Applied Energy 277 (2020) 115552

Contents lists available at ScienceDirect

Applied Energy

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

A novel dynamic wind farm wake model based on deep learning

Jincheng Zhang, Xiaowei Zhao*

School of Engineering, University of Warwick, Coventry, UK

HIGHLIGHTS

- A novel dynamic wind farm wake model is developed based on deep learning.
- This model can predict unsteady flow features while running as fast as static model.
- A ROM method for unsteady distributed fluid systems is proposed to build the model.
- A high-fidelity CFD database for wind farm is generated for model training.

ARTICLE INFO

Keywords:
CFD simulation
Deep learning
Dynamic wake model
Reduced order modelling
Wind farm control

ABSTRACT

A deep learning based reduced order modelling method for general unsteady fluid systems is proposed, which is then applied to develop a novel dynamic wind farm wake model. The proposed method employs the proper orthogonal decomposition technique for reducing the flow field dimension and the long short-term memory network for predicting the reduced representation of the flow field at a future time step. The method is specifically designed to tackle distributed fluid systems (such as wind farm wakes) and to be control-oriented. For wind farm wake modelling, a set of large eddy simulations are first carried out to generate a series of flow field data for wind turbines operating in different conditions. Then the proposed method is employed to develop the data-based wake model. The results show that this novel dynamic wind farm wake model can predict the main features of unsteady wind turbine wakes similarly as high-fidelity wake models while running as fast as the low-fidelity static wake models and that the model's overall prediction error is just 4.8% with respect to the free-stream wind speed. As an illustrative example, the developed model can predict the unsteady turbine wakes of a 9-turbine test wind farm within several seconds based on a standard desktop while it requires tens of thousands of CPU hours on a high-performance computing cluster if a high-fidelity model is used. Thus the developed model can be used for fast yet accurate simulation of wind farms as well as for their predictions and control designs.

1. Introduction

Wind turbines are usually grouped to form a wind farm in order to reduce the overall cost of wind power harvesting. However, the downstream turbines' performance can be significantly undermined if they operate in the wakes generated by the upstream turbines. The turbine rotor wakes are characterized by reduced wind speed and an increased turbulence level, thus the turbines operating in the wakes usually generate less power and experience more severe structural loads than the ones operating in freestream wind. For example, the experimental investigation [1] showed that the downstream turbine's power loss due to the wake effects could be up to 46% compared to the power generation in the designed wind condition. In order to mitigate the wake effects, turbine layout is usually optimized during the design phase while various control techniques are proposed for the operation phase to steer the wake away from the downstream turbines, which include turbine yaw control, individual pitch control, and tilt-based control [2]. However, the design and evaluation of wind farm controllers are very challenging due to the lack of an accurate yet fast and efficient turbine wake model to take account of the wake dynamics. Thus such model is urgently needed for the optimal control design of wind farms. Besides, it can also be used for monitoring wind farms and predicting their extreme and fatigue load and electricity generations, through fast simulation.

Extensive research efforts have been spent on wind farm wake modelling. There exist various wake models in the literature [3], including low-fidelity, medium-fidelity and high-fidelity models. The most widely used wake models are the analytical low-fidelity models,

* Corresponding author.
E-mail addresses: jincheng.zhang@warwick.ac.uk (J. Zhang), xiaowei.zhao@warwick.ac.uk (X. Zhao).

https://doi.org/10.1016/j.apenergy.2020.115552
Received 16 April 2020; Received in revised form 19 June 2020; Accepted 16 July 2020
Available online 30 July 2020

such as the Jensen Park model [4,5], the Frandsen model [6], the FLORIS model [7], the 3D wake model [8,9], and the models developed in [10,11]. These models are formulated analytically and are very fast to evaluate, which makes them suitable for wind farm layout optimization. The further development of analytical models is still an active area, such as taking into account the turbulence effect [12,13], modelling yaw effects [14–16], modelling background flow effects [17], considering the expansion of physical wake boundary [18], and incorporating uncertainty based on high-fidelity data [19]. However, as these models are static, they are mainly used in turbine layout design for optimizing static quantities, such as mean power generation. In order to achieve wind farm design with consideration of unsteady quantities such as power fluctuation and structural load, a dynamic model that can capture the unsteady wakes is needed. More importantly, as static models are not suitable for the control design of wind farms, a fast, efficient, and accurate dynamic model is of great interest to the wind farm control community.

