

Article

Multi-Fidelity Aerodynamic Data Fusion with a Deep Neural Network Modeling Method

Lei He, Weiqi Qian, Tun Zhao * and Qing Wang

Computational Aerodynamics Research Institute, China Aerodynamics Research and Development Center, Mianyang 621000, China; leiluodelei@nudt.edu.cn (L.H.); qianweiqi@cardc.cn (W.Q.); wangqing_mail@163.com (Q.W.)

* Correspondence: zhaotun@cardc.cn

Received: 7 July 2020; Accepted: 8 September 2020; Published: 12 September 2020



Abstract: To generate more high-quality aerodynamic data using the information provided by different fidelity data, where low-fidelity aerodynamic data provides the trend information and high-fidelity aerodynamic data provides value information, we applied a deep neural network (DNN) algorithm to fuse the information of multi-fidelity aerodynamic data. We discuss the relationships between the low-fidelity and high-fidelity data, and then we describe the proposed architecture for an aerodynamic data fusion model. The architecture consists of three fully-connected neural networks that are employed to approximate low-fidelity data, and the linear part and nonlinear part of correlation for the low- and high-fidelity data, respectively. To test the proposed multi-fidelity aerodynamic data fusion method, we calculated Euler and Navier–Stokes simulations for a typical airfoil at various Mach numbers and angles of attack to obtain the aerodynamic coefficients as low- and high-fidelity data. A fusion model of the longitudinal coefficients of lift C_L and drag C_D was constructed with the proposed method. For comparisons, variable complexity modeling and cokriging models were also built. The accuracy spread between the predicted value and true value was discussed for both the training and test data of the three different methods. We calculated the root mean square error and average relative deviation to demonstrate the performance of the three different methods. The fusion result of the proposed method was satisfactory on the test case, and showed a better performance compared with the other two traditional methods presented. The results provide evidence that the method proposed in this paper can be useful in dealing with the multi-fidelity aerodynamic data fusion problem.

Keywords: aerodynamic data fusion; multi-fidelity data; machine learning; deep neural networks; variable complexity modeling; cokriging

1. Introduction

In general, aerodynamic data are the main source for engineers to obtain aerodynamic performance information of aircraft, and they are generated via three types of aerodynamic testing: flight testing, wind-tunnel testing, and computational simulations [1]. Different information is provided from different sources. Flight testing can obtain the most accurate and reliable information and is often used as a final assessment; however, the flight testing is expensive, and the test cycle is long. Therefore, during the engineering development phase, aerodynamic development and analysis rely on wind-tunnel testing. Wind-tunnel testing is an important means to simulate the performance of aircraft, including the prediction of aerodynamic force/heat inside the flight envelope and the establishment of an aerodynamic database, confirming the reliability of the numerical simulation results [2]. Although the test data accuracy is relatively accurate, wind-tunnel testing is not cheap to perform in terms of cost, time or resources.

Computational fluid dynamics (CFD) is also widely applied in the aerodynamic engineering field to simulate the performance of systems, saving the cost of expensive wind tunnel experiments. In recent years, benefiting from the development of computer technology, the ability of the CFD method to simulate complex physical problems has been rapidly and constantly improving. However, such computational experiments still consume time to produce a large amount of accurate aerodynamic data and often show discrepancies in the results compared with experiments, particularly with low-fidelity computational models. Therefore, the CFD method is commonly conducted at an initial stage of aerodynamic development and is often examined by wind-tunnel testing. Combining the information provided by different fidelity data sets is often necessary to successfully produce high-quality aerodynamic data [3].

The aerodynamic characteristics are closely related to the configuration and flow field. Compared with the conventional sensor measurement data, aerodynamic data has its own complexity and particularity. Modeling-based aerodynamic data fusion methods typically need to construct a mathematical model that can describe the aerodynamic characteristics that the data fusion operation is based on. Data modeling methods are often divided into two groups: conventional aerodynamic modeling methods with explicit physical meaning and surrogate-based aerodynamic modeling methods [4]. The aerodynamic data fusion steps based on conventional models with explicit physical meaning are: the analysis of the aircraft aerodynamic characteristics, the construction of aerodynamic model, and the parameter identification of models using data from different sources.

The advantages of this method are that the model can reflect the aerodynamic characteristics in the level of physical laws and the model has the anti-interference ability on the aerodynamic noise data. The deficiencies are that the modeler must have a deep understanding of the physical theory of aerodynamics and the modeling period is always long. Surrogate-based aerodynamic data fusion method require the construction of a surrogate model. The surrogate model is more sensitive to noise data, which causes its anti-interference ability to be inferior to the former. However, the modeling efficiency is higher and the modeling time is shorter. With the development of computer technology and wind tunnel test technology, the quality of the aerodynamic data are continually increasing, and the interference amplitude noise data are decreasing.

Surrogate models are often applied in the domain of engineering design and optimization of aerospace, which can statistically approximate the relationship between a set of design variables and their response, resulting in reducing the resources required for design, search, and optimization. The surrogate models commonly used include polynomial response surface (PRS), spatial correlation models (or “kriging”), multivariate adaptive regression splines (MARS), regression trees and boosting, radial basis function (RBF) networks, and least interpolating polynomials [5–10]. Surrogate modeling is widely used to describe the performance of a system, due to its simplicity and efficiency [11,12].

A surrogate model is an approximation of the relationship between the input variable and the response of a certain system [13], and they are often used in the initial development phases to reduce the resources required for design and optimization [14,15]. However, most surrogate models still have limitations in dealing with high-dimensional problems or large scale data [16].

Taking kriging as an example, the time-consumption will increase significantly with the increase of training samples. It is also difficult for ordinary computers to support the matrix operations required by the kriging model as the amount of data points increase to a large scale. There are many matrix operations that exist in the process of building a surrogate model with the kriging method, in particular the inverse operation of the covariance matrix. If the amount of data points used to build the kriging surrogate model is n , the size of the covariance matrix is $n \times n$. In engineering practice, the amount of low-fidelity data points often reaches tens of thousands or even more; therefore, the covariance matrix involved in the model-building process would be even larger and the matrix operation would be more difficult. Over the past years, deep neural network models have become increasingly practical and important for the development of deep learning approaches. Deep neural networks have shown great

successes in dealing with large-scale data. They can easily handle linear or nonlinear problems at both low- and high-dimensions [17].

In this paper, we applied the deep neural network algorithm in fusing aerodynamic data with different fidelity levels. The paper is organized as follows. In Section 2, we discuss the related work, in particular regarding variable complexity modeling (VCM) and the cokriging method used to handle the aerodynamic data fusion problem. In Section 3, we provide the details for the multilayer perceptron (MLP) and architecture of air aerodynamic data fusion-based deep neural networks along with the training processes. In Section 4, the aerodynamic force data fusion results of a typical airfoil sharp with different methods are discussed, and the accuracy is analyzed. Our concluding remarks are provided in Section 5.

