IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 28, 2020, accepted May 4, 2020, date of publication May 11, 2020, date of current version May 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993562

Neural Networks-Based Aerodynamic Data Modeling: A Comprehensive Review

LIWEI HU, (Student Member, IEEE), JUN ZHANG, YU XIANG, (Member, IEEE), AND WENYONG WANG, (Member, IEEE)
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
Corresponding author: Jun Zhang (zhangjun@uestc.edu.cn)

This work was supported by the National Numerical Wind Tunnel Project No. 13RH19ZT6B1.

ABSTRACT This paper reviews studies on neural networks in aerodynamic data modeling. In this paper, we analyze the shortcomings of computational fluid dynamics (CFD) and traditional reduced-order models (ROMs). Subsequently, the history and fundamental methodologies of neural networks are introduced. Furthermore, we classify the neural networks based studies in aerodynamic data modeling and illustrate comparisons among them. These studies demonstrate that neural networks are effective approaches to aerodynamic data modeling. Finally, we identify three important trends for future studies in aerodynamic data modeling: a) the transformation method and physics informed models will be combined to solve high-dimensional partial differential equations; b) in the research area of steady aerodynamic response predictions, model-oriented studies and data-integration-oriented studies will become the future research directions, while in unsteady aerodynamic response predictions, radial basis function neural networks (RBFNNs) are the best tools for capturing the nonlinear characteristics of flow data, and convolutional neural networks (CNNs) are expected to replace long short-term memories (LSTMs) to capture the temporal characteristics of flow data; and c) in the field of steady or unsteady flow field reconstructions, the CNN-based conditional generative adversarial networks (cGANs) will be the best frameworks in which to discover the spatiotemporal distribution of flow field data.

INDEX TERMS Aerodynamics, convolutional neural networks, neural networks, generative adversarial networks, recurrent neural networks.

I. INTRODUCTION

Aerodynamic data modeling refers to the approximation of the mapping function between input and output data using appropriate models [1].

Generally, the governing equations, usually ordinary differential equations (ODEs) or partial differential equations (PDEs), in aerodynamics express the mapping between input data (design parameters) and output data (response parameters). Computational fluid dynamics (CFD) [2] has the ability to approximate the governing equations to calculate response values under a restricted flow condition. However, the shortcomings of CFD are obvious: a) the accuracy of calculation is easily affected by the mesh density; b) CFD is a time-consuming approach; and c) even worse, some complex governing equations have no numerical solutions.

To overcome the drawbacks of CFD, reduced-order models (ROMs) [3] were proposed. ROMs simplify the governing equations and improve the solution efficiency. However, there are always three problems with aerodynamic ROMs. The first problem is that some mappings between outputs and inputs in aerodynamics are highly nonlinear, which makes it more difficult for ROMs to process aerodynamic data. The second problem is that tens of thousands of samples are needed for training to guarantee the accuracy of ROMs. Paradoxically, using CFD to calculate these sample data is very time-consuming. The third problem is that some special aerodynamic problems possess spatial and temporal characteristics, for example, turbulence [4]. These problems undoubtedly reduce the credibility of ROMs.

Therefore, one tough question in aerodynamic data modeling is what kind of technology is appropriate to build highly reliable aerodynamic data models?

Nerual networks, as a data-driven learning method, has achieved great success in computational vision [5]–[7],

The associate editor coordinating the review of this manuscript and approving it for publication was Vivek Kumar Sehgal.

natural language processing [8], [9], and nonlinear system identification [10]–[12]. These works illustrate that neural networks have the ability to learn the conditional probability $P(X|Y)$ between input data X and output data Y within a short time from the training set. There exist some notable examples for exploring neural network algorithms in physics and engineering [13]. Researchers believe that the combination of neural networks and aerodynamics will be a way to solve many problems in the field of aerodynamic data modeling [14]–[16]. As a matter of fact, neural networks provide many sophisticated models or algorithms to address a number of problems in aerodynamics, including solutions to the governing equations, aerodynamic force predictions, flow field reconstructions, and so on [17].

Compared to traditional ROMs, the advantages of applying neural networks in aerodynamic data modeling lie in the fact that a) neural networks, as a data-driven model, does not rely on aerodynamic theories or physical models, which means it does not require thorough knowledge on aerodynamics; b) neural networks can be used to address high-dimensional [18], multiscale [19] and nonlinear [20] problems that are difficult for traditional ROMs; and c) some neural networks [21], [22] have the ability to process time sequence data. Apparently, these advantages answer the question posed earlier.

Different from images and language, flow data usually have both spatial and temporal variations; particularly, the distribution of flow data is highly nonuniform. Consequently, aerodynamic data modeling presents new challenges to neural networks: a) complex flow changes require the ability of the neural networks to express the physical mechanism of aerodynamics; b) some complex flow phenomena (for example, turbulence) that cannot be effectively recognized in CFD, require neural networks to provide specialized solutions; and c) small aerodynamic flow data require neural networks to learn the underlying aerodynamic features as much and as accurately as possible from a limited number of samples.

To meet these new challenges, some new neural networks that are suitable for aerodynamic data modeling should be studied. Take a look at the current works in this field: the most commonly used models are multilayer perceptrons (MLPs), convolutional neural networks (CNNs), radial basis function neural networks (RBFNNs), recurrent neural networks (RNNs) and generative adversarial networks (GANs), etc. These models are usually used in solving ODEs/PDEs, aerodynamic response predictions, flow field reconstructions and so on. Figure 1 shows some important areas in aerodynamic data modeling, namely solving PDEs/ODEs, nonlinear aerodynamic response predictions, flow field reconstructions and some scattered studies. For example, in nonlinear aerodynamic response predictions, steady and unsteady nonlinear aerodynamic response predictions are discussed. In steady nonlinear aerodynamic response predictions, we divide the existing studies into two categories: predictions based on flow state and predictions based on aerodynamic shapes and flow state. In unsteady nonlinear aerodynamic response predictions, we also divide the existing studies into two categories: spatial nonlinearity oriented predictions and temporal continuity oriented predictions.

The contributions of this paper are a) we sort out the applications of common neural networks in the field of aerodynamic data modeling from the perspective of neural networks; b) we compare these studies and analyze the advantages and disadvantages of them; and c) we highly evaluate some excellent models/methods [14], [23], [24], and point out the future development trend in this field based on them.

The structure of this review is as follows. Section II describes the history of neural networks, as well as the details of the models mentioned above. Section III describes the applications of neural networks in solving ODEs/PDEs. In Section IV, we introduce the applications of neural networks in nonlinear aerodynamic response predictions. In Section V, the applications of neural networks in flow field reconstructions are expounded. In Section VI, we introduce some scattered applications. In Section VII, we summarize the existing research, and future prospects of combining aerodynamics with neural networks are discussed.

II. NEURAL NETWORK FUNDAMENTALS

A. HISTORY OF NEURAL NETWORKS

The combination of aerodynamics and artificial intelligence dates back to the 1940s. Kolmogorov [25] adopted statistical learning methods to address turbulence problems. Neural networks are a science developed from statistical learning. The original concept of MLPs was proposed by McCulloch and Pitts [26]. They attempted to explain the mechanism by which the human brain handles complex tasks. Based on their research, Rosenblatt [27] proposed the single-layer perceptron. However, it was found in 1969 that the perceptron with only one hidden layer were not able to learn the XOR function. Motivated by this drawback, the MLP structure was proposed by Minsky [28], but how to train the parameters of this model is still unresolved. Consequently, the MLPs entered an age of winter (for example, see the ''Lighthill report''of 1974).

In the mid-1980s, Rumelhart *et al.* [29] proposed the back propagation algorithm, which solved the training problem of MLPs. Since then, MLPs have developed into neural networks, which marks the begining of neural networks. However, due to the shortage of the computing power of computers in those days, as well as the limitations of the hardware, the maximum number of hidden layers in MLPs is two. The development of neural networks was restricted and entered an age of winter once again.

At the turn of the century, with the appearance of neural networks, there was an explosion in the study of the combination of aerodynamics and neural networks, especially in the areas of aeroelasticity, particle image velocimetry and flow field reconstruction [30]–[32]. Among these studies,

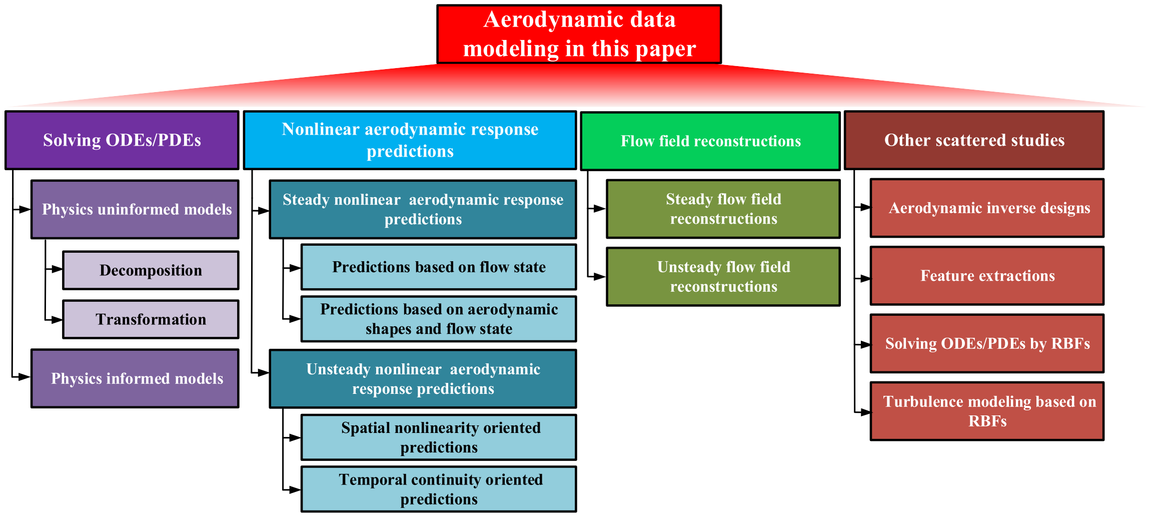
Aerodynamic data modeling in this paper
Solving ODEs/PDEs
Physics uninformed models
Decomposition
Transformation
Physics informed models
Nonlinear aerodynamic response predictions
Steady nonlinear aerodynamic response predictions
Predictions based on flow state
Predictions based on aerodynamic shapes and flow state
Unsteady nonlinear aerodynamic response predictions
Spatial nonlinearity oriented predictions
Temporal continuity oriented predictions
Flow field reconstructions
Steady flow field reconstructions
Unsteady flow field reconstructions
Other scattered studies
Aerodynamic inverse designs
Feature extractions
Solving ODEs/PDEs by RBFs
Turbulence modeling based on RBFs

FIGURE 1. The areas of aerodynamic data modeling discussed in this paper.

Milano [32] was the first to use a true neural network to simulate compressible fluids.

The significant improvement in computing power has motivated the development of neural networks. In 2006, Hinton *et al.* [33] put forward the method of deep belief networks (DBNs), which overcame the difficulty of neural network training with the help of modern computers. Hinton's research opened the era of deep learning. This also marks the formal beginning of the integration of neural networks and aerodynamics.

The goal of neural network models can be summarized as estimating the parameters in a limited number of iterations so that the convergent model can map inputs to outputs [34]. In this section, we mainly introduce the fundamentals of the most commonly used neural networks (i.e., MLPs, CNNs, RBFNNs, RNNs and GANs) in aerodynamic modeling, as well as the training processes.

B. MULTILAYER PERCEPTRONS

Multilayer perceptrons (the MLPs in this paper refer to fully connected neural networks) are the most common nonlinear models in the field of deep learning. It has been proved that an MLP with enough layers has the ability to approximate any function [35]. This proof lays a theoretical foundation for the application of MLPs.

An MLP is composed of layers of artificial neurons (neural nodes), see Figure 2. One neuron in a specific layer is connected to all neurons in the next layer with the corresponding weights and biases. An MLP consists of one input layer, one output layer and several hidden layers. Neurons in the input layer receive input data, neurons in the hidden layers process the input data, and neurons in the output layer output the final results. The training process of an MLP is composed of two parts: forward propagation and back propagation [36].

Forward propagation describes how the input data propagate from the input layer forward to the output layer.

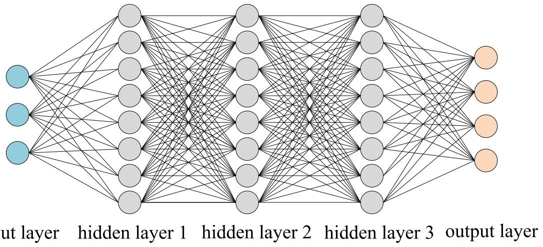

FIGURE 2. A four-layer MLP. Because the input layer does not participate in the calculation, this is a four-layer conceptron.

