

Selection of Surrogate Modeling Techniques for Surface Approximation and Surrogate-Based Optimization

Bianca Williams and Selen Cremaschi*

Department of Chemical Engineering, Auburn University, Auburn, AL

*selen-cremaschi@auburn.edu

Highlights

- Eight surrogate modeling techniques were assessed over several datasets.
- MARS and GP models had the best performance overall for approximating a surface.
- RF, SVR, and GP gave the most robust performance for surrogate-based optimization.
- The results suggest surrogate model performance depends on data characteristics.

Abstract

Surrogate models are used to map input data to output data when the actual relationship between the two is unknown or computationally expensive to evaluate for several applications, including surface approximation and surrogate-based optimization. This work evaluates the performance of eight surrogate modeling techniques for those two applications over a set of generated datasets with known characteristics. With this work, we aim to provide general rules for selecting an appropriate surrogate model form based solely on the characteristics of the data being modeled. The computational experiments revealed that there is a dependence of the surrogate modeling performance on the data characteristics. However, in general, multivariate adaptive regression spline models and Gaussian process regression yielded the most accurate predictions for approximating a surface. Random forests, support vector machine regression, and Gaussian

process regression models most reliably identified the optimum locations and values when used for surrogate-based optimization.

Keywords: surrogate model, surface approximation, surrogate-based optimization, random forests, multivariate adaptive regression splines, Gaussian process regression

1. Introduction

Surrogate models, also known as response surfaces, black-box models, metamodels, or emulators, are simplified approximations of more complex, higher-order models (Wang et al., 2014). These models are used to map input data to output data when the actual relationship between the two is unknown or when the relationship is computationally expensive to evaluate (Han and Zhang, 2012). Surrogate models can also be constructed for use in surrogate-based optimization when a closed analytical form of the connection between the inputs and outputs is not available or is not conducive for use in conventional gradient-based optimization methods. Surrogate modeling techniques are of particular interest where high-fidelity, thus expensive, simulations are used (Han and Zhang, 2012), for example, in computational fluid dynamics (CFD) or computational structural dynamics (CSD). Surrogates are also of interest when the fundamental relationship between design variables and output variables is not well understood, such as in the design of cell or tissue manufacturing processes (Du et al., 2016; Sokolov et al., 2017; Williams et al., 2020).

Surrogate modeling techniques have been receiving increasing attention in a wide range of applications, for example, in the optimization of process design, scheduling, and control (Burnak et al., 2019). They have successfully been used for both regression and classification tasks. Surrogate models have been used in several recent applications in process systems engineering and the manufacturing industry, including for optimization of a sulfur recovery unit (Rahman et

al., 2019), fault detection (Quiroz et al., 2018; Zhang et al., 2018), and surrogate-based optimization of energy consumption in carbon fiber production line (Golkarnarenji et al., 2018).

Construction of a surrogate model is comprised of three steps: (1) selection of the sample points, (2) optimization or "training" of the model parameters, and (3) evaluation of the accuracy of the surrogate model (Wang et al., 2014). Although several machine learning and regression techniques have been developed for surrogate model construction, there has been little work on how to select the appropriate model for a particular application for either surface approximation or surrogate-based optimization. Surface approximation refers to the application of using a surrogate model to mimic the overall behavior or response of an underlying model. In surrogate-based optimization, a surrogate model can be constructed to represent the objective function or any constraints that may be computationally expensive to evaluate or are unavailable in analytical form. The constructed surrogate can be used as a closed functional form in traditional gradient-based optimization methods.

Numerous studies have been conducted to compare the performance of surrogate modeling techniques (Bhosekar and Ierapetritou, 2018; Davis et al., 2017). Previous work on this topic has shown that the performance for approximation is dependent on data characteristics such as the input dimension and the underlying function shape (Davis et al., 2017; Williams and Cremaschi, 2019). The majority of these only compare a few models on a limited number of functions or for specific applications (Ju et al., 2016; Luo and Lu, 2014; Villa-Vialaneix et al., 2012). Recent developments in automatic selection of surrogate models primarily involve training multiple surrogates and selecting the best surrogate based on some criteria using a trial-and-error approach (Ben Salem and Tomaso, 2018; Mehmani et al., 2018). However, the trial-and-error selection approach has the potential to become computationally expensive. With the large number of

techniques that have been developed for constructing surrogate models, there is a need for systematic methods to select an appropriate model form for use in surface approximation and surrogate-based optimization. Current common practices for selecting which surrogate model form is appropriate rely on process-specific expertise.

Selection of an appropriate number of sample points and sampling method to generate those samples is a critical step in the construction of a surrogate model. In general, a higher number of sample points offers more information about the underlying model being approximated, although with a higher computational expense. For low-order functions, after reaching a certain sample size, increasing the number of sample points does not contribute much to the approximation accuracy (Wang and Shan, 2007). Previous studies have investigated the effects of sample size and sampling method on some of the surrogate modeling techniques being studied specifically, including Gaussian process regression (Afzal et al., 2017; Burnaev and Zaytsev, 2015; Iooss et al., 2010) and radial basis function networks (Afzal et al., 2017), as well as on surrogate modeling accuracy in general (Davis et al., 2017). The results of these studies indicate that the accuracy of a surrogate model is dependent upon the number and distribution of samples used in its construction.

The objective of this work is to comprehensively investigate and compare the performance of several different surrogate modeling techniques for both approximating functional relationships and surrogate-based optimization, and to link that performance to the characteristics of the data involved in the application. The results of this analysis are used to develop general "rules of thumb" for selecting an appropriate surrogate modeling technique based on the characteristics of the data being modeled and the desired application. Data sets for training surrogate models are generated from a suite of optimization test functions with different features, such as function shape and

number of inputs. The specific data characteristics being investigated in this study are the shape of the underlying function being modeled, the number of input dimensions, the sampling method used to select sample points to be used in the model training, and the number of sample points. The surrogate modeling techniques considered include Automated Learning of Algebraic Models using Optimization (ALAMO), Artificial Neural Networks (ANN), Extreme Learning Machines (ELM), Gaussian Process Regression (GP), Multivariate Adaptive Regression Splines (MARS), Radial Basis Function Networks (RBFN), Random Forests (RF), and Support Vector Machine Regression (SVR). The following sections contain descriptions of the surrogate modeling techniques used, the sampling methods used to select the training data sets, and the test function sets. Then, the computational experiments and the results are presented, followed by conclusions and future directions.

2. Surrogate Modeling Techniques

2.1 Automated Learning of Algebraic Models using Optimization (ALAMO)

Automated learning of algebraic models (ALAMO) uses a linear summation of nonlinear transformations of the input data to predict output values. Possible nonlinear transformations include polynomial, exponential, logarithmic, ratio, and trigonometric functions (Cozad et al., 2014). The nonlinear transformations allowed for ALAMO models trained for this work were sine, cosine, exponential, logarithmic, polynomial functions. Given a dataset, the approach begins by building a low-complexity, linear model composed of explicit nonlinear transformations of the input variables. Then, the method iteratively refines the model by solving an optimization problem at each iteration to minimize (or maximize) a user-designated error metric. It should be noted that the adaptive sampling scheme of ALAMO is not used in this study. ALAMO is one of the few surrogate modeling techniques developed directly by the chemical engineering community.

2.2 Artificial Neural Networks

Artificial neural networks attempt to mimic the behavior of neurons in the brain. The models consist of an input and an output layer that are connected by a number of hidden layers in between. The artificial neurons have weights and biases that create a network between the layers, with the activation function in the hidden layer determining whether or not a neuron will "fire" and produce a signal (Haykin, 2009). Training of a neural network refers to the process that identifies the values of the weights and biases. Three different types of artificial neural networks are considered here, all with a single hidden layer: a feed-forward artificial neural network with a hyperbolic tangent activation function (ANN), an extreme learning machine (ELM), and a radial basis function network (RBFN). In an ELM, the weights between the input layer and hidden layer are randomly assigned, and the weights between the hidden layer and the output layer are fit using linear regression or other regression techniques (Huang et al., 2006). The activation function used in both the ANN and ELM models is a hyperbolic tangent function. An RBFN is a neural network with a radial basis function as the activation function in the hidden layer (Gomm and Yu, 2000). First, the network calculates the Euclidean distance between the input weights and input values. Then it passes those distances through the Gaussian radial basis activation function. The form of the radial basis function is shown in Eqs. (1) and (2),

$$r = \|x - x'\| \quad (1)$$

$$\varphi(r) = e^{-(\varepsilon r)^2} \quad (2)$$

where the Euclidean distance, r , between points x and x' , is used to calculate the radial basis function, $\varphi(r)$, with the shape tuning parameter ε .

2.3 Gaussian Process Regression (GP)

Gaussian process regression (GP) is a method of interpolation for which the interpolated values are modeled by a Gaussian process governed by prior covariances. Gaussian process regression uses a linear combination of inputs to predict output values. It uses a kernel function as a measure of similarity between points to predict the value for an unseen point from the training data (Rasmussen and Williams, 2005). The radial basis function is used as the kernel function for all GP models trained for this work.

2.4 Multivariate Adaptive Regression Splines (MARS)

Multivariate adaptive regression spline (MARS) models are made up of a linear summation of basis functions. The three types of possible basis functions are a constant, a hinge function (or "spline"), or a product of two or more hinge functions. The training of a MARS model starts with an initial model that is a basis function equal to the mean of the data outputs. On the first pass, the model overfits to the data, adding basis functions to reduce the sum of the squared errors (SSE) between the given and predicted outputs. Then, a backward, pruning pass is performed to remove terms that have little effect on the SSE until the best model is identified based on cross validation criteria (Friedman, 1991).

2.5 Random Forests (RF)

Random forests are machine learning models that make output predictions by combining outcomes from a sequence of regression decision trees, called forests. Each tree is constructed independently and depends on a random vector sampled from the input data, with all the trees in the forest having the same distribution. The predictions from the forests are averaged using bootstrap aggregation and random feature selection (Breiman, 2001). The value that is output for

a tree for given inputs is the value of the final leaf node reached, and the output value for the entire RF model is the average value of the outputs for every decision tree in the forest.

2.6 Support Vector Machine Regression (SVR)

Support vector machine regression transforms input data into m-dimensional space and attempt to construct a set of hyperplanes so that the distance from it to the nearest data point on each side of the plane is maximized using kernel functions (Drucker et al., 2002). The kernel functions transform the data into a higher dimensional feature space to make it possible to perform the linear separation.

3. Computational Experiments

3.1 Test Functions

The test functions used to generate data for constructing the surrogate models are from the Virtual Library of Simulation Experiments optimization test suite (Surjanovic and Bingham, 2013). These test functions are benchmarking optimization problems presented in the form of analytic functions (Hussain et al., 2017). Functions with two, four, six, eight, ten, fifteen, and twenty input dimensions were used in evaluations, resulting in a total of 127 test functions. The functions are divided by their shapes, which include the categories: *multi-local minima* with 39 functions, *bowl-shaped* with 41 functions, *plate-shaped* with 11 functions, *valley-shaped* with 16 functions, and *other-shaped* with 20 functions that do not fit into the other four categories. Example functions from each shape category are provided in Fig. 1.

