Jie Zhang[1]
Postdoctoral Research Associate
Mem. ASME
Multidisciplinary Design and Optimization Laboratory (MDOL),
Department of Mechanical and Aerospace Engineering,
Syracuse University,
Syracuse, NY, 13244
e-mail: jzhang56@syr.edu

Souma Chowdhury
Assistant Research Professor
Mem. ASME
Department of Mechanical Engineering,
Department of Electrical and Computer Engineering,
Mississippi State University,
Mississippi State, MS 39762
e-mail: souma.chowdhury@msstate.edu

Ali Mehmani
Multidisciplinary Design and Optimization Laboratory (MDOL),
Department of Mechanical and Aerospace Engineering,
Syracuse University,
Syracuse, NY, 13244
e-mail: amehmani@syr.edu

Achille Messac[2]
Dean
Professor
Earnest W. and Mary Ann Deavenport Jr. Endowed Chair,
Fellow ASME
Department of Aerospace Engineering,
Bagley College of Engineering,
Mississippi State University,
Mississippi State, MS 39762
e-mail: messac@bagley.msstate.edu

Characterizing Uncertainty Attributable to Surrogate Models

This paper investigates the characterization of the uncertainty in the prediction of surrogate models. In the practice of engineering, where predictive models are pervasively used, the knowledge of the level of modeling error in any region of the design space is uniquely helpful for design exploration and model improvement. The lack of methods that can explore the spatial variation of surrogate error levels in a wide variety of surrogates (i.e., model-independent methods) leaves an important gap in our ability to perform design domain exploration. We develop a novel framework, called domain segmentation based on uncertainty in the surrogate (DSUS) to segregate the design domain based on the level of local errors. The errors in the surrogate estimation are classified into physically meaningful classes based on the user's understanding of the system and/or the accuracy requirements for the concerned system analysis. The leave-one-out cross-validation technique is used to quantity the local errors. Support vector machine (SVM) is implemented to determine the boundaries between error classes, and to classify any new design point into the pertinent error class. We also investigate the effectiveness of the leave-one-out cross-validation technique in providing a local error measure, through comparison with actual local errors. The utility of the DSUS framework is illustrated using two different surrogate modeling methods: (i) the Kriging method and (ii) the adaptive hybrid functions (AHF). The DSUS framework is applied to a series of standard test problems and engineering problems. In these case studies, the DSUS framework is observed to provide reasonable accuracy in classifying the design-space based on error levels. More than 90% of the test points are accurately classified into the appropriate error classes.
[DOI: 10.1115/1.4026150]

Keywords: cross-validation, pattern classification, support vector machine, surrogate modeling, uncertainty

1 Introduction

Uncertainties in a system may come from cognitive (qualitative) and noncognitive (quantitative) sources [1]. Noncognitive, or quantitative, sources of uncertainty can be classified into three types: (i) inherent randomness in all physical observation; (ii) statistical uncertainty; and (iii) modeling uncertainty [1]. Since a surrogate model is an approximation to an unknown function, prediction errors are generally present in the estimated function values. The two major sources of uncertainty in surrogate modeling are (i) the uncertainty in the observations (when they are noisy) and (ii) the uncertainty due to finite sample. One of the major challenges in surrogate modeling is to accurately quantify these uncertainties, and their variation in the design space [2].

The fundamental purpose of a surrogate model is to provide a cheaper and/or a more convenient representation of the actual system. Substantial research has been done in past two decades to improve the fidelity and the robustness of surrogates. However, the most advanced surrogate modeling techniques currently available are still subject to the principles of "no free lunch" theorems [3]. In addition, the adoption of the more sophisticated methods often compromises the generality of application [3]. As a result, it would be uniquely helpful to be able to quantify what level of uncertainty is expected in the surrogate model. Additionally, the knowledge of how the surrogate uncertainty and the levels of errors vary in the variable space will introduce more confidence in the usage of the surrogate (irrespective of its overall level of fidelity). The latter remains a largely uncharted territory in multidisciplinary design optimization.

1.1 An Overview of Surrogate Modeling. The need to quantify economic and engineering performance of complex products often demands highly complex and computationally expensive simulations and/or expensive experiments. Among the various approaches to deal with this problem, surrogate models (or metamodels) have gained wide acceptance from the design community. Surrogate modeling is concerned with the construction of

[1]Present address: National Renewable Energy Laboratory (NREL), Golden, CO 80401.
[2]Corresponding author.
Contributed by the Design Automation Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received December 13, 2012; final manuscript received October 23, 2013; published online January 10, 2014. Assoc. Editor: Irem Y. Tumer.

approximation models to estimate the system performance or to develop relationships between specific system inputs and outputs.

A wide variety of surrogate modeling techniques have been reported in the literature, such as (i) polynomial response surface model (PRSM) [4], (ii) Kriging [5–7], (iii) radial basis functions (RBF) [8,9], (iv) extended radial basis functions (E-RBF) [10], (v) artificial neural networks [11,12], and (vi) support vector regression (SVR) [13–15]. In the literature, the accuracy and the effectiveness of various surrogate models for linear, nonlinear, smooth, and noisy responses have also been investigated [16–19].

PRSM is a statistical tool, primarily developed for fitting analytical models (typically quadratic polynomials) to an available data set. Owing to its tractability, the classical PRSM is still one of the most widely used forms of surrogate models in engineering design [16,17]. PRSM can capture the global trend, and generally involves a smaller set of parameters (unknown coefficients) compared with other advanced models, such as Kriging and RBF. However, PRSM (quadratic polynomial) is often not adequate for capturing the local accuracy, particularly in the case of highly nonlinear functional relationships. The challenge of exact fitting has inspired researchers to explore the so-called kernel-based surrogate modeling techniques, which can provide an interpolating surface through the entire training data set. Kernel-based surrogate modeling techniques offer critical advantages over the traditional PRSM, such as the ease of extending the estimated function to higher dimensions and representation of highly nonlinear functional relationships. Kernel-based surrogate modeling methods typically make use of local information related to each training data point and combine this information to define the overall surrogate model. Kriging, RBF, and E-RBF are among the popular kernel-based surrogate modeling techniques [10]. More recently, researchers have combined different approximate models into a single hybrid model for developing weighted average surrogates [20–24]. In general, the weights of the component surrogates are determined such that the surrogate models with higher accuracy have larger weight values [23]. A considerable amount of work has been done in the literature to determine the weights of the component surrogates, which can be divided into two broad categories based on the error measures used: (i) combining surrogates using global error measures and (ii) combining surrogates using local error measures. We note that the aggregation of the different component surrogates might change the mathematical properties of each component surrogate.

1.2 Uncertainty in Surrogate Modeling. Uncertainties can generally be classified into two categories: (i) aleatoric, or statistical, uncertainties; and (ii) epistemic, or systematic, uncertainties. Epistemic uncertainty represents a lack of knowledge about the appropriate value to use for a quantity, which can be reduced through increased or more relevant data. This paper focuses on the epistemic uncertainty to quantify prediction uncertainties in surrogate models.

Uncertainty and error quantification is a classical theme in surrogate modeling. In surrogate modeling, the uncertainty arises from not knowing the output of the simulation code, except at a finite set of training points. Apley et al. [25] referred to this type of uncertainty as "interpolation uncertainty." In this paper, we use the term "prediction uncertainty," because we believe that surrogate modeling is one type of predictive models.

Bayesian methods [25,26] are widely used to quantify the interpolation uncertainty in computer experiments. Kennedy and O'Hagan [26] developed a Bayesian approach to calibrate a computer code by using observations from the real process, and subsequent prediction and uncertainty analysis of the process. The unknown inputs are represented as a parameter vector $\boldsymbol{\theta}$. Using the training data, Kennedy and O'Hagan derived the posterior distribution of $\boldsymbol{\theta}$, which, in particular, quantifies the uncertainty about $\boldsymbol{\theta}$. Apley et al. [25] developed an approach using Bayesian prediction intervals for quantifying the effects of interpolation uncertainty on the objective function in robust design. Closed-form analytical expressions were derived for the prediction intervals on the response mean, the response variance, and the robust design objective function.

Neufeld et al. [27] assessed the uncertainty introduced by a surrogate model in the conceptual design of the wing box of a generic light jet, by applying reliability based design optimization to obtain a feasible solution. A Kriging surrogate was developed from a database of finite element analysis solutions sampled across 200 evenly distributed design points. The ratios between the finite element method-based and the predicted stress were represented by a normal distribution.