As shown in Figure 3, the input layer has n nodes, and the hidden layer has m nodes. w_{ij}^{l} denotes the weight between the i_{th} neuron at layer $l-1$ and the j_{th} neuron at layer l. b_j^l denotes the bias of the j_{th} neuron at layer l. The forward propagation can be expressed as in equation 1, where z denotes the current sample processed by this MLP and $f(.)$ denotes the activation function of neurons at layer l in this MLP.

$$h_j^l = f(\sum_{i=1}^{n} w_{ij}^l x_{zi} + b_j^l), \quad j = 1, 2, \ldots, m \tag{1}$$

Gradient based back propagation algorithm is the most commonly used method. The latest studies show that back propagation approach based on quasi-Newton method [17] is also effective. In this section, we still focus on gradient based back propagation, because it is widely applied in many different neural networks. Back propagation is a nonlinear optimization algorithm, which describes how errors in an MLP pass from the output layer back to the input layer. The errors of an MLP propagate backwards in the form of a gradient. Through multiple back propagation processes, the parameters (including the weights and biases) of an MLP are determined such that the loss function of the MLP is very near its minimum value. Equations 2 and 3 formulate the process of updating the weights and biases, where E denotes

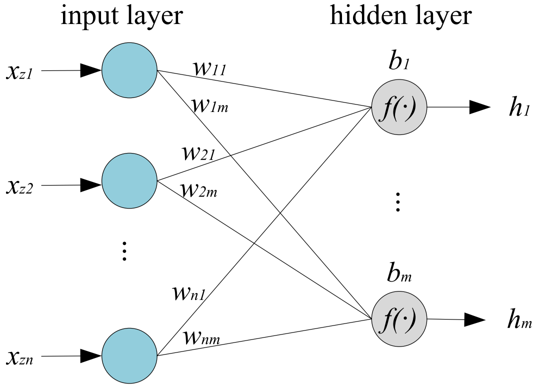

FIGURE 3. Forward propagation. For example, the input of the first node in the hidden layer is $x_{z1} \cdot w_{11} + x_{z2} \cdot w_{21} + \cdots + x_{zn} \cdot w_{n1}$, while the output of this node is $f(x_{z1} \cdot w_{11} + x_{z2} \cdot w_{21} + \cdots + x_{zn} \cdot w_{n1} + b_1)$.

the loss function of an MLP and η denotes the learning rate. Equation 4 describes a feasible error calculation formula, where y_i denotes the output value of the ith neuron in the output layer and d_i denotes the corresponding true value.

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \eta \frac{\partial E}{\partial W_{ij}^{(l)}} \tag{2}$$

$$b_i^{(l)} = b_i^{(l)} - \eta \frac{\partial E}{\partial b_i^{(l)}} \tag{3}$$

$$E = \frac{\sum_{i=1}^{n} (y_i - d_i)^2}{2n} \tag{4}$$

C. CONVOLUTIONAL NEURAL NETWORKS

CNNs for processing image data are special models developed on the basis of MLPs. Image data are different from general statistical data, for example, a pixel in an image is usually relevant with respect to its surrounding pixels or even pixels that are far away. Therefore, CNNs change the form of weights in MLPs: the weights in full connections between two different layers are replaced with convolutional kernels [37]. The appearance of convolutional kernels makes it possible to consider the local information of image data in the training process of CNNs. CNNs learn the distribution of image data by constantly moving convolutional kernels, which is impossible for MLPs. The special structure and training methods are more consistent with how the brain processes what we see. Consequently, CNNs are commonly used in the field of computer vision. Remarkably, the latest studies [21], [38] illustrated that CNNs could address time sequence data processing, even better than RNNs. We will make a detail introduction in IV.B.2).

Figure 4 illustrates a classic CNN structure. The network is composed of several different layers: convolutional layers, pooling layers and fully connected layers.

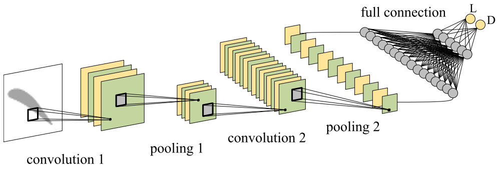

FIGURE 4. A classic CNN in nonlinear steady aerodynamic response predictions. The network consists of two convolutional layers, two pooling layers and two fully connected layers. By the CNN, the corresponding aerodynamic responses, such as the lift (L) and drag (D) of the input wing, can be identified.

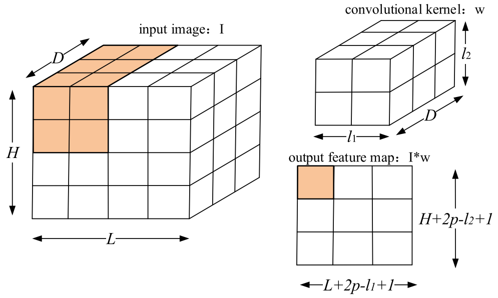

FIGURE 5. Three-dimensional convolution operation with only one convolutional kernel. Equation 5 describes a convolution operation of the red part in this graph.

The function of a convolutional layer is achieved by convolutional kernels (i.e., filters), which can fully use global and local spatial information of the input pictures. The weights in the convolutional kernels and biases are the parameters to learn. Equation 5 and Figure 5 describe one three-dimensional convolution operation with only one convolutional kernel (the stride equals to 1), where I denotes the input image, which is an $L \times H \times D$ matrix, w denotes the weight matrix with a size of $l_1 \times l_2 \times D$, b denotes the bias of this convolutional layer, and $f(.)$ denotes the activation function. In particular, the depth of the input image I is generally equal to the depth of the convolutional kernel w in the field of image processing. The output of this convolutional layer (i.e., $I * w$) is a two-dimensional matrix with shape $(L + 2 \times p - l_1 + 1) \times (H + 2 \times p - l_2 + 1)$, where p denotes the padding value of this convolutional operation. Indeed, the output of a convolution operation with only one convolutional kernel is a two-dimensional matrix. In other words, the number of convolutional kernels determines the depth of the output matrix.

$$(I * w)_{ij} = f(\sum_{m=0}^{l_1-1} \sum_{n=0}^{l_2-1} \sum_{d=0}^{D-1} w_{m,n,d} \cdot I_{i+m,j+n,d} + b) \tag{5}$$

$$0 < i < L + 2 \times p - l_1 + 1 \tag{6}$$

$$0 < j < H + 2 \times p - l_2 + 1 \tag{7}$$

The pooling layer usually follows a convolutional layer, but it is often not required. The objective of the pooling layer is to reduce the dimension of the data to avoid overfitting under the premise of keeping the features unchanged. The most common types of pooling operations are maximum pooling and average pooling. The maximum pooling operation is performed to select the maximum value in the local region of the output matrix of one convolutional layer. In a similar way, the average pooling operation calculates the average value.

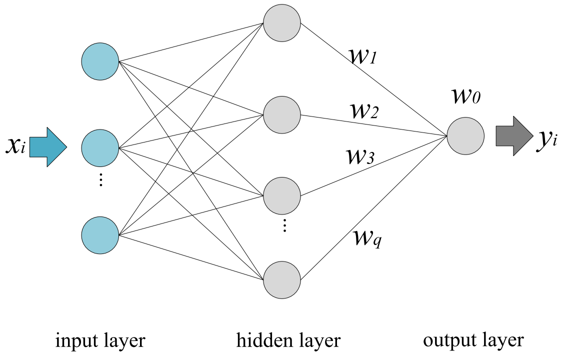

FIGURE 6. The structure of an RBFNN.

A fully connected layer, similar to MLPs, usually follows a pooling layer. In this section, we do not discuss this in further detail.

D. RADIAL BASIS FUNCTION NEURAL NETWORKS

RBFNNs are feedforward neural networks with only one hidden layer. With the maturing of research, RBFNNs have attracted much attention in various fields due to their simple structure, strong nonlinear approximation and good generalization. RBFNNs have been extensively applied in many research fields such as pattern classification, function approximation and data mining [39].

An RBFNN (Figure 6) consists of one input layer, one hidden layer and one output layer. Different from MLPs, the activation function of neurons in the hidden layer is the radial basis function (RBF), for example, the Gaussian basis function (Equation 9), where x_i denotes the ith sample in the training set and v_j and σ_j denote the center and width of the jth hidden neuron, respectively. The mechanism of the output layer in this RBFNN can be described as in Equation 8, where q denotes the number of neurons in the hidden layer and w_j represents the weight between one hidden neuron and one output neuron.

$$y_i = w_0 + \sum_{j=1}^{q} w_j \cdot g\left(x_i, v_j, \sigma_j\right) \tag{8}$$

$$g\left(x_i, v_j, \sigma_j\right) = \exp\left(-\frac{\left\|x_i - v_j\right\|^2}{2\sigma_j^2}\right) \tag{9}$$

Different from MLPs, the weights of full connections between the input neurons and hidden neurons are replaced with an RBF in RBFNNs. It is amazing that this simplification improves the ability of neural networks to map a low-dimensional input space into a higher-dimensional space. Compared with MLPs, RBFNNs not only simplify the network structure and speed up the network training process but also improve the ability to address high-nonlinear mapping problems [40].

E. RECURRENT NEURAL NETWORKS

Different from feedforward neural networks, RNNs get their name from the presence of a feedback loop. The feedback loop points from a hidden neuron to itself, thus constituting a recurrent structure. Because of the presence of this recurrent structure, RNNs can remember previous input data. Therefore, RNNs are always employed to process time sequence data [41], [42].

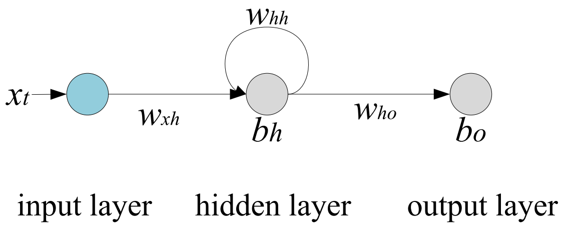

FIGURE 7. The structure of an RNN.

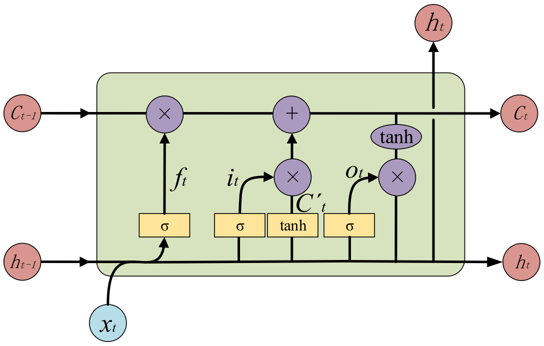

FIGURE 8. The structure of the classic LSTM.

Figure 7 illustrates a simple example of an RNN. Given a time sequence $x = \{x^0, \ldots, x^{T-1}\}$, the output of a hidden neuron at time t can be described by Equation 11, where W_{xh} denotes the connection weight between input neuron x and hidden neuron h and W_{hh} denotes the connection weight of the feedback loop. The output of this RNN at time t can be described by Equation 10, where W_{ho} denotes the connection weight between hidden neuron h and output neuron o. $H(\cdot)$ and $O(\cdot)$ are the activation functions in the hidden layer and the output layer, respectively.

$$y^t = O\left(W_{ho}h^t + b_o\right) \tag{10}$$

$$h^t = H\left(W_{xh}x^t + W_{hh}h^{t-1} + b_h\right) \tag{11}$$

Although RNNs can be used to process time sequence data, it is difficult to train RNNs due to the vanishing gradient problem [43], [44]. To address this problem, LSTMs were proposed by Hochreiter [45]. Figure 8 illustrates a typical LSTM cell. The cell consists of three different gates, namely, the forget gate f, input gate i and output gate o [45].

