PRESTO: Predictive REcommendation of Surrogate models to approximate and Optimize

Bianca Williams[1], Joannah Otashu[2], Simon Leyland[2], Mario R. Eden[1], Selen Cremaschi[1*]

[1]Department of Chemical Engineering, Auburn University, Auburn, AL

[2]Siemens Process Systems Engineering Inc.

[*]selen-cremaschi@auburn.edu

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. Many techniques exist for surrogate modeling; however, selecting suitable techniques for a given application remains an open challenge. This work describes PRESTO, a Random Forest classifier-based tool, to recommend appropriate surrogate modeling techniques for a given dataset for surface approximation and surrogate-based optimization, using attributes calculated only using the input and output data. The tool identifies the techniques for surface approximation with an accuracy of 91% and a precision of 90% and for surrogate-based optimization with an accuracy of 98% and a precision of 99%. PRESTO was tested on data generated from a high fidelity process model of the cumene production. Its performance was comparable to the training data. PRESTO enables computational time savings for selecting surrogate model forms by avoiding expensive trial-and-error methods.

Keywords: model selection, surface approximation, surrogate-based optimization, meta-learning

1. Introduction

Several chemical engineering applications, including process design and process synthesis, operations, and supply chain management, involve complex, high-fidelity simulations and/or

physical experiments, requiring significant resources in terms of both cost and time. High resource requirements can present considerable challenges for modeling these complex processes, as the computational and monetary costs of collecting the necessary data may become prohibitive. In addition, optimization using traditional gradient-based methods may be impractical for these applications because gradient information is not readily available, or approximating gradients is infeasible due to the required expense for multiple simulation evaluations or experiments. To overcome these challenges, cheaper surrogates that mimic the simulations' overall behavior can be constructed and used in their place for both surface approximation and surrogate-based optimization purposes.

Surrogate models, also known as response surfaces, black-box models, metamodels, or emulators, are simplified approximations of more complex, higher-order models. These models map input data to output data when the actual relationship between the two is unknown or computationally expensive to evaluate (Han and Zhang, 2012). Various techniques have been developed for constructing surrogate models for both regression and classification tasks (Breiman, 2001; Cozad et al., 2014; Drucker et al., 2002; Rasmussen and Williams, 2005). The current common practice for choosing a model form from the many available techniques relies on process-specific expertise or expensive trial-and-error methods. When selecting a surrogate model with user expertise, only a small subset of the many possible techniques that the user is most familiar with may be considered as candidates. This selection method, as well as trial and error, which is limited by computational resources, may fail to exploit the large pool of surrogate modeling techniques available and lead to a sub-optimal model selection. A systematic, automated procedure for selecting the appropriate surrogate model for a given application would avoid this issue.

Recent advances in automating the surrogate model selection process include the development of the tool Concurrent Surrogate Model Selection (COSMOS), which uses a genetic search algorithm with sequential k-fold cross-validation to identify the best model for an application (Mehmani et al., 2018). While this method allows users to explore a wide range of candidate surrogates to select the best one, it still involves a considerable computational expense for training multiple models. Progress has been made in recent works in generalizing the process for selecting a surrogate model to approximate a surface by using meta-learning approaches to build selection frameworks, avoiding expensive trial-and-error methods (Cui et al., 2016; Garud et al., 2018). These meta-learning approaches rely on the knowledge pyramid, where the selection framework learns how to best select surrogate models based on past modeling computational experiments results (Vilalta and Drissi, 2002). These frameworks provide "best" recommendations for surrogate modeling techniques based on characteristics, or attributes, calculated from the modeled data. Furthermore, the framework developed by Garud et al. (2018) gives a ranking of all the considered surrogate models based on the predicted accuracy of the model. However, neither framework takes model complexity into account, which can lead to overfitting, or considers that multiple models might perform similarly to the one identified as best in terms of their accuracies. The selection of surrogate models for surrogate-based optimization remains an open challenge.

This work aims to develop a framework to automatically select the set of surrogate models that will perform the best for a particular set of data based on the characteristics of the data and the application that the surrogate model would be used for. Our previous work comparing surrogate model performance demonstrated that there is a link between the characteristics of the dataset and how well different surrogate modeling techniques will perform for it for both surface approximation and surrogate-based optimization (Williams and Cremaschi, 2021). To achieve our

aim, we developed PRESTO (Predictive REcommendation of Surrogate models to approximate and Optimize), a random forest-based surrogate model selection tool. Given a set of data, PRESTO classifies each surrogate modeling technique in a set of candidate models as either recommended or not recommended based on the application, surface approximation or surrogate-based optimization. The set of candidate surrogate modeling techniques considered by PRESTO includes Automated Learning of Algebraic Models using Optimization (ALAMO), single hidden layer feed-forward Artificial Neural Networks (ANN), Extreme Learning Machines (ELM), Gaussian Process Regression (GPR), Multivariate Adaptive Regression Splines (MARS), Radial Basis Function Networks (RBFN), Random Forests (RF), and Support Vector Machine Regression (SVR). The tool provides these recommendations without training any of the models, avoiding much computational expense.

The rest of the paper is organized as follows: Section 2 describes the data sets used to train surrogate models for constructing PRESTO and the computational experiments performed. Section 3 describes the feature engineering and extracted data attributes from the computational experiments used to train PRESTO's random forest classifier-based framework. Section 4 provides details on training PRESTO, a brief overview of random forest model structures, and metrics used for its performance evaluation. Section 5 discusses the results for how PRESTO performed on providing surrogate model recommendations for the data used to train the tool. Section 6 presents a case study using PRESTO to select surrogate models for surface approximation for data generated from different unit operations of a cumene production simulation. Section 7 draws conclusions and discusses future research directions.

2. Surrogate Modeling Experiments

2.1 Dataset Generation

Datasets were generated for training surrogate models from a suite of optimization test functions (Surjanovic and Bingham, 2013). The functions with two, four, six, eight, ten, fifteen, and twenty input dimensions were utilized, resulting in 127 test functions. The test functions are grouped by their underlying functional shape. In this analysis, we have considered five shape categories: *bowl*-shaped, *plate*-shaped, *valley*-shaped, *multi-local-minima*-shaped, and *other*-shaped. Full descriptions of the characteristics of each shape category are provided in Williams and Cremaschi (2021). Input-output pairs were generated from each test function to create datasets at seven different sample sizes (50, 100, 400, 800, 1200, and 1600 samples) using Sobol sequence sampling (Joe and Kuo, 2008), a quasi-random low discrepancy sequence, and resulting in 791 generated datasets. Detailed information on the choice of sample sizes and sampling methods is in Williams and Cremaschi (2021).

2.2 Candidate Surrogate Models

The generated datasets were used to train surrogate models using each of the eight candidate surrogate modeling techniques. The eight techniques are Automated Learning of Algebraic Models using Optimization (ALAMO), single hidden layer feed-forward Artificial Neural Networks (ANNs), Extreme Learning Machines (ELMs), Gaussian Process Regression (GPR), Multivariate Adaptive Regression Splines (MARS), Radial Basis Function Networks (RBFNs), Random Forests (RFs), and Support Vector Machine Regression (SVR). These techniques were chosen as candidates to represent a variety of modeling methods with different functional forms and include some of the more popular techniques in the literature.

