Building and Environment 206 (2021) 108315

Contents lists available at ScienceDirect

Building and Environment

journal homepage: www.elsevier.com/locate/buildenv

Deep learning to replace, improve, or aid CFD analysis in built environment applications: A review

Giovanni Calzolari, Wei Liu *

Division of Sustainable Buildings, Department of Civil and Architectural Engineering, KTH Royal Institute of Technology, Brinellvägen 23, Stockholm, 100 44, Sweden

ARTICLE INFO

Keywords:
Artificial intelligence
Neural networks
Fluid mechanics
Turbulence

ABSTRACT

Fast and accurate airflow simulations in the built environment are critical to provide acceptable thermal comfort and air quality to the occupants. Computational Fluid Dynamics (CFD) offers detailed analysis on airflow motion, heat transfer, and contaminant transport in indoor environment, as well as wind flow and pollution dispersion around buildings in urban environments. However, CFD still faces many challenges mainly in terms of computational expensiveness and accuracy. With the increasing availability of large amount of data, data driven models are starting to be investigated to either replace, improve, or aid CFD simulations. More specifically, the abilities of deep learning and Artificial Neural Networks (ANN) as universal non-linear approximator, handling of high dimensionality fields, and computational inexpensiveness are very appealing. In built environment research, deep learning applications to airflow simulations shows the ANN as surrogate, replacement for expensive CFD analysis. Surrogate modeling enables fast or even real-time predictions, but usually at a cost of a degraded accuracy. The objective of this work is to critically review deep learning interactions with fluid mechanics simulations in general, to propose and inform about different techniques other than surrogate modeling for built environment applications. The literature review shows that ANNs can enhance the turbulence model in various way for coupled CFD simulations of higher accuracy, improve the efficiency of Proper Orthogonal Decomposition (POD) methods, leverage crucial physical properties and information with physics informed deep learning modeling, and even unlock new advanced methods for flow analysis such as super-resolution techniques. These promising methods are largely yet to be explored in the built environment scene. Unavoidably, deep learning models also presents challenges such as the availability of consistent large flow databases, the extrapolation task problem, and over-fitting, etc.

1. Introduction

The study and control of the airflow in the built environment is of great importance since it directly affects human daily life primarily in terms of health, comfort, and productivity. Now more than ever, the urban environment [1,2] and the indoor environment [3,4] are facing challenges in providing acceptable air quality and thermal comfort due to increasing pollution and climate change. Fast and accurate airflow information is therefore desirable when it comes to built environment applications of inverse design, system control, evaluation, and management. Computational Fluid Dynamics (CFD) is a useful tool that enables detailed predictions through numerically solving the Navier–Stokes (N–S) equations. Through the years, CFD has been widely and consistently used in the indoor environment to simulate turbulent airflow [5], heat transfer [6], and contaminant transport [7], as well in the urban environment to simulate wind flow around buildings [8] or at pedestrian level [9] and tracking pollutants [10]. However, CFD still faces many challenges mainly in terms of computational expensiveness and accuracy.

In realistic turbulent flow field cases, the N–S equations cannot be solved analytically, but numerically with space and time discretization. In order to fully capture the flow phenomena, all different scales of turbulence have to be resolved. This CFD approach is called Direct Numerical Simulation (DNS) [11]. Although DNS is conceptually simple, it is extremely computationally expensive and still infeasible on numerous applications. The Large Eddy Simulation (LES) [12] is another method that resolves the large three-dimensional unsteady turbulent motions directly and model the small-scale ones. This approach is justified by the universal shape and isotropy of the small-scale turbulence, but mainly applicable for external flow over large bodies where the boundary layer is of less importance. Although the computational effort compared to DNS is reduced, LES is still expensive and out of reach for

* Corresponding author.
E-mail address: weiliu2@kth.se (W. Liu).

https://doi.org/10.1016/j.buildenv.2021.108315
Received 28 June 2021; Received in revised form 11 August 2021; Accepted 28 August 2021
Available online 16 September 2021

applications with large simulation domain. The industry standard of low computational demand CFD analysis is still the Reynolds Averaged Navier Stokes equations method (RANS) [13], a generalized approach that comprehends all the models that compute the time-averaged N–S equations. Therefore, the RANS simulation only gives the average quantities that constitutes a compromise and it is not possible to reach the same level of accuracy with the DNS approach. Besides RANS equations are still highly non-linear and they introduce the so-called *closure problem* [13]. The closure is often given by modeling the Reynolds stresses using the *eddy-viscosity hypothesis* [14] that establishes a linear relationship between Reynolds stresses and mean strain rate. The *eddy-viscosity hypothesis* is renowned to be physically wrong, but nevertheless used [14]. Such an assumption further increases the uncertainty of the numerical simulation, which is another reason that none of the RANS turbulence models can ever be accurate for all range of flows. These are the main of numerous uncertainties raised by CFD. In a report for the future vision of CFD in 2030, Slotnick et al. [15] view the current situation as stagnant remarking that even though CFD represents a certainly more economic solution than physical experiments, it is still not possible to leave aside the latter in any step of the design process, because CFD simulation are either not to be trusted completely or too computationally expensive to run. Besides, the problem with CFD simulations in built environment is that, even with small indoor domain, the airflow is quite complex, making RANS simulations especially demanding in the design process with numbers of scenarios.

While work is being done to improve CFD techniques themselves with new algorithms and new turbulence models [16,17], recent interest is posed on new tools to either substitute CFD in the analysis of fluid mechanics problems, typically for faster predictions, or use them as aid to the CFD simulation, for improved accuracy. With digitization and the availability of large data, artificial intelligence and data-driven models are receiving much attention in numerous and different applications. More specifically, the abilities of deep learning and artificial neural networks (ANNs) as universal non-linear approximator, handling of high dimensionality fields, and computational inexpensiveness are in general very appealing. There are already several works that substitute CFD simulations with deep learning algorithm with the ANN as surrogate model of the numerical simulation. Depending on the specific problem, surrogate modeling can allow for order of magnitudes [18] faster airflow prediction, even real-time predictions in some cases [19], solving one of the two problems raised by CFD simulations. This is generally what it is done and presented in build environment research [19,20].

Even though the main objective of surrogate modeling is to reduce computational cost while keeping the same order of accuracy [21], the computational inexpensiveness usually comes at the cost of degraded accuracy. Surrogate modeling is often built from and compared with CFD simulations, which still present all the uncertainties stated above. In the best cases ANNs produce very similar results in terms of accuracy, but worse in others. Because of its inexpensiveness, surrogate modeling is the major focus of the current research of built environment application, but there certainly are possible other ways to make use of deep learning architecture. By reviewing some major applications of deep learning that have been attempted in fluid mechanics research to improve the accuracy of CFD simulations, hints and opportunities for future research on integration of ANNs and CFD for built environment applications are proposed.

In this paper, Section 2 reports a brief but complete summary on deep learning and neural networks with an historical perspective. Section 3 presents the current state of the art of the applications in the built environment. Because of the versatility of deep learning algorithm, they can be used in a potentially unlimited number of ways, and are already in numerous different applications, starting from the last century and continuing now with new emerging techniques. Section 4 aims to critically review applications like turbulence model tuning and enhancement, surrogate modeling, POD and super-resolution to assess how ANNs can improve CFD analysis for fluid mechanics in practice. Finally, Section 5 offers a discussion about possible opportunities, but also challenges for deep learning methods to aid, substitute, or improve CFD simulations for the built environment.

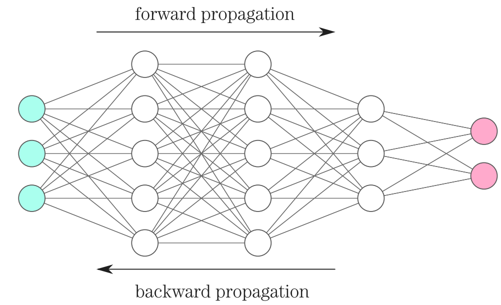


Fig. 1. A four Layer MLP, with a 3 nodes input layer colored in light blue, 3 hidden layers of 5-5-3 nodes respectively in white, and a 2 nodes output colored in pink. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2. Deep learning and artificial neural networks

Before diving into engineering applications, this section is to provide an introduction and summary of deep learning models as guide for the targeted reader, a built environment and fluid mechanics engineer newly approaching to data driven solutions. The history of deep learning is surprisingly not so recent dating back in the 1940s [22], with consequent waves of increased popularity, but also skepticism. Two defining factors of the recent success of ANNs are the rapid increase in computational power from the 90 s making possible the training of more elaborate models, and the world digitization, which makes available huge labeled data-sets. ANNs have consistently improved in their ability to provide accurate recognition or prediction and have been applied to broader and broader sets of applications [22], such as CFD.

ANNs are the basic supervised learning algorithm of deep learning. ANNs are feed forward computing systems with the task of mapping the function f between the input X and the output Y as shown in Eq. (1)

$$Y = f(X) \tag{1}$$

ANNs belong to supervised learning exactly because they are to be *trained* from "labeled" data, where input and output feature vectors are known. The power of ANNs resides in the fact that once the learning procedure is terminated, it is possible to apply the trained and tested algorithm (with the same function f) to "unseen" data of the same group and obtain accurate prediction of output.

To understand how the function f in Eq. (1) is constructed, it is necessary to understand the ANNs nature of distributed representation in their layers composition. Fig. 1 shows an ANN in its most standard architecture: the Multi Layer Perceptron (MLP). In general, every ANN connects the input layer (X) to the output layer (Y) through a number of in-between layers called *hidden layers*, which allow to extract more complex features from the input. Every layer l is structurally composed of basic units called *nodes* stacked together. Each node then calculates its output through a linear weighted relation of the input, which is masked by a non-linear monotonically increasing activation function σ.

