

Article

Ensemble of Regression-Type and Interpolation-Type Metamodels

Cheng Yan ¹, Jianfeng Zhu ^{1,*}, Xiuli Shen ^{2,*}, Jun Fan ³, Dong Mi ⁴ and Zhengming Qian ⁴

¹ School of Aerospace Engineering, Xiamen University, Xiamen 361005, China; yanchengmail@xmu.edu.cn

² School of Energy and Power Engineering, Beihang University, Beijing 100191, China

³ Army Aviation Institute, Beijing 100000, China; fanjun93404@126.com

⁴ AECC Hunan Aviation Powerplant Research Institute, Zhuzhou 412002, China; midcapi@126.com (D.M.); qianzhengming87@163.com (Z.Q.)

* Correspondence: zhjf@xmu.edu.cn (J.Z.); shxl606@buaa.edu.cn (X.S.)

Received: 21 December 2019; Accepted: 21 January 2020; Published: 4 February 2020



Abstract: Metamodels have become increasingly popular in the field of energy sources because of their significant advantages in reducing the computational cost of time-consuming tasks. Lacking the prior knowledge of actual physical systems, it may be difficult to find an appropriate metamodel in advance for a new task. A favorite way of overcoming this difficulty is to construct an ensemble metamodel by assembling two or more individual metamodels. Motivated by the existing works, a novel metamodeling approach for building the ensemble metamodels is proposed in this paper. By thoroughly exploring the characteristics of regression-type and interpolation-type metamodels, some useful information is extracted from the feedback of the regression-type metamodels to further improve the functional fitting capability of the ensemble metamodels. Four types of ensemble metamodels were constructed by choosing four individual metamodels. Common benchmark problems are chosen to compare the performance of the individual and ensemble metamodels. The results show that the proposed metamodeling approach reduces the risk of selecting the worst individual metamodel and improves the accuracy of the used individual metamodels.

Keywords: metamodel; ensemble; individual; regression; interpolation

1. Introduction

Metamodels, which are also referred to as surrogate models, are essentially approximate mathematical models of real physical systems. In the past decade, metamodels have become increasingly popular in the field of energy sources because of their significant advantages in reducing the computational cost of time-consuming tasks [1,2]. Melo et al. [3] pointed out that researchers in many countries are developing metamodels to estimate the energy performance of the building stock. Bornatico et al. [4] used a kind of metamodel to optimize energy systems, and found that the metamodel converged to the same solution at 150 times the speed of the fine model. Westermann and Evins [5] summarized and discussed recent studies on the application of metamodels in sustainable building design. Ferrero Bermejo et al. [6] reviewed and compared two typical metamodels, namely the artificial neural networks and the support vector machine, for energy forecasting and condition-based maintenance in PV plants.

Actually, a good metamodel mainly depends on its accuracy and generality for different design tasks. To enhance the performance of metamodels, researchers have carried out a lot of studies over the past few decades [7–11]. As a result, a large number of metamodels have been proposed, of which several types have gained wide acceptance in various applications. They are polynomial response surface (PRS) [12–14], support vector regression (SVR) [15–17], radial basis functions (RBF) [18,19],

extended radial basis functions (E-RBF) [20], moving least squares (MLS) [21], artificial neural networks (ANN) [22,23], multivariate adaptive regressive splines (MARS) [24] and Kriging (KRG) [25,26]. These different metamodels give us more options for different tasks. However, lacking the prior knowledge of the actual physical systems, it is challenging to find a suitable metamodel in advance for a new task. In particular, the worst metamodel may be chosen for the task.

A simple way to overcome the difficulty is to build a series of metamodels based on a given training dataset at first, and then select the best one on the basis of some statistical techniques like the cross-validation method. Another favorite way is to construct an ensemble metamodel, which assembles two or more individual metamodels by introducing weight factors. The basic idea of such an ensemble metamodel can be traced back to 1990s [27,28], and currently it has become a research hotspot [8,29]. According to the characteristics of the weight factors, the techniques for building the ensemble metamodels can be mainly categorized into methods based on local errors, methods based on global errors, and methods based on regression.

In the first category, the weight factors ($\omega_i = \omega_i(\mathbf{x})$) are functions of design space, which are determined by the local errors of individual metamodels at the point of interest. Zerpa et al. [30] introduced a local weighted average model for the optimization of alkaline-surfactant-polymer flooding processes by using the prediction variances of three individual metamodels (PRS, KRG, and RBF). Sanchez, Pintos, and Queipo [31] proposed a general approach toward the ensemble of kernel-based models based on the local prediction variances. Acar [32] investigated the efficiency of methods based on the local errors, and developed a new approach to determine the weight factors by using the pointwise cross-validation errors instead of the prediction variances. Zhang, Chowdhury, and Messac [33] proposed a new metamodeling technique called adaptively hybrid functions, whose weight factors are determined based on the local measure of accuracy in the pertinent trust region. Lee and Choi [34] presented a new pointwise ensemble of metamodels, of which the weight factors are calculated by using the v nearest points cross-validation errors.

In the second category, the weight factors ($\omega_i = C_i, \forall \mathbf{x}$) are constant values in the entire design space, which are determined by the global errors of individual metamodels. Goel et al. [35] studied a global weight factor selection approach based on the generalized mean square cross-validation errors (GMSE). Acar and Rais-Rohani [36] developed an accurate ensemble of metamodels by solving an optimization problem that minimizes GMSE or root mean square errors (RMSE). Viana, Haftka, and Steffen [37] obtained the optimal weight factors of the optimization problem by using the Lagrange multipliers. This method was also employed by Toal and Keane [38] to construct an ensemble of ordinary, universal, non-stationary and limit KRG models. Additionally, Acar [39] performed the simultaneous optimization of the weight factors and the shape parameters in the ensemble of RBFs.

It should be noted that in the first two categories the weight factors of individual metamodels are restricted to a positive range ($\omega_i > 0$) and the sum of these factors is equal to 1 ($\sum_{i=1}^M \omega_i = 1$). Since they are different from the first two categories, the techniques in the third category mainly use the regression methods (like least squares) to determine the weight factors. Accordingly, there is no longer any restriction on the weight factors, which may even have negative values. Polynkin and Toropov [40] introduced a novel mid-range metamodel assembly for the large-scale optimization problems, which is constructed based on the linear regression method. Ferreira and Serpa [41] developed an augmented least-square approach for creating the ensemble of metamodels, which can be extended to the efficient global optimization. Zhou and Jiang [42] constructed an ensemble of four individual metamodels (PRS, KRG, SVR, and RBF) from the view of the polynomial regression, and proposed a metamodel selection method on the basis of the stepwise regression to eliminate the redundant ones from the set of the candidate metamodels.

Motivated by these existing works, this paper proposes a different method for constructing the ensemble metamodels, which combines the advantages of regression-type and interpolation-type metamodels. The regression-type metamodels have better global trend fitting capacity than the interpolation-type metamodels, while the interpolation-type metamodels perform better than the

regression-type metamodels in the vicinity of the sampling locations. By thoroughly exploring the characteristics of regression-type and interpolation-type metamodels, the proposed method could extract some useful information from the feedback of the regression-type metamodels to further improve the functional fitting capability of the ensemble metamodels.

2. Proposed Ensemble of Metamodels

2.1. Motivation and Basic Characteristics

The existing individual metamodels can be classified into regression-type and interpolation-type metamodels. The regression-type metamodels aim to fit the global trend of the underlying functions of the real physical systems in the entire design space, while the interpolation-type metamodels aim to achieve the local accuracy in the vicinity of the sampling locations. Accordingly, the regression-type metamodels can build smooth surfaces that pass across all the training points, while the interpolation-type metamodels can construct models that go through each training point. That is to say, for the regression-type metamodels there may be obvious deviations between the actual responses and the approximate responses at the sampling locations, while for the interpolation-type metamodels there is no deviation. These different characteristics make the two types of metamodels possess different advantages and limitations. For example: (i) the regression-type metamodels have better global trend fitting capacity than the interpolation-type metamodels, while (ii) the interpolation-type metamodels perform better than the regression-type metamodels in the vicinity of the sampling locations.

It should be noted that obtaining the training dataset required for constructing the metamodels may be time-consuming. Therefore, as much information as possible should be extracted from these data. However, for the regression-type metamodels, there are apparent deviations between the actual responses and the approximate responses at the sampling locations, from where some useful information may be still extracted to further improve the performance of these metamodels. Exploring the underlying knowledge of the training dataset and combining the characteristics of regression-type and interpolation-type metamodels, this paper proposes a novel metamodeling approach for the ensemble metamodels. The flowchart of the proposed metamodeling technique is shown in Figure 1, which involves four main steps as follows.

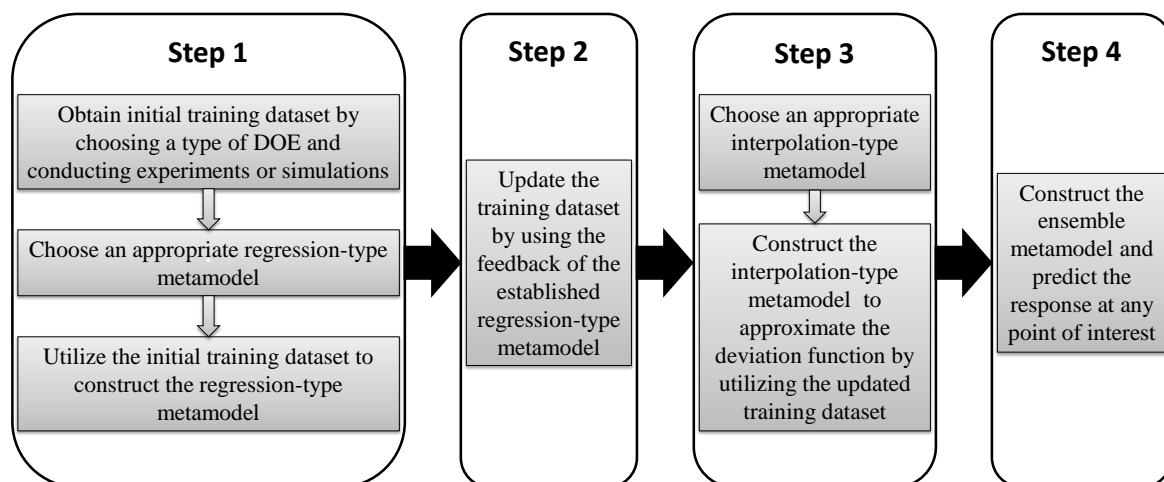


Figure 1. Flowchart of the proposed approach for building ensembles of regression-type and interpolation-type metamodels.

