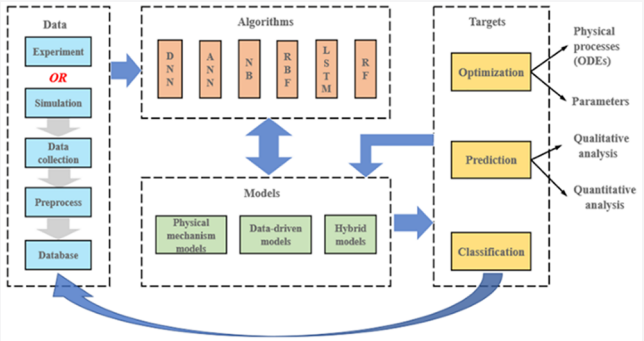
I&EC research
Industrial & Engineering Chemistry Research
pubs.acs.org/IECR
Review

Application of Artificial Intelligence in Computational Fluid Dynamics

Bo Wang and Jingtao Wang*

Cite This: *Ind. Eng. Chem. Res.* 2021, 60, 2772−2790
Read Online
ACCESS | Metrics & More | Article Recommendations

ABSTRACT: This review discusses the recent application of artificial intelligence (AI) algorithms in five aspects of computational fluid dynamics: aerodynamic models, turbulence models, some specific flows, and mass and heat transfer. Currently, there are three main coupling models. The first is the data-driven model to obtain the input−output relationship without involving any physical mechanisms. The second is the physical model to optimize the existing models by AI algorithms. The third is the hybrid model involving both data and physical mechanisms. Among various AI algorithms, artificial neural network is usually applied to build data-driven models and has been successfully employed in the mentioned five fields. Other AI algorithms such as recursive neural network, support vector machine, and naive Bayes are mainly used for the physical models. Finally, the development tendency of coupling models and how to choose an appropriate model are given in the conclusions and prospects.



1. INTRODUCTION

Computational Fluid Dynamics (CFD) is an important interdisciplinary subject, involving mathematics, fluid mechanics, and computer science. Through CFD, the control equations of fluid mechanics could be solved by using numerical methods. It is an effective tool to study fluid flow and the transfer of heat and mass. However, there are still many challenges in CFD development. First, the models describing physical phenomena become more and more complicated with the development of science and technology. Thus, many works are needed to analyze the mechanism and data in order to build and optimize the model, which is time-consuming and laborious. Second, it is difficult to generate an appropriate mesh for the complex model, especially when the geometry is complicated (such as turbine blades). Finally, complex flow fields will generate many data with complicated patterns, which results in high post-processing time.

Recently, more and more scientists are paying attention to artificial intelligence (AI) algorithms and applying them in different areas. The following lists the main reasons:

(1) The improvement of the hardware, such as new computers with higher powers and larger storage medias, lays a solid foundation for AI application.

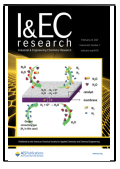
(2) Huge numbers of data are produced nowadays. AI algorithms have been widely used due to their strong capability to process big data.

(3) AI algorithms have the higher accuracy and efficiency in some special fields such as pattern recognition and knowledge engineering.

(4) Similar to the feedback regulation, models established by AI algorithms could be optimized and adjusted through the output data, which proves that AI algorithms have a good ability to generalize.

Because of their advantages, AI algorithms have become effective tools to solve CFD problems and attract more and more researchers. Currently, there are three typical AI-CFD coupling models, including the data-driven model, physical model, and hybrid model. The popular data-driven model requires a large database. When the flow is complicated, CFD data are input into a machine learning (ML) model directly to predict the parameters. This method has advantages such as a simple modeling process, small error, and fast calculation speed, but it is difficult to disclose the physical mechanism. In the physical model, CFD data are employed to build databases in order to optimize the existing models of fluid mechanics. This kind of model requires less data than the data-driven model

Received: October 15, 2020
Revised: February 3, 2021
Accepted: February 5, 2021
Published: February 15, 2021



https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

because of the involvement of physical mechanisms. However, it has a high computation cost and a low calculation accuracy. The hybrid model combines the first two, and usually has a high precision. Although it is still difficult to explain the hidden relationship of data, this model provides a new approach to study the mechanisms.

In this paper, the fundamentals of AI algorithms are given in section 2. Then, the applications of AI in CFD are divided into five parts: the optimization of aerodynamic models (section 3) and turbulence models (section 4), the simulation and prediction of complex flow fields (section 5), and the modeling and verification of heat (section 6) and mass (section 7) transfers. In these five sections, features of CFD models in the corresponding fields are described, respectively, and also, characteristics of AI algorithms and their applications in various CFD fields are discussed in detail. Finally, suggestions how to select the appropriate AI algorithm and build a fine model to solve some specific CFD problems are presented in this paper.

2. FUNDAMENTALS OF ARTIFICIAL INTELLIGENCE

The term "artificial intelligence" was first proposed in 1956. After that, the development of AI was not smooth, but experienced several peaks and valleys. During the past two decades, AI algorithms have been improved rapidly, and attracted more and more attention from different areas all over the world, especially after AlphaGo defeated the best human masters of Go game.

Generally speaking, in the application of AI, the process of training models is to find the suitable risk functions[1] or other similar functions defined by users, and then, minimize them through statistical methods. In the risk functions, there are two most important elements, the loss functions and structural parameters. According to different training strategies, AI algorithms could be divided into three main categories: supervised learning, unsupervised learning, and semisupervised learning.

2.1. Supervised Learning. In the supervised learning, training data are labeled and input into the model to obtain reasonable output results.[2−4] The loss function is usually defined to represent the distance between the calculated and actual values, and thus, the task of training models could be regarded as the minimization of the loss function. Common algorithms of the supervised learning include the neural network (NN), support vector machine (SVM), random forest (RF), and decision tree (DT). Neural network[5] is one of the most famous algorithms and has been widely employed in various fields. By changing the structure parameters and activation functions of the neural network, various neural networks with different functional focuses could be obtained, such as the deep neural network (DNN), convolutional neural network (CNN), and recursive neural network (RNN). Since there are many hidden layers in DNN, it could deal with complex nonlinear problems with more input parameters. At present, certain progresses have been made in the image recognition and market analysis.[6,7] As for CNN, it replaces the activation function of the traditional neural network with a filter, which greatly reduces the number of parameters to be trained in the network. Thus, CNN could extract the features of input parameters effectively, which makes it very successful in the image and pattern recognition.[8,9] Nevertheless, in some applications, it is necessary to consider the order of input data. Thus, the long short-term memory (LSTM) is combined with the neural network, and also, the memory unit and gate control are added into the neural network. This kind of networks could retain the training results of the previous parameters when processing the latter parameters. On this basis, LSTM is widely employed in the natural language process, computer vision, and other fields.[10,11]

2.2. Unsupervised Learning. Compared to the supervised learning, data of the unsupervised learning has no labels and no pretraining process.[12] On the basis of different goals, problems treated by the unsupervised learning could be the dimension reduction, clustering, and quantization. Common algorithms of the unsupervised learning include neural network, principal component analysis (PCA), clustering analysis, anomaly detection, hierarchical clustering algorithm, etc. Among them, PCA is one of the most famous algorithms. It could reduce the data dimension effectively through the orthogonal transformation which could transform a set of correlated variables into a new set of variables which are linearly independent. Thus, PCA is mainly applied for the data compression, especially for image processing.[13]

2.3. Semisupervised Learning. Semisupervised learning, just like its name, combines the supervised learning with the unsupervised learning, which means that the training set includes both labeled and unlabeled data. This learning strategy could get highly accurate results at a lower cost, and thus, gradually become popular.[2,4,14−16] Common algorithms of the semisupervised learning include generative adversarial nets (GANs), low-density separation, graph-based methods and joint training. Among them, GANs is one of the most promising algorithms. The key idea is to construct the generator and the discriminator network, respectively, and then, make them competing in order to judge the true data dimension. GANs has been broadly employed in the digital image processing,[17] video tagging, and other fields due to its low computation cost.

2.4. Artificial Neural Network. Among the three AI strategies mentioned above, the artificial neural network (ANN) is the most typical. It is a mathematical model to mimic the biological neural network in order to realize the information processing like human brains.[18] In neural networks, the smallest computation unit is the neuron. It takes input parameters from other neurons, processes them through activation functions, and then, outputs the results. The functions of a neuron include summing the inputs and activating them. Figure 1 shows the

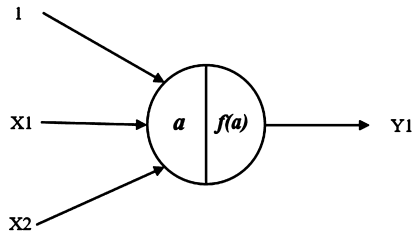


Figure 1. Illustration of a simple neuron structure.

neuron with two input parameters and one output parameter. Each input corresponds to a weight w, and there is another important parameter *bias* (b) for the input. Thus, the summation parameter a could be expressed as $a = x_1w_1 + x_2w_2 + b$. Each neuron will apply the activation function $f(a)$ to calculate the output.

In the engineering application, as the relation between the input and output is usually nonlinear, the corresponding activation function $f(x)$ is nonlinear. Generally, the complex input data are compressed and normalized by the activation

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

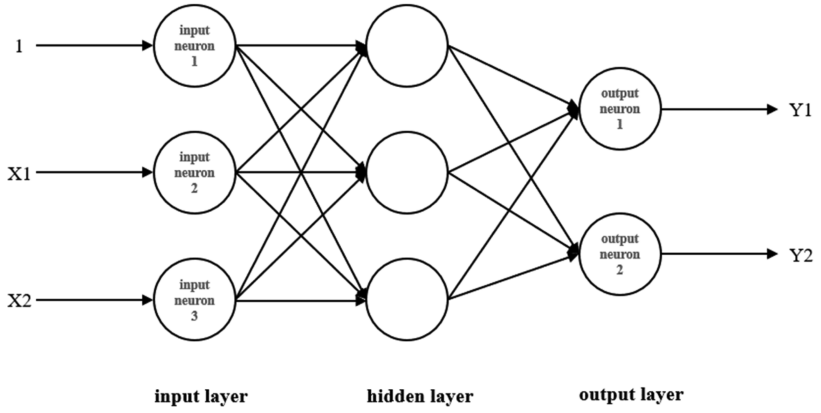


Figure 2. Neural network structures.

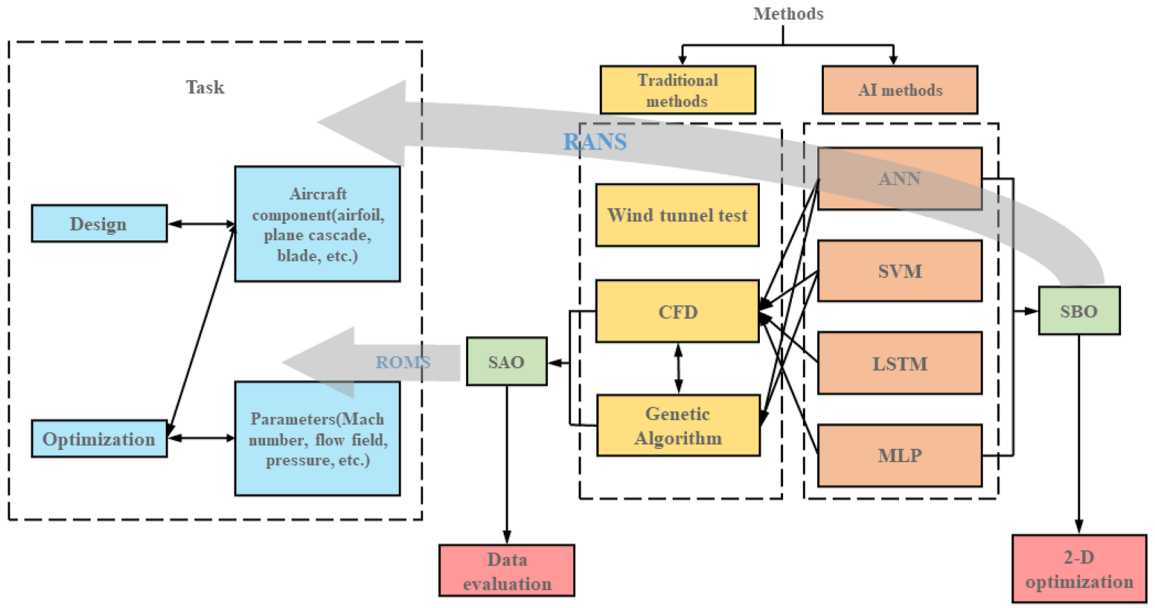


Figure 3. Framework of the application of AI algorithms in aerodynamic models.

function. Common nonlinear activation functions include sigmoid, tanH, and Relu,

$$\text{sigmoid:} \quad f(x) = \frac{1}{1 + e^{-x}} \quad f(x) \in [0, 1] \tag{1}$$

$$\tanh: \quad f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad f(x) \in [-1, 1] \tag{2}$$

$$\text{Relu:} \quad f(x) = \max(0, x) \quad f(x) \in [0, +\infty) \tag{3}$$

The appropriate activation function should be determined on the basis of the processed data.

Similar to the biological neural network, ANN is composed of the input layer, hidden layer, and output layer (see Figure 2). The input layer receives the preprocessed (such as formatting) data and passes them to the hidden layer. The neurons in the hidden layer calculate and transform the data, and then transfer them to the output layer. In the neural network, it is possible that there are multiple hidden layers or no hidden layer at all. As for the latter, the input parameters could be directly calculated by the learning function to get the target output. Thus, it could only process the linear data. As data are mainly nonlinear in engineering applications, the most important function of the hidden layer is to enable the neural network to disclose the nonlinear relationship. The main function of the output layer is to calculate and transform the input from the hidden layer to obtain a reasonable prediction.

