SURROGATE MODEL SELECTION FOR DESIGN SPACE APPROXIMATION AND SURROGATE-BASED OPTIMIZATION

B.A. Williams and S. Cremaschi*
Auburn University
Auburn, AL 36849

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 sensitivity analysis, uncertainty propagation and surrogate based optimization. This work evaluates the performance of eight surrogate modeling techniques for design space approximation and surrogate based optimization 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 solely based on the characteristics of the data being modeled. The computational experiments revealed that, in general, multivariate adaptive regression spline models (MARS) and single hidden layer feed forward neural networks (ANN) yielded the most accurate predictions over the design space while Random Forest (RF) models most reliably identified the locations of the optimums when used for surrogate-based optimization.

Keywords

surrogate model, optimization, design space approximation, multivariate adaptive regression splines, random forests, artificial neural networks

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 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 relationship between input data and output data does not exist or is not conducive for use in traditional gradient based optimization methods. Some recent examples of applications of surrogate modeling approaches include several process synthesis applications, for example, in optimization of carbon fiber production plant energy consumption (Golkaranarenji et al., 2018), and process controls applications in the pharmaceutical production industry (Icten et al., 2015).

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 best select the appropriate model for a particular application for either design space approximation or optimization. For studies applying surrogate modeling techniques for process design and optimization, models are mostly selected using process specific expertise with no systematic basis for the selection.

The majority of studies comparing surrogate model performance only compare a few models on a limited

* To whom all correspondence should be addressed

number of functions or applications (Davis et al., 2017; Bhosekar and Ierapetritou, 2018). Efforts have been made to generalize the process for selecting a surrogate model to approximate a design space by using meta-learning approaches to build selection frameworks (Garud et al., 2018; Cui et al., 2016). However, these frameworks provide little insight into selecting surrogates for optimization purposes. This work aims to address the knowledge gap by comparing the performance of eight different surrogate modeling techniques for two applications of surrogate models: design space approximation, which attempts to model the overall behavior of the dataset, and surrogate based optimization. Data sets for training surrogate models are generated from a large set of test functions with different characteristics, such as function shape and number of inputs, using two sampling methods. The effects of the function characteristics and sampling methods on the surrogate model performance are evaluated. The goal of performing this analysis is to develop general "rules of thumb" for selecting an appropriate surrogate modeling form based on the characteristics of the data being modeled and the desired application. The following sections contain brief descriptions of the surrogate modeling techniques used and the test function sets. Then the design of computational experiments and the results are presented.

Surrogate Modeling Techniques

Multivariate adaptive regression spline 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).

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 (Brieman, 2001).

Single hidden-layer feed forward artificial neural networks (ANNs) 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).

An extreme learning machine is an ANN where 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).

A radial basis function network (RBFNs) is an ANN with a radial basis function as the activation function in the hidden layer. The network calculates the Euclidean distance between the input weights and input values and passes those distances through the radial basis activation function (Jin et al. 2001).

Support vector machines (SVMs) transform input data into m-dimensional space and construct a set of hyperplanes such that the distance from a hyperplane to the nearest data point on each side of the plane is maximized using kernel functions (Jin et al., 2001).

Gaussian process regression (GP) uses a linear combination of inputs to predict output values. It uses a kernel function as measure of similarity between points to predict the value for an unseen point (Mirabagheri, 2001).

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, and trigonometric functions (Cozad et al., 2014). It should be noted that the adaptive sampling scheme of ALAMO is not used in this study.

Test Functions

The test functions used are the optimization set from Virtual Library of Simulation Experiments (Surjanovic and Bingham, 2013). The functions are divided by shape, which include the categories: multi-local minima with 31 functions, bowl-shaped with 31 functions, plate-shaped with 9 functions, valley-shaped with 12 functions, and other-shaped with 18 functions that do not fit into the other four categories. Functions with two, four, six, eight, and 10 inputs were used in evaluations.

Computational Experiments

For evaluating the performances of surrogate modelling techniques, 1000 input-output pairs were generated from each test function using two different sampling methods, and surrogate models were trained using these pairs with each of the surrogate modeling technique for each function. This resulted in 1616 surrogate models.