The structure of a basic node is shown in Fig. 2, and the output a_i^l of node i at a layer l is mathematically expressed in Eq. (2), where w_{ji}^l and b_i^l are the weights and bias of the linear relation. The index j in the sum refers to each node at layer $l-1$, the total node number of which is in general different from that of layer l as shown in Fig. 1. Eq. (3) shows the expression of the most standard activation function, the Rectified Linear Unit (ReLU) [23].

$$a_i^l = \sigma(z_i^l) = \sigma\left(\sum_j w_{ji}^l x_{ji}^l + b_i^l\right) \tag{2}$$

$$\sigma(z) = \max(0, z) \tag{3}$$

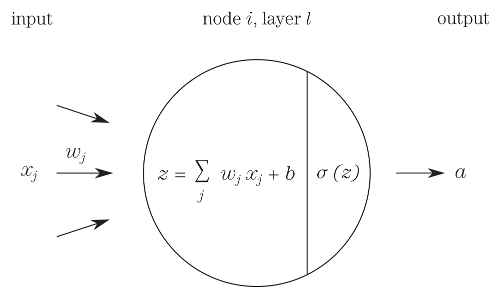


Fig. 2. Basic Structure of a node i of a hidden layer l of an ANN.

In the training process, this procedure of sequentially computing each node output a_i^l from the input layer through all the hidden layers is called forward propagation, which ends with a computed prediction of the last layer L, the output vector $A^L = \hat{Y}$. The final output $\hat{Y}$ will then be measured against the *known* labeled output data vector (Y) considered as the model true value. The error between the prediction and the true value will be counted by a loss function of a minimization algorithm, which will also compute derivatives with respect to the weights w_{ji}^l and bias b_i^l at every node i and layer l through back-propagation and update them. Even though different optimization algorithm and loss functions $\mathcal{L}$ are used to train ANNs, the historical standard is still RMSE (Root-Mean-Square-Error) and the gradient descent algorithm update procedure, which expressions are reported in Eq. (4), (5) and (6), respectively.

$$\mathcal{L} = \|Y - \hat{Y}\|^2 \tag{4}$$

$$w_{ji}^l = w_{ji}^l - \alpha \frac{\partial J}{\partial w_{ji}^l} \tag{5}$$

$$b_i^l = b_i^l - \alpha \frac{\partial J}{\partial b_i^l}, \tag{6}$$

where α is the learning rate hyper-parameter and J is the cost function, meaning the weighted sum of the loss function $\mathcal{L}$ over the entire training set. Once the optimization is complete, the trained ANN will have learned the function mapping input and output with a set of optimized weights (w_{ji}^l and b_i^l). Now it can be applied to give predictions that require only a one-time forward propagation computation. The training of ANNs might be a time-consuming procedure, but applying the trained ANN to obtain prediction is usually extremely fast. Typically, the larger the database is, the better the predictive capabilities will be. Thanks to the distributed representation, inherent non-linearity provided by the activation function, and flexible depth of the architecture in terms of number of hidden layers and nodes, ANN are highly effective in high dimensional problems and have been applied to more and more complex problems over the years. It has been demonstrated that an ANN with at least one hidden layer and arbitrary number of nodes can approximate any continuous function (even non-linear) on a compact subset, i.e. it is an universal non-linear approximator [25].

In the last decades, new and evolved deep learning architectures have been proposed, starting from the standard MLP. In fluid mechanics analysis which appear in the studies reviewed in Sections 3 and 4, the most common deep learning models are MLP, Convolutional Neural Network (CNN) [26], Recurrent Neural Network (RNN) [27], and other architectures below briefly summarized.

2.1. Multi Layer Perceptron (MLP)

MLPs are the most common and standard ANN architecture. They consist of multiple densely fully connected layers of the nodes stacked together in a feed forward algorithm, which working process and structure have already been showed above (see Fig. 1). Deep MLPs are in general able to reveal more complex features from the input, but their full connectivity makes them prone to show *over-fitting*, when the mapping between input and output adapts too closely to the training set. To avoid or reduce over-fitting, regularization techniques have been explored and implemented successfully over the years, but it still remains a challenge.

2.2. Convolutional Neural Networks (CNN)

CNNs are originally developed on the basis of MLPs for computer vision tasks specifically [28]. Taking as an example image data, every pixel in an image is usually relevant with respect to its surrounding pixels or even pixels that are far away. Pattern detection is what makes CNNs so useful. The name convolutional comes from a mathematical operation of **convolution**, which is used by CNNs in at least one layer. In opposition to densely connected MLPs, CNNs replace the standard weights by *convolutional kernels* or *filters*. Their function is to map local space of the input vector into output by constantly sliding through the input vector. Convolution operation is justified by the concepts of sparse interaction and weight sharing [22], making the CNN a regularized version of a MLP and possibly extremely more efficient in terms of both memory and computational costs. Fig. 3 exemplifies the convolution operation by providing visual understanding. The first top left block of output (in green) is the result of a convolution operation on the input (in blue) filtered to a 3 × 3 matrix with the convolutional kernel. The remaining output blocks are computed in the same way sliding the kernel through the input layer.

CNNs can also feature hidden layers with pooling operation [24], which objective is to reduce the dimension of the input vector to highlight patterns and consequently avoid over-fitting. Typically max-pooling or average-pooling operation are used, one selecting and isolating the maximum value in a local region or the average the latter. A typical architecture of a CNN is shown in Fig. 4. The operations between each layers are, in order, convolution, max-pooling, convolution, and max-pooling operations again and finally standard fully densely connected layers.

CNNs can be applied in different fields in problems where patterns are important and processing data that has a known and grid-like topology [22], including fluid mechanics.

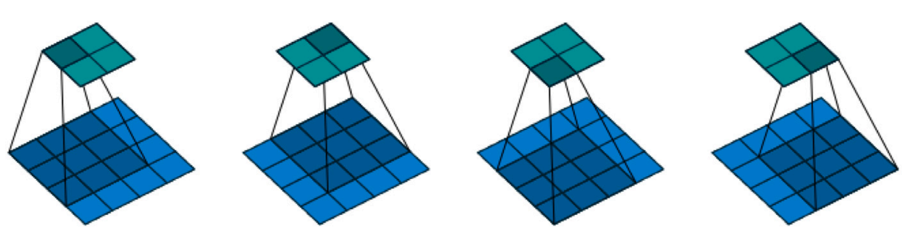


Fig. 3. Convolution operation example with a 3 × 3 *convolutional kernel* over a 4 × 4 input layer.
Source: Taken from [24].

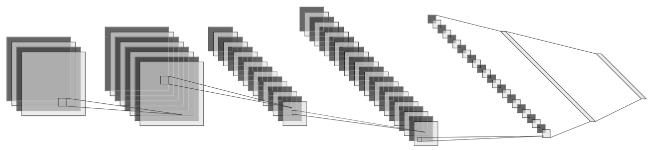


Fig. 4. A six layer example typical architecture of a CNN.

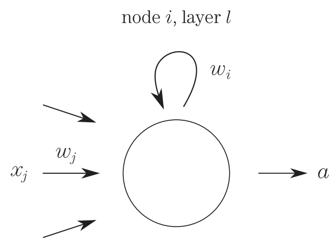


Fig. 5. Basic Structure of a node i of a hidden layer l of a RNN.

2.3. Recurrent Neural Network (RNN)

RNNs are specialized neural network architecture for time sequence data [29]. RNNs differ from standard MLPs because of the presence of a feedback loop, which makes it able to remember previous input data. Fig. 5 shows their node basic structure in its simplest form. The self-loop weights w_i make RNNs specifically adapted for time-sequence data analysis.

Because of the so-called vanishing gradients problem during the training procedure, meaning when the back-propagation computed derivatives decay exponentially and approach to zero, a RNN can be difficult to train [30]. For this reason, Long-Short-Term-Memory RNN (LSTM-RNN) architecture has been developed [31], which is the most common RNN algorithm used today. The main improvement of LSTM-RNNs resides in the use of gates, specifically a *forget* gate applied at the self-loop weights. This allows to discard some information during the training procedure, consequently solving the vanishing gradient problem and showing to learn long-term dependencies more easily than simple recurrent architectures.

2.4. Other deep learning architectures

Besides the main ANN architectures such as MLP, CNN, and RNN, there are several other deep learning algorithms. Some of those deep learning algorithms are new and emerging and some other are modification of the standard architectures to better solve a specific problem.

1. Extreme Learning Machine (ELM)
   ELMs are MLPs with the presence of only 1 hidden layer, with the difference that ELMs randomly choose hidden nodes and analytically determine the weights and bias without needing to tune them during the learning procedure [32]. It is an interesting new approach that goes in the opposite direction of the latest tendency of making deep ANN. The consequent extremely reduced learning computational cost can make it attractive in fluid mechanics since the amount of training data is enormous in size. Further, ELMs have good strong nonlinear generalization performance [33].
2. Support Vector Regression (SVR) and Random Forest (RF)
   There are other supervised learning methods that are based on statistical learning frameworks, such as Support Vector Regression (SVR) [34] or Random Forest (RF) [35]. RFs are special architectures made of decision trees merged together, while SVRs are an evolved version of linear regression approximator, which can also be applied to non-linear problems.

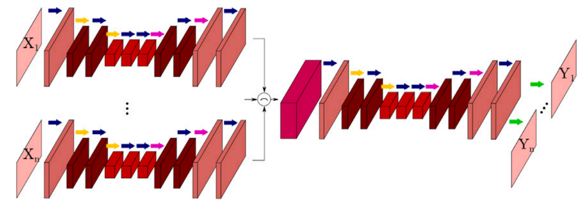


Fig. 6. MIMO-CNN architecture used by Font et al. [36].