3. APPLICATION OF AI IN AERODYNAMIC MODELS

The modeling of aerodynamic applications is an important research area in the aircraft design and optimization. The main goals of studying aerodynamic models have been divided into two categories as shown in Figure 3. One is designing the solid parts or equipment involved in aerodynamics, such as airfoils, turbines, and blowers. The other is parameter optimization, for instance, the aerodynamic optimization of missiles and airfoils under different Mach numbers and pressures. Although traditional methods, such as the wind tunnel test and CFD simulation, are still useful in aerodynamics, it faces some new challenges due to the high complexity of the design object and the demand of high accuracy. Machine learning (ML) algorithms based on the data analysis have unique advantages, and were gradually applied for the optimization and design of aerodynamic models. As early as 1974, these algorithms had been applied in the aerodynamic models by Hicks et al.[19] They successfully solved the optimization problem of the airfoil

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

suitable for a wide range from the low to supersonic velocity by using a kind of optimization algorithm based on the feasible direction method and the equation solving the transonic problem with small disturbances. Nowadays, ML algorithms have been employed in the aerodynamic models to estimate parameters by analyzing and calculating a large number of samples. As shown in Figure 3, there are two common methods: one is to use surrogate models to replace CFD completely, which is called surrogate-based optimization (SBO); and the other uses surrogate models to replace CFD partially, called surrogate-aided optimization (SAO). Both of them are developed by the assistance of ML algorithms, but they are quite different. In SBO, all CFD data are employed to train ML models and no latent mechanisms are disclosed by these models. While, as for SAO, it optimizes the existing aerodynamic models and discloses relevant mechanisms based on partial CFD data. Thus, SAO could solve some complex problems, such as the data evaluation of 3-D unsteady aerodynamic calculation, while, SBO is more suitable to 2-D simulation. The targets of ML modeling are to get more precise results than CFD and accelerate the computation process. Common ML algorithms include MLP, LSTM, and SVM, etc. The combination of ML with traditional algorithms is the key of the future researches in this field. In current works, the combination of GA with ANN and SVM has been done, and all of these four algorithms shown in Figure 3 have been applied successfully in CFD.

3.1. Aerodynamic Optimization. In the aerodynamic optimization, SBO has been widely employed since a large amount of data are available. In SBO, the combination of genetic algorithm (GA) with ANN is an effective technology (see Figure 4). Generally, GA is employed to optimize mesh data, which

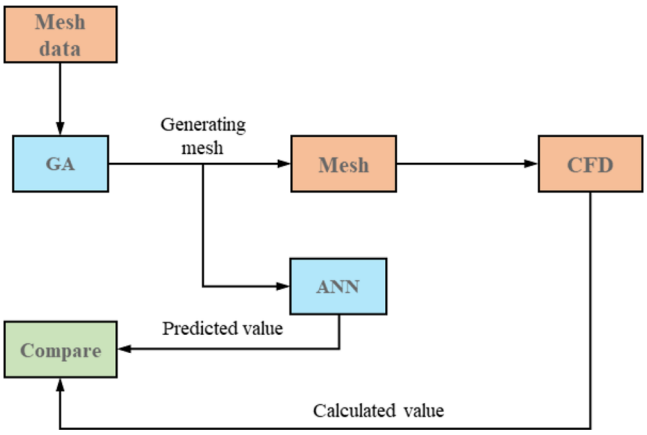


Figure 4. Illustration of the GA-ANN coupling method in the CFD process.

determines the structure of the aircraft components. The optimized (offsprings) mesh data are obtained from the previous mesh data (parents) according to eq 4,

$$\text{crossover}(x_1, x_2) = \text{rand}(0, 1)\cdot(x_2 - x_1) + x_2 \quad (4)$$

ANN can construct an input−output relation between the mesh data optimized by GA and the results of CFD, as shown in Figure 4. By comparing the prediction of ANN to CFD results, the better result is adopted and a better mesh (physical structure) is obtained. Because the time of training and prediction is less than that of a CFD calculation, ANN could accelerate this process significantly. Chen and Agarwal[20] applied the GA and GA-ANN method to optimize airfoils of FB series, respectively, and disclosed that these two methods could obtain the optimal result within a certain attack angle. For the GA-ANN method, ANN could increase the computational efficiency significantly. Su et al.[21] optimized the traditional GA based on MLP and applied this method to the aerodynamic optimization of missiles. The results showed that this GA-MLP model could greatly reduce the computational burden and was superior to GA under the same condition. Similarly, Mengistu and Ghaly[22] combined MLP with GA to optimize the simulation of the blade movement described by RANS equations. This model was highly consistent to the objective function and could shorten the calculation time by 10 times. Zhang et al.[23] proposed a method of GA with hierarchical fair competition dynamic-niche (GA-HFCDN) based on MPL and GA, and optimized the plane cascade and blower blades.

The disadvantage of SBO is that a large number of samples are required to establish a ML model. Thus, SBO is mostly applied for the two-dimensional optimization, and the results sometimes differ greatly from those of CFD. As for SAO, CFD is still employed, and ML algorithms use partial data for the estimation. It could increase the efficiency and ensure the accuracy of the optimization results. Giannakoglou et al.[24] combined MLP and GA for the pre-evaluation of the airfoil optimization. Compared to traditional CFD methods, this algorithm could reduce the calculations by an order of magnitude. Asouti[25] proposed an asynchronous auxiliary evolutionary algorithm based on radial basis function (RBF) response surface and optimized the design of airfoils and compressor cascades. The training set was quite small with only 12 × 10 supporting mesh. Iuliano and Perez[26] combined SVM with the traditional GA, studied the effects of four geometric parameters on the cyclone. Through this combination model, a new type of high performance cyclone was designed by limited data. Compared to SBO, SAO is more suitable for the problem with a low data volume, and thus, it is often employed in the design process.

3.2. Unsteady Aerodynamic Simulation. Unsteady aerodynamic prediction is complicated but important in the modeling of aerodynamics. High-precision CFD methods have been broadly used for this kind of problem. However, there are some disadvantages, such as the high computation cost and low efficiency, especially when calculating some complex engineering cases. To solve these problems, unsteady reduced-order models (ROMs) are established and a series of results have been obtained. Mannarino and Mantegazza[27] proposed a ROM based on a continuous time recurrent neural network (CTRNN) and verified its feasibility through the airfoil design. However, this model is only applicable to steady problems and is no longer suitable when Mach number changes. Kujins and Cesnik[28] proposed a ROM available to the unsteady aerodynamic simulation under different Mach numbers by combining the linear convolution with nonlinear factors. The error of this model increases along with the increase of Mach number. To establish a universal model, Li[29] developed a ROM based on LSTM. This model could capture the time-delayed effect and train a large number of time-series data, and thus, it is suitable for the large Mach number. The RANS method was adopted, and the structure equation of motion was given

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

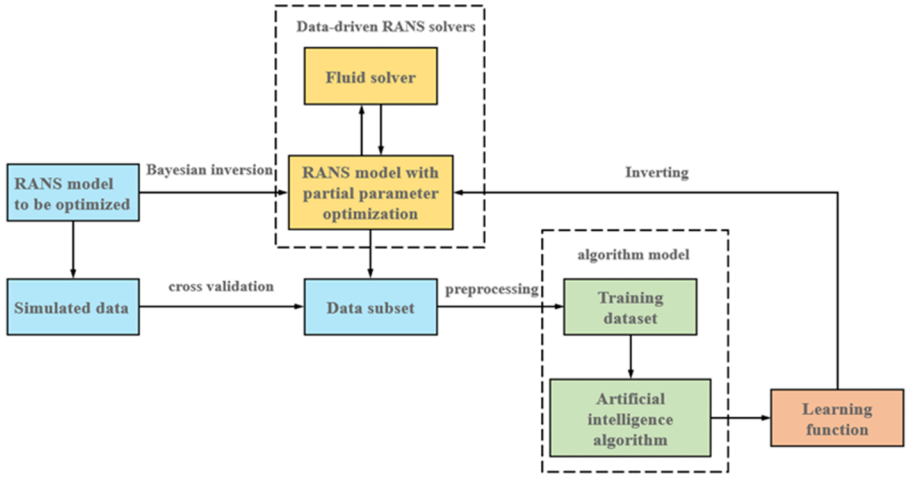


Figure 5. Optimization process of the RANS model based on data

$$\begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix} \begin{Bmatrix} \ddot{h}/b \\ \ddot{\alpha} \end{Bmatrix} + \begin{bmatrix} (\omega_h/\omega_\alpha)^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix} \begin{Bmatrix} h/b \\ \alpha \end{Bmatrix} = \frac{1}{\pi}(V^*)^2 \begin{Bmatrix} -C_l \\ 2C_m \end{Bmatrix} \tag{5}$$

The data set was built by five groups of random motion data with various Mach numbers, which were calculated by a CFD solver. The input parameters were Mach number (Ma), pitching angle (α), and transonic airfoil plunging displacement (h/b). The output variables were the lift coefficient (C_l) and pitch moment coefficient (C_m). Different from the CFD solver, the LSTM neural network disclosed the nonlinear mathematical relationship between the inputs and outputs

$$y_t = f(u_t, y_{t-1}, C_{t-1}) \tag{6}$$

where $u_t = [(h/b)_t\ \alpha_t\ Ma]$ was input parameters vector, and $y_t = [(C_l)_t\ (C_m)_t]$ was output variables vector. By coupling the structure equation of motion with LSTM, the structure equation is solved and the aerodynamic load of airfoils is predicted. This coupling model could improve the generalization capability of the traditional ROM, and its calculation time was only 6% of that of CFD. Because of the complexity of unsteady aerodynamics, the physical mechanism should be considered when AI algorithms are employed to treat relevant data. Thus, LSTM is a strong technique[103,104] and has been commonly employed as it contains the pattern of the data during the training process.

4. APPLICATION OF AI IN TURBULENCE MODELING

Traditional methods to study turbulence include the classical statistical theory and the model theory. However, turbulence equations cannot be established accurately by the above two methods due to the complexity. With the development of AI, ML has been employed to train a large amount of data of turbulent flows in order to obtain high-precision models and prediction results. This has become a new way to solve turbulence problems.

There are three kinds of methods to calculate turbulent flows: Reynolds-averaged Navier−Stokes (RANS), direct numerical simulation (DNS), and the large eddy simulation (LES). RANS transforms the unsteady turbulence problem to the stationary problem and has the advantage of computational tractability. However, compared to Navier−Stokes (N−S) equations, the closure of Reynolds stress in RANS equations has many problems. As early as 2002, Milano and Koumoutsakos et al.[30] employed the neural network algorithm to predict the turbulent flow near walls. They reconstructed and decomposed the predicted results by employing the orthogonal method, which proved the feasibility of using the neural network to solve the closure of RANS equations.

Nowadays, there are two kinds of methods to solve turbulence problems through ML algorithms. One embeds ML methods into the existing turbulence model. By analyzing and processing a large amount of data, the empirical parameters in the turbulence model are optimized to obtain better prediction results. The optimization process of the RANS model through the AI algorithms is shown in Figure 5. Data sets are generated by the RANS model which is to be optimized, and parameters of the RANS model are obtained by Bayesian inversion. By training the data set, RANS parameters are optimized and the new model is embedded into the fluid solver. By repeating these steps, the RANS model could be more and more accurate. Tracey et al.[31] adopted a neural network based on supervised learning algorithms to optimize the source term of Spalart-Allmaras (S-A) models. The full model was

$$\frac{\partial \hat{\nu}}{\partial t} + u_j\frac{\partial \hat{\nu}}{\partial x_j} = c_{b1}(1 - f_{t2})\hat{S}\hat{\nu} - \left(c_w f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right)\left(\frac{\hat{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left(\frac{\partial}{\partial x_j}\left((\nu + \hat{\nu})\frac{\partial \hat{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \hat{\nu}}{\partial x_i}\frac{\partial \hat{\nu}}{\partial x_i}\right) \tag{7}$$

and the source term s included three parts: production (eq 8), destruction (eq 9), and cross production (eq 10),

$$s_p = c_{b1}(1 - f_{t2})\hat{S}\hat{\nu} \tag{8}$$

$$s_d = \left(c_w f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right)\left(\frac{\hat{\nu}}{d}\right)^2 \tag{9}$$

$$s_{cp} = \frac{c_{b2}}{\sigma}\frac{\partial \hat{\nu}}{\partial x_i}\frac{\partial \hat{\nu}}{\partial x_i} \tag{10}$$

The data were obtained by calculating many kinds of flows in S-A model solver. First, the training set was constructed by extracting input (ν, $\hat{\nu}$, Ω, d, $N = \frac{\partial \hat{\nu}}{\partial x_i}\frac{\partial \hat{\nu}}{\partial x_i}$) and an output feature vector ($S_p S_d S_{cp}$) from each mesh location. Then, the ML model

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

was trained on the training set to replace the source term. Finally, the learned ML model was embedded into the S-A solver to generate a ML-CFD coupling solver. This solver could predict various flow situations from a 2-D planar boundary layer to a 3-D airfoil successfully, even those not in the training set. Similarly, Zhang and Duraisamy[32] used a neural network to predict the production term coefficient (an adjustment term optimized by the model) in the turbulence model. This kind of method is always employed to optimize the turbulence models, which means that its application range is narrow and its generalization capability is weak.