Each of the techniques has unique hyperparameters that were optimized in training the models for each dataset. For the MARS models, the number of hinge functions that could be multiplied together was limited to two. RF models grew unpruned without restrictions. The number of ANN and ELM nodes was increased until the root mean squared error of the validation dataset started to increase.

The two sampling methods used were Latin Hypercube Sampling (LHS) and Sobol Sequence. The LHS splits the domain of each input variable into N subsets, where N is the number of sampling points. The subsets are then sampled randomly to produce the input values (McKay, 1992). The Sobol sequence attempts to distribute the sampling points

uniformly across the input space. It is a quasi-random, low-discrepancy sequence (Joe and Kuo, 2003).

After the surrogate models were trained for each dataset and sampling method, 100,000 input-output pairs were generated using the Sobol sequence sampling to test the accuracy of the models. The root mean squared error (RMSE) and the maximum percent error (MaxAPE) 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 Akaike Information Criterion (AIC) and Akaike weights were calculated for each modeling technique (Akaike, 1973).

The global minima of each test function was estimated using the trained surrogate models. The mathematical programs were constructed in Pyomo, a Python based optimization language. The estimated minima were compared to the actual global minima of the test functions for accuracy to provide some insight into the effectiveness of the surrogate models for surrogate based optimization. The solvers used for optimization are provided in Table 1. When local solvers were used for optimization, a multi-start approach was used with 25 starts from different locations in the domain space. Solvers were chosen for each technique based on which provided the best solutions in the shortest time. Computations were carried out on the Auburn University Hopper HPC Cluster (Lenovo System X HPC Cluster) using 12 Intel E5-2650 V3, 2.3 GHz 20 core processors and implemented in Python 3.5 and MATLAB 2017b (for RBFN surrogate models).

Table 1. Solvers for surrogate based optimization

Surrogate Model	Resulting Optimization Model	Solver
MARS	MINLP	ANTIGONE
RF	MILP	CPLEX
ANN	NLP	CONOPT
ELM	NLP	CONOPT
GP	NLP	COUENNE
SVM	NLP	COUENNE
ALAMO	NLP	BARON
RBFN	NLP	BARON

Performance Metrics

The RMSE and MaxAPE values for each dataset-surrogate model combination were normalized by the range of output values for easier comparison across datasets with a variety of ranges for output values.

Surrogate Model Selection by AIC

The formula for AIC is shown in Eq. 1.

$$AIC = 2k + n\ln\frac{SSE}{n} \tag{1}$$

where k is the number of parameters in the model, n is the number of training points and *SSE* is the sum of the squared errors. The AIC is used to calculate the ΔAIC_i (Eq. 2) and Akaike weight (w_i) (Eq. 3), which is the probability that model i is the best one over a set of models for a dataset. This metric considers the model accuracy and complexity. Increased model complexity can lead to both overfitting to the dataset and increased computational time requirements.

$$\Delta AIC_i = AIC_i - min(AIC) \tag{2}$$

$$w_i = \frac{exp(-0.5\Delta AIC_i)}{\sum_j exp(-0.5\Delta AIC_j)} \tag{3}$$

Surrogate-Based Optimization Performance Metrics

We define D_{opt} as the Mahalanobis distance, D_M, (McLachlan, 1999) between the location of the global minimum of the function, x_{opt}, and the location estimated using the surrogate-based optimization, $x_{opt'}$. This value is normalized by the maximum Mahalanobis distance between any two points (x_i, x_j) in the dataset (Eq. 4).

$$D_{opt} = \frac{D_M\left(x_{opt}, x_{opt'}\right)}{\max_{i,j} D_M(x_i, x_j)} \tag{4}$$

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

We define G_{opt}, Eq. 5, 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.

$$G_{opt} = \frac{y_{opt} - y_{opt'}}{y_{max} - y_{min}} \tag{5}$$

where y_{opt} is the actual minimum value, $y_{opt'}$ is the one calculated by the surrogate model, and y_{max} and y_{min} are the maximum and minimum output values of the dataset.