Step 1: An appropriate design of experiment (DOE) should be first chosen to generate n sampling locations (x^1, x^2, \dots, x^n) , at where the actual responses (y^1, y^2, \dots, y^n) are obtained by conducting experiments or simulations. By using the initial training dataset (x^i, y^i) ($i =$

$1, \dots, n$), a regression-type metamodel $\hat{y}_1(\mathbf{x})$ in Equation (1) is subsequently constructed to approximate the actual model $y(\mathbf{x})$.

$$\hat{y}_1(\mathbf{x}) \approx y(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, \dots, x_k)^T \quad (1)$$

where \mathbf{x} denotes any point of interest.

Step 2: We suppose that there is a deviation function $y_d(\mathbf{x})$. It is obtained by subtracting the approximate model $\hat{y}_1(\mathbf{x})$ from the actual model $y(\mathbf{x})$.

$$y_d(\mathbf{x}) = y(\mathbf{x}) - \hat{y}_1(\mathbf{x}) \quad (2)$$

Some useful information may be still extracted from the deviation function $y_d(\mathbf{x})$. To approximate the deviation function, the training dataset should be updated. In detail, this paper first uses the established regression-type metamodel in Equation (1) to predict the approximate responses $(\hat{y}_1^1, \hat{y}_1^2, \dots, \hat{y}_1^n)$ at the initial sampling locations. Subsequently, the deviations $(y_d^1, y_d^2, \dots, y_d^n)$ between the actual responses and approximate responses at these locations are calculated as the updated training dataset.

$$\begin{aligned} & \left\{ (\mathbf{x}^1, y_d^1), (\mathbf{x}^2, y_d^2), \dots, (\mathbf{x}^n, y_d^n) \right\} = \\ & \left\{ (\mathbf{x}^1, y^1 - \hat{y}_1^1), (\mathbf{x}^2, y^2 - \hat{y}_1^2), \dots, (\mathbf{x}^n, y^n - \hat{y}_1^n) \right\} \end{aligned} \quad (3)$$

Step 3: By using the updated training dataset in Equation (3), an interpolation-type metamodel $\hat{y}_2(\mathbf{x})$ in Equation (4) is constructed to approximate the deviation function $y_d(\mathbf{x})$.

$$\hat{y}_2(\mathbf{x}) \approx y_d(\mathbf{x}) \quad (4)$$

Step 4: Finally, the ensemble metamodel $\hat{y}_{ens}(\mathbf{x})$ in Equation (5) is constructed by adding the established regression-type metamodel $\hat{y}_1(\mathbf{x})$ and interpolation-type metamodel $\hat{y}_2(\mathbf{x})$ together. By using Equations (1), (4) and (5), the established ensemble metamodel $\hat{y}_{ens}(\mathbf{x})$ can be used to predict the response at any point of interest in the entire design space.

$$\hat{y}_{ens}(\mathbf{x}) = \hat{y}_1(\mathbf{x}) + \hat{y}_2(\mathbf{x}) \approx \hat{y}_1(\mathbf{x}) + y_d(\mathbf{x}) \approx y(\mathbf{x}) \quad (5)$$

2.2. Detailed Modeling Process

To clearly illustrate the proposed metamodeling technique, this paper selects two common regression-type metamodels (PRS and SVR) and two popular interpolation-type metamodels, namely RBFM (RBF with multiquadric-form basis function) and RBFI (RBF with inverse multiquadric-form basis function). Accordingly, four types of ensemble metamodels can be obtained, which are PrsRbfm (Ensemble Scheme 1, ensemble of PRS and RBFM), PrsRbfi (Ensemble Scheme 2, ensemble of PRS and RBFI), SvrRbfm (Ensemble Scheme 3, ensemble of SVR and RBFM) and SvrRbfi (Ensemble Scheme 4, ensemble of SVR and RBFI). The detailed modeling processes of these involved metamodels are introduced as follows.

2.2.1. Step 1: Construction of Regression-Type Metamodels

PRS is a general designation of a series of polynomial regression functions, of which the most popular one is the second-order polynomial model. This paper adopts the second-order polynomial model $\hat{y}_{1,prs}(\mathbf{x})$, which can be written as

$$\hat{y}_{1,prs}(\mathbf{x}) = \mathbf{z}^T \boldsymbol{\beta} = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \sum_{j=i}^k \beta_{\frac{2j+i(2k-i+1)}{2}} x_i x_j \tag{6}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_{\frac{k^2+3k}{2}})^T$ denotes a coefficient vector, $\mathbf{z} = (1, x_1, x_2, \dots, x_{k-1}x_k, x_kx_k)^T$ denotes a polynomial basis-function vector.

To estimate $\boldsymbol{\beta}$, the regression problem in Equation (6) can be transformed as follows by using the initial training dataset.

$$\begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix} = \begin{bmatrix} y_{d,prs}^1 \\ y_{d,prs}^2 \\ \vdots \\ y_{d,prs}^n \end{bmatrix} + \begin{bmatrix} 1 & x_1^1 & \dots & x_k^1 & \dots & x_1^1 x_1^1 & \dots & x_{k-1}^1 x_k^1 & x_k^1 x_k^1 \\ 1 & x_1^2 & \dots & x_k^2 & \dots & x_1^2 x_1^2 & \dots & x_{k-1}^2 x_k^2 & x_k^2 x_k^2 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 1 & x_1^n & \dots & x_k^n & \dots & x_1^n x_1^n & \dots & x_{k-1}^n x_k^n & x_k^n x_k^n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{\frac{k^2+3k}{2}} \end{bmatrix} \tag{7}$$

where $\mathbf{y}_{d,prs} = (y_{d,prs}^1, y_{d,prs}^2, \dots, y_{d,prs}^n)^T$ denotes the deviation vector.

Equation (7) can be also expressed as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{y}_{d,prs} \tag{8}$$

According to the least squares method, $\boldsymbol{\beta}$ can be calculated as follows.

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \tag{9}$$

SVR is a regression function $\hat{y}_{1,svr}(\mathbf{x})$ in the high-dimensional space, as shown in Equation (10).

$$\hat{y}_{1,svr}(\mathbf{x}) = \boldsymbol{\omega}^T \boldsymbol{\psi}(\mathbf{x}) + b \tag{10}$$

where $\boldsymbol{\omega}$ denotes the weight vector, $\boldsymbol{\psi}(\mathbf{x})$ denotes the mapping function, and b denotes the bias.

To estimate $\boldsymbol{\omega}$ and b , the regression problem in Equation (10) can be transformed as an optimization problem in Equation (11) by introducing ϵ -insensitive loss function.

$$\begin{aligned} \min \quad & \frac{1}{2} \|\boldsymbol{\omega}\|^2 \\ \text{subject to} \quad & \begin{cases} \boldsymbol{\omega}^T \boldsymbol{\psi}(\mathbf{x}^i) + b - y^i \leq \epsilon \\ y^i - \boldsymbol{\omega}^T \boldsymbol{\psi}(\mathbf{x}^i) - b \leq \epsilon \\ i = 1, \dots, n \end{cases} \end{aligned} \tag{11}$$

To solve Equation (11), the regularization parameter, $C (> 0)$, and the slack variables, $\xi^{+(i)}$ and $\xi^{-(i)}$, are introduced. In addition, Equation (12) can be obtained

$$\begin{aligned} \min \quad & \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n (\xi^{+(i)} + \xi^{-(i)}) \\ \text{subject to} \quad & \begin{cases} \omega^T \psi(\mathbf{x}^i) + b - y^i \leq \epsilon + \xi^{+(i)} \\ y^i - \omega^T \psi(\mathbf{x}^i) - b \leq \epsilon + \xi^{-(i)} \\ \xi^{+(i)}, \xi^{-(i)} \geq 0 \\ i = 1, \dots, n \end{cases} \end{aligned} \quad (12)$$

The Lagrange dual model of Equation (12) can be expressed as

$$\begin{aligned} \max \quad & \begin{cases} -\frac{1}{2} \sum_{i,j=1}^n (\alpha^{+(i)} - \alpha^{-(i)})(\alpha^{+(j)} - \alpha^{-(j)}) \\ k \langle \mathbf{x}^i, \mathbf{x}^j \rangle + \sum_{i=1}^n (\alpha^{+(i)} - \alpha^{-(i)}) y^i \\ - \sum_{i=1}^n (\alpha^{+(i)} + \alpha^{-(i)}) \epsilon \end{cases} \\ \text{subject to} \quad & \begin{cases} \sum_{i=1}^n (\alpha^{+(i)} - \alpha^{-(i)}) = 0 \\ 0 \leq \alpha^{+(i)}, \alpha^{-(i)} \leq C \\ i = 1, \dots, n \end{cases} \end{aligned} \quad (13)$$

where $\alpha^{+(i)}$ and $\alpha^{-(i)}$ denote the Lagrange multipliers, $k \langle \mathbf{x}^i, \mathbf{x}^j \rangle = \psi(\mathbf{x}^i)^T \psi(\mathbf{x}^j)$ denotes a kernel function, which has several different forms. This paper chooses the Gaussian kernel function, which can be expressed as

$$k \langle \mathbf{x}, \mathbf{x}^i \rangle = \exp(-\gamma \|\mathbf{x} - \mathbf{x}^i\|^2) \quad (14)$$

According to Equation (13), $\alpha^{+(i)}$ and $\alpha^{-(i)}$ can be first obtained. According to KKT conditions [43], ω and b can be then calculated.