The shape categories are defined by multiple characteristics of the test functions, including modality, basins, and valleys, which describe the resulting surface. Modality refers to the number of peaks on the surface. Multimodal functions have many local solutions but one global one, making the global solution difficult to identify as algorithms may become trapped in

local solutions. A basin is a relatively steep decline surrounding a large area. These basin regions can severely obstruct optimization algorithms due to a lack of information to direct the search towards the optimum (Jamil and Yang, 2013). A valley occurs when a narrow area of little change is surrounded by regions of steep descent. The progress of an optimization algorithm may be hampered significantly on the floor of the valley (Hussain et al., 2017).

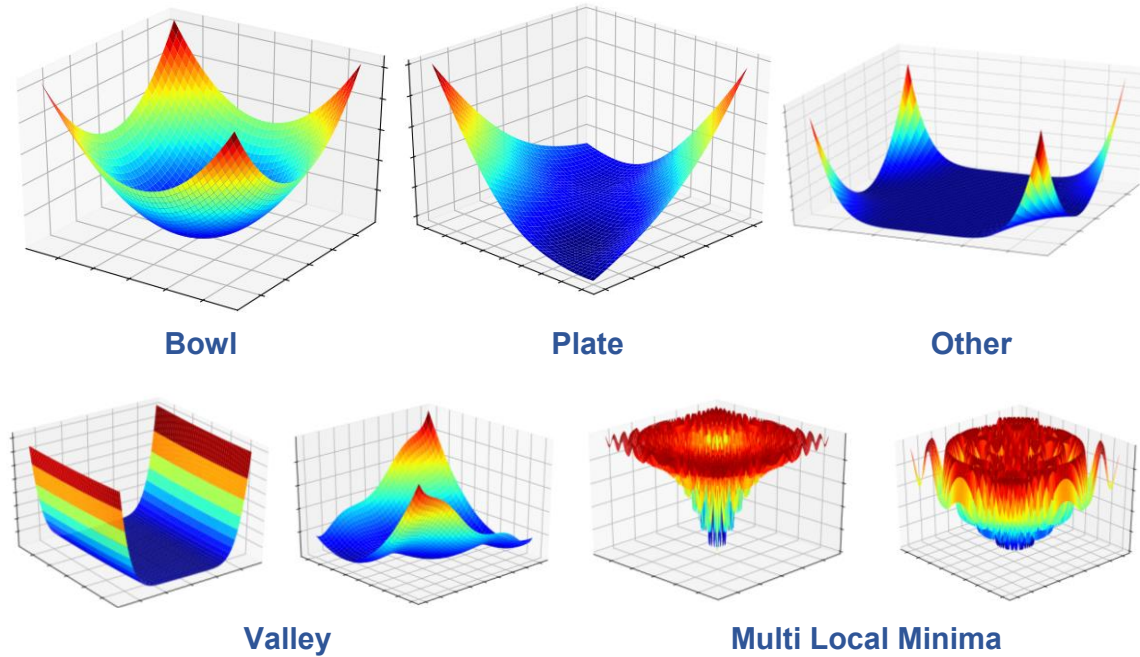


Figure 1. Shape categories for test functions.

The *bowl*-shaped functions are unimodal, convex surfaces that can represent applications where changes in inputs produce smooth, regular changes in output values. The *multi-local minima* functions are multimodal and nonconvex and more representative of real data applications with significant noise in the output. The *plate*-shaped functions contain large basin regions. The *plate*-shape function may be representative of processes where several values of the process inputs or a large section of the design space give a constant value for outputs, creating difficulties with optimization searches. *Valley*-shaped functions have valleys, which may be applicable to processes where small changes in input values produce very large variations in output values. Both the *plate*-

and *valley*-shaped categories contain unimodal and multimodal functions. The *other*-shaped functions contain combinations of the characteristics of the other categories and non-smooth functional behavior, which could encompass several processes where the shape of the output surface is not well-known.

3.2 Surrogate Model Comparison

3.2.1 Surrogate Model Construction

For evaluating the performances of surrogate modeling techniques, input-output pairs were generated from each test function using three different sampling methods at seven different sample sizes (50, 100, 400, 800, 1200, and 1600 samples). The sample sizes were chosen in order to give a range of values for the ratio of sample size to input dimension for each input dimension being studied. In general, a sample size to input dimension ratio of 10 is considered an adequate number samples for most regression techniques (Harrell et al., 1984). Any ratio smaller than 10 can be considered to be a small sample size, with large sample sizes being any ratio of sample size to input dimension larger than 10. Surrogate models were trained using these pairs with each of the surrogate modeling techniques for each generated dataset. This process resulted in a total of 18,984 trained models. Each of the techniques has unique hyperparameters that were optimized in training the models for each dataset to construct the best possible surrogate without overfitting the model. For the MARS models, the number of hinge functions that could be multiplied together was limited to two to avoid overfitting with higher-order hinge functions. The numbers of ANN, ELM, and RBFN nodes, as well as the number of trees in the RF models, were increased until the root mean squared error of a validation dataset stopped improving. For these models, the validation error was estimated using ten-fold cross-validation on the training set. The number of nodes (or trees) was

increased until the average value of the last five validation errors either began to increase or changed by less than 1%.

All of the surrogate modeling techniques except ALAMO and RBFN were implemented in Python with the Sci-Kit Learn library version 0.32.2 (Pedregosa et al., 2011). RBFN models were implemented with MATLAB 2017b, and ALAMO has its own software for model construction (Cozad et al., 2014). All of the training options except for the ones discussed were set to the default values indicated by the implementation package. The specific implementation package used for each technique is listed in the supplementary materials.

The three sampling methods used were Halton Sequence Sampling (Halton), Latin Hypercube Sampling (LHS), and Sobol Sequence Sampling (Sobol). LHS partitions the domain of each input variable into N subsets to be sampled from, where N is the number of sampling points (Mckay, 1992). Both Halton and Sobol sequence sampling are quasi-random, low discrepancy sequences that attempt to distribute the sampling points uniformly across the sample space (Halton and Smith, 1964; Joe and Kuo, 2008). These sampling methods were chosen because they have been shown to sample input space uniformly for functions up to ten dimensions (Diwekar, 2003; Garud et al., 2017).

3.2.2 Evaluation of Surface Approximation and Surrogate-Based Optimization Performance

After the surrogate models were trained for each dataset, sample size, and sampling method, a densely sampled set of 100,000 input-output pairs were generated as test dataset for assessing the accuracy of the models. Because there was no significant difference between the samples or results obtained from any of the sampling methods at this large size, only results for the dense set produced using Sobol sequence sampling are presented here. The root mean squared error, adjusted R^2 value, and the maximum percent error were calculated for each dataset-surrogate

model combination based on the difference between the outputs of the given function and the outputs predicted by the surrogate model.

The global minimum of each test function was estimated using the trained surrogate models. The mathematical programs for estimating the minima were constructed in Pyomo (version 5.6) (Hart et al., 2017; Hart et al., 2011), a Python-based optimization language. The estimated minimum location and value are compared to the actual global minimum and value of each function for accuracy to provide some insight into the effectiveness of each surrogate modeling technique for surrogate-based optimization. Computations were carried out on the Auburn University Hopper HPC Cluster (Lenovo System X HPC Cluster) using Intel E5-2650 V3, 2.3 GHz 20 core processors and implemented in Python 3.5 and MATLAB 2017b (for RBFN surrogate models).

3.3 Performance Metrics

3.3.1 Surface Approximation Performance Metrics

Two performance metrics were used for evaluating the surface approximation ability of the surrogate models: normalized root mean square error (nRMSE) and adjusted- R^2 . The adjusted- R^2 (Miles, 2014) takes into account both the surrogate model accuracy and the size, or complexity, of the model. Balancing the complexity of the model with the sample size is essential in ensuring that the model is not overfit, as overfit models do not generalize well to new conditions. However, adjusted- R^2 can unfairly penalize some of the surrogate models that are larger by nature of their structure, for example, Random Forests, which need to grow larger because of their decision tree framework. The nRMSE metric was chosen to assess how well the surrogates approximated the test function without penalizing them for their size. The formula for (nRMSE) is given in Eq. (3).

The nRMSE value for each dataset-surrogate model combination is normalized by the range of output values for easier comparison across datasets with a variety of ranges for output values.

$$nRMSE = \sqrt{\frac{\sum_{n=1}^N (\hat{y}_n - y_n)^2}{N}} / (y_{max} - y_{min}) \quad (3)$$

In Eq. (3), y_n is the output for point n for a dataset, $\hat{y}_n$ is the output predicted by a surrogate model for point n , N is the total number of sample points in the dataset, and y_{max} and y_{min} are the maximum and minimum output values in a dataset, respectively.

The formula for calculating adjusted- R^2 ($\hat{R}^2$) is shown in Eq. (4).

$$\hat{R}^2 = 1 - (1 - R^2) \left[\frac{N - 1}{N - (k + 1)} \right] \quad (4)$$

In Eq. (4), R^2 is the R-squared regression coefficient, N is the number of data points in the training set, and k is the number of model parameters (or hyperparameters). R^2 values typically fall between zero and one, with an R^2 of one indicating a perfect fit. However, with the adjustment for model size, adjusted- R^2 values can become negative. The number of model hyperparameters, k , was estimated as the number of nodes in the trained ANN, RBFN, and ELM models. For MARS models, k was estimated as the total number of hinge functions. The k for the ALAMO models was estimated as the number of nonlinear transformation terms in the final model. The k for SVR models was estimated as the number of support vectors in the trained model. For GP models, k was estimated as the number of input dimensions, which corresponds to the number of hyperparameters that are fit for the length scale used in the radial basis function (the kernel function used in the GP models). For RF models, k was estimated as the average number of decision threshold values per tree in the forest.

The nRMSE and adjusted- R^2 metrics were calculated using the densely sampled 100,000 point test sets generated using Sobol Sequence sampling. One-way analysis of variance (ANOVA)

was applied to determine which dataset characteristics had a statistically significant effect on the surrogate model performance metrics at a 95% confidence level.

3.3.2 Surrogate-Based Optimization Performance Metrics

We define D_{opt} as the Mahalanobis distance, D_M , (De Maesschalck et al., 2000) between the location of the global minimum of a test function, x_{opt} , and the location estimated using a trained surrogate model, $\hat{x}_{opt}$. This value is normalized by the maximum Mahalanobis distance between any two points (x_i, x_j) in the dataset (Eq. 5),

$$D_{opt} = \frac{D_M(x_{opt}, \hat{x}_{opt})}{\max_{i,j} D_M(x_i, x_j)} \quad (5)$$

where x_i and x_j are points in the domain space of the dataset.