ALAMO uses a linear summation of nonlinear transformations of the input data to predict output values (Cozad et al., 2014). ANN models attempt to mimic the behavior of neurons in the brain. 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' (Haykin, 2009). ELM models are ANNs where input-to-hidden layer weights are randomly assigned, and hidden layer-to-output weights are estimated using regression (Huang et al., 2006). RBFN models are ANNs where the activation function of the nodes in the hidden layer is a radial basis function (Gomm and Yu, 2000). GPR uses a kernel function to measure similarity between points to predict the value for an unseen point from the training data (Rasmussen and Williams, 2005). MARS models are made up of a linear summation of basis functions, which include constants, a hinge function (or "spline"), or a product of two or more hinge functions (Friedman, 1991). RFs are machine learning models that make output predictions by combining outcomes from a sequence of regression decision trees, called forests (Breiman, 2001). Finally, SVR models transform 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).

2.3 Surrogate Model Training

A model was trained using each of the eight surrogate modeling techniques for each of the generated datasets, resulting in 6328 trained models. Each technique has a unique set of hyperparameters that was optimized while training the models for each dataset to construct the best possible surrogate model without overfitting. More details on the surrogate model hyperparameter optimization are discussed in (Williams and Cremaschi, 2021). After the models were trained for each dataset and sampling method, 100,000 input-output pairs were generated

from the test functions using the Sobol sequence sampling method to test the accuracy of the surrogate models' predictions. To evaluate the performance of the surrogate models for surrogate-based optimization, the optimization models to determine the global minimum of each trained surrogate model were constructed in Pyomo (version 5.6), a Python-based optimization language (Hart et al., 2017; Hart et al., 2011). The resulting optimization problems were solved with a global solver most appropriate for the form of the problem (MINLP, MILP, or NLP) (Williams and Cremaschi, 2021). 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 3.7 and MATLAB 2017b (for RBFN surrogate models).

3. Feature Engineering and Attribute Extraction for Training PRESTO

Attributes calculated based only on the input and output values of each dataset were used as inputs for PRESTO's surrogate model recommendation classification. For surface approximation, the performance metric used to determine if a model would be considered recommended or not is the estimated adjusted R-squared for the model (Eq. 1). The adjustments used for each surrogate modeling technique are listed in (Williams and Cremaschi, 2021). The performance metric used to make recommendations for surrogate-based optimization is the normalized Mahalanobis distance between the optimum point(s) estimated by the surrogate models and the actual optimum location of the underlying test function used to generate the model (Eq. 2).

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

$$D_{opt} = \frac{M\left(x_{opt}, \hat{x}_{opt}\right)}{\max\limits_{i,j} M\left(x_i, x_j\right)} \tag{2}$$

In Eq. (1), 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). In Eq. (2), x_i and x_j are points in the domain space of the dataset, M is the Mahalanobis distance (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, $\hat{x}_{opt}$. M is normalized by the maximum Mahalanobis distance between any two points (x_i, x_j) in the dataset (Eq. 3). Mahalanobis distance is the distance between two points in multivariate space. This distance between two objects, x and y, can be calculated as

$$M(x,y) = \sqrt{(x-y)^T C^{-1}(x-y)} \tag{3}$$

where C^{-1} is the sample covariance matrix. It has an advantage over Euclidean distance as it considers correlations in the dataset, and large scaling differences between the dimensions, because the distances are normalized with variance. The Mahalanobis distance is thus unitless and scale-invariant (De Maesschalck et al., 2000).

3.1 Dataset Attributes

The attributes are used to capture the overall behavior of the datasets using numeric measures. A total of 38 attributes were defined for the datasets. Twenty of the attributes were previously defined and described in detail in Garud et al. (2018). These include attributes related to estimated gradients and curvatures, attributes related to the distribution of the outputs (such as the first four moments of the output value distribution), and attributes related to the dataset's extreme minimum and maximum values.

An additional 18 new attributes are defined in this work. LEAPS2, the model selection framework described in (Garud et al., 2018), was only trained to select models for surface

approximation. In addition, there were no attributes directly related to the distributions of the inputs in the dataset. These additional attributes were constructed to include information about the arrangement of the inputs for the datasets and provide characteristics of the data that may have a larger effect on the optimization performance.

3.1.1 Input Related Attributes

In Eqs. (3) – (5), $\mathrm{M}(x_i, x_j)$ is the Mahalanobis distance between any two points x_i and x_j in the domain space. Let D be equal to the number of input dimensions in the dataset and N be equal to the total number of datapoints.

Minimum Mahalanobis distance: This is the minimum Mahalanobis distance between any two points in the input space (Eq. 4).

$$M_{min} = \min_{i,j} \mathrm{M}(x_i, x_j) \tag{4}$$

Maximum Mahalanobis distance: This is the maximum Mahalanobis distance between any two points in the input space (Eq. 5).

$$M_{max} = \max_{i,j} \mathrm{M}(x_i, x_j) \tag{5}$$

Average Mahalanobis distance: This is the average Mahalanobis distance between any two points in the input space (Eq. 6).

$$M_{avg} = \frac{1}{N^2} \sum \mathrm{M}(x_i, x_j) \tag{6}$$

Euclidean to Mahalanobis distance ratio: This ratio of the average pairwise Euclidean distance to the average Mahalanobis distance estimates the level of correlation, if any, between the dataset inputs and the magnitude of variance. When there is no correlation between the variables, the covariance matrix used to calculate the Mahalanobis distance becomes the identity covariance

matrix, and the Euclidean and Mahalanobis distances become equal to each other, which makes the value of this one. The average Euclidean distance is calculated as

$$E_{avg} = \frac{1}{N^2} \sum \left\| x_i - x_j \right\| \tag{7}$$

where $\left\| x_i - x_j \right\|$ is the Euclidean distance between any two points x_i and x_j. The Euclidean to Mahalanobis distance ratio can then be estimated by

$$R_{E/M} = \frac{E_{avg}}{M_{avg}} \tag{8}$$

3.1.2 Gradient-Based Attributes

From Garud et al. (2018), for any point in the data set, $x^{(i)}$, let , $x^{(j)}$ be its nearest neighbor based on the Euclidean distance and $y^{(i)}$ be its response. Then, the gradient vector of the response, $g^{(i)}$, at $x^{(i)}$ can be estimated using Eq. (9), where $x_d^{(i)}$ is the value of input dimension d for point $x^{(i)}$, and e is a small number related to the precision of the numbers in the dataset. These gradient-based attributes were added to the attribute set because the gradients indicate the overall shape of the surface, and previous studies have shown that surrogate-based optimization performance is dependent on the underlying shape of the surface being modeled (Williams and Cremaschi, 2021).

$$g^{(i)} = \left[g_d^{(i)} = \frac{\left(y^{(i)} - y^{(j)}\right) sign\left(x_d^{(i)} - x_d^{(j)}\right)}{max\left[e, \left|x_d^{(i)} - x_d^{(j)}\right|\right]} \middle| \, d = 1,2, \ldots, D \right] \tag{9}$$