2. Related Work

Various surrogate modeling methods have been studied to fuse the information provided by multi-fidelity aerodynamic data. Hutchinson [18,19] studied a variable-complexity strategy with a scaling factor of combining simple and detailed analysis methods for the design optimization of a high-speed civil transport (HSCT) wing. A. J. Keane et al. [20] used a fusion-based method with an experimental design to solve the transonic wing optimization problem. The key of the method was to build a response surface (RSM) model of the differences between the empirical and CFD data, which have different levels of fidelity. The fusion-based method was shown to be more accurate than the initial empirical model or a simple RSM using only data from CFD. Stephen J. Leary et al. [21] described several a priori knowledge-based approaches to multi-fidelity modeling problems, in particular, the knowledge-based artificial neural networks and a new knowledge-based kriging model. Approaches that used a low-fidelity model as the prior knowledge were more effective than RSM approaches built on expensive models alone. C.Y. Tang [22] applied a variable complexity modeling method with an increment function to merge various fidelity solutions into a single, coherent database. A crew transfer vehicle (CTV), which can provide an excellent test case for the generation of aerodynamic data for its flight envelope and contains subsonic, transonic, and supersonic flows, was selected to evaluate the method. Jun Zheng et al. [5] constructed a hybrid variable-fidelity global approximation model to fuse data, where RBF was used to approximate the low-fidelity data and kriging was used to build a correction model. Maxim Tyan et al. [23] also studied the data fusion approach to construct aerodynamic tables for flight simulation using data obtained from various sources, and an additive scaling function was created to correct low-fidelity functions to match the high-fidelity data in this approach. Renganathan et al. [24] proposed a Bayesian framework to fuse two aerodynamic data sets originating from differing fidelity physical or computer experiments that could be corrupted by noise, bias, and incompleteness. M. Ghoreyshi et al. [25] described a framework based on sampling and data fusion technology to generate aerodynamic tables for flight simulation. The cokriging method was used in this study to build a data fusion model for combining samples from different fidelity sources that were expensive and cheap to evaluate. Y. Kuya et al. [26] constructed a multi-fidelity surrogate model of an inverted wing with vortex generators (VGs) in the ground effect based on cokriging regression to combine wind-tunnel experimental data and Reynolds Averaged Navier Stokes equations (RANS) computational data. Various types of sampling designs for the low-fidelity data were examined to study to what extent the low-fidelity data contributed to improving the surrogate model versus only using limited high-fidelity data. Q. Zhang [27] applied the cokriging based data fusion algorithm on CFD data and data from industrial semi-empirical methods, like the US Air Force DATCOM (data compendium) to construct an aerodynamic characteristics database.

From the above analysis, two data fusion methods are commonly used in the domain of aerodynamics, that is, variable complexity modeling and cokriging. The target of the two methods is to generate a data set that is more accurate than low-fidelity data and greater in quantity than high-fidelity data. The implementation of the methods rely on the assumption that information of low-fidelity data are used to predict global trends while high-fidelity data are used to provide absolute

value information and correct the global trends. The main ideas of the two methods are briefly explained below.

2.1. VCM

The two VCM approaches used in previous works were scaled approximations [28,29] and increment approximations. The scaling function $\sigma(x)$ defines the ratio between a high-fidelity (f_h) and a low-fidelity (f_l) solution [30].

$$\sigma(x_s) = \frac{f_h(x_s)}{f_l(x_s)}, \quad (1)$$

where x_s is the input variable at the observation points. The values of this scaling function $\sigma(x)$ are interpolated throughout the whole design space. Then the function was approximated using the expression:

$$f(x) = \sigma(x)f_l(x). \quad (2)$$

However, there are some potential problems associated with the scaling function approach when it combines low- and high-fidelity data. If the value of the low-fidelity data is exactly zero, the approach would not work. If the low-fidelity data are close to zero, $\sigma(x)$ may be quite large and may amplify any approximation errors. To avoid these possible problems, the increment function $\beta(x)$ was proposed instead of computing the ratio between the high- and low-quality data.

$$\beta(x_s) = f_h(x_s) - f_l(x_s). \quad (3)$$

Similarly, the function is approximated using the equation:

$$f(x) = \beta(x) + f_l(x). \quad (4)$$

This increment function is more reliable than a scaling ratio as the subtraction of small values does not result in any amplification errors. Kriging methods are commonly used to construct the increment surrogate model. The kriging surrogate model goes through the data points exactly, and can approximate the true model with fewer data. The fusion results can be guaranteed to be equal to the high-fidelity data completely, and the quality of the model is valid with small sample data sets.

2.2. Cokriging

To discuss the cokriging method, we first assume that low- and high-fidelity data sets are given as:

$$X = \begin{bmatrix} X_L \\ X_H \end{bmatrix} = \begin{bmatrix} X_L^{(1)} \\ \vdots \\ X_L^{(n_L)} \\ X_H^{(1)} \\ \vdots \\ X_H^{(n_H)} \end{bmatrix}, Y = \begin{bmatrix} Y_L(X_L) \\ Y_H(X_H) \end{bmatrix} = \begin{bmatrix} Y_L(X_L^{(1)}) \\ \vdots \\ Y_L(X_L^{(n_L)}) \\ Y_H(X_H^{(1)}) \\ \vdots \\ Y_H(X_H^{(n_H)}) \end{bmatrix}, \quad (5)$$

where X is the sample point, Y is the response, and we assume Y_L and Y_H are two static stochastic processes:

$$\begin{cases} Y_L = \mu_L + Z_L(x) \\ Y_H = \mu_H + Z_H(x) \end{cases} \quad (6)$$

Z is the Gaussian correlation process, and we assume that the relation between Z_L and Z_H is:

$$Z_H(x) = \rho Z_L(x) + Z_d(x), \quad (7)$$

where ρ is the scaling factor, and Z_d represents the difference between Z_H and ρZ_L . Then, a covariance matrix can be constructed as follows:

$$C = \begin{bmatrix} cov(Y_L, Y_L) & cov(Y_H, Y_L) \\ cov(Y_L, Y_H) & cov(Y_H, Y_H) \end{bmatrix} = \begin{bmatrix} \sigma_L^2 \Psi(X_L, X_L) & \rho \sigma_L^2 \Psi(X_H, X_L) \\ \rho \sigma_L^2 \Psi(X_L, X_H) & \rho^2 \sigma_L^2 \Psi(X_H, X_H) + \sigma_d^2 \Psi(X_H, X_H) \end{bmatrix} \quad (8)$$

where Ψ is the correlations between sample data points:

$$\psi(X, X') = \prod_{k=1}^n e^{-\theta_k |x_k^i - x_k^j|^{p_k}}. \quad (9)$$

We can assume $\theta_k = \theta$ and $p_k = p$ based on the isotropic hypothesis [31]. The cokriging prediction model is given by:

$$\hat{y}(x^{n_e+1}) = \hat{\mu} + cC^{-1}(y - I\hat{\mu}). \quad (10)$$

The meaning of $\hat{\mu}$, c , and other details of cokriging was discussed by A. I. J. Forrester [32]. The hyper-parameters in cokriging could be numerically estimated using a genetic algorithm or particle swarm optimization.

3. Methodology

3.1. Multilayer Perceptron

The multilayer perceptron (MLP) [33] has the ability to extract the deep hidden features of information from data efficiently and accurately. The MLP is composed of several neurons, which are connected together in a complex manner to form a network [34]. Neurons are the basic elements of the MLP. Figure 1 is a typical neuron model: $n + 1$ input, one output, and two computation functions.

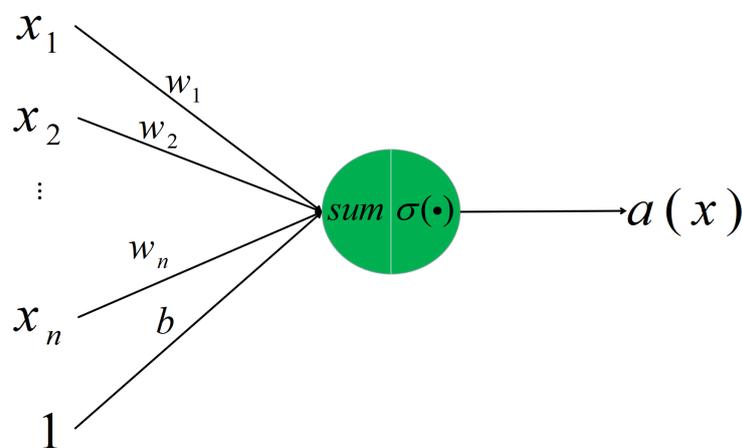


Figure 1. Typical neuron model.