The other method is to disclose the relationship of the input and output by only analyzing the relevant data. In this method, neural network algorithms and naive Bayesian estimation algorithms are commonly employed. Tracey et al.[33] predicted and reconstructed the tensor eigenvalues of the regularized Reynolds stress term by means of a ML model based on kernel regression. The data set contained high-fidelity DNS data and low-fidelity RANS data. There were four input features: x_{bary}, y_{bary}, marker function m, and the ratio of production of turbulent kinetic energy to its dissipation P/ε, where x_{bary} and y_{bary} were calculated by eqs 11−13,

$$C_{1c} = \lambda_1 - \lambda_2, \quad C_{2c} = 2(\lambda_2 - \lambda_3), \quad C_{3c} = 3\lambda_3 + 1 \tag{11}$$

$$x_{bary} = C_{1c}x_{1c} + C_{2c}x_{2c} + C_{3c}x_{3c} \tag{12}$$

$$y_{bary} = C_{1c}y_{1c} + C_{2c}y_{2c} + C_{3c}y_{3c} \tag{13}$$

As the learning algorithm of ML models, the kernel function was

$$K(x_i, x_j) = \exp(-\sum_k (x_{i,k} - x_{j,k})^2/\tau_k) \tag{14}$$

The output features were new eigenvalues $\tilde{\lambda}_i$, which could be injected into Reynolds stresses by the following equation

$$\overline{u_i'u_j'} = \frac{2}{3}\kappa\delta_{ij} + \kappa(v_i\tilde{\lambda}_i v_j) \tag{15}$$

This technique could improve the accuracy of low-fidelity models by using data sets which contained high-fidelity data, and thus, saving computation costs indirectly. Cheung et al.[34] employed Bayesian uncertainty quantification technology (BUQT) to predict quantities of interest (QoI) and calibrate complex mathematical models. It proved that some model classes could be rejected by combing model plausibility with predicted QoI. Oliver et al.[35] used BUQT to optimize RANS turbulence model and analyzed the flow field of channel flows under different Reynolds Numbers. Through the uncertainty analysis and parameter calibration, the uncertainty is reduced to 4%. Portwood et al.[36] did not optimize an existing turbulence model, but captured the potential turbulence phenomenon. Then, they established a continuous-time, generative neural ordinary differential equation (ODE) model (called latent ODE) driven by data and predicted the evolution of dissipation rates accurately under several targets. It proved that this model has higher accuracy than DNS. Duraisamy et al.[37] studied the turbulence by analyzing the uncertainty of the RANS model. Four kinds of uncertainties are disclosed: the uncertainty of ensemble average, of the functional form, of the coefficient in the model, and the uncertainty expressed by Reynolds stress function. By using AI algorithms for the uncertainty analysis and optimization, it was greatly reduced, and a stable turbulence model was obtained. This proves that the uncertainty analysis could be a new idea to solve the turbulence and other complex flow problems.

Unlike RANS, DNS has a high fidelity to solve turbulence problems and has become a popular method.[38−41] As the data generated by DNS usually have high accuracy, it is used to construct a database for the data-driven model. Wang et al.[39] built a random forests model, trained discrepancy functions by DNS databases, and then, predicted Reynolds stress of two different types of flows. The feature vector $q(x)$, as the input, contained 10 features ($q_1 \sim q_{10}$) and was calculated by the RANS model. The output (or response) was the discrepancies field $\Delta\tau_\alpha(x)$, which was the difference between the high-fidelity data and Reynolds stresses of the RANS model. The regression functions f_α mapped the feature vector $q(x)$ to the response $\Delta\tau_\alpha(x)$ as shown in eq 16,

$$f_\alpha : q \mapsto \Delta\tau_\alpha \tag{16}$$

This method has a good performance in the prediction and could be generalized to other prediction problems.

LES is another popular method. It can solve flows with strong transients, and it usually has different grid resolutions for different time/length scales. This method requires higher grid density in the flow field, especially for the region near the walls, and thus, it requires more computation resources.[42−45] To solve these problems, ML algorithms have been coupled with LES, such as the equation discretization and model optimization.[46−48] Maulik et al.[49] reported the application of AI algorithms in LES and spatiotemporally dynamic turbulence model classification. The nodes of the grid were classified, and for a hybrid closure, the turbulence model is blended with the prediction of the conditional probability of the same neural network. Yang and Zafar[50] trained the neural network in a wall-modeled LES to study the wall-bounded flows. As the input was velocity u and the output was wall stress τ, the objective function was

$$e = \frac{1}{N}\sum_{n=1}^{N} \omega_n[\tau_n - f_{NN}(u_n)]^2 \tag{17}$$

To improve the generalization capability of this model, the wall stress τ was corrected by the following equation,

$$\tau^+_{\omega,n} = \begin{cases} h_n^+, & y^+ \le 11.3 \\ \left[\dfrac{\kappa u_n^+}{\ln(h_n/y_0)}\right]^2, & y^+ > 11.3 \end{cases} \tag{18}$$

This model could capture the wall law under any Reynolds numbers and was superior to the traditional subgrid scale (SGS) model in nonequilibrium flows. Thus, it has gradually replaced the traditional models.

5. APPLICATION OF AI IN SOME SPECIFIC FLOWS

In this section, the application of AI in some specific flows, including flows passing a cylinder, boundary layer flows, multiphase flows without solid particles, and multiphase flows containing solid particles will be discussed. CFD is a common method to study these specific flows, and various models have been employed. For example, volume of fluid (VOF) model is used to capture the interface of two phases (liquid−liquid or liquid−gas). Discrete element method (DEM) is employed to study the multiphase flows with solid particles. However, the interaction of fluid-structure and phase−phase, and some

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

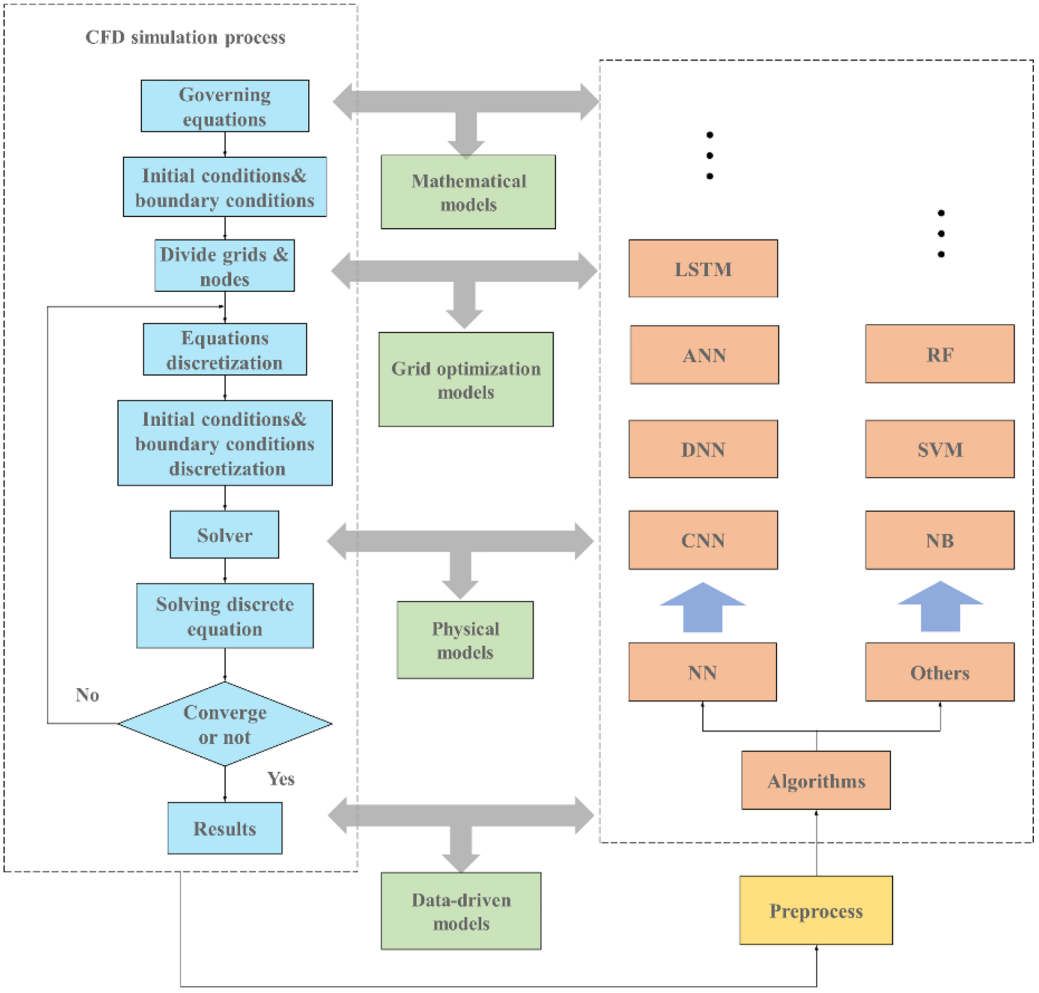


Figure 6. Four models generated by the combination of CFD with AI algorithms.

microphysical mechanisms (such as eddies or microscale) are still unclear, which means that the accuracy of CFD needs to be improved further. As AI algorithms have unique advantages in data processing and analysis, the combination of AI with CFD models has become an important area.

The process of CFD simulation is shown in Figure 6, in which each step could generate a large amount of discretized equations and data. By coupling these data or equations with AI algorithms, four types of models could be obtained: the mathematical model, grid optimization model, physical model, and data-driven model (see Figure 6). For the mathematical model, AI algorithms are usually used to solve the governing equations and optimize their numerical solutions, such as optimizing the numerical solutions of ODE by DNN. This type of model is designed to search more accurate numerical solutions of the equations, and could improve the result accuracy effectively. The grid optimization model could reduce the computation time greatly. It takes the advantage of AI algorithms in the data processing to optimize the local grid, or to generate the domain grid more reasonably and efficiently. The physical model is a combination of ML with CFD solver, and the algorithms are employed to analyze the flow data or CFD model. It could make the calculation results approach the real values by optimizing and adjusting the parameters of the original models. Thus, this kind of model is a semidata-driven model (or hybrid model), which involves not only data but also the physical mechanism. As for the data-driven model, since it could build the database and analyze the input−output relationship or data pattern, it is often employed to predict the results or verify the accuracy of a new model.

From section 4, it has been known that AI algorithms have become popular and made some progresses in the field of the general turbulence. In this section, we will discuss the application of AI algorithms in several special flows which have their own complexity.

5.1. Studies of Flows Passing a Cylinder. As the flow passing a cylinder has spatiotemporal instabilities when *Re* number changes, it has become a focus in recent years. Erichson et al.[51] proposed an automatic encoders model based on physical attributes, which could add some physical mechanisms in the learning process, and thus, enhance the model stability. The sequence of training data $y_0,y_1,\cdots,y_T \in R^m$ was obtained by CFD, and the target of this model was to provide a mapping from y_T to y_{t+1}. The model contained encoder ψ, dynamics Ω, and decoder ϕ, which could be described as

$$\hat{y}_{t+1} = \Phi \cdot \Omega \cdot \Psi(y_t) \tag{19}$$

where ψ mapped the high-dimensional y_t to a low-dimensional feature space ($R^m \rightarrow R^n, m \gg n$). The dynamics Ω, $R^n \rightarrow R^n$, which evolved the state in time, was given as

$$z_{t+1} = \Omega z_t \tag{20}$$

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

The decoder ϕ mapped the low-dimensional feature space to the high-dimensional y_t ($R^m \rightarrow R^m$). Finally, the model was trained by minimizing the mean squared error according to

$$\mathrm{MSE} = \min \frac{1}{T-1} \sum_{t=0}^{T-1} \left\| y_{t+1} - \Phi \cdot \Omega \cdot \Psi(y_t) \right\|_2^2 \tag{21}$$

This new training method was effective to deal with the problem with limited data and to improve the generalization performance. It predicted the flow field around the cylinder successfully, and proves that the neural network had obvious advantages in the processing of data with physical properties. In addition, they also reconstructed the high-dimensional flow field with the data obtained from CFD and experiments.[52] In their work, they constructed a data-driven shallow learning model based on the neural network without preprocessing the data and without using any existing physical models. This data-driven method is superior to the traditional CFD.

Lui and Wolf[53] proposed an order reduction model of flow fields by combining the fluid modal decomposition with the regression analysis. The key idea is to filter the time mode of an orthogonal decomposition and to use the orthogonal decomposition to reduce the model dimension. The regression step is realized by the feed-forward DNN based on the nonlinear dynamic sparse identification algorithm. They evaluated the compressible fluid passing a cylinder and verified the accuracy of the model. Moreover, they calculated the flow field of the downdip airfoil with dynamic stall by using the large eddy simulation, and compared their results to the experimental data. This model could capture the nonlinear flow behaviors of the leading stall vortex and the trailing edge vortex. Compared to the sparse regression method, DNN could capture the transient characteristics of flows and improve the accuracy and stability of the model. These works disclose that the neural network is suitable to solve this type of problems.

5.2. AI and the Boundary Layer. In the boundary layer, the influence of viscous forces cannot be neglected, which results in a large velocity gradient and a complex velocity field. For this reason, it is necessary to apply AI algorithms for the flow calculation in the boundary layer. Safarzadeh et al.[54] studied the convection in the boundary layer by using the DE-NN coupling method. The initial population of differential evolution (DE) algorithm could be generated by

$$X = X_{\min} + \alpha(X_{\max} = X_{\min}) \tag{22}$$

In this study, two hybrid DE models were established to calculate the flows. They were the multilayer perceptron model based on differential evolution (DE-MLP) and the RBF model based on differential evolution (DE-RBF). These two models could calculate the separation of the middle and the low layer, the downward deflection of the upper streamlines, and the development of the horseshoe vortex. In addition, DE-MLP could calculate the distribution of Reynolds stresses along the flow direction of the mixed layer. However, DE-RBF could not predict the flow features accurately, and the results are often smaller than the actual values when the parameters are changed. Besides, it could not predict the inflection point of the transverse velocity distribution. Nevertheless, this method still provides a new idea to describe the generation and evolution of the boundary layer.