Picheny [2] showed that uncertainty in conservative predictions can be compensated by adding bias to the surrogate models in order to increase safety, using constant or pointwise margins. The margins are designed based on the error distribution measures given by the model (Kriging), or based on the model-independent accuracy measures (cross-validation) of the model.

Surrogate-based optimization under uncertainty has been conducted in the literature [28–32]. Uncertainty estimates are used in adaptive sampling and optimization methods to select the next sampling point(s). The efficient global optimization approach [29] and the sequential Kriging optimization algorithm [33] used the Kriging uncertainty to seek the point of maximum expected improvement as the next infill point. Viana and Haftka [30] proposed the importation of uncertainty estimates from one surrogate to another. A SVR with an uncertainty model was developed by combining the prediction from SVR and the standard error from Kriging. Xiong et al. [31] developed cheap surrogate models to integrate information from both low-fidelity and high-fidelity models based on the Bayesian-Gaussian process modeling. The interpolation uncertainty of the surrogate model due to the lack of sufficient high-fidelity simulations is quantified in the Bayesian-Gaussian process modeling.

1.3 Research Objectives. The existing surrogate-based uncertainty modeling methods are model dependent. A generalized methodology that can be applied to a majority of surrogate models to characterize the surrogate-based uncertainty can be more helpful from the standpoint of the average end-user of optimization. To this end, this paper develops a novel approach to segregate the design domain based on levels of fidelity and quantify the uncertainty in the surrogate model. This technique, domain segmentation based on uncertainty in the surrogate (DSUS) framework, estimates the relative errors of a surrogate by investigating the cross-validation errors. Assuming that the designer/researcher does not have a definitive insight into the functional relationships that we are seeking to model, the measured (or simulated) sample data are all the information that we have at our disposal. A broad objective of evaluating the uncertainty in a surrogate is to characterize the uncertainty that is introduced during the surrogate modeling process itself. An uncertainty modeling technique like the one presented in this paper is expected to streamline the overall system-design efforts, by providing the user more confidence in the surrogates being used. Key features of the DSUS framework are

(1) This method segregates the design domain into regions based on the level of local modeling errors (or level of fidelity). Pattern classification methods are used for this purpose.
(2) This method can classify any point/design, for which the actual functional response is not known, into an error class, and quantify the uncertainty in its predicted function response.
(3) This method is readily applicable to a majority of interpolative surrogate models.

1.4 Applications of the DSUS Model. The DSUS framework is useful for a variety of applications, such as optimization, system analysis (involving unknown functional relationships), and

surrogate modeling improvement. In surrogate-based optimization, optimal solutions in regions with smaller errors are more reliable than solutions in regions with larger errors. If optimal solutions are in a region with large errors, the user may add more training points in that region to improve the local surrogate accuracy. In addition, the DSUS framework can be used to quantify the uncertainty in the optimal solutions based on their locations in the design space. For instance, in surrogate-based wind farm layout optimization [34], power generation is represented as a function of turbine locations. The uncertainty in the power generation can be estimated based on the corresponding candidate farm layout design.

In surrogate-based system analysis, the knowledge of the errors and uncertainties in the surrogate (determined by DSUS) is also helpful for the user/designer. For example, in the response surface-based wind farm cost (RS-WFC) model (which estimates the total annual cost of a wind farm using surrogate modeling methods) [35], DSUS can quantify the errors and uncertainties attributable to surrogate models in the predicted wind farm cost. The RS-WFC model was developed using surrogate models for onshore wind farms in the U.S. The RS-WFC model is composed of three components, which are (i) the installation cost, (ii) the annual operation and maintenance (O&M) cost, and (iii) the total annual cost of a wind farm. The input parameters to the RS-WFC total annual cost model are (i) the number and (ii) the rated power of each wind turbine installed in the wind farm. The design-domain based uncertainty information provided by DSUS can be used to implement adaptive sampling strategies during surrogate development. In addition, the domain segmentation in the DSUS can also be implemented based on other errors as well (irrespective of the source of errors).

The remainder of the paper is organized as follows: Sections 2 and 3 present the overall framework and components of the DSUS framework, respectively; The surrogate modeling methods used to illustrate the DSUS framework are provided in Sec. 4; Sec. 5 describes different standard problems and engineering problems to which the DSUS framework is applied; the numerical settings and results of case studies are shown and discussed in Sec. 6; and Sec. 7 provides the concluding remarks.

2 DSUS

In this section, we develop a framework to segregate the design domain of a surrogate into classes based on the local prediction errors. The uncertainty in each class is represented by the mean and standard deviation values of cross-validation errors.

For an engineering design model, it is important to balance the accuracy of model predictions with the computational complexity of the model used. Model prediction errors that lie within a specified range may be acceptable. Based on prior experience or practical design requirements, the designer may know what levels of errors are acceptable for a particular problem. For instance, for a wind farm power generation model (which estimates the power generation of a wind farm using analytical methods) [36,37], 2% estimation error might be desirable; 2–10% error is acceptable; and higher than 10% error is unacceptable. If the whole design domain is divided into physically meaningful classes based on the level of local prediction errors, new designs could be classified into appropriate levels of fidelity based on their locations. Figure 1 illustrates the concept of determining the predictive modeling errors in a two design variable system. In Fig. 1, the local errors of the model are classified into three classes, and each color represents one class of errors. The boundaries between classes can be determined using pattern classification methods. The designer can estimate the confidence of a new design based on the region into which the design point is classified; these regions can correspond to "desirable," "acceptable," and "unacceptable" levels of accuracy. Note that Fig. 1 is the domain segmentation result of the DSUS application to the Dixon & Price function (in conjunction with the AHF), which is further illustrated in Sec. 6. The DSUS framework is implemented by observing the following sequence of three steps:

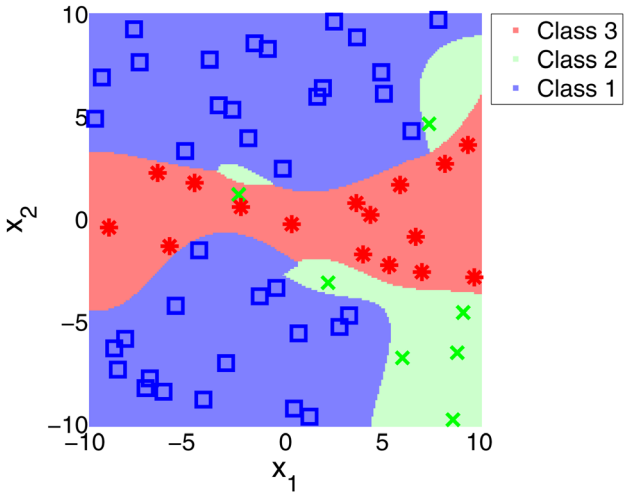

Fig. 1 The illustration of the prediction uncertainty modeling (class 1: low error; class 2: medium error; class 3: high error)

(1) An interpolating surrogate modeling method is used to develop the predictive model, e.g., Kriging, RBF, or AHF.
(2) Cross-validation errors are evaluated using the leave-one-out strategy, and we classify the training points into classes based on the cross-validation error at each training point. The error classes are determined based on the level of error which depends on the design problem. For example, three error classes are defined in the wind farm power generation model discussed above.
(3) A model is developed to determine the class boundaries in the input variable space using SVM. The input variables of the surrogates are thus considered as input features in the classification process.

It is important to note that in steps 1 and 2, regression models (e.g., PRSM) can also be used. However, in that case, a direct error estimation at each training point may be used instead of cross-validation errors. The development of the DSUS framework is illustrated in Fig. 2. In Sec. 3, we discuss the details of the DSUS framework.

3 Major Components of the DSUS Model

The two major components of the DSUS framework, cross-validation, and pattern classification are developed in detail in this section.