A forget gate determines what information should be discarded by an LSTM. The function of a forget gate can be formulated as in Equation 12, where h_{t-1} denotes the output of this cell at time $t-1$, x_t denotes the input data of this cell at time t, and W_f and b_f denote the weights and biases of the forget gate, respectively. The activation function σ maps the input data to a specific range between 0 and 1. Consequently, some part of the input value will be discarded by the forget gate.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{12}$$

An input gate determines what information should be updated to the cell state. The function of an input gate can be formulated as in Equation 13 and 14, where C'_t represents the provisional cell state calculated by the input gate, and i_t determines which parts of C'_t should be added to the final state C_t. Equation 15 describes how the input gate updates the cell state C_t.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{13}$$

$$C'_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{14}$$

$$C_t = f_t \times C_{t-1} + i_t \times C'_t \tag{15}$$

An output gate determines the output value of an LSTM cell. The function of an output gate can be formulated as in Equation 16 and 17, where o_t represents a provisional ratio that determines which parts of the cell should be exported and h_t is the final output of this cell.

$$o_t = \sigma\left(W_o\,[h_{t-1}, x_t] + b_o\right) \tag{16}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{17}$$

F. GENERATIVE ADVERSARIAL NETWORKS

Different from the above basic models, GANs [46], proposed by Goodfellow in 2014, are generative models that are pervasively used to produce content. GANs consist of two submodels, namely, the generator G and the discriminator D. Both G and D can be MLPs, CNNs, RBFNNs, RNNs, LSTMs or other neural networks. The goal of D is to train a function D(x) that has the capability to distinguish between real data and generated data. The goal of G is to train a function G(z) that transfers a simple data distribution $p_z(z)$ (e.g., a Gaussian distribution [47]) to a desired data distribution $p_{data}(x)$ that confuses D. G and D are trained alternately until Nash equilibrium is reached. Equation 18 and 19 respectively describe the loss functions of D and G.

$$L_D = -\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] - \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{18}$$

$$L_G = \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{19}$$

After 2014, the number of varieties of GANs exploded. In general, GAN series models are divided into two categories: conditional and unconditional GANs. The Wasserstein-GAN [48]–[50], DC-GAN [51], etc. are outstanding examples of unconditional GANs, which are committed to producing realistic and high-resolution image data. The cGAN [52] and info-GAN [53] are good examples of conditional GANs, which focus on how to control the output by adding some additional variants related to the target function in the input data.

G. OTHER NEURAL NETWORKS OR MECHANISMS

In this section, we introduce some neural networks or mechanisms that are not commonly used in aerodynamic data modeling. However, these state-of-the-art models or mechanisms represent the front of artificial intelligence. Besides, we believe that these new models or mechanisms can be used to analyze the spatiotemporal characteristics of the flow field data and key parts of the aerodynamic shape, so it is necessary to provide an overview of them.

1) GRAPH NEURAL NETWORKS

Graph neural networks (GNNs) were firstly proposed by Scarselli in 2009 [54]. Subsequently, many varieties of GNNs emerged, for example gated graph neural networks [55], graph attention networks [56], graph LSTM [57], and graph convolutional networks [58], etc. Different from standard neural networks, every node in a GNN maintain a state embedding that expresses information from its neighborhood. Therefore, GNNs are unique models that obtain the dependence of graphs via information passing among the nodes of graphs. undoubtedly, GNNs are widely used in learning tasks with complex dependencies, for example social network prediction [59], traffic prediction [60], recommender systems [61], graph representation [62], and image processing [55], etc.

According to the current situation, the application of GNN in aerodynamic data modeling is rare. However, we believe that the powerful ability of GNNs to analyze dependency can be used in the analysis of spatiotemporal characteristics of flow field data.

2) ATTENTION MECHANISM IN NEURAL NETWORKS

Attention mechanism was proposed by Bahdanau in 2014 [63]. Similar to human visual processing system, attention mechanism always focus on some parts of the input information, ignoring the irrelevant parts [64]. Similarly, some input data, for example languages, speeches or images, possess the characteristic that some parts are more relevant compared to others. Consequently, attention mechanism has been widely used in natural language processing [65], statistical learning [66], and computational vision [67]. There are many variations of attention mechanism, for example soft attention [68], multi-level attention [69] and multi-dimensional attention [70], etc. More details of the category of attention mechanism can be found in [71].

From the perspective of aerodynamic data modeling, we believe that attention mechanism can be used to identify key aerodynamic features (such as key dimensions in the flow state, key parts in aerodynamic shape, etc.)

III. SOLVING ORDINARY/PARTIAL DIFFERENTIAL EQUATIONS

In recent decades, the cross research of CFD and neural networks has attracted much attention [17]. It has become a common practice in the field of aerodynamics to adopt data driven models to overcome the shortcomings of CFD. Differential equations (DEs) play a significant role in CFD. DEs are divided into two categories: ODEs and PDEs. Most of the DEs in aerodynamics are complex PDEs. It is unrealistic to pursue an ultraprecise numerical solution to every PDE. Consequently, many types of approximation methods including MLPs are utilized to solve PDEs. In this section, we classify the existing studies into two categories: physics uninformed neural networks and physics informed neural networks.

A. PHYSICS UNINFORMED MODELS

Classic neural networks, as a data driven model, do not involve the physical laws contained in the governing equations when solving them. Therefore, we regard classic neural networks as physics uninformed neural networks. The studies based on physics uninformed neural networks can be divided into two categories: decomposition method and transformation method.

1) DECOMPOSITION METHOD

Early studies often use the decomposition method: dividing one ODE/PDE into different parts, some of which can be fitted by MLPs. Lagaris *et al.* [72] proposed a method to divide one ODE/PDE into two parts. The first part is a simple function that satisfies the boundary conditions. The second part is an unexplainable function that is not affected by the boundary conditions. The advantage of this approach is that the second function can be easily simulated by an MLP. The result shows that MLPs can be used to solve ODEs and PDEs, with an accuracy of 10^{-5}. In this research, the number of nodes in the hidden layers is determined by trial, which is the most common way in deep learning or neural networks.

Based on Lagaris's research, Mall *et al.* [16] proposed a method for determining the number of nodes and the initial weights of an MLP, which was used to simulate the second part of a specific PDE. The structure of an MLP with a single hidden layer was considered. The numbers of nodes in the hidden layers are related to the degree of the polynomial. For example, a degree n polynomial will generate a specific MLP with the number of nodes in the hidden layer equal to $n + 1$, and the coefficients of the polynomial will be taken as the initial weights both from the input layer to the hidden layer and from the hidden layer to the output layer. Mall considered three-, four-, and five-degree polynomials for the experiments. The results demonstrate that the proposed method is preferable over a random initialization. However, an abnormal phenomenon is observed: increasing the degree of polynomials does not lead to a high accuracy.

The above studies demonstrate that MLPs are good tools for solving ODEs/PDEs. However, in the above papers, the ODEs/PDEs are relatively simple, which means that it is still not clear weather MLPs can be applied to solve high-dimensional ODEs/PDEs.

2) TRANSFORMATION METHOD

Transforming a high-dimensional ODE/PDE into another convex optimization problem is a solution for the "curse of dimensionality" of ODEs/PDEs. Han [73] introduced an MLP-based transformation approach that can solve high-dimensional parabolic PDEs. The target PDEs, which could be approximated by MLPs, are reformulated by backward stochastic DEs from the original PDEs. The relative approximation error is stable at 10^{-3}. Based on the principle of MLPs, Sirignano [23] proposed a learning method, i.e., the deep Galerkin method (DGM), to solve high-dimensional PDEs. The test results show that the DGM is able to solve PDEs in up to 200 dimensions.

Compared with decomposition method, the transformation method verified the feasibility of using MLPs to solve the high-dimensional ODEs/PDEs. However, neither the decomposition method nor the transformation method incorporate aerodynamic laws into MLPs, which means that the above research is not of physical significance.

B. PHYSICS INFORMED MODELS

Some recent aerodynamic studies showed that neural networks can solve ODEs/PDEs with respecting the physical laws contained in them. Usually, physical laws can be fused with neural networks as prior knowledge. Based on MLPs, Raissi *et al.* [74], [75] proposed a continuous time model and a discrete time model to approximate the governing equations. These two state-of-the art physics informed neural networks followed the symmetry, invariance, principles coming from the physical laws that govern the observed data. Similar studies are [76]–[78]. These studies regard physical laws as prior knowledge and integrate them into neural networks. The resulting physics informed models meet the challenge a) that we introduced in Section I very well. However, Raissi [75] mentioned that the proposed methods which may introduce a severe bottleneck in high-dimensional ODEs/PDEs are supplementary methods for solving ODEs/PDEs because traditional ROMs (for example finite elements [79], spectral methods [80], etc.) have already matured. Therefore, these physics informed methods, as a combination between neural network and aerodynamic potential laws, remain to be further studied.

C. A CONCLUSION OF SOLVING ORDINARY/PARTIAL DIFFERENTIAL EQUATIONS

In terms of physics uninformed models, the earlier research focused on how to solve ODEs/PDEs using MLPs, leaving out the high-dimensional problem. However, recent research has been consumed with the "curse of dimensionality" of ODEs/PDEs. In terms of physics informed models, the idea of incorporating physical laws expressed by governing equations into neural networks is catching on. Table 1 illustrates the differences between the above studies. The studies on the first four lines are physics uninformed models, while those on the last lines are physics informed models. Compared with the decomposition methods, transformations are able to address high-dimensional ODEs/PDEs, but they are more complex and difficult to accomplish. Compared with physics uninformed models, physics informed models are more suitable to solving ODEs/PDEs in aerodynamics, but how to

TABLE 1. Comparison of the above studies on solving ODEs/PDEs.

researcher	method	error (MAE)	complexity	High dimension	physics informed
Lagaris [72]	D	10^{-5}	simple	No	No
Mall [16]	D	10^{-3}	simple	No	No
Han [73]	T	10^{-3}	complex	Yes	No
Sirignano [23]	T	10^{-3}	complex	Yes	No
Raissi [74]–[76]	P	10^{-4}	complex	No	Yes

In the column related to the prediction accuracy, MAE denotes the mean absolute error. In the column related to method, D denotes decomposition method, T denotes transformation method and P denotes physics informed model.

deal with high-dimensional ODEs/PDEs are still unknown. From our perspective, combining transformation methods and physics informed models may be a feasible approach to solving high-dimensional ODEs/PDEs in aerodynamics.

IV. NONLINEAR AERODYNAMIC RESPONSE PREDICTIONS

Aerodynamic response predictions refer to the utilization of appropriate methods to build aerodynamic data models that can express the variation in aerodynamic response parameters (force and moment, etc.) with the design parameters (height, Mach number and Reynolds number, etc.) [17], and finally output the predicted value of response parameters in the case of given design parameters. Because most aerodynamic problems are nonlinear, we mainly discuss nonlinear aerodynamic response predictions. In addition, since the flow state of fluid mechanics is generally divided into steady and unsteady flow [81], the corresponding aerodynamic response predictions in this paper are also divided into steady and unsteady aerodynamic response predictions.

A. STEADY NONLINEAR AERODYNAMIC RESPONSE RREDICTIONS

In the field of steady nonlinear aerodynamic response predictions, the response parameters, for example aerodynamic forces or moment coefficients, are related to flow state and aerodynamic shape. So we analyze the studies in this field from two aspects: steady nonlinear aerodynamic response predictions based on flow state and steady nonlinear aerodynamic response predictions based on both aerodynamic shapes and flow state.

1) STEADY NONLINEAR AERODYNAMIC RESPONSE PREDICTIONS BASED ON FLOW STATE

The applications of MLPs in steady aerodynamic response predictions dates back to the 1990s. Koumoutsakos *et al.* [82] established a low-order turbulence model by using MLPs. Studies in this field has exploded since 2015. Ling *et al.* [83] proposed a tensor basis neural network (TBNN) to improve the accuracy of predictions of the Reynolds average Navier–Stokes (RANS) equation. Tenney *et al.* adopted MLPs to predict the noise of a rectangular jet. It was clearly shown that the essence of the aerodynamic response predictions is multivariable nonlinear regression [84]. These studies demonstrate that the architecture of MLPs does work, but the results of these experiments show that the accuracy of the predicted data still has much room for improvement. Usually, the accuracy can be enhanced by additional hidden layers or units in each hidden layer. However, these additional layers or units will lead to a long training period for the model. In summary, one of the biggest problems in steady aerodynamic modeling is how to predict the aerodynamic responses more quickly with high accuracy [85].

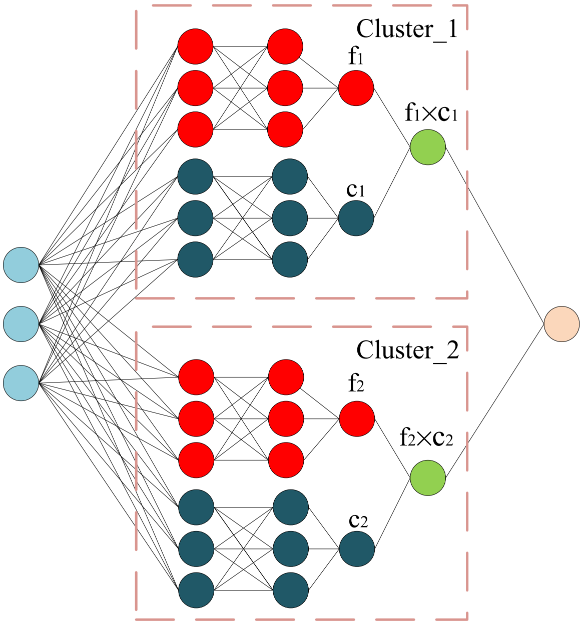

FIGURE 9. The structure of a cluster network with 2 clusters. The red network is the function network, while the dark blue network is the context network. A function network and a context network constitute a cluster. Furthermore, multiple clusters constitute a clustering network.