2.2.2. Step 2: Update of Training Dataset

First, β calculated by Equation (9) can be used to substitute the one in Equation (6). Second, the approximate responses of established PRS ($\hat{y}_{1,prs}^1, \hat{y}_{1,prs}^2, \dots, \hat{y}_{1,prs}^n$) at the initial sampling locations ($\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^n$) can be calculated according to Equation (6). Then, the updated training dataset of PRS can be expressed as

$$\begin{aligned} & \left\{ (\mathbf{x}^1, y_{d,prs}^1), (\mathbf{x}^2, y_{d,prs}^2), \dots, (\mathbf{x}^n, y_{d,prs}^n) \right\} = \\ & \left\{ (\mathbf{x}^1, y^1 - \hat{y}_{1,prs}^1), (\mathbf{x}^2, y^2 - \hat{y}_{1,prs}^2), \dots, (\mathbf{x}^n, y^n - \hat{y}_{1,prs}^n) \right\} \end{aligned} \quad (15)$$

Similarly, according to Equation (10), the updated training dataset of SVR can be obtained and expressed as

$$\begin{aligned} & \left\{ (\mathbf{x}^1, y_{d,svr}^1), (\mathbf{x}^2, y_{d,svr}^2), \dots, (\mathbf{x}^n, y_{d,svr}^n) \right\} = \\ & \left\{ (\mathbf{x}^1, y^1 - \hat{y}_{1,svr}^1), (\mathbf{x}^2, y^2 - \hat{y}_{1,svr}^2), \dots, (\mathbf{x}^n, y^n - \hat{y}_{1,svr}^n) \right\} \end{aligned} \quad (16)$$

2.2.3. Step 3: Construction of Interpolation-Type Metamodels

The general form of RBF can be expressed as

$$\hat{y}_{rbf}(\mathbf{x}) = \sum_{i=1}^n \lambda_i \phi(\|\mathbf{x} - \mathbf{x}^i\|) \quad (17)$$

where λ_i denotes an interpolation coefficient, $r = \|\mathbf{x} - \mathbf{x}^i\| = \sqrt{(\mathbf{x} - \mathbf{x}^i)^T(\mathbf{x} - \mathbf{x}^i)}$ denotes the distance between points \mathbf{x} and \mathbf{x}^i . $\phi(r)$ denotes a radially symmetric basis function, which has several different forms, such as:

- Gaussian $\phi(r) = e^{(-r^2/c^2)}$
- Multiquadric $\phi(r) = (r^2 + c^2)^{\frac{1}{2}}$
- Inverse multiquadric $\phi(r) = (r^2 + c^2)^{-\frac{1}{2}}$
- Thin plate spline $\phi(r) = (r^2) \log(r)$

The interpolation coefficient λ_i can be calculated by using the given training dataset (\mathbf{x}^i, y^i) ($i = 1, \dots, n$).

$$\boldsymbol{\lambda} = \mathbf{A}^{-1} \mathbf{y} \quad (18)$$

where

$$\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_n]^T$$

$$\mathbf{A} = \begin{bmatrix} \phi(\|\mathbf{x}^1 - \mathbf{x}^1\|), & \phi(\|\mathbf{x}^1 - \mathbf{x}^2\|) & \dots & \phi(\|\mathbf{x}^1 - \mathbf{x}^n\|) \\ \phi(\|\mathbf{x}^2 - \mathbf{x}^1\|), & \phi(\|\mathbf{x}^2 - \mathbf{x}^2\|) & \dots & \phi(\|\mathbf{x}^2 - \mathbf{x}^n\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}^n - \mathbf{x}^1\|), & \phi(\|\mathbf{x}^n - \mathbf{x}^2\|) & \dots & \phi(\|\mathbf{x}^n - \mathbf{x}^n\|) \end{bmatrix}$$

After choosing the multiquadric-form basis function, **RBFM** ($\hat{y}_{rbfm}(\mathbf{x})$) can be constructed to approximate the actual model $y(\mathbf{x})$ by replacing $\hat{y}_{rbf}(\mathbf{x})$ and λ_i in Equation (17) with $\hat{y}_{rbfm}(\mathbf{x})$ and $\lambda_{i,rbfm}$. The coefficient $\lambda_{i,rbfm}$ can be calculated based on Equation (18). Similarly, after choosing the inverse multiquadric-form basis function, **RBFi** ($\hat{y}_{rbfi}(\mathbf{x})$) can be constructed to approximate the actual model $y(\mathbf{x})$. The coefficient $\lambda_{i,rbfi}$ of $\hat{y}_{rbfi}(\mathbf{x})$ can be calculated based on Equation (18).

Additionally, by choosing the multiquadric-form basis function, a model $\hat{y}_{2,rbfm1}(\mathbf{x})$ can be constructed to approximate the deviation function of PRS $\mathbf{y}_{d,prs}$. By replacing the initial training dataset (\mathbf{x}^i, y^i) ($i = 1, \dots, n$) with the updated training dataset of PRS $(\mathbf{x}^i, y_{d,prs}^i)$ ($i = 1, \dots, n$), the coefficient $\lambda_{i,2,rbfm1}$ of $\hat{y}_{2,rbfm1}(\mathbf{x})$ can be calculated on the basis of Equation (18). Similarly, by choosing the inverse multiquadric-form basis function, a model $\hat{y}_{2,rbfi1}(\mathbf{x})$ can be constructed to approximate the deviation function of PRS $\mathbf{y}_{d,prs}$.

Finally, by choosing the multiquadric-form basis function, a model $\hat{y}_{2,rbfm2}(\mathbf{x})$ can be constructed to approximate the deviation function of SVR $\mathbf{y}_{d,svr}$. By choosing the interpolation-type metamodel, a model $\hat{y}_{2,rbfi2}(\mathbf{x})$ can be constructed to approximate the deviation function of SVR $\mathbf{y}_{d,svr}$.

2.2.4. Step 4: Construction of Ensemble Metamodels

By adding the established $\hat{y}_{1,prs}(\mathbf{x})$ and $\hat{y}_{2,rbfm1}(\mathbf{x})$ together, **PrsRbfm** ($\hat{y}_{prsrbfm}(\mathbf{x})$) can be subsequently constructed as follows.

$$\hat{y}_{prsrbfm}(\mathbf{x}) = \hat{y}_{1,prs}(\mathbf{x}) + \hat{y}_{2,rbfm1}(\mathbf{x}) \quad (19)$$

Being similar to PrsRbfm, **PrsRbfi** ($\hat{y}_{prsrbfi}(\mathbf{x})$) can be constructed as follows.

$$\hat{y}_{prsrbfi}(\mathbf{x}) = \hat{y}_{1,prs}(\mathbf{x}) + \hat{y}_{2,rbfi1}(\mathbf{x}) \quad (20)$$

SvrRbfm ($\hat{y}_{svrrbfm}(\mathbf{x})$) can be constructed as follows.

$$\hat{y}_{svrrbfm}(\mathbf{x}) = \hat{y}_{1,svr}(\mathbf{x}) + \hat{y}_{2,rbfm2}(\mathbf{x}) \quad (21)$$

SvrRbfi ($\hat{y}_{svrrbfi}(\mathbf{x})$) can be constructed as follows.

$$\hat{y}_{svrrbfi}(\mathbf{x}) = \hat{y}_{1,svr}(\mathbf{x}) + \hat{y}_{2,rbfi2}(\mathbf{x}) \quad (22)$$

The established ensemble metamodels, namely PrsRbfm, PrsRbfi, SvrRbfm, and SvrRbfi, can be used to predict the response at any point of interest in the entire design space by using Equations (19)–(22).

3. Numerical Experiments

3.1. Benchmark Problems

Referred to the website (<http://www.sfu.ca/~ssurjano/index.html>) and relevant literature [32], six common benchmark problems (BPs) are selected to compare the performance of the individual metamodels (PRS, SVR, RBFM, and RBFI) and the ensemble metamodels (PrsRbfm, PrsRbfi, SvrRbfm, and SvrRbfi).

BP1: Goldstein Price Function

$$f(\mathbf{x}) = \left[1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \right] \quad (23)$$

where $x_i \in [-2, 2]$, for $i = 1, 2$.

BP2: Friedman Function

$$f(\mathbf{x}) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \quad (24)$$

where $x_i \in [0, 1]$, for all $i = 1, \dots, 5$.

BP3: Power Sum Function

$$f(\mathbf{x}) = \sum_{j=1}^6 \left[\left(-\sum_{i=1}^6 x_i^j \right) - 36 \right]^2 \quad (25)$$

where $x_i \in [0, 6]$, for all $i = 1, \dots, 6$.

BP4: Rosenbrock Function

$$f(\mathbf{x}) = \sum_{i=1}^6 \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right] \quad (26)$$

where $x_i \in [-5, 10]$, for all $i = 1, \dots, 7$.

BP5: Zakharov Function

$$f(\mathbf{x}) = \sum_{i=1}^9 x_i^2 + \left(\sum_{i=1}^9 0.5ix_i \right)^2 + \left(\sum_{i=1}^9 0.5ix_i \right)^4 \quad (27)$$

where $x_i \in [-5, 10]$, for all $i = 1, \dots, 9$.

BP6: Powell Function

$$f(\mathbf{x}) = \sum_{i=1}^2 \left[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4 \right] \quad (28)$$

where $x_i \in [-4, 5]$, for all $i = 1, \dots, 10$.