In Eq. (9), the sign function is defined as

$$sign\left(x_d^{(i)} - x_d^{(j)}\right) = \begin{cases} 1, & if\ x_d^{(i)} - x_d^{(j)} \geq 0 \\ -1, & otherwise \end{cases} \tag{10}$$

Average magnitude of gradient vector: This attribute (Eq. 11), which is the average value of the magnitude of the gradient vector $g^{(i)}$, across all sample points, aims to provide a measure of the average steepness of the surface being modeled.

$$g_{avg} = \frac{1}{N}\sum_{i=1}^{N}\left|g^{(i)}\right| \tag{11}$$

Standard deviation of gradient vector magnitudes: The standard deviation of the magnitude of the gradient vector $g^{(i)}$ across all sample points gives an estimate of the non-linearity of the surface (Eq. 12).

$$g_{std} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(\left|g^{(i)}\right| - g_{avg}\right)^2} \tag{12}$$

Minimum and maximum gradient vector magnitudes: These attributes describe the minimum (Eq. 13) and maximum (Eq. 14) values for the magnitudes of the gradient vectors for sample points in the data set.

$$g_{min} = \min_{N} g^{(i)} \tag{13}$$

$$g_{max} = \max_{N} g^{(i)} \tag{14}$$

Ratios of gradient vector magnitudes: These attributes aim to capture the average "bumpiness" or noisiness of the surface by measuring how sharply the gradients change on average throughout the surface (Eqs. 15-17). Higher values of these ratios indicate a noisier surface.

$$g_{avg,min} = \frac{g_{avg}}{g_{min}} \tag{15}$$

$$g_{avg,max} = \frac{g_{avg}}{g_{max}} \tag{16}$$

$$g_{max,min} = \frac{g_{max}}{g_{min}} \tag{17}$$

Skewness of gradient vector magnitudes: The skewness of the gradient magnitudes estimates a measure of the asymmetry of the distribution of the gradient vector magnitudes (Eq. 18).

$$g_{skew} = \frac{\sum_{i=1}^{N}\left(\left|g^{(i)}\right| - g_{avg}\right)^3}{N(g_{std})^3} \tag{18}$$

3.1.3 Response (Output)-Based Attributes

These response-based attributes were developed and added to the attribute set to provide insights into how concentrated (or sparse) the output values are distributed at the extreme high and low output values. Data that is concentrated in certain areas and not well-distributed over the entire possible output space may produce models whose predictions do not generalize well over the space. However, if data is concentrated at extreme values, a trained model may be better able to closely locate the optimum for the dataset.

Upper and lower tail average: These attributes calculate the average value of the response values in the top 5% and bottom 5% of responses.

Upper and lower tail relative size: This is the ratio of the number of output responses in the top (Eq. 19) and bottom (Eq. 20) 5% of values to the total number of data points. These attributes aim to estimate how well distributed the response values are over the range of responses. In Eqs. (19) and (20), N_u and N_l represent the number of output responses in the top 5% and bottom 5% of values, respectively.

$$u_{size} = \frac{N_u}{N} \tag{19}$$

$$l_{size} = \frac{N_l}{N} \tag{20}$$

Ratio of lower to upper tail size: This ratio (Eq. 21) describes how evenly the output responses are distributed between the upper and lower extremes of the output values.

$$r_{l/u} = \frac{l_{size}}{u_{size}} \tag{21}$$

3.1.4 Other Attributes

Average local convex deviation: This deviation (Eq. 27) aims to estimate the convexity of the function from which the input-output data was generated. We hypothesize that this metric may be important for determining the appropriate surrogate modeling technique for surrogate-based optimization.

Let $c^{(m)}, m = 1,2, \dots, M$ be some sample points generated on the input domain of the dataset using Latin hypercube sampling.

$$d_{min} = \min_{m \neq n} \left\| c^{(m)} - c^{(n)} \right\| \tag{22}$$

We can construct a hypersphere of diameter d_{min} (Eq. 22) around each point $c^{(m)}$ to create a local "neighborhood" of dataset points $x^{(i)}$ around each center, where $d_{min}$is the minimum distance between $c^{(m)}$ and any other generated sample point, $c^{(n)}$. Then, we define the convex difference for each point in the neighborhood as in Eq. (23),

$$C_{diff}^{(i),(m)} = \left| y^{(i)} - y_{convex}^{(i)} \right| \tag{23}$$

and the average convex difference in the neighborhood is given in Eq. (24),

$$\bar{C}_{diff}^{(m)} = \frac{1}{K}\sum_{i=1}^{K} C_{diff}^{(i),(m)} \tag{24}$$

where $y_{convex}^{(i)}$ is the response of a known convex function (Eq. (25))

$$y_{convex}^{(i)} = 0.1\left(x^{(i)}\right)^4 \tag{25}$$

for the input $x^{(i)}$. Figure 1 illustrates the process for calculating $C_{diff}^{(i),(m)}$.

The local convex deviation in each neighborhood can then be calculated as in Eq. (26)

$$C_{dev}^{(m)} = \sqrt{\frac{1}{K-1}\sum_{i=1}^{K}\left(C_{diff}^{(i),(m)} - \bar{C}_{diff}^{(m)}\right)^2} \tag{26}$$

where K is the number of points from the dataset in the neighborhood m. The average local convex deviation is given in Eq. (27).