3. Hybrid and *ad-hoc* Architectures
   Deep learning algorithm also possess the flexibility to create hybrid and *ad-hoc* architectures for improved performance on specific tasks. For example later in Section 4 an application of a MIMO (Multiple-Input-Multiple-Output) CNN architecture will be reviewed, which consists of different CNNs merged together with a final CNN [36]. Its structure is shown in Fig. 6. From each input field, an encoding–decoding CNN branch develops. The branches are eventually concatenated together as input for another final encoding–decoding CNN branch that leads to the output layer.

3. Built environment application

Numerous built environment applications are strictly connected to the study of fluid flows and in need of fluid mechanics simulations. Some examples of CFD simulations that are consistently performed in the indoor environment for inverse design are thermal comfort analysis, and pollutant dispersion, etc. For the outdoor environment, the simulations of wind and tracking pollutants are conducted at various urban levels. Inverse design is a design process that in the last years has heavily made use of numerical simulation techniques. In opposition to trial-and-error methods, we refer with the inverse design term to the design process which starts and poses at the center an objective in performance to satisfy and then uses a method that automatically search for a suitable system that satisfies that objective. Thermal comfort in the indoor environment, such as vehicle cabins of cars but also aircraft is a challenging problem which has been attempted to solve with increase popularity by inverse design. Liu et al. [37] and Wang et al. [38] for example, proposed respectively a CFD based adjoint and POD-CFD based method to address this kind of problems. Together with these methods, even deep learning models have started to cover an important place in these applications. Deep learning techniques have already been widely applied since the early 2000s in specific topics such as building energy predictions [39–41] or HVAC system control [42,43]. However, data driven interactions with CFD simulations are still limited in quantity and diversity of approach. The vast majority of applications in the built environment are limited to substituting surrogate modeling for expensive CFD simulations to achieve faster predictions. CFD simulations can in fact be very computationally demanding, especially in the design process and emergency management and control, where fast predictions are necessary, or in the outdoor environment where the simulation domain is usually large. The main objective of surrogate modeling is to reduce computational cost while keeping the same order of accuracy [21]. Section 3.1 reviews the studies on the topic, also summarized by Table 1.

3.1. Deep learning surrogate modeling in the built environment

The analysis of the indoor flow of vehicles and cabin environment to assess the thermal comfort of the occupants is one important built environment application. In the work by Hintea et al. [19], from a minimalistic set of cabin environment sensors, several different machine learning algorithm, including a standard MLP network and a

Table 1
Current research works with interaction between deep learning models and CFD simulations in the built environment reviewed in Section 3.1.

Authors	Deep Learning model					Application	
	MLP	CNN	RNN/LSTM	O	S/H	IE	UE
Hintea [19]	•			•		•	
Zhang [20]	•					•	
Zhang [44]	•					•	
Warey [45]	•			•		•	
Tanaka [18]		•					•
Ding [46]				•			•

Note: MLP = Multi layer perceptron; CNN = Convolutional neural network; RNN/LSTM = Recurrent neural network/Long short term memory; O = Other (architectures), such as ELM, SVG, etc; S/H = Special/hybrid (architecture); IE = Indoor environment; UE = Urban Environment.

RF algorithm, are used and compared to approximate equivalent temperature inside a car to obtain real-time predictions. The equivalent temperature is measured at eight body locations to provide a direct estimation of thermal comfort. Eventually the fastest prediction is delivered by a simpler linear regression model, while the MLP obtains the highest accuracy overall. As shown instead by other works (Zhang et al. [20,44] and Warey et al. [45] for instance), CFD simulations are not discarded completely, but still used as high fidelity solution necessary for training of the deep learning algorithm. Warey et al. [45] (see Fig. 7(a)) use deep learning models, such as an MLP, a RF and more, as surrogate models to quantify the thermal comfort of indoor vehicle cabins for different boundary conditions of air temperature, but also air velocity, humidity, glazing conditions, etc. To better assess the indoor thermal comfort this time not only an equivalent temperature analysis is performed, but also on more advanced evaluation scores, the predicted mean vote and the predicted percentage of dissatisfied people. More rationally, these scores evaluates thermal comfort from an overall perspective [47]. CFD simulations previously validated with wind tunnel experimental measurements are generated to be used to train the deep learning algorithm, which was then applied also to different fields with real time predictions. Instead, Zhang et al. [20] first use a shallow standard MLP in general indoor environments with CFD as training data to solve the inverse design problem and identify a possible relationship between thermal comfort and inlet boundary condition. Later on in another work, Zhang et al. [44] apply the based knowledge on a simplified first class aircraft cabin environment. This time the deep learning surrogate model is integrated inside a genetic algorithm and the results are compared against a 57% more computationally expensive classic genetic algorithm without deep learning tools. Three shallow MLPs are singularly trained to obtain the predicted mean vote, the air age and the draft rate, the latter assessing the local discomfort for human.

Upgrading the scale of simulations from indoor to external environments, like urban cities, accuracy deterioration and struggle to consider other boundary conditions such as traffic or weather are inevitable. Tanaka et al. [18] (see Fig. 7(b)), study urban flow simulations where the location, dimension and shape of four tall buildings were optimized in a restricted area with the construction of a CFD optimization tool. The objective is to reduce wind forces on buildings and mitigate local strong winds at pedestrian level. The focus of the paper still resides on CFD RANS simulations, which are eventually used to train a deep CNN encoder–decoder for *ultra fast* (0.005 s) predictions. They estimate the CNN predictions to be about fifty thousand times faster than RANS. However, the ANN predictions show lack of accuracy compared to CFD simulations especially in specific case that the network was not trained for, such as different wind directions. In Tanaka et al. [18] case, the deep learning method has to be viewed more as a parallel tool to the CFD simulation, useful in the early design process for example. A similar example is also given by the work of Ding et al. [46], which develop data driven regression model for coupled indoor–outdoor flow analysis together with CFD simulations. Eventually, surrogate modeling is an useful technique, which main advantage resides in the possibility to obtain really fast and inexpensive predictions, otherwise unfeasible with more expensive CFD simulations. It also comes with limitations such as being strongly case dependent, without good generalizability, and in need of large training data sets. In most indoor environment studies, only the case related data were used for training. Then the trained ANN was very likely unfeasible for another case that the network was not trained for. Moreover, its main disadvantage is the limited accuracy, which can become an important factor in practice. Many possibilities of utilization of deep learning models that Section 4 will cover are yet to be explored in the built environment, possibly to further increase also the accuracy of the analysis and create more trusty models.

4. Deep learning applied to Fluid Mechanics

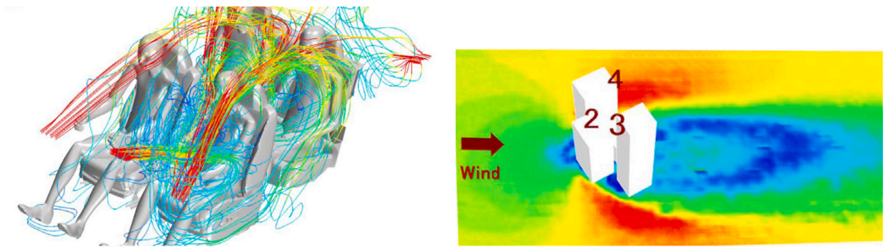
Section 3 highlighted that the current use of deep learning models to flow simulations in the built environment is limited to surrogate modeling for faster predictions. However, looking at the past and current research in fluid mechanics the range of applicability of ANNs is in general much richer. This section aims to inform about possible different techniques and application of ANNs to drive forward the research in the built environment. More specifically, Section 4.1 gives a brief historical perspective on the first research works of interaction between deep learning and fluid mechanics, while Sections 4.2 and 4.3 focus on the main current applications.

4.1. Deep learning and fluid mechanics history

In the 1980s, Kutler et al. [48] speculate about an approach between fluid mechanics (aerodynamics) and Artificial Intelligence (AI) and how AI would make computational complexity manageable. The first practical applications of deep learning applied to a fluid mechanics problem emerge around the 1990s, beginning with flow-related applications that are not limited to numerical simulations. In 1991, Teo et al. [49] developed an ANN for Particle Image Velocimetry (PIV) and Lee et al. [50] applied an ANN to turbulence control of a inflow jet for drag reduction. Towards the end of the XXth and the beginning of the XXIth century, the first proper CFD applications adopted ANNs as surrogate modeling for faster prediction in heat transfer data analysis [51], flow pattern estimation [52], aerodynamic design [53], engine modeling [54], and design optimization [55]. All these first cases span only shallow neural network with one hidden layer.

Another approach given by Milano et al. [56] is modeling near wall turbulent flows with Proper Orthogonal Decomposition (POD) in 2002. POD belongs to the branch of order reduction models and decomposes a physical fields into a basis along the principal component analyses, allowing for example to quantify the structure of turbulence through recognized basic patterns [57]. The limitations raised by POD reside in the inherent linearity of the mode decomposition. Milano et al. [56] showed that a linear neural network with only one single hidden layer can be trained to be the exact equivalent to a POD. ANNs can actually be viewed as generalization of POD, but with improved architecture consisting of non-linear layers that eliminates the linear limitations of POD. The structure of the ANN used by Milano et al. [56] is a 4 layers MLP similar to Fig. 1, with a combination of linear and non-linear layers. In 2002, the multiple hidden layers architecture was really ahead of the time and this work opens a future research in order reduction models using the concept of *non-linear POD* with deep learning models.