Currently, most investigations of unsteady wakes are carried out using numerical simulations, such as the studies of the Lillgrund wind farm in [20] and the Horns Rev offshore wind farm in [21]. For numerical simulations, turbine rotors were usually modelled by the actuator disk method (ADM) [22,23] or the actuator line method (ALM) [24,25]. The comparisons of the ADM and ALM methods in wind farm simulations were also investigated, by using the PALM model [26], the UTDWF model [27], and the model developed in [28]. The further development of wind farm solvers is still an active area, such as the Nalu-Wind solver in [29]. Although these high-fidelity models can capture detailed wake dynamics, such as wake recovery and wake meandering, they are expensive to run. For instance, in [7] about 60 h of distributed computation with 512 processors on high-performance computing (HPC) clusters were used for 1000s large eddy simulation (LES) of a 3 km × 3 km wind farm with 6 turbines. The requirement of long simulation time and enormous computing resources makes high-fidelity models not suitable for control design purposes. In the existing literature, there are also a few medium-fidelity dynamic models, such as the dynamic wake meandering (DWM) in [30], WFSim in [31], and the continuous-time dynamic wake model in [32]. The development of such control-oriented dynamic wake models is becoming very active now. In this work, contrary to [30–32] which are based on assumption and approximation from physical observation, we develop a machine learning based reduced order modelling (ROM) method to build a novel wake model that can be evaluated as fast as low-fidelity static models while capturing the unsteady wake details similarly as high-fidelity models. With the fast development of machine learning, in particular deep learning [33], this work paves the way for developing novel wake models using advanced machine learning techniques based on high-fidelity flow field data.

Recently non-intrusive ROM using machine learning is attracting more and more research attention in fluid dynamics [34], including the investigations of both steady [35] and unsteady flow problems [36,37]. One approach is to directly formulate the ROM as a supervised machine learning problem to train a model with the flow parameters as training input and the full flow field as training output [38,39]. This kind of approach makes use of the-state-of-art machine learning algorithms, which can thus mimic the fluid system to high accuracy if there are enough training data available. Another approach is to first reduce the flow field dimension by a dimensionality reduction technique, and then formulate a supervised machine learning problem to predict the reduced coefficients instead of the full flow field, with the flow parameters as input. Such studies include the ROM of both steady [40,41] and unsteady [42,43] flow problems. In this way, the trained model can capture the main dynamics of the fluid systems while alleviating the need for a large amount of high-fidelity training data.

A ROM method for unsteady distributed fluid systems is proposed in this work. Distributed fluid systems are quite common in daily life and industrial applications, such as the natural convection of heater array in heat exchangers [44,45], the heat transfer of building array in turbulent boundary layer [46], and the wake interactions of wind turbines within a wind farm [20,21]. In the proposed ROM method, the high-dimensional flow field data is first reduced to low-dimensional coefficients by a dimensionality reduction technique called proper orthogonal decomposition (POD) [47]. Then a deep recurrent neural network (RNN), called long-short term memory (LSTM) [33,48], is employed to predict the reduced coefficients at current time step based on the reduced coefficients in the previous time steps. POD is chosen as it can capture the main flow features according to the flow field's energy content while LSTM is chosen as it is very powerful in handling time-series predictions. The proposed method can be used for the ROM of general unsteady fluid systems but is not directly suitable for distributed fluid systems. The present paper handles this challenge by including the flow boundary conditions in the model input to enable the constructed reduced order model to predict the flow field around distributed structures of different layouts/scales. The proposed method is hereby referred as POD-LSTM.