$$\bar{C}_{dev} = \frac{1}{M}\sum_{i=1}^{M} C_{dev}^{(m)} \tag{27}$$

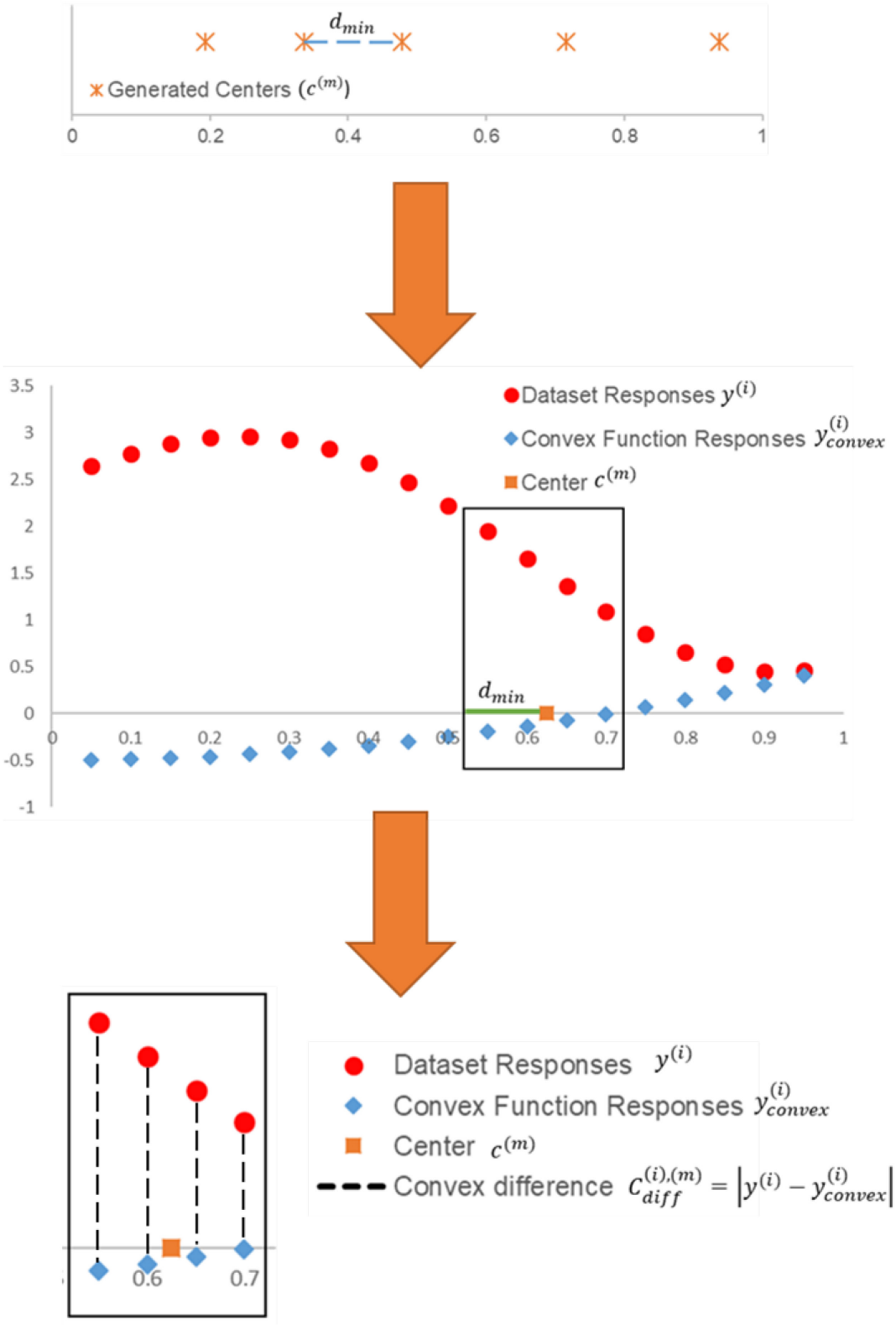

Figure 1. Steps for generating neighborhoods for convex difference calculations.

4. PRESTO Construction and Performance Evaluation

4.1 PRESTO Framework Construction

Random forest classification models were trained for each surrogate modeling technique to predict whether the surrogate should be recommended or not recommended for a dataset. Random forests are surrogate models that make output predictions based on inputs by combining predictions from a collection of decision trees. Each tree in a random forest model 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 (Brieman, 2001). Random forests have successfully been used for both regression and classification tasks, performing with high prediction accuracy for both small sample sizes and high dimensional data.

Separate classification models were trained for surface approximation and surrogate-based optimization. The calculated attributes were used as inputs, and the assigned recommendation classes ("recommended" or "not recommended") were used as outputs. To assign recommendation classes for a dataset, the highest or lowest value out of all the eight surrogate techniques for $\hat{R}^2$ and D_{opt}, respectively, were assigned as "recommended", as they had the "best" performance for that dataset. Then, surrogate models with performance metric values within 1% of those best values were also assigned as "recommended." Any surrogate models with metric values outside of the 1% range was assigned a recommendation class of "not recommended."

The built-in feature selection method of random forest models was performed to determine which attributes had the most influence on the predicted recommendation class for each surrogate modeling technique. Input features are assigned an importance value in random forest models based on how much they reduce the Gini impurity (Menze et al., 2009) at each decision node in the forest. The Gini impurity measures how well the decision threshold separates the training

samples into the two classes at a particular node (Menze et al., 2009). The feature importances of all the input features (in this case, the attributes) sum to 100%. Attributes were ranked from highest to lowest feature importance. The attributes were added to the input feature set starting with the highest importance up to a sum of 90% of the total importance to reduce the attribute set for the classification model for each surrogate modeling technique. Here, the goal is to consider the attributes with the highest impact on the classification model outcome in the random forest classifier model. The remaining features in the lower 10% of the importance sum were discarded. Figure 2 summarizes the PRESTO construction steps. PRESTO is available for testing in the Cremaschi research group GitHub repository (https://github.com/CremaschiLab/PRESTO).

