00	KPLS	
	8.51E+01	1.00E+00	7.08E-04	3.23E-03	KPLSK	
	7.05E+00	9.92E-01	7.06E-02	3.32E-01	GPX	
	7.91E-02	9.90E-01	7.80E-02	3.30E-01	RBF	
	0.00E+00	2.25E-02	7.95E-01	3.73E+00	IDW	
Scikit-learn	3.45E+00	-9.20E-04	8.05E-01	3.75E+00	GPy	
	2.24E+00	9.52E-01	1.74E-01	7.41E-01	scikit	
Dakota	6.70E+01	6.47E-01	4.66E-01	2.72E+00	gaussian_process dakota	
	6.38E+01	1.21E-01	7.50E-01	4.23E+00	mars	
	6.48E+01	8.57E-02	7.60E-01	3.73E+00	neural_network	
	6.50E+01	#####	2.77E+86	5.54E+89	radial_basis	
	6.44E+01	4.19E-02	7.80E-01	3.57E+00	polynomial_linear	
	6.49E+01	1.00E+00	7.31E-10	3.22E-09	polynomial_quadratic	
	3.04E+02	1.00E+00	2.81E-07	1.91E-06	polynomial_cubic	
total time (m): 13.24002858						

Figure A8. Surrogate model results for function 9

OASIS BASELINE (Function 10)						
	R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions
	1.00E+00	9.30E-03	4.27E-02	20	1031	2000
Comments:						
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	0.00E+00	9.38E-01	1.74E-01	2.88E+00	LS	
	1.61E-02	total time	2.37E-02	5.94E-01	QP	
	8.16E+02	1.00E+00	9.98E-05	3.94E-04	KRG	
	2.71E+01	9.84E-01	8.93E-02	1.51E+00	KPLS	
	7.31E+02	1.00E+00	1.69E-02	2.07E-01	KPLSK	
	1.24E+02	1.00E+00	1.70E-02	2.07E-01	GPX	
	5.73E-01	9.99E-01	2.28E-02	5.71E-01	RBF	
	0.00E+00	1.02E-01	7.41E-01	6.24E+00	IDW	
Scikit-learn	2.98E+01	9.39E-01	1.10E+00	9.27E+00	GPy	
	1.19E+01	1.00E+00	1.59E-02	2.22E-01	scikit	
Dakota	2.10E+02	9.59E-01	1.24E-01	2.47E+00	gaussian_process dakota	
	1.91E+02	8.56E-01	2.75E-01	3.15E+00	mars	
	1.99E+02	9.64E-01	1.15E-01	2.91E+00	neural_network	
	1.90E+02	9.02E-01	2.07E-01	3.38E+00	radial_basis	
	1.89E+02	9.21E-01	1.73E-01	3.43E+00	polynomial_linear	
	1.78E+02	9.98E-01	2.73E-02	8.22E-01	polynomial_quadratic	
	3.07E+02	2.51E-01	6.62E-01	4.91E+00	polynomial_cubic	
total time (m): 53.39498667						

Figure A9. Surrogate model results for function 10

OASIS BASELINE (Function 11)						
R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions	
7.93E-01	3.43E-01	2.04E+00	11	543	2000	
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	0.00E+00	4.88E-01	4.56E-01	8.49E+00	LS	
	1.58E-02	7.14E-01	3.48E-01	6.48E+00	QP	
	1.03E+03	9.00E-01	2.06E-01	3.36E+00	KRG	
	1.94E+01	7.80E-01	2.96E-01	5.89E+00	KPLS	
	1.90E+02	9.00E-01	2.06E-01	3.36E+00	KPLSK	
	7.63E+00	8.27E-01	2.78E-01	3.97E+00	GPX	
	3.24E-01	7.19E-01	3.44E-01	6.45E+00	RBF	
	0.00E+00	1.03E-01	6.35E-01	9.86E+00	IDW	
Scikit-learn	2.19E+02	8.96E-01	9.55E-01	1.19E+01	GPy	
	2.00E+01	7.40E-01	3.11E-01	7.37E+00	scikit	
Dakota	1.09E+02	8.09E-01	3.13E-01	2.79E+00	gaussian_process dakota	
	1.03E+02	7.13E-01	3.85E-01	2.86E+00	mars	
	1.05E+02	5.51E-01	4.41E-01	5.83E+00	neural_network	
	1.04E+02	4.26E-01	5.25E-01	5.72E+00	radial_basis	
	1.03E+02	4.41E-01	5.11E-01	6.33E+00	polynomial linear	
	1.03E+02	6.25E-01	4.26E-01	4.63E+00	polynomial quadratic	
	1.04E+02	4.89E-01	5.31E-01	4.82E+00	polynomial cubic	
total time (m): 36.96971633						