The green circle represents a neuron that has $n + 1$ inputs $x_1, x_2, \dots, x_n, 1$ and one output a , where w_1, w_2, \dots, w_n are the corresponding weights to x_1, x_2, \dots, x_n and b is a bias term. The arrow represents a weighted operation, through which the input x_i will become $w_i x_i$. The neuron contains one summation function and one nonlinear activation function. The summation of the weighted inputs is

$$\sum_{i=1}^n w_i x_i + b. \quad (11)$$

The weighted input is also called a weighted signal to the neuron. The output is obtained by passing the weight signal through a nonlinear activation function $\sigma(g)$. Therefore, the output of the neuron is written as follows:

$$a = \sigma\left(\sum_{i=1}^n w_i x_i + b\right). \quad (12)$$

Figure 2 is a typical MLP neural network, which consists of four layers with full connection. Layer 1 is called the input layer, layer 2 and layer 3 are called hidden layers, and layer 4 is called the output layer. The input layer receives the input signal from the user, and the other layers receive input signals from the previous layers. The output neuron is obtained as follows:

$$a_j^{(k)} = \sigma\left(\sum_{i=1}^n w_{ji}^k x_i^{(k-1)} + b_j^k\right), \quad (13)$$

where $a_j^{(k)}$ represents the output of the j th neuron in the k th layer, w_{ji}^k denotes the weight of the connection from the i th neuron in the $(k-1)$ th layer to the j th neuron in the k th layer, b_j^k indicates the bias term of the j th neuron in the k th layer, and $\sigma(\cdot)$ is the activation function. The process of obtaining the neuron activation is called feed forward. The weights and biases of the MLP are trained using a back propagation (BP) algorithm [35].

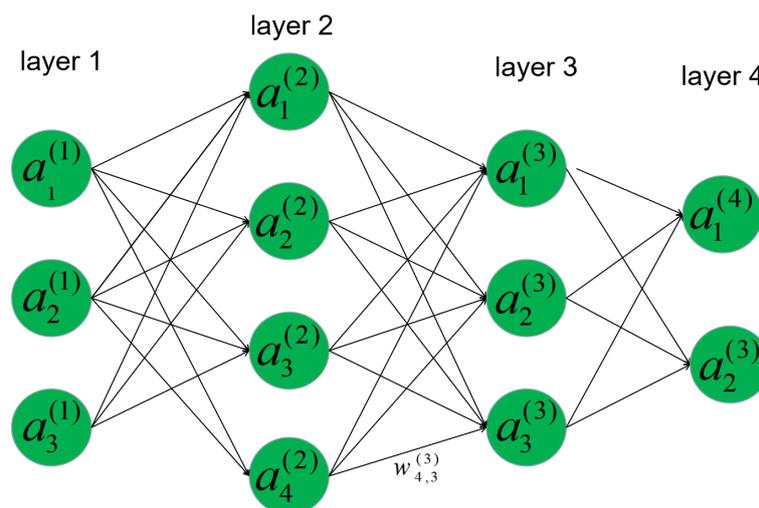


Figure 2. Typical multilayer perceptron (MLP) neural network.

3.2. Multi-Fidelity Aerodynamic Data Fusion with Deep Neural Networks

The following descriptions of multi-fidelity deep neural networks are inspired by the works of Meng [17] and Babaei [36]. We assume that low-fidelity and high-fidelity data sets are given as (X, Y_L) and (X, Y_H) . The correlation between low-fidelity and high-fidelity data can be expressed as:

$$Y_H = f(X, Y_L), \quad (14)$$

where $f(\cdot)$ is a function that maps the data from the low-fidelity level to the high-fidelity level. Generally, there exist linear and nonlinear correlations between low-fidelity and high-fidelity data [16]; then, the function $f(\cdot)$ can be decomposed into the linear and nonlinear parts, which are expressed as

$$f = f_l + f_{nl}, \quad (15)$$

where f_l and f_{nl} denote the linear and nonlinear terms in f , respectively. To describe the contribution degree of the linear and nonlinear parts to the correlation between low-fidelity and high-fidelity data, a scaling hyper-parameter ρ is used. We can further write Equation (14) as

$$Y_H = \rho f_l(X, Y_L) + (1 - \rho) f_{nl}(X, Y_L), \rho \in [0, 1]. \tag{16}$$

The value of ρ is auto determined by training data. Now, we have constructed the correlation equation.

As shown in Figure 3, the proposed architecture of the multi-fidelity aerodynamic data fusion model based on deep neural networks is composed of three fully-connected neural networks with user flow conditions as inputs, for example, the Mach number, angle of attack α , sideslip angle β , and Reynolds number Re ; and aerodynamic coefficients as outputs, for example, the coefficients of lift C_L , drag C_D , pitching moment C_M , normal force C_N , and axial force C_A . The three fully-connected neural networks can be employed to approximate the low-fidelity data, the linear part and nonlinear part of correlation for the low- and high-fidelity data. The hyperbolic tangent activation function is employed in the green neural network, and no activation function is included in the gray neural network due to the fact that it is used to approximate the linear part of the correlation. As to the number of fully connected layers, this depends on the complexity of the problem to be solved. $C_{N,L}$ and $C_{N,H}$ can be replaced by other coefficients.

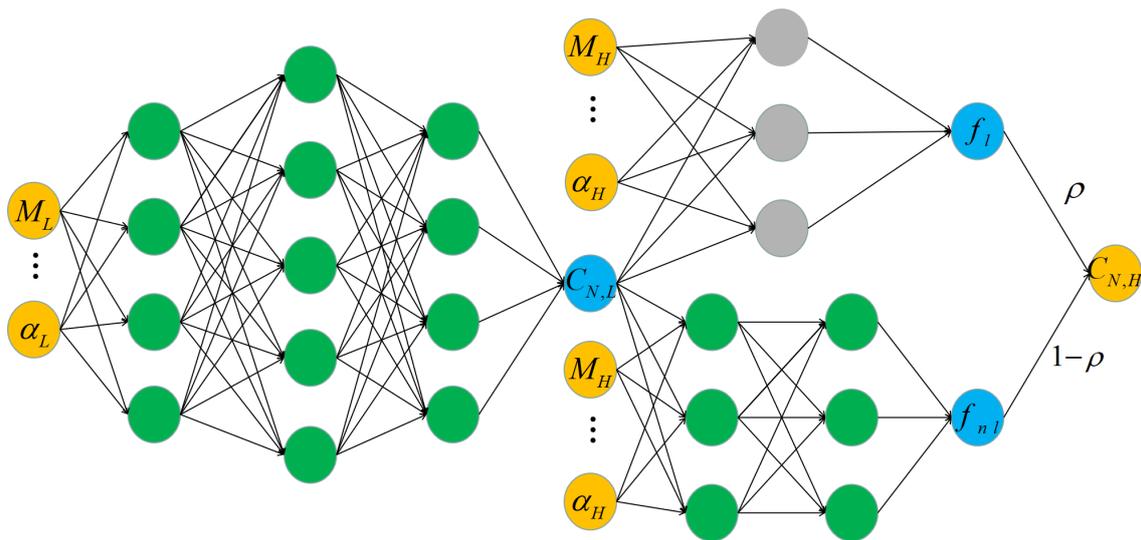


Figure 3. Architecture of the multi-fidelity aerodynamic data fusion model.

The implementation, training, and predictions of the model were performed with the open-source software Pytorch. Machine learning techniques are broadly classified into supervised and unsupervised techniques. The work here is limited to supervised machine learning techniques, in which the neural network is trained using flow conditions as input data and aerodynamic coefficients as labels. Neural network training is an optimization process in which the unknown parameters are learned by minimizing the loss function. The commonly used loss functions include the mean square error (MSE), cross entropy, categorical hinge, and so forth. In the present study, the loss function for training samples is defined as follows:

$$MSE = \frac{1}{N_L} \sum_{i=1}^{N_L} |y_L^t - y_L|^2 + \frac{1}{N_H} \sum_{i=1}^{N_H} |y_H^t - y_H|^2 + \lambda \sum w^2, \tag{17}$$

where y^t denotes the true value of the aerodynamic coefficients (labels). y is the output value of the networks, and w is weight of networks. N_L and N_H represent the number of low- and high-fidelity data sets, respectively. λ is the $L2$ regularization rate.

