Special Issue Article

Advances in Mechanical Engineering

Advances in Mechanical Engineering
2016, Vol. 8(5) 1–14

DOI: 10.1177/1687814016647317
aime.sagepub.com
SAGE

Artificial intelligence metamodel comparison and application to wind turbine airfoil uncertainty analysis

Yaping Ju[1], Chuhua Zhang[1] and Lin Ma[2]

Abstract

The Monte Carlo simulation method for turbomachinery uncertainty analysis often requires performing a huge number of simulations, the computational cost of which can be greatly alleviated with the help of metamodeling techniques. An intensive comparative study was performed on the approximation performance of three prospective artificial intelligence metamodels, that is, artificial neural network, radial basis function, and support vector regression. The genetic algorithm was used to optimize the predetermined parameters of each metamodel for the sake of a fair comparison. Through testing on 10 nonlinear functions with different problem scales and sample sizes, the genetic algorithm–support vector regression metamodel was found more accurate and robust than the other two counterparts. Accordingly, the genetic algorithm–support vector regression metamodel was selected and combined with the Monte Carlo simulation method for the uncertainty analysis of a wind turbine airfoil under two types of surface roughness uncertainties. The results show that the genetic algorithm–support vector regression metamodel can capture well the uncertainty propagation from the surface roughness to the airfoil aerodynamic performance. This work is useful to the application of metamodeling techniques in the robust design optimization of turbomachinery.

Keywords

Support vector regression, artificial neural network, radial basis function, uncertainty analysis, wind turbine airfoil

Date received: 22 December 2015; accepted: 10 April 2016

Academic Editor: Amir Alavi

Introduction

Almost all of the real-world engineering systems suffer from a wide variety of uncertainties, such as manufacturing tolerance, material nonuniformity, mathematical and/or numerical approximation, environment variation, and in-service degradation. Due to the lack of proper treatment of these uncertainties, design engineers have seen too many unpleasant cases of performance deficiencies in their own products. It does pose a significant challenge for engineers to include rigorous uncertainty analysis as much as possible within the cycle of design to make cost-effective products.[1]

The primary task of the uncertainty analysis is to evaluate the impact of various uncertainties on the performance of the product, in which both deterministic simulations and statistical analysis are required. According to the manners by which the deterministic solvers are implemented with the statistical methods, the approaches of uncertainty analysis are classified into two groups: the intrusive methods and the nonintrusive methods.[2] The former, such as the polynomial

[1]Xi'an Jiaotong University, Xi'an, China
[2]The University of Sheffield, Sheffield, UK

Corresponding author:
Chuhua Zhang, Xi'an Jiaotong University, No. 28, Xianning West Road, Xi'an 710049, China.
Email: chzhang@mail.xjtu.edu.cn

chaos (PC) techniques, can directly yield the uncertain outputs through a one-time solution of certain reformulated equations; however, they are severely limited to low-dimensional, low-order, and short-time systems due to the mathematical and/or numerical difficulties arising from the reformulated equations. The latter, such as Monte Carlo simulation (MCS) method or its variants, are often easy to implement in combination with the available deterministic solvers, yet requiring a huge number of simulations for a statistical convergence.

Modern engineering analysis and design are frequently based on high-fidelity simulation tools such as the computational fluid dynamics (CFD) for fluid flow and the finite element analysis (FEA) for structural strength and vibration. These high-fidelity simulation tools are often difficult, if not impossible, to be directly used for the uncertainty analysis and design due to the heavy computational load in spite of the continuing advances of computing power. The MCS-based uncertainty analysis would be prohibitively expensive because of the requirements of even more CFD and/or FEA simulations of sampling points. An efficient measure to tackle this issue is to build variable-fidelity metamodels, which are regarded as a "model of the model,"[3]that is, a fast, inexpensive, effective but approximate model of the high-fidelity model. So far, various metamodels have been developed, such as polynomial regression (PR),[4] Kriging (KG),[5] multivariate adaptive regression splines (MARS),[6] artificial neural network (ANN),[7,8] radial basis function (RBF),[9,10] and support vector regression (SVR).[11,12] Due to the greatly reduced computational cost, the metamodeling techniques have been widely used for the analysis and design of engineering devices.[13,14] Typically, the applications of metamodeling techniques in turbomachinery design and analysis can be found in Chahine et al.,[15] Mackman et al.,[16] Sugimura et al.,[17] Ma et al.,[18] and Ju and Zhang.[19]

One of the most important issues in metamodeling engineering problems is to evaluate the performances of metamodels and then select the most appropriate one for their design and analysis. Jin et al.[20] performed a systematic comparison among PR, KG, MARS, and RBF. They aimed at developing standard procedures and multiple criteria for metamodel evaluation. In the following studies, some other metamodels were also compared in some particular applications.[16,21–25] All these comparative studies, to some extent, revealed the great potential and promise of ANN, RBF, and SVR for approximation. These three models are acknowledged as the artificial intelligence (AI) metamodels, which allow the behavior of self-learning and can smartly extract the mechanisms that underlie the complex systems. However, it remains unknown yet which of them has the best approximation capability in general cases and whether they could be used for turbomachinery uncertainty analysis. Therefore, in this study, an intensive comparison among these three AI metamodels was particularly conducted, where the performance evaluation strategies were carefully designed and tested through a total of 10 nonlinear functions with different problem scales and sample sizes.

Additional efforts were then devoted to applying the AI metamodeling technique to the uncertainty analysis of a wind turbine airfoil. As exposed to harsh environmental conditions, the in-service wind turbine blades unavoidably suffer from insect and/or dust contaminations, blade erosion and/or corrosion, and massive sand build-up, which cause increased blade surface roughness to varying degrees. So far, extensive numerical and experimental studies have concluded that the increased surface roughness, in particular around the blade leading edge (LE), can induce boundary layer disturbances, earlier turbulence transition and eventually wind turbine performance degradation.[26–31] However, most of these studies only obtained deterministic results and ignored the wind turbine performance variations due to the stochastic nature of the surface roughness in real conditions. In view of this, the motivation of the present uncertainty analysis was to investigate the impact of surface roughness uncertainties on the aerodynamic performance of a wind turbine airfoil, although the used methodology can be extended without modifications to other types of uncertainties.

This article is organized as follows. Section "AI metamodels" mainly presents the basic formulations of the three AI metamodels: ANN, RBF, and SVR. In section "Numerical assessment of metamodeling performance," a total of 10 analytical functions are tested to compare these three AI models in terms of accuracy and robustness, where the impacts of problem scale and sample size are carefully examined. The focus of section "Uncertainty analysis of wind turbine airfoil" is on the application of the metamodeling techniques to the uncertainty analysis of a wind turbine airfoil under surface roughness uncertainties. Some conclusions are drawn in section "Conclusion."

AI metamodels

ANN

The mathematical form of an ANN model with one hidden layer and one output neuron can be written as[7]

$$\hat{y} = \hat{f}(\boldsymbol{X}) = f_2\left(\sum_{h=1}^{H} w_h f_1\left(\sum_{i=1}^{N} v_{ih} x_i + b_h\right) + b_2\right) \quad (1)$$

where $\boldsymbol{X}$ is a N-dimensional input vector with x_i ($i = 1, 2,\ldots, N$) as its elements; $\hat{y}$ is the model output for the approximation result; $f_1(\cdot)$ and $f_2(\cdot)$ represent the input-hidden and hidden-output transfer functions,

respectively; v_{ih} is the weight connecting the ith input neuron and the hth hidden neuron; b_h is the bias in the hth hidden layer; w_h is the weight connecting the hth hidden neuron with the output neuron; b_2 is the bias in the output layer; and H is the total number of the hidden neurons. In this study, both the weights and bias are calculated via the back propagation (BP) algorithm,[7] and the sigmoid function is selected for the input-hidden transfer function while the linear function for the hidden-output.