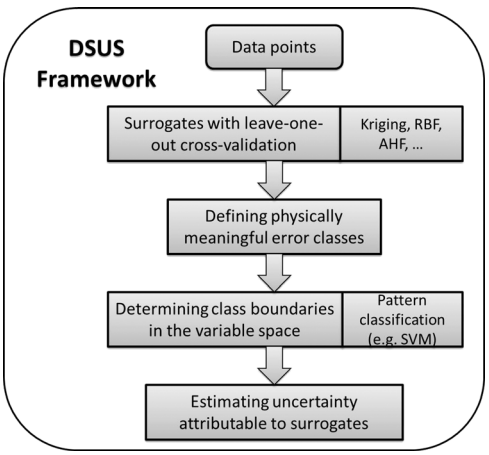

Fig. 2 The framework of the DSUS methodology

3.1 Cross-Validation. Cross-validation errors are used in this paper as a measure of the accuracy of the surrogate. Cross-validation is a technique that is used to analyze and improve the accuracy of a surrogate model. Cross-validation error is the error estimated at a data point, when the response surface is fitted to all the data points except that point (also called the leave-one-out strategy). A vector of cross-validation errors, $\tilde{\boldsymbol{e}}$, can be obtained, when the response surfaces are fitted to all the other points. This vector is known as the prediction sum of squares (the PRESS vector). Leave-one-out cross-validation has been widely used as an estimate of prediction errors, to evaluate the performance or accuracy of a surrogate [16,24,38–41].

In order to obtain the error at each training point, the leave-one-out strategy is adopted in the DSUS framework. The relative accuracy error (RAE) [35] is used to classify the training points into classes. The RAE is evaluated at each training point as

$$\mathrm{RAE}(X^k) = \left| \frac{\tilde{f}_c(X^k) - f(X^k)}{f(X^k)} \right|, \quad k = 1, 2, \cdots, n_{\mathrm{tr}} \tag{1}$$

where n_{tr} is the number of training points; $f(x^k)$ represents the actual function value at the training point x^k; and the term $\tilde{f}_c(x^k)$ is the corresponding function value at x^k, estimated by the surrogate that is trained without the data point x^k (based on the leave-one-out cross-validation principle). According to the RAE values, we can manually classify the training points into error classes, and define the lower and upper limits of each class. The definition of the error levels (error ranges) depends on the intended use of the domain-based surrogate uncertainty information, e.g., using it for increasing the confidence of a particular system analysis.

It is important to note that, when the number of training points is extremely small, the discrepancy between the intermediate (leave-one-out) surrogate performance and the actual surrogate performance might be significant. However, when the number of training points is relatively large, the performance of the surrogate is not affected significantly by removing one point, which is illustrated in Sec. 3.3. To the best of the authors' knowledge, there does not exist accurate model-independent measure of local errors. Hence, the leave-one-out cross-validation error is an acceptable choice for representing local errors, especially when the number of sample points is 10–20 times the problem dimension (common in engineering design [24,29]). However, more accurate model-independent measures of local errors (when developed) can further improve the effectiveness and the applicability of the DSUS method.

3.2 Pattern Classification. The classes generated in the previous step are used to determine the classification boundaries. A wide variety of pattern classification methods are available in the literature [11], such as (i) linear discriminant analysis; (ii) principal components analysis; (iii) Kernel estimation and K-nearest-neighbor algorithms; (iv) Perceptrons; (v) neural network; and (vi) SVM. In this paper, the uncertainty classification in surrogate modeling is a multiclass classification problem. SVM, which is known to be a competitive approach for multiclass classification problem [42], is adopted in this paper.

SVM is a popular machine learning technique that has been used for classification, regression, and other learning tasks. Given a training set of instance-label pairs $(x_i, y_i), i = 1, \cdots, m$, where $x_i \in R^n$ and $y \in \{1, -1\}^m$, the determination of the support vectors requires the solution of the following optimization problem:

$$\begin{aligned} &\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i \\ &\text{subject to } y_i\left(w^T \phi(x_i) + b\right) \geq 1 - \xi_i \\ &\xi_i \geq 0 \end{aligned} \tag{2}$$

Here, the training vectors x_i are mapped onto a higher (maybe infinite) dimensional space by the function ϕ. SVM finds a hyperplane with the maximum margin in this higher dimensional space. The vector $\boldsymbol{w}$ denotes the normal vector to the hyperplane; and the term $(b/\| \boldsymbol{w} \|)$ represents the distance between the hyperplane and the origin along the normal vector $\boldsymbol{w}$. The parameter $C > 0$ is the penalty parameter in the error term. The generic term ξ is a slack variable, which measures the degree of misclassification of the datum x_i. The function $K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j)$ is called the kernel function. Four popular kernels are

(1) Linear: $K(x_i, x_j) = x_i^T x_j$
(2) Polynomial: $K(x_i, x_j) = \left(\gamma x_i^T x_j + r\right)^d, \gamma > 0$
(3) Radial basis function: $K(x_i, x_j) = \exp\left(-\gamma \| x_i - x_j \|^2\right), \gamma > 0$
(4) Sigmoid: $K(x_i, x_j) = \tanh\left(\gamma x_i^T x_j + r\right)$

where γ, r, and d are the kernel parameters.

The conventional support vector machine is a powerful tool for binary classification, which is capable of generating fast classifier functions following a training period. There exist several advanced SVM approaches to solve classification problems that involve three or more classes [43]:

(1) One-against-all classification, in which one binary SVM is used to separate members of each class from the members of other classes.
(2) One-against-one classification, which constructs $k(k–1)/2$ classifiers, where each classifier is trained on data from two classes (k is the number of classes).
(3) Directed acyclic graph SVM, in which the training phase is the same as the one-against-one method.

Hsu and Lin [43] provided a detailed comparison of the above three approaches and concluded that "one-against-one" classification is a competitive approach; this approach is adopted in this paper. For training data from the i^{th} and the j^{th} classes, the following two-class classification problem is solved [44]:

$$\begin{aligned} &\min_{w^{ij}, b^{ij}, \xi^{ij}} \frac{1}{2} \left(w^{ij}\right)^T w + C \sum_t \left(\xi^{ij}\right)_t \\ &\text{subject to} \\ &\left(w^{ij}\right)^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \quad \text{if } x_{\mathrm{t}} \text{ in the } i^{\mathrm{th}} \text{ class} \\ &\left(w^{ij}\right)^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \quad \text{if } x_{\mathrm{t}} \text{ in the } j^{\mathrm{th}} \text{ class} \\ &\xi_t^{ij} \geq 0 \end{aligned} \tag{3}$$

A voting strategy is used in classification. Each binary classification is considered to be a voting process where votes can be cast for all data points x; and in the end, a point is designated to be in the class with the maximum number of votes [44]. In this paper, we use an efficient SVM package, library for support vector machines, developed by Chang and Lin [44].

3.3 Illustrating Cross-Validation Errors as Local Error Measures. Both global and local error measure metrics were used to assess the accuracy of surrogate models. Generally, cross-validation (prediction sum of squares) is used as one of the global measure metrics. Local error metrics include maximum absolute error and RAE. In the DSUS framework, the leave-one-out cross-validation errors at all training points, in conjunction with the RAE metric, are used as local error measures. A method is proposed in this paper to evaluate the performance of this local error measure metric.

3.3.1 Surrogate Prediction in the Neighborhood of Training Points. The local errors of the surrogate are evaluated in the neighborhood of each training point. A local hypercube is constructed to include one training point, and the accuracy of the surrogate is estimated within the local hypercube. To construct the local hypercube in a n_{d} dimensional space, for a point $X^j = (x_1^j, x_2^j, \cdots, x_{n_{\mathrm{d}}}^j)$, we first find its nearest neighbor ${}_{\mathrm{ne}}X^j$ in

terms of Euclidean distance. The length of the hypercube along each dimension is determined as follows:

$$L_i = \left| x_i^j - {}_{\text{ne}} X_i^j \right|, \quad \forall i = 1, 2, \cdots, n_{\text{d}} \tag{4}$$

The j^{th} hypercube can be expressed by

$$H_j = \left\{ (x_1^{jL}, x_1^{jU}), (x_2^{jL}, x_2^{jU}), \cdots, (x_{n_{\text{d}}}^{jL}, x_{n_{\text{d}}}^{jU}) \right\} \tag{5}$$

where

$$x_i^{jL} = x_i^j - \frac{L_i}{2}, \quad \forall i = 1, 2, \cdots, n_{\text{d}}$$
$$x_i^{jU} = x_i^j + \frac{L_i}{2}, \quad \forall i = 1, 2, \cdots, n_{\text{d}}$$

The parameters x_i^{jL} and x_i^{jU} define the boundaries of the local hypercube.