Over the years more and more fluid mechanics research on different problems is looking at deep learning methods, not only for computational efficiency but also increased accuracy. The current state of the art of the research in fluid mechanics using deep learning techniques sees numerous possible applications. Some of them are not discarding CFD



(a) Indoor environment [45] (b) Urban environment [18]

Fig. 7. Two examples of deep learning surrogate modeling for built environment applications. On the left 7(a) the study of thermal comfort for indoor vehicles cabin by Warey et al. [45]. On the right 7(b) the optimization study of urban wind flow simulations by Tanaka et al. [18]-

numerical simulation, but make use of data-driven machine learning models as an aid for augmenting the turbulence models. Some others instead present deep learning architectures as complete substitute and surrogate model or emulator of fluid dynamics numerical simulation, similarly of what is also done in the built environment and reviewed in Section 3. Besides straightforward substitution of CFD with ANN, model order reduction and POD techniques are also reviewed. There are some other applications that use completely different new emerging methods such as super-resolution techniques. The fluid mechanics studies object of analyses are schematically reported in Table 2.

4.2. Turbulence modeling

As remarked already, deep learning techniques can be utilized in conjunction with CFD simulations to increase the accuracy of simulations. What is often done is acting on the turbulence models that usually presents the highest degree of uncertainties.

4.2.1. Tuning coefficients

For the RANS turbulence models, the closure problem together with the eddy-viscosity hypothesis introduce a number of constant coefficients varying from model to model. Those coefficients are determined from combination of dimensional analysis and experimental measurements usually in simplified configuration [81,82]. Therefore, the RANS turbulence models are not universally capable for every flow, but actually depends on the specific flow features. For a complex flow with multiple characteristics, the chosen standard values are a compromise. A first obvious way to utilize deep learning algorithm on turbulence models is for *constant tuning*, finding for different flow applications the best values of the coefficients of a particular turbulence model. For example, Yarlanki et al. [58] estimated the temperature in data centers using a tuned standard $k-\epsilon$ RANS turbulence model [83], where past studies were already showing that the turbulence model was performing poorly in that specific application [84]. The introduction of the MLP model allowed a reduction of the absolute average error in by 35%. Another example is the work by Luo et al. [59]. The main difference between the two is that Luo et al. [59] provide a **physics informed** (PI) neural network by combining the high-fidelity labeled output of DNS simulations with prior knowledge on the mathematical representation of RANS model. The PI-ANN architecture embeds the turbulence physics in the loss function, the sum of a standard RMSE (Eq. (4)) and two weighted parameters which enforce the two transport equations of the standard $k-\epsilon$ turbulence model. This eventually acts as sort of regularization mechanism for the neural network allowing high accuracy even with smaller amount of data for training. In general tuning coefficients can increase the accuracy of CFD predictions, but the value of the coefficients of the turbulence models are problem related, with scarce generalizability. Furthermore, it is important to notice that without proper caution, tuning coefficient techniques could actually introduce more uncertainties to the analysis; the new tuned coefficient could in fact compensate for example for experimental errors, inaccurate boundary conditions or invalid grid resolution, etc. Physics informed deep learning modeling can help in this regard, to help contain this problem and increase generalizability so that the same architecture can be applied to different flow fields.

4.2.2. Turbulence models enhancements

Even though improving turbulence model coefficients is still extremely useful, deep learning algorithm can actually achieve much more when interacting with turbulence modeling. Research is focusing on new ways of enhancing turbulence models by acting directly to the source of uncertainties such as the closure modeling and the evident failings of eddy-viscosity hypothesis [14]. An idea is to use deep learning algorithm to build a *representation* of turbulence modeling closure terms. Tracey et al. [60,85] take a one-equation turbulence model, Spalart–Allmaras [86] model, as the true model and try to reproduce the same results by replacing the source term of the model with an ANN, a classic shallow MLP. They established a methodology for future studies where the training data becomes more accurate DNS or well resolved LES simulations. The enhanced model in practice learns the behavior of turbulence leading to better prediction over a wider range of flows. Singh et al. [61] do a similar work, where experimental measurements are used together with the Spalart–Allmaras model for minimizing the difference between the measured lift coefficient and the model output through an inverse approach. The possibility to use DNS data as labeled output, which theoretical accuracy easily tops RANS's, is the clear next step in the research. In 2016, Ling et al. [62], for instance, explore this opportunity in a work which already became a classic in the field. What the paper achieves is not limited to the use of DNS data, but much more. First of all, the objective is to act directly on the eddy-viscosity hypothesis, substituting it with an ANN model. The ANN used is a very deep architecture of more than 10 hidden layers, even proposing an advanced alternative to MLP, called Tensor Basis Neural Network (TBNN) (shown in Fig. 8, together with the standard MLP). As the name remarks, this custom deep architecture makes use of a tensor basis set to impose within the ANN prior physical knowledge about fluid flows. In particular, the TBNN architecture embeds symmetry and physical property of Galilean-rotational invariance, which is ultimately fundamental for accuracy of the CFD solution. Using 9 different flows for testing and training, with both an *a priori* analysis on the anisotropy tensor and *a posteriori* analysis on the velocity profile and shape of the flow they are able to obtain much better results using the TBNN compared to the simpler standard MLP and even conventional RANS models too. Similar applications focusing on augmenting turbulence modeling are for instance the work of Geneva et al. [63] using the same TBNN architecture from Ling et al. [62] coupled with bayesian statistics, or Wang et al. [64] which makes use of random forest algorithm. Consistent reviews on the topic have been written by Duraisamy in two different articles [87,88].

In some cases deep learning can even reveal new hidden correlations where the physical laws are still not known *a priori*. A perfect example of this interaction is given by Rudy et al. [89] for parametric partial differential equations or by Font et al. [36] for turbulent flow analysis. Font et al. [36] present a novel set of 2D N–S equations based on local

Table 2
Current research works with interaction between deep learning data driven models and fluid mechanics simulations reviewed in Section 4.

Authors	Deep Learning model					TM		Surrogate Modeling			PI
	MLP	CNN	RNN/LSTM	O	S/H	CT	TME	SSM	POD/ROM	SR	
Yarlanki [58]	•					•					
Luo [59]	•					•					•
Tracey [60]	•						•				
Singh [61]	•						•				
Ling [62]	•				•		•				•
Geneva [63]	•				•		•				•
Wang [64]				•			•				•
Font [36]		•			•		•				
Guo [65]		•						•			
Maulik [66]	•							•			
Fukami [33]	•	•		•	•			•		•	
Srinivasan [67]	•		•					•			
Guastoni [68]		•						•			
Guastoni [69]			•					•			
Guastoni [70]		•			•			•	•		
Fresca [71]	•	•			•				•		
Wang [72]			•						•		
Mohan [73]			•						•		
Lui [74]	•								•		
Erichson [75]	•									•	
Fukami [21]		•			•					•	
Fukami [76]		•								•	
Liu [77]		•			•					•	
Gao [78]		•								•	•
Deng [79]				•	•					•	
Bai [80]				•						•	

Note: TM = Turbulence modeling; MLP = Multi layer perceptron; CNN = Convolutional neural network; RNN/LSTM = Recurrent neural network/Long short term memory; O = Other (architectures), such as ELM, SVG, etc; S/H = Special/hybrid (architecture); CT = constant tuning; TME = turbulence model enhancement; SSM = simple surrogate modeling; POD/ROM = Proper orthogonal decomposition/Reduced order modeling; SR = Super-resolution; PI = Physics-informed (model).

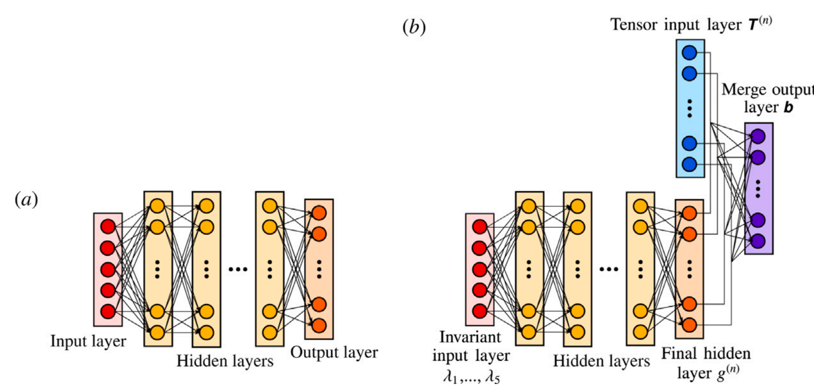


Fig. 8. Standard MLP architecture (a) versus TBNN architecture (b) used by Ling et al. [62]. The TBNN presents a further input layer called Tensor input layer, which makes sure that the ANN embeds the physical property of rotational invariance.

spanwise average of the flow (called SANS equations). These equations actually take into account the 3D spanwise effect of turbulence which is otherwise ignored in 2D equations. The SANS closure problem, which is physically unknown, is then handled by a CNN. Comparing to the difficulties of conventional eddy-viscosity models and 2D RANS equations, the method successfully allows to use computationally inexpensive 2D simulations to provide solutions with the same level of accuracy of much more computationally demanding 3D simulations. The deep learning architecture is actually a deep Multiple Input-Multiple Output CNN (MIMO-CNN) architecture, where every input presents itself a CNN, which are then concatenated into a single encoding–decoding branch as shown by Fig. 6 in Section 2. The last CNN branch is useful to provide a clean output in regions where closure terms are negligible. Overall, turbulence model enhancement techniques allow for impressive accuracy improvements, faster simulations in some cases and perhaps most importantly even the discoveries of new equations and correlations about turbulence phenomena that were not known before.