The training process of the model is shown in Figure 4. The main training processes of the model are feed forward calculation and error back propagation. Forward calculation obtains prediction values through the output layer. Error back propagation transfers MSE errors backward through the algorithms, such as the gradient descent, and updated trainable parameters of the network, using an optimizer. This indicates a single training iteration with a batch, which is a subset of the training set. An epoch is the full pass of the training process over the entire training set. After feed forward calculation and error back propagation, if the desired convergence was not achieved, the above steps were repeated. The end of training can be determined by the error threshold or the maximal number of epochs. The latter is used as the end condition in this paper. In this paper, the trainable parameters were initialized with Xavier's initialization method. The loss function was optimized using a stochastic gradient descent algorithm called Adam [37]. Adam is a first-order gradient-based algorithm that uses an adaptive learning rate based on past gradient information for each parameter update. So far, Adam has proved to be a good choice of algorithm for deep learning [38].

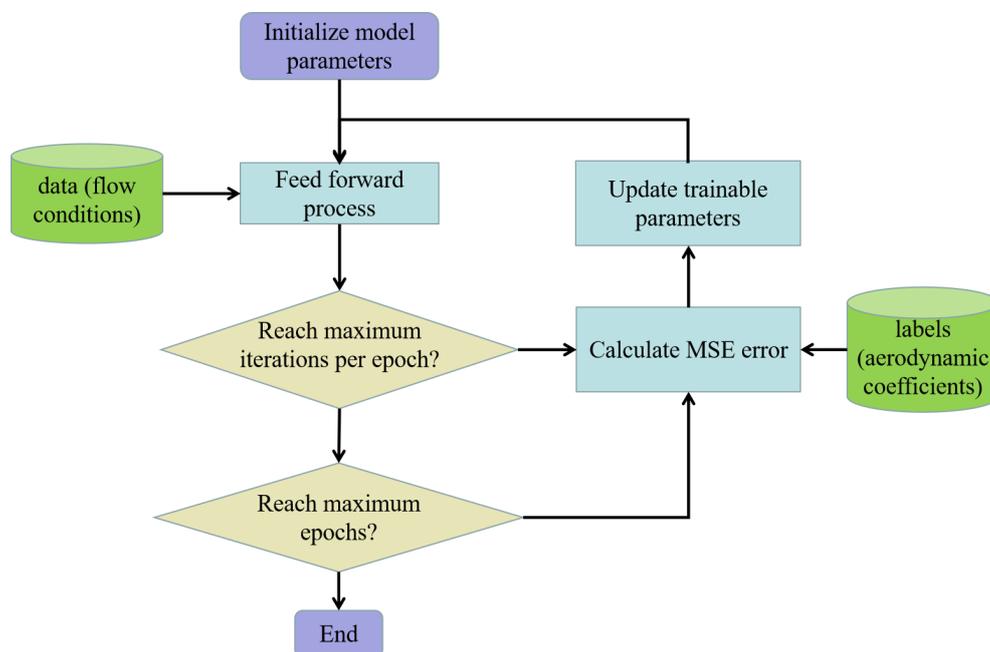


Figure 4. The training process of the proposed model.

4. Results and Discussion

We studied a two-dimensional case to verify the proposed aerodynamic data fusion method. As a comparison, VCM and the cokriging method were used to predict the aerodynamic coefficients. The VCM in this case was an increment approximation that used the kriging model to approximate both low-fidelity data and increment data, and we called this the VCM-kriging method in the following.

4.1. Data Preparation

Data are the carriers of information and data preparation is an important task in machine learning. As a test of the proposed aerodynamic data fusion approach, low-fidelity (Euler) and high-fidelity (Navier–Stokes) simulations were calculated for an airfoil sharp at various Mach numbers and angles of attack to obtain the aerodynamic force data. The tested airfoil shape, which is a variant of symmetric airfoil NACA0012, is shown in Figure 5. It was generated by improved Hicks-Henne bump function [39]. In this study, the number of bump functions was set to 8, and the values of the 8 control points were set to 0.0, 0.005, 0.0, 0.0, 0.005, 0.01, 0.005 and 0.01 respectively.

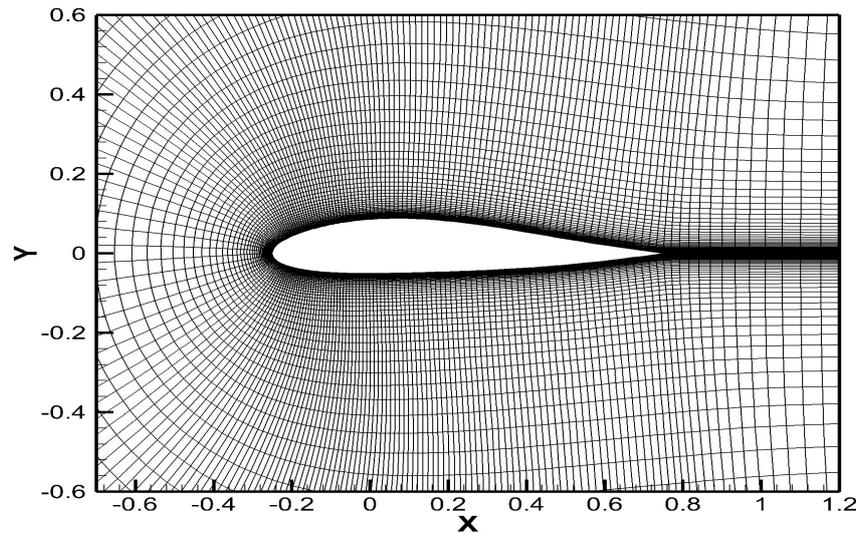


Figure 5. The tested airfoil sharp and computational grid.

The aerodynamic coefficients of the airfoil were calculated using the computational fluid software MBNS2D [40], which was independently developed by our department. A Navier–Stokes (NS) equation, Roe scheme, and a two-equation $k-\omega$ SST (Shear Stress Transfer) turbulence model were adopted for this simulation. The Reynolds number took a fixed value of 6.5×10^6 . Figure 5 also shows the X–Y plane of the computational grid. The number of grids was set to 300×100 , the grids of the leading and trailing edges were encrypted, and the first layer height in the wall-normal direction was less than $10^{-5} C$ (C is the chord length). It took approximately 350 seconds to calculate one aerodynamic coefficient at certain flow conditions with a personal computer (Intel Core i5-8250U CPU, 8G memory, and GeForce MX150 graphics card).

To study the grid convergence, four grid levels were chosen, and the grid refinement ratio was $\sqrt{2}$. Table 1 shows the simulation results of C_D and C_L at a fixed-flow condition. The relative error of Grid 2 was 3.42% and 0.8%. The convergence ratios R_G of Grids 1, 2, and 3 were 0.42 and 0.53. $0 < R_G < 1$ indicates that the simulations were monotonically convergent [41].

Table 1. The computational-fluid-dynamics (CFD) results of C_D and C_L ($\alpha = 2$, $Ma = 0.4$, $Re = 6.5 \times 10^6$).