Figure A10. Surrogate model results for function 11

OASIS BASELINE (Function 12)						
R square	RAAE	RMAE	Dimensionality	Number of Training Samples	Number of predictions	
0.0167	4.83E-02	4.11E+01	11	266	2000	
Comments: Very difficult function to characterize						
Python/Dakota						
	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
SMT	1.58E-02	-5.08E-04	2.26E-02	4.47E+01	LS	
	1.56E-02	-5.07E-04	2.26E-02	4.47E+01	QP	
	1.42E+02	-5.08E-04	2.26E-02	4.47E+01	KRG	
	2.01E+00	-5.08E-04	2.26E-02	4.47E+01	KPLS	
	3.38E+01	-5.05E-04	2.26E-02	4.47E+01	KPLSK	
	1.04E+00	-5.01E-04	2.24E-02	4.47E+01	GPX	
	1.90E-03	-5.07E-04	2.26E-02	4.47E+01	RBF	
	0.00E+00	-5.08E-04	2.26E-02	4.47E+01	IDW	
Scikit-learn	2.31E+00	-5.08E-04	2.26E-02	4.47E+01	GPy	
	1.67E+00	-5.08E-04	2.26E-02	4.47E+01	scikit	
Dakota	4.89E+01	-5.15E-04	2.26E-02	4.47E+01	gaussian_process dakota	
	5.12E+01	-5.21E-05	2.26E-02	4.47E+01	mars	
	4.93E+01	-5.02E-04	2.25E-02	4.47E+01	neural_network	
	5.07E+01	-5.03E-04	2.24E-02	4.47E+01	radial_basis	
	4.99E+01	-4.78E-04	2.26E-02	4.47E+01	polynomial linear	
	4.95E+01	-4.81E-04	2.29E-02	4.47E+01	polynomial quadratic	
	6.94E+01	-5.03E-04	2.24E-02	4.47E+01	polynomial cubic	
total time (m): 9.201970333						

Figure A11. Surrogate model results for function 12

OASIS BASELINE (Function 13)						
R square	RAAE	RMAE	Dimensionality		Number of Training Samples	Number of predictions
6.73E-01	1.27E-01	5.54E+00	14		325	2000
Comments: KRG had the same result as KPLSK, taking 10x longer to run						
	Python/Dakota					
SMT	time (s)	R square	RAAE	RMAE	Method Names	Comments/Specifications
	1.65E-02	1.18E-01	3.95E-01	1.80E+01	LS	
	0.00E+00	-7.48E-02	6.07E-01	1.73E+01	QP	
	2.13E+02	5.26E-01	3.28E-01	1.42E+01	KRG	
	1.33E+00	6.69E-01	3.39E-01	1.17E+01	KPLS	
	2.92E+01	4.17E-01	3.41E-01	1.68E+01	KPLSK	
	2.48E+00	4.26E-01	3.40E-01	1.67E+01	GPX	
	9.51E-02	-7.27E-02	6.04E-01	1.73E+01	RBF	
Scikit-learn	0.00E+00	2.34E-02	3.89E-01	1.85E+01	IDW	
	5.49E+01	8.63E-01	4.96E-01	1.91E+01	GPy	
	1.49E+00	8.90E-02	3.51E-01	1.80E+01	scikit	
Dakota	6.67E+01	5.68E-01	2.87E-01	1.07E+01	gaussian_process dakota	
	6.25E+01	2.37E-01	4.03E-01	1.43E+01	mars	
	6.11E+01	7.71E-02	3.48E-01	1.71E+01	neural_network	
	6.20E+01	9.72E-02	5.46E-01	1.51E+01	radial_basis	
	6.50E+01	1.09E-01	4.09E-01	1.69E+01	polynomial linear	
	5.79E+01	-1.47E-01	6.10E-01	1.61E+01	polynomial quadratic	
	2.33E+02	-4.94E-02	1.71E+01	6.95E+01	polynomial cubic	
total time (m): 15.17809267						