Reddy and Das[55] employed a series of methods including Runge−Kutta fourth-order method, targeting technology, ANN and back-propagation neural network to predict the boundary layer sliding flow field with chemical reactions. First, the nonlinear partial differential equations (PDEs) of the control flow were transformed into a system of highly nonlinear ordinary differential equations by the similarity transformation as eqs 23−25,

$$f' = z, \quad z' = p,$$
$$p' = (z^2 - fp - 2Mp + (D + k_1)z - \lambda\theta - \delta\phi) \tag{23}$$

$$\theta' = q, \quad q' = -[\mathrm{Pr}(fq - Nz\theta) + 2Mq]/(1 + 2M\eta) \tag{24}$$

$$\phi' = r,$$
$$r' = -[sc(fr - Pz\phi - \beta\phi) + 2Mr]/(1 + 2M\eta) \tag{25}$$

Second, the system was solved numerically by Runge−Kutta method and shooting technology, and then, the intermediate data were trained by ANN. To verify the accuracy of this method, the back-propagation neural network was employed to predict the parameters, including surface friction coefficient, Nussel number, and Sherwood number. The numerical results with the neural network are in good agreement to those of the back-propagation method. Thus, this new model was fully applicable to the boundary layer slip flow on a stretched cylinder. It not only considers the physical meaning of each parameter in the flow field, but also makes use of the advantage of the neural network in data processing. In order to predict the boundary layer change in the system, AI algorithms were employed in the above study to optimize the physical model.

5.3. Multiphase Flows without Solid Particles. Similar to the turbulent flow, DNS is generally employed to calculate multiphase flows without solid particles. In this section, two-phase (gas−liquid and liquid−liquid) and three-phase (liquid−liquid-gas) flows are discussed. The data produced by DNS are the basis of AI algorithms to simulate the behavior of multiphase flows. Ma et al.[56] used the mean artificial neural network (MANN) to calculate the bubble rising in a vertical channel, so as to study the relationship between the unknown closure terms and variables in N−S equations. Data in the database were averaged over the planes parallel to the walls, and the closure relationship was verified by changing the initial conditions. In this work, the predicted results of the proposed model are consistent to those of DNS. Furthermore, they also simulated the gas−liquid flow with bubbles generated periodically through a data-driven model.[57] This model could greatly simplify the calculation of the multiphase flows. Although a statistical learning algorithm could also be employed in the simulation of multiphase flows, the neural network has become more popular as it requires less input data and has a better prediction accuracy.

In addition to two-phase flows, AI algorithms could also be applied to simulate three-phase flows. Roshani and Feghhi[58] constructed a multilayer perceptor (MLP) neural network to predict the concentration of each phase in the oil−water−gas three-phase flow. There are two input parameters, one output parameter, and a detector in the neural network. By comparing the predicted results to those of CFD, the error between the prediction and simulation is quite small and negligible. However, MLP neural network does not involve the mechanics of this three-phase flow, and thus, whether it is the best AI algorithm to solve this kind of problems remains to be discussed.

5.4. Multiphase Flows Containing Solid Particles. Multiphase flows containing solid particles are very complicated due to complex interactions such as particle-to-fluid and

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

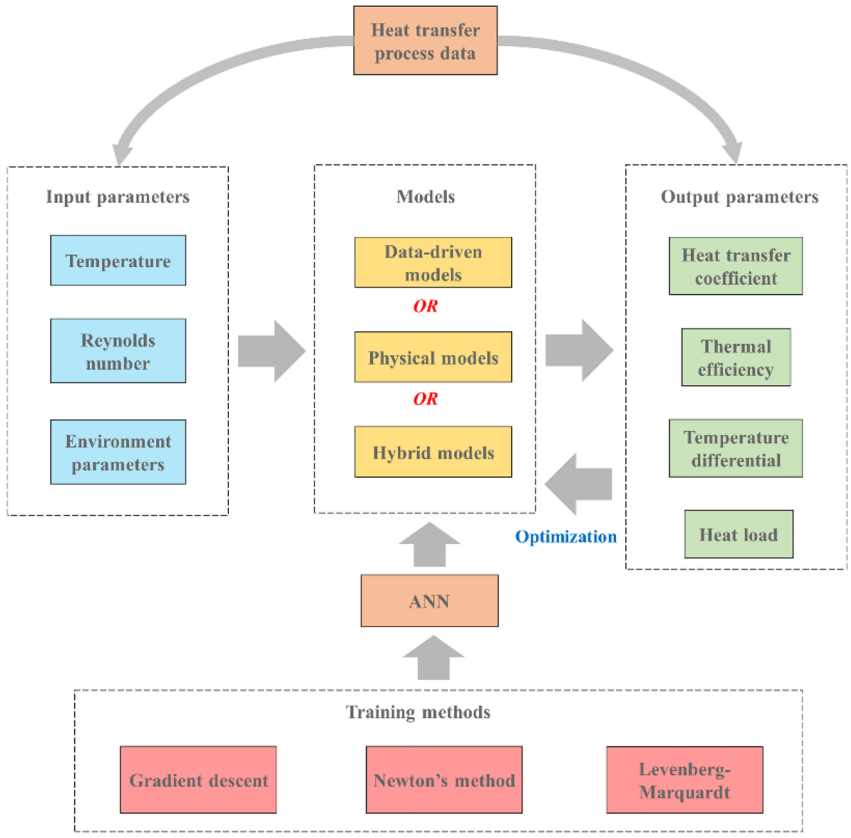


Figure 7. Models and parameters in heat transfer.

particle-to-particle. The solid particles could enhance the chaos of flows, and the change of flows will vary the movement of those particles. Thus, developing a suitable method to calculate the interaction force between the solid and liquid is very important. In CFD of multiphase flows, generally there are two kinds of methods to simplify the solid particles. In the Euler−Euler model, the solid particles are treated as continuous phase. Indeed, the computation cost is relatively small. However, the interface (between the solid and liquid) model is not complete and its accuracy is not quite high due to the oversimplification and indirect numerical simulation. In the Euler-Lagrangian model, the solid particles are treated as the dispersed phase. As the dispersed and continuous phase are calculated in each own computation unit, this method has a higher precision. It could calculate the particle−fluid interaction theoretically, but the calculation cost is big when the volume fraction of particles is high. Nevertheless, these methods are not universal and depend on the specific system, since the variable mapping between different methods and the interaction between different phases are different.

The neural network based on the statistical learning method has made progresses to solve PDEs and nonlinear problems, which means that it is possible to employ the neural network to solve the particle−fluid system.

Pirnia et al.[59] constructed an ANN model to predict drags of small spherical particles in a particle−fluid system. The drags were calculated through the rough grid method and were trained by ANN. The drag of an individual particle was calculated by the weight of the particles on a large scale. Compared to the results of finite element method, ANN could predict the drags of small particles with a high accuracy, but it could not give the exact drags of large particles. This study combined the traditional CFD with ANN, and provided a new idea to verify the rationality of the drag model.

He and Tafti[60] proposed a particle resolved simulations (PRS) method to construct the optimal drag model. It is a data-driven method based on the supervised machine learning. PRS was employed to generate the drag model under the framework of DEM, and ANN was employed to train the model. The governing equation of this model, which generated data, included the continuity eq (eq 26), momentum eq (eq 27) and mean pressure gradient eq (eq 28),

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{26}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_j u_i)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re_{\text{ref}}}\left(\frac{\partial^2 u_i}{\partial x_j \partial x_j}\right) + \beta \vec{e_x} \tag{27}$$

$$P(\vec{x}, t) = -\beta \cdot x + p(\overrightarrow{x, t}) \tag{28}$$

The inputs of ANN were the porosity, adjacent particle positions, and Reynolds numbers, while the output was the drag. Results of this work showed that the drags of 68% of the particles could be predicted within the error of 15%. Although this technology has a wide margin of errors, it shows the possibility of using the supervised learning to develop drag models with more precision.

Besides spherical particles, the simulation of solid−liquid flows of nonspherical particles through ANN has been done. Yan et al.[61] constructed a back-propagation neural network and a RBF-ANN, and predicted the drag coefficients of particles with different sphericities in the dense gas. Compared to experimental results, these two networks could predict the

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

drag coefficients accurately. On the basis of the results of the neural network, the curves of the resistance coefficients, Reynolds numbers, and particle sphericities were all fitted, and a new model of the drag coefficient was developed. To testify the accuracy of the model, it was used in the simulation of the fixed bed and the bubbling fluidized bed, and the results were consistent to those of experiments. This meant that ANN could be applied in the prediction of the drags of nonspherical particles in gas−solid systems. This work is a good example to predict the drag coefficient of particles with complex shapes in multiphase flow systems.

Currently, the application of AI in this area tends to avoid analyzing complex physical processes through the black-box operation, and only focuses on the data relationship. It is a kind of preliminary application and has some drawbacks, such as the overfitting, the output difficult to explain, and so on. In the future, it is important to analyze the physical processes through AI, which might disclose the hidden mechanisms and avoid the overfitting to some degree.

6. APPLICATION OF AI IN HEAT TRANSFER

Heat transfer is ubiquitous in nature, daily life, and industry. It has always been important in the energy, metallurgy, machinery, chemical engineering, and other fields. Recently, many interdisciplinary subjects such as the multiphase heat transfer, superlow temperature heat transfer, microscale heat transfer, and biological heat transfer have appeared. Also, the measurement of relevant parameters, such as the heat transfer coefficient, entropy change and heat load, and the optimization of heat-transfer equipment, has attracted more and more attention.

Experiments[62−64] and CFD calculations[65−68] are the two most common methods to study heat transfer. However, the cost is large and the useful samples are limited when there are too many experimental variables. As for CFD, similar to that in the previous section, the computation cost is high when the problem is complicated especially for nonlinear PDEs. Besides, it is also difficult to calculate problems without clear physical mechanisms. Thus, AI has become very popular to solve these problems.

On the basis of the data produced by experiments and calculations, AI could be employed to generate different models.[69−72] In the data set, sample labels have been given due to the physical relationship, and thus, the supervised learning (SL) is applicable. Among different algorithms of SL, ANN is the most popular and has been employed in many works successfully.[73−80] Three typical methods, including gradient descent, Newton's method, and Levenberg−Marquardt have been used to train ANN model (minimizing the loss function), as shown in Figure 7. Among them, the gradient descent is the slowest and requires the least memory. In contrast, Levenberg−Marquardt is the fastest, but requires the most for memory, while Newton's method could balance the two requirements very well. In heat transfer, parameter numbers in training data are limited[82−90] (usually not more than 100), and the Levenberg−Marquardt method could meet the requirement of high precision. If there are more than 100 parameters and the data set is quite large, Newton's method is suitable. Generally speaking, the gradient descent is not so popular. According to the size of the data set, models are typically divided into physical models, hybrid models, and data-driven models. The data-driven model requires the largest database, the physical model requires the minimum, while the hybrid is in between. Input parameters of these three models could be various, including temperature, Reynolds number, and some environment parameters, while the output parameters are those typical in heat transfers, such as the heat transfer coefficient, thermal efficiency, temperature differential and heat load (Figure 7).

As early as 1997, ANN had been employed to solve heat transfer problems. Kalogirou et al.[81] predicted the heat loads of rooms with different building materials. The error of results is 9% compared to those of experiments. With the further development of AI, ANN has demonstrated its advantages to deal with the heat transfer. In this section, two representative areas involving heat transfers, the nanofluids and the heat transfer equipment involving solar energy, will be discussed and the suggestion how to select suitable AI algorithms is given.

6.1. Optimization of Heat Transfer Parameters of Nanofluids. Nanofluids are generated by dispersing metal or nonmetal nanoparticles into water, alcohol, or other heat transfer mediums. Owing to their good stability and strong thermal conductivity, they are important in the field of heat transfers. Hemmat et al.[82] constructed a NN with one hidden layer containing four neurons and used experimental data of the viscosity of TiO_2/water nanofluids to train this model. The regression coefficient was 0.9998, and the prediction error of the nanofluid viscosity was only 0.5%. Besides the viscosity, they also predicted the thermal conductivity of nanofluids by using the neural network.[83] It is disclosed that the thermal conductivity of Al_2O_3/ethylene glycol nanofluids was proportional to the volume fraction of nanoparticles. The Levenberg algorithm was employed for the training through the training set from experimental data. The maximum prediction error was about 1.3%. Rostamian et al.[84] constructed a neural network to analyze a complex aqueous nanofluid of CuO-SWcnts/EG. The prediction error was less than 0.544%, and the average error was only 0.0438%. Toghraie et al.[85] proposed an optimal neural network structure generation algorithm, and used only 42 groups of experimental samples to train the model to predict the viscosity of Ag/ethanol nanofluids. Compared to ANN, the maximum error of this algorithm was large, but the mean square error (MSE) was only 0.12%, which means that the new algorithm was still valuable and further study is needed.