Grid	C_D	ϵ (%)	R_G	C_L	ϵ (%)	R_G
gird1 (210 × 70)	0.01088	9.46		0.34306	1.2	
gird1 (300 × 100)	0.01028	3.42	0.42	0.34443	0.8	0.53
gird1 (420 × 140)	0.01003	0.9		0.34516	0.6	
gird1 (600 × 200)	0.00994	-	-	0.34740	-	-

In this case, we considered a two-dimensional aerodynamic force data fusion problem. To illustrate the problem, the case presented here includes the Mach number M and angle of attack α as input variables, and the longitudinal coefficients lift C_L and drag C_D were modeled. The data obtained by Euler simulation was used as the low-fidelity level and the data obtained by the Navier–Stokes simulation was used as the high-fidelity level. There were 120 and 24 training points at the low- and high-fidelity levels, respectively. The training data encompassed α and M values in the ranges:

$$\left\{ \begin{array}{l} M_L^{train} = \{-0.1, 0.2, 0.3, 0.4, 0.5, 0.6\} \\ \alpha_L^{train} = \{-4^\circ, -3^\circ, -2^\circ, -1^\circ, 0^\circ, 1^\circ, 2^\circ, 3^\circ, 4^\circ, 5^\circ, 6^\circ, 7^\circ, 8^\circ, 9^\circ, 10^\circ, 11^\circ, 12^\circ, 13^\circ, 14^\circ, 15^\circ\} \\ M_H^{train} = \{-0.1, 0.4, 0.6\} \\ \alpha_H^{train} = \{-4^\circ, 0^\circ, 1^\circ, 5^\circ, 7^\circ, 9^\circ, 11^\circ, 13^\circ, 15^\circ\}. \end{array} \right. \quad (18)$$

As illustrated in Figure 6, the green and yellow surfaces represent the high-fidelity solution (Navier–Stokes simulation) and low-fidelity solution (Euler simulation) of C_L and C_D , respectively, and the white and red points are the training data obtained from the green and yellow surfaces.

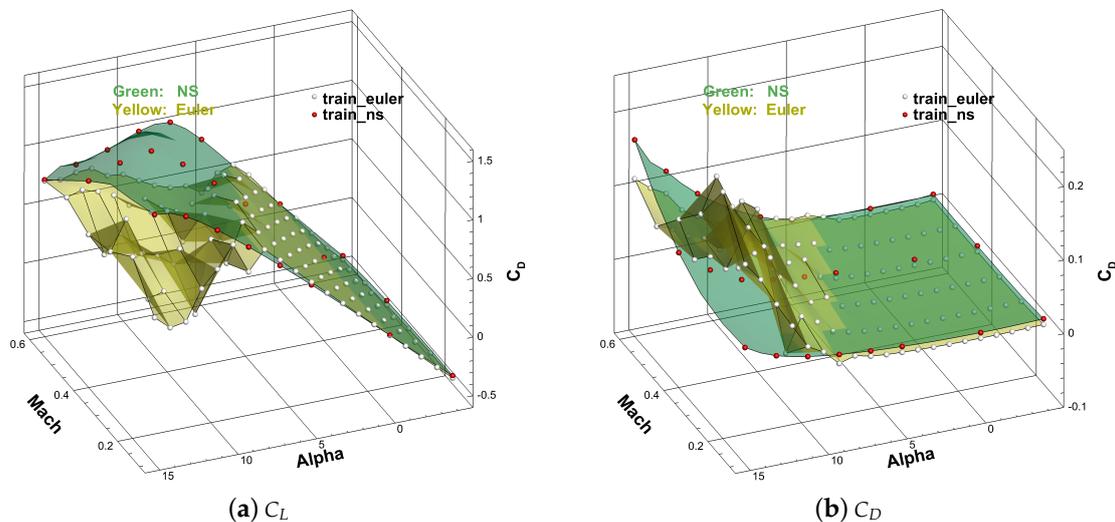


Figure 6. The low- and high-fidelity solution surfaces and training points: (a) the lift coefficient; (b) drag coefficient.

4.2. Model Training

In this case, there were four hidden layers used within the low-fidelity neural network, while two and one hidden layers were used within the nonlinear and linear neural networks, respectively. The training parameters of the proposed model were as follows: the maximal number of epochs was set to 80,000, the regularization rate was set to $\lambda = 2.5 \times 10^{-5}$ with a learning rate of 0.0001. The training convergence of the MSE for DNN architectures is shown in Figure 7, where the green and red lines indicate the convergence process of C_L and C_D . For better observation, only MSE of the first 2000 epochs are drawn. The training MSE of C_L and C_D already reached steady-state values at around 1000 epochs and 200 epochs, before which the weights of the network were rapidly tuned to optimize the prediction model. As illustrated in Table 2, it cost approximately 204 and 201 seconds to train the prediction model of C_L and C_D with the same personal computer. The training time of the model was closely related to the number of training data and the number of epochs. The VCM-kriging and cokriging methods cost less time to train.

Table 2. The time consumption of different methods.

Time	Coefficient	Proposed Method	Variable Complexity Modeling (VCM)-Kriging	Cokriging
Training time	C_L	204 s	53 s	61 s
	C_D	201 s	50 s	60 s
Prediction time	C_L	<1 s	<1 s	<1 s
	C_D	<1 s	<1 s	<1 s

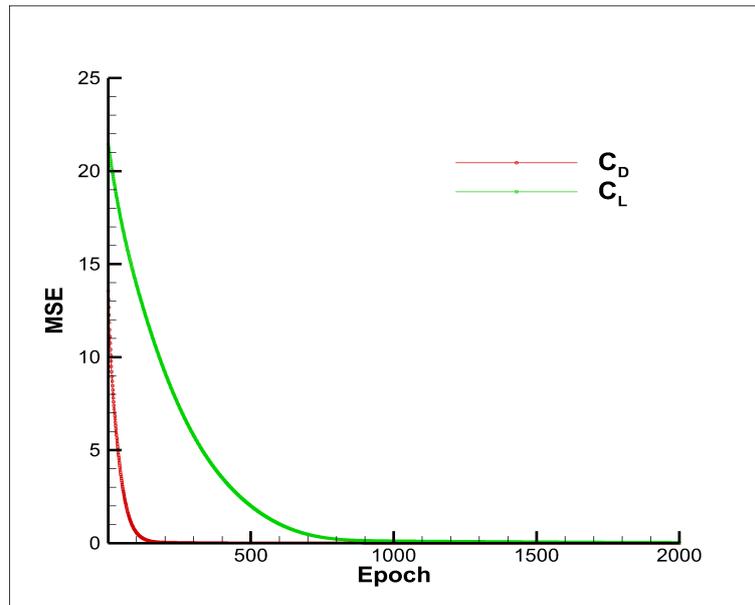


Figure 7. The training history.

4.3. Discussion

In order to test the performance of the prediction model constructed before, a validation data set was generated with the Navier–Stokes simulation. There were 96 test points that encompass the Mach number and the angle of attack values in the ranges:

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} M^{test} = \{0.2, 0.3, 0.5\} \\ \alpha^{test} = \{-4^\circ, -3^\circ, -2^\circ, -1^\circ, 0^\circ, 1^\circ, 2^\circ, 3^\circ, 4^\circ, 5^\circ, 6^\circ, 7^\circ, 8^\circ, 9^\circ, 10^\circ, 11^\circ, 12^\circ, 13^\circ, 14^\circ, 15^\circ\} \end{array} \right. \\ \left\{ \begin{array}{l} M^{test} = \{-0.1, 0.4, 0.6\} \\ \alpha^{test} = \{-3^\circ, -2^\circ, -1^\circ, 1^\circ, 2^\circ, 3^\circ, 4^\circ, 6^\circ, 8^\circ, 10^\circ, 12^\circ, 14^\circ\}. \end{array} \right. \end{array} \right. \quad (19)$$

As illustrated in Table 2, the prediction time of 96 test points with all three methods was less than 1 s, which was very short and could be ignored; therefore, the time consumption of this method was mainly during the model training. The training time of the three surrogate models was shorter than that of one CFD evaluation cost, which was approximately 6 min (350 s). CFD often needs to calculate hundreds of flight states in practical applications; thereby, the CFD method consumes more time.