The application of the POD-LSTM method to wind farm wake modelling is then investigated. A series of large eddy simulations are carried out with wind turbine rotors operating in different yaw conditions under different turbulent inflows. Then the machine learning model is trained to learn the wake dynamics from the generated LES database. The results show that the so-constructed reduced order dynamic wake model can be evaluated as fast as low-fidelity static models while capturing the main unsteady flow features (such as the streamwise convection of flow structures, the wake meandering, the turbine's yaw effects, and the wake interactions between wind turbines) similarly as high-fidelity dynamic models. Therefore, this data-based model can be used for fast wind farm simulations, predictions and control design.

The main contributions of this paper are summarized as follows:

(i) A novel data-based dynamic wind farm wake model is developed and validated through a series of simulation tests including single turbine wakes, multiple turbine wakes, yawed wakes, and wake interactions within a large wind farm. As the existing wake models in the literature are either unable to predict unsteady wake details (low-fidelity models) or too time-consuming to run (high-fidelity models), this work bridges the research gap by developing a wake model that can predict unsteady wind turbine wakes similarly as high-fidelity wake models while running as fast as the low-fidelity static wake models. The comparison of the developed model with existing wake models is summarised in Table 1.
(ii) A deep learning based ROM method, called POD-LSTM, is proposed to build this novel wake model. As the existing ROM methods in the literature are not suitable for distributed systems, this work fills the research gap by proposing a ROM method specifically designed for distributed systems. The proposed method is generic and can also be used to model other distributed fluid systems, such as tidal farms and building arrays in the atmospheric boundary layer.
(iii) A high-fidelity CFD database of wind flow around turbine rotors is

Table 1
The comparison of the developed machine learning based wake model with existing wake models.

Type	Analytical	Numerical	Machine learning
Models	Jensen, Frandsen, FLORIS, 3D wake, …	PALM, UTDWF, Nalu-Wind, SOWFA, …	The model developed in this paper
Based on	flow observations	NS equations	LES database
Method	flow analysis	CFD	deep learning
Speed	fast	slow	fast
Accuracy	low	high	moderate/high
Flow details	no	yes	yes

Table 2
The main terminologies (including abbreviations, parameters and variables) mentioned in this paper.

List of terminologies (abbreviations)			
ADM	Actuator disk method	NREL	National Renewable Energy Laboratory
ALM	Actuator line method	NS	Navier–Stokes
CFD	Computational fluid dynamics	POD	Proper orthogonal decomposition
DWM	Dynamic wake meandering	RMSE	Root-mean-squared-errors
FSTI	Freestream turbulence intensity	RNN	Recurrent neural network
HPC	High-performance computing	ROM	Reduced order modelling
LES	Large eddy simulation	SCRTP	Scientific Computing Research Technology Platform
LSTM	Long-short term memory	SOWFA	Simulator fOr Wind Farm Applications
MSE	Mean-squared error	SVD	Singular value decomposition
List of terminologies (parameters and variables)			
d	The dimension of the spatial domain	u_i^0	The inflow velocity at i^{th} time step
d_i	The i^{th} distributed control parameter	u_i^f	The reduced coefficient at i^{th} time step
$\mathscr{D}$	The matrix of designed input variables	U	The matrix of flow snapshots
M	The number of distributed structures	$\mathscr{U}$	The matrix of all the flow fields
N_h	The LSTM cell's output feature dimension	$\mathscr{U}^r$	The reduced representation of all the flow fields
N_t	The number of time steps in LES simulations	$\mathscr{U}^{POD}$	The flow fields estimated by exact POD coefficients
N_x	The number of points in each subdomain	$\widehat{\mathscr{U}}$	The flow field predicted by POD-LSTM
P	The dimension of each distributed parameter	v_k	The k^{th} POD basis
R	The number of POD basis	$\mathscr{X}$	The spatial coordinates of all the grid points
S	The number of simulations in the LES database	α	The dropout rate of the LSTM cell
T	The number of lookback time steps	α_k	The k^{th} POD coefficient
T_{tot}	The number of prediction steps	ϵ_{POD}	The POD model reduction error
$\widetilde{u}$	The true value of a flow snapshot	ϵ_{total}	The total prediction error
$\widehat{u}$	The approximate value of a flow snapshot		

generated, through a series of large eddy simulations which takes around 7×10^5 CPU hours' run time on local HPC clusters. The above deep learning based dynamic wind farm wake model is then trained to learn the complex wind farm wake dynamics from this valuable database.