4.3. Surrogate modeling and reduced order modeling

As shown in Section 3 for the built environment, ANNs can also be used to marginally or completely substitute CFD simulations in the analysis of fluid flows, probably one of the most common deep learning application also in general fluid mechanics. This philosophy is also close to reduced order modeling techniques [90] such as POD, which aim to reduce the high dimensionality that CFD computations present (especially DNS). In general, computational efficiency of reduced order based methods can be 1–2 orders higher compared to conventional CFD methods [91], even allowing real-time prediction in some cases. As reported in Table 2, some of the works belonging to surrogate modeling and ROMs techniques are reviewed and presented below. The surrogate modeling fits into three categories: simple surrogate modeling, POD, and super-resolution.

4.3.1. Simple surrogate modeling

For example, Guo et al. [65] develop a CNN algorithm that performs at least two orders of magnitude faster than equivalent CFD solution (using Lattice-Boltzmann method). The comparisons on velocity profiles for different external flow problems shows that the CNN solutions are slightly less accurate than that of CFD. Fukami et al. [33] instead analyze the feasibility of five different deep learning architecture (ELM, RF, SVR, MLP and CNN) for aerodynamic design predictions in three different regression problems, including the estimation of force coefficients of wake flows. Sensor measurements were used as labeled output for the network training. Maulik et al. [66] train a standard deep MLP to develop a surrogate turbulent eddy-viscosity model and bypass any extra equations for closure. The training input is the potential flow solution to the flow. Consequently only velocity and pressure equations needs to be solved. Lastly, the multiple works done primarily by Guastoni et al. [67–70], apply deep learning models to the open channel extremely expensively boundary layer DNS simulation [92]. It is one clear example of the computational efficiency advantage of data driven models surrogate over brutal numerical simulations. In general, simple surrogate modeling provides a computational cost-effective solution to otherwise unfeasible or expensive problems, often compromising on the level of accuracy.

4.3.2. POD

Since the work by Milano et al. [56], which shows the possibility to consider specific ANNs as generalization of non-linear POD and principal component analysis, the development of model order reduction using POD with ANNs has been uninterrupted. In essence, the deep learning POD problem consists of a dual offline-online computational splitting procedure: the POD low-dimensional subspace is built in the offline step, which will then be used to give ideally very fast prediction in the latter online stage. However, the offline procedure can actually become quite expensive for fluid mechanics applications [93]. Fresca et al. [71] develop a strategy to make the offline training dramatically faster by performing a prior dimensionality reduction through POD and a pre-training stage where different models are combined to initialize the parameters of the algorithm. By enhancing the deep learning algorithm, the high-dimensionality bottleneck of the equations is overcome. Improved ANNs architectures have also been researched over the years. For example in 2018 Wang et al. [72] applies the LSTM-RNN architecture shown in Section 2.3 with POD techniques in the study of ocean currents and flow around cylinder. The same architecture is used later by Mohan et al. [94] for flow control applications. Other examples of POD decomposition with deep learning techniques are given recently by Lui et al. [74] or the already mentioned work by Guastoni et al. [70]. The first proposes a numerical methodology for construction of POD, using a deep MLP for regression analysis, comparing the algorithm against sparse regression, which is outperformed. Guastoni et al. [70], instead, together with a conventional CNN architecture applies also a neural network combination which reconstructs the flow fields using a linear combination of orthonormal basis functions, obtained by POD. The POD based ANN obtains better prediction further away from the wall, while the standard CNN better close to the walls. POD deep learning techniques are eventually an interesting and efficient form of order reduction modeling, which allows much faster predictions while maintaining reasonable accuracy [72].

4.3.3. Super-resolution

Amongst the benefits that data driven model brought to fluid mechanics research, *super-resolution* is a preeminent and new one for the analysis of fluid mechanics problems. Borrowed from computer vision and image recognition, super-resolution imaging refers to the class of techniques that aims at obtaining a high resolution image output from a low-resolution image. Deep learning has already been extensively used for super-resolution and lately mainly through CNNs [95], given their predisposition for visual pattern recognition as shown in Section 2.2. The same concept could be applied to flow fields by treating them as *images*. Using an offline database of high resolution snapshot of flow field (for example DNS data) as labeled output, it is possible to give input from low resolution data either from experimental measurements as done by Erichson et al. [75] or from fast and computationally inexpensive simulation. The method consequently reconstructs the turbulent field to high resolution with the ANN. More specifically Erichson et al. [75] make use of shallow standard MLP to first reconstruct a 2D cylinder wake at Reynold number of 100, with 10 sensors producing low-resolution input and getting DNS data as high resolution output. They also apply the same architecture to the study of isotropic turbulence, using high resolution snapshot from DNS as output and its filtered low-resolution image to form the input, as shown in Fig. 9.

Fukami et al. [21,76] take DNS high-resolution data and purposely down-scale them with a pooling operation to obtain the low-resolution input data. Using two different neural network architecture, one a classic CNN and another improved hybrid ANN algorithm that can handle the multi-scale nature of the flow, Fukami et al. [21,76] successfully reconstruct the same cylinder wake of Erichson et al. [75] and subsequently even 2D decaying isotropic turbulence. Liu et al. [77] adopt a specific novel architecture called multiple temporal paths convolutional neural network (MTPC) to perform super-resolution. Leveraging from the spatiotemporal coupled nature of turbulent flow, the main improvement of MTPC consists in taking a temporal sequence of snapshots as input instead of only instantaneous snaphots as done with a standard, referred by Liu et al. [77] as *static*, CNNs. Therefore, the MTPC obtains extra temporal information from adjacent frames, eventually resulting in increased performance.

In all these cases, the model super-resolves the low resolution field without assuming any *a priori* knowledge of the physics, which demonstrate the strength of data-driven super-resolution techniques. At the same time, the super-resolution opens the possibility to incorporate the knowledge of the physics into the learning process for improved accuracy in future studies. Gao et al. [78] develop a physics-informed deep learning based super resolution solution using a CNN. The aim of a physics-informed algorithm is to guarantee that the super-resolved fields are faithful to the physical laws and principles. Moreover, by leveraging the physical laws and boundary conditions of fluid flows, Gao et al. [78] CNN solution does not rely on high-resolution data for training. Other notable deep learning super-resolution applications in fluid mechanics are Deng et al. [79], which applied generative adversarial networks (GAN) or Bai et al. [80], for impressive smoke flow visualizations. As shown, super-resolution techniques eventually offer a new and efficient way to study turbulent flows, with enormous potential especially when coupled with CNNs.

5. Opportunities and challenges in the built environment

This study reviews that the current main focus of deep learning applications in built environment resides in the construction of surrogate model for faster prediction. Nevertheless, Section 4 shows that the range of applicability of deep learning methods for fluid mechanics problems is actually much broader. This section focuses on a discussion about possible opportunities that deep learning and ANNs offer to the built environment, as well as challenges to overcome.

Most RANS turbulence models are designed for aeronautics applications with high speed flow, while built environment simulations usually present low velocity fields. The standard coefficients of the models might be inadequate in various scenarios as highlighted by Yarlanki et al. [58]. Tuning coefficients with the use of ANNs is a practical scope that can already decrease error in quantities of interest. The problem that a simple coefficient tuning procedure raises stands in its limited ability to be applicable to different flow fields other than the one studied and in the possibility to misuse the ANNs for new coefficients that compensate for inaccuracies and errors in the analysis. More advanced research topic to explore in the built environment is the enhancement

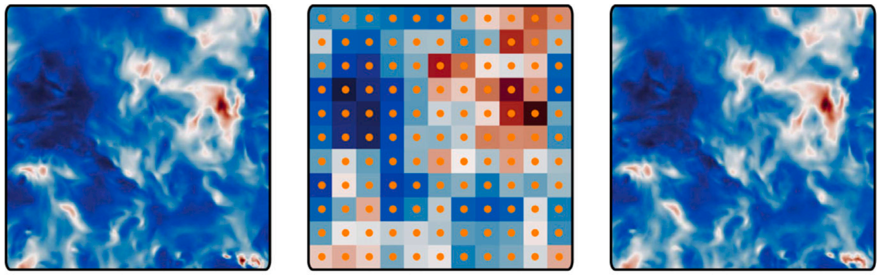


Fig. 9. Super-resolution study methodology of isotropic turbulence by Erichson et al. [75]. From the test snapshot data 9(a) and the low-resolution input 9(b) derived from sensor measurements, the deep learning algorithm is able to super-reconstruct the flow field to obtain a high-resolution output 9(c).

of the turbulence models in a similar fashion to what was presented in Section 4.2.2. By acting directly to the source of the uncertainties such as the eddy-viscosity hypothesis in RANS cases, and replacing it with deep learning algorithm, substantial accuracy improvements can be reached. However, together with higher CFD simulations accuracy, the computational efficiency in built environment design problems still needs to be addressed. Depending on the specific case, CFD simulations have a threshold of grid resolution under which it is not possible to obtain accurate results. Refining the grid domain results in an exponential increase of computational power required, which makes coupled deep learning CFD unfeasible for some applications, especially when fast predictions are needed, such as in control, design or optimization tasks.

On the other hand surrogate modeling inevitably allows faster computations, but the predictions quality is usually affected. Surrogate modeling is also able to obtain results without any prior physical knowledge about the problem, thanks to a large amount of training data. This is advantageous in some aspects, as it is not required previous deep physical knowledge and experience on the problem, but it can be problematic in others. For example without prior physical knowledge, the trained ANN could provide adequate level of accuracy on the variables trained, but critically non-physical results overall. Physics informed modeling is another possible step into higher quality predictions, which could also increase the generalizability of the model. About the direction of turbulence modeling research, Duraisamy et al. [96] note that one should not throw away the existing knowledge-base in turbulence modeling but rather build on top of it. Some modeling base concepts, such as dimensional analysis and Galilean invariance in turbulence modeling should be preserved in the deep learning architecture. Numerous physics informed modeling have been reviewed in Section 4, such as Luo et al. for constant tuning [65], Ling et al. for turbulence model enhancement [62], and Gao et al. for super-resolution [78], etc. However, it is necessary to proceed with caution as sometimes it is not clear as to what type of physical information is useful for predictive outcomes [87]. Deep learning augmentations is often an extremely complex scenario, and the choice of what to physically inform specifically can be hard.