Figures 8 and 9 show the predicted results of the lift and drag coefficients at training points of high-fidelity data and test points, respectively. The colored surface in the diagram represents the true high-fidelity values. The red, green, and blue points represent the predicted values with the proposed method, VCM-kriging, and cokriging, respectively. Clearly, the predicted values of the training points by all three methods almost coincide with the expected surface. As to the test points, the predicted results of the proposed method were satisfactory, and the results of other two methods were not. Thus, we can draw an intuitive conclusion that the model established in this paper could accurately predict the two aerodynamic coefficients.

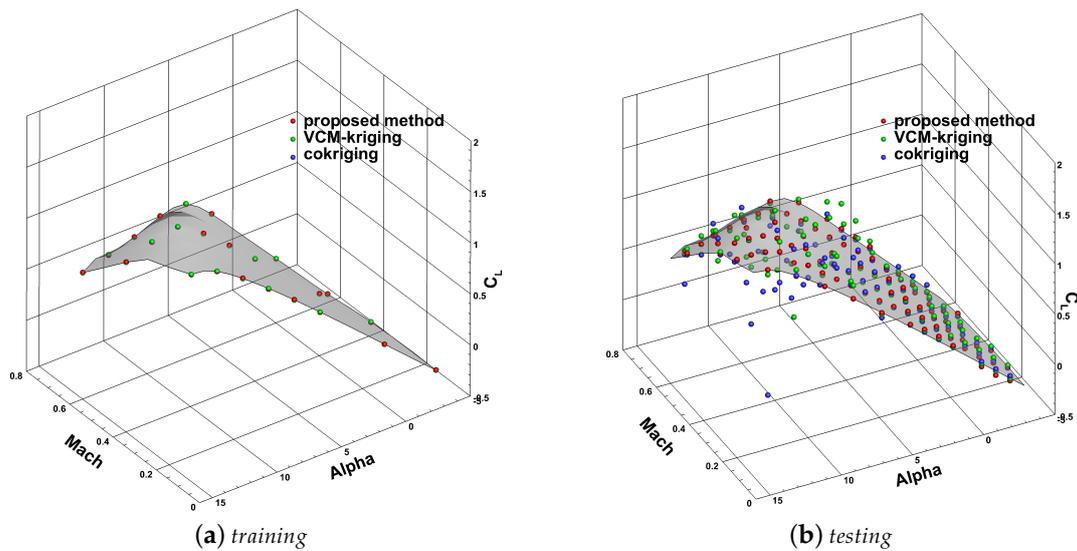


Figure 8. The predicted results of C_L with different methods: (a) training; (b) testing.

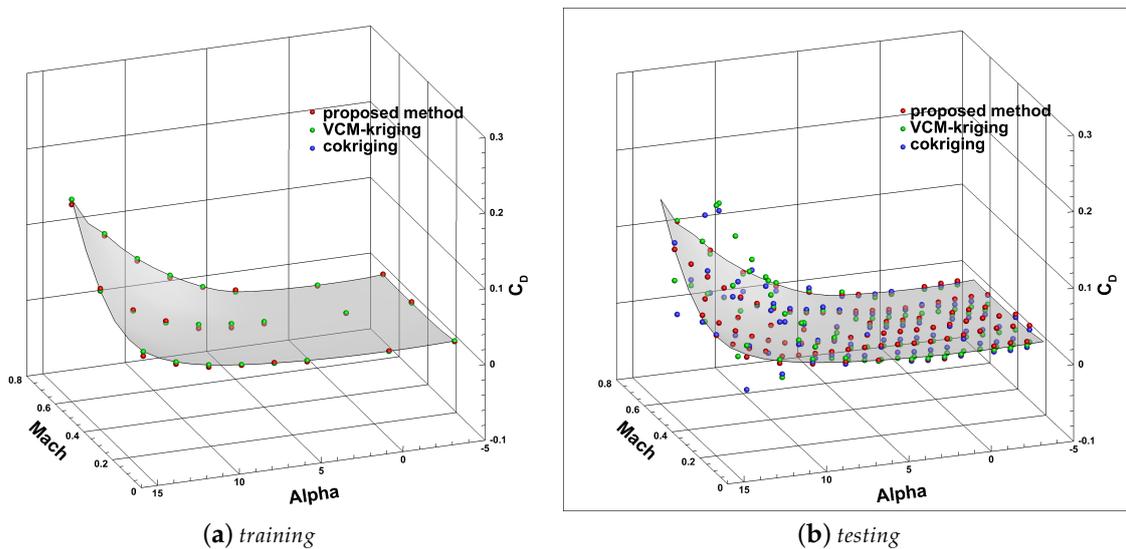


Figure 9. The predicted results of C_D with different methods: (a) training; (b) testing.

Figures 10–12 show the linear regressions of true and predicted aerodynamic coefficients. The qualitative accuracy spreads of the training samples of the three methods are shown in Figures 10 and 12. All points are clustered along the 45 deg line, the predicted values are close to the true value, which shows that all three methods show good performance in the training data sets, and the points fall accurately on the 45 deg line in subfigures (b) and (c). The qualitative accuracy spreads of the testing samples are shown in Figures 11 and 13, the points are clustered along the 45 deg line in subfigure (a), while many points are located far away the 45 deg line in subfigures (b) and (c). Thus, the performance of the proposed method is better than that of VCM-kriging and cokriging in the testing data set for the test problem.

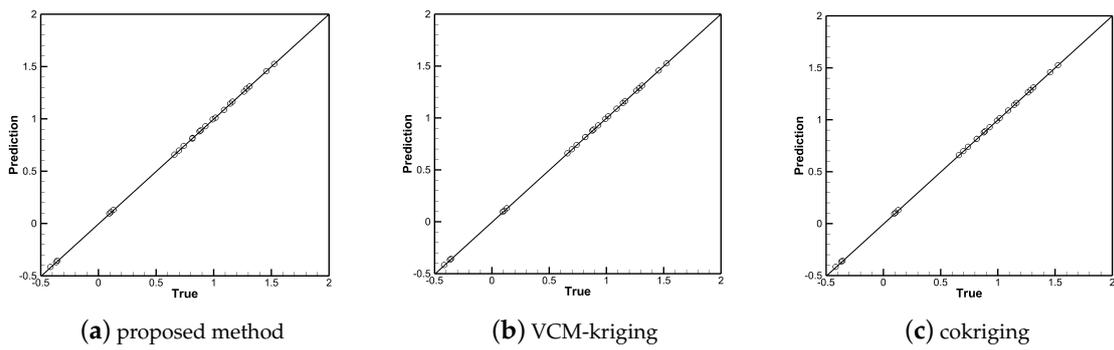


Figure 10. Accuracy spread (true vs. prediction) of the high-fidelity training points of C_L with different methods.

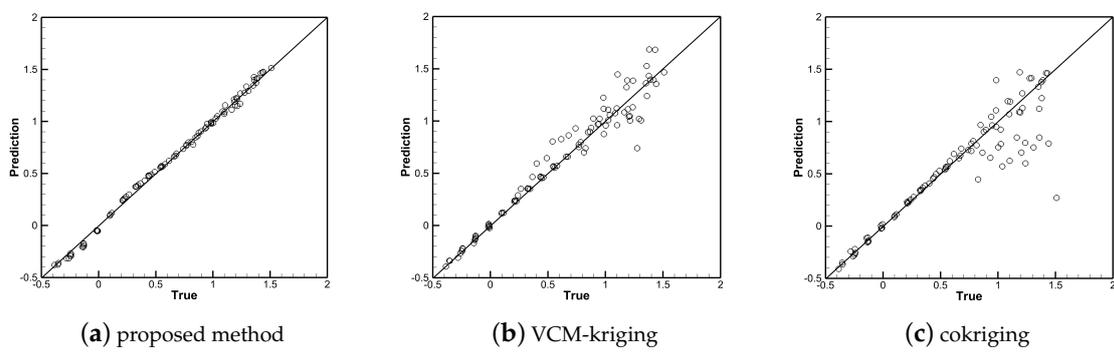


Figure 11. Accuracy spread (true vs. prediction) of the testing points of C_L with different methods.