Within each local hypercube, a set of test points are generated. The responses of the test points are estimated for the surrogate trained using all training points. The RAE at each test point within the j^{th} local hypercube is given by

$$\text{RAE}^{\text{te}}(X^{jk}) = \left| \frac{\tilde{f}(X^{jk}) - f(X^{jk})}{f(X^{jk})} \right| \tag{6}$$

$$\forall j = 1, 2, \cdots, n_{\text{tr}}, k = 1, 2, \cdots, n_{\text{te}} \tag{7}$$

where $f(X^{jk})$ represents the actual function value at the test point X^{jk}; $\tilde{f}(X^{jk})$ is the estimated response at the point X^{jk} using the surrogate trained by all training points; and n_{te} is the number of test points within each local hypercube. For the j^{th} local hypercube, the average of the RAE values for the n_{te} points, $\overline{\text{RAE}^{\text{te}}}_j$, is given by

$$\overline{\text{RAE}^{\text{te}}}_j = \frac{1}{n_{\text{te}}} \sum_{k=1}^{n_{\text{te}}} \text{RAE}^{\text{te}}(X^{jk}) \tag{8}$$

where $\text{RAE}^{\text{te}}(X^{jk})$ represents the RAE value for the k^{th} test point within the j^{th} local hypercube.

The local errors estimated by the leave-one-out cross-validation strategy at each training point are compared with the $\overline{\text{RAE}^{\text{te}}}$ values estimated within the corresponding local hypercubes (Eq. (8)).

3.3.2 Analytical Examples. Two analytical examples are used to show the surrogate prediction in the neighborhood of training points, which are (i) a one-variable function [38] and (ii) the two-variable Dixon & Price function [24]. The expressions of the two functions are given in Eqs. (9) and (10). Two different surrogate modeling methods are used: (i) the Kriging method and (ii) the AHF [45].

Test function 1: one-variable function

$$f(x) = (6x_1 - 2)^2 \sin[2(6x_1 - 2)] \quad \text{where } x_1 \in [0\ 1] \tag{9}$$

Test function 2: two-variable Dixon & Price function

$$f(x) = (x_1 - 1)^2 + 2\left(2x_2^2 - x_1\right)^2 \quad \text{where } x_i \in [-10\ 10] \tag{10}$$

For the one-variable function, 15 training points are generated to construct the surrogate. The 15 training points are generated using the optimal Latin hypercube sampling strategy which maximizes the minimum distance between points. For the Dixon & Price problem, 30 training points are generated to construct the surrogate. The Latin hypercube sampling strategy developed by Audze and Eglais [46] is adopted to determine the locations of sample points. The following optimality criterion is used in the sampling method:

$$\sum_{p}^{n_{\text{tr}}} \sum_{q=p+1}^{n_{\text{tr}}} \frac{1}{L_{pq}^2} \rightarrow \min \tag{11}$$

where L_{pq} is the distance between the points p and q and n_{tr} is the total number of sample points.

The RAE error at each training point and the $\overline{\text{RAE}^{\text{te}}}$ value for each local hypercube are estimated. The values of RAE and $\overline{\text{RAE}^{\text{te}}}$ are classified into four groups based on the mean values of RAE and $\overline{\text{RAE}^{\text{te}}}$ for all the training points. Figures 3 and 4 show the comparison of the cross-validation errors and the actual local errors. The four classes are defined as follows:

$$\text{RAE}_i \in \begin{cases} \text{Class 1}, & \text{if } \text{RAE}_i < 0.5\mu_{\text{RAE}} \\ \text{Class 2}, & \text{if } 0.5\mu_{\text{RAE}} \le \text{RAE}_i < \mu_{\text{RAE}} \\ \text{Class 3}, & \text{if } \mu_{\text{RAE}} \le \text{RAE}_i < 1.5\mu_{\text{RAE}} \\ \text{Class 4}, & \text{if } \text{RAE}_i \ge 1.5\mu_{\text{RAE}} \end{cases} \tag{12}$$

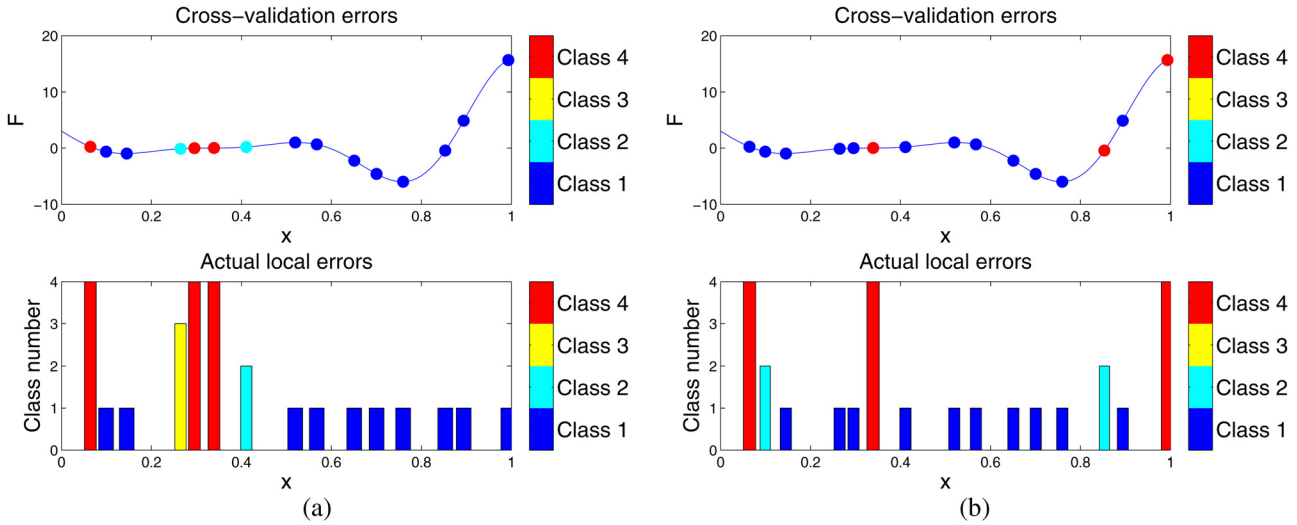

Fig. 3 Comparison of cross-validation errors and actual local errors (1-variable function): (*a*) AHF and (*b*) Kriging

where RAE_i represents the RAE value at the i^{th} training point and μ_{RAE} is the mean RAE value for the 30 training points

$$\overline{RAE^{te}}_i \in \begin{cases} \text{Class 1,} & \text{if } \overline{RAE^{te}} < 0.5\mu_{\overline{RAE^{te}}} \\ \text{Class 2,} & \text{if } 0.5\mu_{\overline{RAE^{te}}} \le \overline{RAE^{te}} < \mu_{\overline{RAE^{te}}} \\ \text{Class 3,} & \text{if } \mu_{\overline{RAE^{te}}} \le \overline{RAE^{te}} < 1.5\mu_{\overline{RAE^{te}}} \\ \text{Class 4,} & \text{if } \overline{RAE^{te}} \ge 1.5\mu_{\overline{RAE^{te}}} \end{cases} \quad (13)$$

where $\overline{RAE^{te}}_i$ represents the $\overline{RAE^{te}}$ value for the i^{th} local hypercube and $\mu_{\overline{RAE^{te}}}$ is the mean $\overline{RAE^{te}}$ value for all the local hypercubes.

The evaluation results of the one-variable function are shown in Fig. 3. In Fig. 3(*a*), the figure on the top illustrates the leave-one-out cross-validation errors using differing colors, while the figure below it shows the actual local errors evaluated in the neighborhood (hypercube) of training points. The levels of errors are represented using different colors. It is observed that the cross-validation errors and the actual local errors belong to the same class for 14 out of 15 training points. It is also observed from Fig. 3(*b*) that for the Kriging surrogate, the cross-validation errors and the actual local errors belong to the same class for 12 out of 15 training points.

For the Dixon & Price function, the leave-one-out cross-validation errors at the training points are illustrated in Figs. 4(*b*) and 4(*d*). Figures 4(*a*) and 4(*c*) show the local hypercubes at all the 30 training points. The levels of errors are represented using different colors. It is observed that (i) for the AHF surrogate, the cross-validation errors and the actual local errors belong to the same class for 25 out of 30 training points and (ii) for the Kriging surrogate, the cross-validation errors and the actual local errors belong to the same class for 27 out of 30 training points. The comparison results show that the leave-one-out cross-validation strategy can capture the local errors of a surrogate with a reasonable accuracy of more than 80% for these low-dimensional problems. It is important to note that (i) the results of the validation might be slightly different for other test problems and (ii) the leave-one-out cross-validation errors are the approximations of the actual errors in surrogate modeling.