3.2. Numerical Setting

For all the benchmark problems, the MATLAB routine “lhsdesign” is used to generate training points and test points. Referred to Jin, Chen, and Simpson [44], $n = \frac{3(k+1)(k+2)}{2}$ training points are selected for a k -dimension problem. Moreover, as many test points as possible should be used in practice, since insufficient test points may increase the uncertainty of the results. This paper selects $n_{tst} = 20,000$ test points for each benchmark problem. Since the DOE sampling scheme may have an obvious influence on the performance of the metamodels, 100 different training and test sets are selected for each problem. The detailed numerical settings for all the benchmark problems are listed in Table 1. The shape parameters (c) of RBFM and RBFI are both selected as 1 by referring to relevant literature [34,45,46]. The parameters (ϵ , C , and γ) of SVR are selected by using the cross-validation method, which was introduced in detail in the published paper of the authors [47].

Table 1. Detailed numerical settings for the benchmark problems.

Benchmark Problem	NO. of Variables	NO. of Training Points	NO. of Test Points	NO. of Training and Test Sets
BP1	2	18	20,000	100
BP2	5	63	20,000	100
BP3	6	84	20,000	100
BP4	7	108	20,000	100
BP5	9	165	20,000	100
BP6	10	198	20,000	100

3.3. Performance Criteria

The root mean square error (RMSE) and the max absolute error (MAE) are selected as the performance criteria.

RMSE can be expressed as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n_{tst}} (y^i - \hat{y}^i)^2}{n_{tst}}} \quad (29)$$

where n_{tst} denotes the number of test points.

MAE can be expressed as

$$MAE = \max |y_i - \hat{y}_i|, \quad i = 1, 2, \dots, n_{tst} \quad (30)$$

4. Results and Discussion

4.1. RMSE

Figure 2 shows the boxplots of RMSE of the metamodels over 100 test sets for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points. It can be seen that: (1) for all the benchmark problems, the most accurate ensemble metamodels outperform the most accurate individual metamodels; (2) without exception, the least accurate individual metamodels perform worse than the least accurate ensemble metamodels; (3) for each benchmark problem, the performance differences among the four individual metamodels are greater than that among the four ensemble metamodels.

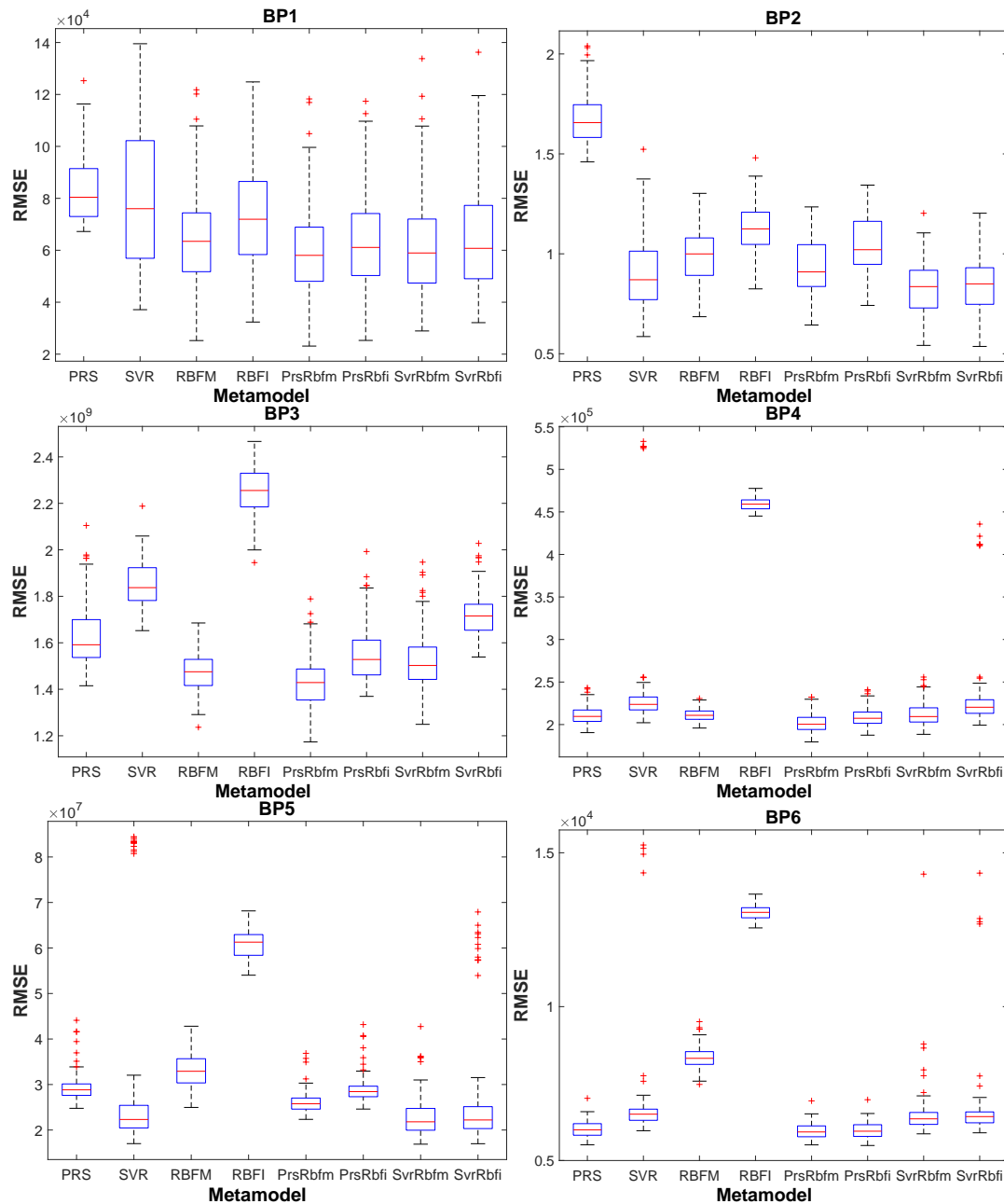


Figure 2. Boxplots of RMSE of the metamodels over 100 test sets for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points.

To provide a better comparison for these metamodels, the error values are normalized with respect to the most accurate individual metamodel for each benchmark problem. Table 2 shows the normalized means of RMSE of the metamodels for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points. The bold values in Table 2 are the most accurate individual/ensemble metamodels, the italic values are the least accurate individual/ensemble metamodels, the underlined values are the ensemble metamodels that perform better than all the individual metamodels, the “Best & Best” values denote the differences between the most accurate ensemble metamodels and individual metamodels, and the “Worst & Worst” values denote the differences between the least accurate ensemble metamodels and individual metamodels. From Table 2, it can be seen that: (1) compared with the most accurate individual metamodels, the means of RMSE of the most accurate ensemble metamodels are reduced, ranging from 1.1% to 22.2%; (2) compared with the least accurate individual metamodels, the means of RMSE of the least accurate ensemble metamodels are reduced, ranging from 21.1% to 52.5%; (3) except

for BP3, more than two ensemble metamodels perform better than the most accurate individual metamodels; (4) for BP5, all the four ensemble metamodels perform better than the most accurate individual metamodel.

Table 2. Normalized means of RMSE of the metamodels for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points.

BPs	BP1	BP2	BP3	BP4	BP5	BP6
PRS	1.280	1.866	1.113	1.000	1.000	1.000
SVR	1.224	1.000	1.262	1.149	1.006	1.133
RBFM	1.000	1.108	1.000	1.001	1.123	1.385
RBFI	1.133	1.261	1.536	2.175	2.073	2.166
PrsRbfm	0.929	1.043	0.981	0.957	0.889	0.989
PrsRbfi	0.977	1.173	1.062	0.990	0.985	0.994
SvrRbfm	0.968	0.922	1.039	1.006	0.778	1.080
SvrRbfi	1.010	0.937	1.176	1.102	0.910	1.109
Best & Best	−7.1%	−7.8%	−1.9%	−4.3%	−22.2%	−1.1%
Worst & Worst	−21.1%	−37.1%	−23.5%	−49.3%	−52.5%	−48.8%

Table 3 shows the frequency of the accuracy ranking (using RMSE) of the metamodels for the six benchmark problems with $\frac{3(k+1)(k+2)}{2}$ training points. It can be seen that: (1) the frequency of the ensemble metamodels that rank 1st or 2nd is 11, yet the frequency of the individual metamodels is only one; (2) the frequency of the individual metamodels that rank 7th or 8th is 12, yet the frequency of the ensemble metamodels is zero; (3) considered the frequency of the metamodels that rank the top/bottom two, all the ensemble metamodels have better performance than the individual metamodels; (4) PrsRbfm performs best among the four ensemble metamodels, followed by SvrRbfm, PrsRbfi, and SvrRbfi.

Table 3. Frequency of the accuracy ranking (using RMSE) of the metamodels for the six benchmark problems with $\frac{3(k+1)(k+2)}{2}$ training points.

Ranking	1st	2nd	3rd	4th	5th	6th	7th	8th
PRS	0	0	2	0	2	0	0	2
SVR	0	0	1	0	0	2	3	0
RBFM	0	1	0	2	1	0	2	0
RBFI	0	0	0	0	0	1	1	4
Total	0	1	3	2	3	3	6	6
PrsRbfm	4	1	0	1	0	0	0	0
PrsRbfi	0	2	1	2	0	1	0	0
SvrRbfm	2	1	1	1	1	0	0	0
SvrRbfi	0	1	1	0	2	2	0	0
Total	6	5	3	4	3	3	0	0

To clearly compare the accuracy of each ensemble metamodel with their corresponding individual metamodels, Figure 3 shows the normalized means of RMSE of each ensemble scheme for the six benchmark problems with $\frac{3(k+1)(k+2)}{2}$ training points. It can be seen that: (1) in Scheme 1, PrsRbfm ranks 1st among PRS, RBFM, and PrsRbfm for all the benchmark problems; (2) in Scheme 2, PrsRbfi ranks 1st for all the benchmark problems; (3) in Scheme 3, SvrRbfm ranks 1st for four benchmark problems and 2nd for two benchmark problems; although RBFM ranks 1st for two benchmark problems, it is the worst performer for three benchmark problems; (4) in Scheme 4, without exception, the accuracy of SvrRbfi outperforms that of SVR and RBFI.