The remaining part of this paper is organized as follows: the proposed deep learning based ROM method is described in Section 2. Its application on wind farm wake modelling is described in Section 3. After developing the wake model, the model validation and prediction are carried out in Section 4, where the prediction results on wakes behind a turbine with changing yaw and wake interactions in a 9-turbine wind farm are demonstrated. Finally, the conclusions are drawn in Section 5. The main terminologies mentioned in this paper are presented in Table 2.

2. Methodology

An example distributed fluid system, a wind farm, is illustrated in Fig. 1, where M distributed structures (wind turbines in this example) with distributed control parameters $[d_1, d_2, ..., d_M]$ (such as the yaw angle and blade pitch angle for a wind turbine) are shown in the rectangular flow domain. With a CFD approach, a mesh is first generated for the whole flow domain and then the discretized governing equation (such as the Navier–Stokes equations) is solved on the mesh. This approach is usually costly as the degree of freedom (the number of cells) is very high. In this section, a machine learning based ROM approach is developed to build a reduced order model that can predict the flow field efficiently given the distributed control parameters (wind turbines' operating parameters $[d_1, d_2, ..., d_M]$ in this example).

A ROM approach is to generate a set of flow field data using CFD, followed by training a model that takes all the distributed control parameters as the input to predict the whole flow field as the output. There are two fundamental flaws in this approach: i) after training, the model can only predict the flow field of a fixed layout and scale. ii) it is not feasible to generate the training data for the prediction of a fluid system containing many distributed structures, due to the curse of dimensionality (a large number of CFD simulations are required to cover the input parameter space which includes all the distributed control parameters).

These challenges are addressed in the proposed ROM approach here. The main idea is to first divide the whole fluid domain into

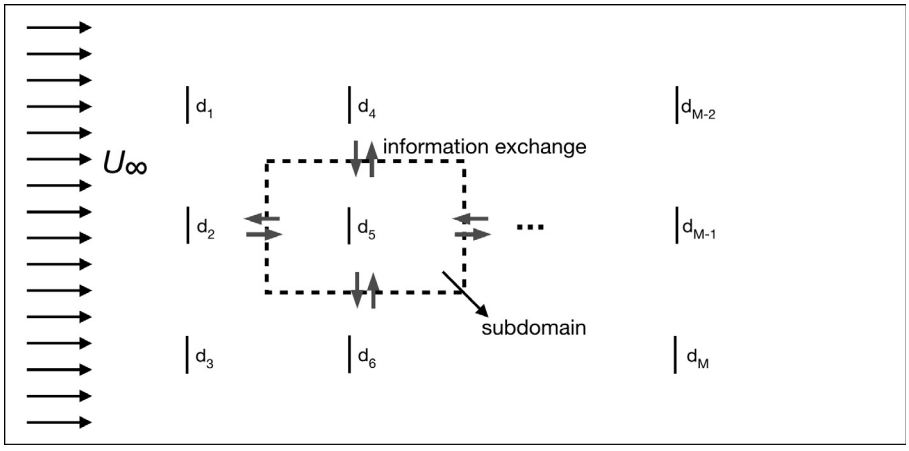


Fig. 1. A typical example of a distributed fluid system, where a subdomain containing one distributed structure is illustrated by the dashed rectangular.