4 Surrogate Modeling

The DSUS framework presented in this paper is readily applicable to a majority of standard interpolative methods, such as Kriging, RBF, and E-RBF. For regression methods (e.g., polynomial and support vector regression), the local error at each training point need not be quantified using the leave-one-out cross-validation technique. In the case of regression models, the local errors could be directly evaluated by comparing the approximated values with the actual function values at the training points. In this paper, the DSUS framework is illustrated using two different surrogate modeling methods: (i) the Kriging method and (ii) a newly developed hybrid surrogate, the AHF [34,45].

4.1 Kriging. Kriging [5,6] is an approach to approximate irregular data. The Kriging approximation function consists of two components: (i) a global trend function and (ii) a functional departure from the trend function. The trend function is generally a polynomial (e.g., constant, linear, or quadratic). The general form of the Kriging surrogate model is given by

$$\tilde{f}(x) = G(x) + Z(x) \quad (14)$$

where $\tilde{f}(x)$ is the unknown function of interest, G(x) is the user-defined approximation (usually polynomial) function, and $Z(x)$ is the realization of a stochastic process with a zero mean and a non-zero covariance. The $(i,j)^{th}$ element of the covariance matrix of $Z(x)$ is given as

$$\mathrm{COV}[Z(x^i), Z(x^j)] = \sigma_z^2 R_{ij} \quad (15)$$

where R_{ij} is the correlation function between the i^{th} and the j^{th} data points and σ_z^2 is the process variance. In the present paper, a Gaussian function is used as the correlation function, defined as

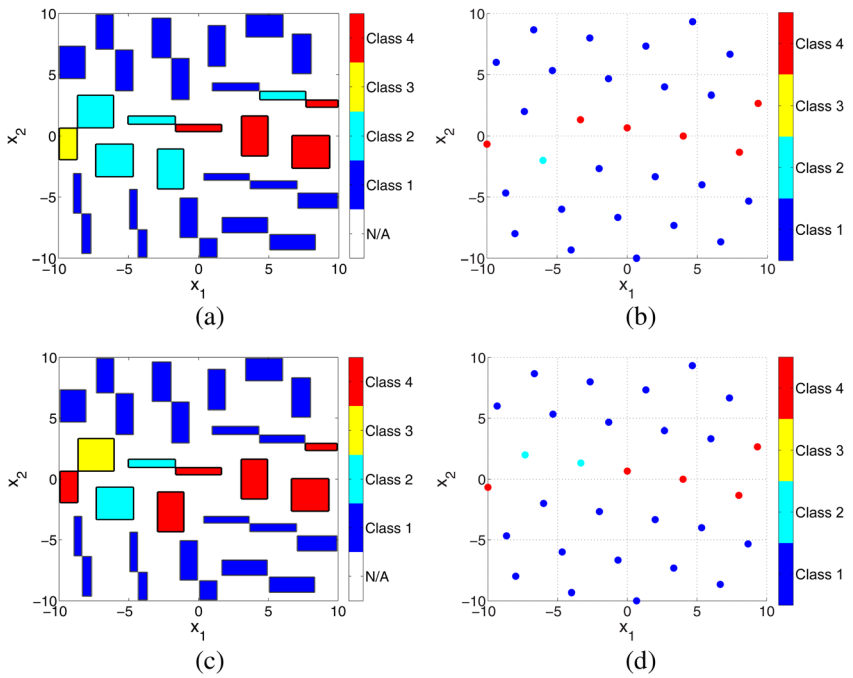

Fig. 4 Comparison of cross-validation errors and actual local errors (Dixon & Price function): (*a*) actual local errors (AHF), (*b*) cross-validation errors (AHF), (*c*) actual local errors (Kriging), and (*d*) cross-validation errors (Kriging)

$$R(x^i, x^j) = R_{ij} = \exp\left\{-\Sigma_{k=1}^{n_d} \theta_k \left(x_k^i - x_k^j\right)^2\right\} \quad (16)$$

where θ_k is distinct for each dimension and these unknown parameters are generally obtained by solving a nonlinear optimization problem. In this paper, we use an efficient MATLAB implementation of the Kriging surrogate called design and analysis of computer experiments, developed by Lophaven et al. [47]. The order of the global polynomial trend function was specified to be zero.

4.2 AHF. The AHF methodology, recently developed by Zhang et al. [45], formulates a reliable trust region, and adaptively combines characteristically different surrogate models. The weight of each contributing surrogate model is represented as a function of the input domain, based on a local *measure of accuracy* of that surrogate model. Such an approach exploits the advantages of each component surrogate, thereby capturing both the global and the local trend of complex functional relationships. In this paper, the AHF combines three component surrogate models by characterizing and evaluating the local *measure of accuracy* of each model. The three models are (i) RBF, (ii) E-RBF, and (iii) Kriging. The AHF methodology introduces a three-step approach

(1) Determination of a trust region: numerical bounds of the estimated parameter (output) as functions of the independent parameters (input vector),
(2) Definition of a local *measure of accuracy* (using kernel functions) of the estimated function value and representation of the corresponding distribution parameters as functions of the input vector,
(3) Weighted summation of characteristically different surrogate models (component surrogates) based on the local *measure of accuracy* (defined in the previous step).

The AHF is a weighted summation of function values estimated by the component surrogates as given by

$$\tilde{f}_{\text{AHF}} = \sum_{i=1}^{n_s} w_i(x)\tilde{f}_i(x) \quad (17)$$

where n_s is the number of component surrogates in the AHF and $\tilde{f}_i(x)$ represents the value estimated by the i^{th} component surrogate. The weights, w_i, are expressed in terms of a *measure of accuracy* (based on a crowding distance-based trust region), which is given by

$$w_i(x) = \frac{P_i(x)}{\sum_{i=1}^{n_s} P_i(x)} \quad (18)$$

where $P_i(x)$ is the *measure of accuracy* of the i^{th} surrogate for point x. The detailed formulation of the AHF method can be found in the paper by Zhang et al. [45].

5 Case Studies

We apply the DSUS framework to a series of standard problems, including (i) one-variable function (Eq. (9)) and (ii) two-variable Dixon & Price function (Eq. (10)). The prediction uncertainties in (i) the *RS-WFC model* [35], (ii) the wind power potential (WPP) model [48], and (iii) the *commonality representation in product family design* [49] are also characterized to illustrate the applicability of DSUS to practical engineering design problems. In the case of problems where the user has control over the design of experiments, the choice of an appropriate sampling technique is generally considered crucial. Two efficient sampling methods are adopted in this paper: (i) Latin hypercube sampling and (ii) Sobol's quasirandom sequence generator.

5.1 Onshore Wind Farm Cost Model. A RS-WFC model was developed for wind farm design [35,50]. The RS-WFC model was developed using E-RBF for onshore wind farms in the U.S. The RS-WFC model is composed of three components that estimate the effects of engineering and economic factors on (i) the installation cost, (ii) the annual operation and maintenance (O&M) cost, and (iii) the total annual cost of a wind farm.

In this paper, we adopted the RS-WFC model to estimate *the total annual cost of a wind farm per kilowatt installed* and to characterize the uncertainty in the cost. The AHF (or Kriging) surrogate is used to develop the cost model instead of using the E-RBF method. The input parameters to the RS-WFC total annual cost model are (i) the number and (ii) the rated power of each wind turbine installed in the wind farm; and the output is the total annual cost of the wind farm. Cost data collected for the state of North Dakota are used to train the cost model.

5.2 WPP Model. The WPP model [48] was developed to quantify the maximum wind farm capacity factor that can be accomplished (for specified farm-land area and installed capacity) for a given site. For any farm site, according to the recorded wind data, we can (i) estimate the parameters of the joint distribution of wind speed and direction (bivariate normal distribution is adopted in this paper) and (ii) predict the maximum capacity factor for a specified farm size and capacity, using the WPP model. Further details of the WPP model can be found in the paper by Zhang et al. [48].

In this paper, we explore two different scenarios in the WPP estimation, and the uncertainty in the WPP is characterized using the DSUS framework. We consider a fixed-size (land) rectangular wind farm that comprised a defined turbine-type. Two different scenarios are considered here, given as (i) case 1: evaluating the WPP (maximum capacity factor) of a farm, comprising four turbines; and (ii) case 2: evaluating the WPP (maximum capacity factor) of a farm, comprising nine turbines. The *GE-1.5 MW-xle* [51] turbine is chosen as the specified turbine-type in both cases 1 and 2. The input parameters to the WPP model include (i) the mean of the wind speed distribution; (ii) the mean of the wind direction distribution; (iii) the standard deviation of the wind speed distribution; (iv) the standard deviation of the wind direction distribution; and (v) the correlation coefficient between wind speed and wind direction. The output is the maximum capacity factor of the wind farm.