As demonstrated in our previous work,[19] the initial weights used in the BP algorithm are often difficult to determine but significantly affect the ANN performance. These parameters are thus taken as the predetermined ones and optimized by the genetic algorithm (GA), as will be discussed in section "Use of GA."

RBF

The mathematical form of a RBF can be written as[9,10]

$$\hat{y} = \hat{f}(\boldsymbol{X}) = \sum_{i=1}^{H} w_i \phi(||\boldsymbol{X} - \boldsymbol{X}_i||) + b \tag{2}$$

where $||\boldsymbol{X} - \boldsymbol{X}_i||$ represents the Euclidean distance between the input vector $\boldsymbol{X}$ and the central point vector $\boldsymbol{X}_i$ of the ith hidden neuron, $\phi(\cdot)$ is a RBF, H is the hidden neuron number which may be equal or smaller than the sample size, w_i is the weight connecting the ith hidden neuron and the output neuron, and b is the bias in the output layer. Therefore, the RBF actually expresses a weighted linear combination of H RBFs. In this study, the most commonly used Gaussian function is chosen as the RBF, which reads[9]

$$\phi(||\boldsymbol{X} - \boldsymbol{X}_i||) = \exp\left(-\frac{||\boldsymbol{X} - \boldsymbol{X}_i||^2}{2{\sigma_i}^2}\right) \tag{3}$$

where σ_i is the spread factor and directly affects the smoothness of the approximation result.

During the learning procedure of a RBF model, the predetermined parameters are the spread factors in equation (3) and the initial weights in equation (2).

SVR

With the concept of ε-insensitive loss function, the mathematical form of a SVR is[11]

$$\hat{y} = \hat{f}(\boldsymbol{X}) = \boldsymbol{W}\boldsymbol{\Phi}(\boldsymbol{X}) + b \tag{4}$$

where $\boldsymbol{\Phi}(\cdot)$ represents the nonlinear transformation imposed on the input vectors. The coefficients $\boldsymbol{W}$ and b can be obtained by solving the following convex quadratic programming optimization problem[11]

$$\min \frac{1}{2}||\boldsymbol{W}||^2 + C\sum_{p=1}^{P}\left(\xi_p + \xi_p^*\right) \tag{5}$$

$$\begin{aligned} &\text{s.t.}((\boldsymbol{W}\cdot\boldsymbol{X}_p) + b) - y_p \le \varepsilon + \xi_p \\ &y_p - ((\boldsymbol{W}\cdot\boldsymbol{X}_p) + b) \le \varepsilon + \xi_p^* \\ &\xi_p, \xi_p^* \ge 0, \quad p = 1, \ldots, P \end{aligned} \tag{6}$$

where $\boldsymbol{X}_p$ and y_p are the input and output of the pth sample point, P is the sample size, ξ_p and ξ_p^* are the slack variables used to satisfy the constraints, C is the penalty coefficient for the tradeoff between training error minimization and model complexity, and ε is the tube size controlling the penalty range. After solving the above minimization problem, the final prediction function can be expressed as[11]

$$\hat{f}\left(\boldsymbol{X}, \alpha_p, \alpha_p^*\right) = \sum_{p=1}^{P}\left(\alpha_p - \alpha_p^*\right)k(\boldsymbol{X}, \boldsymbol{X}_p) + b \tag{7}$$

where α_p, α_p^*, and b are the obtained coefficients and $k(\boldsymbol{X}, \boldsymbol{X}_p)$ is the kernel function measuring the distance between the new input $\boldsymbol{X}$ and the pth sample point $\boldsymbol{X}_p$. Table 1 lists some common kernel functions for the SVR. The predetermined parameters of SVR are the penalty coefficient C in equation (5) and the kernel function variable d or σ shown in Table 1.

Use of GA

The approximation performances of AI metamodels are closely associated with the predetermined parameters, which in most studies were specified or adjusted randomly. This may be less convincing or even misleading in that method A with a good set of parameters is likely to achieve better approximations than method B with a poor set of parameters even if method B is generally more accurate than method A. With this in mind, Li et al.[24] employed the hold-out validation plus grid search scheme to optimize the predetermined parameters of metamodels. However, this scheme is time-consuming and only limited to discrete parameter values. By contrast, modern evolutionary algorithms, for example, GA, can be more efficient and versatile for finding out the global optimums.

Table 1. Common kernel functions for the SVR model.[11]

Kernel function	Mathematical expression
Linear	$k(\boldsymbol{X}, \boldsymbol{X}') = (\boldsymbol{X}\cdot\boldsymbol{X}')$
Polynomial	$k(\boldsymbol{X}, \boldsymbol{X}') = ((\boldsymbol{X}\cdot\boldsymbol{X}') + 1)^d$
Gaussian	$k(\boldsymbol{X}, \boldsymbol{X}') = \exp\left(-\frac{||\boldsymbol{X}-\boldsymbol{X}'||^2}{2\sigma^2}\right)$

Table 2. Training and testing sample sizes for different problems.

Problems	Number of variables	Training sample size			Testing sample size
		Scarce set	Small set	Large set	
Small-scale	2	–	9	20	1000
	3	–	11	25	1000
Large-scale	5	11	30	55	1500
	8	17	40	60	2000
	10	19	50	100	2000

In this work, GA was adopted to optimize the predetermined model parameters through minimizing errors between the metamodel outputs and the actual outputs, the motivations behind which are twofold: first, it eliminates the uncontrollable randomness in the modeling process and thus enhances the metamodel training efficiency and robustness; second, GA helps to maximize the approximation accuracy of each metamodel, which thus strengthens the fairness of comparison in the following section.

Here, it has to be remarked that the GA parameters were tuned for different metamodels and different test cases. Detailed implementations of GA in metamodel optimization can be found in Ju and Zhang.[19]

Numerical assessment of metamodeling performance

Descriptions of test problems

So far, the number of independent design variables is generally limited below 10 in the simulation-based and metamodel-assisted design optimization of turbomachinery.[13,16–18] A total of 10 analytic functions with different degrees of nonlinearity and problem scales[22,32,33] were first chosen in this section for a comprehensive evaluation of the three AI metamodeling techniques. Among them, the first six functions have not more than three independent variables and were classified as small-scale problems, while the other functions with 5, 8 or 10 variables were large-scale problems in this study. The mathematical expressions of these functions are listed in Appendix 1.

The training sample sets were generated by means of the uniform design of experiment (DOE), which is capable of providing fewer training samples with better uniformity and representativeness.[34] In a uniform DOE table $U_n(t^q)$, n denotes the number of experiments or the sample size while q and t represent the number of factors (i.e. variables) and their levels, respectively. Accordingly, different training sampling sets were generated for different functions, that is, small and large sample sets for small-scale problems, while scarce, small, and large sample sets for large-scale problems (see Table 2). In summary, totally $6 \times 2 + 4 \times 3 = 24$ training sample sets were generated for approximation. Here, the terms "scarce,""small," and "large" did not denote fixed values but were defined with respect to the problem scale, that is, the factor q in the uniform DOE table. Besides the training sampling sets, additional 10 testing sample sets were randomly generated within the variation ranges to verify the approximation performance.

Performance measures

To evaluate the approximation performance of the metamodels, a group of measures have been developed, such as accuracy, robustness, efficiency, transparency, and simplicity.[21] Among them, accuracy and robustness were the primary focus of this study because they play essential roles in determining whether a metamodel is acceptable or not for approximation.