Figure A12. Surrogate model results for function 13

OASIS BASELINE (Function 14)						
R square	RAAE	RMAE	Dimensionality		Number of Training Samples	Number of predictions
2.31E-01	7.46E-01	2.72E+00	30		660	2000
Comments:						

Figure A13. Surrogate model results for function 14



## Appendix B.

Conglomerated scores for all functions for all error metrics

sheet name	LS	QP	KRG	KPLS	KPLSK	GPX	RBF	GPY	scikit	GP Dakota	mars	NN Dakota	RBF Dakota	Linear Dakota	Quad Dakota	Cubic Dakota	OASIS
1	-0.36	-0.08	-0.03	-0.03	-0.03	-0.04	-0.05	-0.05	-0.23	-0.23	-0.05	-0.39	-0.19	-0.43	-0.08	\	0.93
2	-0.49	-0.13	0.01	-0.02	0.01	0.01	-0.13	0.01	5.91	5.91	-0.04	-0.23	-0.20	-0.56	-0.17	-0.07	0.99
3	\	0.00	0.00	-0.87	0.00	-0.01	-0.01	\	-0.03	-0.03	-0.94	-1.08	\	\	0.00	0.00	1.00
4	0.41	-0.29	-0.19	-0.08	-0.19	-0.18	-0.29	-0.44	-0.30	-0.30	-0.11	-0.54	\	-0.55	-0.22	\	1.00
5	0.90	-0.20	-0.24	-0.86	-0.24	-0.24	-0.20	-0.64	0.27	-0.27	-0.56	-0.85	\	-0.95	-0.10	\	0.89
7	-0.34	0.02	#VALUE!	0.02	0.01	0.02	-0.96	0.01	-0.02	-0.02	-0.23	-0.36	-0.12	-0.63	0.00	\	0.97
8	-0.58	-0.99	#VALUE!	-0.50	-0.63	-0.63	-0.99	-1.00	-0.62	-0.62	-0.83	-0.89	-0.27	-0.82	-0.53	\	1.00
9	-0.89	0.00	#VALUE!	-0.84	0.00	-0.01	-0.01	-1.00	-0.05	-0.05	-0.88	-0.91	\	-0.96	0.00	0.00	1.00
10	-0.06	#VALUE!	0.00	-0.02	0.00	0.00	0.00	-0.06	0.00	0.00	-0.14	-0.04	-0.10	-0.08	0.00	-0.75	1.00
11	0.38	0.10	0.14	-0.02	0.14	0.04	-0.09	0.13	-0.07	-0.07	-0.10	0.31	-0.46	-0.44	-0.21	-0.38	0.79
12	\	\	\	\	\	\	\	\	\	\	\	\	\	\	\	\	0.02
13	-0.82	\	-0.22	-0.01	-0.38	-0.37	\	0.28	-0.87	-0.87	-0.65	-0.89	-0.86	-0.84	\	\	0.67
14	3.31	\	#VALUE!	3.31	3.31	3.31	\	-1.00	3.31	3.31	3.11	3.06	3.15	3.19	\	\	0.23
Adjusted R^2 Scores (- (OASIS - Model) / OASIS)																	