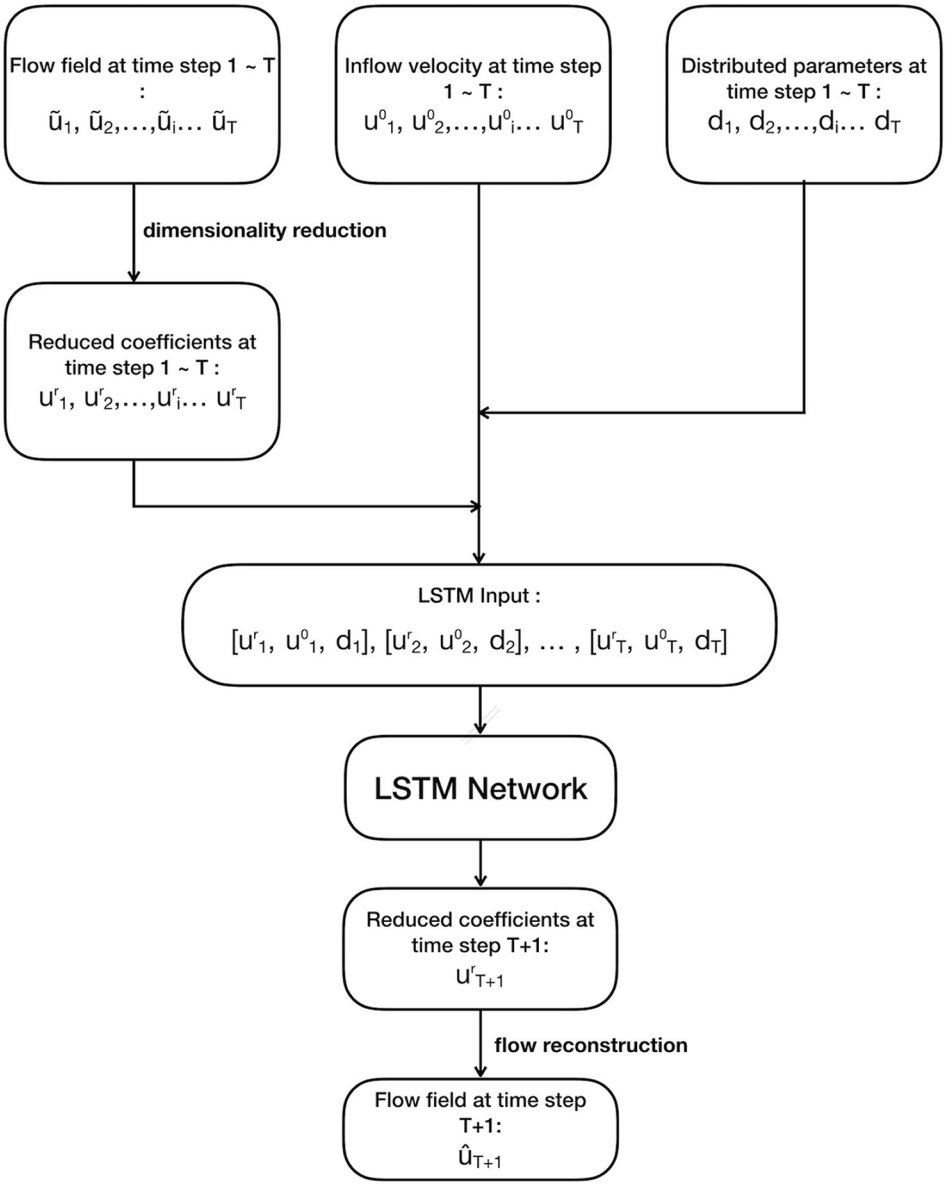


Fig. 2. An illustration of the data pipeline of the proposed POD-LSTM method.

individual subdomains (as shown by the dashed rectangular in Fig. 1), then carry out the ROM for each subdomain with only one single distributed control parameter included as model input, and finally obtain the whole flow field prediction by combining all the subdomain predictions. One missing ingredient is how to take account of the interactions between subdomains, which depends on whether the problem at hand is convection-dominant or diffusion-dominant. Here, the convection-dominant fluid problems are of our interest, which are omnipresent in industrial applications and natural phenomena. For this kind of problem, the impact of downstream structures on upstream flow can be ignored. Thus the upstream flow information can be directly used to impose the inflow condition for the downstream subdomain. The ROM for the whole domain can thus be carried out for each subdomain sequentially from upstream to downstream. The proposed ROM procedure includes four steps as shown below.