We define G_{opt} , Eq. (6), as the normalized gap between the global minimum value and the estimated one. This value is normalized by the range of output values in the dataset.

$$G_{opt} = \frac{y_{opt} - \hat{y}_{opt}}{y_{max} - y_{min}} \quad (6)$$

In Eq. (6), y_{opt} is the actual global minimum value, $\hat{y}_{opt}$ is the one calculated by the surrogate model, and y_{max} and y_{min} are the maximum and minimum output values in a dataset, respectively.

4. Results and Discussion

4.1 Effect of Sampling Method and Sample Size

The surface approximation performance for MARS models is shown in Fig. 2a as a function of both sample size and sampling method. The average value of the performance metrics for all 127 test functions is given as a function of the sample size for models trained with datasets generated from each of the three sample methods tested. Similar to previous studies, the results here show a general trend of improving performance with the addition of more sample points for

training the surrogate model. Surface approximation performance for all the surrogate modeling techniques showed a comparable behavior to that of the MARS models. Plots of performance as a function of sample size and sampling method for all surrogate modeling techniques and performance metrics are available in the supplementary materials. The 90% confidence interval error bars for the sampling methods have some overlap at all of the sample sizes. From this result, there does not appear to be any significant difference in the surrogate modeling performance among the three sampling methods investigated. ANOVA analysis did not indicate any statistically significant difference in surface approximation performance for any of the surrogate model techniques with changing sampling methods ($p > 0.05$). The selection of the space-filling sampling method does not appear to have any effect on the approximation performance.

For surrogate-based optimization, only RF and RBFN models showed any statistically significant differences for the surrogate model performance among the three sampling methods. The other six surrogate modeling techniques' performance was not significantly affected by choice of the sampling method. The average D_{opt} value for RF and RBFN models as a function of the training set sample size is shown in Fig. 2b and Fig. 2c for Sobol, Halton, and LHS sampling. For RF models (Fig. 2b), the D_{opt} values for models trained with Sobol sequence sampling data tend to be in general lower than those of models trained with data generated from the other two sampling methods, meaning that the optimum locations predicted by RF models trained with Sobol samples are on average closer than those given by other sampling methods. ANOVA analysis further confirmed that the models trained using Sobol sequence samples had statistically significantly lower values for D_{opt} at each of the sample sizes investigated. The models trained using Sobol sequence samples also had statistically lower values of D_{opt} for RBFN models (Fig. 2c) at a sample size of 50 ($p = 0.002$). These results indicate that while sampling method does not affect surface

approximation performance, for some surrogate modeling techniques, the choice of sampling method can have a substantial impact on the performance for the model for surrogate-based optimization, especially at lower sample sizes

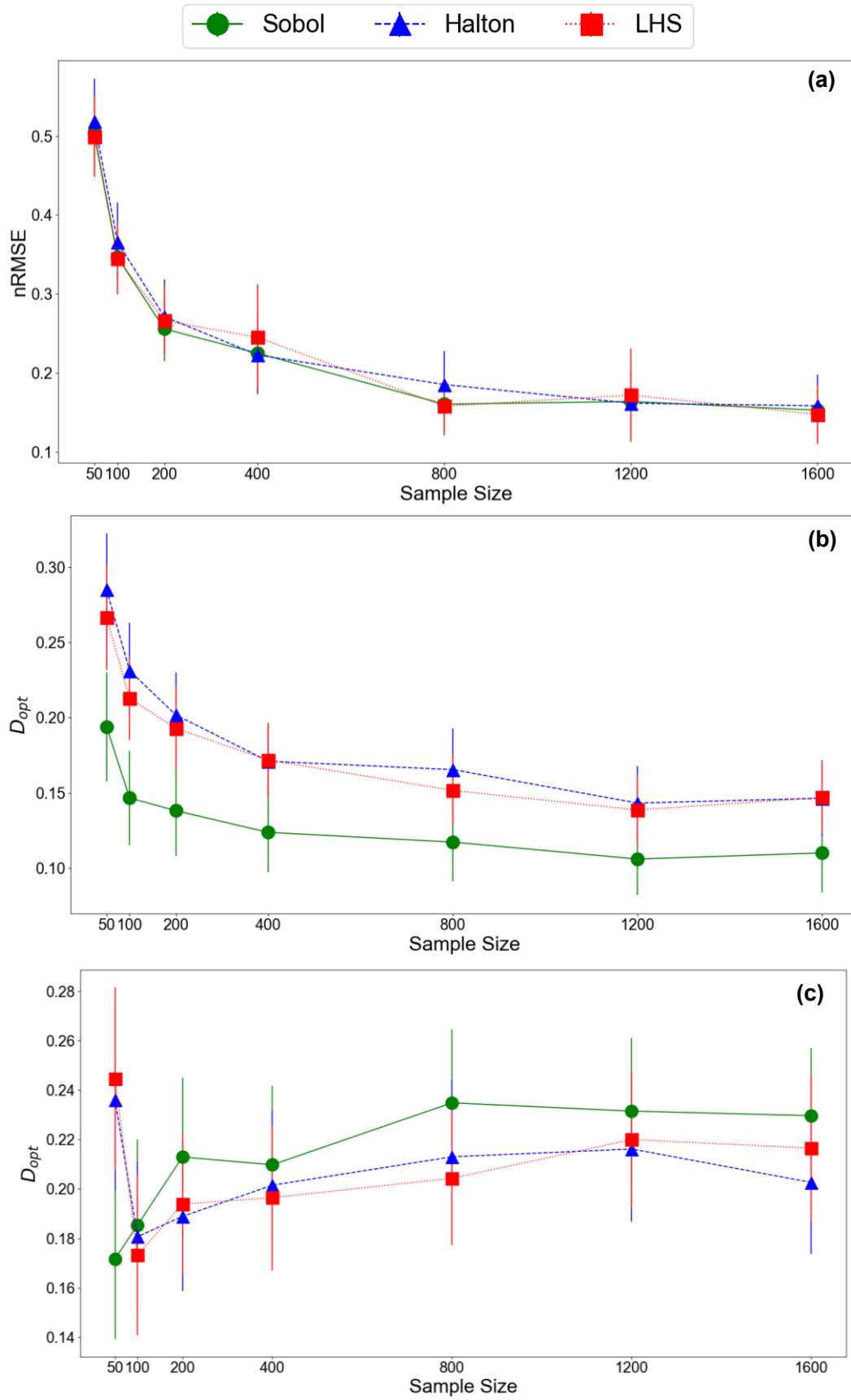


Figure 2. (a) MARS performance for different sampling methods as a function of sample size for average nRMSE on all 127 test functions. (b) Average D_{opt} vs. sample size for RF models. (c) Average D_{opt} vs. sample size for RBFN models. Error bars represent 90% confidence intervals on the averages.

4.2 Comparison of surrogate modeling technique performance for surface approximation

There was no significant difference in the surface approximation performance of the surrogate models trained using the sample points generated using Sobol and Halton sequences and LHS. Therefore, results presented in this section only include surrogate models trained with datasets generated via Sobol sequence sampling. Results are presented in this section for three selected sample sizes. However, results for all sample sizes are available in the supplementary material. The surface approximation performance metric results are presented in violin plot format. The shape of each violin represents the probability density distribution of the data values. The top and bottom of the violin represent minimum and maximum values, with the black bar within each violin representing the interquartile range of the values. Median values are indicated on each violin by a white circle.

4.2.1 Selection of Surrogate Modeling Technique for Surface Approximation by Adjusted- R^2

Results obtained based on the adjusted- R^2 are summarized in Fig. 3. The adjusted- R^2 was used to take into account the model size and complexity in addition to its accuracy (Miles, 2014). This metric can be used to select a “best” model to use for approximation while controlling for overfitting by selecting the technique that provides the highest value of adjusted- R^2 . Adjusted- R^2 values were calculated for all the trained surrogate models for each dataset. For each dataset category (either input dimension or shape), the number of times each surrogate modeling technique was selected as best (had the highest adjusted- R^2) was tabulated for the datasets in that category. These tabulated values were divided by the total number of datasets in the category to calculate the fraction of datasets for which each surrogate modeling technique was selected as best performing. The number of datasets included in each category is given below the x -axis for Fig. 3.

Figure 3 shows which surrogate modeling techniques are selected most frequently when the datasets are grouped by their input dimension and by their function shape. Although at the smallest sample size tested, RBFN models give the highest adjusted- R^2 more frequently at the higher input dimensions, ALAMO provides the highest adjusted- R^2 , and thus the closest approximation, the highest percentage of the time at low input dimension. The superior approximation performance of RBFN models is not observed at higher sample sizes when considering selecting the “best” model by the adjusted- R^2 . In general, ALAMO provides the most robust performance for yielding the highest adjusted- R^2 . However, as the dimension and sample size increase, GP and MARS models begin to perform as well or better than ALAMO models.

At the largest sample size (Fig. 3b), GP and MARS models demonstrate opposite trends with increasing input dimension until eight inputs. The selection frequency of GP models as having the highest adjusted- R^2 (Fig. 5b) deteriorates while that of MARS improves. At input dimensions higher than eight, both GP and MARS models have decreasing selection frequencies. GP models use interpolation between the given training points to estimate outputs at new input conditions. As the number of input dimensions increases, GP models require a higher number of training points to more accurately capture a surface’s behavior in a region for interpolating to new conditions. This may explain why MARS models begin to outperform GP models at higher dimensions, as the hinge functions of the MARS models are not dependent on interpolation.

Figs. 3c and 3d show which surrogate modeling techniques are selected most frequently when the datasets are grouped by function shape. When the datasets are grouped by the function shape, different techniques yield the best adjusted- R^2 values at different sample sizes. For *bowl* and *multi-local minima* shaped functions, MARS and ALAMO models give the highest values for the largest percentage of the datasets at smaller sample sizes. The hinge functions of the MARS

models and the several available nonlinear transformations of ALAMO models may make them particularly suitable for mimicking the convex behavior of the *bowl*-shaped functions and for approximating the somewhat “noisy” surface of the *multi-local minima* functions. When the sample size grows, GP models also begin to perform well for multi-local minima functions as they gain more information for more accurate interpolations. When the sample size grows, GP models also begin to perform well for *multi-local minima* functions. Also, GP models are selected the most frequently at all sample sizes for the *other* functions, which do not fit into any of the other four defined shape categories. ANN models provide the best models for *plate-shaped* functions with smaller samples but are outperformed as sample size increases. The model selection for *valley* functions is spread fairly evenly among a few modeling techniques, which may suggest that additional characteristics should be considered when selecting a surrogate model for a similar dataset.

RF models did not perform the best for any of the datasets considered. SVR performed best for very few, indicating that if adjusted- R^2 is the performance metric of interest, these models may not be suitable choices. These results indicate that there is some dependence of the surrogate model surface approximation performance on the overall shape of the function the dataset was generated from, the input dimension, and the sample size, especially when all these factors are considered together.