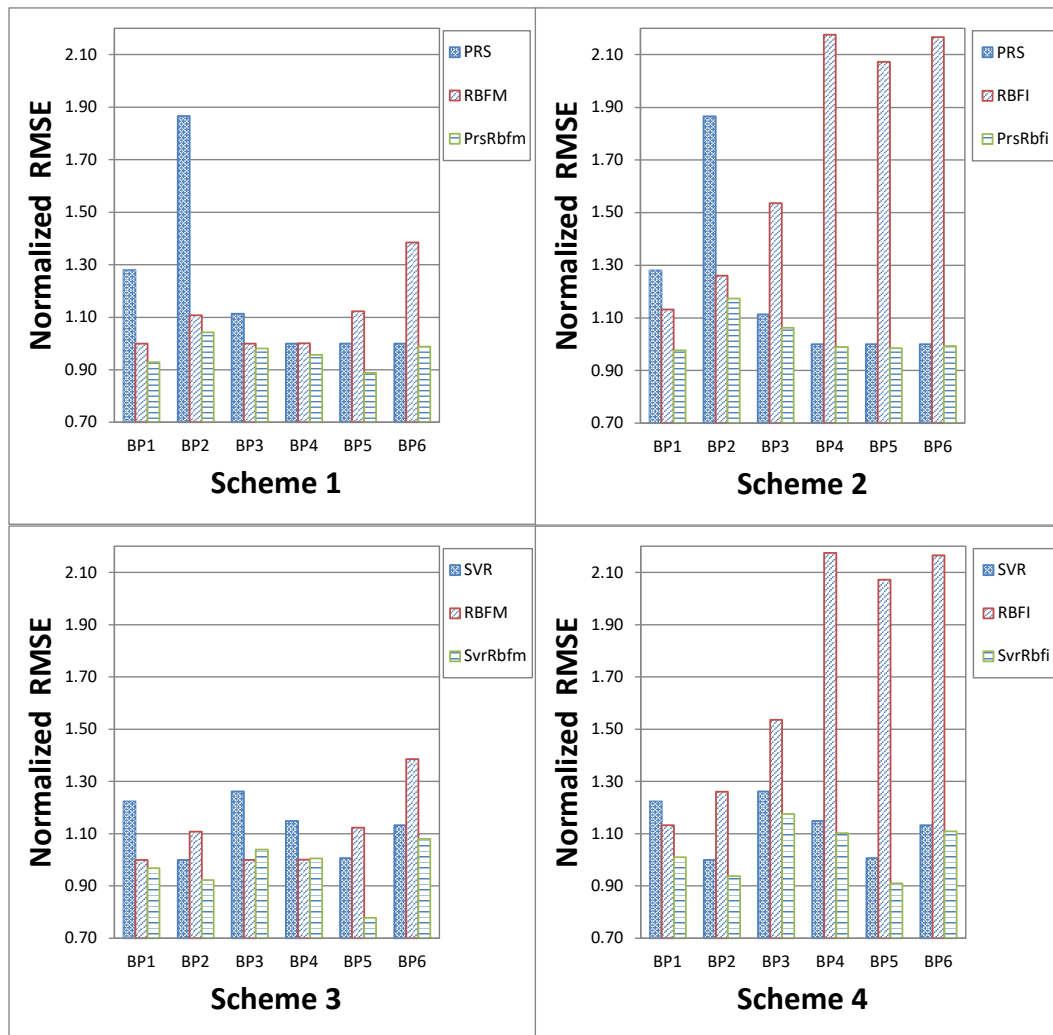


Figure 3. Normalized means of RMSE of each ensemble scheme for the six benchmark problems with $\frac{3(k+1)(k+2)}{2}$ training points.

Table 4 shows the normalized standard deviations of RMSE of the metamodels for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points. It can be seen that: (1) compared with the most accurate individual metamodels, the standard deviations of RMSE of the most accurate ensemble metamodels are reduced for BP5 and BP6, yet the standard deviations are increased for the other four benchmark problems; (2) compared with the least accurate individual metamodels, the standard deviations of RMSE of the least accurate ensemble metamodels are reduced, ranging from 8.4% to 35.5%.

According to the above experimental results, we think the proposed metamodeling approach could reduce the risk of selecting the worst individual metamodel, and the constructed ensemble metamodels perform better than the used individual metamodels in terms of accuracy. In particular, PrsRbfm performs best among the four ensemble metamodels, followed by SvrRbfm, PrsRbfi, and SvrRbfi.

To provide an explicit explanation for the better performance of the proposed approach, a low-dimensional problem (BP1) and an ensemble scheme (ensemble of SVR and RBFM) are selected as examples. Figure 4 shows the contour plot of the actual function and the approximate functions of SVR, RBFM, and SvrRbfm. It can be seen that: (1) SVR has better global trend fitting capacity than RBFM, such as in the red box area; (2) RBFM performs better in the vicinity of the sampling locations, such as in the red ellipse region; (3) SvrRbfm combines the global trend of SVR and the local accuracy of RBFM, such as in the red box area and the red ellipse region.

Table 4. Normalized standard deviations of RMSE of the metamodels for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points.

BPs	BP1	BP2	BP3	BP4	BP5	BP6
PRS	1.000	1.000	1.646	1.566	1.001	1.072
SVR	2.067	1.542	1.121	10.663	6.188	6.844
RBFM	1.462	1.167	1.000	1.000	1.204	1.488
RBFI	1.660	1.100	1.235	1.117	1.000	1.000
PrsRbfm	1.397	1.185	1.304	1.592	0.744	0.995
PrsRbfi	1.423	1.193	1.493	1.577	0.953	1.055
SvrRbfm	1.581	1.143	1.509	2.021	1.427	3.714
SvrRbfi	1.696	1.199	1.123	7.041	3.991	5.542
Best & Best	39.7%	14.3%	12.3%	57.7%	−25.6%	−0.5%
Worst & Worst	−17.9%	−22.2%	−8.4%	−34.0%	−35.5%	−19.0%

Therefore, the reason for the better performance of the ensemble metamodels may be that the proposed metamodeling approach combines the advantages of the regression-type and interpolation-type metamodels. The actual model is regarded as the sum of a regression-type model and a deviation function. Some useful information is first extracted by the regression-type metamodel to capture the global trend of the actual model in the entire design space. Then, some other information is extracted from the deviations at the sampling locations by using the interpolation-type metamodel to achieve the local accuracy in the vicinity of sampling locations.

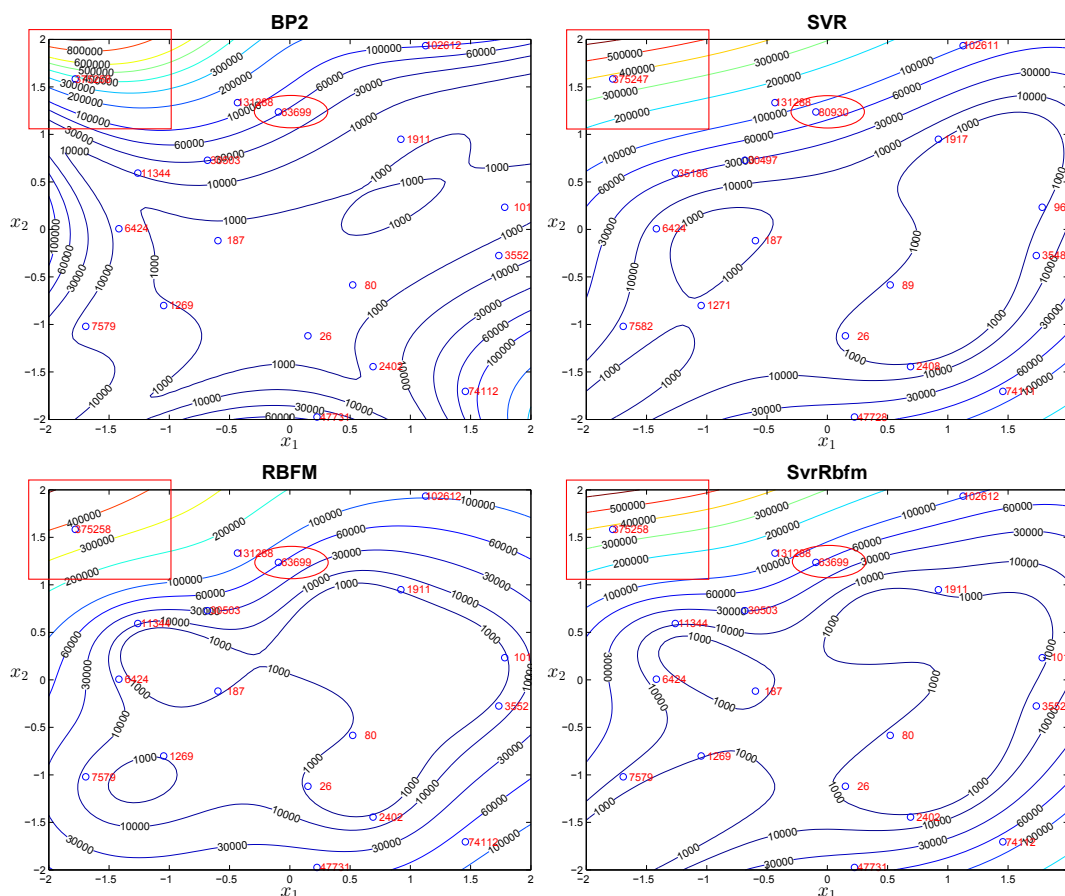


Figure 4. Contour plot of the actual function and the approximate functions of SVR, RBFM, and SvrRbfm.