5.3 Commonality Representation in Product Family Design. A product family is a group of related products that are derived from a common product platform to satisfy a variety of market niches. The sharing of a common platform by different products is expected to result in (i) reduced overhead, (ii) lower per-product cost, and (iii) increased profit. The recently developed comprehensive product platform planning (CP^3) framework [49] formulated a generalized mathematical model to represent the complex platform planning process. The CP^3 model formulates a generalized equality constraint (the *commonality constraint*) to represent the variable-based platform formation. The presence of a combination of integer variables (specifically binary variables) and continuous variables can be attributed to the combinatorial process of platform identification. The authors [52] developed a methodology to reduce the high dimensional binary integer problem to a more tractable integer problem, where the commonality matrix is represented by a set of integer variables. The detailed formulation of the *commonality matrix* can be found in the paper by Chowdhury et al. [49]. This commonality index is essentially based on the ratio of "the number of unique parts" to "the total number of parts" in the product family.

In this paper, we develop a surrogate model to represent the *commonality index* as a function of the integer variables, and the

Table 1 Numerical setup for test problems

Problem	Number of variables	Number of training points	Number of test points	SVM kernel
One-variable function	1	15	20	Linear
Dixon & Price function	2	60	20	Radial basis function
Wind farm cost	2	60	20	Radial basis function
WPP (four turbines)	5	100	20	Radial basis function
WPP (nine turbines)	5	100	20	Radial basis function
Commonality (two products)	7	105	35	Radial basis function
Commonality (three products)	7	105	21	Radial basis function

Table 2 The uncertainty scale in each class

Problem	Class 1	Class 2	Class 3
One-variable function	RAE $<$ 10%	RAE $\geq$ 10%	—
Dixon & Price function	RAE $<$ 10%	10% $\leq$ RAE $<$25%	RAE $\geq$ 25%
Wind farm cost	RAE $<$ 0.5%	0.5% $\leq$ RAE $<$1%	RAE $\geq$ 1%
WPP (four turbines)	RAE $<$ 5%	5% $\leq$ RAE $<$10%	RAE $\geq$ 10%
WPP (nine turbines)	RAE $<$ 5%	5% $\leq$ RAE $<$10%	RAE $\geq$ 10%
Commonality (two products)	RAE $<$ 0.5%	0.5% $\leq$ RAE $<$1%	RAE $\geq$ 1%
Commonality (three products)	RAE $<$ 5%	5% $\leq$ RAE $<$10%	RAE $\geq$ 10%

uncertainty in the surrogate is characterized by the DSUS method. The input parameters to the surrogate are the seven integer variables; and the output is the value of *commonality index*.

6 Results and Discussion

The results from the application of the DSUS methodology are discussed in this section. The parameter selections and numerical settings of the DSUS framework are also summarized. In addition, we define the measure of *prediction uncertainty* in the surrogate. The SVM kernels and parameters are determined for the studied cases, followed by the discussion of the uncertainty prediction results.

6.1 Numerical Settings. The AHF surrogate is an ensemble of RBF, E-RBF, and Kriging methods. Detailed formulations of RBF and E-RBF can be found in the paper [45]. The parameter values are the same as specified in that paper. For the Kriging method used in this paper, the bounds on the correlation parameters in the nonlinear optimization, θ_l and θ_u, are selected as 0.1 and 20. Under the Kriging approach, the order of the global polynomial trend function is specified to be zero. The parameters of the SVM depend on the kernel used.

The numerical settings for each problem are summarized in Table 1, which includes (i) the number of input variables, (ii) the number of training points, (iii) the number of test points, and (iv) the SVM kernel. The sampling points for the first two analytical problems (one-variable function and two-variable Dixon & Price function) are generated using the Latin hypercube sampling method. The data used to develop and test the RS-WFC model are obtained from the energy efficiency and renewable energy program at the U.S. Department of Energy [53]. For the WPP estimation, the training points are generated using Sobol's quasirandom sequence generator. For the *commonality representation in product family design* problem, two different sampling methods are used to generate training and test points. The training points are generated using Sobol's quasirandom sequence generator; and test points are generated using the Latin hypercube sampling method. This sampling strategy differentiates the training points and the test points. We select the kernel function through numerical experiments. The linear kernel is selected for the one-variable function; and the radial basis function kernel is adopted for the other problems.

We classify the training points into different error classes based on the RAE values. The user-defined lower and upper limits of each class are listed in Table 2. It can be seen from Table 2 that (i) two classes are selected for the one-variable problem and (ii) three classes are defined for the other problems. However in practical engineering design, the user is expected to specify the classes based on the physical implication of the errors (e.g., desirable, acceptable, unacceptable).

6.2 Representation of Prediction Uncertainty. Tables 3 and 4 show the uncertainty in each class, in terms of the mean and standard deviation values of the prediction errors. The parameter n_{sv} (in Tables 3 and 4) represents the number of points in each class. Classes 1, 2, and 3 represent uncertainty levels of low errors, medium errors, and high errors, respectively. For any new point (candidate design), the DSUS framework can classify it into one of these error classes. In addition, the *prediction uncertainty* of the surrogate-based functional response of this new design is given by the mean (μ) and the standard deviation (σ) of errors in that class.

For the one-variable problem, the RAEs are classified into two classes that represent the low and high errors, respectively. For the other problems, the RAEs are classified into three classes to, respectively, represent the low, medium, and high errors of the surrogate. In real-life engineering design, the user might use "desirable," "acceptable," and "unacceptable" levels to characterize the system design. It is observed from Tables 3 and 4 (as expected) that the mean error in the Class 3 is higher than those in Classes 1 and 2. This observation shows that there is significant scope to improve the surrogate accuracies in Class 3 regions. This design-space information provided by DSUS could be helpful in advancing surrogate modeling fidelity and in effective surrogate-based optimization.

Table 3 Uncertainty scale (mean and standard deviation) of each class with AHF surrogate

	Class 1			Class 2			Class 3		
Problem	n_{sv}	μ_1	σ_1	n_{sv}	μ_2	σ_2	n_{sv}	μ_3	σ_3
One-variable function	10	0.0341	0.0227	5	0.5092	0.3791	—	—	—
Dixon & price function	37	0.0409	0.0293	7	0.1718	0.0582	16	2.6109	2.4950
Wind farm cost	39	0.0018	0.0011	13	0.0066	0.0013	8	0.0216	0.0102
WPP (four turbines)	15	0.0312	0.0148	10	0.0779	0.0153	75	0.7426	1.3464
WPP (nine turbines)	12	0.0290	0.0124	10	0.0701	0.0172	78	0.5374	0.6207
Commonality (two products)	53	0.0027	0.0012	29	0.0063	0.0011	23	0.0506	0.0536
Commonality (three products)	48	0.0242	0.0124	37	0.0740	0.0149	20	0.1590	0.0906

Table 4 Uncertainty scale (mean and standard deviation) of each class with Kriging surrogate

Problem	Class 1			Class 2			Class 3		
	n_{sv}	μ_1	σ_1	n_{sv}	μ_2	σ_2	n_{sv}	μ_3	σ_3
One-variable function	7	0.0129	0.0106	8	0.3224	0.2445	—	—	—
Dixon & price function	21	0.0106	0.0200	11	0.1503	0.0437	28	1.6421	2.0346
Wind farm cost	40	0.0018	0.0015	9	0.0069	0.0015	11	0.0179	0.0080
WPP (four turbines)	14	0.0214	0.0153	12	0.0702	0.0159	74	0.7160	1.3311
WPP (nine turbines)	16	0.0247	0.0173	15	0.0750	0.0130	69	0.5392	0.6653
Commonality (two products)	25	0.0024	0.0011	28	0.0061	0.0009	52	0.0829	0.0204
Commonality (three products)	32	0.0257	0.0145	27	0.0708	0.0162	46	0.1861	0.1126