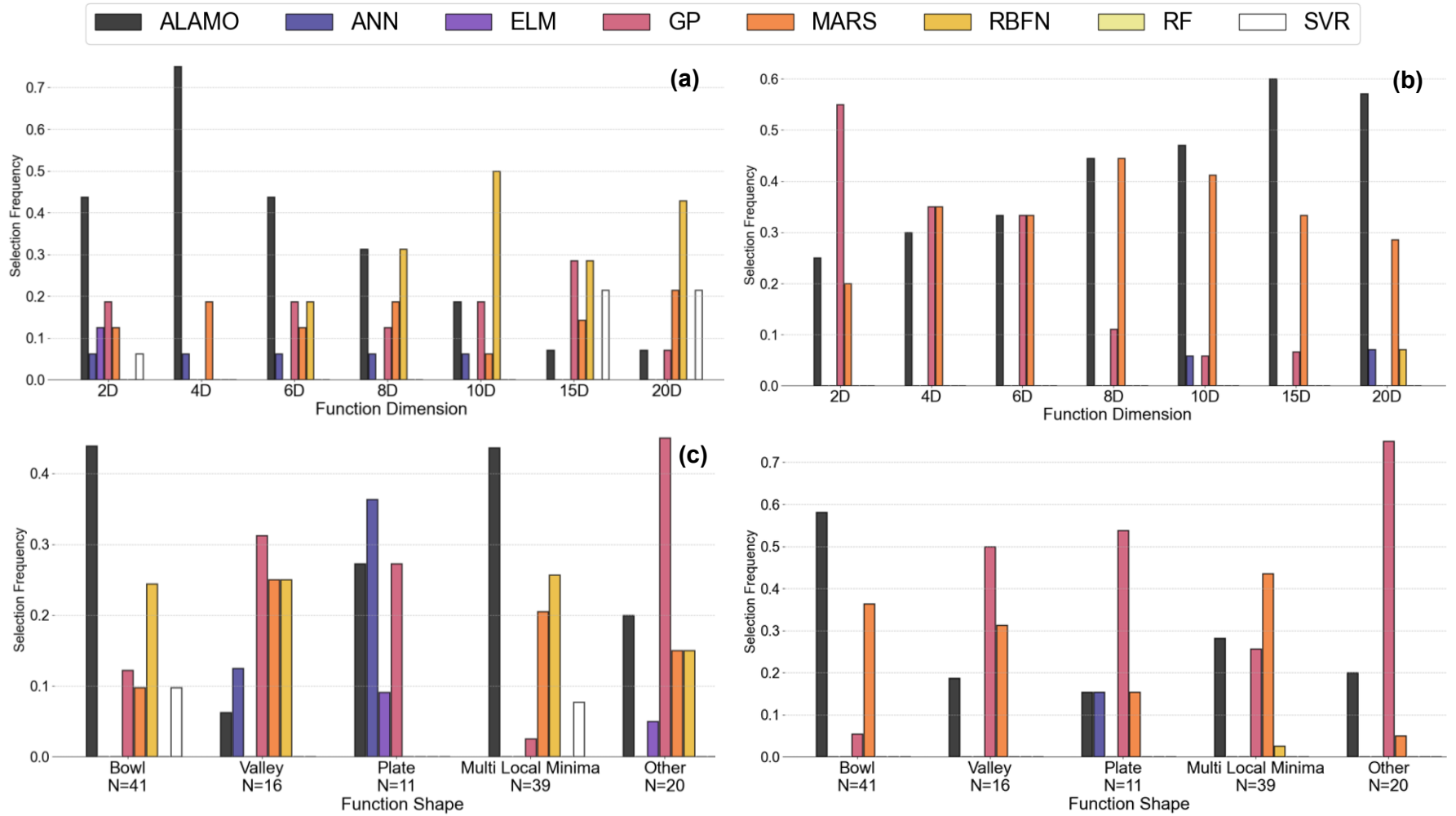


Figure 3. Percentage of datasets grouped by input dimension for which each surrogate modeling technique had the highest adjusted- R^2 for sample sizes: (a) 50 and (b) 1600. Percentage of datasets grouped function shape for which each surrogate modeling technique had the highest adjusted- R^2 for sample sizes: (c) 50 and (d) 1600.

4.2.2 Effect of Underlying Function Input Dimension and Function Shape on Surface

Approximation Performance

Results obtained for the effect of input dimension of the test function (and resulting training dataset) on the nRMSE and adjusted- R^2 for each surrogate modeling technique at a sample size of 50 are summarized in Fig. 4. RBFN and MARS models have better performance than the other techniques at the smaller sample sizes tested. Although many of the techniques appear to perform comparably for approximation based on their nRMSE's, the performance metric values deteriorate when adjusted for the model size with the adjusted- R^2 . This indicates that while many of the techniques can capture the general surface of the test functions at small sample sizes, they do so at the expense of overfitting. This overfitting trend is particularly apparent for ELM and ANN models, for example. With increasing sample sizes, the adjusted- R^2 values and nRMSE follow similar trends, as increased sample sizes allow for larger models that can still avoid overfitting.

The results for adjusted- R^2 are summarized for sample sizes of 400 and 1600 in Fig. 5. In general, at these larger sample sizes, MARS models perform the most robustly with respect to the input dimensions. ANOVA analysis confirms this robust behavior with respect to dimensions for MARS models, revealing no significant difference between the nRMSE values of each dimension ($p = 0.43$). MARS and GP models at lower input dimensions yield higher values, close to one, of adjusted- R^2 . However, the GP model performance worsens as the dimension increases, which matches the trend from the results for model selection (Fig. 3b), illustrating the dependence of model performance on dimension. The robust performance of MARS models may be due to their effective partitioning of the design space with the hinge functions and the accurate modeling of nonlinearities in these partitions by the products of hinge functions. Input dimension has different levels of effects on the surrogate modeling technique performance at larger sample sizes. RF and

RBFN model performance becomes progressively worse with increasing dimensions, while ALAMO model performance does not change much at different input dimensions. ALAMO's robust approximation performance with respect to input dimension may be due to its ability to perform multiple nonlinear transformations for each input dimension separately.

While the selection of a modeling form by adjusted- R^2 can be useful, selecting a single surrogate model as the best for a dataset may be misleading, as multiple models may perform almost identically for the same dataset. For example, although ALAMO models are selected most frequently as best for *bowl*-shaped test functions (Fig. 3c and Fig. 3d), MARS and GP models are selected most frequently as second-best when ALAMO models are the best performing. However, statistical analysis revealed that, on average, MARS models give higher adjusted- R^2 values than ALAMO for *bowl*-shaped functions, and GP model performance was not significantly different from that of ALAMO (at a significant level of 0.05). Furthermore, ANN models were selected as best (with the highest adjusted- R^2) for *plate*-shaped functions most frequently (Fig. 3b and Fig. 3c), but their adjusted- R^2 values (Fig. 5b) were only significantly different from RF, SVR, and ELM models for that shape category. Based on these results, multiple surrogate modeling techniques can be successfully applied to a dataset to produce similarly accurate approximations, and one may not need to rely on a single best choice.

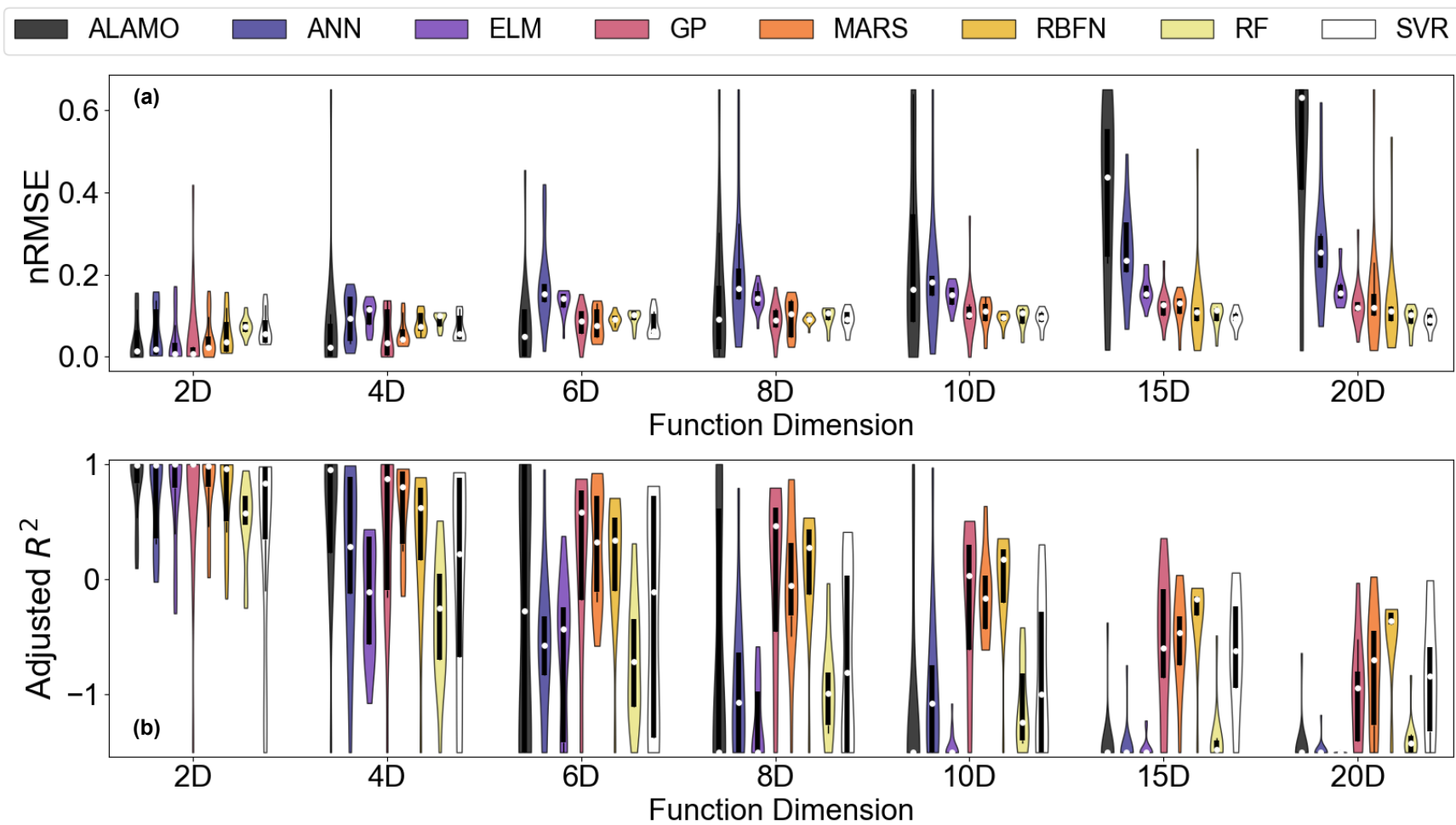


Figure 4. (a) nRMSE and (b) adjusted- R^2 for datasets grouped by underlying function dimension.