Besides single physical fields, ANN could also be applied to multiple physical fields, and the calculation accuracy is quite good. To predict the thermal conductivity of Fe_3O_4/water nanofluids under various magnetic fields, Khosravi and Malekan[86] developed four AI methods: a multilayer feedforward neural network, a grouping method of data processing neural network, a support vector regression model, and an adaptive neural fuzzy inference system. The volume fraction of nanoparticles, the magnetic field intensity, and the magnetic field frequency, etc., were the input parameters of these models, and the heat conduction coefficient was the output parameter. From results, all the four models could be employed to predict the heat conduction coefficient, but the accuracy of the feedforward neural network is the best.

Other AI algorithms, such as RBF neural network, group method of data handling (GMDH), and generalized regression neural network (GRNN) have also been applied in this field. Barati-harooni and Najafi-Marghmaleki[87] employed a RBF neural network and ANN to predict the viscosity of nanofluids, respectively, and found out that the accuracy of the former is lower than that of ANN. Tashrouz et al.[88] combined GMDH with ANN to predict the viscosity of nine types of nanofluids. The regression coefficient of this new model was 0.9978, and the average error was 2.14%, which meant that this model was more

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

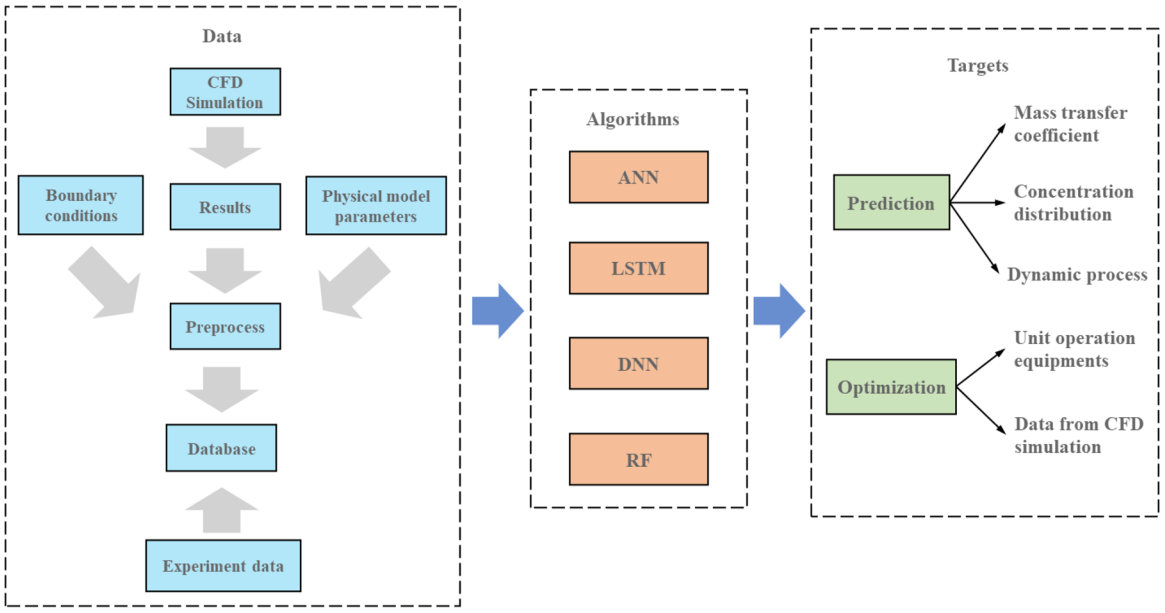


Figure 8. AI models and parameters in the mass transfer.

accurate than the traditional ANN model. Selimefendigil and Oztop[89] constructed the radial basis network, feed-forward network, and generalized regression neural network to predict the heat transfer coefficient. The finite volume method was employed to generate the training set. The heat transfer performance of the system is proportional to the volume fraction of nanoparticles, and the generalized regression neural network has a better performance than the first two.

6.2. Optimization of Solar Heat Transfer Equipment. Solar energy, as an important renewable energy, has been widely studied in the 21st century due to the increasing energy demand. The conversion of solar energy into heat is the key point of solar energy utilization, and it is particularly important to design a solar-thermal conversion device with a high efficiency to improve the energy conversion rate. The successful application of AI in nanofluids shows its applicability in heat transfers, and it might be employed to optimize the solar energy conversion. Some works have been done to analyze the solar collector, solar heater, solar distiller, and other equipment by using ANN.

Tomy et al.[90] studied the flat-panel solar collector through ANN. The input parameters were the solar radiation heat flow, mass flow rate, and inlet temperature. The range of the solar radiation heat flow was 900 W/m^2 ~ 1000 W/m^2, and Reynolds numbers were from 5000 to 25 000. The output parameters were the heat transfer coefficient and the thermal efficiency. The thermal efficiency was predicted by the model, and the error is only 2% compared to the experimental results.

Ghritlahre and Prasad[91] constructed an ANN to predict the thermal performance of a solar air heater with an unidirectional flow porous bed. The input parameters of ANN were the ambient temperature, inlet temperature, mass flow rate, and solar radiation intensity, while the outputs were the air temperature difference, air heat gain, and thermal efficiency. In this model, four training functions (TRAINCGP, TRAINSCG, TRAINLM, and TRAINOSS) were employed to train the feedforward learning process, respectively. In this work, TRAINLM has better training results than other training functions and it could be employed in relevant area.

Mashaly and Alazba[92] proposed MLP neural network and multiple linear regression (MLR) neural network to calculate the instantaneous thermal efficiency of the solar distillers. MLP had nine variables as the inputs, and the output layer had three parameters the same for MLP and MLR. By comparing the results of two models, it is presented that the average determination coefficient of MLP was 11.23%, higher than that of MLR, but the root mean squared error (RMSE) of MLP was much lower. Generally speaking, MLP is a model with a high-precision to predict the instantaneous thermal efficiency of the solar distiller.

These new technologies could calculate the parameters which are difficult to obtain through traditional methods. Among these algorithms, ANN is powerful in the construction of the data-driven model and has performed well. Nevertheless, as the data-driven model could not disclose the hidden mechanisms, it is necessary to develop appropriate algorithms which take physical mechanisms into account in the future.

7. APPLICATION OF AI IN MASS TRANSFER

In chemical industries, almost all the unit operations, such as the absorption, adsorption, distillation, rectification, extraction, and drying, involve mass transfers. How to simulate the mass transfer and calculate the relevant parameters accurately has been concerned more and more and become a big problem in CFD.

The successful application of AI algorithms in the momentum transfer (fluid dynamics) and the heat transfer makes it possible to apply them in mass transfers. In 2000, a preliminary study on the mass transfer through ANN was conducted. Alvarez et al.[93] built an ANN model to predict the mass transfer coefficient of Newtonian and non-Newtonian fluids in a bubble column. The error between the predicted and experimental result was only 1%, and the computation speed was 50 times that of CFD. The database in mass transfers is generally produced by experiments and CFD, as shown in Figure 8. As for CFD data, which include the boundary conditions, calculation results and model parameters, are always preprocessed to reduce the errors in database. Experimental data are usually employed as the part of the test set to evaluate the generalization ability of models.

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

Similar to the heat transfer, SL algorithms, such as ANN, LSTM, DNN, and RF, are effective in mass transfers because the training data are easy to label and analyze. On the basis of these data, high-precision models could be built through these AI algorithms to solve problems of prediction and optimization. Currently, the mass transfer coefficient, concentration distribution, and dynamic process are three parameters which have been successfully predicted by these models. In addition, some parameters of CFD models and equipment of unit operations have also been optimized by these techniques. In this section, three representative areas, the prediction of mass transfer coefficient, calculation of concentration distribution and simulation of mass transfer are discussed, and the algorithms involved in each area will be analyzed.

7.1. Prediction of Mass Transfer Coefficients. The mass transfer coefficient is defined to evaluate mass fluxes, and it is convenient to show the transfer capability of the component in the mixture. Thus, it is of great significance to build a high-precision model to calculate the mass transfer coefficient. Generally, mass transfer coefficients are obtained through experiments or CFD calculations. Currently, the neural network has been applied to predict them in different systems.

Saha[94] developed a model of ANN based on RBF to predict the mass transfer coefficient of the air in the gas−liquid system in a rotating packed bed. By inputting flow rates of the liquid and gas, and the rotor speed into the network, the mass transfer coefficient of the air could be calculated. Through their work, the fitting of the multidimensional data was realized. The error of the traditional simulation method was about 20−30%, while the error of ANN was only 5%.

ElShazly[95] predicted the mass transfer coefficient of the liquid phase in a stirring tank by constructing a model of ANN based on the Levenberg-Marquardt back-propagation optimization algorithm. The neuron number of each hidden layer in the neural network varied from 1 to 12. The input parameters included the slope angle of the stirring tank, the impeller speed, and the viscosity and density of the solution. Different mass transfer coefficients were obtained by changing the number of hidden neurons, and were compared to those calculated by the conventional mass transfer correlation method. It is evident that the neural network has a better accuracy and a faster speed. Thus, it is feasible to calculate the mass transfer coefficient directly through the neural network without estimating the mass transfer correlation coefficient. In addition, the effects of the number of samples and hidden neurons on the prediction results were analyzed, and the neural network had the best performance when the number of hidden neurons was about 3−7.

The neural network can also predict the mass transfer coefficient accurately under some complex conditions (such as a process with chemical reactions). Liu et al.[96] used Higee technology to enhance the mass transfer of the ozone oxidation absorption process, and established a model of ANN to predict the mass transfer coefficient of the ozone absorption. The neural network had three hidden layers, each with 15, 20, and 25 neurons. The data set was labeled as the gas Reynolds number, liquid Reynolds number, liquid Weber number, and liquid Froude number, and the output parameters were the absorption coefficient and mass transfer coefficient of the ozone. To improve the model accuracy, the random grid search method was used to optimize the superparameters of the model. Finally, 50 000 samples were employed to train the model based on the optimized hyperparameters. The regression coefficient was 0.9896 and the RMS error was only 0.01801. Thus, combining ANN with the random grid search method is an effective technology which could generate more reasonable results than traditional methods.

7.2. Calculation of the Concentration Distribution. The concentration distribution in a system is also very important in the problems involving the mass transfer. Khazaei et al.[97] proposed a model of MLP neural network to predict the concentration change of the carbon dioxide in a room. The inputs of the network were the temperature and the relative humidity. Based on Levenberg−Marquardt back-propagation algorithm, the model was trained by the open loop, closed loop, and multistep prediction, respectively. From results, the neural network trained by the five-step prediction method had the highest accuracy, with a difference of only 17 ppm between the predicted and actual value. Although the open-loop training method had a low accuracy in global results, it was still worth studying further due to its simple modeling process, fast training speed, and high accuracy in local results. Kim et al.[98] constructed a feed-forward neural network (FNN) model and a LSTM-RNN coupling model based on the concentration data (collected from the sensor) of harmful substances to predict the location of the leakage source. This work presented that the LSTM-RNN had a higher accuracy than the FNN model. Then, they simulated the leak source through CFD, and trained the LSTM-RNN with the calculated data. When the test set contained only the simulation data, the prediction accuracy of the model was about 97.08%. However, the accuracy could reach 99.96% when the test set included both the simulated and measured data. This work confirmed that the neural network performed well in the concentration prediction.

Besides ANN, other AI algorithms such as DNN and RF have also been successfully applied in the concentration prediction. Cho et al.[99] developed a DNN model and a RF model, and trained them with the calculation data of the chemical leakage though CFD. There were 25 hidden layers in the DNN model and 100 decision trees in the RF model. The data set included the concentration distribution around the leakage location and the concentration evolution. The prediction accuracy of DNN and the RF model was 75.43% and 86.33%, respectively. They could predict not only the spacial variation of the concentration, but also the temporal change. Thus, this technology could be applicable to the real-time diagnosis of the chemical leakage sources, and are of great significance for the chemical emergency management.

7.3. Simulation of Mass Transfer Processes. In addition to the prediction and optimization of some parameters in the mass transfer, the neural network could also be employed to build a high-precision model to simulate the entire mass transfer process. D'Arisbo et al.[100] constructed two models, a phenomenological model based on the law of mass transfers and a mixed model based on ANN, to simulate the ion exchange in a fixed bed. The total concentration of the liquid phase, the equivalent fraction of the liquid phase, and the liquid phase sodium ion were input into ANN as the input parameters, while the equivalent fractions of the solid phase and the solid phase sodium ion were the outputs. Both models could simulate the ion exchange dynamics in the fixed bed accurately, but the average calculation time of the mixed model was only 18% of the phenomenological model. Thus, ANN is efficient to describe the mass transfer processes.