Results and Discussion

With the training set size of 1000 sample points, there was no significant difference in the performance of the surrogate models trained using the points generated using Sobol sequence and LHS. Therefore, results presented in this section only include surrogate models trained with datasets generated via LHS.

Design Space Approximation Performance

Results obtained based on the Akaike weights are summarized in Fig. 1. The weights were used to take into account the model size and complexity in addition to its accuracy (Akaike, 1973). Akaike weights were calculated for all of the surrogate modeling techniques for each dataset, and the percentage of the time each surrogate model

was selected as the best model by Akaike weight was tabulated, which was used to calculate the fraction of a surrogate modeling technique being the best among the datasets (Fig. 1). The number of datasets included in each category is included below the *x*-axis for Figs. 1b and 1c.

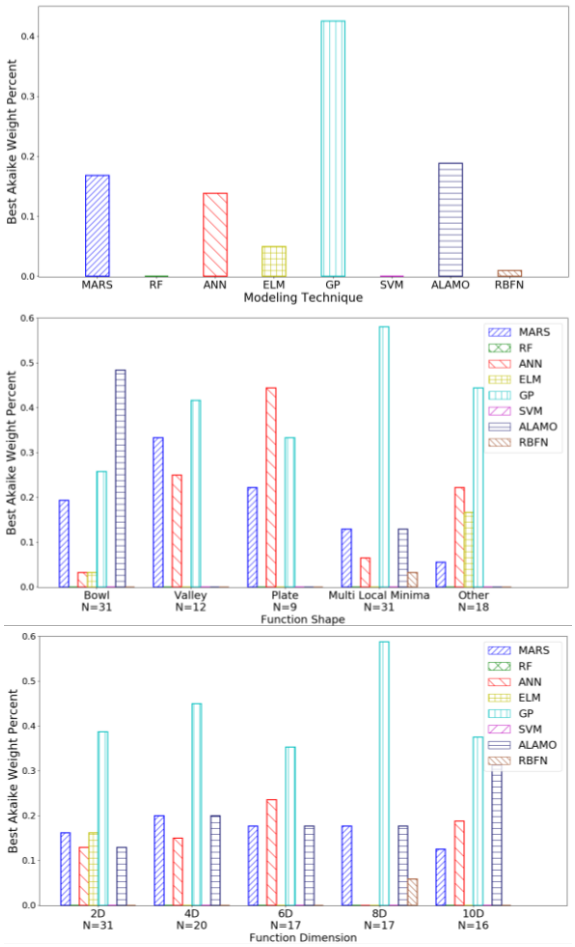

Figure 1. (a) Percentage of datasets for which each model has the highest Akaike weight, Percentage of best weights grouped by (b)function shape, and (c) input dimension

Figure 1a shows that GP models were selected as the best model the highest percentage of the time for all datasets when they are all grouped together. Figure 1b shows slightly different results when the datasets are grouped by shape, with GP being selected as the best the most often for three out of the five shape categories. For bowl-shaped functions, ALAMO models were selected as the best most frequently, while for plate-shaped functions ANN models were selected as the best most frequently. This result indicates that there is some dependence of the surrogate model performance on the overall shape of the function the dataset was generated from. When the datasets are grouped by dimension, however, GP models are selected as the best most frequently across all of the dimensions tested, indicating less of a dependence on AIC performance on input dimension of the dataset. While GP models had the most robust AIC performance, RF and SVM models did not perform the best for any of the datasets considered indicating that if AIC is the performance metric of interest, these models are not suitable choices.

The RMSE and MaxAPE were calculated using the 100,000 sample point test sets to investigate how well the surrogate models approximated the actual function surfaces. Results for these performance metrics are shown in Figs. 2 and 3 in boxplot format. For each box, the bottom, middle, and top lines of the box represent the 25th, 50th, and 75th percentiles, respectively. Outliers are plotted as individual points. In contrast to the AIC performance, no model was the worst performing for the RMSE and MaxAPE for all of the considered datasets.