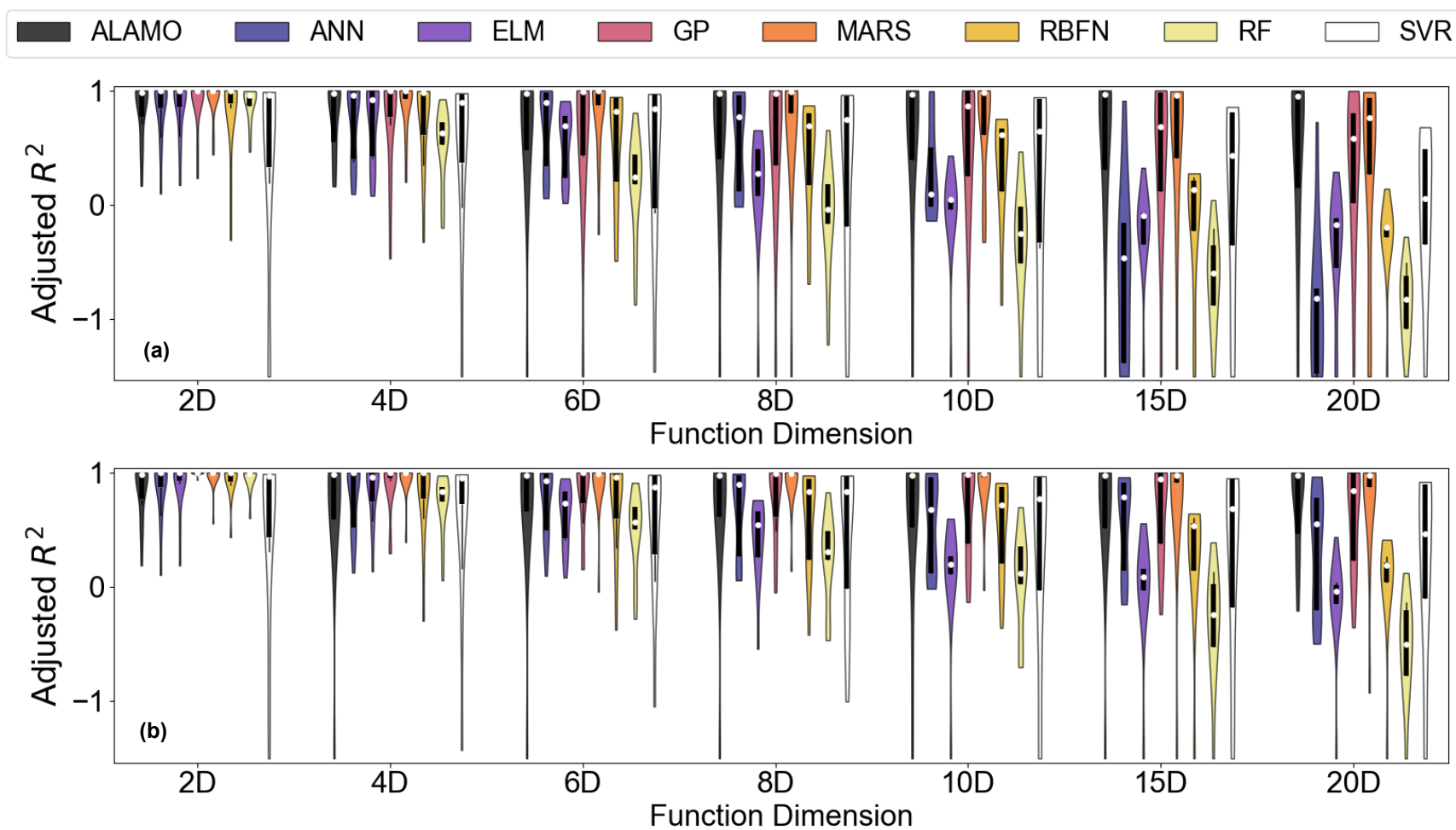


Figure 5. Adjusted- R^2 for models trained with sample sizes of (a) 400 and (b) 1600.

Results for the effects of the underlying function shape on the performance of the surrogate models for surface approximation are summarized in Fig. 6. All of the surrogate models had poor approximation performance and a high level of overfitting, indicated by negative adjusted- R^2 values, with respect to the function shape at sample sizes less than 200. Therefore, adjusted- R^2 results for sample sizes of 400 and 1600 are presented. GP and MARS models provide the most robust performance when considering the test function shape, though none of the techniques perform well overall for the test functions with *multi-local minima* shape.

The function shape does have an impact on the surrogate models' performance for some of the other techniques. Although overall GP and MARS models give significantly lower values of nRMSE than the other techniques ($p < 0.05$), when considering only *bowl*-shaped functions, ALAMO models provided the lowest nRMSE values and best performance ($p < 0.05$). In addition, while ELM's have poor performance in general for *bowl* and *valley*-shaped functions, they perform very well in approximating *plate*-shaped functions, with adjusted- R^2 values close to one. Both ANN and ELM models demonstrate improved performance for *plate*-shaped functions in comparison to the other shape categories. The on-or-off nature of the nodes and activation functions in these model types may make them especially suitable to approximate the flat or nearly so portions of the *plate*-shaped surfaces. results for surface approximation suggest that for datasets where specific characteristics are not available, a MARS or GP model would be appropriate to select as a general guideline. However, if characteristics are available, other models might provide a better approximation.

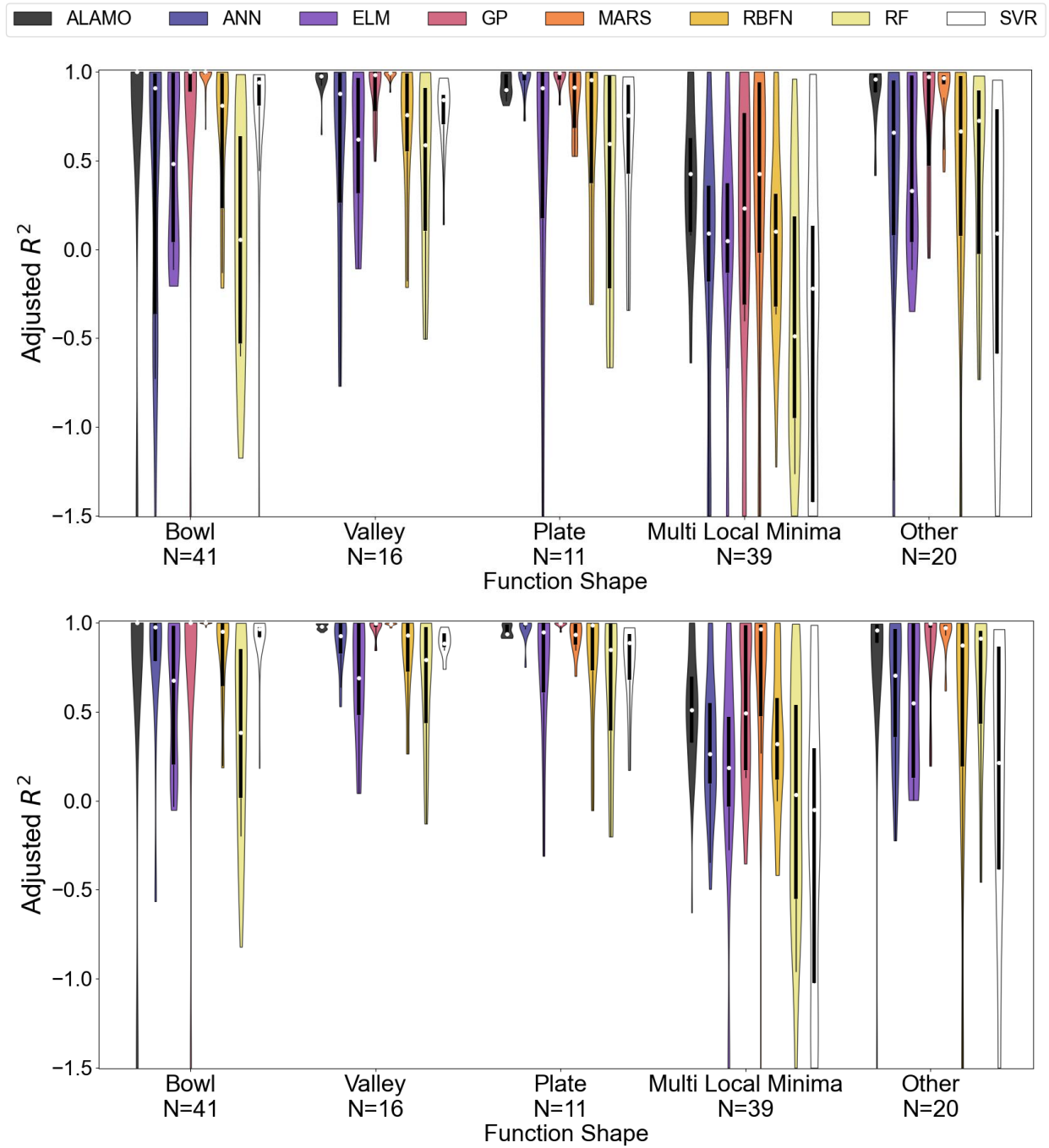


Figure 6. Adjusted R^2 for models trained with sample sizes of (a) 400 and (b) 1600 group by underlying function shape. N values below the function dimensions indicate the number of test functions used for each shape category.

4.3 Comparison of surrogate modeling technique performance for surrogate-based optimization

The computational experiments for surrogate-based optimization were executed by using each surrogate model to estimate the minimum of each function and the location of the minimum. Then, these results were compared to the global minimum and its true location using two metrics, D_{opt} (Eq. 5) and G_{opt} (Eq. 6). Results are summarized in Figs. 7, 8 and 9, where we define a model as having located the optimum when it obtains a D_{opt} or G_{opt} value less than a threshold. The thresholds for D_{opt} and G_{opt} are 5% and 0.01%, respectively. Threshold for D_{opt} was selected in terms of how close the estimate needed to be to still get a reasonable estimate of the optimum value with the predicted location (within 1% error of the output range) and to also yield a reasonable separation in the performance of models. The G_{opt} threshold was selected using 1% of the output range as a measure of a good model fit.

Figure 7 shows the results for how well the surrogate models locate the global minimum of each test function when they are grouped by the function dimension for sample sizes of 50 (Fig. 7a) and 400 (Fig. 7b). Surrogate-based optimization performance with respect to underlying function shape did not differ significantly with sample size, so only results for 1600 samples are presented here (Figs. 8 and 9). RF and SVR models, in general, locate the minima for the highest fraction of the datasets when datasets are grouped by both shape (Fig. 9a) and input dimension (Fig. 8a). ANOVA analysis on the mean D_{opt} of those two techniques versus that of the others indicates that the locations given SVR and RF values are significantly lower ($p < 0.05$). Contrastingly, both techniques had some of the worst performances for approximating the design space, with higher values for nRMSE and lower values of adjusted- R^2 .