Accuracy. Accuracy is defined as a metamodel's capacity of predicting the system response over the pre-specified design space. To take a comprehensive measure of a metamodel's accuracy performance, three error metrics described in equations (8)–(10) were investigated, that is, root mean square error (RMSE), averaged absolute error (AAE), and maximum absolute error (MAE)[21]

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{8}$$

$$\mathrm{AAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{9}$$

$$\mathrm{MAE} = \max|\hat{y}_i - y_i| \tag{10}$$

The above errors should be calculated from testing samples rather than training samples. Among them, N denotes the number of testing samples, y_i is the actual (analytic) output of the ith testing sample, while $\hat{y}_i$ is the corresponding metamodel output. Both RMSE and AAE provide the overall indications of the difference between metamodel outputs and actual data, while

MAE reflects the presence of local poor approximation. The lower values of these errors mean the higher accuracy of a metamodel.

Considering the different output ranges for the 10 test functions (see Appendix 1), it is necessary to normalize the obtained error metrics to facilitate the comparison among the three metamodels across various approximation problems. For each metamodel's error metrics for a test function, two types of normalizations were defined in this study:

- *Normalization 1*

$$\mathrm{RMSE}^* = \frac{\mathrm{RMSE} \times 100}{\sum \mathrm{RMSE} \text{under } \textbf{\textit{each}} \text{ sample size of the test function}} \quad (11)$$

- *Normalization 2*

$$\mathrm{RMSE}^* = \frac{\mathrm{RMSE} \times 100}{\sum \mathrm{RMSE} \text{ under } \textbf{\textit{all}} \text{ sample sizes of the test function}} \quad (12)$$

where RMSE^* denotes the normalized value of RMSE and $\sum \mathrm{RMSE}$ is the sum of RMSE of the three metamodels. Two other normalized errors, that is, AAE^* and MAE^*, were defined similarly. Through normalization, the error metrics are closely associated with the concept of percentage, share the same order of magnitude for different test functions, and can be added up to calculate the averaged accuracies of a specific metamodel

$$\mu(\mathrm{RMSE}^*) = \mathrm{mean}(\mathrm{RMSE}^*) \quad (13)$$

Two other mean values, $\mu(\mathrm{AAE}^*)$ and $\mu(\mathrm{MAE}^*)$, were defined similarly. By such definition, lower values of these measures represent higher averaged accuracy of a metamodel for different approximation problems.

Robustness. Robustness measures a metamodel's capacity of achieving similar accuracies for different approximation problems. Mathematically, here it was defined as the standard deviation of error metrics across all the problems of interest

$$\sigma(\mathrm{RMSE}^*) = \mathrm{Standard\ deviation}(\mathrm{RMSE}^*) \quad (14)$$

Two other standard deviations, $\sigma(\mathrm{AAE}^*)$ and $\sigma(\mathrm{MAE}^*)$, were defined similarly. Accordingly, a more robust and problem-independent metamodel should have lower values of $\sigma(\mathrm{RMSE}^*)$, $\sigma(\mathrm{AAE}^*)$, and $\sigma(\mathrm{MAE}^*)$.

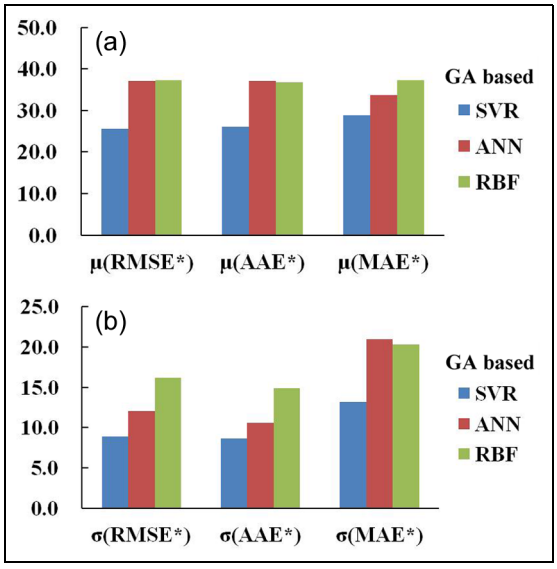

Figure 1. Overall performance comparison among the three metamodels: (a) mean values and (b) standard deviations.

Comparison results

Overall performance. The overall performances of the GA-based AI (GA-AI) metamodeling techniques were calculated as follows: for each metamodel, Normalization 1 (equation (11)) was first implemented to calculate its RMSE^*, AAE^*, and MAE^* for each test function under a certain sample size; then, the mean values and the standard deviations of RMSE^*, AAE^*, and MAE^* were calculated through averaging across all the 24 approximation problems.

The corresponding results are illustrated in Figure 1. Compared with the other two metamodels, the GA-based SVR (GA-SVR) achieved both lower mean values and lower standard deviations of the RMSE^*, AAE^*, and MAE^*, indicating that GA-SVR has the best overall performance and could be the best choice for metamodeling, especially when there is no knowledge about the problem scales and sample sizes. Except for the $\sigma(\mathrm{MAE}^*)$, the GA-based ANN (GA-ANN) generally performed the next best, followed by the GA-based RBF (GA-RBF).

Impact of problem scale. To examine each GA-AI metamodel's performances in approximating problems with different scales, Normalization 1 (equation (11)) was adopted, while a different averaging strategy from the above was used: for each metamodel, the mean values and standard deviations of errors were calculated first for small-scale functions through averaging across a total of $6 \times 2 = 12$ approximation problems (i.e. the small and large sample sets for each of the first six functions); and then, the average was performed for large-

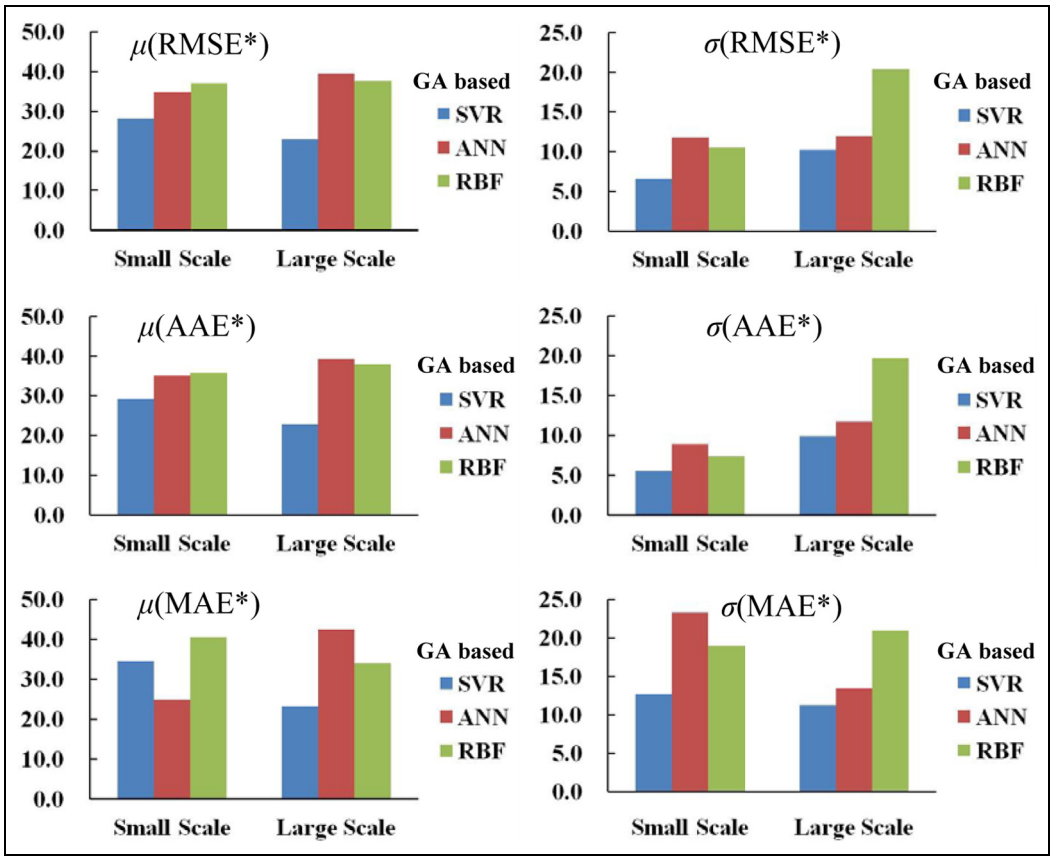

Figure 2. Performance comparison among the three metamodels for problems with different scales.

scale functions across the $4 \times 3 = 12$ approximation problems (i.e. the scarce, small and large sample sets for each of the last four functions). The comparison results are given in Figure 2.