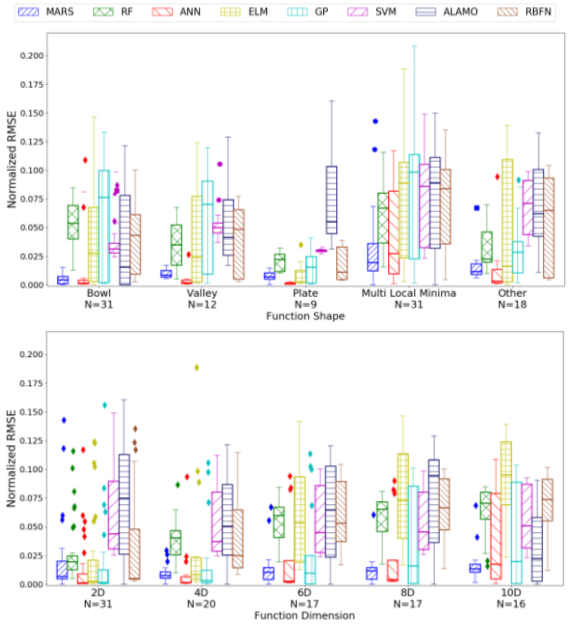

Figure 2. RMSE for datasets grouped (a) by function shape, (b) by input dimension

The RMSE and MaxAPE exhibit similar trends with MARS models having the most robust performance with respect to both function shape and input dimension. The ANN models also exhibit similarly accurate performance. The difference between the suggested models by the Akaike weights and these metrics may be due to the differences in resulting surrogate model complexities. While MARS and ANN models tend to get larger as the function shape complexity and input dimension increase, GP model size tends to remain constant. Because AIC takes model complexity into consideration, it favors GP models more. 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 products of hinge functions.

While MARS and ANN models perform well for each shape and dimension investigated, the performances of other models change with different function characteristics. GP and ELM models have performances similar to MARS and ANN at low input dimensions, but their performances worsen as the dimension increases, for example, further

illustrating the dependence of model performance on dimension. These results suggest that for datasets where specific characteristics are not available a MARS or ANN model would be appropriate to select as a general guideline. However, if characteristics are available, other models might provide a better design space approximation.

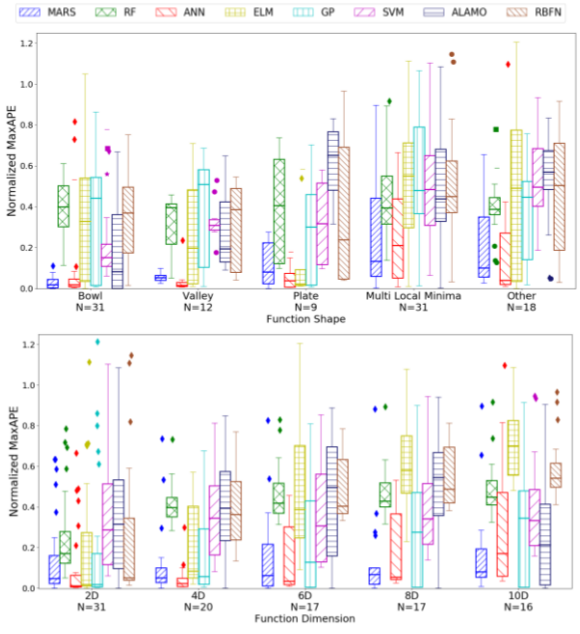

Figure 3. MaxAPE for datasets grouped (a) by function shape, (b) by input dimension

Surrogate Based Optimization Performance

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. 4) and G_{opt} (Eq. 5). Results are summarized in Figs. 4 and 5, where we define a model as having located the optimum when it obtains a D_{opt} or G_{opt} value less than 1%. The numbers in parentheses (in x-axis) next to the number of datasets in each category represent the number of datasets for which a solution was obtained using RF models because some of the RF models resulted in mixed integer linear programs (MILP) that were too large to solve.

Random forest (RF) models in general locate the minima for the highest fraction of the datasets, when datasets are grouped by both shape and dimension. However, for approximating the design space RF models had some of the worst performances, with higher values for both RMSE and MaxAPE. 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 the decision tree nature of RF models. The 'rules' of the decision tree 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 accurately pinpoint the location of the minimum. GP and RBFN models perform most robustly in estimating the actual values of the global minima, in general, with respect to both shape and dimension. This result is in agreement with results from the Akaike weights.