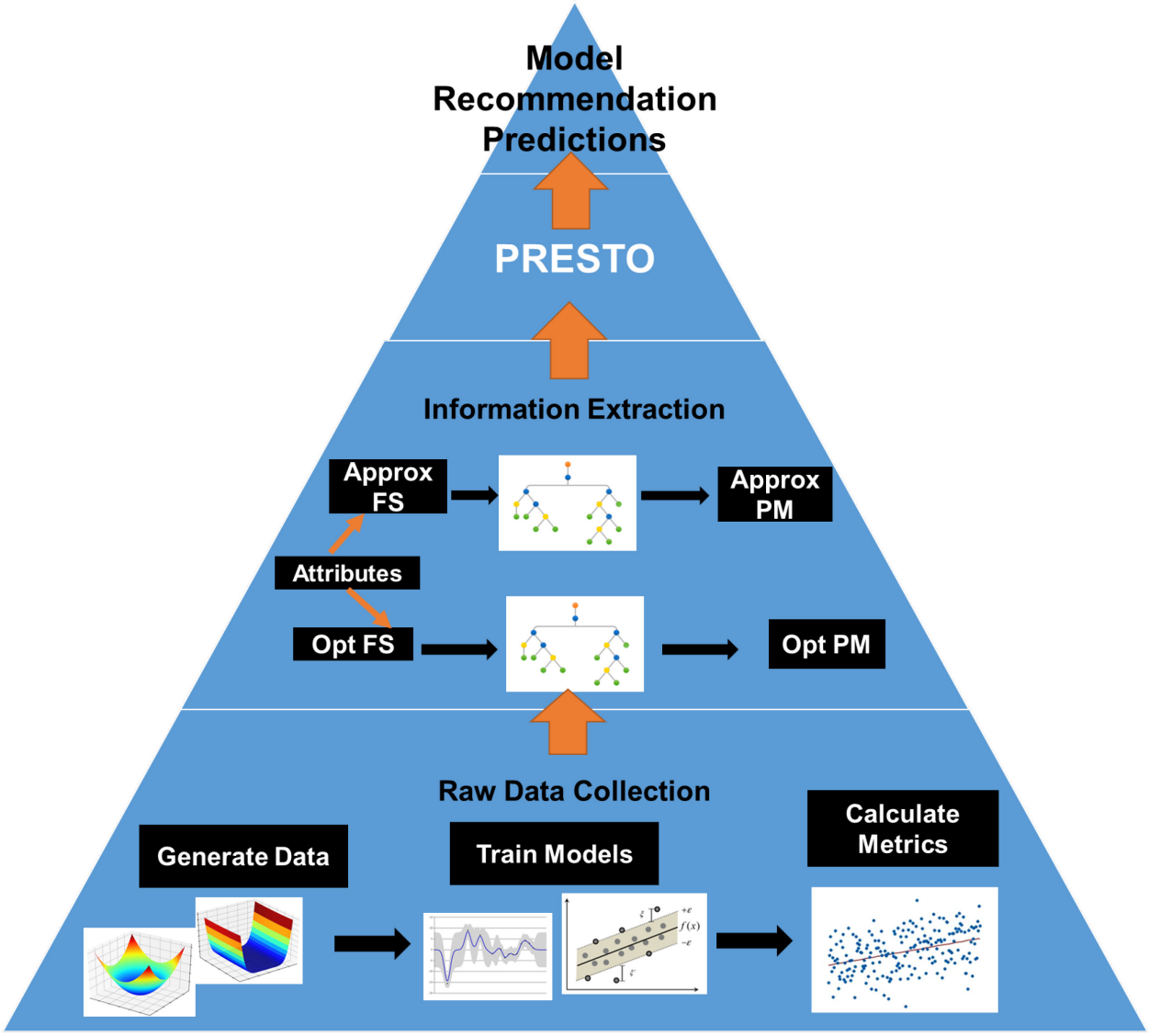

Figure 2. Summary of PRESTO construction (FS = Feature Selection, Approx = Surface Approximation, Opt = Surrogate-based optimization, PM = Classification Performance Metrics)

4.2 PRESTO Performance Evaluation Criteria

The performance of PRESTO was evaluated using three performance metrics: accuracy, precision, and hit ratio. These metrics are calculated based on the classification confusion matrix

(Sokolova and Lapalme, 2009) (Figure 3), which describes the quality of classifications made by a classification model.

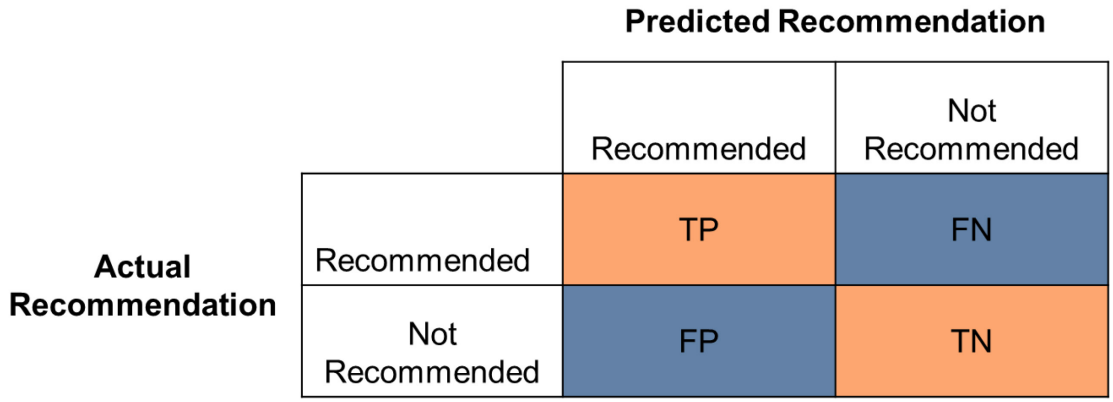

Figure 3. Classification confusion matrix (TP = true positive, TN = true negative, FP = false positive, FN = false negative).

The accuracy (Eq. 28) measures the percentage of recommendation classifications made by PRESTO that is correct. The precision (Eq. 29) is the probability that a model classified as recommended should actually be recommended and will perform well for a dataset. The hit ratio (Eq. 30) is the percentage of models that will perform well for a dataset that PRESTO is recommending.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{28}$$

$$Precision = \frac{TP}{TP + FP} \tag{29}$$

$$Hit\ ratio = \frac{TP}{TP + FN} \tag{30}$$

Figure 4 provides a flowchart depicting how the performance metrics were calculated for evaluating PRESTO's performance. The first step is similar to how a user would use PRESTO, where input-output data is generated and passed to PRESTO. PRESTO then provides recommendations for which surrogate modeling techniques to employ. For the second step in the analysis, all eight of the candidate surrogate modeling techniques were used to train models, and their performance metrics were calculated. These metrics of actual performance were compared to the recommendations for models PRESTO predicted would perform well for the data to evaluate the quality of PRESTO's recommendations. Training all of the models as was done in this analysis is not necessary for a PRESTO user. In practice, a user could train as few or as many of the recommended models as desired.

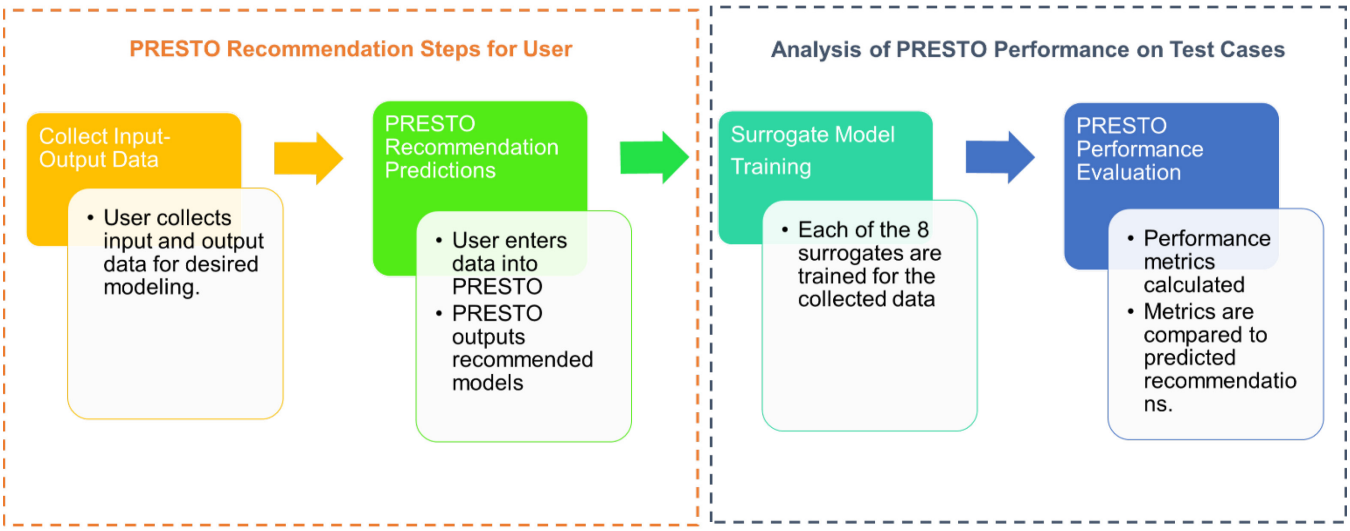

Figure 4. Summary of PRESTO performance evaluation.

5. PRESTO Model Selection Performance Results

5.1 Application Dependent PRESTO Attributes for Surrogate Modeling Techniques

The numbers of attributes selected for classifying the eight candidate surrogate modeling techniques as being recommended or not recommended for surface approximation and surrogate-based optimization are given in Table 1. For example, for surface approximation, 21 attributes out of the initial set of 38 were selected as inputs for the classifier trained to make predictions regarding ALAMO. Based on these results, the random forest classifier required approximately 39 - 55% of the attributes for making recommendations. There were no significant differences between the number of attributes selected for each surrogate modeling technique or application.