Figure B1. Relative R^2 scores

sheet name	LS	QP	KRG	KPLS	KPLSK	GPX	RBF	GPY	scikit	GP Dakota	mars	NN Dakota	RBF Dakota	Linear Dakota	Quad Dakota	Cubic Dakota	OASIS
1	1.51E+00	-5.30E-01	-2.34E-01	-1.81E-01	-2.34E-01	-2.58E-01	-3.62E-01	-4.21E+00	-1.04E+00	-1.26E-01	-3.40E-01	-1.46E+00	-9.58E-01	-1.71E+00	-4.61E-01	-2.79E+01	1.99E-01
2	-6.30E+00	-2.72E+00	5.50E-01	-3.64E-01	5.50E-01	4.99E-01	-2.70E+00	-1.22E+01	-1.73E+00	-3.54E-01	-1.05E+00	-3.25E+00	-3.70E+00	-6.57E+00	-3.14E+00	-1.85E+00	6.75E-02
3	-1.16E+15	-1.44E+02	-1.93E+11	-1.07E+15	1.98E+11	-1.01E+14	-8.32E+13	-1.15E+15	-1.94E+14	-7.43E+14	-1.10E+15	-2.24E+15	-1.16E+15	-1.16E+15	5.89E+05	-1.78E+07	7.00E-16
4	-6.52E+14	-5.49E+14	-4.35E+14	-2.86E+14	-4.35E+14	-4.29E+14	-5.47E+14	-1.23E+15	-5.45E+14	-5.47E+14	-3.34E+14	-7.23E+14	-1.22E+15	-7.49E+14	-4.66E+14	-7.18E+14	7.85E-16
5	-1.91E+00	-5.89E-01	-6.87E-01	-1.86E+00	-6.87E-01	-6.89E-01	-5.31E-01	-2.25E+00	-7.46E-01	-8.24E-01	-1.32E+00	-1.75E+00	-2.24E+00	-1.94E+00	-3.07E-01	-4.22E+02	2.63E-01
7	-1.72E+00	4.55E-01	#VALUE!	4.37E-01	2.62E-01	4.66E-01	-5.11E+00	3.85E-01	-2.41E-01	#VALUE!	-2.05E+00	-2.68E+00	-1.24E+00	-3.94E+00	2.16E-01	-2.55E+01	1.27E-01
8	-2.84E+14	-3.68E+14	#VALUE!	-2.60E+14	-2.95E+14	-2.94E+14	-3.70E+14	-3.75E+14	-2.90E+14	-1.72E+14	-3.39E+14	-3.49E+14	-5.54E+14	-3.39E+14	2.68E+14	-2.38E+15	2.14E-15
9	-6.50E+14	-0.54E+00	#VALUE!	-6.39E+14	-6.10E+11	-6.09E+13	-6.73E+13	-6.94E+14	-1.50E+14	-4.02E+14	-6.47E+14	-6.55E+14	-1.39E+101	-6.73E+14	-6.30E+05	-2.42E+08	1.18E-15
10	-1.77E+01	-1.55E+00	9.89E-01	8.60E+00	-8.18E-01	8.35E-01	-1.45E+00	-1.17E+02	-7.08E-01	-1.73E+01	-2.85E+01	-1.14E+01	-2.13E+01	-1.76E+01	-1.94E+00	-7.01E+01	9.30E-03
11	-3.27E-01	-1.31E-02	1.01E-01	1.98E-01	1.01E-01	1.92E-01	-8.74E-04	-1.78E+00	9.47E-02	8.80E-02	-1.23E-01	-2.83E-01	-5.30E-01	-4.87E-01	-3.42E-01	-5.46E-01	1.43E-01
12	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	4.83E-02
13	-2.11E+00	-3.78E+00	1.58E+00	-1.67E+00	-1.68E+00	-1.68E+00	-3.75E+00	-2.91E+00	-1.77E+00	-1.26E+00	-2.18E+00	-1.74E+00	-3.30E+00	-2.22E+00	-3.81E+00	-1.34E+02	1.27E-01
14	9.50E-01	1.00E+00	#VALUE!	9.55E-01	9.47E-01	9.47E-01	9.80E-01	7.78E-02	9.48E-01	1.00E+00	7.86E-01	7.99E-01	8.09E-01	8.17E-01	1.00E+00	1.00E+00	7.46E-01
Adjusted RAAE Scores ((OASIS - model) / OASIS)																	