Except for the $\mu(\text{MAE}^*)$ of small-scale problems, GA-SVR was generally better than the other two metamodels. As for GA-ANN and GA-RBF, however, it is difficult to determine which one is better: GA-ANN was more accurate but less robust than GA-RBF for small-scale problems, while for large-scale problems, GA-ANN showed less accurate but much more robust performances.

Impact of training sample size. To further investigate the impact of training sample size on the performance of metamodels, particular attention was paid to the approximation results under different training sample sets. First, the values of RMSE^*, AAE^*, and MAE^* were obtained using Normalization 2 (equation (12)), by which the best combination of the metamodel and sample size could be obtained for each test function. Second, for each group of the small- and large-scale test functions, more detailed average was performed across approximation problems with the same type of sample set.

Figure 3 summarizes the performances of the three GA-AI metamodeling techniques with different sample sets for small-scale problems. GA-SVR generally performed the best in spite of a slight inferiority in $\sigma(\text{AAE}^*)$ under the large sample size. Moreover, all the three metamodels were found more accurate and robust under the large sample size than under the small sample size, which could be attributed to the relatively rich information involved in the large sample sets. The impact of sample size on the accuracy and robustness was the smallest for GA-RBF and the largest for GA-ANN.

Compared with the small-scale problems, the large-scale problems were less influenced by sample sizes (see Figure 4). This is interesting and encouraging due to the absence of the great expense for getting the large sample information. The averaged accuracies of GA-SVR were almost equivalent for the three kinds of sample sets, while significantly better than those of the other two metamodels. GA-RBF achieved the second best accuracy performance, followed by GA-ANN. In terms of the robustness performance, GA-SVR still performed excellently for all the sample sets. GA-ANN was the most robust metamodel for the scarce and small sample sets but its performance deteriorated

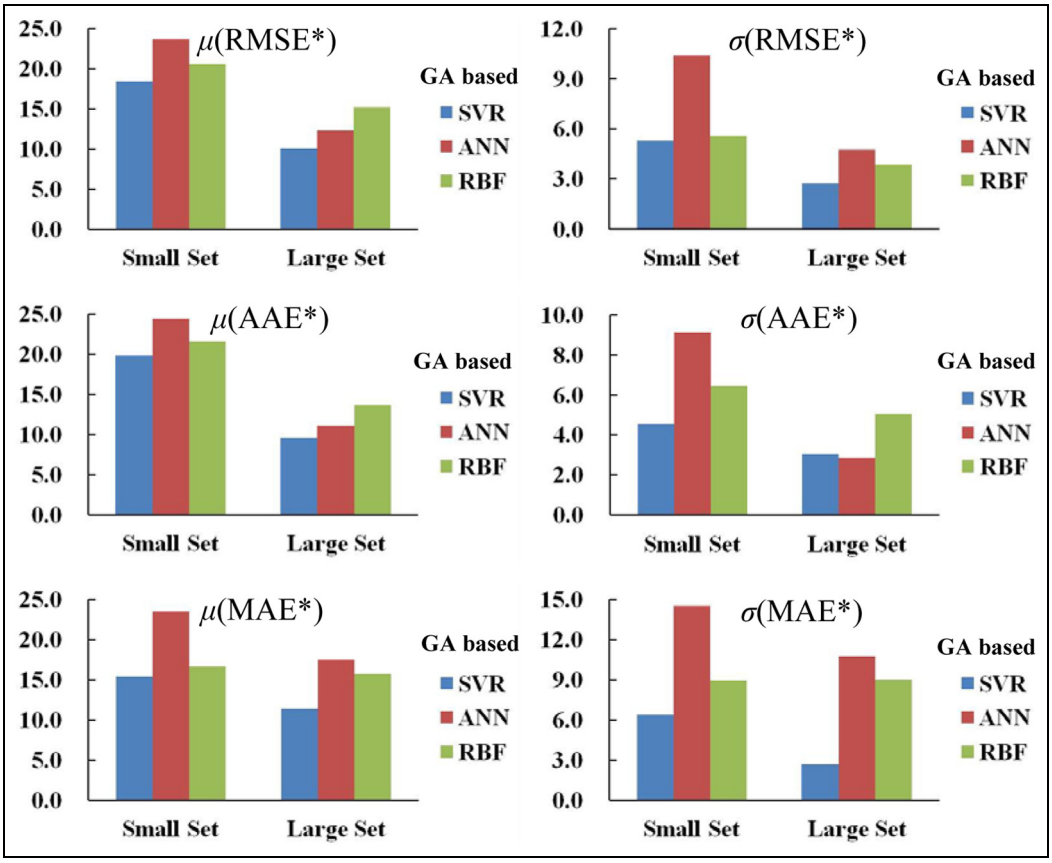

Figure 3. Performance comparison among the three metamodels for small-scale problems under different sample sizes.

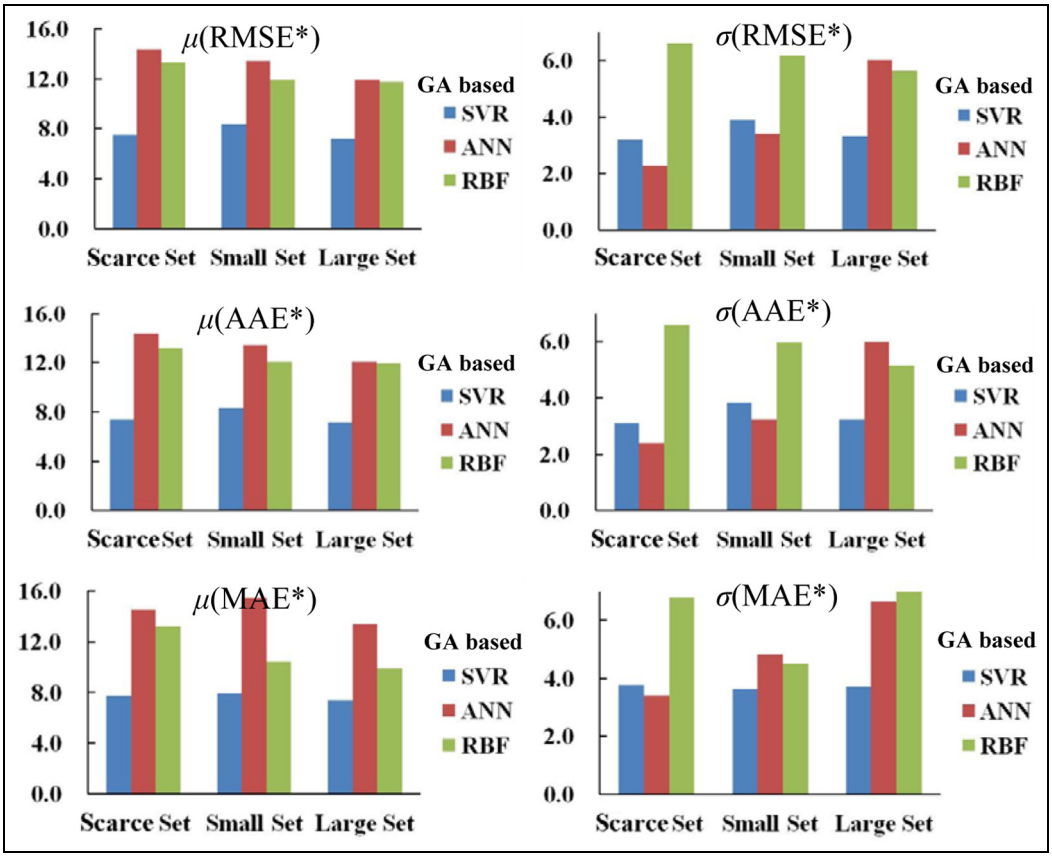

Figure 4. Performance comparison among the three metamodels for large-scale problems under different sample sizes.

significantly when the sample size became large. GA-RBF achieved the worst robustness performance in the group. Therefore, the combination of the GA-SVR metamodel and a scarce sample set could be acceptable for approximating large-scale problems. For most practical problems, the most time-consuming part during the metamodeling process is spent on the evaluation of training samples rather than the model construction itself. From this point of view, GA-SVR metamodel could be the most efficient selection for an accurate approximation of the large-scale problem.