Yurtsever et al.[101] developed a model of ANN to predict and simulate the adsorption and kinetics of valonia tannin resin (VTR) to chromium ions in an aqueous solution. The input

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

Table 1. Application Summary of AI Algorithms in CFD

application areas	authors	input parameters	output parameters	algorithms
aerodynamic model	Chen and Agarwal[2]	two points on the edge of an airfoil	drag coefficient, lift coefficient	GA and ANN
	Mannarino and Mantegazza[9]	pitch parameters and dive parameters	dimensionless load coefficient	RNN
turbulence model	Zhang et al.[14]	300 sets of multidimensional simulation results	RANS product parameter	ANN and multiscale GP regression
	Cheung et al.[16]	seven sets of definite and uncertain parameters	turbulent boundary layer shear stress	Bayesian uncertainty analysis
specific flows	Safarzadeh et al.[24]	three sets of dimensionless velocity parameters and one set of dimensionless length	shear reynolds stress	DE-MLP and DE-RBF
	Roshani and Feghhi[28]	two sets of parameters that the detector outputs	oil−water ratio	MLP(ANN)
	He and Tafti[30]	relative adjacent particle locations, Reynolds Numbers, and porosity	drag	ANN
heat transfer	Khosravi and Malekan[44]	the volume fraction of nanoparticles, magnetic field intensity, magnetic field frequency, Reynolds number and dimensionless distance of tube	thermal conductivity coefficient	MLFNN, ANN, SVM, ANFIS
	Ghritlahre and Prasad[49]	ambient temperature, inlet temperature, mass flow rate, and solar radiation intensity	air temperature difference, air gain heat and thermal efficiency	ANN
mass transfer	Saha[52]	liquid flow, gas flow and rotor speed	gas phase mass transfer coefficient	RBF-ANN
	Khazaei et al.[55]	relative humidity and temperature	CO2 concentration	LM-ANN

layer contained the initial PH, temperature, agitation rate, particle size, contact time, *etc.*; the output layer was adsorption rate, and the structure of the hidden layer could be changed in order to find an optimal model. When the neuron numbers in the hidden layer were 2 and 25, the prediction accuracy was the highest, the correlation coefficient was as high as 0.9997, and the mean square error was only 6.51×10^{-6}. The chromium ion adsorption could be simulated effectively by ANN. Compared to the traditional isothermal model, this neural network had a better adjustment to the balance data of the adsorption. Liu et al.[102] established a ANN model to simulate the biosorption of a rotating packed bed. The liquid Reynolds number, the average gravity factor, the contact time, the ratio of the particle size and bed depth, etc., were chosen as the inputs, the ratio of the biosorption capacity at some time to the maximum biosorption capacity was the output parameter. The number of hidden layers was three, and the neuron numbers in each layer were 6, 15, and 14, respectively. The neural network was trained by the back-propagation (BP) and Levenberg-Marquart (LM) algorithm, respectively. The ANN model based on the BP algorithm had the best results with a correlation coefficient of 0.996 and a mean square error of 0.00904%. Compared to the complex CFD model and the inaccurate semiempirical model in the biosorption simulation, ANN modeling is simple and has a shorter computation time, which makes it promising to model complex systems.

Table 1 summarizes the application of AI in the five fields mentioned above and Table 2 presents the advantages and disadvantage of these algorithms. Among them, as ANN could deal with nonlinear relations, is faster in computation, and has a strong generalization ability, it has been applied widely in many various fields. However, for some special situations, such as the optimization of physical models and the solving of PDEs, ANN is not appropriate sometimes. By contrast, other AI algorithms, such as Bayesian neural network and SVM, might be more effective. There is no doubt that developing algorithms with a higher accuracy and faster speed is the focus of the research in the future. It is also the purpose of the application of AI in the field of CFD.

8. CONCLUSIONS AND PROSPECTS

This paper discusses the application of AI algorithms in different fields of CFD. According to our best knowledge, the papers published in the past 10 years about these topics have been collected, and the paper numbers of each year are shown in Figure 9. Before 2015, the number of published papers was relatively small, and the contents were mainly about the aerodynamic model and the turbulence model. After 2015, the paper numbers increased greatly and maintained a high growth rate until today.

In recent papers, the algorithm employed most widely, about 85%, is the neural network, as shown in Figure 10. Within this 85%, ANN is the most popular and accounts for about 73%, due to its high accuracy of classification, its strong ability of parallel distribution processing, and the ability to approximate nonlinear relations. Other algorithms, such as RNN, DNN, and LSTM, also have some certain applications in CFD. In addition to the neural network, SVM is the second most popular technology. It has a strong generalization ability, requires small databases and could avoid the structure selection and the local minimum points which are problems of the neural network. What is more, RF, NB, and other algorithms which are not sensitive to data and require less parameters have unique advantages to deal with some specific problems.

In this paper, five aspects of CFD have been discussed: the optimization of aerodynamic and turbulence models, the simulation and prediction of complex flow fields, the modeling and verification of heat and mass transfers. Table 1 summarizes some representative algorithms and input−output parameters. Table 2 shows AI algorithms employed in the five fields and their advantages and disadvantages, which is helpful for readers to choose the appropriate one. Currently, SL technology is the most popular as data from CFD and experiments generally have specific physical background and are easy to label.

The general application modes of AI algorithms in the field of CFD is shown in Figure 11. The data from calculations and experiments are collected and preprocessed to build the database. According to the problem to be solved and the

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

Table 2. Advantages and Disadvantages of Various AI Algorithms in CFD

AI algorithms		application areas	advantages	disadvantages
NN	ANN	aerodynamic model	(1) capability to solve complex nonlinear problems	(1) large data set
		turbulence model	(2) high accuracy of parameters (RANS, LES, DNS) prediction	(2) long training time
		specific flows	(3) strong robustness and fault tolerance	(3) cannot observe learning process
		heat transfer		(4) the output is hard to interpret
		mass transfer		
	DNN	specific flows	(1) higher precision than traditional NN (ANN)	(1) too many neurons result in too many DNN parameters
		mass transfer	(2) solving the problem of gradient disappearance	(2) difficult to verify the correctness of the model
	CNN	specific flows	strong feature detection capability	(1) the full join mode is redundant and inefficient
				(2) less feature understanding
	RNN	mass transfer	(1) few model parameters	(1) gradient extinction occurs in the timeline
			(2) capability to training time-sequence data set	(2) the problem of long-term dependence cannot be solved
	LSTM	aerodynamic model	(1) sequence modeling capability	(1) difficult to parallelization
		specific flows	(2) using gate mechanism can solve gradient explosion and gradient disappearance	(2) poor performance and long training time (four times the number of parameters than RNN)
	GRNN	heat transfer	(1) fast convergence	high computation complexity (ach test sample is calculated against the entire training sample)
			(2) good nonlinear approximation performance (based on radial basis network)	
	Bayesian Networks	turbulence model	few parameters required	having error rate in classification decision
	RF	turbulence model	(1) an integration algorithm (high accuracy)	1) long training time when there are many decision trees
		specific flows	(2) default values can be handled (as a separate class) without additional processing	2) a black box model
		mass transfer	(3) fast training speed	
			(4) strong capability to parallelize	
	GA	aerodynamic model	(1) simple process (using evaluation to search QoI)	(1) the training depend on the initial population (mesh data)
	SVM	aerodynamic model	(1) capability to solve high dimensional (many features) problems	(1) Sensitive to missing data
		specific flows	(2) small data set	(2) kernel function should be selected carefully
			(3) capability to avoid the problem of local minimum point	(3) poor performance in dealing with nonlinear problems (multiphase flow with particles)
	kernel regression	turbulence model	having strong robustness to outliers	unconstrained and easy to overfit
	NB	specific flows	a traditional classification algorithm with stable performance and high accuracy	model assumptions (attributes are independent) are not valid in practical applications

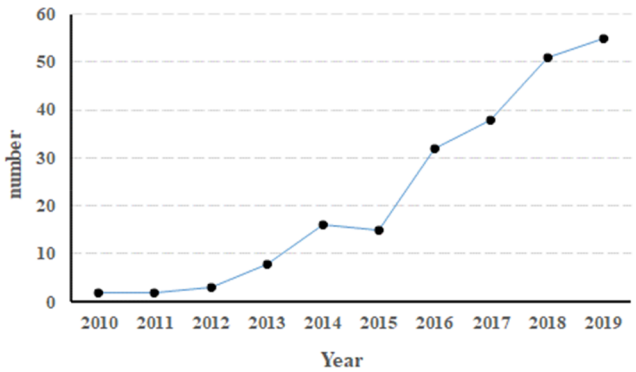


Figure 9. Numbers of the publications in the past decade.

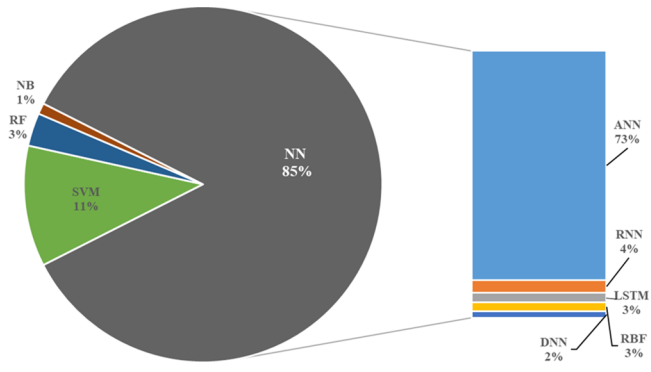


Figure 10. Percentage of publications on various AI algorithms.

characteristics of the database, different AI algorithms are selected to generate different models. Normally, there are three kinds of CFD models based on AI algorithms. One is so-called physical model, which is to disclose or optimize the existing physical mechanisms. For example, the neural network is employed to solve PDEs involved in the physical process so as to optimize the parameters of the physical model. It has practical physical meanings and is helpful to optimize the existing models. Although physical models are usually of high precision, their computation costs are also quite high. The next is the data-driven model, which need not solve complex mathematical formulas. Instead, it establishes the input−output relations, and trains the model by selecting appropriate algorithms, in order to obtain high-quality models suitable for the actual situations. Generally, data-driven models have simple generation processes, high calculation speeds, and reliable outcomes. The last is the

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

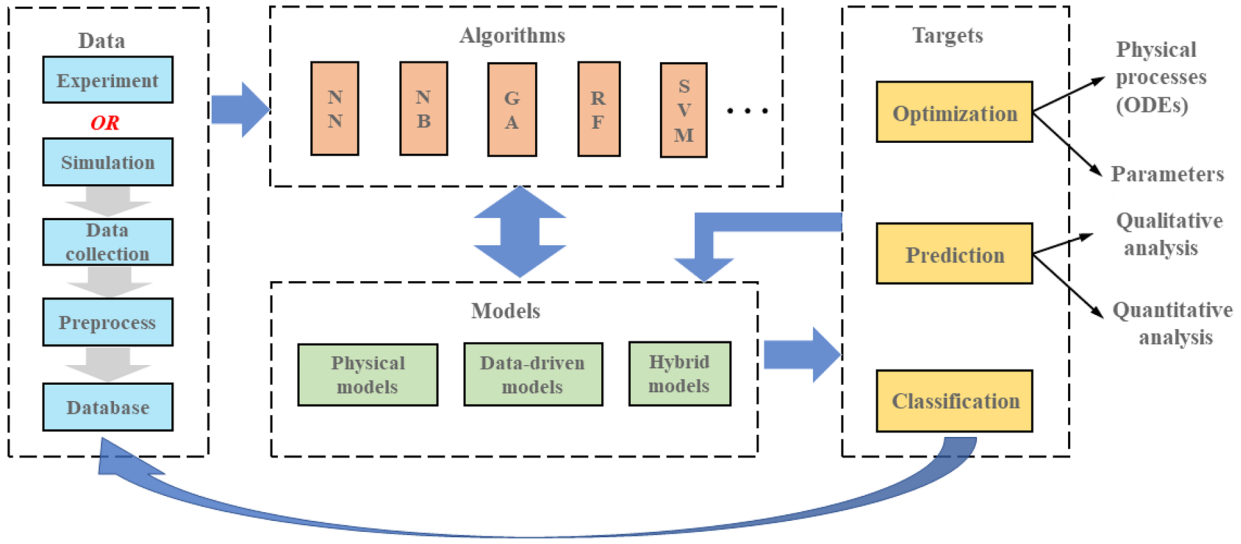


Figure 11. Application status of AI algorithm in CFD.

hybrid model, which not only takes into account certain physical mechanisms but also requires a lot of data. According to the applications, the above models are mainly employed to solve optimization, prediction, and classification problems. On the basis of the results of optimization and prediction, models could be further optimized to achieve the expected goals. By classifying the data, the stability of the training set could be enhanced, and the accuracy of the model could be improved.

Nowadays, the data-driven model is the commonest in the application of AI in CFD, since a large amount of data are generated in CFD and could be employed for the establishment of neural networks. Although it has a high accuracy and wide application, this model cannot obtain the hidden physical mechanism to essentially solve the CFD problems. Other algorithms, such as SVM and RF, which contain mathematical theories and require less data, are mostly employed for the optimization of physical models and the exploration of mechanisms. They are common in the application of the turbulence model.

Although the model combining AI with CFD has a high accuracy and wide application, it still has some problems in the following aspects. First, data sources are not broad enough. The data-driven model could be quite accurate to solve some specific problems, such as heat and mass transfer. However, as most data are from CFD and experiments, the model is prone to overfitting and is weak in generalization. Additionally, more data should be added during the data collection and preprocessing, and the uncertainty of the data when training the model needs to be considered. Second, the goal of learning is unclear. There is no criterion for the input and output features of models for specific problems, and most of them are selected only based on experience. Moreover, the best choice of how to generalize the model to solve other problems is still uncertain. In the future, it is important to select appropriate input and output features and to modify the data set according to the activation function and physical mechanisms. Finally, there are no clear criteria how to balance data and physical mechanisms in physical models and hybrid models in between. On the one hand, AI algorithms are employed to reconstruct the numerical solution of N−S equations. For example, DNN is effective to solve PDEs and optimize the numerical solution, but its calculation cost is high. On the other hand, although the pure use of data could also obtain accurate prediction models, the relationship between data is difficult to explain. Thus, adding mathematical and physical constraints on the basis of the data-driven model is a major focus of future research.