Figure B2. Relative RAAE scores

sheet name	LS	QP	KRG	KPLS	KPLSK	GPX	RBF	GPp	scikit	GP Dakota	mars	NN Dakota	RBF Dakota	Linear Dakota	Quad Dakota	Cubic Dakota	OASIS
1	7.08E+00	-5.67E-01	-2.94E-01	-7.38E-01	-2.96E-01	-1.12E-01	-4.75E-01	-5.93E+00	-1.34E+00	-3.66E-01	-7.60E-01	-2.86E+00	-1.10E+00	-3.04E+00	-6.95E-01	-3.67E+01	9.69E-01
2	7.43E+00	-4.63E+00	2.87E-02	-2.52E+00	2.87E-02	-1.70E-01	-4.66E+00	1.06E+01	-4.90E+00	-2.94E+00	-1.39E+00	-6.27E+00	-5.56E+00	-8.25E+00	-4.79E+00	-2.22E+00	7.57E-01
3	5.17E+14	-3.95E-01	1.02E+14	-4.57E+14	-1.57E+11	-5.25E+13	-4.52E+13	3.19E+14	-1.03E+14	-4.87E+14	-6.47E+14	-8.51E+14	-5.54E+16	-8.43E+14	-3.26E+05	-1.56E+07	5.80E-15
4	-3.99E+14	-3.70E+14	-3.38E+14	-3.38E+14	-3.38E+14	-3.41E+14	-3.54E+14	-1.08E+15	-4.56E+14	-2.26E+14	-2.81E+14	-5.94E+14	-9.33E+15	-5.11E+14	-1.45E+14	-5.40E+18	9.80E-15
5	3.14E+00	-5.70E-01	-1.71E+00	-3.11E+00	-1.71E+00	-1.72E+00	-5.99E-01	4.02E+00	-1.89E+00	-1.78E+00	-2.46E+00	-2.92E+00	-1.70E+01	-3.57E+00	-6.50E-01	-7.43E+02	1.02E+00
7	2.86E+00	1.14E-01	#VALUE!	2.07E-01	-3.94E-01	1.29E-01	-5.73E+00	1.84E-01	-3.43E-01	#VALUE!	-2.73E+00	-3.89E+00	-3.80E+00	-4.86E+00	-1.05E+00	-3.06E+01	6.08E-01
8	-2.45E+14	-3.48E+14	#VALUE!	-2.20E+14	-2.51E+14	-2.60E+14	-2.48E+14	-3.28E+14	-2.63E+14	-1.54E+14	-3.44E+14	-3.54E+14	-2.24E+14	-2.94E+14	-2.61E+14	-2.38E+15	1.11E-14
9	-1.85E+15	-2.35E+01	#VALUE!	1.63E+15	-1.51E+12	-1.55E+14	-1.34E+14	1.19E+15	-1.46E+14	-1.27E+15	1.98E+15	-1.74E+15	-2.59E+104	-1.47E+15	-1.51E+06	-4.93E+08	2.14E-15
10	-6.04E-01	-1.29E+01	5.91E-01	-3.43E-01	-3.84E+00	-3.84E+00	-1.24E+01	2.18E+02	-4.20E+00	-9.09E-01	-7.28E+01	-6.72E+01	-7.81E+01	-7.94E+01	-1.82E+01	-1.14E+02	4.27E-01
11	3.16E+00	-2.17E+00	-6.44E-01	1.88E+00	-6.44E-01	-9.43E-01	-2.16E+00	4.83E+00	-2.61E+00	3.64E-01	4.00E-01	1.85E+00	-1.80E+00	-2.10E+00	-1.27E+00	-1.36E+00	2.04E+00
12	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	4.11E-01
13	-2.25E+00	-2.12E+00	-1.56E+00	-1.10E+00	-2.03E+00	-2.01E+00	-2.12E+00	-2.44E+00	-2.25E+00	-5.26E-01	-1.57E+00	-2.08E+00	-1.91E+00	-1.72E+00	-2.05E+00	-1.15E+01	5.54E+00
14	9.37E-01	1.05E+00	#VALUE!	9.40E-01	9.76E-01	9.76E-01	9.74E-01	1.68E+01	9.26E-01	1.03E+00	7.03E-01	6.24E-01	-3.04E-02	7.66E-01	1.00E+00	1.00E+00	2.72E+00
Adjusted RMAE Scores ((OASIS - model) / OASIS)																	