While the RF models perform well in capturing the overall curvature of the underlying function in each dataset, they perform poorly for predicting the actual output values. This may be

due to their utilization of decision trees. The “rules” of the decision trees that determine movement between nodes provide less accurate, more noisy predictions for outputs but may be effective in dividing the domain of the dataset in a way that allows the solver to pinpoint the location of the minimum accurately. The support vectors in SVR models may serve a similar function to the decision tree rules in RF models. GP models perform most robustly in estimating the actual global minima values, in general, with respect to both shape and dimension, which may be related to their ability to approximate the surfaces for the datasets accurately.

Both function input dimension and function shape impacted the surrogate models’ estimation of optimum values. While ANN models only identify the optimum value for about 25% of the *bowl*-shaped test functions (Fig. 9b), they can identify close to 80% of the optimum values for the *plate*-shaped functions. On the other hand, ALAMO models can identify optimum values much more accurately for *bowl*-shaped functions than for *plate*-shaped ones. The optimum value estimation seems to be more closely linked to the approximation performance than is the estimation of the optimum location, as ALAMO models were more accurate in approximating *bowl*-shaped functions and ANN models were more accurate for *plate*-shaped ones. At the higher input dimensions of 10 and 15, the optimization problems of the many surrogate models were not solved to 0.001% optimality gap within 48 hours (wall time). Specifically, none of the optimization problems for GP, ELM or RBFN models, and very few of the SVR and ANN could be solved within the allotted computational time. In contrast, the optimization problems constructed using RF, MARS, and ALAMO models were solved to optimality within 72 hours (wall time) for all test functions at high input dimension. Therefore, our computational test results recommend using only those three techniques for surrogate-based optimization at input dimensions higher than 10. A

summary of these findings for surrogate-based optimization, as well as those for surface approximation is provided in Table 1.

Table 1. *Summary of findings for surrogate modeling technique performance*

Model	Advantages	Disadvantages
ALAMO	-Accurate for approximation and optimization of convex (“bowl”-shaped functions) -Relatively short optimization solution times	
ANN	-Accurate approximation and optimization of <i>plate</i> -shaped functions	-Requires a relatively large number of samples for approximating several function types accurately -High computational time for optimization solutions, particularly at high input dimension
ELM	-Accurate approximation of <i>plate</i> -shaped functions -Relatively short model training times	-Requires a relatively large number of samples for approximating several function types accurately -High computational time for optimization solutions, particularly at high input dimension
GP	-Accurate approximation of several function types	-High computational time for optimization solutions, particularly at high input dimension -High model training times
MARS	-Accurate approximation of several function types -Optimization problems remain tractable, even at high input dimension	-Not as accurate for optimization of test functions
RBFN	-More accurate than other techniques for optimization at smaller sample sizes	-High computational time for optimization solutions, particularly at high input dimension
RF	-MILP structure of optimization problem provides accurate optimization solutions with relatively low solution times	-Less accurate than other techniques for approximation surfaces in general
SVR	-Relatively short model training times -Accurate optimization of several function types, particularly at small sample sizes	-Less accurate than other techniques for approximation surfaces in general -High computational time for optimization solutions, particularly at high input dimension

4.3.1 Computational Efficiency of Solving the Resulting Optimization Problems

The solvers used for optimization are provided in Table 2. For each modeling technique, the selected solver was the most appropriate for the resulting optimization model (Table 2). The average computational times required for solving the optimization problems to estimate the global minima of the test functions for each surrogate modeling technique are also included in Table 2. The average solution times reported in Table 2 are for the optimization problems that were solved to optimality within 48 hours. The solution time is dependent on final model size and structure, with larger, more complex models taking a much longer time to solve than linear models or models with fewer parameters.

GP models have the highest average solution times because the radial basis kernel function used and the interpolation of the model based on the training data points result in a large, highly nonlinear optimization model. The high degree of nonlinearity and number of parameters in the optimization problems of ANN, ELM, RBFN, and SVR models also presented difficulties, with a large proportion of the problems not being solved within 48 hours (wall time). The range of optimality gaps for the models that did not reach a gap of 0.001% within the set time limit were 1% - 10% for ANN, 0.5% - 14% for ELM, 0.009% - 1×10^7 % for GP, 4% - 1×10^3 % for RBFN, and 0.14% to 195% for SVR.

Although they are some of the more accurate models for locating optima and the resulting optimization models do not, in general, become computationally intractable, the optimization problems of RF models have the highest average value for the solution time. The solution time for RF-based optimization problems may be reduced by developing specialized algorithms that exploit the special structure of RF model MILPs as RF models were successful in pinpointing the location

of the minimum. While MARS models had relatively low optimization solution times, the solutions given by MARS models were less accurate than those of other methods.

Table 2. *Solvers and solution times for surrogate-based optimization (NLP = Non-linear program, MINLP = Mixed integer non-linear program, MILP = Mixed integer linear program).*

Surrogate Model	Resulting Optimization Model	Solver	Average Solution Time (min)
ALAMO	NLP	BARON	4.4
ANN	NLP	BARON	664
ELM	NLP	BARON	9.4
GP	NLP	BARON	2169
MARS	MINLP	ANTIGONE	7.9
RBFN	NLP	BARON	33
RF	MILP	CPLEX	27
SVR	NLP	BARON	288

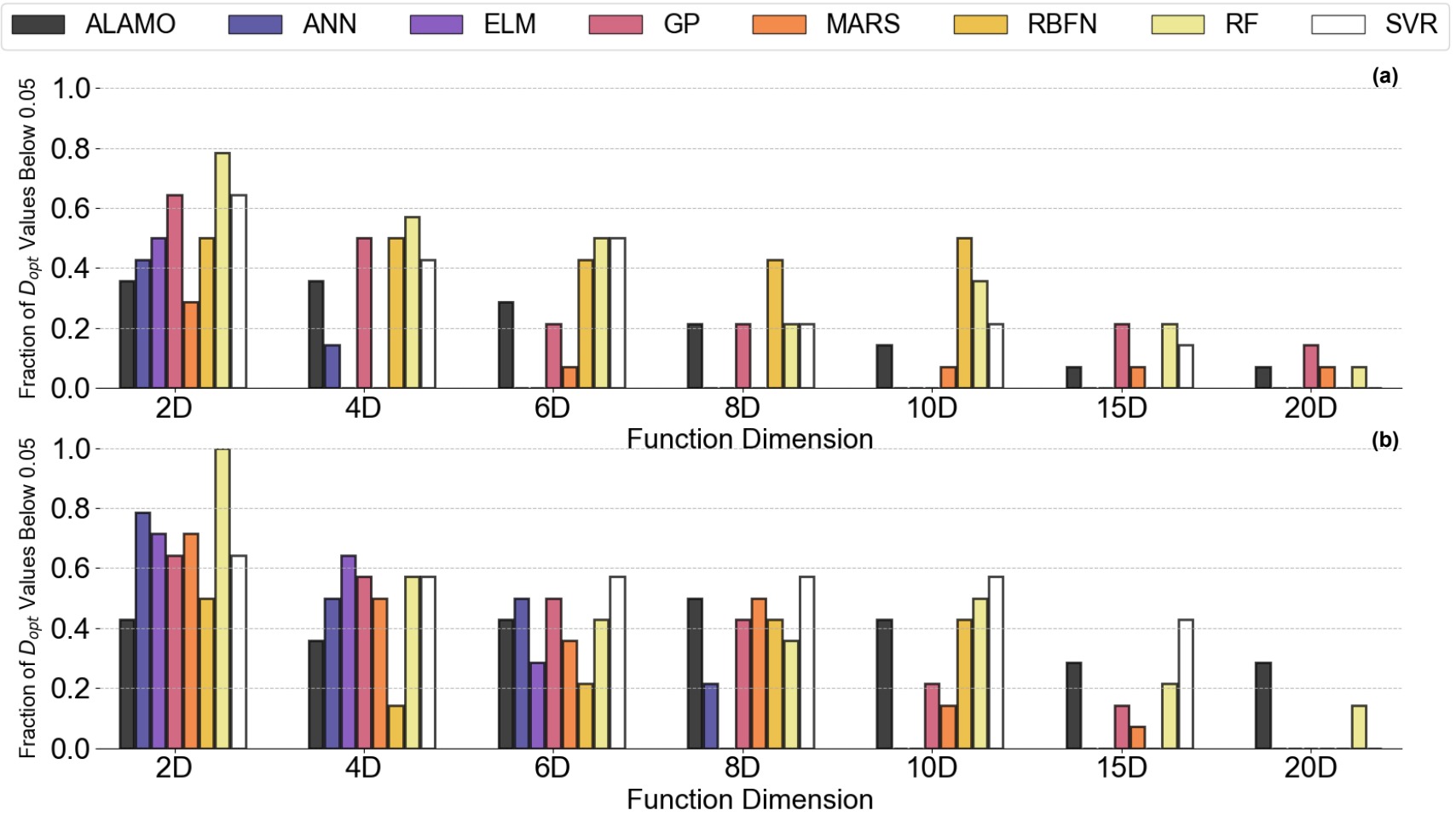


Figure 7. Fraction of datasets with D_{opt} less than 5% grouped by input dimension for sample size (a) 50 and (b) 400.