To alleviate the above problem of steady aerodynamic models based on MLPs, White *et al.* [85] proposed a novel network architecture (named the cluster network, Figure 9) based on MLPs to predict fluid dynamic solutions from limited samples. The cluster network is unique because it consists of a number of fully connected networks (called clusters). Each cluster automatically addresses different part of the training set. One cluster is composed of two parts: a function network and a context network. The function network learns the conditional probability $P(X|Y)$, while the context network determines how much the result of the corresponding function network should be added to the final predicted result.

White designed two experiments based on different data sets. One data set comes from Burgers' equation (Equation 20), which appears in many fields of applied mathematics, such as fluid mechanics, nonlinear acoustics, aerodynamics, and traffic flow [86]. Burgers' equation is a one-dimensional PDE used to express the movement of a shockwave across a tube in the first experiment. The other comes from the shock bubble test case of the Navier-Stokes (N-S) equation (Equation 21). The N-S equation describes the flow of many fluids, such as ocean currents, water in a pipe,

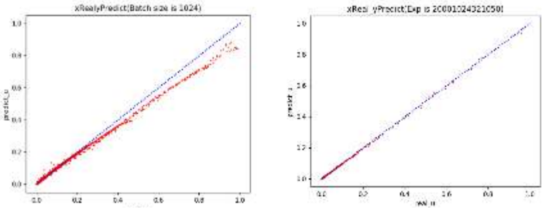

FIGURE 10. Comparison between the MLP (left) and cluster network (right) on Burgers' data (velocity *u*) prediction.

TABLE 2. Comparison of the prediction errors between the MLP and our cluster network based on Burgers' equation.

network structure	epoch	MAE	MSE
MLP	1000	10^{-3}	10^{-4}
our proposed structure [87]	1000	10^{-4}	10^{-6}

the air around wings, etc. [77]. In the second experiment, the N-S equation was used to describe how a shockwave moves across a 2D bubble. White chose only 5 numerical solutions from these two cases. In Burgers' case, the training set contains only three samples, with the viscosity μ equal to 1.0, 3.0 and 4.0. The remaining two samples, with μ equal to 2.0 and 5.0, constitute the validation set. In the N-S case, the training set contains three samples, with the velocity v equal to 1.4, 2.0 and 5.0; in a similar way, the remaining two samples, with v equal to 1.8 and 3.0, constitute the validation set. The result illustrate that both MLPs and cluster networks can closely approximate Burgers' equation and the N-S equation, but cluster networks extrapolate better than MLPs in the test sets.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2} \tag{20}$$

$$\rho\left[\frac{\partial V}{\partial t} + (V.\nabla)V\right] = -\nabla P + \rho g + \mu\nabla^2 V \tag{21}$$

We [87] noticed that White's paper did not mention how to divide the training set into different subsets for every cluster nor the conditions these subsets should satisfy. Motivated by this, in our design, the uniformly distributed Burgers' data set was split into 4 disjoint subsets according to the value of time t, with each subset processed by one cluster. We designed 14 different structures for the MLPs and 32 different structures for the cluster networks, with their parametric variation occurring in a similar way. We compared these MLPs and cluster networks based on Burgers' equation and the N-S equation. Table 2 shows the results of the most accurate MLP and cluster network based on Burgers' equation. Notably, we tried to add both the hidden layers and nodes for the most accurate MLP structure, but doing so did not improve the accuracy. Figure 10 illustrates the predicted velocity u, and visualizations of the prediction of data are shown in Figure 11. As for the N-S equation, Table 3 shows the comparison results based on the N-S equation, and visualizations of prediction of data are shown in Figure 12.

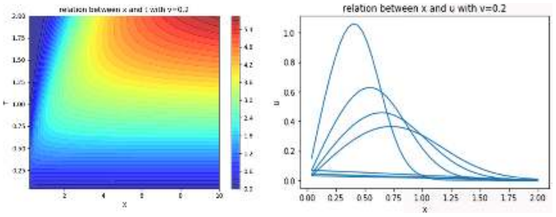

FIGURE 11. Visualization of the data predicted by our proposed structure based on Burgers' equation. The figure on the left is a prediction of the velocity *u*, and the graph on the right shows the relationship between the velocity *u* and displacement *x*.

TABLE 3. Comparison of the prediction errors between the MLP and our cluster network based on the N-S equation.

network structure	epoch	MAE	MSE
MLP	1000	10^{-2}	10^{-4}
our proposed structure [87]	1000	10^{-2}	10^{-4}

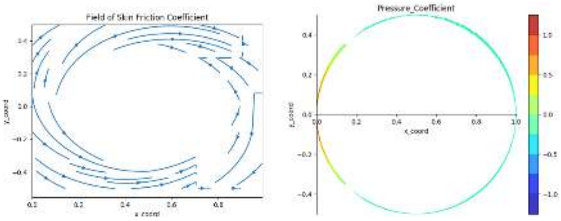

FIGURE 12. Visualization of the data predicted by our proposed structure based on cylindrical laminar case of the N-S equation. The figure on the left describes the friction field on the surface of a cylinder, and the figure on the right shows the distribution of pressure coefficients on the surface of a cylinder.

TABLE 4. Comparison of the above research on aerodynamic performance prediction.

researcher	prediction accuracy	epoch	efficiency	complexity
Koumoutsakos [82]	—	—	—	complex
Ling [83]	0.08 (MSE)	—	—	complex
Tenney [84]	0.699 (MAE)	5000	No	simple
White [85]	10^{-3} (MAE)	—	Yes	complex
We [87]	10^{-4} (MAE)	1000	Yes	complex

Notably, "—" denotes that some indicators were not mentioned in their papers. On the last two lines, the studies based on small data [90] set meet the challenge c) that we introduced in Section I verywell.

As a conclusion, the comparison results of the above studies based on MLPs are shown in Table 4. By contrast, we noticed that cluster networks are better than MLPs in terms of accuracy and efficiency, because it decomposes a complex nonlinear problem into several independent and simple nonlinear problems, which greatly reduces the difficulty of data processing for each cluster. However, cluster networks are more complex than MLPs, especially in the partition of subsets when dealing with uniformly distributed data. We tried to adopt different partition methods, e.g. k-means [88], but this did not bring an improvement in accuracy or efficiency. Therefore, we believe that cluster neural networks are more suitable for the processing of non-uniform distributed data. Of course, this speculation still needs to be proven by a lot of experimental and theoretical analysis.

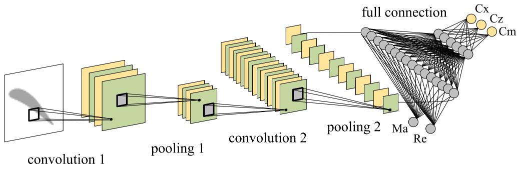

FIGURE 13. The structure of AeroCNN-I.

2) STEADY NONLINEAR AERODYNAMIC RESPONSE PREDICTIONS BASED ON BOTH AERODYNAMIC SHAPES AND FLOW STATE

The ability of CNNs to process image data has motivated the development of neural networks in the field of steady aerodynamic response predictions based on aerodynamic shapes and flow state. Compared with MLPs, CNNs take both the geometric profiles of the airfoil and the flow state into consideration. Theoretically, they are more reliable than MLPs. In this section, we classified the existing studies into three typical categories: traditional CNNs, improved CNNs and artificial images.

Yilmza *et al.* [90] adopted traditional CNNs to probe the feasibility of CNNs in the field of aerodynamic shapes modeling. In this study, the author compared the effects of MLPs and CNNs on predicting the pressure coefficient on the airfoil surface. For MLPs, they constructed a training data set, e.g., $[(x_i, y), C_p^i]_N^{tr}$, where (x_i, y) denotes a specific point on the airfoil surface and C_p^i denotes the corresponding pressure coefficient of the corresponding point. For CNNs, the input data are images of the airfoil profile. The accuracy of CNNs can exceed 80%, which demonstrates that CNNs are much better than MLPs in addressing the geometry of airfoils. An obvious drawback of this research is that the process of this study involved only the shapes of the airfoils, leaving out the flow state.

To address the problem of traditional CNNs in steady aerodynamic response predictions, improved CNNs were proposed. Considering the importance of the flow state, Zhang *et al.* [14] presented a new model named AeroCNN-I (Figure 13), which can take both the flow state and aerodynamic shapes into consideration. AeroCNN-I was used to learn the conditional probability $P(Y|[X_1, X_2])$, where Y, the output value, denotes the specific aerodynamic forces or moment coefficients, X_1 denotes the aircraft geometric shapes, and X_2 denotes the flow state (i.e., the flow Mach number and Reynolds number in this case). The prediction accuracy of AeroCNN-I is better than that of MLPs.

Subsequently, the artificial image method appeared. Thuerey *et al.* [91] adopted CNNs to predict steady aerodynamic responses. Some flow state parameters and airfoil shapes were encoded into a $N \times 128 \times 128 \times 3$ matrix, where N is the number of samples. A CNN was used to predict the velocity of a specific airfoil shapes under a given flow condition. This research provided a feasible scheme for considering the shapes of the airfoil and the flow state parameters simultaneously. Similar to Thuerey's research, Zhang *et al.* [14] further proposed AeroCNN-II based on AeroCNN-I. The core idea of AeroCNN-II is the "artificial image", in which the flow state parameters are fused. Consequently, a synthetic image can express both the flow state and the geometric shapes of the airfoils. Compared with AeroCNN-I, the prediction accuracy of AeroCNN-II is higher, which demonstrates that the "artificial image" is a working concept.

TABLE 5. Comparisons of the above research on aerodynamic performance predictions.

researcher	prediction accuracy	epoch	efficiency	complexity
Yilmza [90]	0.01 (MAE)	300	Yes	simple
AeroCNN-I [14]	0.005 (MSE)	200	Yes	simple
Thuerey [91]	0.01 (MAE)	800	No	simple
AeroCNN-II [14]	0.005 (MSE)	400	Yes	complex

Table 5 illustrates the comparisons of the mentioned models in this section. It is clear that a) CNNs can take both airfoil shapes and flow state parameters into consideration and b) the use of an "artificial image" is an effective way to integrate these two different parameters.

3) A CONCLUSION OF STEADY NONLINEAR AERODYNAMIC RESPONSE RREDICTIONS

As a matter of fact, in steady aerodynamic response predictions, there are two parallel approaches: a) approaches based on MLPs and b) approaches based on CNNs. [90] believes that the approaches based on CNNs are better than those based on MLPs. The main reasons are that a) aerodynamic forces and moments not only depend on the flow state parameters but also are affected by the aircraft profile and b) MLPs are not powerful enough to learn the geometric shapes of airfoils. In fact, the data in the training set of MLPs are measurements of one specific aircraft shape. Consequently, the training set of MLPs implies a representation of this specific geometric profile. Therefore, we believe that both MLPs and CNNs are feasible, but the key factor that affects the prediction accuracy is the high reliability of the original data in the training set. Besides, the models based on MLPs need to explore new structures to improve the prediction accuracy, we call this type of research model-oriented research. The models based on CNNs focus on how to merge flow state and aerodynamic shapes, we call this type of research data-integration-oriented research.

Despite that CNNs-based models are successful in the field of steady nonlinear aerodynamic response predictions, mechanisms of CNNs to learn aerodynamic shapes are still unknown. Understanding the mechanisms of CNNs to perceive aerodynamic information can help us to optimize the network structure or super parameters. In computational vision, there are some studies [92]–[94] based on visualization method try to explain the mechanisms for CNNs to perceive image data. However, in aerodynamic modeling, the study on the interpretability [95] of CNNs is at the initial stage. The research on the interpretability of CNNs-based aerodynamic models is not only convenient to optimize the structure and parameters of the models, but also to interpret the potential aerodynamic law. Consequently, we believe that the interpretability research will be one of the hot topics in the future.

B. UNSTEADY NONLINEAR AERODYNAMIC RESPONSE PREDICTIONS

Different from the steady aerodynamic response predictions, the unsteady aerodynamic response predictions in this section refer to the predictions of aerodynamic responses in unsteady air flows. In unsteady nonlinear aerodynamic response predictions, the relative movement between an aircraft and airflow changes with time (e.g. aerodynamic force presents a periodic or aperiodic change law), which makes unsteady aerodynamics one of the difficulties in the study of aerodynamics [96]. In addition, this high-nonlinear dynamic system often has a large number of degrees of freedom, which increasing the computational cost of CFD. Compared with the computational cost of the current CFD methods, neural networks are an efficient and feasible surrogate model for nonlinear unsteady aerodynamics [97].