In summary, SVR showed generally the best approximation performance in terms of accuracy and robustness among the three AI metamodels for the tested mathematical cases. The strong generalizability of SVR can be attributed to the structural risk minimization (SRM) inductive principle which strikes a good balance between the empirical risk and problem complexities.[12]

Uncertainty analysis of wind turbine airfoil

In this section, the above-mentioned GA-SVR metamodeling technique has been applied to the uncertainty analysis of a wind turbine airfoil under surface roughness uncertainties. The airfoil to be investigated was FX 63-137.[35] Prior to the uncertainty analysis, the deterministic CFD solver was outlined in the next subsection.

Deterministic CFD simulation

CFD model. The deterministic aerodynamic performances of both smooth and rough airfoils were obtained by numerical solution of two-dimensional steady incompressible Reynolds-averaged Navier–Stokes equations using a finite-volume method solver, Fluent 6.3. The pressure-based SIMPLEC algorithm was employed to treat the flow velocity–pressure coupling. The convection terms of the governing equations were discretized by the second-order upwind scheme and the diffusion term by the second-order central difference scheme. The shear-stress transport (SST) k-ω model was used to close the turbulence terms, in which the calculation of turbulence vorticity ω near solid wall carefully considers the surface roughness height k_s. To consider the effect of surface roughness, this turbulence model requires that the computational grid near the rough surface should be fine enough to include the roughness height k_s, and the y^+ of the first nodes near the rough surface should be kept below unit.

The inlet boundary of the computational domain was placed at the location about 12.5 chord-length upstream of the airfoil and the outlet was at about 20 chord-length downstream of the airfoil. The boundary condition at the inlet was the free stream flow velocity, while a static pressure was specified at the outlet. The nonslip wall boundary condition was applied on the airfoil surface. As shown in Figure 5, the generated computational grids comprised the C-type grids around the airfoil and the H-type grid in the far field region. The area-averaged y^+ of the first nodes close to the airfoil surface was about 0.89. The number of nodes assigned inside the boundary layer was 15. Besides, the grid independency was examined via successively increasing the grid number along the airfoil surface until the evaluated aerodynamic quantity was essentially no longer changed. Finally, the total number of grid nodes was 153 on the airfoil surface and 42571 all over the computational region.

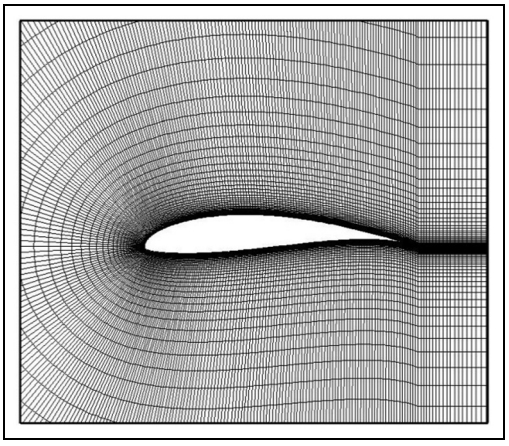

Figure 5. Local grids around the wind turbine airfoil.

Note that the capacity of the above CFD model in predicting the airfoil performance in both clean and rough conditions has been validated by Ferrer and Munduate[28] in their work. In this study, the CFD model was directly employed to evaluate the effect of surface roughness on the aerodynamic performance of the FX 63-137 airfoil.

Performance comparison. As shown in Figure 6 (top), the rough airfoil to be investigated was assumed to have deterministic roughness zones uniformly distributed around LE, and the roughness height (k_s) was fixed to be 0.3 mm, covering 20% (x/c) of the chord from the LE. The airfoil chord was 305 mm in this study. In Figure 6 (down), the calculated aerodynamic performances of the rough airfoil were characterized by a reduced lift and an increased drag relative to the smooth one. The performance deteriorations became more apparent as the incidence angle increased. The maximum lift coefficient was reduced by 4.8%, while the related drag coefficient had a relatively small increase of about 1.9%, in comparison with the clean airfoil. In addition, the stall incidence angle of the rough airfoil shifted from 12° to 11°, indicating the occurrence of an earlier stall.

Figure 7 shows the calculated pressure distributions of the smooth and rough airfoils at near-stall points.

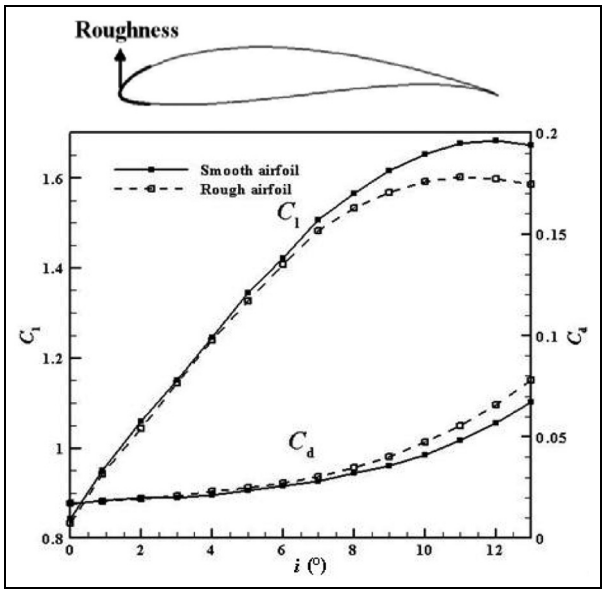

Figure 6. Comparison of the lift-drag performance between the smooth and rough airfoils.

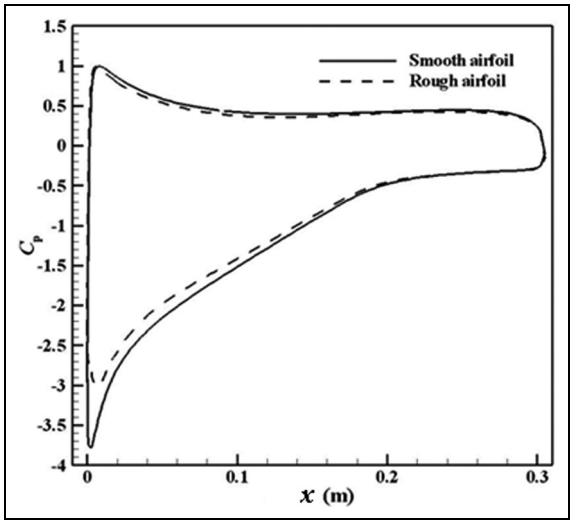

Figure 7. Comparison of the pressure coefficients C_p between the smooth and rough airfoils.