With the availability of large amount of data, deep learning models are also capable to find hidden correlations or equations that were unknown [96]. This can be extremely important since human preconception and knowledge can harm new discoveries instead of helping achieve them. For example forcing to fit deep learning models from RANS models when the closure is renowned to fail in multiple cases will certainly not allow great performance improvements. Instead, data driven models might even bypass the traditional ways of hypothesis-driven model creation and instead generate models free from human intuition [96]. In this regard, Font et al. [36] present a new set of Reynolds averaged equations and manage to make CFD-deep learning coupled simulation drastically faster than 3D simulation. A similar philosophy could also be applied in the built environment.

Another research field that directly interest built environment flow simulations is super-resolution. The ability of leveraging high-resolution data on a smaller domain to enhance the resolution of the system is very appealing in terms of computational efficiency. As shown by Gao et al. [78], it is possible to introduce physics prior knowledge into the algorithm for performance improvements. Moreover, the promising results obtained with oceanography task by Erichson et al. [75] makes super-resolution also applicable for observation tasks, tracking pollutants in urban cities at various level, and climate modeling. Specifically, CNNs are a very promising architecture for super-resolution which made huge step forward in the recent years.

Eventually, to provide direct quantitative comparison of general accuracy performance between different deep learning techniques reviewed in this paper is extremely hard if not unfeasible. This is because the deep learning architecture in every single application is applied in a different way. For example, the difference in training data sets comes not only from different distributions, but also from different sources, such as CFD simulations or experimental measurements. The accuracy of simple surrogate modeling and coefficient tuning is highly dependent on the training procedure. The corresponding ANNs show the highest struggle in adapting to cases where the ANN was not trained for, even for simple variables changes as seen in the work by Tanaka et al. [18]. Super-resolution and POD are in general less sensitive in this regard, but they still require a large training data set to achieve good accuracy. Turbulence model augmentations provide a tool to overcome the limitations of standard turbulence models and they should provide the highest generalizability and robustness in applications to different flow scenarios. Finally the addition of physical information with physics informed modeling can definitely help in improving overall performance of deep learning.

It is important to notice that the power of ANNs in specific tasks reside almost completely in their capacity to interpolate the data. Given a sufficiently big enough training data-set, the performance on the test set will be adequate if the test and training set are under similar distribution. The regression function is in fact well approximated only in the span (or under the probability distribution) of the sample data [97]. A big challenge for deep learning instead resides in the extrapolation process, where ANN can fail even in simple cases [75]. One obvious example of extrapolation task is when the data has the form of a time sequence and the objective of the deep learning algorithm is to inference about future predictions, given historical data as training data (crucial in climate modeling for instance). The specific failure of the deep learning algorithm developed by Erichson is shown in Fig. 10. In the top row, the true test snapshot of a future distribution after $t = 1$10(a), $t = 20$ 10(b) and $t = 50$ 10(c). In the bottom row, the super-resolved field after $t = 1$ 10(d), $t = 20$ 10(e) and $t = 50$ 10(f). The deep learning model fails in extrapolating the fields which belong to a different statistical distribution. Therefore, for transient indoor/outdoor airflow, the application of super-resolution requires further investigation.

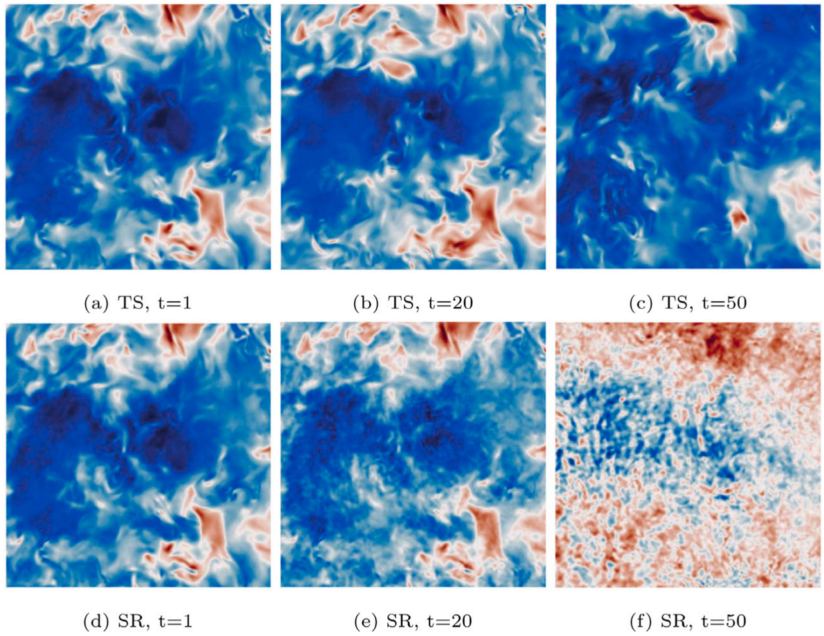


Fig. 10. Extrapolation task problem of the super-resolution study by Erichson et al. [75].

Directly linked to interpolation ability is the renowned ANN hunger for large data-set. In computer vision tasks, the available training data are usually massive. In fluid mechanics, the availability of data is still limited, despite various efforts in the latest years to create fluid mechanics databases of simulations, such as the John Hopkins turbulence database [98], or the DNS database of Li and Perlman [99]. The research community is still not used to consistently work with open and large shared databases and often prefer to generate simulation data themselves, especially in built environment, which eventually harm the development of better data driven models. Besides, the scarce availability of large data makes the classic over-fitting problem during the ANN training procedure even more relevant.

6. Conclusions

The objective of this study is to perform a comprehensive review of the current state of the art of the interaction between deep learning models and fluid mechanics simulations, update on the state of the built environment research in the topic, and propose possible advancements in the field. The vast majority of applications in fluid mechanics analysis in the built environment involves deep learning as surrogate modeling for faster predictions, which is justified by the expensiveness of CFD simulations especially in the design process. However, most often fast predictions comes at a cost of degraded accuracy. Fluid mechanics research and applications in general offer inspiration for possible different interaction which could benefit not only prediction speed but also accuracy performance. Above all, physics informed deep learning modeling, turbulence model enhancement with different techniques, and super-resolution techniques are the most promising methods that are largely yet to be explored in the built environment, for both indoor and outdoor simulations. Unfortunately, together with promising advancements, deep learning methods come with challenges to overcome, such as the availability of consistent large flow databases, the extrapolation task problem, over-fitting and others. The entire spectrum of possible interactions, opportunities, and challenges may not be captured in this paper despite our best efforts towards it. It is expected that this study will be useful to built environment researchers on deep learning models for flow simulations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was partially supported by the Digital Futures, C3.ai Digital Transformation Institute, and the Energimyndigheten (Swedish Energy Agency, grant No. 50057-1).

References

[1] R. Albers, P. Bosch, B. Blocken, A. Van Den Dobbelsteen, L. Van Hove, T. Spit, F. Van de Ven, T. Van Hooff, V. Rovers, Overview of challenges and achievements in the climate adaptation of cities and in the climate proof cities program, Elsevier, 2015.
[2] H. Bulkeley, M.M. Betsill, et al., Cities and Climate Change, vol. 382, Routledge London, 2013.
[3] D. Coley, T. Kershaw, Changes in internal temperatures within the built environment as a response to a changing climate, Build. Environ. 45 (1) (2010) 89–93.
[4] Z. Huijbregts, R. Kramer, M. Martens, A. Van Schijndel, H. Schellen, A proposed method to assess the damage risk of future climate change to museum objects in historic buildings, Build. Environ. 55 (2012) 43–56.
[5] M. Wang, Q. Chen, Assessment of various turbulence models for transitional flows in an enclosed environment (RP-1271), Hvac R Res. 15 (6) (2009) 1099–1119.
[6] A. Stamou, I. Katsiris, Verification of a CFD model for indoor airflow and heat transfer, Build. Environ. 41 (9) (2006) 1171–1181.
[7] J. Srebric, V. Vukovic, G. He, X. Yang, Cfd boundary conditions for contaminant dispersion, heat transfer and airflow simulations around human occupants in indoor environments, Build. Environ. 43 (3) (2008) 294–303.
[8] J. Liu, J. Niu, Cfd simulation of the wind environment around an isolated high-rise building: An evaluation of SRANS, LES and DES models, Build. Environ. 96 (2016) 91–106.
[9] T. van Druenen, T. van Hooff, H. Montazeri, B. Blocken, Cfd evaluation of building geometry modifications to reduce pedestrian-level wind speed, Build. Environ. 163 (2019) 106293.
[10] T. Lauriks, R. Longo, D. Baetens, M. Derudi, A. Parente, A. Bellemans, J. Van Beeck, S. Denys, Application of improved CFD modeling for prediction and mitigation of traffic-related air pollution hotspots in a realistic urban street, Atmos. Environ. 246 (2021) 118127.
[11] P. Moin, K. Mahesh, Direct numerical simulation: a tool in turbulence research, Annu. Rev. Fluid Mech. 30 (1) (1998) 539–578.