4.2. Effect of Performance Criteria

The choice of different performance criteria may influence the results of the metamodels. To reduce the source of uncertainty in the results as much as possible, the max absolute error (MAE) is selected as another performance criterion.

Figure 5 shows the boxplots of MAE of the metamodels over 100 test sets for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points. Table 5 shows the normalized means of MAE of the metamodels for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points. From Figure 5 and Table 5, it can be seen that: (1) for each benchmark problem, the performance differences among the four ensemble metamodels are less than that among the four individual metamodels; (2) except for BP6, more than two ensemble metamodels perform better than the most accurate individual metamodels; (3) compared with the most accurate individual metamodels, the means of MAE of the most accurate ensemble metamodels are reduced for five benchmark problems; (4) compared with the least accurate individual metamodels, the means of MAE of the least accurate ensemble metamodels are reduced, ranging from 14.2% to 48.9%.

Table 5. Normalized means of MAE of the metamodels for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points.

BPs	BP1	BP2	BP3	BP4	BP5	BP6
PRS	1.055	1.583	1.000	1.000	1.000	1.000
SVR	1.149	1.000	1.405	1.312	1.080	1.325
RBFM	1.000	1.148	1.165	1.174	1.385	1.708
RBFI	1.189	1.200	1.626	2.515	1.939	2.353
PrsRbfm	0.910	1.111	0.952	0.980	0.944	1.015
PrsRbfi	<u>0.965</u>	1.278	<u>0.999</u>	<u>0.999</u>	<u>0.996</u>	1.002
SvrRbfm	<u>0.950</u>	0.933	1.164	1.168	<u>0.957</u>	1.244
SvrRbfi	1.021	<u>0.956</u>	1.338	1.285	1.052	1.302
Best & Best	−9.0%	−6.7%	−4.8%	−2.0%	−5.6%	0.2%
Worst & Worst	−14.2%	−19.3%	−17.7%	−48.9%	−45.7%	−44.7%

Table 6 shows the frequency of the accuracy ranking (using MAE) of the metamodels for the six benchmark problems with $\frac{3(k+1)(k+2)}{2}$ training points. It can be seen that: (1) considered the frequency of the metamodels that rank the top/bottom two, PrsRbfm, PrsRbfi, and SvrRbfm outperform all the individual metamodels; (2) although SvrRbfi is a little worse than PRS, it still performs better than its corresponding individual metamodels (SVR and RBFI); (3) PrsRbfm is the best performer of the four ensemble metamodels, followed by SvrRbfm, PrsRbfi, and SvrRbfi.

Table 6. Frequency of the accuracy ranking (using MAE) of the metamodels for the six benchmark problems with $\frac{3(k+1)(k+2)}{2}$ training points.

Ranking	1st	2nd	3rd	4th	5th	6th	7th	8th
PRS	1	0	2	1	0	1	0	1
SVR	0	0	1	0	0	2	3	0
RBFM	0	0	0	1	3	0	2	0
RBFI	0	0	0	0	0	1	0	5
Total	1	0	3	2	3	4	5	6
PrsRbfm	4	0	1	1	0	0	0	0
PrsRbfi	0	3	2	0	0	0	1	0
SvrRbfm	1	2	0	3	0	0	0	0
SvrRbfi	0	1	0	0	3	2	0	0
Total	5	6	3	4	3	2	1	0

In summary, the choice of the performance criteria influence the results slightly, but the conclusions obtained by the two criteria remain unchanged.

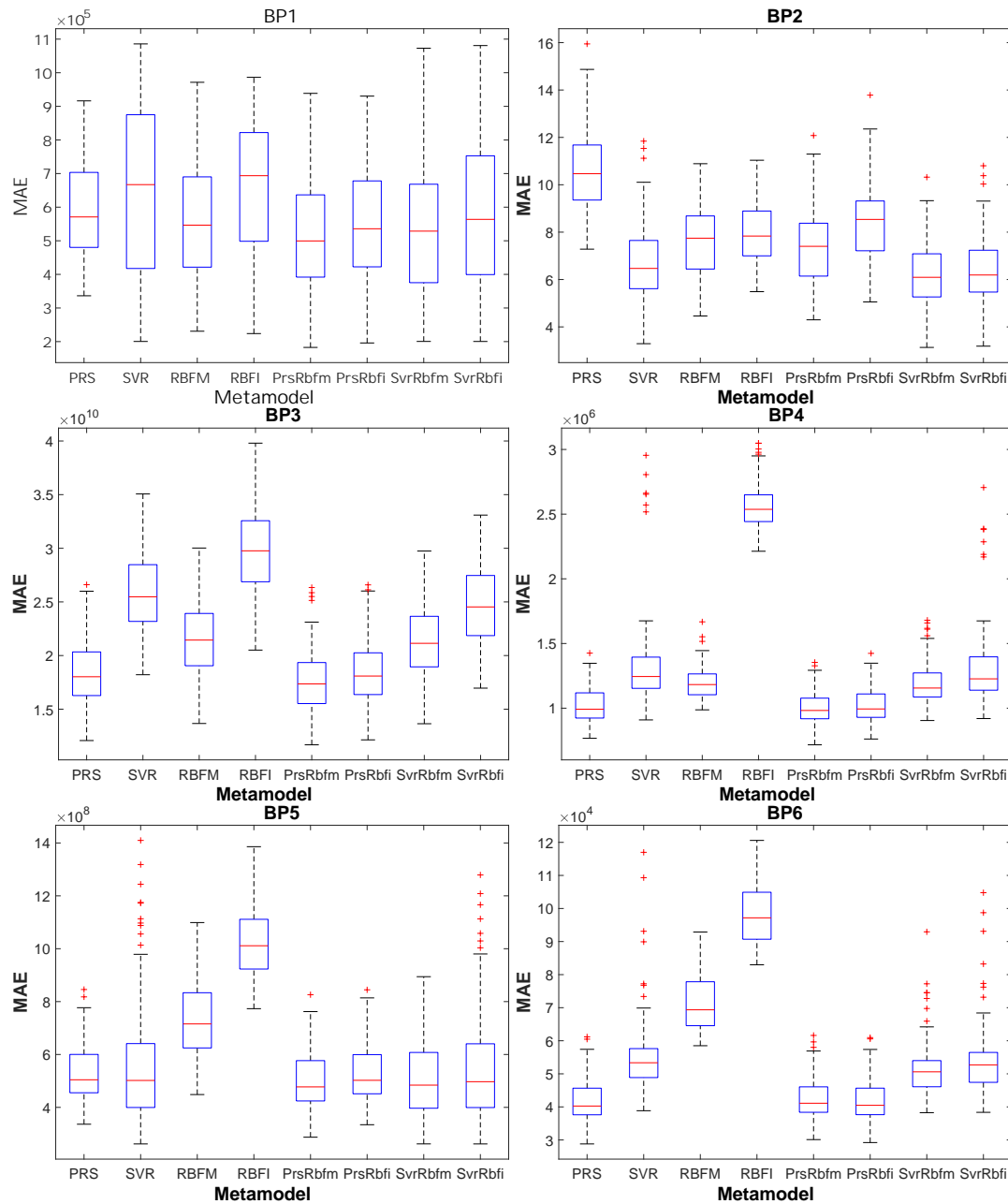


Figure 5. Boxplots of MAE of the metamodels over 100 test sets for each benchmark problem with $\frac{3(k+1)(k+2)}{2}$ training points.

4.3. Effect of Sampling Densities

The choice of different sampling densities may also influence the results of the metamodels. To investigate the effect of the sampling densities, this paper selects another two schemes with different sampling densities, which are $n = \frac{5(k+1)(k+2)}{4}$ and $n = \frac{7(k+1)(k+2)}{4}$.

Table 7 shows the normalized means of RMSE of the metamodels for each benchmark problem with $\frac{7(k+1)(k+2)}{4}$ training points. It can be seen that: (1) compared with the most accurate individual metamodels, the means of RMSE of the most accurate ensemble metamodels are reduced, ranging from 0.9% to 8.1%; (2) compared with the least accurate individual metamodels, the means of RMSE of the least accurate ensemble metamodels are reduced, ranging from 23.4% to 53.8%; (3) except for BP3, more than two ensemble metamodels perform better than the most accurate individual metamodels;

(4) all the ensemble metamodels perform better than the four individual metamodels; (5) PrsRbfm is the best performer among the four metamodels, while SvrRbfi is the worst performer.

Table 7. Normalized means of RMSE of the metamodels for each benchmark problem with $\frac{7(k+1)(k+2)}{4}$ training points.

BPs	BP1	BP2	BP3	BP4	BP5	BP6
PRS	1.400	2.282	1.104	1.000	1.342	1.000
SVR	1.209	1.000	1.283	1.154	1.000	1.130
RBFM	1.000	1.186	1.000	1.012	1.496	1.371
RBFI	1.174	1.383	1.555	2.188	2.847	2.195
PrsRbfm	0.921	1.102	0.952	0.946	1.158	0.991
PrsRbfi	0.985	1.271	1.044	0.987	1.316	0.992
SvrRbfm	0.958	0.937	1.040	1.000	0.919	1.076
SvrRbfi	1.014	0.949	1.191	1.102	0.972	1.105
Best & Best	−7.9%	−6.3%	−4.8%	−5.4%	−8.1%	−0.9%
Worst & Worst	−27.6%	−44.3%	−23.4%	−49.6%	−53.8%	−49.7%

Table 8 shows the normalized means of RMSE of the metamodels for each benchmark problem with $\frac{5(k+1)(k+2)}{4}$ training points. It can be seen that: (1) compared with the most accurate individual metamodels, the means of RMSE of the most accurate ensemble metamodels are reduced for five benchmark problems, ranging from 0.9% to 16.9%; (2) compared with the least accurate individual metamodels, the means of RMSE of the least accurate ensemble metamodels are reduced, ranging from 20.9% to 51.3%; (3) all the ensemble metamodels have better performance than the four individual metamodels.