As can be seen, the most remarkable difference between smooth and rough airfoils was the flow behavior near the LE on the suction surface (SS): the rough airfoil showed a much weaker velocity acceleration from the LE, corresponding to a thicker boundary layer on the airfoil surface. This gives rise to the deteriorated airfoil performance with reduced lift, increased drag, as well as earlier stall.

Stochastic roughness model

To quantify the stochastic nature of the surface roughness distributed on the airfoil surface, two stochastic

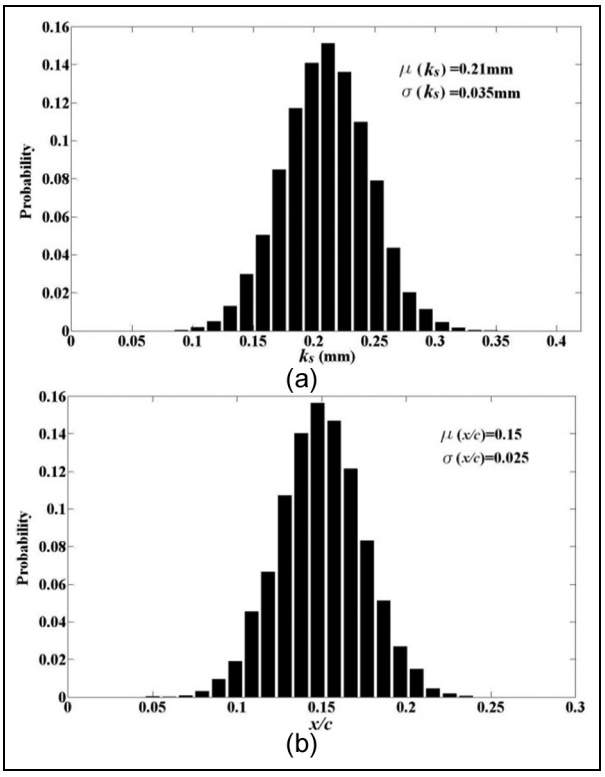

Figure 8. Gaussian probability distributions of the surface roughness parameters: (a) roughness height and (b) dimensionless covering length.

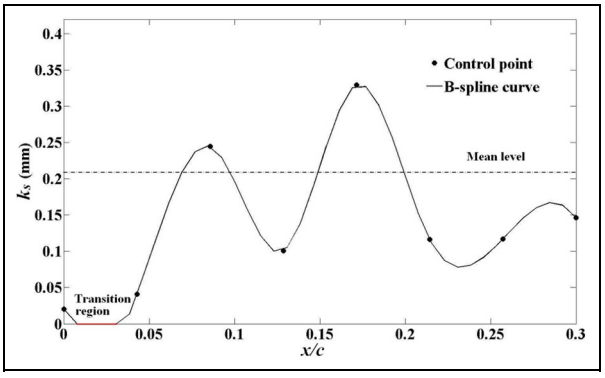

Figure 9. Roughness distribution by B-spline curve.

models were built based on the Gaussian probability distributions and the B-spline function, respectively. In the first model, the surface roughness distributions on the airfoil were assumed to be spatially uniform, while the uniformly distributed roughness height k_s and the dimensionless covering length x/c were considered to be randomly varied, both following the Gaussian probability distributions, as shown in Figure 8. Compared with the first model, the second one is more irregular and thus closer to the real conditions. As shown in Figure 9, this model used the random-shaped B-spline

Table 3. Different parameter combinations and numerical results of the uncertainty analysis in Case 1.

Parameter combinations	SS		PS		MCS	CFD
	k_s (mm)	x/c	k_s (mm)	x/c	$\mu(\Delta C_{lmax})$	ΔC_{lmax}
1	√	√	√	√	−2.25%	−2.26%
2	√	√	0	–	−1.62%	−1.60%
3	0	–	√	√	−0.58%	−0.58%
4	√	0.15	0	–	−1.60%	−1.60%
5	0	–	√	0.15	−0.59%	−0.58%
6	0.21	√	0	–	−1.61%	−1.60%
7	0	–	0.21	√	−0.58%	−0.58%

MCS: Monte Carlo simulation; CFD: computational fluid dynamics; SS: suction surface; PS: pressure surface.

curve to characterize the nonuniformity of the surface roughness distribution around the airfoil LE. The interpolating points of the B-spline curve were supposed to be randomly varied around the mean value, and any negative values on the curve were reassigned to be zero to simulate the abrupt transitions between the rough and smooth regions in real conditions. The roughness parameters of both models had the same variation ranges: the roughness height k_s was from 0 to 0.42 mm and the dimensionless covering length x/c varied from 0 to 0.3.

Uncertainty analysis and results

In order to alleviate the huge computational cost in the uncertainty analysis by MCS, the GA-SVR metamodels presented in the previous sections were constructed to fast predict the C_{lmax} of the airfoil with different surface roughness distributions. The two roughness uncertainty quantifications mentioned above were considered, respectively, as follows.

Case 1: Gaussian probability distribution. The first uncertainty analysis concerned the roughness uncertainties shown in Figure 8. A total of four uncertain inputs were considered, that is, k_s and x/c for each of the SS and pressure surface (PS) of the wind turbine airfoil. Therefore, this case was classified as a small-scale case.

A GA-SVR model was constructed to link the four roughness uncertain parameters and the C_{lmax} of the airfoil. According to the uniform DOE table $U_{30}(30^4)$, a family of 30 roughness profiles were obtained for the airfoil as the training samples, which is in line with the suggestion obtained from Figure 3 that the GA-SVR combined with a large sample set could achieve the best approximation performance for small-scale problems. Additional 10 roughness profiles were randomly generated for the airfoil to be the testing samples. For each of the training and testing samples, C_{lmax} was calculated by means of CFD. As a result, the averaged prediction error of the constructed GA-SVR metamodel was 0.93%.

Once the GA-SVR model was built, it was integrated with the MCS method to conduct the uncertainty analysis. To further investigate how roughness uncertainties affect the C_{lmax} of the airfoil, several analyses were conducted with different combinations of uncertain inputs. As shown in Table 3, the parameters marked with "√" were considered as the active uncertainties, while the others were inactive with fixed deterministic values. For the variations of C_{lmax} relative to the smooth airfoil, that is, ΔC_{lmax}, the related deterministic results were calculated using the CFD method by setting each uncertain parameter to its mean level (0.21 mm for k_s and 0.15 mm for x/c). From Table 3, it can be seen that the MCS results of $\mu(\Delta C_{lmax})$ agreed well with the CFD deterministic results of the ΔC_{lmax}, implying the high accuracy of the constructed GA-SVR metamodel.

The calculated probability distributions of ΔC_{lmax} for both multiple and single uncertainties are shown in Figure 10. In Figure 10(a), the ΔC_{lmax} probability distribution curve subjected to PS roughness uncertainties was much slender and located on the right of the probability distribution curve subjected to SS roughness uncertainties. This suggests that the SS roughness distributions do have the dominant impact on ΔC_{lmax} and further explain why the SS flow behavior is more sensitive to the surface roughness as illustrated in Figure 8. From Figure 10(b), it can be seen that for either SS or PS of the airfoil, the roughness height k_s and covering length x/c had almost the same impacts on the ΔC_{lmax}.

According to the above results, the SS roughness distributions are the primary uncertainties that affect the aerodynamic performance and should be focused on in the future design and analysis of the wind turbine airfoil.