AUTHOR INFORMATION

Corresponding Author

Jingtao Wang − *School of Chemical Engineering and Technology, Tianjin University, Tianjin 300072, P. R. China; Tianjin Key Laboratory of Chemical Process Safety and Equipment Technology, Tianjin 300072, P. R. China;* orcid.org/0000-0002-1712-7898; Email: wjingtao928@tju.edu.cn

Author

Bo Wang − *School of Chemical Engineering and Technology, Tianjin University, Tianjin 300072, P. R. China*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.iecr.0c05045

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was supported by National Key R&D Program of China (2019YFC1905805) and National Natural Science Foundation of China (22078229 and 21576185). In addition, we are grateful to Prof. Jing Guan for her valuable communication and help.

ABBREVIATION

AI = artificial intelligence
CFD = computational fluid dynamics
ANN = artificial neural network
RNN = recursive neural network
SVM = support vector machine
NB = naive Bayes
NN = neural network
RF = random forest
DT = decision tree
DNN = deep neural network
CNN = convolutional neural network

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

LSTM = long short-term memory
PCA = principal component analysis
GANs = generative adversarial nets
ML = machine learning
SBO = surrogate based optimization
SAO = surrogate aided optimization
GA = genetic algorithm
GA-HFCDN = GA with hierarchical fair competition dynamic-niche
RBF = radial basis function
PDEs = partial differential equations
ROMs = reduced-order models
CTRNN = continuous time recurrent neural network
RANS = Reynolds-averaged Navier−Stokes
DNS = direct numerical simulation method
LES = large eddy simulation
N−S = Navier−Stokes
S-A = Spalart-Allmaras
QoI = quantities of interest
ODE = ordinary differential equation
SGS = subgrid scale
VOF = volume of fluid
DEM = discrete element method
DE = differential evolution
MANN = mean artificial neural network
MLP = multilayer perceptor
PRS = particle resolved simulations
SL = supervised learning
MSE = mean square error
GMDH = group method of data handling
GRNN = generalized regression neural network
MLR = multiple linear regression
RMSE = root mean squared error
FNN = feed-forward neural network
VTR = valonia tannin resin
BP = back-propagation
LM = Levenberg-Marquart
PDE = partial differential equation

REFERENCES

(1) Brunton, S. L.; Noack, B. R.; Koumoutsakos, P. Machine Learning for Fluid Mechanics. In *Annual Review of Fluid Mechanics*; Davis, S. H., Moin, P., Eds.; 2020; Vol. *52*, pp 477−508.

(2) Zhou, Z. H. A brief introduction to weakly supervised learning. *Natl. Sci. Rev.* **2018**, *5* (1), 44−53.

(3) Yang, H. F.; Lin, K.; Chen, C. S. Supervised Learning of Semantics-Preserving Hash via Deep Convolutional Neural Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40* (2), 437−451.

(4) Miyato, T.; Maeda, S. I.; Koyama, M.; Ishii, S. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE. Trans. Pattern Anal. Mach. Intell.* **2019**, *41* (8), 1979−1993.

(5) Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks.* **2015**, *61*, 85−117.

(6) Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270* (2), 654−669.

(7) Chan, T. H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. PCANet: A Simple Deep Learning Baseline for Image Classification? *IEEE Trans. Image Process.* **2015**, *24* (12), 5017−5032.

(8) Quan, W.; Wang, K.; Yan, D. M.; Zhang, X. Distinguishing Between Natural and Computer-Generated Images Using Convolutional Neural Networks. *IEEE T. Inf. Foren. Sec.* **2018**, *13* (11), 2772−2787.

(9) Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; Chen, T. Recent advances in convolutional neural networks. *Pattern Recogn.* **2018**, *77*, 354−377.

(10) Graves, A.; Mohamed, Ar.; Hinton, G. Speech Recognition with Deep Recurrent Neural Networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada*, May 26−31, 2013; IEEE: New York, 2013.

(11) Donahue, J.; Hendricks, L. A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T.; Long-term Recurrent Convolutional Networks for Visual Recognition and Description. *2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA*, Jun 07−12, 2015; IEEE: New York, 2015.

(12) Huang, G.; Song, S.; Gupta, J. N. D.; Wu, C. Semi-Supervised and Unsupervised Extreme Learning Machines. *IEEE Trans. Cybern.* **2014**, *44* (12), 2405−2417.

(13) Ahonen, T.; Hadid, A.; Pietikainen, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28* (12), 2037−2041.

(14) Zhao, H.; Zheng, J.; Deng, W.; Song, Y. Semi-Supervised Broad Learning System Based on Manifold Regularization and Broad Network. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67* (3), 983−994.

(15) Sheikhpour, R.; Sarram, M. A.; Gharaghani, S.; Chahooki, M. A. Z. A Survey on semi-supervised feature selection methods. *Pattern Recogn.* **2017**, *64*, 141−158.

(16) Ashfaq, R. A. R.; Wang, X. Z.; Huang, J. Z.; Abbas, H.; He, Y. L. Fuzziness based semi-supervised learning approach for intrusion detection system. *Inf. Sci.* **2017**, *378*, 484−497.

(17) Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; Shi, W. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, U.S.*, Jul 21−26, 2017; IEEE: New York, 2017.

(18) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521* (7553), 436−444.

(19) Hicks, R. M.; Murman, E. M.; Vanderplaats, G. N. An assessment of airfoil design by numerical optimization. *NASA Report*, 1974; No. 19740020369.

(20) Chen, X.; Agarwal, R. Optimization of flatback airfoils for wind-turbine blades using a genetic algorithm. *J. Aircr.* **2012**, *49* (2), 622−629.

(21) Su, W.; Zuo, Y.; Gao, Z. Preliminary aerodynamic shape optimization using genetic algorithm and neural network. *11th AIAA/ISSMO multidisciplinary analysis and optimization conference, Portsmouth, VA, USA*, Sept 6−8, 2006; American Institute of Aeronautics and Astronautics, Inc., 2006.

(22) Mengistu, T.; Ghaly, W. Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models. *Optim. Eng.* **2008**, *9* (3), 239−255.

(23) Zhang, B.; Wang, T.; Gu, Cg.; Shu, X. W. An integrated blade optimization approach based on parallel ANN and GA with hierarchical fair competition dynamic-niche. *J. Mech. Sci. Technol.* **2011**, *25* (6), 1457.

(24) Giannakoglou, K. C.; Giotis, A. P.; Karakasis, M. K. Low-cost genetic optimization based on inexact pre-evaluations and the sensitivity analysis of design parameters. *Inverse Probl. Eng.* **2001**, *9* (4), 389−412.

(25) Asouti, V.; Kampolis, I. C.; Giannakoglou, K. C. A grid-enabled asynchronous metamodel-assisted evolutionary algorithm for aerodynamic optimization. *Genet. Program Evolvable Mach.* **2009**, *10* (4), 373.

(26) Iuliano, E.; Pérez, E. A. Application of surrogate-based global optimization to aerodynamic design. In *Chapter 2: Adaptive Sampling Strategies for Surrogate-Based Aerodynamic Optimization*; Choi, S. B., Duan, H., Fu, Y., Guardiola, C., Sun, J. Q., Kwon, Y. W., Cavas-Martínez, F., Chaari, F. Eds.; Springer: London, 2016; pp 25−46.

(27) Mannarino, A.; Mantegazza, P. Nonlinear aeroelastic reduced order modeling by recurrent neural networks. *J. Fluids Struct.* **2014**, *48*, 103−121.

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

(28) Skujins, T.; Cesnik, C. E. Reduced-order modeling of unsteady aerodynamics across multiple mach regimes. *J. Aircr.* **2014**, *51* (6), 1681−1704.
(29) Li, K.; Kou, J.; Zhang, W. Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple Mach numbers. *Nonlinear Dyn.* **2019**, *96* (3), 2157−2177.
(30) Milano, M.; Koumoutsakos, P. Neural network modeling for near wall turbulent flow. *J. Comput. Phys.* **2002**, *182* (1), 1−26.
(31) Tracey, B. D.; Duraisamy, K.; Alonso, J. J. A machine learning strategy to assist turbulence model development. *53rd AIAA aerospace sciences meeting, Kissimmee, Florida*, Jan 5−9, 2015; American Institute of Aeronautics and Astronautics, Inc., 2015.
(32) Zhang, Z. J.; Duraisamy, K. Machine learning methods for data-driven turbulence modeling. *22nd AIAA Computational Fluid Dynamics Conference, Dallas, TX*, Jun 22−26, 2015; American Institute of Aeronautics and Astronautics, Inc., 2015.
(33) Tracey, B.; Duraisamy, K.; Alonso, J. Application of supervised learning to quantify uncertainties in turbulence and combustion modeling. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Dallas, TX*, Jan 7−10, 2013; American Institute of Aeronautics and Astronautics, Inc., 2013.
(34) Cheung, S. H.; Oliver, T. A.; Prudencio, E. E.; Prudhomme, S.; Moser, R. D. Bayesian uncertainty analysis with applications to turbulence modeling. *Reliab. Eng. Syst. Saf.* **2011**, *96* (9), 1137−1149.
(35) Oliver, T. A.; Moser, R. D. Bayesian uncertainty quantification applied to RANS turbulence models. *J. Phys. Conf.* **2011**, *318*, 042032.
(36) Portwood, G. D.; Mitra, P. P.; Ribeiro, M. D.; Nguyen, T. M.; Nadiga, B. T.; Saenz, J. A.; Chertkov, M.; Garg, A.; Anandkumar, A.; Dengel, A. Turbulence forecasting via Neural ODE. *arXiv preprint.* **2019**. arXiv:1911.05180 [physics.comp-ph].
(37) Duraisamy, K.; Iaccarino, G.; Xiao, H. Turbulence Modeling in the Age of Data. *Annu. Rev. Fluid Mech.* **2019**, *51* (1), 357−377.
(38) Zhu, L.; Zhang, W.; Kou, J.; Liu, Y. Machine learning methods for turbulence modeling in subsonic flows around airfoils. *Phys. Fluids* **2019**, *31* (1), 015105.
(39) Wang, J. X.; Wu, J. L.; Xiao, H. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. *Phys. Rev. Fluids.* **2017**, *2* (3), 013403.
(40) Lee, M.; Moser, R. D. Direct numerical simulation of turbulent channel flow up to Re-tau approximate to 5200. *J. Fluid Mech.* **2015**, *774*, 395−415.
(41) Khan, N. B.; Ibrahim, Z. Numerical investigation of vortex-induced vibration of an elastically mounted circular cylinder with One-degree of freedom at high Reynolds number using different turbulent models. *P. I. Mech. Eng. M-j. Eng.* **2019**, *233* (2), 443−453.
(42) Tabor, G. R.; Baba-Ahmadi, M. H. Inlet conditions for large eddy simulation: A review. *Comput. Fluids* **2010**, *39* (4), 553−567.
(43) Nicoud, F.; Toda, H. B.; Cabrit, O.; Bose, S.; Lee, J. Using singular values to build a subgrid-scale model for large eddy simulations. *Phys. Fluids* **2011**, *23* (8), 085106.
(44) Maronga, B.; Gryschka, M.; Heinze, R.; Hoffmann, F.; Kanani-Suehring, F.; Keck, M.; Ketelsen, K.; Letzel, M. O.; Suehring, M.; Raasch, S. The Parallelized Large-Eddy Simulation Model (PALM) version 4.0 for atmospheric and oceanic flows: model formulation, recent developments, and future perspectives. *Geosci. Model Dev.* **2015**, *8* (8), 2515−2551.
(45) Long, Y.; Long, X.; Ji, B.; Xing, T. Verification and validation of Large Eddy Simulation of attached cavitating flow around a Clark-Y hydrofoil. *Int. J. Multiphase Flow* **2019**, *115*, 93−107.
(46) Wang, Z.; Luo, K.; Li, D.; Tan, J.; Fan, J. Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation. *Phys. Fluids* **2018**, *30* (12), 125101.
(47) Maulik, R.; San, O.; Jacob, J. D. Spatiotemporally dynamic implicit large eddy simulation using machine learning classifiers. *Phys. D* **2020**, *406*, 132409.
(48) Huang, X. L. D.; Yang, X. I. A.; Kunz, R. F. Wall-modeled large-eddy simulations of spanwise rotating turbulent channels-Comparing a physics-based approach and a data-based approach. *Phys. Fluids* **2019**, *31* (12), 125105.
(49) Maulik, R.; San, O.; Jacob, J. D.; Crick, C. Sub-grid scale model classification and blending through deep learning. *J. Fluid Mech.* **2019**, *870*, 784−812.
(50) Yang, X. I. A.; Zafar, S.; Wang, J. X.; Xiao, H. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Phys. Rev. Fluids.* **2019**, *4* (3), 034602.
(51) Erichson, N. B.; Muehlebach, M.; Mahoney, M. W. Physics-informed Autoencoders for Lyapunov-stable Fluid Flow Prediction. *arXiv preprint.* **2019**. arXiv:1905.10866 [physics.comp-ph].
(52) Erichson, N. B.; Mathelin, L.; Yao, Z.; Brunton, S. L.; Mahoney, M. W.; Kutz, J. N. Shallow learning for fluid flow reconstruction with limited sensors and limited data. *arXiv preprint.* **2019**. arXiv:1902.07358 [physics.comp-ph].
(53) Lui, H. F. S.; Wolf, W. R. Construction of reduced-order models for fluid flows using deep feedforward neural networks. *J. Fluid Mech.* **2019**, *872*, 963−994.
(54) Safarzadeh, A.; Zaji, A. H.; Bonakdari, H. 3D flow simulation of straight groynes using hybrid DE-based artificial intelligence methods. *Soft comput.* **2019**, *23* (11), 3757−3777.
(55) Reddy, P. B. A.; Das, R. Estimation of MHD boundary layer slip flow over a permeable stretching cylinder in the presence of chemical reaction through numerical and artificial neural network modeling. *Eng. Sci. Technol.* **2016**, *19* (3), 1108−1116.
(56) Ma, M.; Lu, J.; Tryggvason, G. Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels. *Int. J. Multiphase Flow* **2016**, *85*, 336−347.
(57) Ma, M.; Lu, J. C.; Tryggvason, G. Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system. *Phys. Fluids* **2015**, *27* (9), 092101.
(58) Roshani, G. H.; Feghhi, S. A. H.; Mahmoudi-Aznaveh, A.; Nazemi, E.; Adineh-Vand, A. Precise volume fraction prediction in oil-water-gas multiphase flows by means of gamma-ray attenuation and artificial neural networks using one detector. *Measurement.* **2014**, *51*, 34−41.
(59) Pirnia, P.; Duhaime, F.; Ethier, Y.; Dube, J. S. Drag Force Calculations in Polydisperse DEM Simulations with the Coarse-Grid Method: Influence of the Weighting Method and Improved Predictions Through Artificial Neural Networks. *Transp. Porous Media* **2019**, *129* (3), 837−853.
(60) He, L.; Tafti, D. K. A supervised machine learning approach for predicting variable drag forces on spherical particles in suspension. *Powder Technol.* **2019**, *345*, 379−389.
(61) Yan, S. N.; He, Y. R.; Tang, T. Q.; Wang, T. Y. Drag coefficient prediction for non-spherical particles in dense gas-solid two-phase flow using artificial neural network. *Powder Technol.* **2019**, *354*, 115−124.
(62) Azad, E. Experimental analysis of thermal performance of solar collectors with different numbers of heat pipes versus a flow-through solar collector. *Renewable Sustainable Energy Rev.* **2018**, *82*, 4320−4325.
(63) Asadzadeh, F.; Esfahany, M. N.; Etesami, N. Natural convective heat transfer of Fe3O4/ethylene glycol nanofluid in electric field. *Int. J. Therm. Sci.* **2012**, *62*, 114−119.
(64) Dalkilic, A. S.; Yalcin, G.; Kucukyildirim, B. O.; Oztuna, S.; Eker, A. A.; Jumpholkul, C.; Nakkaew, S.; Wongwises, S. Experimental study on the thermal conductivity of water-based CNT-SiO2 hybrid nanofluids. *Int. Commun. Heat Mass Transfer* **2018**, *99*, 18−25.
(65) Yuan, B.; Zhang, Y.; Hu, G.; Zhong, W.; Qian, F. Analytical models for heat transfer in the tube bundle of convection section in a steam cracking furnace. *Appl. Therm. Eng.* **2019**, *163*, 113947.
(66) Malekan, M.; Khosravi, A.; Syri, S. Heat transfer modeling of a parabolic trough solar collector with working fluid of Fe3O4 and CuO/Therminol 66 nanofluids under magnetic field. *Appl. Therm. Eng.* **2019**, *163*, 114435.
(67) Chen, S.; Liu, D.; Liu, M.; Xiao, Y.; Gu, H. Numerical simulation and analysis of flow and heat transfer in a 5 × 5 vertical rod bundle with buoyancy effects. *Appl. Therm. Eng.* **2019**, *163*, 114221.
(68) Buzzi, F.; Pucciarelli, A.; Ambrosini, W. On the mechanism of final heat transfer restoration at the transition to gas-like fluid at