In this section, we focus on two areas: spatial nonlinearity oriented unsteady aerodynamic response predictions and temporal continuity oriented unsteady aerodynamic response predictions.

1) SPATIAL NONLINEARITY ORIENTED UNSTEADY AERODYNAMIC RESPONSE PREDICTIONS

Radial basis function neural networks (RBFNNs) are considered as nonlinear input–output models that have been found to be very useful for multivariate scattered data interpolation [98]. Besides, Poggio *et al.* [40] proved that RBFNNs are the best approximator for nonlinear continuous function while MLPs are not. Therefore, In this section, we focus on the high-nonlinear characteristics of unsteady aerodynamic response predictions. We divide recent studies into two categories: pure nonlinear models and hybrid models.

The most prevailing RBFNNs-based unsteady aerodynamic models are pure nonlinear models. Hoceva *et al.* [99] adopted RBFNNs to predict the fluctuations in the passive-tracer concentration for the turbulent wake behind an airfoil. This study is just the beginning of the application of RBFNNs to turbulence modeling. The veritable explosion came after 2010 [100].

Ghoreyshi [101] summarized in detail the definition and formulation of nonlinear unsteady aerodynamic problems and applied RBFNNs to study unsteady aerodynamic modeling at low-speed flow conditions. The results show that the converged RBFNN can predict a nonlinear unsteady aerodynamic load in a few seconds, making it faster than MLPs. Zhang *et al.* [102], [103] probed the feasibility of RBFNNs for high-Reynolds-number turbulent flows. In Zhang's works, the whole flow field was divided into three parts: near-wall region, wake region, and far-field region. Three different RBFNNs were built for every region to predict the eddy viscosity. Zhang validated the accuracy and generalization capability of RBFNNs in turbulence eddy viscosity prediction. Compared with Hoceva, Zhang greatly improved the effectiveness of the RBFNNs in turbulence modeling. Kou *et al.* [47] proposed a novel model called the multikernel neural network and used it to model unsteady aerodynamics. Based on the recurrent radial basis function neural networks (RRBFNNs), Kou realized a multikernel neural network by adopting the linear combination of the Gaussian kernel function and wavelet kernel function instead of using the Gaussian kernel function alone. The experimental results show that the multikernel neural networks have fewer errors than do RRBFs in both fixed Mach number and variant Mach number experiments.

TABLE 6. Comparison of the above RBF neural network models on nonlinear unsteady aerodynamic response predictions.

researcher	uncertainty	accuracy (MAE)	relative error [103]	hybrid	multikernel
Hoceva [99]	20%~90%	—	—	No	No
Ghoreyshi [101]	—	<0.1	—	No	No
Zhang [102], [103]	—	<0.001	3%~60%	No	No
Kou [47]	—	—	2%~20%	No	Yes
Kou [97]	—	—	<2%	Yes	No

Hybrid denotes the ability of a model to address both linear and nonlinear problems.

Although research on nonlinear unsteady aerodynamics based on neural networks has made great progress, the theoretical research on unsteady aerodynamics is limited due to the inability to consider linear and nonlinear characteristics at the same time. For example, the pure nonlinear ROMs mentioned above may fail to capture the flutter boundary accurately [104]. Hybrid models in nonlinear unsteady aerodynamics began to emerge. Kou *et al.* [97] proposed a hybrid model containing both linear and nonlinear characteristics for nonlinear unsteady aerodynamics. Combining the linear autoregressive with exogenous input (ARX) model [105] and RBFNNs, the hybrid model can predict unsteady aerodynamic forces, limit cycle oscillations (LCOs) [106] and flutter behaviors.

In summary, the unsteady aerodynamic response predictions in this section focus on addressing linear or nonlinear problems. The comparison results of the above studies are shown in Table 6. By contrast, we notice that a) multikernel models are more accurate than single-kernel models and b) the hybrid model achieves better precision than do the pure nonlinear models.

2) TEMPORAL CONTINUITY ORIENTED UNSTEADY AERODYNAMIC RESPONSE PREDICTIONS

In this section, we focus on the temporal continuity of unsteady aerodynamic response predictions. We classify the existing studies into two categories: RNNs and CNNs.

The ability of RNNs to deal with time sequence data in other fields (e.g. natural language processing) promotes its development in the field of temporal continuity oriented unsteady aerodynamic response predictions. Mannarino *et al.* [107] presented a continuous time recurrent neural network (CTRNN)-based ROM for nonlinear unsteady aerodynamic load predictions. Based on traditional RNNs, Mannarino added a simple direct integration in the time domain to obtain the CTRNNs, which gave the whole model continuity in the time domain. Compared with RBFs and AeroFoam [108], the validity and accuracy of the CTRNN were verified. However, the author had to assign delay orders for the CTRNN, and the vanishing gradient problem was still unsolvable.

As one of the most popular models for processing time sequence data, the LSTM models overcome the above drawbacks of RNNs. Li *et al.* [109] adopted an LSTM to predict aerodynamic and aeroelastic responses. In their research, the input variables of this LSTM were the Mach number, pitching angle and plunging displacement of the transonic NACA64A010 airfoil, while the output variables were the corresponding lift and pitch moment coefficients. Compared with the CFD and proper orthogonal decompositions (POD)-RBF [110] approaches, it was demonstrated that LSTMs can effectively capture the dynamic characteristics of aerodynamic and aeroelastic systems.

The combination of PODs [111] and LSTMs is another common approach. Because PODs are mathematically optimal [112] for any given data set, combining PODs and LSTMs is feasible for nonlinear unsteady aerodynamic response predictions. Wang *et al.* [113] combined PODs and LSTMs to construct a novel ROM, which was applied to study the changing process when air flows past a cylinder. Compared with the RBF methods, the new novel ROM has a smaller error in the velocity prediction. Mohan *et al.* [114] adopted PODs and LSTMs/BiLSTMs [115] to learn the spatiotemporal feature of turbulent flows based on high-fidelity simulation databases coming from the N-S equation. The results show that LSTM is more accurate than BiLSTM in predicting the turbulence amplitude. Similar studies are [116], [117].

Another approach is CNN-based model. Recent studies [21], [38] illustrated that CNNs outperform RNNs (e.g. LSTMs) on problems of sequence data modeling, while demonstrating longer effective memory. This breakthrough makes CNNs be used in time sequence modeling gradually. Fukami *et al.* [118] developed a turbulent inflow generator (Figure 14) based on CNNs to generate time-dependent turbulent inflow data. Fukami combined a convolutional encoder, an MLP and a convolutional decoder to form a generative model to produce the turbulent data in next time. Han *et al.* [119] proposed a novel hybrid DNN which consists of convolutional layers, convolutional long short term memory layers, and deconvolutional layers. The new architecture was designed to capture the spatial-temporal features of unsteady flows. Similar studies are [120]–[122].

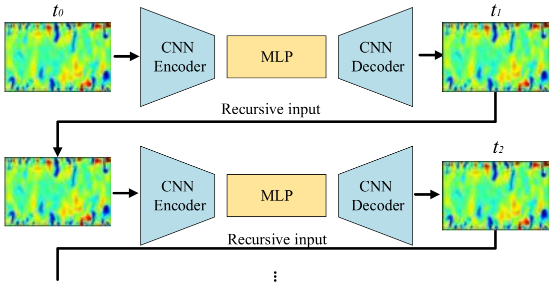

FIGURE 14. The structure of the turbulent inflow generator by Fukami [118].

TABLE 7. Comparisons of the above models on nonlinear unsteady aerodynamic response predictions.

researcher	method	prediction error	necessity of delay orders
Mannarino [107]	RNN	better than RBF*	Yes
Li [109]	LSTM	0.83%~1.90%+	No
Kou [110]	POD-RBF	2.91%~8.49%+	Yes
Wang [113]	POD&LSTM	10^{-2}*	No
Mohan [114]	POD&LSTM	large span*	No
Fukami [118]	CNN	10^{-2}*	No
Han [119]	CNN&LSTM	<0.02*	No

In the column related to the prediction error, * denotes MSE, while + denotes the relative error [103].

In summary, the comparison results of the above studies are shown in Table 7. By contrast, we notice that LSTMs are more accurate than PODs, RBFs and traditional RNNs. In addition, there is no requirement for assigning delay orders for LSTMs, which undoubtedly makes nonlinear unsteady aerodynamic modeling more convenient. Although it is not mentioned that CNNs keep longer effective memory than LSTMs in studies of Fukami and Han, CNNs are similar to LSTMs in prediction error, which makes CNNs a promising model in temporal continuity oriented unsteady aerodynamic response predictions. From our point of view, proving that CNNs have longer effective memory than LSTMs in this field is the next research point, which will determine the dominant position of CNNs.

3) A CONCLUSION OF UNSTEADY NONLINEAR AERODYNAMIC RESPONSE PREDICTIONS

There are two problems in unsteady nonlinear aerodynamic response predictions: spatial nonlinearity oriented and temporal continuity oriented modeling. As for spatial nonlinearity oriented modeling, the RBFNNs are a feasible method. An RBFNN constructs a hyperplane by linear combination of several RBFs to approximate the sample point. As for temporal continuity oriented modeling, both RNNs and CNNs can learn the temporal continuity contained in the training set. However, CNNs have shown a long effective memory in the field of natural language processing, which makes the model gradually concerned in the field of aerodynamic data modeling. RBFNNs, RNNs and CNNs mentioned in this

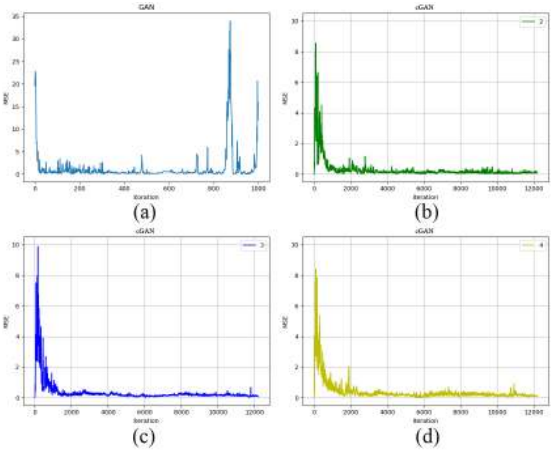

FIGURE 15. Comparison of the classic GAN with our cGAN. (a) The training process of the classic GAN. (b),(c),(d) The training processes of our cGAN in three times.

section meet the challenge b) that we introduced in Section I very well.

V. FLOW FIELD RECONSTRUCTIONS

Similar to nonlinear aerodynamic response predictions, we still analyze the studies in this field from two aspects: steady flow field reconstructions and unsteady flow field reconstructions.

A. STEADY FLOW FIELD RECONSTRUCTIONS

There are two striking indicators in steady flow field reconstructions: a) the accuracy of the generated data and b) the stability of the model. The accuracy indicator ensures the reliability of the output data, which represent the spatial distribution of the steady flow field data. The stability indicator guarantees the feasibility and practicality of the data model. In steady flow field reconstructions, a good model should have both of these indicators.

Early research adopted MLPs to reconstruct the steady flow field [15], [32], which is better than the use of PODs. Although the stability of the model was guaranteed, the accuracy of generated data was unsatisfactory.

To improve the accuracy of generated data, cGANs were gradually adopted for use in steady flow field reconstructions. Farmani [123] adopted a cGAN to study the transport phenomena. Transport phenomena refer to the exchange of energy, momentum, temperature, and so on. In Farmani's paper, cGANs were compared with numerical finite difference (FD) methods. The MAE of the cGANs was lower than 0.01, while the training time was more than an order of magnitude faster than that of the FD methods. However, the stability of cGANs was omitted.

We [87] adopted cGANs to generate steady flow field data. Figure 15 illustrates the training process of our model, which shows that, to some extent, we have improved the stability of cGANs. The MSE and MAE of our model are shown in Table 8.

TABLE 8. Comparison of MLPs and our cGANs on steady flow field reconstruction.

researcher	accuracy	stability
Yu [15]	0.01 (MSE)	Yes
Milano [32]	—	Yes
Farimani [123]	<0.01 (MAE)	No
We [87]	0.008 (MSE) 0.01 (MAE)	Yes

In summary, we infer from Table 8 that cGANs are better than MLPs in steady flow field reconstructions. These studies show that cGANs can capture the spatial distribution of aerodynamic data in a steady flow field. Unexpectedly, cGANs are not only applicable to the steady flow field reconstructions but also made great breakthroughs in the field of unsteady flow field reconstructions.