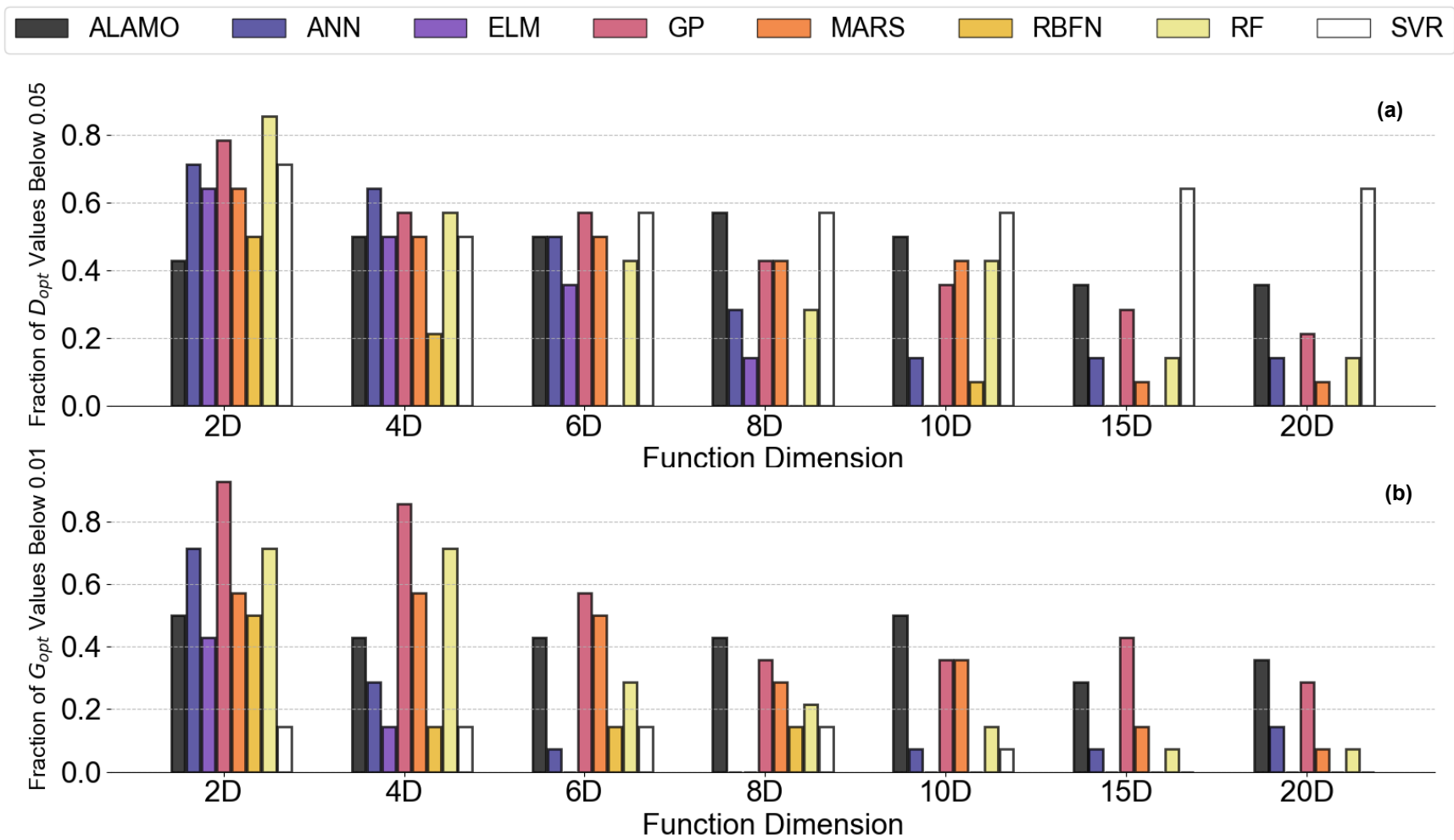


Figure 8. Fraction of datasets with (a) D_{opt} and (b) G_{opt} less than threshold grouped by input dimension for sample size of 1600.

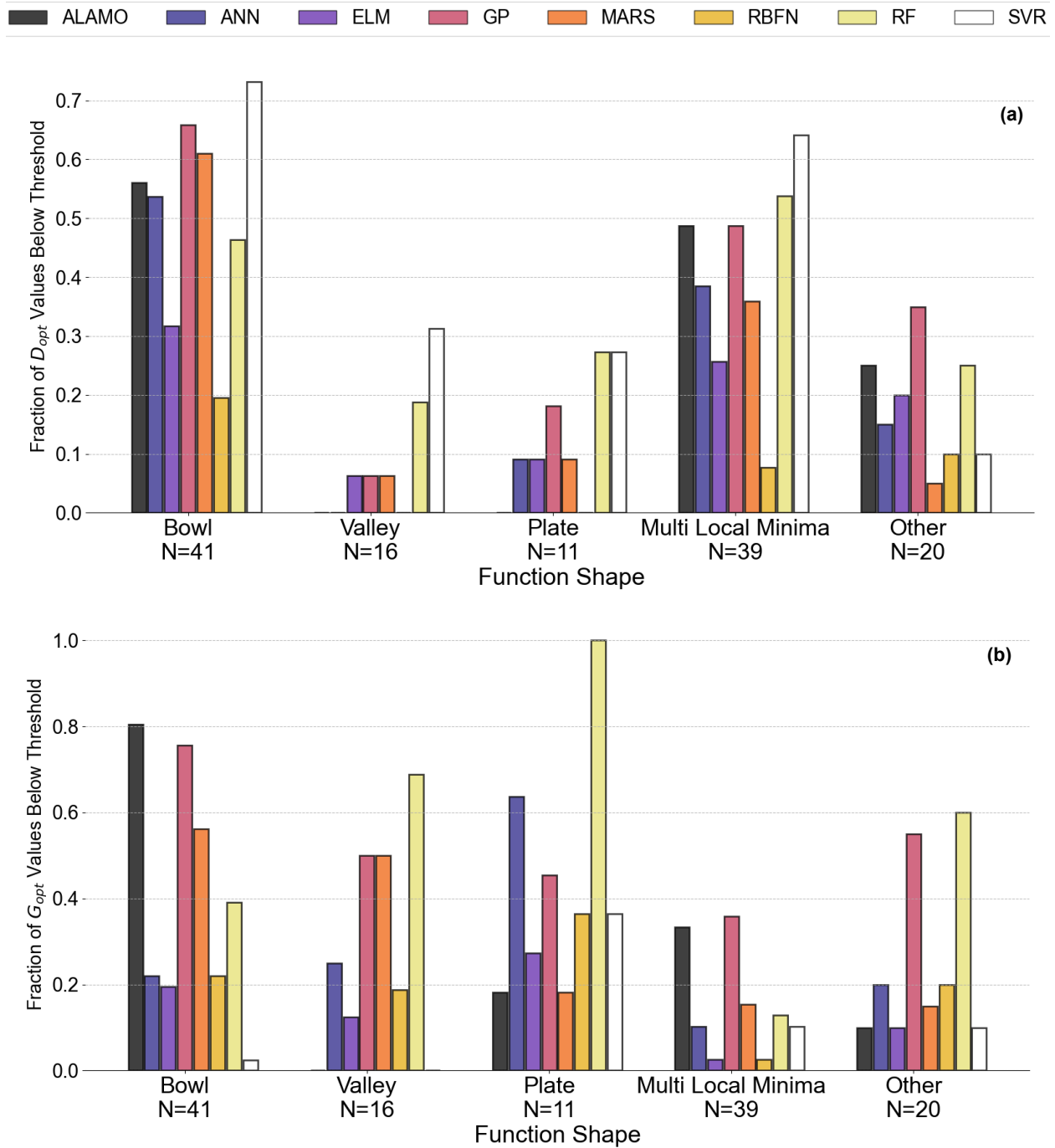


Figure 9. Fraction of datasets with (a) D_{opt} and (b) G_{opt} less than threshold grouped by function shape for sample size of 1600. N values below the function dimensions indicate the number of test functions used for that input dimension.

4.4 Functions for Which None of the Surrogate Modeling Techniques were Accurate

For both the surface approximation and surrogate-based optimization applications, there were some test functions for which none of the surrogate modeling techniques investigated were able to achieve accurate estimates, even at the largest sample size. The two-dimensional projections of the three functions that none of the surrogate modeling techniques were able to fit with an adjusted- R^2 of at least 0.90 are shown in Fig. 10(a) – (c). These functions all come from the *multi-local minima* shape category. The frequency of these functions' peaks may make the surfaces too noisy for approximating with any of the techniques, and other modeling approaches may be necessary to get an accurate approximation.

The two-dimensional projections of a selection of the functions that none of the surrogate modeling techniques located the optimum within a D_{opt} value of 5% are shown in Fig. 10(d) – (f). There were seven of these functions. When compared to the rest of the test functions, these seven had a range of output values that were several orders of magnitude higher, which may have given the solvers used difficulty in locating the optimum point. Most came from the *plate* and *valley*-shaped function categories. The large flat segments of these surfaces could have caused difficulty in locating the optimums, causing the solvers to get trapped in them. There was no overlap between the functions that were not modeled accurately for approximation and the functions whose optimum locations could not be found, further indicating that selection of a surrogate model for the two different applications may be unrelated. Although there were some common characteristics for the functions that could not be adequately modeled using these approaches, further work is needed on the specific characteristics of a dataset that may make it an inappropriate candidate for these traditional surrogate modeling methods.

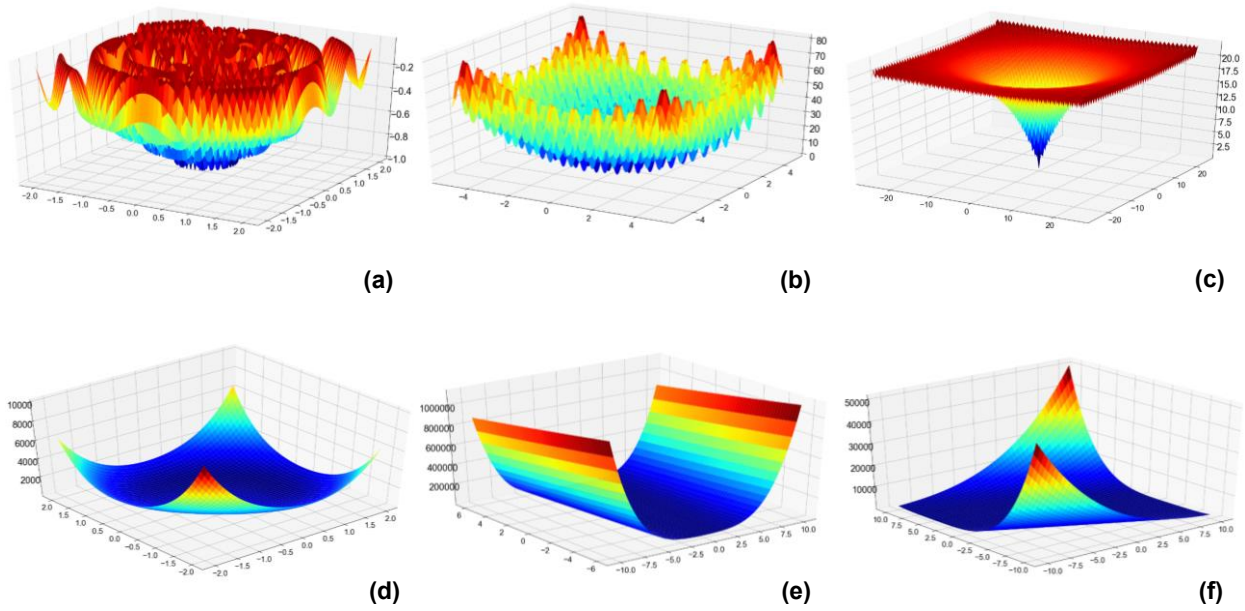


Figure 10. Functions that could not be approximated by any of the surrogate models (a) – (c) or for which the optimum could not be located (d) – (f). (a) Eggholder function (*multi local minima-shaped*) (b) Rastrigin Function (*multi local minima-shaped*) (c) Ackley function (*multi local minima-shaped*) (d) Perm function (*bowl-shaped*) (e) Rosenbrock function (*valley-shaped*) (f) Zakharov function (*plate-shaped*)

5. Conclusions and Future Directions

The selection of the appropriate surrogate modeling technique depends on both the desired application of the surrogate model and the characteristics of the dataset being modeled. Although surface approximation using surrogate models is not significantly impacted by the choice of space-filling sampling method, the quality of solutions obtained from surrogate-based optimization can be dependent upon the sampling method, particularly at small sample sizes. For general selection rules, MARS and GP models give the most accurate predictions for design space approximation, and RF, SVR and GP models give the most accurate estimations for surrogate-based optimization.