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

supercritical pressure: A description by CFD analyses. *Nucl. Eng. Des.* **2019**, *355* (Dec), 110345.
(69) Yadav, A. K.; Chandel, S. S. Solar radiation prediction using Artificial Neural Network techniques: A review. *Renewable Sustainable Energy Rev.* **2014**, *33*, 772−781.
(70) Vafaei, M.; Afrand, M.; Sina, N.; Kalbasi, R.; Sourani, F.; Teimouri, H. Evaluation of thermal conductivity of MgO-MWCNTs/EG hybrid nanofluids based on experimental data by selecting optimal artificial neural networks. *Phys. E* **2017**, *85*, 90−96.
(71) Moradikazerouni, A.; Hajizadeh, A.; Safaei, M. R.; Afrand, M.; Yarmand, H.; Zulkifli, N. W. B. M. Assessment of thermal conductivity enhancement of nano-antifreeze containing single-walled carbon nanotubes: Optimal artificial neural network and curve-fitting. *Phys. A* **2019**, *521*, 138−145.
(72) Mohanraj, M.; Jayaraj, S.; Muraleedharan, C. Applications of artificial neural networks for thermal analysis of heat exchangers - A review. *Int. J. Therm. Sci.* **2015**, *90*, 150−172.
(73) Mohanraj, M.; Jayaraj, S.; Muraleedharan, C. Applications of artificial neural networks for refrigeration, air-conditioning and heat pump systems-A review. *Renewable Sustainable Energy Rev.* **2012**, *16* (2), 1340−1358.
(74) Le Thi, L.; Hoang, N.; Dou, J.; Zhou, J. A Comparative Study of PSO-ANN, GA-ANN, ICA-ANN, and ABC-ANN in Estimating the Heating Load of Buildings' Energy Efficiency for Smart City Planning. *Appl. Sci.* **2019**, *9* (13), 2630.
(75) Karimipour, A.; Bagherzadeh, S. A.; Taghipour, A.; Abdollahi, A.; Safaei, M. R. A novel nonlinear regression model of SVR as a substitute for ANN to predict conductivity of MWCNT-CuO/water hybrid nanofluid based on empirical data. *Phys. A* **2019**, *521*, 89−97.
(76) Jahani, B.; Mohammadi, B. A comparison between the application of empirical and ANN methods for estimation of daily global solar radiation in Iran. *Theor. Appl. Climatol.* **2019**, *137* (1−2), 1257−1269.
(77) Hemmat Esfe, M.; Esfandeh, S.; Saedodin, S.; Rostamian, H. Experimental evaluation, sensitivity analyzation and ANN modeling of thermal conductivity of ZnO-MWCNT/EG-water hybrid nanofluid for engineering applications. *Appl. Therm. Eng.* **2017**, *125*, 673−685.
(78) Bahrami, M.; Akbari, M.; Bagherzadeh, S. A.; Karimipour, A.; Afrand, M.; Goodarzi, M. Develop 24 dissimilar ANNs by suitable architectures & training algorithms via sensitivity analysis to better statistical presentation: Measure MSEs between targets & ANN for Fe-CuO/Eg-Water nanofluid. *Phys. A* **2019**, *519*, 159−168.
(79) Ahmadi, M. H.; Baghban, A.; Sadeghzadeh, M.; Hadipoor, M.; Ghazvini, M. Evolving connectionist approaches to compute thermal conductivity of TiO2/water nanofluid. *Phys. A* **2020**, *540* (Feb 15), 122489.
(80) Afram, A.; Janabi-Sharifi, F.; Fung, A. S.; Raahemifar, K. Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system. *Energy Build.* **2017**, *141*, 96−113.
(81) Kalogirou, S.; Neocleous, C.; Schizas, C. Building heating load estimation using artificial neural networks. In *Proceedings of CLIMA 2000 Conference*; AIVC: Brussels, Belgium, 1997.
(82) Hemmat Esfe, M.; Ahangar, M. R. H.; Rejvani, M.; Toghraie, D.; Hajmohammad, M. H. Designing an artificial neural network to predict dynamic viscosity of aqueous nanofluid of TiO2 using experimental data. *Int. Commun. Heat Mass Transfer* **2016**, *75*, 192−196.
(83) Hemmat Esfe, M.; Rostamian, H.; Toghraie, D.; Yan, W. M. Using artificial neural network to predict thermal conductivity of ethylene glycol with alumina nanoparticle. *J. Therm. Anal. Calorim.* **2016**, *126* (2), 643−648.
(84) Rostamian, S. H.; Biglari, M.; Saedodin, S.; Hemmat Esfe, M. An inspection of thermal conductivity of CuO-SWCNTs hybrid nanofluid versus temperature and concentration using experimental data, ANN modeling and new correlation. *J. Mol. Liq.* **2017**, *231*, 364−369.
(85) Toghraie, D.; Sina, N.; Jolfaei, N. A.; Hajian, M.; Afrand, M. Designing an Artificial Neural Network (ANN) to predict the viscosity of Silver/Ethylene glycol nanofluid at different temperatures and volume fraction of nanoparticles. *Phys. A* **2019**, *534* (Nov 15), 122142.
(86) Khosravi, A.; Malekan, M. Effect of the magnetic field on the heat transfer coefficient of a Fe3O4-water ferrofluid using artificial intelligence and CFD simulation. *Eur. Phys. J. Plus.* **2019**, *134* (3), 18.
(87) Barati-Harooni, A.; Najafi-Marghmaleki, A. An accurate RBF-NN model for estimation of viscosity of nanofluids. *J. Mol. Liq.* **2016**, *224*, 580−588.
(88) Atashrouz, S.; Pazuki, G.; Alimoradi, Y. Estimation of the viscosity of nine nanofluids using a hybrid GMDH-type neural network system. *Fluid Phase Equilib.* **2014**, *372*, 43−48.
(89) Selimefendigil, F.; Oztop, H. F. Estimation of the mixed convection heat transfer of a rotating cylinder in a vented cavity subjected to nanofluid by using generalized neural networks. *Numer. Heat Transfer, Part A* **2014**, *65* (2), 165−185.
(90) Tomy, A. M.; Ahammed, N.; Subathra, M. S. P.; Asirvatham, L. G. Analysing the Performance of a Flat Plate Solar Collector with Silver/Water Nanofluid Using Artificial Neural Network. *In Proceedings of the 6th International Conference on Advances in Computing and Communications, Kochi, India*, Sept 6−8 2016; Mathew, J., DasKrishna, D., Jose, J., Eds.; Elsevier: Amsterdam, 2016; Vol. *93*, pp 33−40.
(91) Ghritlahre, H. K.; Prasad, R. K. Prediction of thermal performance of unidirectional flow porous bed solar air heater with optimal training function using Artificial Neural Network. *In International Conference on Recent Advancement in Air Conditioning and Refrigeration, Bhubaneswar, India*, Nov 10−12, 2016; Choudhury, B. K., Rout, S. K., Muthukumar, P., Sahoo, R. K., Sahoo, S. S., Eds.; Elsevier: Amsterdam, 2016; Vol. *109*, pp 369−376.
(92) Mashaly, A. F.; Alazba, A. A. MLP and MLR models for instantaneous thermal efficiency prediction of solar still under hyper-arid environment. *Comput. Electron. Agric.* **2016**, *122*, 146−155.
(93) Alvarez, E.; Correa, J. M.; Riverol, C.; Navaza, J. M. Model based in neural networks for the prediction of the mass transfer coefficients in bubble columns. Study in Newtonian and Non-Newtonian fluids. *Int. Commun. Heat Mass Transfer* **2000**, *27* (1), 93−98.
(94) Saha, D. Prediction of mass transfer coefficient in rotating bed contactor (Higee) using artificial neural network. *Heat Mass Transfer* **2009**, *45* (4), 451−457.
(95) ElShazly, Y. M. S. The use of neural network to estimate mass transfer coefficient from the bottom of agitated vessel. *Heat Mass Transfer* **2015**, *51* (4), 465−475.
(96) Liu, T. R.; Liu, Y. R.; Wang, D.; Li, Y. W.; Shao, L. Artificial neural network modeling on the prediction of mass transfer coefficient for ozone absorption in RPB. *Chem. Eng. Res. Des.* **2019**, *152*, 38−47.
(97) Khazaei, B.; Shiehbeigi, A.; Kani, A. Modeling indoor air carbon dioxide concentration using artificial neural network. *Int. J. Environ. Sci. Technol.* **2019**, *16* (2), 729−736.
(98) Kim, H.; Park, M.; Kim, C. W.; Shin, D. Source localization for hazardous material release in an outdoor chemical plant via a combination of LSTM-RNN and CFD simulation. *Comput. Chem. Eng.* **2019**, *125*, 476−489.
(99) Cho, J.; Kim, H.; Gebreselassie, A. L.; Shin, D. Deep neural network and random forest classifier for source tracking of chemical leaks using fence monitoring data. *J. Loss Prev. Process Ind.* **2018**, *56*, 548−558.
(100) D'Arisbo, T.; Silva, E. A.; Borba, C. E.; Ostroski, I. C.; Arroyo, P. A.; Barros, M. Comparison of Phenomenological and Hybrid Models in the Description of the Ion Exchange Process in a Fixed-Bed Column. *Sep. Sci. Technol.* **2013**, *48* (7), 1102−1110.
(101) Yurtsever, U.; Yurtsever, M.; Sengil, I. A.; Yilmazcoban, N. K. Fast artificial neural network (FANN) modeling of Cd(II) ions removal by valonia resin. *Desalin. Water Treat.* **2015**, *56* (1), 83−96.
(102) Liu, Z. W.; Liang, F. N.; Liu, Y. Z. Artificial neural network modeling of biosorption process using agricultural wastes in a rotating packed bed. *Appl. Therm. Eng.* **2018**, *140*, 95−101.
(103) Wang, Z.; Xiao, D.; Fang, F.; Govindan, R.; Pain, C. C.; Guo, Y. Model identification of reduced order fluid dynamics systems using deep learning. *Int. J. Numer. Methods Fluids* **2018**, *86* (4), 255−268.
(104) Mohan, A. T.; Gaitonde, D. V. A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790

using LSTM Neural Networks. *arXiv preprint.* **2018**. arXiv:1804.09269v1 [physics.comp-ph].

https://dx.doi.org/10.1021/acs.iecr.0c05045
Ind. Eng. Chem. Res. 2021, 60, 2772−2790