Table 8. Normalized means of RMSE of the metamodels for each benchmark problem with $\frac{5(k+1)(k+2)}{4}$ training points.

BPs	BP1	BP2	BP3	BP4	BP5	BP6
PRS	1.268	1.577	1.115	1.022	1.000	1.000
SVR	1.248	1.000	1.242	1.206	1.027	1.174
RBFM	1.000	1.020	1.000	1.000	1.134	1.379
RBFI	1.100	1.126	1.517	2.151	2.030	2.093
PrsRbfm	0.937	1.008	1.015	0.991	0.918	0.991
PrsRbfi	0.980	1.098	1.079	1.014	0.989	0.995
SvrRbfm	0.965	0.924	1.053	1.035	0.831	1.073
SvrRbfi	1.002	0.938	1.167	1.152	0.951	1.139
Best & Best	−6.3%	−7.6%	1.5%	−0.9%	−16.9%	−0.9%
Worst & Worst	−20.9%	−30.4%	−23.1%	−46.4%	−51.3%	−45.6%

In summary, the choice of different sampling densities influences the results slightly, but the conclusions obtained by the three schemes with different sampling densities remain unchanged.

4.4. Significance of Results

The results above have proven the effectiveness of the proposed method to some extent. To further demonstrate the advantages, the proposed method is compared with some other popular ensemble metamodels, which are BPS (Best PRESS surrogate), PWS (PRESS weighted average surrogate), and OWSD (Optimal weighted surrogate using the diagonal elements). The detailed descriptions of these ensemble metamodels can be found in relevant literature [35,37]. Additionally, Kriging with first order polynomial regression function (KRG1) and Kriging with second-order polynomial regression function (KRG2) are also included in the performance comparison. To be noted, the principle and modeling process of Kriging are different from that of the proposed metamodeling approach in this paper.

Figure 6 compares the performance of PrsRbfm, SvrRbfm, KRG1, KRG2, BPS, PWS, and OWSD. It can be seen that: (1) for BP1, PrsRbfm and SvrRbfm perform better than the other five metamodels; (2) for BP2, SvrRbfm and BPS are the best two performers; (3) for BP3, the accuracy of PrsRbfm and BPS are better than that of the other metamodels; (4) for BP4, PrsRbfm and KRG2 are the best two performers; (5) for BP5, SvrRbfm and BPS are more accurate than other metamodels; (6) for BP6, PrsRbfm and KRG2 perform better the other metamodels.

In summary, the proposed metamodeling approach possesses some advantages when compared with KRG1, KRG2, BPS, PWS, and OWSD.

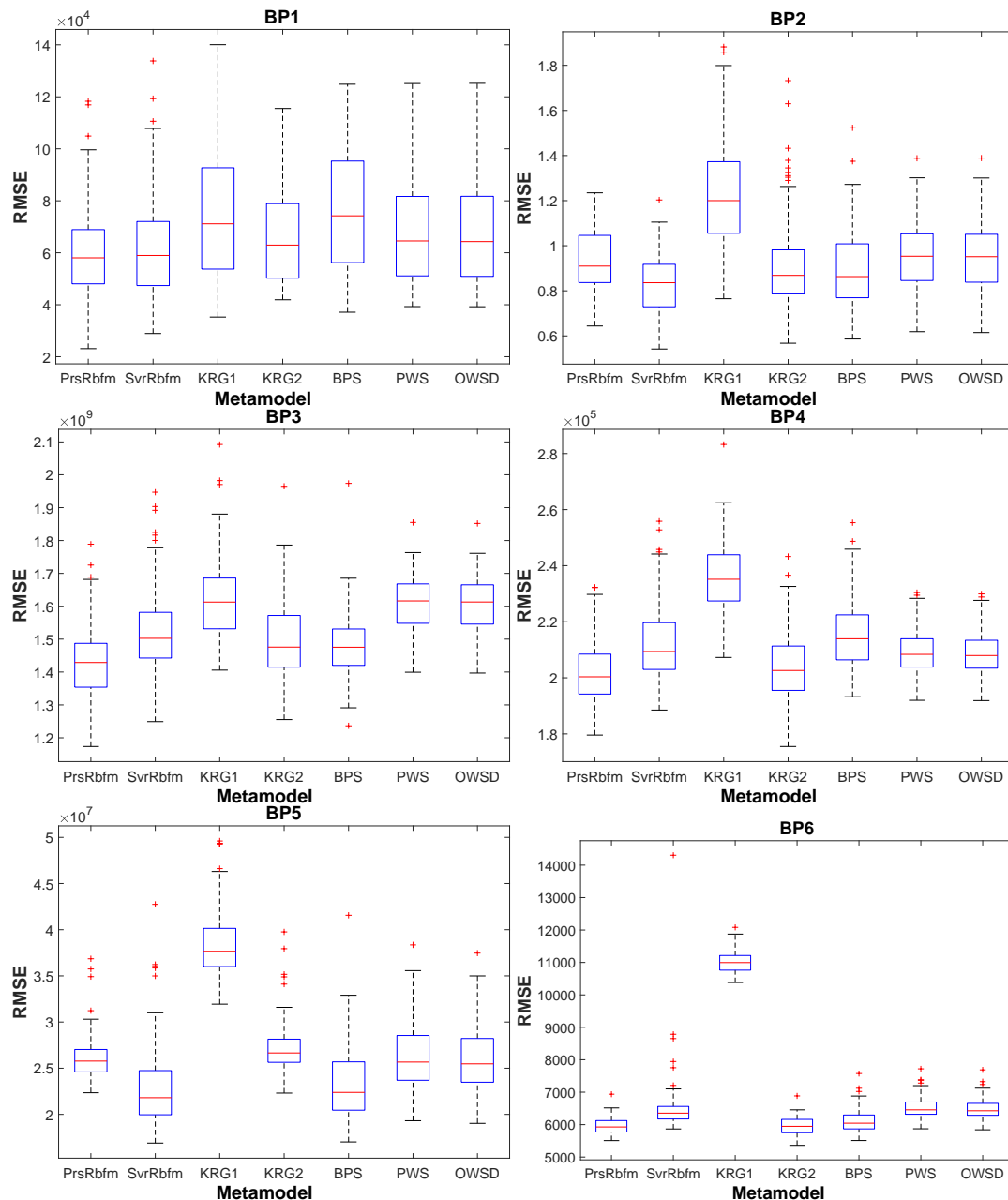


Figure 6. Boxplots of RMSE of PrsRbfm, SvrRbfm, KRG1, KRG2, BPS, PWS, and OWSD for the benchmark problems.

5. Conclusions

This paper proposed a novel metamodeling approach for building ensemble metamodels. Four types of ensemble metamodels, namely PrsRbfm, PrsRbfi, SvrRbfm, and SvrRbfi, were constructed

by choosing four individual metamodels, namely PRS, SVR, RBFM, and RBFI. The performance of these metamodels was investigated through six popular benchmark problems. The effects of the performance criteria and sampling densities on the performance of the metamodels were studied. Additionally, the significance of the results was discussed by comparing the proposed method with some other popular ensemble metamodels. According to the results, some findings of this work could be concluded as follows:

- (1) According to the experimental results, the proposed metamodeling approach could reduce the risk of choosing the worst individual metamodel, and the constructed ensemble metamodels perform better than the selected individual metamodels in terms of accuracy.
- (2) The reason for the better performance of the ensemble metamodels may be that the proposed metamodeling approach combines the advantages of the regression-type and interpolation-type metamodels. The ensemble metamodels not only capture the global trend of the actual model in the entire design space, but also achieve the local accuracy in the vicinity of sampling locations.
- (3) The choices of different performance criteria and sampling densities influence the results slightly, but the obtained conclusions remain unchanged.
- (4) The proposed metamodeling approach possesses some advantages when compared with some other popular ensemble metamodels.

Author Contributions: Conceptualization, C.Y. and J.Z.; Formal analysis, C.Y. and J.F.; Methodology, C.Y. and J.Z.; Software, Z.Q. and J.F.; Validation, C.Y. and Z.Q.; Investigation, C.Y. and Z.Q.; Resources, X.S. and D.M.; Data curation, J.Z., Z.Q. and J.F.; Writing—original draft, C.Y.; Writing—review and editing, C.Y., J.F. and J.Z.; Visualization, J.Z. and D.M.; Supervision, X.S. and D.M.; Project administration, X.S. and D.M.; Funding acquisition, X.S. and D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. An, X.; Song, B.; Mao, Z.; Ma, C. Layout Optimization Design of Two Vortex Induced Piezoelectric Energy Converters (VIPECs) Using the Combined Kriging Surrogate Model and Particle Swarm Optimization Method. *Energies* **2018**, *11*, 2069. [[CrossRef](#)]
2. Wang, D.; Hu, Q.; Tang, J.; Jia, H.; Li, Y.; Gao, S.; Fan, M. A kriging model based optimization of active distribution networks considering loss reduction and voltage profile improvement. *Energies* **2017**, *10*, 2162. [[CrossRef](#)]
3. Melo, A.; Versage, R.; Sawaya, G.; Lamberts, R. A novel surrogate model to support building energy labelling system: A new approach to assess cooling energy demand in commercial buildings. *Energy Build.* **2016**, *131*, 233–247. [[CrossRef](#)]
4. Bornatico, R.; Hüseyin, J.; Witzig, A.; Guzzella, L. Surrogate modeling for the fast optimization of energy systems. *Energy* **2013**, *57*, 653–662. [[CrossRef](#)]
5. Westermann, P.; Evins, R. Surrogate modelling for sustainable building design—A review. *Energy Build.* **2019**, *198*, 170–186. [[CrossRef](#)]
6. Ferrero Bermejo, J.; Gómez Fernández, J.F.; Pino, R.; Crespo Márquez, A.; Guillén López, A.J. Review and Comparison of Intelligent Optimization Modelling Techniques for Energy Forecasting and Condition-Based Maintenance in PV Plants. *Energies* **2019**, *12*, 4163. [[CrossRef](#)]
7. Asher, M.J.; Croke, B.F.W.; Jakeman, A.J.; Peeters, L.J.M. A review of surrogate models and their application to groundwater modeling. *Water Resour. Res.* **2015**, *51*, 5957–5973. [[CrossRef](#)]
8. Viana, F.A.C.; Simpson, T.W.; Balabanov, V.; Toropov, V. Special section on multidisciplinary design optimization: metamodeling in multidisciplinary design optimization: How far have we really come? *AIAA J.* **2014**, *52*, 670–690. [[CrossRef](#)]
9. Razavi, S.; Tolson, B.A.; Burn, D.H. Review of surrogate modeling in water resources. *Water Resour. Res.* **2012**, *48*, 54–62. [[CrossRef](#)]