Table 1. Number of attributes selected for recommendation predictions

	Attributes Selected	
	Surface Approximation	Surrogate-Based Optimization
ALAMO	21	21
ANN	20	19
ELM	19	20
GPR	21	20
MARS	21	20
RBFN	20	20
RF	15	21
SVR	21	21

Tables 2 and 3 list the five highest important attributes selected for surface approximation and surrogate-based optimization, respectively. Attributes related to the dataset inputs, including the average, minimum, and maximum Mahalanobis distances between input data points, were selected frequently for the majority of the surrogate modeling techniques for both surface approximation and surrogate-based optimization performance predictions. Other commonly selected features include those related to the distributions of output values, specifically the relative

size of the output distribution tails and the output distribution skewness and kurtosis. These results suggest that the distribution and location of the sample points and the relative steepness and smoothness of the surface have a high level of influence on how well each of the surrogate models can approximate that surface and locate the optimum of the underlying function.

Attributes related to the distributions of the input and output locations were commonly selected among all of the candidate techniques. These attributes may affect surface approximation performance as having data unevenly concentrated (or sparse) at the extreme values may skew models to predict more accurately in areas of data concentration and less so for other areas of the design space. For example, in the case of RF models, uneven tails could cause decision nodes in the model trees to split more frequently at the extremes of the output values while more finely split partitions were really needed elsewhere, such as where the gradients were steep. For the neural network-based models, the on-off nature of the hidden layer nodes may make them more suitable for making accurate predictions for surfaces where large areas of the design space have similar output values, creating flat or nearly flat areas, similar to the *plate*-shaped functions.

For surface approximation, the attributes of the empirical mean of fractional local fluctuations and the empirical standard deviation in fractional local fluctuations were also frequently selected in the top five attributes. These attributes measure the average bumpiness and non-linearity variations, respectively, of the surface being modeled (Garud et al., 2018) and can be considered to give a measure of the noisiness of the surface. The noise level has a significant effect on some models' ability to fit a surface. For example, for SVR model performance, the support vectors fitted in the model construction can easily become sensitive to noise as they are only dependent on a small set of the data used to train the model (Sabzekar et al., 2011).

Table 2. Five highest important attributes selected for surface approximation

ALAMO	
Attribute	Importance
Euclidean to Mahalanobis ratio	11.1%
Skewness of outputs	8.7%
Kurtosis of outputs	7.2%
Upper tail average	5.8%
Coefficient of variation of outputs	5.4%

ANN	
Attribute	Importance
Coefficient of variation of outputs	10.4%
Upper tail average	9.3%
Average Mahalanobis distance	6.6%
Average gradient cosine direction	6.0%
Kurtosis of outputs	5.5%

ELM	
Attribute	Importance
Average Mahalanobis distance	12.2%
Minimum Mahalanobis distance	12.0%
Input dimensions	9.7%
Empirical standard deviation in fractional local fluctuations	8.9%
Kurtosis of outputs	6.6%

GPR	
Attribute	Importance
Upper tail average	7.6%
Coefficient of variation of outputs	7.6%
Empirical mean of fractional local fluctuations	6.4%
Skewness of outputs	6.3%
Empirical standard deviation in fractional local fluctuations	5.7%

MARS	
Attribute	Importance
Kurtosis of outputs	11.5%
Empirical standard deviation in fractional local fluctuations	7.6%
Skewness of outputs	7.1%
Relative size of upper tail	6.7%
Upper tail average	5.0%

RBFN	
Attribute	Importance
Average Mahalanobis distance	10.4%
Minimum Mahalanobis distance	9.2%
Empirical mean of fractional local fluctuations	7.0%
Empirical standard deviation in fractional local fluctuations	7.0%
Skewness of outputs	5.5%

RF	
Attribute	Importance
Empirical standard deviation in fractional local fluctuations	21.3%
Empirical mean of fractional local fluctuations	15.6%
Coefficient of variation of gradient magnitudes	7.3%
Average Mahalanobis distance	7.1%
Minimum Mahalanobis distance	7.0%

SVR	
Attribute	Importance
Skewness of outputs	8.0%
Empirical mean of fractional local fluctuations	7.7%
Kurtosis of outputs	6.3%
Empirical standard deviation in fractional local fluctuations	5.7%
Skewness of gradient magnitudes	5.1%

Table 3. Five highest important attributes selected for surrogate-based optimization

ALAMO	
Attribute	Importance
Upper tail average	9.3%
Coefficient of variation of outputs	9.2%
Skewness of outputs	8.0%
Lower tail relative size	6.2%
Average Mahalanobis distance	5.8%

ANN	
Attribute	Importance
Input dimensions	12.3%
Average Mahalanobis distance	12.1%
Maximum Mahalanobis distance	9.3%
Minimum Mahalanobis distance	7.7%
Kurtosis of outputs	7.7%

ELM	
Attribute	Importance
Average Mahalanobis distance	18.6%
Maximum Mahalanobis distance	12.9%
Minimum Mahalanobis distance	10.7%
Input dimensions	10.0%
Coefficient of variation of outputs	4.4%

GPR	
Attribute	Importance
Input dimensions	9.5%
Average Mahalanobis distance	9.1%
Coefficient of variation of outputs	7.0%
Maximum Mahalanobis distance	6.6%
Skewness of outputs	6.5%

MARS	
Attribute	Importance
Input dimensions	13.8%
Average Mahalanobis distance	13.0%
Minimum Mahalanobis distance	8.0%
Maximum Mahalanobis distance	6.2%
Kurtosis of outputs	5.2%

RBFN	
Attribute	Importance
Minimum Mahalanobis distance	13.9%
Average Mahalanobis distance	10.4%
Input dimensions	10.3%
Maximum Mahalanobis distance	6.6%
Skewness of outputs	4.8%

RF	
Attribute	Importance
Average Mahalanobis distance	8.8%
Minimum Mahalanobis distance	6.4%
Input dimensions	6.3%
Maximum Mahalanobis distance	5.7%
Euclidean to Mahalanobis ratio	5.6%

SVR	
Attribute	Importance
Skewness of outputs	11.0%
Upper tail average	8.8%
Coefficient of variation of outputs	8.0%
Average Mahalanobis distance	5.3%
Kurtosis of outputs	5.2%

For surrogate-based optimization, although not selected as one of the top five important attributes, the average local convex deviation, and attributes related to the estimated gradients were selected frequently in the set of important attributes. The convexity of a model has a significant

effect on the relative “ease” of finding its global minimum. Gradient information is also crucial in the application of gradient-based methods for optimization. The complete listing of all attributes selected is available in the supplemental materials of the paper.

5.2 PRESTO Recommendation Classification Results

The selected attributes were used as inputs to train random forest classification models for the eight techniques to classify each technique as being “recommended” or “not recommended” for a given dataset. Separate classifiers were trained for surface approximation and surrogate-based optimization. This recommendation scheme allows for multiple similarly performing surrogate modeling techniques to be suggested for use. The performance metrics were calculated using Monte Carlo cross-validation with 75 Monte Carlo trials. Each trial had a test set size of 20% of the total dataset.