Table 5 Classification accuracy of each test problem

Problem	AHF		Kriging	
	Parameters	Accuracy	Parameters	Accuracy
One-variable function	$C=1$	90% (18/20)	$C=1$	95% (19/20)
Dixon & Price function	$C=20, \gamma=0.5$	85% (17/20)	$C=1, \gamma=0.5$	95% (19/20)
Wind farm cost	$C=1, \gamma=0.8$	100% (20/20)	$C=1, \gamma=0.8$	95% (19/20)
WPP (four turbines)	$C=1, \gamma=1$	90% (18/20)	$C=1, \gamma=0.2$	90% (18/20)
WPP (nine turbines)	$C=1, \gamma=0.2$	95% (19/20)	$C=1, \gamma=0.2$	95% (19/20)
Commonality (two products)	$C=0.0313, \gamma=0.0078$	94% (33/35)	$C=0.0313, \gamma=0.0078$	100% (35/35)
Commonality (three products)	$C=2, \gamma=0.12$	95% (20/21)	$C=2, \gamma=0.06$	86% (18/21)

6.3 Uncertainty Prediction Results. There are (i) one parameter for the SVM linear kernel, C; and (ii) two parameters for the SVM radial basis function kernel: C and γ. Cross-validation technique [44] is used to achieve a high training accuracy. A grid-search technique [44] is performed on C and γ using cross-validation. Various pairs of (C, γ) values are tested and the one with the best cross-validation accuracy is selected. Table 5 gives the best C and γ values for all the test problems. It is important to note that the cross-validation used to determine the SVM parameter values is not related in any way to the cross-validation error of the surrogates. For instance, the C and γ values for the Dixon & Price function with AHF are found to be 20 and 0.5, respectively. In this case, the design domain of the function is segregated into three error classes, as shown in Fig. 1.

Table 5 shows the prediction accuracy of each problem. In the table, the prediction accuracy is given by the ratio of *the number of test points correctly classified* to *the total number of test points*. It can be seen that the DSUS framework performs fairly well for all the problems. The classification accuracy of the DSUS prediction is more than 90% for most test problems. The DSUS framework characterized the uncertainty levels with 100% accuracies in two cases: (i) wind farm cost with AHF surrogate and (ii) commonality (two products) index approximation with Kriging surrogate.

The number of training points required to obtain an acceptable surrogate accuracy may be significantly different from that required for developing the classification. We should increase the number of training points if it is not enough for classification. Future research should investigate the training points balance between surrogate modeling and pattern classification, to promote acceptable accuracy both in the surrogate modeling stage and the classification stage, while maintaining acceptable system evaluation expense.

6.3.1 Uncertainty in the Wind Farm Cost. Table 6 shows the uncertainty in the estimated AHF-based wind farm cost for three new wind farm designs. For the third design, the DSUS method classifies the design into error class 3 based on the input parameters of the cost model (44 wind turbines with rated power of 1.5 MW). The mean and the standard deviation values of errors within class 3 are 0.0216 and 0.0102, respectively. The total annual cost per kilowatt installed (for this farm) is estimated to be 122.28 $/kW. Assuming a lifetime of 20 years, the 2.16% (mean value) error in the cost estimation translates to approximately 3.5 million dollars, which is an appreciable value for such a medium-scale wind farm.

Table 6 Uncertainty characterization for new designs (AHF-based wind farm cost)

New design	Number of turbines	Rated power	Class	Uncertainty (μ, σ)
1	40	1.25 MW	1	0.0018, 0.0011
2	7	1.00 MW	2	0.0066, 0.0013
3	44	1.50 MW	3	0.0216, 0.0102

Table 7 Uncertainty in the estimated capacity factors (AHF-based wind power potential)

Station	WPP (four-turbine farm)		WPP (nine-turbine farm)	
	Class	Uncertainty (μ, σ)	Class	Uncertainty (μ, σ)
Ada	3	0.7426, 1.3464	3	0.5374, 0.6207
Baker	3	0.7426, 1.3464	3	0.5374, 0.6207
Beach	2	0.0779, 0.0153	1	0.0290, 0.0124
Bottineau	3	0.7426, 1.3464	3	0.5374, 0.6207

6.3.2 Uncertainty in the Wind Power Potential. Four locations are selected to evaluate their wind power potentials. The wind data for these four locations are obtained from the *North Dakota Agricultural Weather Network* [54]. The daily averaged data (wind speed and wind direction) in 2010 were measured and recorded at the four differing stations (Ada, Baker, Beach, and Bottineau). The parameters of the bivariate normal distributions for the four stations are estimated using maximum likelihood estimators.

Table 7 shows the uncertainty in the wind power potential estimations for the four wind farm sites. For the Ada, Baker, and Bottineau stations, we observe that errors of the estimated WPP are

classified into Class 3 for both the four-turbine and the nine-turbine wind farms. For the Beach station, the error of the estimated WPP is classified into Class 2 for the four-turbine wind farm, and is classified into Class 1 for the nine-turbine wind farm. The corresponding mean and standard deviation of the errors are also given in Table 7. Classes 1, 2, and 3 represent the low, medium, and high uncertainty levels, respectively. For the nine-turbine farm at the Ada station, the capacity factor is estimated to be 48.52%. The installed capacity of the nine-turbine farm is 13.5 MW. The 53.74% (mean value) error in the capacity factor estimation results in a discrepancy of approximately 3×10^7 kWh (calculated by $13.5 \times 48.52\% \times 365 \times 24 \times 1000 \times 53.74\%$) of annual energy production, which is significant for such a small-scale wind farm.

7 Conclusion

This paper developed a method to characterize the uncertainty attributable to surrogate models. This method was called the DSUS framework. In this framework, the whole design domain was divided into physically meaningful error classes. SVM was implemented to determine the boundaries between error classes in the design variable space, and to classify any new point/design into the appropriate error class. Such classification of the design-domain based on model errors facilitated more informed decision-making during the design process than possible with the conventional application of surrogates.

The DSUS framework was applied to a series of standard test problems and engineering problems in conjunction with two surrogate modeling methods. The results showed that the DSUS framework could successfully characterize and quantify the uncertainty (predictive errors) in surrogates. The mean errors in the wind farm cost and wind power potential were estimated to be 3.5 million dollars and 3×10^7 kWh of annual energy production, respectively, for specific case studies.

Both global and local error measure metrics were used to assess the accuracy of surrogate models. In this paper, the leave-one-out cross-validation errors at all training points were used as local error measures. When compared with errors estimated on test points in the local neighborhood, the leave-one-out cross-validation error was found to be reasonably accurate in capturing local errors as long as there was a reasonable number of training points. In future research, more accurate measures of local error could be formulated and used in the DSUS framework.