10. Forrester, A.I.J.; Keane, A.J. Recent advances in surrogate-based optimization. *Prog. Aerosp. Sci.* **2009**, *45*, 50–79. [[CrossRef](#)]
11. Wang, G.G.; Shan, S. Review of metamodeling techniques in support of engineering design optimization. *J. Mech. Des.* **2007**, *129*, 370–380. [[CrossRef](#)]
12. González-Fernández, C.; Molinuevo-Salces, B.; García-González, M.C. Evaluation of anaerobic codigestion of microalgal biomass and swine manure via response surface methodology. *Appl. Energy* **2011**, *88*, 3448–3453. [[CrossRef](#)]
13. Yan, C.; Shen, X.; Guo, F. Novel two-stage method for low-order polynomial model. *Math. Probl. Eng.* **2018**, *2018*, 8156390. [[CrossRef](#)]
14. Yan, C.; Yin, Z.; Shen, X.; Guo, F.; Wu, Y. Axisymmetric hub-endwall profile optimization for a transonic fan to improve aerodynamic performance based on an integrated design optimization method. *Struct. Multidiscip. Optim.* **2019**, *60*, 1267–1282. [[CrossRef](#)]
15. Yan, C.; Shen, X.; Guo, F.; Zhao, S.; Zhang, L. A novel model modification method for support vector regression based on radial basis functions. *Struct. Multidiscip. Optim.* **2019**, *60*, 983–997. [[CrossRef](#)]
16. Lee, C.W.; Lin, B.Y. Applications of the chaotic quantum genetic algorithm with support vector regression in load forecasting. *Energies* **2017**, *10*, 1832. [[CrossRef](#)]
17. Hong, W.C.; Fan, G.F. Hybrid Empirical Mode Decomposition with Support Vector Regression Model for Short Term Load Forecasting. *Energies* **2019**, *12*, 1093. [[CrossRef](#)]
18. Fang, H.; Horstemeyer, M.F. Global response approximation with radial basis functions. *Eng. Optim.* **2006**, *38*, 407–424. [[CrossRef](#)]
19. Zhou, Q.; Cao, L.; Zhou, H.; Huang, X. Prediction of angular distortion in the fiber laser keyhole welding process based on a variable-fidelity approximation modeling approach. *J. Intell. Manuf.* **2018**, *29*, 719–736. [[CrossRef](#)]
20. Mullur, A.; Messac, A. Metamodeling using extended radial basis functions: A comparative approach. *Eng. Comput.* **2006**, *21*, 203–217. [[CrossRef](#)]
21. Kim, C.; Wang, S.; Choi, K.K. Efficient response surface modeling by using moving least-squares method and sensitivity. *AIAA J.* **2005**, *43*, 2404–2411. [[CrossRef](#)]
22. Runge, J.; Zmeureanu, R. Forecasting Energy Use in Buildings Using Artificial Neural Networks: A Review. *Energies* **2019**, *12*, 3254. [[CrossRef](#)]
23. Silitonga, A.S.; Mahlia, T.M.I.; Shamsuddin, A.H.; Ong, H.C.; Milano, J.; Kusumo, F.; Sebayang, A.H.; Dharma, S.; Ibrahim, H.; Husin, H.; et al. Optimization of Cerbera manghas Biodiesel Production Using Artificial Neural Networks Integrated with Ant Colony Optimization. *Energies* **2019**, *12*, 3811. [[CrossRef](#)]
24. Crino, S.; Brown, D.E. Global optimization with multivariate adaptive regression splines. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2007**, *37*, 333–40. [[CrossRef](#)]
25. Nam, S.; Hur, J. Probabilistic Forecasting Model of Solar Power Outputs Based on the Naïve Bayes Classifier and Kriging Models. *Energies* **2018**, *11*, 2982. [[CrossRef](#)]
26. Venturelli, G.; Benini, E.; L, L.W. A Kriging-assisted multiobjective evolutionary algorithm. *Appl. Soft Comput.* **2017**, *58*, 155–175. [[CrossRef](#)]
27. Perrone, M.P.; Cooper, L.N. When networks disagree: Ensemble methods for hybrid neural networks. In *Artificial Neural Networks for Speech and Vision*; Mammone, R.J., Ed.; Chapman and Hall: London, UK, 1993; pp. 126–142.
28. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: New York, NY, USA, 1995; pp. 364–371.
29. Zhou, Q.; Rong, Y.; Shao, X.; Jiang, P. Optimization of laser brazing onto galvanized steel based on ensemble of metamodels. *J. Intell. Manuf.* **2018**, *29*, 1417–1431. [[CrossRef](#)]
30. Zepa, L.E.; Queipo, N.V.; Pintos, S.; Salager, J.L. An optimization methodology of alkaline-surfactant-polymer flooding processes using field scale numerical simulation and multiple surrogates. *J. Pet. Sci. Eng.* **2005**, *47*, 197–208. [[CrossRef](#)]
31. Sanchez, E.; Pintos, S.; Queipo, N.V. Toward an optimal ensemble of kernel-based approximations with engineering applications. *Struct. Multidiscip. Optim.* **2008**, *36*, 247–261. [[CrossRef](#)]
32. Acar, E. Various approaches for constructing an ensemble of metamodels using local measures. *Struct. Multidiscip. Optim.* **2010**, *42*, 879–896. [[CrossRef](#)]

33. Zhang, J.; Chowdhury, S.; Messac, A. An adaptive hybrid surrogate model. *Struct. Multidiscip. Optim.* **2012**, *46*, 223–238. [[CrossRef](#)]
34. Lee, Y.; Choi, D.H. Pointwise ensemble of meta-models using v nearest points cross-validation. *Struct. Multidiscip. Optim.* **2014**, *50*, 383–394. [[CrossRef](#)]
35. Goel, T.; Haftka, R.T.; Shyy, W.; Queipo, N.V. Ensemble of surrogates. *Struct. Multidiscip. Optim.* **2007**, *33*, 199–216. [[CrossRef](#)]
36. Acar, E.; Rais-Rohani, M. Ensemble of metamodels with optimized weight factors. *Struct. Multidiscip. Optim.* **2009**, *37*, 279–294. [[CrossRef](#)]
37. Viana, F.A.C.; Haftka, R.T.; Steffen, V. Multiple surrogates: How cross-validation errors can help us to obtain the best predictor. *Struct. Multidiscip. Optim.* **2009**, *39*, 439–457. [[CrossRef](#)]
38. Toal, D.J.; Keane, A.J. Performance of an ensemble of ordinary, universal, non-stationary and limit Kriging predictors. *Struct. Multidiscip. Optim.* **2013**, *47*, 893–903. [[CrossRef](#)]
39. Acar, E. Simultaneous optimization of shape parameters and weight factors in ensemble of radial basis functions. *Struct. Multidiscip. Optim.* **2014**, *49*, 969–978. [[CrossRef](#)]
40. Polynkin, A.; Toropov, V.V. Mid-range metamodel assembly building based on linear regression for large scale optimization problems. *Struct. Multidiscip. Optim.* **2012**, *45*, 515–527. [[CrossRef](#)]
41. Ferreira, W.G.; Serpa, A.L. Ensemble of metamodels: The augmented least squares approach. *Struct. Multidiscip. Optim.* **2016**, *53*, 1019–1046. [[CrossRef](#)]
42. Zhou, X.; Jiang, T. Metamodel selection based on stepwise regression. *Struct. Multidiscip. Optim.* **2016**. [[CrossRef](#)]
43. Fletcher, R. *Practical Methods of Optimization*; Wiley: New York, NY, USA, 2013.
44. Jin, R.; Chen, W.; Simpson, T.W. Comparative Studies Of Metamodeling Techniques Under Multiple Modeling Criteria. *Struct. Multidiscip. Optim.* **2001**, *23*, 1–13. [[CrossRef](#)]
45. Forrester, A.; Sobester, A.; Keane, A. *Engineering Design via Surrogate Modelling: A Practical Guide*; Wiley: New York, NY, USA, 2008. [[CrossRef](#)]
46. Chen, R.; Liang, C.Y.; Hong, W.C.; Gu, D.X. Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm. *Appl. Soft Comput.* **2015**, *26*, 435–443. [[CrossRef](#)]
47. Yan, C.; Shen, X.; Guo, F. An improved support vector regression using least squares method. *Struct. Multidiscip. Optim.* **2017**, *57*, 2431–2445. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).