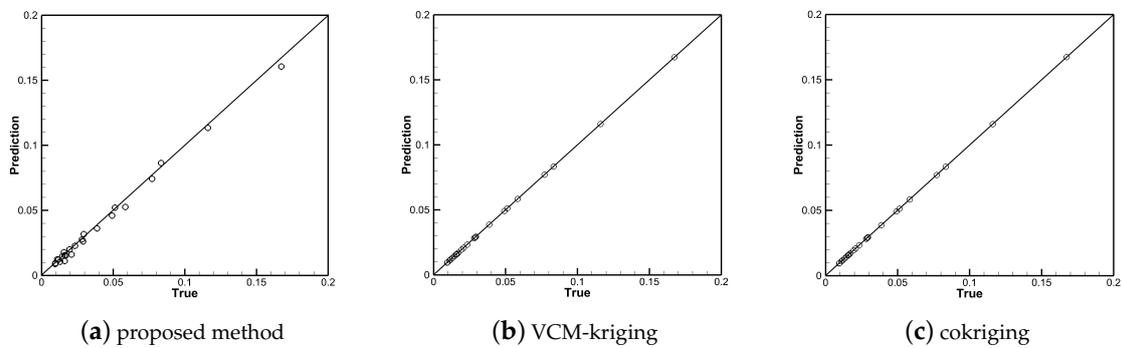


Figure 12. Accuracy spread (true vs. prediction) of the high-fidelity training points of C_D with different methods.

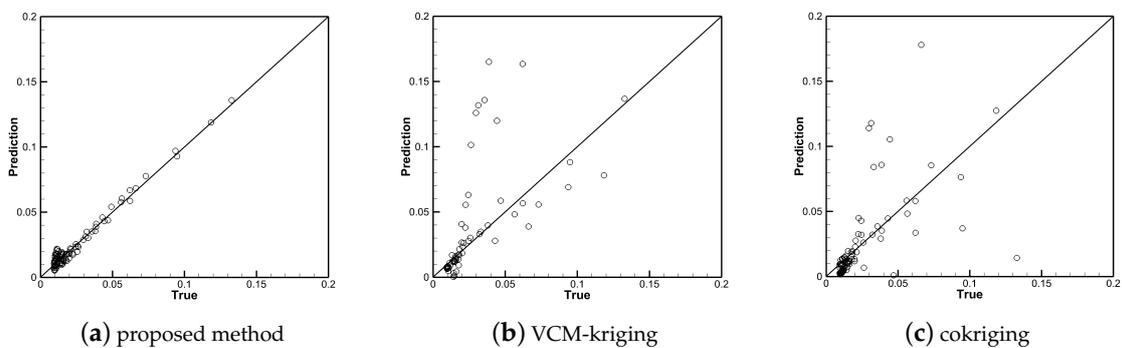


Figure 13. Accuracy spread (true vs. prediction) of the testing points of C_D with different methods.

In addition, for further performance demonstration purposes, the root mean square error (RMSE) and average relative deviation error (ζ) were used to describe the accuracy of the results. The definitions are as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_t^i - y_p^i)^2} \quad (20)$$

$$\zeta = \frac{\sum_{i=1}^N (y_t^i - y_p^i)}{\sum_{i=1}^N (y_t^i)}, \quad (21)$$

where N represents the total number of training or testing samples, y_t is the true value (high-fidelity value), and y_p is the predicted value. The errors of the training and testing data with different methods are listed in Table 3. The proposed method has a small root mean square error and average relative deviation error for the training and testing prediction, which indicates that the proposed method in this paper was able to perform well for both C_L and C_D prediction. The VCM-kriging and cokriging method demonstrated perfect performance for the training set prediction, where both the root mean square error and average relative deviation error were 0. This is because the kriging and cokriging surrogate models go through the training data points exactly. However, the VCM-kriging and cokriging method were not able to perform as well as the proposed method for both C_L and C_D prediction on the testing set, where the average relative deviation error value was 58.40% and 82.98% for C_D prediction compared with 12.89% of the proposed method.

Table 3. Errors of the training and testing data with different methods.

Set	Method	Root Mean Square Error (RMSE)		ζ	
		C_L	C_D	C_L	C_D
train	proposed method	0.0024	0.0029	0.0023	0.0584
	VCM-kriging	0.0	0.0	0.0	0.0
	cokriging	0.0	0.0	0.0	0.0
test	proposed method	0.0333	0.0038	0.0389	0.1289
	VCM-kriging	0.1220	0.0382	0.1110	0.8298
	cokriging	0.2240	0.0296	0.1627	0.5840

5. Conclusions

Data fusion technology offers great potential and prospects in the field of aerodynamics analysis. In this paper, we applied a deep neural network algorithm to fusion information contained in the multi-fidelity aerodynamic data, which provides an effective way to generate more high-quality aerodynamic data with less cost. The proposed method can learn the linear and nonlinear correlations between the low- and high-fidelity using the training sample data. The structure, training process, optimizer, and so forth of the proposed model were introduced. In the study case, both the root mean square error and average relative deviation showed that the proposed method had better performance for the test problem compared with the VCM-kriging and cokriging methods.

Although, the case only studied the multi-fidelity data obtained by different CFD solutions (Euler and Navier–Stokes), the method can also be extended to wind tunnel or flight test data. Compared with the traditional VCM-kriging and cokriging methods, the deep neural network-based fusion method had obvious advantages in high-dimensional or large-scale data problems. Therefore, more input variables can be dealt with using the method for complex aerodynamic problems involving more influence factors, and more data points can be used to build the fusion model. Thus, a large amount of historical data can be fully utilized. Despite the advantages above, an iterative process is used to determine the optimal hyper-parameters of the network, and this is tedious work for

even experienced engineers to adjust the hyper-parameters (e.g., the regularization rate) in dealing with new tasks. Future work may include experimental design of the training points by considering the historical information, expert experience, or wing geometry features. Overall, the deep neural network based multi-fidelity aerodynamic data fusion method is a promising method and can be widely applied across the spectrum of engineering.

Author Contributions: Conceptualization, L.H. and W.Q.; Methodology, L.H. and Q.W.; Validation, L.H., T.Z. and W.Q.; Formal Analysis, L.H. and T.Z.; Investigation, L.H. and W.Q.; Resources, L.H.; Data Curation, L.H. and T.Z.; Writing—Original Draft Preparation, L.H.; Writing—Review & Editing, Q.W. and T.Z.; Supervision, W.Q. 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