Acknowledgment

Support from the National Science Foundation Awards CMMI-1100948 and CMMI-0946765 is gratefully acknowledged. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] Haldar, A., and Mahadevan, S., 2000, *Probability, Reliability, and Statistical Methods in Engineering Design*, Wiley, New York.
[2] Picheny, V., 2009, "Improving Accuracy and Compensating for Uncertainty in Surrogate Modeling," Ph.D. thesis, Aerospace Engineering, University of Florida, Gainesville, FL.
[3] Keane, A. J., and Nair, P. B., 2005, *Computational Approaches for Aerospace Design: The Pursuit of Excellence*, Wiley, New York.
[4] Myers, R. H., and Montgomery, D. C., 2002, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 2nd ed., Wiley, New York.
[5] Giunta, A. A., and Watson, L., 1998, "A Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models," NASA, Technical Report AIAA-98-4758.
[6] Sakata, S., Ashida, F., and Zako, M., 2003, "Structural Optimization Using Kriging Approximation," Comput. Methods Appl. Mech. Eng., **192**(7–8), pp. 923–939.
[7] Cressie, N., 1993, *Statistics for Spatial Data*, Wiley, New York.
[8] Hardy, R. L., 1971, "Multiquadric Equations of Topography and Other Irregular Surfaces," J. Geophys. Res., **76**, pp. 1905–1915.
[9] Jin, R., Chen, W., and Simpson, T., 2001, "Comparative Studies of Metamodelling Techniques Under Multiple Modelling Criteria," Struct. Multidiscip. Optim., **23**(1), pp. 1–13.
[10] Mullur, A. A., and Messac, A., 2005, "Extended Radial Basis Functions: More Flexible and Effective Metamodeling," AIAA J., **43**(6), pp. 1306–1315.
[11] Duda, R. O., Hart, P. E., and Stork, D. G., 2000, *Pattern Classification*, 2nd ed., Wiley, New York.
[12] Yegnanarayana, B., 2004, *Artificial Neural Networks*, PHI Learning Pvt. Ltd., New Delhi, India.
[13] Clarke, S. M., Griebsch, J. H., and Simpson, T. W., 2005, "Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses," ASME J. Mech. Des., **127**(6), pp. 1077–1087.
[14] Vapnik, V., 1995, *The Nature of Statistical Learning Theory*, Springer, New York.
[15] Basudhar, A., and Missoum, S., 2008, "Adaptive Explicit Decision Functions for Probabilistic Design and Optimization Using Support Vector Machines," Comput. Struct., **86**(19–20), pp. 1904–1917.
[16] Forrester, A. I. J., and Keane, A. J., 2009, "Recent Advances in Surrogate-Based Optimization," Prog. Aerosp. Sci., **45**(1–3), pp. 50–79.
[17] Queipo, N., Haftka, R., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P., 2005, "Surrogate-Based Analysis and Optimization," Prog. Aerosp. Sci., **41**(1), pp. 1–28.
[18] Wang, G., and Shan, S., 2007, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," ASME J. Mech. Des., **129**(4), pp. 370–380.
[19] Simpson, T. W., Toropov, V., Balabanov, V., and Viana, F. A. C., 2008, "Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How Far we Have Come or Not," 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA.
[20] Zerpa, L. E., Queipo, N. V., Pintos, S., and Salager, J., 2005, "An Optimization Methodology of Alkaline-Urfactant-Polymer Flooding Processes Using Field Scale Numerical Simulation and Multiple Surrogates," J. Pet. Sci. Eng., **47**(3–4), pp. 197–208.
[21] Goel, T., Haftka, R. T., Shyy, W., and Queipo, N. V., 2007, "Ensemble of Surrogates," Struct. Multidiscip. Optim., **33**(3), pp. 199–216.
[22] Sanchez, E., Pintos, S., and Queipo, N. V., 2008, "Toward an Optimal Ensemble of Kernel-Based Approximations With Engineering Applications," Struct. Multidiscip. Optim., **36**(3), pp. 247–261.
[23] Acar, E., and Rais-Rohani, M., 2009, "Ensemble of Metamodels With Optimized Weight Factors," Struct. Multidiscip. Optim., **37**(3), pp. 279–294.
[24] Viana, F. A. C., Haftka, R. T., and Steffen, V., 2009, "Multiple Surrogates: How Cross-Validation Errors Can Help us to Obtain the Best Predictor," Struct. Multidiscip. Optim., **39**(4), pp. 439–457.
[25] Apley, D. W., Liu, J., and Chen, W., 2006, "Understanding the Effects of Model Uncertainty in Robust Design With Computer Experiments," ASME J. Mech. Des., **128**(4), pp. 945–958.
[26] Kennedy, M. C., and O'Hagan, A., 2001, "Bayesian Calibration of Computer Models," J. R. Stat. Soc.: Ser. B, **63**(3), pp. 425–464.
[27] Neufeld, D., Behdinan, K., and Chung, J., 2010, "Aircraft Wing Box Optimization Considering Uncertainty in Surrogate Models," Struct. Multidiscip. Optim., **42**(5), pp. 745–753.
[28] Eldred, M., Giunta, A., Wojtkiewicz, S. F., and Trucano, T., 2002, "Formulations for Surrogate-Based Optimization Under Uncertainty," 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA.
[29] Jones, D., Schonlau, M., and Welch, W., 1998, "Efficient Global Optimization of Expensive Black-Box Functions," J. Global Optim., **13**(4), pp. 455–492.
[30] Viana, F. A. C., and Haftka, R. T., 2009, "Importing Uncertainty Estimates From one Surrogate to Another," 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA.
[31] Xiong, Y., Chen, W., and Tsui, K., 2008, "A New Variable-Fidelity Optimization Framework Based on Model Fusion and Objective-Oriented Sequential Sampling," ASME J. Mech. Des., **130**(11), p. 111401.
[32] Chen, S., Xiong, Y., and Chen, W., 2009, "Multiresponse and Multistage Metamodeling Approach for Design Optimization," AIAA J., **47**(1), pp. 206–218.
[33] Huang, D., Allen, T. T., Notz, W. I., and Zeng, N., 2006, "Global Optimization of Stochastic Black-Box Systems Via Sequential Kriging Meta-Models," J. Global Optim., **34**(3), pp. 441–466.
[34] Zhang, J., Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., 2013, "Adaptive Hybrid Surrogate Modeling for Complex Systems," AIAA J., **51**(3), pp. 643–656.
[35] Zhang, J., Chowdhury, S., Messac, A., and Castillo, L., 2012, "A Response Surface-Based Cost Model for Wind Farm Design," Energy Policy, **42**, pp. 538–550.
[36] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., 2012, "Unrestricted Wind Farm Layout Optimization (UWFLO): Investigating Key Factors Influencing the Maximum Power Generation," Renewable Energy, **38**(1), pp. 16–30.
[37] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., 2013, "Optimizing the Arrangement and the Selection of Turbines for Wind Farms Subject to Varying Wind Conditions," Renewable Energy, **52**, pp. 273–282.
[38] Forrester, A. I. J., Sóbester, A., and Keane, A. J., 2008, *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley, New York.
[39] Joseph, V. R., Hung, Y., and Sudjianto, A., 2008, "Blind Kriging: A New Method for Developing Metamodels," ASME J. Mech. Des., **130**(3), p. 031102.
[40] Viana, F. A. C., and Haftka, R. T., 2009, "Cross Validation Can Estimate How Well Prediction Variance Correlates With Error," AIAA J., **47**(9), pp. 2266–2270.
[41] Viana, F. A. C., Picheny, V., and Haftka, R. T., 2010, "Using Cross Validation to Design Conservative Surrogates," AIAA J., **48**(10), pp. 2286–2298.

[42] Duan, K., and Keerthi, S. S., 2005, "Which is the Best Multiclass SVM Method? An Empirical Study," Multiple Classifier Syst., **3541**, pp. 732–760.
[43] Hsu, C. W., and Lin, C. J., 2002, "A Comparison of Methods for Multiclass Support Vector Machines," IEEE Trans. Neural Networks, **13**(2), pp. 415–425.
[44] Chang, C. C., and Lin, C. J., 2011, "LIBSVM: A Library for Support Vector Machines," ACM Trans. Intell. Syst. Technol., **2**(3), pp. 27:1–27:27.
[45] Zhang, J., Chowdhury, S., and Messac, A., 2012, "An Adaptive Hybrid Surrogate Model," Struct. Multidiscip. Optim., **46**(2), pp. 223–238.
[46] Audze, P., and Eglais, V., 1997, "New Approach for Planning Out of Experiments," Prob. Dyn. Strengths, **35**, pp. 104–107.
[47] Lophaven, S. N., Nielsen, H. B., and Sondergaard, J., 2002, Dace—A Matlab Kriging Toolbox, Version 2.0," Technical University of Denmark, Technical Report, Informatics and Mathematical Modelling Report IMM-REP-2002-12.
[48] Zhang, J., Chowdhury, S., Messac, A., and Castillo, L., 2011, "A Comprehensive Measure of the Energy Resource Potential of a Wind Farm Site," ASME 2011 5th International Conference on Energy Sustainability, ASME.
[49] Chowdhury, S., Messac, A., and Khire, R. A., 2011, "Comprehensive Product Platform Planning (cp^3) Framework," ASME J. Mech. Des., **133**(10), p. 101004.
[50] Zhang, J., Chowdhury, S., Messac, A., Castillo, L., and Lebron, J., 2010, "Response Surface Based Cost Model for Onshore Wind Farms Using Extended Radial Basis Functions," ASME 2010 International Design Engineering Technical Conferences (IDETC), ASME.
[51] GE, 2010, *GE Energy 1.5MW Wind Turbine Brochure*, General Electric, http://www.gepower.com/
[52] Chowdhury, S., Messac, A., and Khire, R. A., 2013, "Investigating the Commonality Attributes for Scaling Product Families Using Comprehensive Product Platform Planning (cp3)," Struct. Multidiscip. Optim., **48**, pp. 1089–1107.
[53] Goldberg, M., 2009, *Jobs and Economic Development Impact (JEDI) Model*, National Renewable Energy Laboratory, Golden, CO.
[54] NDSU, 2012, "The North Dakota Agricultural Weather Network," http://ndawn.ndsu.nodak.edu