Figure B3. Relative RMAE scores

sheet name	LS	QP	KRG	KPLS	KPLSK	GPX	RBF	GPp	scikit	GP Dakota	mars	NN Dakota	RBF Dakota	Linear Dakota	Quad Dakota	Cubic Dakota	OASIS
1	5.05E-01	1.04E-01	2.46E-01	2.35E-01	2.46E-01	2.50E-01	2.71E-01	1.04E+00	4.05E-01	2.24E-01	2.57E-01	4.90E-01	1.89E-01	5.40E-01	2.91E-01	5.74E+00	1.99E-01
2	4.93E-01	2.51E-01	2.96E-02	2.96E-02	2.96E-02	1.65E-02	2.50E-01	8.89E-01	1.84E-01	9.14E-02	1.58E-01	2.86E-01	1.17E-01	5.11E-01	2.79E-01	1.92E-01	6.75E-02
3	8.15E-01	1.02E-13	1.35E-04	7.52E-01	2.08E-04	7.10E-02	5.83E-02	8.05E-01	1.36E-01	5.20E-01	7.70E-01	1.57E+00	8.12E-01	8.15E-01	4.08E-10	1.24E+06	7.00E-16
4	5.12E-01	4.31E-01	3.41E-01	2.26E-01	3.41E-01	3.37E-01	4.29E-01	8.66E-01	4.28E-01	2.72E-01	2.62E-01	5.68E-01	9.54E-01	5.88E-01	3.64E-01	5.57E+01	7.85E-16
5	7.63E-01	4.17E-01	4.43E-01	7.51E-01	4.43E-01	4.44E-01	4.18E-01	8.53E-01	4.59E-01	4.79E-01	6.10E-01	7.22E-01	8.50E-01	7.72E-01	3.83E-01	1.11E+02	2.63E-01
7	4.70E-01	4.89E-02		7.11E-02	9.33E-02	6.75E-02	7.73E-01	8.04E-02	1.57E-01			3.89E-01	4.66E-01	2.84E-01	6.24E-01	9.91E-02	3.38E+00
8	6.07E-01	7.88E-01		5.57E-01	6.31E-01	6.29E-01	7.91E-01	8.02E-01	6.21E-01	3.78E-01	7.24E-01	7.46E-01	4.15E-01	7.25E-01	5.73E-01	5.12E+00	2.14E-15
9	7.54E-01	1.22E-14		7.34E-01	7.08E-04	7.04E-02	7.80E-02	8.05E-01	1.74E-01	4.66E-01	7.50E-01	7.60E-01	2.77E+06	7.80E-01	7.31E-10	2.81E-07	1.16E-15
10	1.74E-01	2.37E-02	9.98E-01	8.93E-02	1.69E-02	1.70E-02	2.28E-02	1.10E+00	1.59E-02	1.24E-01	2.79E-01	1.11E-01	2.07E-01	1.73E-01	2.73E-02	6.62E-01	9.90E-01
11	4.56E-01	3.48E-01	2.06E-01	2.96E-01	2.06E-01	2.78E-01	3.44E-01	9.15E-01	3.11E-01	3.13E-01	3.85E-01	4.41E-01	5.25E-01	5.11E-01	4.26E-01	5.31E-01	3.43E-01
12																	4.83E-02
13	3.95E-01	6.07E-01	1.26E-01	3.39E-01	2.41E-01	3.40E-01	6.04E-01	4.96E-01	3.51E-01	2.87E-01	4.03E-01	3.48E-01	5.46E-01	4.09E-01	6.10E-01	1.71E+01	1.27E-01
14	3.73E-01	1.40E-14		1.35E-02	3.98E-02	1.92E-02	1.47E-02	8.09E-01	3.91E-02	3.83E-10	3.75E-01	1.95E-01	1.43E-01	1.37E-01	3.85E-10	3.83E-10	7.48E-01
RAW RAE Scores																	