- Chen, J.; Zhang, Y.; Zhang, Y.; Chen, L. Review of correlation analysis of aerodynamic data between flight and ground prediction for hypersonic vehicle. *Acta Aerodyn. Sin.* **2014**, *32*, 587–599. [[CrossRef](#)]
- Hall, R.M.; Biedron, R.T.; Ball, D.N.; Bogue, D.R.; Chung, J.; Green, B.E.; Grismer, M.J.; Brooks, G.P.; Chambers, J.R. Computational Methods for Stability and Control (COMSAC): The Time Has Come. *AIAA J.* **2005**. [[CrossRef](#)]
- Kuya, Y.; Takeda, K.; Zhang, X. Optimal Surrogate Modelling Approaches for Combining Experimental and Computational Fluid Dynamics Datasets. In Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Palm Springs, CA, USA, 4–7 May 2009; p. 2216.
- Kaifeng, H.E.; Qian, W.; Wang, Q.; Kong, Y.; Wang, W. Application of data fusion technique in aerodynamics studies. *Acta Aerodyn. Sin.* **2014**, *32*, 777–782.
- Zheng, J.; Shao, X.; Liang, G.; Ping, J.; Li, Z. A hybrid variable-fidelity global approximation modelling method combining tuned radial basis function base and kriging correction. *J. Eng. Des.* **2013**, *24*, 604–622. [[CrossRef](#)]
- Rosenbaum, B.; Schulz, V. Comparing sampling strategies for aerodynamic Kriging surrogate models. *ZAMM J. Appl. Math. Mech.* **2012**, *92*, 852–868. [[CrossRef](#)]
- Queipo, N.V.; Haftka, R.T.; Shyy, W.; Goel, T.; Vaidyanathan, R.; Tucker, P.K. Surrogate-based analysis and optimization. *Prog. Aerosp. Sci.* **2005**, *41*, 1–28. [[CrossRef](#)]
- Ai, Y. Research on Response Surface Method Optimisation Based on Radial Basis Function. Master's Thesis, Huazhong University of Science and Technology, Wuhan, China, 2012.
- Barton, R.R. Simulation optimization using metamodels. In Proceedings of the 2009 Winter Simulation Conference (WSC), Austin, TX, USA, 13–16 December 2009; pp. 230–238.
- Younis, A.; Dong, Z. Trends, features, and tests of common and recently introduced global optimization methods. *Eng. Optim.* **2010**, *42*, 691–718. [[CrossRef](#)]
- Laurenceau, J.; Sagaut, P. Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging. *AIAA J.* **2008**, *46*, 498–507. [[CrossRef](#)]
- Forrester, A.; Sobester, A.; Keane, A. *Engineering Design via Surrogate Modelling: A Practical Guide*; John Wiley & Sons: Chichester, UK, 2008.
- Chen, V.C.P.; Tsui, K.; Barton, R.R.; Meckesheimer, M. A review on design, modeling and applications of computer experiments. *IIE Trans.* **2006**, *38*, 273–291. [[CrossRef](#)]
- Santner, T.J.; Williams, B.J.; Notz, W.; Williams, B.J. *The Design and Analysis of Computer Experiments*; Springer: New York, NY, USA, 2003; Volume 1.
- Park, C.; Haftka, R.T.; Kim, N.H. Remarks on multi-fidelity surrogates. *Struct. Multidiscip. Optim.* **2017**, *55*, 1029–1050. [[CrossRef](#)]
- Perdikaris, P.; Raissi, M.; Damianou, A.; Lawrence, N.D.; Karniadakis, G.E. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proc. Math. Phys. Eng. Sci.* **2017**, *473*, 20160751. [[CrossRef](#)] [[PubMed](#)]
- Meng, X.; Karniadakis, G.E. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems. *J. Comput. Phys.* **2020**, *401*, 109020. [[CrossRef](#)]

18. Hutchison, M.; Mason, W.; Grossman, B.; Haftka, R. Aerodynamic Optimization of an HSCT Configuration Using Variable-Complexity Modeling. In Proceedings of the 31st Aerospace Sciences Meeting, Reno, NV, USA, 11 January 1993. [[CrossRef](#)]
19. Hutchison, M.G.; Unger, E.R.; Mason, W.H.; Grossman, B.; Haftka, R.T. Variable-complexity aerodynamic optimization of a high-speed civil transport wing. *J. Aircr.* **1994**, *31*, 110–116. [[CrossRef](#)]
20. Keane, A.J. Wing Optimization Using Design of Experiment, Response Surface, and Data Fusion Methods. *J. Aircr.* **2003**, *40*, 741–750. [[CrossRef](#)]
21. Leary, S.J.; Bhaskar, A.; Keane, A.J. A Knowledge-Based Approach To Response Surface Modelling in Multifidelity Optimization. *J. Glob. Optim.* **2003**, *26*, 297–319. [[CrossRef](#)]
22. Tang, C.; Gee, K.; Lawrence, S. Generation of aerodynamic data using a design of experiment and data fusion approach. In Proceedings of the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 10–13 January 2005; p. 1137.
23. Tyan, M.; Kim, M.; Pham, V.; Choi, C.K.; Nguyen, T.L.; Lee, J.W. Development of Advanced Aerodynamic Data Fusion Techniques for Flight Simulation Database Construction. In Proceedings of the 2018 Modeling and Simulation Technologies Conference, Atlanta, GA, USA, 25–29 June 2018.
24. Renganathan, A.; Harada, K.; Mavris, D.N. Multifidelity Data Fusion via Bayesian Inference. In Proceedings of the AIAA Aviation 2019 Forum, Dallas, TX, USA, 17–21 June 2019.
25. Ghoreyshi, M.; Badcock, K.J.; Woodgate, M.A. Accelerating the Numerical Generation of Aerodynamic Models for Flight Simulation. *J. Aircr.* **2009**, *46*, 972–980. [[CrossRef](#)]
26. Kuya, Y.; Takeda, K.; Xin, Z.; Forrester, A.I.J. Multifidelity Surrogate Modeling of Experimental and Computational Aerodynamic Data Sets. *AIAA J.* **2011**, *49*, 289–298. [[CrossRef](#)]
27. Zhang, Q. Development of a Data Fusion Framework for the Aerodynamic Analysis of Launchers. Master's Thesis, Delft University of Technology, Delft, The Netherlands, 2017.
28. Unger, E.R.; Hutchinson, M.G.; Rais-Rohani, M.; Haftka, R.T.; Grossman, B. Variable-Complexity Design of a Transport Wing. *Int. J. Syst. Autom. Res. Appl.* **1992**, *2*, 87–113.
29. Knill, D.; Giunta, A.; Baker, C.; Grossman, B.; Mason, W.; Haftka, R.; Watson, L. HSCT Configuration Design Using Response Surface Approximations of Supersonic Euler Aerodynamics. In Proceedings of the 36th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, USA, 12–15 January 1998. [[CrossRef](#)]
30. Baker, C.A.; Grossman, B.; Haftka, R.T.; Mason, W.H.; Watson, L.T. High-speed civil transport design space exploration using aerodynamic response surface approximations. *J. Aircr.* **2002**, *39*, 215–220. [[CrossRef](#)]
31. Han, Z. Kriging surrogate model and its application to design optimization: A review of recent progress. *Acta Aeronaut. Astronaut. Sin.* **2016**, *37*, 3197–3225.
32. Forrester, A.I.; S̄abester, A.; Keane, A.J. Multi-fidelity optimization via surrogate modelling. *Proc. R. Soc. Math. Phys. Eng. Sci.* **2007**, *463*, 3251–3269. [[CrossRef](#)]
33. Santos, M.; Mattos, B.; Girardi, R. Aerodynamic Coefficient Prediction of Airfoils Using Neural Networks. In Proceedings of the AIAA Aerospace Sciences Meeting & Exhibit, Reno, NV, USA, 7–10 January 2008.
34. Sekar, V.; Zhang, M.; Shu, C.; Khoo, B.C. Inverse Design of Airfoil Using a Deep Convolutional Neural Network. *AIAA J.* **2019**, *57*, 993–1003. [[CrossRef](#)]
35. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Representations by Back Propagating Errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
36. Babae, H.; Perdikaris, P.; Chryssostomidis, C.; Karniadakis, G.E. Multi-fidelity modelling of mixed convection based on experimental correlations and numerical simulations. *J. Fluid Mech.* **2016**, *809*, 895–917. [[CrossRef](#)]
37. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
39. Wang, J.J.; Gao, Z.H. Analysis and Improvement of HicksHenne Airfoil Parameterization Method. *Aeronaut. Comput. Tech.* **2010**, *40*, 47–49.

40. Wang, J.; Yi, X.; Xiao, Z. Numerical simulation of ice shedding from ARJ21-700. *Acta Aerodyn. Sin.* **2013**, *31*, 430–436.
41. Chen, K.; Huang, D.; Li, Y. Grid convergence study in the resistance calculation of a trimaran. *J. Mar. Sci. Appl.* **2008**, *7*, 174–178. [[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/>).