PRESTO identified which techniques should be recommended for the simulated datasets for surface approximation with an accuracy of 91%. The precision, or the probability that a recommended technique should actually be recommended, was 90%. The hit ratio, the percentage of the surrogate models that should have been recommended for surface approximation that PRESTO successfully captured, was 87%.

For surrogate-based optimization, PRESTO recommended surrogate modeling techniques with an accuracy of 98% and a precision of 99%. This result indicates that if PRESTO identifies a model as being recommended for a dataset, there is a 99% probability that the model will accurately locate the global optimum of the underlying model for the dataset. The hit ratio for surrogate-based optimization was 98%.

6. Performance Evaluation of PRESTO for a Chemical Engineering Application - Cumene Production Case Study

A simulation model of the cumene production process was employed to test the performance of PRESTO's recommendations for a chemical engineering application and on datasets that were not used for its training. The entire process for cumene production was simulated in gPROMS Process. Input-output datasets were generated for a subset of the unit operations in the flowsheet, using the gPROMS Global System Analysis capabilities. PRESTO was used to provide surrogate modeling technique recommendations to predict each output for surface approximation. Then, surrogate models were trained for each output using the eight candidate surrogate modeling techniques, and the corresponding adjusted R-squared values were calculated using Monte Carlo cross-validation (Xu et al., 2004) with a test set size of 20% of the dataset and 50 Monte Carlo trials. The adjusted R-squared results were compared to the recommendation classifications made by PRESTO for surface approximation.

6.1 Process and Simulation Description for Cumene Production

The process to produce cumene, a petrochemical used in the production of several chemicals, involves the reaction of benzene with propylene to form cumene and the undesirable reaction of cumene with propylene to form p-diisopropyl benzene (PDIB) (Luyben, 2011). The flowsheet for the process is given in Figure 5. The case study focuses on the cooled tubular reactor (Reactor) and two distillation columns (C1 and C2). These are the three most complex unit-level models in the overall flowsheet, meaning that replacing them with surrogate models will have the greatest impact on improving computational speed. In the process, the liquid fresh feed streams are combined with a benzene recycle stream, vaporized, preheated to 360 °C, and fed to the cooled tubular reactor. The first distillation column, C1, produces a mostly benzene distillate, which is

recycled back to the reactor. This recycle stream is necessary to keep benzene from exiting in the bottoms product and affecting the purity of the cumene product in the distillate of the second column, C2. Column C2 is designed to attain high-purity cumene in the distillate and minimize the loss of cumene in the bottoms (Luyben, 2011).

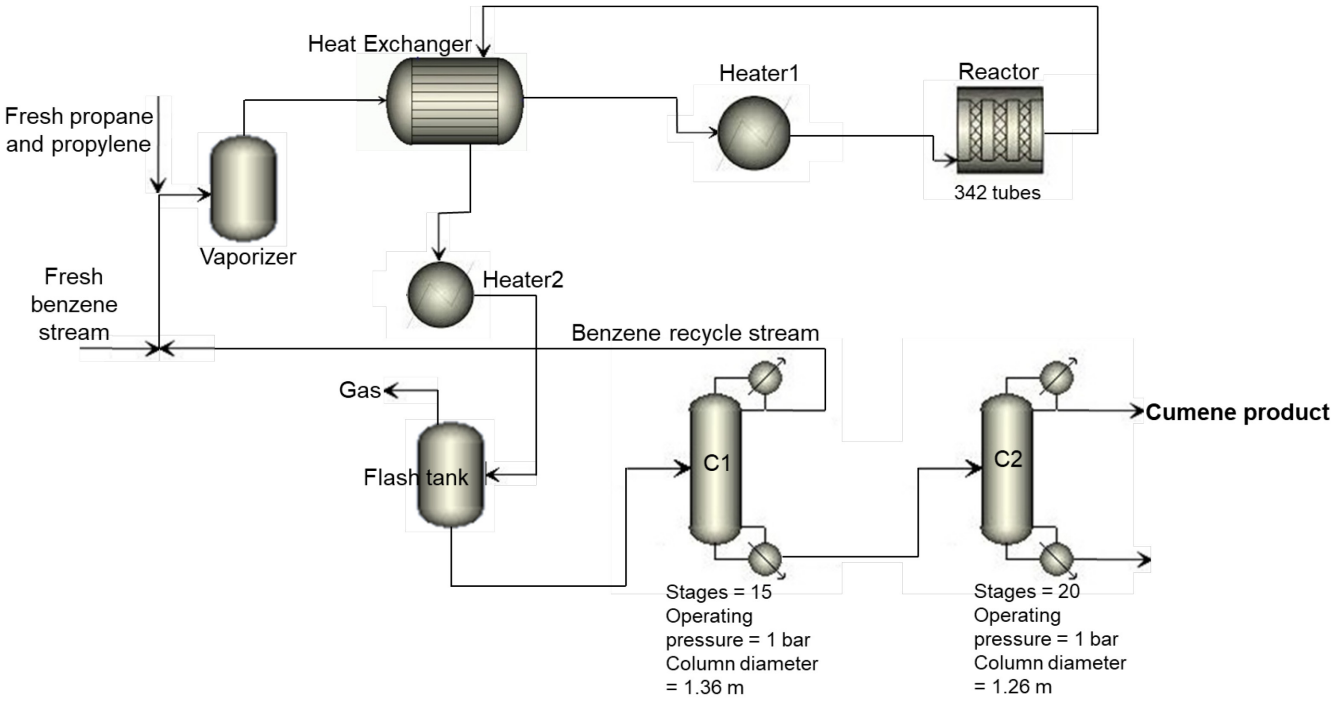

Figure 5. Flowsheet for cumene production case study.

The entire process was simulated in gPROMS Process, where the data for the three complex unit operations was generated. Data was generated for a total of 27 outputs, with seven outputs for the reactor, 11 outputs for the first distillation column (C1), and nine outputs for the second distillation column (C2). Outputs for the distillation columns include heat duties and top and bottoms product compositions. Outputs for the reactor include outlet temperatures and reaction product compositions. For each output, data was randomly selected at 4 different sample sizes (100, 300, 1000, and 3000 samples) for a total of 108 case study datasets for evaluating PRESTO's performance. The input values for each dataset are operating specifications for a fixed design of

the respective unit operations, such as inlet temperatures and inlet compositions. All inputs and outputs for each dataset and all the case study data are provided in the supplementary materials.

6.2 PRESTO Performance

Table 4 lists the performance metrics for PRESTO on the case study data compared to the metrics for the data that PRESTO was trained on, or the PRESTO data. Performance metrics were calculated for both to compare how well PRESTO performed on the simulated data from the test functions to how it performed on data from a real-world application. PRESTO's performance on the cumene case study data was similar to that of the tool training data for accuracy and precision. However, the hit ratio was slightly lower. The lower hit ratio for the case study data indicates that PRESTO was not identifying all the possible models that could be recommended, only a subset of them, which also resulted in a higher precision. These results suggest that PRESTO may successfully identify a set of surrogate models that will perform well for approximating the behavior of a data set for some relevant cases without the need for expensive trial-and-error methods.