Case 2: B-spline distribution. In this case, the uncertainty analysis was based on the second model shown in Figure 9, which highlights the nonuniformity and randomness of the surface roughness distributed around

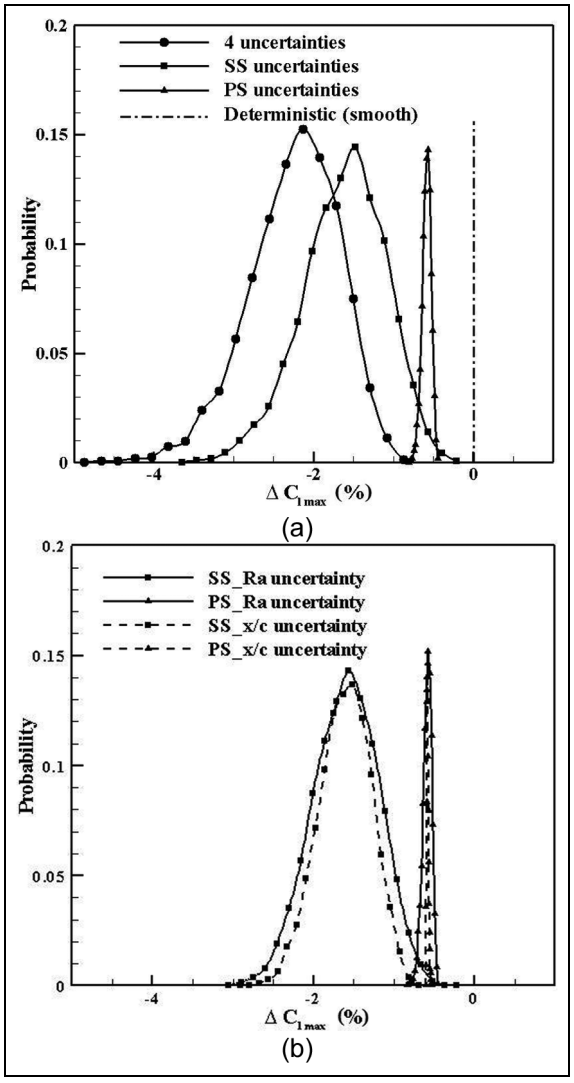

Figure 10. Probability distributions of the ΔC_{lmax} in Case 1: (a) under multiple uncertainties and (b) under single uncertainty.

the wind turbine airfoil LE. Given the results obtained in Case 1, here primary attention was focused on the SS roughness uncertainties. The eight interpolating points of the B-spline curve were taken as uncertain inputs in this analysis.

A second GA-SVR model was required to build up the relationship between the eight roughness uncertain parameters and C_{lmax}. For this large-scale approximation problem, it is recommended to use the GA-SVR with the combination of a scarce sample set (see Figure 4). Hence, the uniform DOE table $U_{17}(17^8)$ was used to generate 17 training samples. And 10 other roughness profiles were randomly generated to be the testing samples. In this case, the averaged prediction error of the constructed GA-SVR model was 2.6%.

Table 4 shows the numerical results of this uncertainty analysis, and the resulting probabilistic

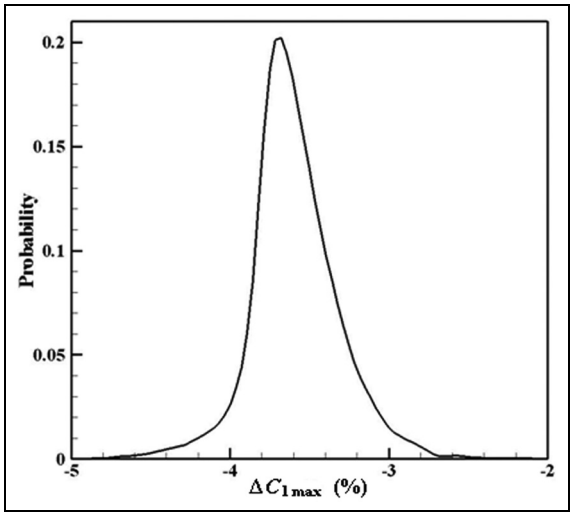

Figure 11. Probability distribution of ΔC_{lmax} in Case 2.

Table 4. Numerical results of the uncertainty analysis in Case 2.

MCS $\mu(\Delta C_{\text{lmax}})$	CFD ΔC_{lmax}	Training sample $\overline{\Delta C_{\text{lmax}}}$
−3.60%	−2.03%	−3.65%

MCS: Monte Carlo simulation; CFD: computational fluid dynamics.

distribution of ΔC_{lmax} is given in Figure 11. Compared with those in Figure 10, the probabilistic distribution curve in Figure 11 was obviously slender and shifted toward left, indicating more deteriorated performance for the wind turbine airfoil. In Table 4, the CFD deterministic results ΔC_{lmax} were calculated by setting k_s to 0.21 mm (mean level) and x/c to 0.3. Significant deviations can be observed between the MCS results $\mu(\Delta C_{\text{lmax}})$ and the CFD deterministic result ΔC_{lmax}. This is interesting and may be attributed to the spatial nonuniformity and abrupt transitions of roughness considered in this case. These particular features complicate the flow phenomenon, produce an even lower C_{lmax}, and finally worsen the aerodynamic performance of the wind turbine airfoil. Nevertheless, the MCS result of $\mu(\Delta C_{\text{lmax}})$ was observed to agree well with $\overline{\Delta C_{\text{lmax}}}$ which is the average value of the CFD deterministic results of the training samples, still confirming the high accuracy of the GA-SVR metamodel.

Above all, the GA-SVR metamodel can capture well airfoil performance variations due to different surface roughness uncertainties, which demonstrates the potential applicability of the GA-SVR metamodel in turbomachinery uncertainty analysis and provides reference for the future robust design optimization.

Conclusion

An intensive comparison has been conducted among three AI metamodeling techniques, that is, SVR, ANN, and RBF, of which the predetermined parameters were optimized using the GA method for the sake of a fair comparison. The metamodel performance evaluation strategies were carefully designed with the consideration of problem scales and sample sizes and tested by a total of 10 analytic functions. The GA-SVR metamodel was found to have the most accurate and robust performances for almost all the cases examined. It has to be remarked that the effect of noise on algorithm performance was not considered in this study, which, however, is worth studying in the future work.

The GA-SVR metamodel has been combined with the MCS method to conduct the uncertainty analysis of the wind turbine airfoil. For the investigated roughness uncertainty quantifications, the GA-SVR was capable of capturing the effect of surface roughness uncertainties on the aerodynamic performance of the wind turbine airfoil. This uncertainty analysis method based on GA-SVR and MCS is fairly generic. Work concerning the application of GA-SVR metamodel combined with MCS method for multi-objective robust design optimization of turbomachinery is currently ongoing.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: the work was supported by the National Natural Science Foundation of China (Grant No. 51406148) and the Postdoctoral Scientific foundation of China (Grant No. 2014M552444).