B. UNSTEADY FLOW FIELD RECONSTRUCTIONS

Compared with steady flow field reconstructions, unsteady flow field reconstructions are more difficult because of the temporal continuity of the unsteady flow. Consequently, the assessment indicators are a) the accuracy of the generated data; b) the stability of the model; c) the resolution of the reconstructed flow field image; and d) the temporal continuity of the unsteady flow field. Super-resolution images of generated data describe the spatial data distribution, namely, the spatial characteristics. Generally, super-resolution images have more powerful representations of spatial characteristics than a single numerical value. Combining the spatial characteristics and the temporal continuity of an unsteady flow, we can understand the variation in flow in both the space and time dimensions.

We classify the existing studies into two categories: CNNs and CNN(or RNN)-based GANs.

Since CNNs can be used in the field of time sequence data modeling [21], [38], CNNs can be used to generate unsteady flow field data. Fukami *et al.* [124] adopted the CNNs and the hybrid Downsampled Skip-Connection Multi-Scale (DSC/MS) models to perform super-resolution and temporal continuity analysis to reconstruct the high-resolution turbulent flow field based on only 50 training data. Liu *et al.* [125] proposed a novel multiple temporal paths convolutional neural network (MTPC). The MTPC model takes a time series of velocity fields as input. The experimental results showed that MTPC remarkably improve the spatial resolution in turbulent flow field reconstructions. Significantly, MTPC performed well in temporal continuity as well as other features (for example kinetic energy spectra). The satisfy indicators of Fukami and Liu's work are a), c) and d). They did not consider the stability of the model. However, from the accuracy of generated data, we know that the stability of these models are not satisfactory.

CNNs can also be combined with GANs to capture the spatio-temporal characteristics of unsteady flow data. Xie *et al.* [24] proposed a novel generative model named tempoGAN to generate super-resolution flow field images.

TABLE 9. Comparison of CNNs and GANs on steady flow field reconstruction.

researcher	accuracy	stability	super-resolution image	temporal continuity
Fukami [124]	10^{-1}~10^{-3}	—	Yes	Yes
Liu [125]	10^{-3}~10^{-4}	—	Yes	Yes
Xie [24]	—	No	Yes	Yes
Kim [126]	—	No	Yes	Yes
Lee [127]	10^{-2}~10^{-3}	No	Yes	Yes

TempoGAN is a conditional GAN with double discriminators. One discriminator (CNN-based discriminator) is used to determine the spatial distribution of the generated data, while the other (CNN-based discriminator) is used to determine the temporal distribution. The flow velocity v, vorticity w and density ρ are the three-dimensional input data of the two discriminators, while the input data of the generator are z (stochastic noise) and ρ, which means that tempoGAN generates v and w when the density equals ρ. The satisfy indicators of tempoGAN are a), c) and d). Kim *et al.* [126] developed a generative model named RNN-GAN to generate an inlet boundary condition of turbulent channel flow. RNN-GAN can generate time-varying flow for a long time. RNN-GAN satisfies indicator a) and d). Lee *et al.* [127] developed a generative model based on CNNs and GANs to predict unsteady flow over a circular cylinder. This model is composed of multiple CNNs that generate flow field data at the next time according to the data at the current moment. The studies of Lee satisfies indicator a) and d).

In summary, it is clear that the way to satisfy the four indicators will be a vital research topic. However, it is not easy to satisfy all these indicators simultaneously. Hence, how to measure the importance of these indicators is something that researchers need to consider.

C. A CONCLUSION OF FLOW FIELD RECONSTRUCTIONS

In flow field reconstructions, MLPs, CNNs and GANs are widely used to generate steady/unsteady flow field data. Among these models, we suppose that CNN-based GANs are the most promising models to meet the requirements of accuracy, super-resolution images and temporal continuity at the same time. However, the stability of GAN itself limits the application and promotion of CNN-based GANs. Therefore, how to address the stability problem of them will become a research hotspot in this field.

VI. OTHER SCATTERED STUDIES

In this section, we introduce some scattered research areas or methods. Research areas are aerodynamic inverse design and dimension reduction.

A. AERODYNAMIC INVERSE DESIGNS

Inverse designs [128] play a vital role in aircraft designs. The inverse design approaches are clearly divided into gradient-free and gradient-based algorithms [129]. Gradient-free algorithms (traditional approaches), for example particle swarm optimization [130], are not based on neural networks, therefore, we focus on the gradient-based algorithms in this section. Gradient-based algorithms calculate the continuous gradient of the objective function.

Early gradient-based inverse design algorithms were based on MLPs [2], [131], [132], which has proven to be a poor approach. The problems are as follows: a) these works, to some extent, are less accurate; and b) these works still require the parametrization of the geometric shapes of wings. In contrast, CNNs can automatically extract high-dimensional features, which are more suitable for the processing of aircraft shapes. Consequently, to further improve the accuracy of inverse design and reduce the amount of manual intervention, MLPs were replaced with CNNs in the inverse design of airfoils. Sekar *et al.* [128] applied deep CNNs to obtain the airfoil shape based on the distribution of the pressure coefficient. The advantages of CNNs in aerodynamic inverse are apparent: CNNs are more accurate than MLPs, and there is no need to parameterize the airfoils.

B. FEATURE EXTRACTION

Aerodynamic data (flow state or aerodynamic shapes) usually possess multi-dimensional features (for example, after an airfoil is parameterized, thousands of dimensional features can be obtained). This problem increases the difficulty of learning potential aerodynamic laws. Neural networks are natural dimension reduction models to extract key features. Sekar *et al.* [121] use a deep convolutional neural network and a deep multilayer perceptron to predict incompressible laminar steady flow field over airfoils. In Sekar's work, the deep CNN was only used to reduce the dimension of input data. In the work of Wang *et al.* [133] and Omata *et al.* [134], a deep convolutional autoencoder was used for dimensionality reduction in unsteady flow fields. Murata *et al.* [135] proposed the mode decomposing convolutional neural network autoencoder (MD-CNN-AE) to visualize the decomposed flow fields. The results suggest a great potential for the nonlinear MD-CNN-AE to be used for feature extraction of flow fields in lower dimension. Chen *et al.* [136] applied CNN-based GANs to reduce the dimension of input airfoil shapes to realize the design optimization of an airfoil. CNNs are better than traditional approaches (e.g. PODs [111]), especially in the aerodynamic shape processing.

C. SOLVING ORDINARY/PARTIAL DIFFERENTIAL EQUATIONS BY RADIAL BASIS FUNCTION

Because RBFs are the basis of RBFNNs, we briefly introduce the applications of RBFs in this section and the next section.

RBFs are usually used to solve ODEs/PDEs in aerodynamics [137]–[139]. Compared with the traditional methods [140]–[142] of solving PDEs, their real meshless characteristic is the biggest advantage, which has promoted the application of RBFs in this field [143]–[145]. Different from the solving of ODEs/PDEs presented in Section III, RBFs do not involve artificial neural networks. RBFs are methods based on function approximation by reducing the error between the approximation functions and ODEs/PDEs to obtain the approximate solutions. However, considering that many ODEs/PDEs in aerodynamics possess highly nonlinear properties, MLPs are still the most promising approaches in this field.

D. TURBULENCE MODELING BASED ON RADIAL BASIS FUNCTION

Another application of RBFs in aerodynamics is turbulence modeling. Duraisamy *et al.* [146], [147] and Singh *et al.* [148] used an RBF (a Gaussian kernel function) to build turbulence and transition models and embedded them into CFD solvers. Similar studies are [149]–[151]. Because RBFs (especially Gaussian kernel functions) are quite simple, these models reduce the time consumption of turbulence simulation. Compared with RBFs, RBFNNs play an important role in aerodynamic data modeling, especially in nonlinear unsteady aerodynamic response predictions.

VII. CONCLUSION AND DISCUSSION

In this review, we analyzed the shortcomings of CFD and traditional ROMs in aerodynamic data modeling, and we introduced existing neural network based approaches in solving ODEs/PDEs, steady/unsteady aerodynamic response predictions and steady/unsteady flow field reconstructions, etc. We emphasized some successful studies, such as the DGM by Sirignano, AeroCNN-II by Zhang, and tempoGAN by Xie *et al.* These successful studies indicate the future prospects in their respective fields.

In the field of solving ODEs/PDEs, we introduced physics uninformed models and physics informed models, and further pointed out the problem of sovling high-dimensional ODEs/PDEs. As for physics uninformed models, the decomposition of ODEs/PDEs is easy, but it is not powerful enough to deal with high-dimensional ODEs/PDEs, while the transformation of high-dimensional ODEs/PDEs is suitable to address the ‘‘curse of dimensionality’’ of ODEs/PDEs. As for physics informed models, physical laws can be fused with neural networks as prior knowledge. However, physics informed models also cannot address the ‘‘curse of dimensionality’’ problems. Hence, we believe that the combination of transformation methods and physics informed models will be a feasible and aerodynamics informed approach to solve high-dimensional ODEs/PDEs.

In aerodynamic response predictions, we analyzed them from two perspectives: steady and unsteady aerodynamic response predictions.

In steady aerodynamic response predictions, both MLPs and CNNs are feasible. The difference is the representation of the geometric profiles of airfoils. In the approaches based on flow state, more complex model structures and training algorithms are the trend of future research, namely, model-oriented research. In the approaches based on aerodynamic shapes and flow state, those methods that can integrate both flow state parameters and the geometric profile will direct future studies, namely, data-integration-oriented research. In addition, the interpretability of neural networks in the field of aerodynamics will also be a hot topic in the future.

In unsteady aerodynamic response predictions, RBFNNs, RNNs and CNNs are commonly used for different targets. RBFNNs focus on the spatial nonlinearity of unsteady aerodynamic modeling. Hybrid models and multikernel models will be topics of future nonlinear research. RNNs and CNNs focus on the temporal continuity of unsteady aerodynamic modeling. It should be noted that CNNs can process time sequence data in the way of a recursive input, and show longer effective memory than LSTM. However, this feature of CNNs, at present, is still unexplainable and needs further study.

In the field of flow field reconstruction, the steady and unsteady flow field reconstructions were analyzed. In steady flow field reconstructions, the prediction accuracy and the stability of models are key indicators. Compared with steady flow field reconstructions, there are two new indicators for unsteady flow field reconstructions, namely, the resolution of the generated images and the temporal continuity, which reflect the spatial distribution of the flow data and the rule governing the change in flow data with time, respectively. By combing existing studies, we can conclude that CNN-based cGANs are most likely to meet all the indicators in future studies.

We believe that neural networks will certainly solve many difficult problems in aerodynamics. In the next few decades, aerodynamic response predictions based on neural networks will gradually mature. In the long run, the solving of ODEs/PDEs and the flow field reconstructions are sure to be the most considered topics in both artificial intelligence and aerodynamics research.

REFERENCES

[1] H.-Y. Fan, G. S. Dulikravich, and Z.-X. Han, “Aerodynamic data modeling using support vector machines,” *Inverse Problems Sci. Eng.*, vol. 13, no. 3, pp. 261–278, Jun. 2005.

[2] M. M. Rai and N. K. Madavan, “Aerodynamic design using neural networks,” *AIAA J.*, vol. 38, pp. 173–182, Jan. 2000.

[3] D. J. Lucia, P. S. Beran, and W. A. Silva, “Reduced-order modeling: New approaches for computational physics,” *Prog. Aerosp. Sci.*, vol. 40, nos. 1–2, pp. 51–117, Feb. 2004.

[4] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu, “Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison,” *Comput. Methods Appl. Mech. Eng.*, vols. 237–240, pp. 10–26, Sep. 2012.

[5] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–26.

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, 5967–5976.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[8] R. Collobert and J. Weston, “A unified architecture for natural language processing,” in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 160–167.

[9] A. Kumar *et al.*, “Ask me anything: Dynamic memory networks for natural language processing,” in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1378–1387.

[10] E. de la Rosa and W. Yu, "Randomized algorithms for nonlinear system identification with deep learning modification," *Inf. Sci.*, vols. 364–365, pp. 197–212, Oct. 2016.

[11] J. Qiao, G. Wang, W. Li, and X. Li, "A deep belief network with PLSR for nonlinear system modeling," *Neural Netw.*, vol. 104, pp. 68–79, Aug. 2018.

[12] S. Genc, "Parametric system identification using deep convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2112–2119.

[13] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Phys.*, vol. 13, no. 5, pp. 431–434, May 2017.

[14] Y. Zhang, W.-J. Sung, and D. Mavris, "Application of convolutional neural network to predict airfoil lift coefficient," 2018, *arXiv:1712.10082*. [Online]. Available: https://arxiv.org/abs/1712.10082

[15] J. Yu and J. S. Hesthaven, "Flowfield reconstruction method using artificial neural network," *AIAA J.*, vol. 57, no. 2, pp. 482–498, Feb. 2019.

[16] S. Mall and S. Chakraverty, "Comparison of artificial neural network architecture in solving ordinary differential equations," *Adv. Artif. Neural Syst.*, vol. 2013, pp. 1–12, 2013.