The lower hit ratio may be due to the fact that the values for the Mahalanobis distances between the data points for the case study data were outside the ranges of the distances for the data that the tool was trained on. For example, the maximum value for the PRESTO data for the maximum Mahalanobis distance between data points is approximately 6.7, while both the average and maximum value of that same attribute for the case study data are higher at 6.8 and 7.1, respectively. The simulated data was generated using the same space-filling method, while the inputs for the case study data were generated randomly. We can conclude that the position of the sample points, or the distances between them, are critical in providing accurate recommendations as features related to the Mahalanobis distances were selected as important for almost all of the

classification models that were trained. The performance of PRESTO could be improved by adding datasets that use a variety of sampling methods in order to provide better ranges for the attributes related to data point distribution.

In addition, all of the datasets in the PRESTO data were created using relatively smooth, continuous functions. Data from real applications may not share the same characteristics. An example of this difference in behavior can be seen in the average local convex deviation attribute. The average value for this attribute is 2.66×10^{6} for the PRESTO data and several orders of magnitude higher for the case study data at 9.1×10^{12}. The recommendations of PRESTO for real data could be enhanced by the addition of real datasets to the PRESTO training data. However, with a 94% precision for the case study data, PRESTO's predictions for which surrogate models to use are still accurate. All of the compiled results for adjusted R-squared and recommended models for the case study data are available in the supplementary materials.

Table 4. PRESTO case study performance comparison

	Case Study Data	PRESTO Data
Accuracy	89%	91%
Precision	94%	90%
Hit Ratio	76%	87%

PRESTO did not recommend any candidate surrogate models for two process outputs: the bottom product temperature and top liquid recovery of cumene. These classifications of "not recommended" for all the surrogates were correct, as when the models were trained, none of the techniques could successfully approximate these outputs with an adjusted R-squared above 0.7. Our previous work demonstrates that there are some test functions that were used to train PRESTO,

for which none of the surrogate modeling techniques were able to approximate the surface (Williams and Cremaschi, 2021). In these instances, alternative modeling strategies, such as ensemble modeling, deep learning algorithms, or another surrogate modeling technique not included in the candidate set, may be considered. It should be noted that when PRESTO does not recommend any surrogate modeling techniques, it could also indicate that the current data set size is too small for them to model the input-output relationship accurately. We observed that for some datasets, increasing its size also increased the number of recommended models. Hence, we also recommend increasing the dataset size and re-running PRESTO. The case study results reveal that PRESTO can capture the qualities of datasets that would make them unsuitable for modeling with the eight candidate techniques studied.

7. Conclusions and Future Work

Selecting an appropriate surrogate modeling technique depends on the characteristics of the dataset being modeled and the application domain of the surrogate model, surface approximation vs. optimization. We identified attributes of datasets appropriate for selecting surrogate models for both surface approximation and surrogate-based optimization. Using these attributes, a recommendation tool, PRESTO, was constructed to recommend surrogate modeling techniques for approximating a dataset with 91% accuracy and 90% precision and for performing surrogate based-optimization with 98% accuracy and 99% precision. Although PRESTO could not capture the full set of models that could be recommended on a set of test data generated from a cumene production process simulation, the recommended models did provide higher values of adjusted R-squared and better predictions for the outputs. Future work on PRESTO will include adding more real datasets to the training data for the tool, focusing on using a wider variety of sampling methods, not just space-filling ones, and incorporating the impact of noisy data.

Acknowledgments

This work was partially funded by 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

Breiman, L., 2001. Random forests. Machine Learning 45, 5-32.

Cozad, A., Sahinidis, N.V., Miller, D.C., 2014. Learning surrogate models for simulation-based optimization. Aiche Journal 60, 2211-2227.

Cui, C., Hu, M.Q., Weir, J.D., Wu, T., 2016. A recommendation system for meta-modeling: A meta-learning based approach. Expert Systems with Applications 46, 33-44.

De Maesschalck, R., Jouan-Rimbaud, D., Massart, D.L., 2000. The Mahalanobis distance. Chemometrics and Intelligent Laboratory Systems 50, 1-18.

Drucker, H., Shahrary, B., Gibbon, D.C., 2002. Support vector machines: relevance feedback and information retrieval. Information Processing & Management 38, 305-323.

Friedman, J.H., 1991. Multivariate Adaptive Regression Splines - Rejoinder. Annals of Statistics 19, 123-141.

Garud, S.S., Karimi, I.A., Kraft, M., 2018. LEAPS2: Learning based Evolutionary Assistive Paradigm for Surrogate Selection. Computers & Chemical Engineering 119, 352-370.

Gomm, J.B., Yu, D.L., 2000. Selecting radial basis function network centers with recursive orthogonal least squares training. Ieee Transactions on Neural Networks 11, 306-314.

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.

Hart, W.E., Laird, C.D., Watson, J.-P., Woodruff, D., L., Hackebail, G.A., Nicholson, B.L., Siirola, J.D., 2017. Pyomo - Optimization Modeling in Python, 2 ed. Springer, Boston, MA.

Hart, W.E., Watson, J.-P., Woodruff, D.L., 2011. Pyomo: modeling and solving mathematical programs in Python. Mathematical Programming Computation 3, 219 - 260.

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.

Joe, S., Kuo, F.Y., 2008. Constructing Sobol' Sequences with Better Two-Dimensional Projections. Siam Journal on Scientific Computing 30, 2635-2654.

Luyben, W., 2011. Design and Control of the Cumene Process, Principles and Case Studies of Simultaneous Design. John Wiley & Sons, Inc, Hoboken, NJ, pp. 135-158.

Mehmani, A., Chowdhury, S., Meinrenken, C., Messac, A., 2018. Concurrent surrogate model selection (COSMOS): optimizing model type, kernel function, and hyper-parameters. Structural and Multidisciplinary Optimization 57, 1093-1114.

Menze, B.H., Kelm, B.M., Masuch, R., Himmelreich, U., Bachert, P., Petrich, W., Hamprecht, F.A., 2009. A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. Bmc Bioinformatics 10.

Rasmussen, C.E., Williams, C.K.I., 2005. Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning, 1-247.

Sabzekar, M., Yazdi, H.S., Naghibzadeh, M., 2011. Relaxed constraints support vector machines for noisy data. Neural Computing & Applications 20, 671-685.

Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. Information Processing & Management 45, 427-437.

Surjanovic, S., Bingham, D., 2013. Virtual Library of Simulation Experiments, Simon Fraser University.

Vilalta, R., Drissi, Y., 2002. A perspective view and survey of meta-learning. Artificial Intelligence Review 18, 77-95.

Williams, B., Cremaschi, S., 2021. Selection of surrogate modeling techniques for surface approximation and surrogate-based optimization. Chemical Engineering Research & Design 170, 76-89.

Xu, Q.S., Liang, Y.Z., Du, Y.P., 2004. Monte Carlo cross-validation for selecting a model and estimating the prediction error in multivariate calibration. Journal of Chemometrics 18, 112-120.