The main limitation of this study is that the analysis was carried out only on relatively smooth functions (with the exception of a few) with only continuous outputs. The results may not be applicable to more noisy data or to data that has binary or integer inputs and/or outputs. In addition, the “shape” data characteristic is not one that can readily be applied to other data in determining which surrogate might be the most appropriate. Future work will focus on developing specific, quantifiable data characteristics related to the shape that can be calculated based only on available inputs and outputs and capture the overall data behavior to make the recommendations for surrogate modeling selection more generalizable to other data.

Acknowledgements

This work was partially funded by the Department of Education GAANN grant #P200A150075, NSF grant #1743445, and RAPID Manufacturing Institute, USA. The authors would also like to acknowledge the Auburn HPC cluster for support on this work.

References

- Afzal, A., Kim, K., Seo, J., 2017. Effects of Latin hypercube sampling on surrogate modeling and optimization. *International Journal of Fluid Machinery and Systems* 10, 240-253. <http://dx.doi.org/10.5293/IJFMS.2017.10.3.240>.
- Ben Salem, M., Tomaso, L., 2018. Automatic selection for general surrogate models. *Struct Multidiscip O* 58, 719-734. 10.1007/s00158-018-1925-3.
- Bhosekar, A., Ierapetritou, M., 2018. Advances in surrogate based modeling, feasibility analysis, and optimization: A review. *Comput Chem Eng* 108, 250-267. 10.1016/j.compchemeng.2017.09.017.
- Breiman, L., 2001. Random forests. *Mach Learn* 45, 5-32. Doi 10.1023/A:1010933404324.
- Burnaev, E.V., Zaytsev, A.A., 2015. Surrogate modeling of multifidelity data for large samples. *J Commun Technol El+* 60, 1348-1355. 10.1134/S1064226915120037.
- Burnak, B., Diangelakis, N.A., Katz, J., Pistikopoulos, E.N., 2019. Integrated process design, scheduling, and control using multiparametric programming. *Comput Chem Eng* 125, 164-184. 10.1016/j.compchemeng.2019.03.004.

Cozad, A., Sahinidis, N.V., Miller, D.C., 2014. Learning surrogate models for simulation-based optimization. *Aiche J* 60, 2211-2227. 10.1002/aic.14418.

Davis, S., Cremaschi, S., Eden, M., 2017. Efficient Surrogate Model Development: Optimum Model Form Based on Input Function Characteristics, in: Espuna, A., Graells, M., Puigjaner, L. (Eds.), 27th European Symposium on Computer Aided Process Engineering (ESCAPE 27). Elsevier, Barcelona, Spain, pp. 457-462.

De Maesschalck, R., Jouan-Rimbaud, D., Massart, D.L., 2000. The Mahalanobis distance. *Chemometr Intell Lab* 50, 1-18. Doi 10.1016/S0169-7439(99)00047-7.

Diwekar, U.M., 2003. A novel sampling approach to combinatorial optimization under uncertainty. *Comput Optim Appl* 24, 335-371. Doi 10.1023/A:1021866210039.

Drucker, H., Shahrory, B., Gibbon, D.C., 2002. Support vector machines: relevance feedback and information retrieval. *Inform Process Manag* 38, 305-323. Doi 10.1016/S0306-4573(01)00037-1.

Du, D., Yang, H., Ednie, A.R., Bennett, E.S., 2016. Statistical Metamodeling and Sequential Design of Computer Experiments to Model Glyco-Altered Gating of Sodium Channels in Cardiac Myocytes. *IEEE J Biomed Health Inform* 20, 1439-1452. 10.1109/JBHI.2015.2458791.

Friedman, J.H., 1991. Multivariate Adaptive Regression Splines - Rejoinder. *Ann Stat* 19, 123-141. DOI 10.1214/aos/1176347973.

Garud, S.S., Karimi, I.A., Kraft, M., 2017. Design of computer experiments: A review. *Comput Chem Eng* 106, 71-95. 10.1016/j.compchemeng.2017.05.010.

Golkarnarenji, G., Naebe, M., Badii, K., Milani, A.S., Jazar, R.N., Khayyam, H., 2018. Support vector regression modelling and optimization of energy consumption in carbon fiber production line. *Comput Chem Eng* 109, 276-288. 10.1016/j.compchemeng.2017.11.020.

Gomm, J.B., Yu, D.L., 2000. Selecting radial basis function network centers with recursive orthogonal least squares training. *Ieee T Neural Networ* 11, 306-314. Doi 10.1109/72.839002.

Halton, J.H., Smith, G.B., 1964. Algorithm-247 - Radical-Inverse Quasi-Random Point Sequence [G5]. *Commun Acm* 7, 701-702. Doi 10.1145/355588.365104.

Han, Z., Zhang, K., 2012. Surrogate-Based Optimization, in: Roeva, O. (Ed.), *Real-World Applications of Genetic Algorithms*. InTech Open, Rijeka, Croatia, pp. 343-362. 10.5772/2674.

Harrell, F.E., Lee, K.L., Califf, R.M., Pryor, D.B., Rosati, R.A., 1984. Regression Modeling Strategies for Improved Prognostic Prediction. *Stat Med* 3, 143-152. DOI 10.1002/sim.4780030207.

Hart, W.E., Laird, C.D., Watson, J.-P., Woodruff, D., L., Hackebail, G.A., Nicholson, B.L., Sirola, J.D., 2017. *Pyomo - Optimization Modeling in Python*, 2 ed. Springer, Boston, MA. <https://doi.org/10.1007/978-1-4614-3226-5>.

Hart, W.E., Watson, J.-P., Woodruff, D.L., 2011. Pyomo: modeling and solving mathematical programs in Python. *Math Program Comput* 3, 219 - 260. <https://doi.org/10.1007/s12532-011-0026-8>.

Haykin, S., 2009. *Neural Networks and Learning Machines*, 3rd ed. Pearson Education, Inc., Upper Saddle River, New Jersey.

Huang, G.B., Zhu, Q.Y., Siew, C.K., 2006. Extreme learning machine: Theory and applications. *Neurocomputing* 70, 489-501. 10.1016/j.neucom.2005.12.126.

Hussain, K., Salleh, M.N.M., Cheng, S., Naseem, R., 2017. Common Benchmark Functions for Metaheuristic Evaluation: A Review. *International Journal of Informatics Visualization* 1, 218-223. 10.30630/ijoiv.1.4-2.65. .

Iooss, B., Boussouf, L., Feuillard, V., Marrel, A., 2010. Numerical studies of the metamodel fitting and validation process. *International Journal of Advances in Systems and Measurements* 3, 11-21.

Jamil, M., Yang, X.-S., 2013. A Literature Survey of Benchmark Functions for Global Optimization Problems. *International Journal of Mathematical Modelling and Numerical Optimization* 4, 150-194. 10.1504/IJMMNO.2013.055204.

Joe, S., Kuo, F.Y., 2008. Constructing Sobol' Sequences with Better Two-Dimensional Projections. *Siam J Sci Comput* 30, 2635-2654. 10.1137/070709359.

Ju, Y.P., Zhang, C.H., Ma, L., 2016. Artificial intelligence metamodel comparison and application to wind turbine airfoil uncertainty analysis. *Adv Mech Eng* 8. Artn 1687814016647317 10.1177/1687814016647317.

Luo, J.N., Lu, W.X., 2014. Comparison of surrogate models with different methods in groundwater remediation process. *J Earth Syst Sci* 123, 1579-1589. DOI 10.1007/s12040-014-0494-0.

McKay, M.D., 1992. Latin Hypercube Sampling as a Tool in Uncertainty Analysis of Computer-Models. 1992 Winter Simulation Conference Proceedings, 557-564.

Mehmani, A., Chowdhury, S., Meinrenken, C., Messac, A., 2018. Concurrent surrogate model selection (COSMOS): optimizing model type, kernel function, and hyper-parameters. *Struct Multidiscip O* 57, 1093-1114. 10.1007/s00158-017-1797-y.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. *J Mach Learn Res* 12, 2825-2830.

Quiroz, J.C., Mariun, N., Mehrjou, M.R., Izadi, M., Mison, N., Radzi, M.A.M., 2018. Fault detection of broken rotor bar in LS-PMSM using random forests. *Measurement* 116, 273-280. 10.1016/j.measurement.2017.11.004.

- Rahman, R.K., Ibrahim, S., Raj, A., 2019. Multi-objective optimization of sulfur recovery units using a detailed reaction mechanism to reduce energy consumption and destruct feed contaminants. *Comput Chem Eng* 128, 21-34. 10.1016/j.compchemeng.2019.05.039.
- Rasmussen, C.E., Williams, C.K.I., 2005. *Gaussian Processes for Machine Learning*. *Adapt Comput Mach Le*, 1-247.
- Sokolov, M., Ritscher, J., MacKinnon, N., Souquet, J., Broly, H., Morbidelli, M., Butte, A., 2017. Enhanced process understanding and multivariate prediction of the relationship between cell culture process and monoclonal antibody quality. *Biotechnol Prog* 33, 1368-1380. 10.1002/btpr.2502.
- Surjanovic, S., Bingham, D., 2013. *Virtual Library of Simulation Experiments*, Simon Fraser University. <http://www.sfu.ca/~ssurjano>.
- Villa-Vialaneix, N., Follador, M., Ratto, M., Leip, A., 2012. A comparison of eight metamodeling techniques for the simulation of N₂O fluxes and N leaching from corn crops. *Environ Modell Softw* 34, 51-66. 10.1016/j.envsoft.2011.05.003.
- Wang, C., Duan, Q.Y., Gong, W., Ye, A.Z., Di, Z.H., Miao, C.Y., 2014. An evaluation of adaptive surrogate modeling based optimization with two benchmark problems. *Environ Modell Softw* 60, 167-179. 10.1016/j.envsoft.2014.05.026.
- Wang, G.G., Shan, S., 2007. Review of metamodeling techniques in support of engineering design optimization. *J Mech Design* 129, 370-380. 10.1115/1.2429697.
- Williams, B., Cremaschi, S., 2019. Surrogate Model Selection for Design-Space Approximation and Surrogate-Based Optimization, in: Munoz, S., Laird, C., Realff, M. (Eds.), *Ninth International Conference on Foundations of Computer-Aided Process Design (FOCAPD-19)*. Elsevier B.V., Copper Mountain, CO, USA, pp. 353-358.
- Williams, B., Lobel, W., Finklea, F., Halloin, C., Ritzenhoff, K., Manstein, F., Mohammadi, S., Hashemi, M., Zweigerdt, R., Lipke, E., Cremaschi, S., 2020. Prediction of Human Induced Pluripotent Stem Cell Cardiac Differentiation Outcome by Multifactorial Process Modeling. *Front Bioeng Biotechnol* 8, 851. 10.3389/fbioe.2020.00851.
- Zhang, D.H., Qian, L.Y., Mao, B.J., Huang, C., Huang, B., Si, Y.L., 2018. A Data-Driven Design for Fault Detection of Wind Turbines Using Random Forests and XGboost. *Ieee Access* 6, 21020-21031. 10.1109/Access.2018.2818678.