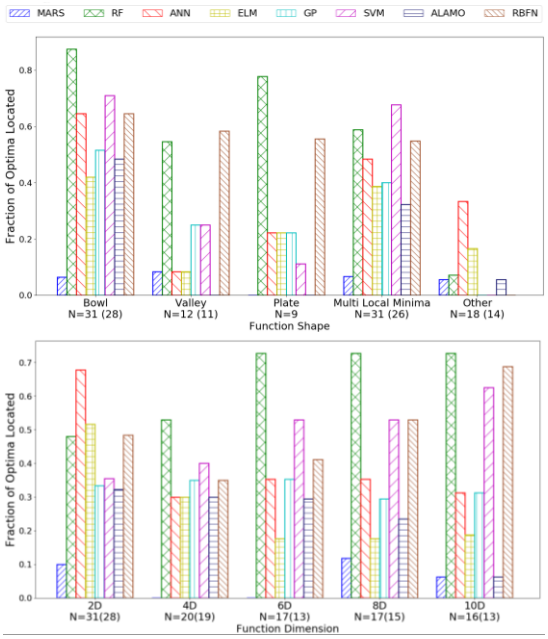

Figure 4. Fraction of datasets with D_{opt} less than 0.1% grouped by (a) function shape, (b) input dimension

Computational Efficiency

The computational time required for training each model, evaluating the test set predictions, and solving the optimization problems are shown in Figure 6. MARS models had some of the lowest times for all three values, which reinforces the suggestion that MARS models would be in general an appropriate selection for a variety of datasets if specific characteristics are not known for design space approximation. RF models have the highest average value for optimization solution times. These solution times may be reduced by developing algorithms that exploit the special structure of RF model MILPs as RF models were successful in pinpointing the location of the minimum.

Conclusions and Future Work

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. However, for general selection rules, MARS and ANN models give the most accurate predictions for design space approximation, and RF, RBFN and GP models give the most accurate estimations for surrogate based approximation. Future work will include consideration of additional dataset characteristics and investigating the

effect of changing training dataset sizes on surrogate model performance.

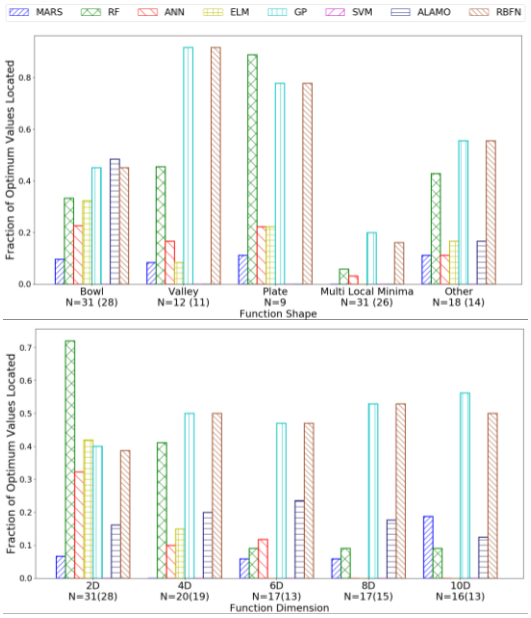

Figure 5. Fraction of datasets with G_{opt} less than 0.1% grouped by (a) function shape, (b) input dimension