References

1. Egorov IN, Kretinin GV and Leshchenko IA. How to execute robust design optimization. In: *Proceedings of the 9th AIAA/ISSMO symposium on multidisciplinary analysis and optimization*, Atlanta, GA, 4–6 September 2002, paper no. 2002-5670. Reston, VA: AIAA.
2. Le Maître OP and Knio OM. *Spectral methods for uncertainty quantification: with applications to computational fluid dynamics*. New York: Springer, 2010.
3. Kleijnen JPC. *Statistical tools for simulation practitioners*. New York: Marcel Dekker, 1987.
4. Myers RH and Montgomery DC. *Response surface methodology: process and product optimization using designed experiments*. New York: Wiley, 1995.
5. Sacks J, Welch WJ, Mitchell TJ, et al. Design and analysis of computer experiments. *Stat Sci* 1989; 4: 409–435.
6. Friedman JH. Multivariate adaptive regression splines. *Ann Stat* 1991; 19: 1–67.
7. Hagan MT, Demuth HB and Beale M. *Neural network design*. Boston, MA: PWS Publishing Company, 1996.
8. Hsu KL, Gupta HV, Gao XG, et al. Self-organizing linear output map (SOLO): an artificial neural network suitable for hydrologic modeling and analysis. *Water Resour Res* 2002; 38: 1302.
9. Fang H and Horstemeyer MF. Global response approximation with radial basis functions. *Eng Optimiz* 2006; 38: 407–424.
10. Chen S, Cowan CFN and Grant PM. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE T Neural Networ* 1991; 2: 302–309.
11. Smola AJ and Schölkopf B. A tutorial on support vector regression. *Stat Comput* 2004; 14: 199–222.
12. Clarke SM, Griebsch JH and Simpson TW. Analysis of support vector regression for approximation of complex engineering analyses. *J Mech Des: T ASME* 2005; 127: 1077–1087.
13. Wang GG and Shan S. Review of metamodeling techniques in support of engineering design optimization. *J Mech Des: T ASME* 2007; 129: 370–380.
14. Simpson TW, Toropov V, Balabanov V, et al. Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come—or not. In: *Proceedings of the 12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, Victoria, BC, Canada, 10–12 September 2008, paper no. 2008-5802. Reston, VA: AIAA.
15. Chahine C, Seume JR and Verstraete T. The influence of metamodeling techniques on the multidisciplinary design optimization of a radial compressor impeller. In: *Proceedings of the ASME turbo expo 2012: turbine technical conference and exposition*, Copenhagen, 11–15 June 2012, paper no. GT2012-68358. New York: ASME.
16. Mackman TJ, Allen CB, Ghoreyshi M, et al. Comparison of adaptive sampling methods for generation of surrogate aerodynamic models. *AIAA J* 2013; 51: 797–808.
17. Sugimura K, Jeong S, Obayashi S, et al. Kriging-model-based multi-objective robust optimization and trade-off rule mining of a centrifugal fan with dimensional uncertainty. *J Comput Sci Technol* 2009; 3: 196–211.
18. Ma Y, Engeda A, Cave M, et al. Improved centrifugal compressor impeller optimization with a radial basis function network and principle component analysis. *Proc IMechE, Part C: J Mechanical Engineering Science* 2010; 224: 935–945.
19. Ju YP and Zhang CH. Multi-point and multi-objective optimization design method for industrial axial compressor cascades. *Proc IMechE, Part C: J Mechanical Engineering Science* 2011; 225: 1481–1493.
20. Jin R, Chen W and Simpson TW. Comparative studies of metamodelling techniques under multiple modelling criteria. *Struct Multidiscip O* 2001; 23: 1–13.
21. Fang H, Rais-Rohani M, Liu Z, et al. A comparative study of metamodeling methods for multiobjective crashworthiness optimization. *Comput Struct* 2005; 83: 2121–2136.
22. Tsao CC. Comparison between response surface methodology and radial basis function network for core-center

drill in drilling composite materials. *Int J Adv Manuf Tech* 2008; 37: 1061–1068.
23. Kim BS, Lee YB and Choi DH. Comparison study on the accuracy of metamodeling technique for non-convex functions. *J Mech Sci Technol* 2009; 23: 1175–1181.
24. Li YF, Ng SH, Xie M, et al. A systematic comparison of metamodeling techniques for simulation optimization in decision support systems. *Appl Soft Comput* 2010; 10: 1257–1273.
25. Paiva RM, Carvalho ARD, Crawford C, et al. Comparison of surrogate models in a multidisciplinary optimization framework for wing design. *AIAA J* 2010; 48: 995–1005.
26. Madsen HA. *Aerodynamics of a horizontal-axis wind turbine in natural conditions*. Risø-M 2903, 1991, http://orbit.dtu.dk/en/publications/aerodynamics-of-a-horizontalaxis-wind-turbine-in-natural-conditions(1f4904c3-819d-400a-9715-fcb3ffe74650).html
27. Dalili N, Edrisy A and Carriveau R. A review of surface engineering issues critical to wind turbine performance. *Renew Sust Energ Rev* 2009; 13: 428–438.
28. Ferrer E and Munduate X. CFD predictions of transition and distributed roughness over a wind turbine airfoil. In: *Proceedings of the 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, Orlando, FL, 5–8 January 2009, paper no. 2009-269. Reston, VA: AIAA.
29. Pechlivanoglou G, Fuehr S, Nayeri CN, et al. The effect of distributed roughness on the power performance of wind turbines. In: *Proceedings of the ASME turbo expo 2010: power for land, sea, and air*, Glasgow, 14–18 June 2010, paper no. GT2010-23258. New York: ASME.
30. Huang CW, Yang K, Liu Q, et al. A study on performance influences of airfoil aerodynamic parameters and evaluation indicators for the roughness sensitivity on wind turbine blade. *Sci China Technol Sci* 2011; 54: 2993–2998.
31. Sagol E, Reggio M and Ilinca A. Issues concerning roughness on wind turbine blades. *Renew Sust Energ Rev* 2013; 23: 514–525.
32. Hock W and Schittkowski K. *Test examples for nonlinear programming codes*. New York: Springer-Verlag, 1981.
33. Brainin FK. A widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM J Res Dev* 1972; 16: 504–521.
34. Simpson TW, Lin DKJ and Chen W. Sampling strategies for computer experiments: design and analysis. *Int J Reliab Appl* 2001; 2: 209–240.
35. Selig MS and McGranahan BD. *Wind tunnel aerodynamic tests of six airfoils for use on small wind turbines*. NREL/SR-500-34515, 2004, http://m-selig.ae.illinois.edu/pubs/SeligMcGranahan-2004-NREL-SR-500-34515-SixNRLAirfoils.pdf

Appendix 1

Test functions

1. $f(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1.0$
$-1.5 \le x_1 \le 4, \ -3 \le x_2 \le 3$

2. $f(x) = \left(x_2 - \frac{5.1{x_1}^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$
$-5 \le x_1 \le 10, 0 \le x_2 \le 15$

3. $f(x) = x_1 \sin(x_2) + x_2 \sin(x_1)$
$-2\pi \le x_1, x_2 \le 2\pi$

4. $f(x) = \sin\left(\frac{\pi x_1}{12}\right)\cos\left(\frac{\pi x_2}{16}\right)$
$-10 \le x_1 \le 10, \ -20 \le x_2 \le 20$

5. $f(x) = (30 + x_1 \sin(x_1)) \cdot (4 + \exp(-x_2^2))$
$0 \le x_1 \le 10, 0 \le x_2 \le 6$

6. $f(x) = \sum_{i=1}^{99} (f_i(x))^2$
$f_i(x) = -0.01i + \exp\left(-\frac{1}{x_1}(u_i - x_2)^{x_3}\right);$
$u_i = 25 + (-50\ln(0.01i))^{\frac{2}{3}}; 0.1 \le x_1 \le 100, 0 \le x_2 \le 25.6, 0 \le x_3 \le 5$

7. $f(x) = \sum_{j=1}^{5}\left[\frac{3}{10} + \sin\left(\frac{16}{15}x_j - 1\right) + \sin^2\left(\frac{16}{15}x_j - 1\right)\right]$
$-1 \le x_j \le 1$

8. $$f(x) = \sum_{j=1}^{10} \left[\frac{3}{10} + \sin\left(\frac{16}{15} x_j - 1 \right) + \sin^2\left(\frac{16}{15} x_j - 1 \right) \right]$$
$$-1 \le x_j \le 1$$

9. $$f(x) = \sum_{j=1}^{8} \left[2^{x_j - 4} + (6 - x_j) \right]$$
$$0 \le x_j \le 10$$

10. $$f(x) = \sum_{j=1}^{10} x_j \left(t_j + \ln \frac{x_j}{\sum_{j=1}^{10} x_j} \right)$$
$$t_j = 6.089, -17.164, -34.054, -5.914, -24.721, -14.986, -24.100, -10.708, -26.662, -22.179;$$
$$0.5 \le x_j \le 10$$