[12] J.W. Deardorff, et al., A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers, J. Fluid Mech. 41 (2) (1970) 453–480.
[13] S.B. Pope, Turbulent flows, IOP Publishing, 2001.
[14] F.G. Schmitt, About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity, C. R. Méc. 335 (9–10) (2007) 617–627.
[15] J. Slotnick, A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, D. Mavriplis, Cfd vision 2030 study: a path to revolutionary computational aerosciences, National Aeronautics and Space Administration, Langley Research Center, 2014.
[16] M.J. Berger, M.J. Aftosmis, D. Marshall, S.M. Murman, Performance of a new CFD flow solver using a hybrid programming paradigm, J. Parallel Distrib. Comput. 65 (4) (2005) 414–423.
[17] P.R. Spalart, A.V. Garbaruk, A new "λ 2" term for the spalart–allmaras turbulence model, active in axisymmetric flows, Flow Turbul. Combust. (2021) 1–12.
[18] H. Tanaka, Y. Matsuoka, T. Kawakami, Y. Azegami, M. Yamamoto, K. Ohtake, T. Sone, Optimization calculations and machine learning aimed at reduction of wind forces acting on tall buildings and mitigation of wind environment, Int. J High Rise Build. 8 (4) (2019) 291–302, http://dx.doi.org/10.21022/IJHRB.2019.8.4.291, URL https://doi.org/10.21022/IJHRB.2019.8.4.291.
[19] D. Hintea, J. Brusey, E. Gaura, A study on several machine learning methods for estimating cabin occupant equivalent temperature, in: Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics, 2015, p. nil, http://dx.doi.org/10.5220/0005573606290634, URL https://doi.org/10.5220/0005573606290634.
[20] T. hu Zhang, X. yi You, Applying neural networks to solve the inverse problem of indoor environment, Indoor Built Environ. 23 (8) (2013) 1187–1195, http://dx.doi.org/10.1177/1420326x13499596, URL https://doi.org/10.1177/1420326x13499596.
[21] K. Fukami, K. Fukagata, K. Taira, Super-resolution reconstruction of turbulent flows with machine learning, J. Fluid Mech. 870 (nil) (2019) 106–120, http://dx.doi.org/10.1017/jfm.2019.238, URL https://doi.org/10.1017/jfm.2019.238.
[22] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, http://www.deeplearningbook.org.
[23] A.F. Agarap, Deep learning using rectified linear units (relu), 2018, arXiv preprint arXiv:1803.08375.
[24] V. Dumoulin, F. Visin, A guide to convolution arithmetic for deep learning, 2018, arXiv:1603.07285.
[25] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366, http://dx.doi.org/10.1016/0893-6080(89)90020-8, URL https://doi.org/10.1016/0893-6080(89)90020-8.
[26] Y. LeCun, Y. Bengio, et al., Convolutional networks for images, speech, and time series, Handb. Brain Theory Neural Netw. 3361 (10) (1995) 1995.
[27] L.R. Medsker, L. Jain, Recurrent neural networks, Des. Appl. 5 (2001).
[28] A. Beck, D. Flad, C.-D. Munz, Deep neural networks for data-driven les closure models, J. Comput. Phys. 398 (nil) (2019) 108910, http://dx.doi.org/10.1016/j.jcp.2019.108910, URL https://doi.org/10.1016/j.jcp.2019.108910.
[29] J.-S. Zhang, X.-C. Xiao, Predicting chaotic time series using recurrent neural network, Chin. Phys. Lett. 17 (2) (2000) 88.
[30] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, A field guide to dynamical recurrent neural networks. IEEE Press, 2001.
[31] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[32] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: Theory and applications, Neurocomputing 70 (1–3) (2006) 489–501, http://dx.doi.org/10.1016/j.neucom.2005.12.126, URL https://doi.org/10.1016/j.neucom.2005.12.126.
[33] K. Fukami, K. Fukagata, K. Taira, Assessment of supervised machine learning methods for fluid flows, Theor. Comput. Fluid Dyn. 34 (4) (2020) 497–519, http://dx.doi.org/10.1007/s00162-020-00518-y, URL https://doi.org/10.1007/s00162-020-00518-y.
[34] M. Awad, R. Khanna, Support vector regression, in: Efficient Learning Machines, Springer, 2015, pp. 67–80.
[35] T.K. Ho, Random decision forests, in: Proceedings of 3rd International Conference on Document Analysis and Recognition, 1, IEEE, 1995, pp. 278–282.
[36] B. Font, G.D. Weymouth, V.-T. Nguyen, O.R. Tutty, Deep learning of the spanwise-averaged Navier-Stokes equations, J. Comput. Phys. 434 (nil) (2021) 110199, http://dx.doi.org/10.1016/j.jcp.2021.110199, URL https://doi.org/10.1016/j.jcp.2021.110199.
[37] W. Liu, R. Duan, C. Chen, C.-H. Lin, Q. Chen, Inverse design of the thermal environment in an airliner cabin by use of the cfd-based adjoint method, Energy Build. 104 (nil) (2015) 147–155, http://dx.doi.org/10.1016/j.enbuild.2015.07.011, URL https://doi.org/10.1016/j.enbuild.2015.07.011.
[38] J. Wang, T.T. Zhang, H. Zhou, S. Wang, Inverse design of aircraft cabin environment using computational fluid dynamics-based proper orthogonal decomposition method, Indoor Built Environ. 27 (10) (2017) 1379–1391, http://dx.doi.org/10.1177/1420326x17718053, URL https://doi.org/10.1177/1420326x17718053.
[39] S.L. Wong, K.K. Wan, T.N. Lam, Artificial neural networks for energy analysis of office buildings with daylighting, Appl. Energy 87 (2) (2010) 551–557.
[40] R. Yokoyama, T. Wakui, R. Satake, Prediction of energy demands using neural network with model identification by global optimization, Energy Convers. Manage. 50 (2) (2009) 319–327.
[41] S.A. Kalogirou, Artificial neural networks in energy applications in buildings, Int. J. Low Carbon Technol. 1 (3) (2006) 201–216.
[42] T. Wei, Y. Wang, Q. Zhu, Deep reinforcement learning for building HVAC control, in: Proceedings of the 54th Annual Design Automation Conference 2017, 2017, pp. 1–6.
[43] Z. Wang, T. Hong, M.A. Piette, Data fusion in predicting internal heat gains for office buildings through a deep learning approach, Appl. Energy 240 (2019) 386–398.
[44] T. hu Zhang, X. yi You, A simulation-based inverse design of preset aircraft cabin environment, Build. Environ. 82 (nil) (2014) 20–26, http://dx.doi.org/10.1016/j.buildenv.2014.08.002, URL https://doi.org/10.1016/j.buildenv.2014.08.002.
[45] A. Warey, S. Kaushik, B. Khalighi, M. Cruse, G. Venkatesan, Data-driven prediction of vehicle cabin thermal comfort: Using machine learning and high-fidelity simulation results, Int. J. Heat Mass Transfer 148 (nil) (2020) 119083, http://dx.doi.org/10.1016/j.ijheatmasstransfer.2019.119083, URL https://doi.org/10.1016/j.ijheatmasstransfer.2019.119083.
[46] C. Ding, K.P. Lam, Data-driven model for cross ventilation potential in high-density cities based on coupled CFD simulation and machine learning, Build. Environ. 165 (2019) 106394.
[47] S. Carlucci, L. Bai, R. de Dear, L. Yang, Review of adaptive thermal comfort models in built environmental regulatory documents, Build. Environ. 137 (2018) 73–89.
[48] P. Kutler, U. Mehta, Computational aerodynamics and artificial intelligence, in: 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, 1984, p. nil, http://dx.doi.org/10.2514/6.1984-1531, URL https://doi.org/10.2514/6.1984-1531.
[49] C. Teo, K. Lim, G. Hong, M. Yeo, A neural net approach in analyzing photograph in PIV, in: Conference Proceedings 1991 IEEE International Conference on Systems, Man, and Cybernetics, 1991, p. nil, http://dx.doi.org/10.1109/icsmc.1991.169906, URL https://doi.org/10.1109/icsmc.1991.169906.
[50] C. Lee, J. Kim, D. Babcock, R. Goodman, Application of neural networks to turbulence control for drag reduction, Phys. Fluids 9 (6) (1997) 1740–1747, http://dx.doi.org/10.1063/1.869290, URL https://doi.org/10.1063/1.869290.
[51] J. Thibault, B.P. Grandjean, A neural network methodology for heat transfer data analysis, Int. J. Heat Mass Transfer 34 (8) (1991) 2063–2070.
[52] L. Zhang, M. Akiyama, K. Huang, H. Sugiyama, N. Ninomiya, Estimation of flow patterns by applying artificial neural networks, in: 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929), 2, IEEE, 1996, pp. 1358–1363.
[53] M.M. Rai, N.K. Madavan, Application of artificial neural networks to the design of turbomachinery airfoils, J. Propul. Power 17 (1) (2001) 176–183.
[54] Y. He, C. Rutland, Application of artificial neural networks in engine modelling, Int. J. Engine Res. 5 (4) (2004) 281–296.
[55] R. Duvigneau, M. Visonneau, Hybrid genetic algorithms and neural networks for fast CFD-based design, in: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, 2002, pp. 5465.
[56] M. Milano, P. Koumoutsakos, Neural network modeling for near wall turbulent flow, J. Comput. Phys. 182 (1) (2002) 1–26, http://dx.doi.org/10.1006/jcph.2002.7146, URL https://doi.org/10.1006/jcph.2002.7146.
[57] G. Berkooz, P. Holmes, J. Lumley, The proper orthogonal decomposition in the analysis of turbulent flows, Annu. Rev. Fluid Mech. 25 (1) (1993) 539–575, http://dx.doi.org/10.1146/annurev.fl.25.010193.002543, URL https://doi.org/10.1146/annurev.fl.25.010193.002543.
[58] S. Yarlanki, B. Rajendran, H. Hamann, Estimation of turbulence closure coefficients for data centers using machine learning algorithms, in: 13th InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems, IEEE, 2012, http://dx.doi.org/10.1109/itherm.2012.6231411, URL http://dx.doi.org/10.1109/ITHERM.2012.6231411.
[59] S. Luo, M. Vellakal, S. Koric, V. Kindratenko, J. Cui, Parameter identification of RANS turbulence model using physics-embedded neural network, in: Lecture Notes in Computer Science, in: Lecture Notes in Computer Science, Springer International Publishing, 2020, pp. 137–149, http://dx.doi.org/10.1007/978-3-030-59851-8_9, URL https://doi.org/10.1007/978-3-030-59851-8_9.
[60] B.D. Tracey, K. Duraisamy, J.J. Alonso, A machine learning strategy to assist turbulence model development, in: 53rd AIAA Aerospace Sciences Meeting, 2015, p. nil, http://dx.doi.org/10.2514/6.2015-1287, URL https://doi.org/10.2514/6.2015-1287.
[61] A.P. Singh, S. Medida, K. Duraisamy, Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils, AIAA J. 55 (7) (2017) 2215–2227, http://dx.doi.org/10.2514/1.j055595, URL https://doi.org/10.2514/1.j055595.
[62] J. Ling, A. Kurzawski, J. Templeton, Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, J. Fluid Mech. 807 (nil) (2016) 155–166, http://dx.doi.org/10.1017/jfm.2016.615, URL https://doi.org/10.1017/jfm.2016.615.
[63] N. Geneva, N. Zabaras, Quantifying model form uncertainty in Reynolds-averaged turbulence models with Bayesian deep neural networks, J. Comput. Phys. 383 (nil) (2019) 125–147, http://dx.doi.org/10.1016/j.jcp.2019.01.021, URL https://doi.org/10.1016/j.jcp.2019.01.021.