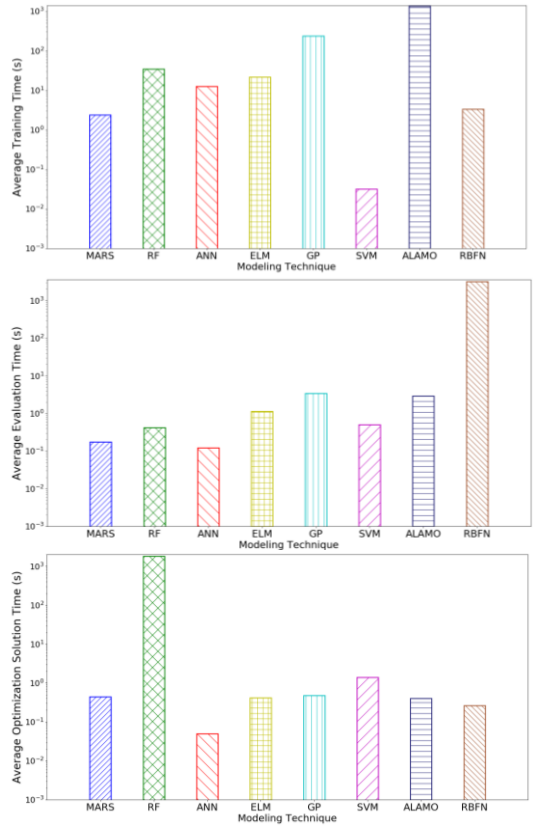

Figure 6: (a) Average model training time in seconds, Average time to (b) evaluate test sets (c) solve optimization problem for global minimum

Acknowledgments

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

References

Akaike, H. (1973). Information Theory and an Extension of the Maximum Likelihood Principle. *Second Intl. Symposium on. Info. Theory*. 1,200.

Bhosekar, A., Ierapertritou, M. (2018). Advances in surrogate based modeling feasibility analysis, and optimization: A review. *Comput. Chem. Eng.*, 108, 250.

Brieman, L. (2001). Random Forests. *Machine Learning*, 45, 5.

Cozad, A. et al. (2014). Learning Surrogate Models for Simulation Based Optimization. *AICHE Journal*, 60, 2211.

Cui, C. et al. (2016). A recommendation system for meta-modeling: A meta-learning based approach. *Expert Systems with Applicatons*, 46, 33.

Davis, S. et al. (2017). Efficient surrogate model development: Optimum model form based on input function characteristics. *Proceedings of the 27th European Symposium on Comp. Aided Process Engineering*, 27.

Friedman, J. (1991). Multivariate Adaptive Regression Splines. *Annals of Statistics,* 19, 1.

Garud, S. et al. (2018). Learning based Evolutionary Assistive Paradigm for Surrogate Selection. *Comput. Chem. Eng.*, 119, 352.

Golkaranarenji, G. et al. (2018). Support vector regression modelling and optimization of energy consumption in carbon fiber production line. *Comput. Chem. Eng.*,109, 276.

Han, Z. and Zhang, K. (2012). Surrogate-Based Optimization. *Real-World Applications of Genetic Algorithms*. InTech Europe, Rijeka, Croatia.

Haykin, S. (2009). Neural Networks and Learning Machines. Prentice Hall PTR.

Huang, G. et al. (2006). Extreme Learning Machine Theory and Applications. *Neurocomputing*, 70, 489.

Icten, E. et al. (2015). Process control of a dropwise additive manufacturing system for pharmaceuticals using polynomial chaos expansion based surrogate model. *Comput. Chem. Eng.*, 83, 221.

Jin, R. et al. (2001). Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria. *Structural and Multidisciplinary Optimization,* 23, 1.

Joe, F. and Kuo,F. (2008). Constructing Sobol' sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing*, 30, 2635.

McKay, M. (1992). Latin hypercube sampling as a tool in uncertainty analysis of computer models. *WSC '92 Proceedings of the 24th conference on Winter simulation,* 557.

McLachlan, G. (1999). Mahalanobis Distance. *Resonance,* 1, 20.

Mirabagheri, S. (2015). Evaluation and Prediction of Membrane Fouling in a Submerged Membrane Bioreactor using Artificial Neural Network-Genetic Algorithm. *Proc. Saf. And Evn. Protection,* 96, 111.

Surjanovic, S., Bingham, D. (2013). "Virtual Library of Simulation Experiments: Test Functions and Datasets. http://www.sfu.ca/~ssurjano.

Wang, C. et al. (2014). An evaluation of adaptive surrogate modeling based optimization with two benchmark problems. *Environ. Modell. Softw.,* 60, 167.