Figure B4. Raw RAE scores

sheet name	LS	QP	KRG	KPLS	KPLSK	GPX	RBF	GPp	scikit	GP Dakota	mars	NN Dakota	RBF Dakota	Linear Dakota	Quad Dakota	Cubic Dakota	OASIS
1	2.99E+00	1.52E+00	1.26E+00	1.68E+00	1.26E+00	1.27E+00	1.43E+00	6.72E+00	2.27E+00	1.32E+00	1.71E+00	3.74E+00	2.23E+00	3.92E+00	1.64E+00	3.89E+01	9.69E-01
2	6.40E+00	4.26E+00	7.25E-01	2.67E+00	7.33E-01	9.62E-01	4.28E+00	8.80E+00	4.47E+00	2.99E+00	1.81E+00	5.50E+00	4.97E+00	7.00E+00	4.39E+00	2.44E+00	7.57E-01
3	3.00E+00	2.35E-13	5.93E-04	2.88E+00	5.11E-04	3.07E-01	2.62E-01	3.07E+00	5.95E-01	2.83E+00	3.75E+00	4.94E+00	3.27E+02	4.89E+00	1.88E-09	5.07E-08	5.80E-15
4	3.31E+00	2.15E+00	1.86E+00	1.15E+00	1.96E+00	1.90E+00	2.09E+00	6.28E+00	2.44E+00	1.31E+00	1.51E+00	3.44E+00	5.41E+01	2.96E+00	2.00E+00	3.13E+02	5.80E-15
5	4.24E+00	2.02E+00	2.77E+00	4.41E+00	2.77E+00	2.79E+00	2.04E+00	5.14E+00	2.94E+00	2.85E+00	3.34E+00	4.01E+00	1.85E+01	4.68E+00	1.89E+00	7.62E+02	1.02E+00
7	2.35E+00	3.99E-01		4.83E-01	8.48E-01	5.30E-01	4.09E+00	4.57E-01	8.17E-01			2.27E+00	2.98E+00	2.52E+00	3.56E+00	1.92E+01	6.08E-01
8	2.72E+00	3.86E+00		2.44E+00	2.79E+00	2.89E+00	3.88E+00	3.73E+00	2.92E+00	1.71E+00	3.82E+00	3.93E+00	2.49E+00	3.27E+00	2.90E+00	2.86E+01	1.11E-14
9	3.96E+00	5.21E-14		3.47E+00	1.23E-03	3.32E-01	3.30E-01	3.75E+00	7.41E-01	2.72E+00	4.23E+00	3.73E+00	5.54E+09	3.57E+00	3.22E-09	1.91E-06	2.14E-15
10	2.88E+00	5.94E-01	3.94E-04	1.53E+00	2.07E-01	2.07E-01	5.71E-01	9.27E+00	2.22E-01	2.47E+00	3.15E+00	2.91E+00	3.38E+00	3.43E+00	8.12E-01	4.91E+00	4.27E-02
11	8.49E+00	6.48E+00	1.38E+00	5.89E+00	3.36E+00	3.37E+00	6.43E+00	1.19E+02	7.37E+00	2.79E+00	2.86E+00	5.83E+00	5.72E+00	6.33E+00	4.63E+00	4.82E+00	2.04E+00
12																	4.11E+01
13	1.80E+01	1.73E+01	1.42E+01	1.17E+01	1.68E+01	1.67E+01	1.73E+01	1.91E+01	1.80E+01	1.07E+01	1.43E+01	1.71E+01	1.51E+01	1.69E+01	1.61E+01	8.95E+01	5.54E+00
14	1.71E-01	7.50E-14		1.64E-01	2.01E-01	2.01E-01	7.02E-02	3.17E+00	2.01E-01	1.35E-04	6.07E-01	1.02E+00	2.80E+00	5.36E-01	1.35E-09	1.35E-09	2.72E+00
Raw RMAE Scores																	