[17] S. Brunton, B. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," 2020, *arXiv:1905.11075*. [Online]. Available: https://arxiv.org/abs/1905.11075

[18] L. Baert, E. Chérière, C. Sainvitu, I. Lepot, A. Nouvellon, and V. Leonardon, "Aerodynamic optimization of the low-pressure turbine module: Exploiting surrogate models in a high-dimensional design space," *J. Turbomachinery*, vol. 142, no. 3, Mar. 2020, Art. no. 031005.

[19] A. C. Huang, H. A. Carson, S. R. Allmaras, M. C. Galbraith, D. L. Darmofal, and D. S. Kamenetskiy, "Correction: A variational multiscale method with discontinuous subscales for output-based adaptation of aerodynamic flows," in *Proc. AIAA Scitech Forum*, Jan. 2020, p. 1563.

[20] E. Kantor, D. E. Raveh, and R. Cavallaro, "Nonlinear structural, nonlinear aerodynamic model for static aeroelastic problems," *AIAA J.*, vol. 57, no. 5, pp. 2158–2170, May 2019.

[21] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*. [Online]. Available: http://arxiv.org/abs/1803.01271

[22] Y. Lakretz, G. Kruszewski, T. Desbordes, D. Hupkes, S. Dehaene, and M. Baroni, "The emergence of number and syntax units in LSTM language models," 2019, *arXiv:1903.07435*. [Online]. Available: http://arxiv.org/abs/1903.07435

[23] J. Sirignano and K. Spiliopoulos, "DGM: A deep learning algorithm for solving partial differential equations," *J. Comput. Phys.*, vol. 375, pp. 1339–1364, Dec. 2018.

[24] Y. Xie, E. Franz, M. Chu, and N. Thuerey, "TempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–15, Aug. 2018.

[25] A. N. Kolmogorov, "The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers," *Proc. Math. Phys. Sci.*, vol. 434, pp. 9–13, 1991. [Online]. Available: http://www.jstor.org/stable/51980

[26] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biol.*, vol. 52, nos. 1–2, pp. 99–115, Jan. 1990.

[27] F. Rosenblatt, "The perceptron, a perceiving and recognizing automaton project para," Cornell Aeronaut. Lab., Buffalo, NY, USA, Tech. Rep. 85-460-1, 1957.

[28] M. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry, Expanded Edition*. Cambridge, MA, USA: MIT Press, 1988, p. 28.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[30] C. L. Teo, K. B. Lim, G. S. Hong, and M. H. T. Yeo, "A neural net approach in analyzing photograph in PIV," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 1991, pp. 1535–1538.

[31] I. Grant and X. Pan, "An investigation of the performance of multi layer, neural networks applied to the analysis of PIV images," *Experim. Fluids*, vol. 19, no. 3, pp. 159–166, Jul. 1995.

[32] M. Milano and P. Koumoutsakos, "Neural network modeling for near wall turbulent flow," *J. Comput. Phys.*, vol. 182, no. 1, pp. 1–26, Oct. 2002.

[33] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[34] R. A. Lordo, "Learning from data: Concepts, theory, and methods," *Technometrics*, vol. 43, no. 1, pp. 105–106, Feb. 2001.

[35] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, Jan. 1989.

[36] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Proc. Int. Joint Conf. Neural Netw.*, Aug. 1989, pp. 593–605.

[37] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid, "Convolutional kernel networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2627–2635.

[38] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 649–657.

[39] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, no. 2, pp. 281–294, Jun. 1989.

[40] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, Sep. 1990.

[41] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.

[42] J.-S. Zhang and X.-C. Xiao, "Predicting chaotic time series using recurrent neural network," *Chin. Phys. Lett.*, vol. 17, no. 2, pp. 88–90, Feb. 2000.

[43] A. Graves, "Supervised sequence labelling with recurrent neural networks," Ph. D. dissertation, Tech. Univ. Munich, Munich, Germany, 2008.

[44] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," in *A Field Guideto Dynamical Recurrent Neural Networks*, S. C. Kremer and J. F. Kolen, Eds. Piscataway, NJ, USA: IEEE Press, 2001.

[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[46] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2014, pp. 2672–2680.

[47] J. Kou and W. Zhang, "Multi-kernel neural networks for nonlinear unsteady aerodynamic reduced-order modeling," *Aerosp. Sci. Technol.*, vol. 67, pp. 309–326, Aug. 2017.

[48] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875

[49] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–17.

[50] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 5768–5778.

[51] J. Li, J. Jia, and D. Xu, "Unsupervised representation learning of image-based plant disease with deep convolutional generative adversarial networks," in *Proc. 37th Chin. Control Conf. (CCC)*, Jul. 2018, pp. 1–16.

[52] A. Nguyen *et al.*, "Plug & play generative networks: Conditional iterative generation of images in latent space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4467–4477.

[53] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2180–2188.

[54] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[55] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*. [Online]. Available: http://arxiv.org/abs/1511.05493

[56] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*. [Online]. Available: http://arxiv.org/abs/1710.10903

[57] N. Peng, H. Poon, C. Quirk, K. Toutanova, and W.-T. Yih, "Cross-sentence N-ary relation extraction with graph LSTMs," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 101–115, Dec. 2017.

[58] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[59] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[60] A. Rahimi, T. Cohn, and T. Baldwin, ''Semi-supervised user geolocation via graph convolutional networks,'' 2018, *arXiv:1804.08049*. [Online]. Available: http://arxiv.org/abs/1804.08049

[61] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, ''Graph convolutional neural networks for Web-scale recommender systems,'' in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.

[62] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, ''Hierarchical graph representation learning with differentiable pooling,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4800–4810.

[63] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural machine translation by jointly learning to align and translate,'' 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[64] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, ''Show, attend and tell: Neural image caption generation with visual attention,'' in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[65] A. Galassi, M. Lippi, and P. Torroni, ''Attention, please! A critical review of neural attention models in natural language processing,'' 2019, *arXiv:1902.02181*. [Online]. Available: http://arxiv.org/abs/1902.02181

[66] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, ''Learning phrase representations using RNN encoder-decoder for statistical machine translation,'' 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[67] F. Wang and D. M. J. Tax, ''Survey on the attention based RNN model and its applications in computer vision,'' 2016, *arXiv:1601.06823*. [Online]. Available: http://arxiv.org/abs/1601.06823

[68] D. Jain, A. Kumar, and G. Garg, ''Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN,'' *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106198.

[69] Y. Song, J. Wang, L. Ma, Z. Yu, and J. Yu, ''Weakly-supervised multi-level attentional reconstruction network for grounding textual queries in videos,'' 2020, *arXiv:2003.07048*. [Online]. Available: http://arxiv.org/abs/2003.07048

[70] Y. Yao, S. Zhang, S. Yang, and G. Gui, ''Learning attention representation with a multi-scale CNN for gear fault diagnosis under different working conditions,'' *Sensors*, vol. 20, no. 4, p. 1233, 2020.

[71] S. Chaudhari, G. Polatkan, R. Ramanath, and V. Mithal, ''An attentive survey of attention models,'' 2019, *arXiv:1904.02874*. [Online]. Available: http://arxiv.org/abs/1904.02874

[72] I. E. Lagaris, A. Likas, and D. I. Fotiadis, ''Artificial neural networks for solving ordinary and partial differential equations,'' *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 987–1000, Sep. 1998.

[73] J. Han, A. Jentzen, and E. Weinan, ''Solving high-dimensional partial differential equations using deep learning,'' *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 34, pp. 8505–8510, Aug. 2018.

[74] M. Raissi, P. Perdikaris, and G. E. Karniadakis, ''Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,'' *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.

[75] M. Raissi, P. Perdikaris, and G. Em Karniadakis, ''Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations,'' 2017, *arXiv:1711.10561*. [Online]. Available: http://arxiv.org/abs/1711.10561

[76] M. Raissi, A. Yazdani, and G. Em Karniadakis, ''Hidden fluid mechanics: A Navier–Stokes informed deep learning framework for assimilating flow visualization data,'' 2018, *arXiv:1808.04327*. [Online]. Available: http://arxiv.org/abs/1808.04327

[77] M. Raissi and G. E. Karniadakis, ''Hidden physics models: Machine learning of nonlinear partial differential equations,'' *J. Comput. Phys.*, vol. 357, pp. 125–141, Mar. 2018.

[78] M. Raissi, ''Deep hidden physics models: Deep learning of nonlinear partial differential equations,'' *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 9320–955, 2018.

[79] J. Urban and J. Preinhaelter, ''Adaptive finite elements for a set of second-order ODEs,'' *J. Plasma Phys.*, vol. 72, no. 6, p. 1041, Dec. 2006.

[80] L. N. Trefethen. (1996). *Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations*. [Online]. Available: http://web.comlab.ox.ac.uk/oucl/work/nick.trefethen/pdetext.html

[81] Y. Liu, L. Sankar, R. Englar, and K. Ahuja, ''Numerical simulations of the steady and unsteady aerodynamic characteristics of a circulation control wing airfoil,'' in *Proc. 39th Aerosp. Sci. Meeting Exhib.*, Jan. 2001, p. 704.

[82] S. Müller, M. Milano, and P. Koumoutsakos, ''Application of machine learning algorithms to flow modeling and optimization,'' *Annu. Res. Briefs*, pp. 169–178, 1999.

[83] J. Ling, A. Kurzawski, and J. Templeton, ''Reynolds averaged turbulence modelling using deep neural networks with embedded invariance,'' *J. Fluid Mech.*, vol. 807, pp. 155–166, Nov. 2016.

[84] A. S. Tenney, M. N. Glauser, and J. Lewalle, ''A deep learning approach to jet noise prediction,'' in *Proc. AIAA Aerosp. Sci. Meeting*, Jan. 2018, pp. 1–9.

[85] C. White, D. Ushizima, and C. Farhat, ''Fast neural network predictions from constrained aerodynamics datasets,'' 2019, *arXiv:1902.00091*. [Online]. Available: https://arxiv.org/abs/1902.00091

[86] C. Basdevant, M. Deville, P. Haldenwang, J. M. Lacroix, J. Ouazzani, R. Peyret, P. Orlandi, and A. T. Patera, ''Spectral and finite difference solutions of the burgers equation,'' *Comput. Fluids*, vol. 14, no. 1, pp. 23–41, Jan. 1986.

[87] L. Hu, J. Zhang, and Y. Xiang, ''Research on deep learning in aerodynamic modeling,'' UESTC, Chengdu, China, Tech. Rep. 93-220-1, 2019.

[88] A. K. Jain, ''Data clustering: 50 years beyond K-means,'' *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[89] G.-J. Qi and J. Luo, ''Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods,'' 2019, *arXiv:1903.11260*. [Online]. Available: http://arxiv.org/abs/1903.11260

[90] E. Yilmaz and B. German, ''A convolutional neural network approach to training predictors for airfoil performance,'' in *Proc. 18th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, Jun. 2017, p. 3660.

[91] N. Thuerey, K. Weißenow, L. Prantl, and X. Hu, ''Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows,'' *AIAA J.*, vol. 58, no. 1, pp. 25–36, Jan. 2020.

[92] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, ''Learning deep features for discriminative localization,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[93] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, ''Grad-CAM: Visual explanations from deep networks via gradient-based localization,'' in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 618–626.

[94] Q. Zhang, Y. N. Wu, and S.-C. Zhu, ''Interpretable convolutional neural networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8827–8836.

[95] S. Lee and D. You, ''Mechanisms of a convolutional neural network for learning three-dimensional unsteady wake flow,'' 2019, *arXiv:1909.06042*. [Online]. Available: http://arxiv.org/abs/1909.06042

[96] D. E. Raveh, ''Reduced-order models for nonlinear unsteady aerodynamics,'' *AIAA J.*, vol. 39, pp. 1417–1429, Jan. 2001.

[97] J. Kou and W. Zhang, ''Layered reduced-order models for nonlinear aerodynamics and aeroelasticity,'' *J. Fluids Struct.*, vol. 68, pp. 174–193, Jan. 2017.

[98] M. D. Buhmann, ''Radial basis functions,'' *Acta Numer.*, vol. 9, pp. 1–38, Jan. 2000.

[99] M. Hočevar, B. Širok, and I. Grabec, ''A turbulent-wake estimation using radial basis function neural networks,'' *Flow, Turbulence Combustion*, vol. 74, no. 3, pp. 291–308, Apr. 2005.

[100] K. Duraisamy, P. R. Spalart, and C. L. Rumsey, ''Status, emerging ideas and future directions of turbulence modeling research in aeronautics,'' Tech. Rep. NASA/TM-2017-219682, 2017.

[101] M. Ghoreyshi, A. Jirásek, and R. M. Cummings, ''Computational approximation of nonlinear unsteady aerodynamics using an aerodynamic model hierarchy,'' *Aerosp. Sci. Technol.*, vol. 28, no. 1, pp. 133–144, Jul. 2013.