[64] J.-X. Wang, J.-L. Wu, H. Xiao, Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on dns data, Phys. Rev. Fluids 2 (3) (2017) 034603, http://dx.doi.org/10.1103/physrevfluids.2.034603, URL https://doi.org/10.1103/physrevfluids.2.034603.

[65] X. Guo, W. Li, F. Iorio, Convolutional neural networks for steady flow approximation, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, p. nil, http://dx.doi.org/10.1145/2939672.2939738, URL https://doi.org/10.1145/2939672.2939738.

[66] R. Maulik, H. Sharma, S. Patel, B. Lusch, E. Jennings, Accelerating RANS turbulence modeling using potential flow and machine learning, 2019, arXiv preprint arXiv:1910.10878.

[67] P. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, R. Vinuesa, Predictions of turbulent shear flows using deep neural networks, Phys. Rev. Fluids 4 (5) (2019) 054603, http://dx.doi.org/10.1103/physrevfluids.4.054603, URL https://doi.org/10.1103/physrevfluids.4.054603.

[68] L. Guastoni, M.P. Encinar, P. Schlatter, H. Azizpour, R. Vinuesa, Prediction of wall-bounded turbulence from wall quantities using convolutional neural networks, J. Phys. Conf. Ser. 1522 (nil) (2020) 012022, http://dx.doi.org/10.1088/1742-6596/1522/1/012022, URL https://doi.org/10.1088/1742-6596/1522/1/012022.

[69] L. Guastoni, P.A. Srinivasan, H. Azizpour, P. Schlatter, R. Vinuesa, On the use of recurrent neural networks for predictions of turbulent flows, 2020, arXiv preprint arXiv:2002.01222.

[70] L. Guastoni, A. Güemes, A. Ianiro, S. Discetti, P. Schlatter, H. Azizpour, R. Vinuesa, Convolutional-network models to predict wall-bounded turbulence from wall quantities, 2020, arXiv preprint arXiv:2006.12483.

[71] S. Fresca, A. Manzoni, Pod-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition, 2021, arXiv preprint arXiv:2101.11845.

[72] Z. Wang, D. Xiao, F. Fang, R. Govindan, C.C. Pain, Y. Guo, Model identification of reduced order fluid dynamics systems using deep learning, Internat. J. Numer. Methods Fluids 86 (4) (2018) 255–268.

[73] A.T. Mohan, D.V. Gaitonde, A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks, 2018, arXiv preprint arXiv:1804.09269.

[74] H.F. Lui, W.R. Wolf, Construction of reduced-order models for fluid flows using deep feedforward neural networks, J. Fluid Mech. 872 (nil) (2019) 963–994, http://dx.doi.org/10.1017/jfm.2019.358, URL https://doi.org/10.1017/jfm.2019.358.

[75] N.B. Erichson, L. Mathelin, Z. Yao, S.L. Brunton, M.W. Mahoney, J.N. Kutz, Shallow neural networks for fluid flow reconstruction with limited sensors, Proc. R. Soc. A 476 (2238) (2020) 20200097, http://dx.doi.org/10.1098/rspa.2020.0097, URL https://doi.org/10.1098/rspa.2020.0097.

[76] K. Fukami, K. Fukagata, K. Taira, Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows, J. Fluid Mech. 909 (2021).

[77] B. Liu, J. Tang, H. Huang, X.-Y. Lu, Deep learning methods for super-resolution reconstruction of turbulent flows, Phys. Fluids 32 (2) (2020) 025105.

[78] H. Gao, L. Sun, J.-X. Wang, Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels, 2020, arXiv preprint arXiv:2011.02364.

[79] Z. Deng, C. He, Y. Liu, K.C. Kim, Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework, Phys. Fluids 31 (12) (2019) 125111.

[80] K. Bai, W. Li, M. Desbrun, X. Liu, Dynamic upsampling of smoke through dictionary-based learning, ACM Trans. Graph. 40 (1) (2020) 1–19.

[81] G.H. Hoffman, Improved form of the low Reynolds number k- ε turbulence model, Phys. Fluids 18 (3) (1975) 309–312.

[82] B. Launder, A. Morse, W. Rodi, D. Spalding, Prediction of free shear flows: a comparison of the performance of six turbulence models, Int. J. Heat Mass Transfer (1973) http://dx.doi.org/10.1016/0017-9310(73)90125-7.

[83] B.E. Launder, D.B. Spalding, The numerical computation of turbulent flows, in: Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion, Elsevier, 1983, pp. 96–116.

[84] M. Iyengar, R.R. Schmidt, H. Hamann, J. VanGilder, Comparison between numerical and experimental temperature distributions in a small data center test cell, in: International Electronic Packaging Technical Conference and Exhibition, 42770, 2007, pp. 819–826.

[85] B. Tracey, K. Duraisamy, J. Alonso, Application of supervised learning to quantify uncertainties in turbulence and combustion modeling, in: 51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2013, p. nil, http://dx.doi.org/10.2514/6.2013-259, URL https://doi.org/10.2514/6.2013-259.

[86] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, 1992, p. 439.

[87] K. Duraisamy, Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence, Phys. Rev. Fluids 6 (5) (2021) 050504.

[88] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. 51 (1) (2019) 357–377, http://dx.doi.org/10.1146/annurev-fluid-010518-040547, URL http://dx.doi.org/10.1146/annurev-fluid-010518-040547.

[89] S. Rudy, A. Alla, S.L. Brunton, J.N. Kutz, Data-driven identification of parametric partial differential equations, SIAM J. Appl. Dyn. Syst. 18 (2) (2019) 643–660.

[90] D.J. Lucia, P.S. Beran, W.A. Silva, Reduced-order modeling: new approaches for computational physics, Prog. Aerosp. Sci. 40 (1–2) (2004) 51–117.

[91] W. Zhang, B. Wang, Z. Ye, High efficient numerical method for limit cycle flutter analysis based on nonlinear aerodynamic reduced order model reduced order model, in: 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 18th AIAA/ASME/AHS Adaptive Structures Conference 12th, 2010, p. 2723.

[92] P. Schlatter, R. Örlü, Q. Li, G. Brethouwer, J.M. Fransson, A. Johansson, P. Alfredsson, D. Henningson, Turbulent boundary layers up to reθ=2500 studied through simulation and experiment, Phys. Fluids 21 (5) (2009) 051702, http://dx.doi.org/10.1063/1.3139294, URL https://doi.org/10.1063/1.3139294.

[93] F. Ballarin, A. Manzoni, A. Quarteroni, G. Rozza, Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations, Internat. J. Numer. Methods Engrg. 102 (5) (2015) 1136–1161.

[94] A. Mohan, D. Daniel, M. Chertkov, D. Livescu, Compressed convolutional LSTM: An efficient deep learning framework to model high fidelity 3D turbulence, 2019, arXiv preprint arXiv:1903.00033.

[95] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, Q. Liao, Deep learning for single image super-resolution: a brief review, IEEE Trans. Multimed. 21 (12) (2019) 3106–3121, http://dx.doi.org/10.1109/tmm.2019.2919431, URL https://doi.org/10.1109/tmm.2019.2919431.

[96] K. Duraisamy, P.R. Spalart, C.L. Rumsey, Status, emerging ideas and future directions of turbulence modeling research in aeronautics, National Aeronautics and Space Administration, Langley Research Center, 2017.

[97] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. 52 (2020) 477–508.

[98] K. Kanov, R. Burns, C. Lalescu, G. Eyink, The johns hopkins turbulence databases: an open simulation laboratory for turbulence research, Comput. Sci. Eng. 17 (5) (2015) 10–17.

[99] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, G. Eyink, A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence, J. Turbul. (9) (2008) N31.