Figure B5. Raw RMAE scores

## Appendix C

### Pre-Processed Accuracy Metric Graphs

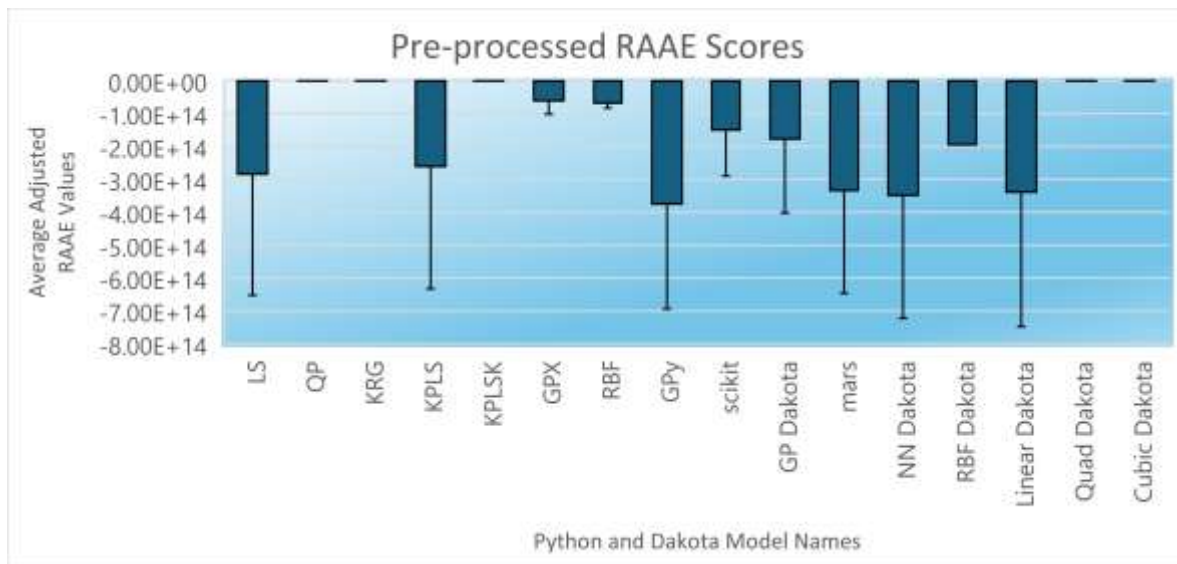


Figure C1. Pre-processed RAAE scores of surrogate tests against OASIS.AI benchmark

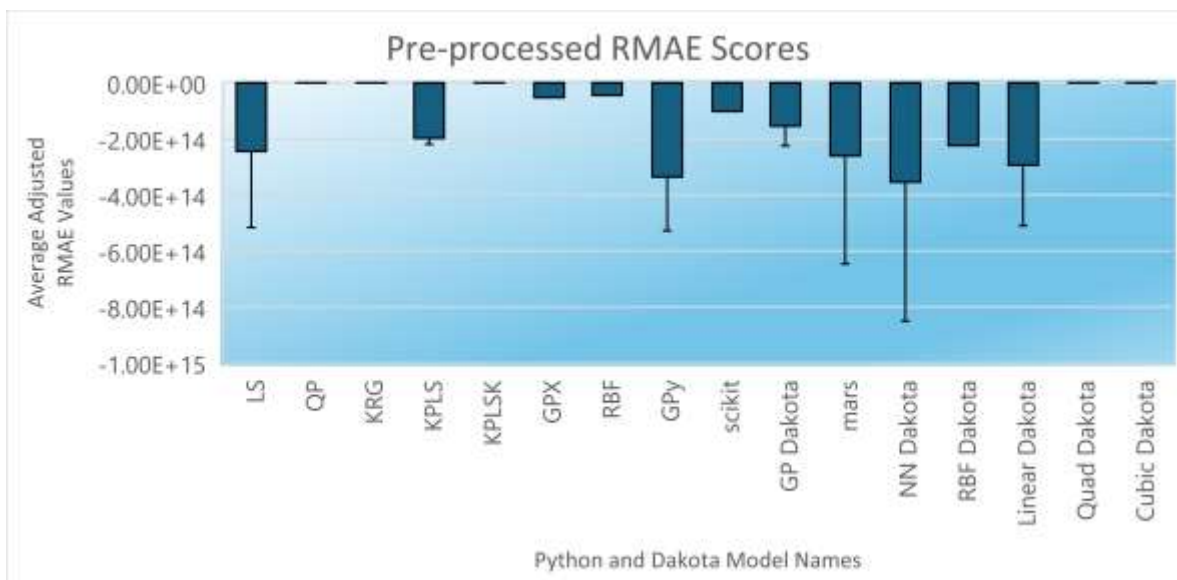


Figure C2. Pre-processed RMAE scores of surrogate tests against the OASIS.AI benchmark