[102] W. Zhang, L. Zhu, Y. Liu, and J. Kou, ''Machine learning methods for turbulence modeling in subsonic flows over airfoils,'' 2018, *arXiv:1806.05904*. [Online]. Available: http://arxiv.org/abs/1806.05904

[103] W. Zhang, B. Wang, and Z. Ye, ''High efficient numerical method for limit cycle flutter analysis based on nonlinear aerodynamic reduced order model reduced order model,'' in *Proc. 51st AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn., Mater. Conf., 18th AIAA/ASME/AHS Adapt. Struct. Conf. 12th*, Apr. 2010, p. 2723.

[104] W. Zhang, B. Wang, Z. Ye, and J. Quan, ''Efficient method for limit cycle flutter analysis based on nonlinear aerodynamic reduced-order models,'' *AIAA J.*, vol. 50, no. 5, pp. 1019–1028, May 2012.

[105] C. S. Huang, S. L. Hung, W. C. Su, and C. L. Wu, ''Identification of time-variant modal parameters using time-varying autoregressive with exogenous input and low-order polynomial function,'' *Comput.-Aided Civil Infrastruct. Eng.*, vol. 24, no. 7, pp. 470–491, Oct. 2009.

[106] J. P. Thomas, E. H. Dowell, and K. C. Hall, "Nonlinear inviscid aerodynamic effects on transonic divergence, flutter, and limit-cycle oscillations," *AIAA J.*, vol. 40, pp. 638–646, Jan. 2002.

[107] A. Mannarino and P. Mantegazza, "Nonlinear aeroelastic reduced order modeling by recurrent neural networks," *J. Fluids Struct.*, vol. 48, pp. 103–121, Jul. 2014.

[108] G. Romanelli and L. Mangani, "Object-oriented redesign of density-based rans solver aerofoam for aerodynamic/aeroelastic industrial applications," in *Proc. 5-th OpenCFD Conf.*, 2010.

[109] K. Li, J. Kou, and W. Zhang, "Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple mach numbers," *Nonlinear Dyn.*, vol. 96, no. 3, pp. 2157–2177, May 2019.

[110] J. Kou and W. Zhang, "Reduced-order modeling for nonlinear aeroelasticity with varying mach numbers," *J. Aerosp. Eng.*, vol. 31, no. 6, Nov. 2018, Art. no. 04018105.

[111] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annu. Rev. Fluid Mech.*, vol. 25, no. 1, pp. 539–575, Jan. 1993.

[112] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[113] Z. Wang, D. Xiao, F. Fang, R. Govindan, C. C. Pain, and Y. Guo, "Model identification of reduced order fluid dynamics systems using deep learning," *Int. J. Numer. Methods Fluids*, vol. 86, no. 4, pp. 255–268, Feb. 2018.

[114] A. T. Mohan and D. V. Gaitonde, "A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks," 2018, *arXiv:1804.09269*. [Online]. Available: http://arxiv.org/abs/1804.09269

[115] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[116] S. Pawar, S. M. Rahman, H. Vaddireddy, O. San, A. Rasheed, and P. Vedula, "A deep learning enabler for nonintrusive reduced order modeling of fluid flows," *Phys. Fluids*, vol. 31, no. 8, Aug. 2019, Art. no. 085101.

[117] Z. Deng, Y. Chen, Y. Liu, and K. C. Kim, "Time-resolved turbulent velocity field reconstruction using a long short-term memory (LSTM)-based artificial intelligence framework," *Phys. Fluids*, vol. 31, no. 7, 2019, Art. no. 075108.

[118] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata, "Synthetic turbulent inflow generator using machine learning," *Phys. Rev. Fluids*, vol. 4, no. 6, Jun. 2019, Art. no. 064603.

[119] R. Han, Y. Wang, Y. Zhang, and G. Chen, "A novel spatial-temporal prediction method for unsteady wake flows based on hybrid deep neural network," *Phys. Fluids*, vol. 31, no. 12, 2019, Art. no. 127101.

[120] X. Jin, P. Cheng, W.-L. Chen, and H. Li, "Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder," *Phys. Fluids*, vol. 30, no. 4, Apr. 2018, Art. no. 047105.

[121] V. Sekar and B. C. Khoo, "Fast flow field prediction over airfoils using deep learning approach," *Phys. Fluids*, vol. 31, no. 5, May 2019, Art. no. 057103.

[122] J. Kim and C. Lee, "Prediction of turbulent heat transfer using convolutional neural networks," *J. Fluid Mech.*, vol. 882, p. A18, Jan. 2020.

[123] A. B. Farimani, J. Gomes, and V. S. Pande, "Deep learning the physics of transport phenomena," 2017, *arXiv:1709.02432*. [Online]. Available: https://arxiv.org/abs/1709.02432

[124] K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," *J. Fluid Mech.*, vol. 870, pp. 106–120, Jul. 2019.

[125] B. Liu, J. Tang, H. Huang, and X.-Y. Lu, "Deep learning methods for super-resolution reconstruction of turbulent flows," *Phys. Fluids*, vol. 32, no. 2, Feb. 2020, Art. no. 025105.

[126] J. Kim and C. Lee, "Deep unsupervised learning of turbulence for inflow generation at various Reynolds numbers," *J. Comput. Phys.*, vol. 406, Apr. 2020, Art. no. 109216.

[127] S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," *J. Fluid Mech.*, vol. 879, pp. 217–254, Nov. 2019.

[128] V. Sekar, M. Zhang, C. Shu, and B. C. Khoo, "Inverse design of airfoil using a deep convolutional neural network," *AIAA J.*, vol. 57, no. 3, pp. 993–1003, Mar. 2019.

[129] S. N. Skinner and H. Zare-Behtash, "State-of-the-art in aerodynamic shape optimisation methods," *Appl. Soft Comput.*, vol. 62, pp. 933–962, Jan. 2018.

[130] G. Venter and J. Sobieszczanski-Sobieski, "Particle swarm optimization," *AIAA J.*, vol. 41, no. 8, pp. 1583–1589, 2003.

[131] A. Kharal and A. Saleem, "Neural networks based airfoil generation for a given using Bezier–PARSEC parameterization," *Aerosp. Sci. Technol.*, vol. 23, no. 1, pp. 330–344, Dec. 2012.

[132] G. Sun, Y. Sun, and S. Wang, "Artificial neural network based inverse design: Airfoils and wings," *Aerosp. Sci. Technol.*, vol. 42, pp. 415–428, Apr. 2015.

[133] M. Wang, H.-X. Li, X. Chen, and Y. Chen, "Deep learning-based model reduction for distributed parameter systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 12, pp. 1664–1674, Dec. 2016.

[134] N. Omata and S. Shirayama, "A novel method of low-dimensional representation for temporal behavior of flow fields using deep autoencoder," *AIP Adv.*, vol. 9, no. 1, Jan. 2019, Art. no. 015006.

[135] T. Murata, K. Fukami, and K. Fukagata, "Nonlinear mode decomposition with convolutional neural networks for fluid dynamics," *J. Fluid Mech.*, vol. 882, p. A13, Jan. 2020.

[136] W. Chen, K. Chiu, and M. Fuge, "Aerodynamic design optimization and shape exploration using generative adversarial networks," in *Proc. AIAA Scitech Forum*, Jan. 2019, p. 2351.

[137] C. Audouze, F. De Vuyst, and P. B. Nair, "Reduced-order modeling of parameterized PDEs using time-space-parameter principal component analysis," *Int. J. Numer. Methods Eng.*, vol. 80, no. 8, pp. 1025–1057, Nov. 2009.

[138] C. Audouze, F. De Vuyst, and P. B. Nair, "Nonintrusive reduced-order modeling of parametrized time-dependent partial differential equations," *Numer. Methods Partial Differ. Equ.*, vol. 29, no. 5, pp. 1587–1628, Sep. 2013.

[139] W. Zongmin, "Hermite-Birkhoff interpolation of scattered data by radial basis functions," *Approximation Theory Appl.*, vol. 8, no. 2, pp. 1–10, 1992.

[140] C. A. Duarte and J. T. Oden, "A new meshless method to solve boundary-value problems," in *Proc. 16th CILAMCE-Iberian Latin Amer. Conf. Comput. Methods Eng.*, Curitiba, Brazil, 1995, pp. 90–99.

[141] E. Oñate, S. Idelsohn, O. C. Zienkiewicz, and R. L. Taylor, "A finite point method in computational mechanics. applications to convective transport and fluid flow," *Int. J. Numer. Methods Eng.*, vol. 39, no. 22, pp. 3839–3866, Nov. 1996.

[142] S. N. Atluri and T. Zhu, "A new meshless local petrov-Galerkin (MLPG) approach in computational mechanics," *Comput. Mech.*, vol. 22, no. 2, pp. 117–127, Aug. 1998.

[143] C. S. Chen and C. A. Brebbia, "The dual reciprocity method for helmholtz-type operators," *WIT Trans. Model. Simul.*, vol. 21, pp. 1–10, Jul. 1998.

[144] W. Chen and M. Tanaka, "A meshless, integration-free, and boundary-only RBF technique," *Comput. Math. Appl.*, vol. 43, nos. 3–5, pp. 379–391, Feb. 2002.

[145] Y. C. Hon and Z. Wu, "A quasi-interpolation method for solving stiff ordinary differential equations," *Int. J. Numer. Methods Eng.*, vol. 48, no. 8, pp. 1187–1197, Jul. 2000.

[146] K. Duraisamy, Z. J. Zhang, and A. P. Singh, "New approaches in turbulence and transition modeling using data-driven techniques," in *Proc. 53rd AIAA Aerosp. Sci. Meeting*, Jan. 2015, p. 1284.

[147] K. Duraisamy and P. Durbin, "Transition modeling using data driven approaches," in *Proc. Summer Program*, 2014, p. 427.

[148] A. P. Singh, S. Medida, and K. Duraisamy, "Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils," *AIAA J.*, vol. 55, no. 7, pp. 2215–2227, Jul. 2017.

[149] D. Xiao, F. Fang, A. G. Buchan, C. C. Pain, I. M. Navon, and A. Muggeridge, "Non-intrusive reduced order modelling of the Navier–Stokes equations," *Comput. Methods Appl. Mech. Eng.*, vol. 293, pp. 522–541, Aug. 2015.

[150] D. Xiao, F. Fang, C. Pain, and G. Hu, "Non-intrusive reduced-order modelling of the Navier–Stokes equations based on RBF interpolation," *Int. J. Numer. Methods Fluids*, vol. 79, no. 11, pp. 580–595, Dec. 2015.

[151] D. Xiao, F. Fang, C. Pain, I. M. Navon, and A. Muggeridge, "Non-intrusive reduced order modelling of waterflooding in geologically heterogeneous reservoirs," in *Proc. 15th Eur. Conf. Math. Oil Recovery (ECMOR)*, Aug. 2016, p. cp-494.

LIWEI HU (Student Member, IEEE) received the B.S. degree in software engineering from the School of Software, Hebei Normal University, Shijiazhuang, Hebei, China, in 2014, and the M.S. degree in computer technology from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 2018, where he is currently pursuing the Ph.D. degree in computer science and technology with the School of Computer Science and Engineering. His research fields include aerodynamic data modeling, deep learning, and pattern recognition.

JUN ZHANG received the B.S. and M.S. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), in 1995 and 1998, respectively. From 1998 to 2008, he worked as a Senior Researcher and an Engineer with the China Education and Research Network (CERNET). He is currently a Lecturer with the School of Computer Science and Engineering, UESTC. His current research interests include software-defined networking, machine learning applied in network traffic engineering, and aerodynamics.

YU XIANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, Sichuan, China, in 1995, 1998, and 2003, respectively. He joined the UESTC, in 2003, and became an Associate Professor, in 2006. From 2014 to 2015, he was a Visiting Scholar with The University of Melbourne, Australia. His current research interests include computer networks, intelligent transportation systems, and deep learning.

WENYONG WANG (Member, IEEE) received the B.S. degree in computer science from Beihang University, Beijing, China, in 1988, and the M.S. and Ph.D. degrees from the University of Electronic Science and Technology (UESTC), Chengdu, China, in 1991 and 2011, respectively. He has been a Professor with the School of Computer Science and Engineering, UESTC, in 2009. He has served as the Director of the Information Center of UESTC and the Chairman of the UESTC-Dongguan Information Engineering Research Institute, from 2003 to 2009. He is currently a Visiting Professor with the Macau University of Technology. His main research interests include next-generation Internet, software-designed networks, software engineering, and artificial intelligence. He is a member of the expert board of CERNET and China Next-Generation Internet Committee and a Senior Member of the Chinese Computer Federation.