2.1. Design of experiments

This step serves to generate/collect high-fidelity training data. For a system with M distibuted structures as in Fig. 1, a set of design input variables, denoted as $\mathscr{D}$, are generated according to a sampling strategy (such as Latin hypercube sampling), where $\mathscr{D}$ is a matrix of shape $[M, N_t, P]$ with $\mathscr{D}_{m,i_t,p}$ representing the design value of the p^{th} control parameter of the m^{th} distributed structure at time step i_t. N_t is the total number of time steps and P is the dimension of each distributed control parameter (for example, $P = 2$ for the wind turbine case if two control parameters, e.g., turbine yaw angle and turbine blade pitch angle, are considered). The corresponding output variables, the flow field data $\mathscr{U}$, are obtained by running CFD or wind tunnel experiment with the designed input $\mathscr{D}$. Here $\mathscr{U}$ is a matrix of shape $[M, N_t, N_x]$ with $\mathscr{U}_{m,i_t,i_x}$ representing the value of the flow velocity magnitude in the m^{th} subdomain at spatial coordinate indexed by i_x at time step i_t. N_x is the total number of grid points in each subdomain. A matrix $\mathscr{X}$ of shape $[N_x, d]$ is used to record the location of all the grid points within a subdomain relative to the corresponding distributed structure location, where $\mathscr{X}_{i_x,1:d}$ represents the d-dimension spatial coordinate (for example $d = 2$ for 2-D plane) indexed by i_x.

In order to expand the training dataset to include more scenarios, a set of CFD/experiments can be carried out with different inflow conditions and design input variables. All the resulting data can then be

collected and reshaped together as the final training dataset. If in total S simulations/experiments are carried out, the final design input matrix $\mathscr{D}$ is of shape $[M \times S, N_t, P]$ and the design output matrix $\mathscr{U}$ containing all the flow field data is of shape $[M \times S, N_t, N_x]$.

2.2. Dimensionality reduction

Here, the design output matrix $\mathscr{U}$ obtained in previous step is first reshaped into a matrix U of shape $[M \times S \times N_t, N_x]$, with each row in U representing a snapshot of the flow field. The POD as in [47] is done by the singular value decomposition (SVD) as

$$U^T = V\Sigma W^T, \tag{1}$$

where the k^{th} column vector of V, denoted as v_k, is the k^{th} POD basis. Setting the total number of POD basis as R, each snapshot of the flow field, $\widetilde{u}$, can then be approximated by

$$\widehat{u} = \sum_{k=1}^{R} \alpha_k v_k, \tag{2}$$

where the reduced coefficients α_k are calculated by

$$\alpha_k = < \widetilde{u}, v_k >, 1 \leqslant k \leqslant R. \tag{3}$$

In this way, the output matrix $\mathscr{U}$ can be reduced into its reduced representation $\mathscr{U}^r$ of shape $[M \times S, N_t, R]$, where $\mathscr{U}^r_{m,i_t,r}$ represents the r^{th} reduced coefficient of the flow field $\mathscr{U}_{m,i_t,1:N_x}$. This dimensionality reduction process thus reduces the original flow field dimension from N_x to R. We mention that other dimensionality reduction techniques can also be used in the proposed ROM framework. The independent component analysis [49] and the auto-encoder [50] have been implemented and tested for the applications in this work, and the results are omitted here as their performances are not as good as POD.

2.3. Neural network training

After dimensionality reduction, a supervised machine learning problem is formulated to predict the reduced coefficients at the current time step based on historical data of the flow. The LSTM network, which is particularly powerful in time-series predictions, is employed here.

The overall data pipeline is illustrated in Fig. 2, where the flow field, the inflow velocity, and the distributed control parameters from time steps 1 to T are required as the input in order to predict the flow field at time step $T + 1$. The flow fields from 1 to T are first reduced to their POD coefficients, which are then fed into the LSTM network along with the inflow and distributed control parameter history to predict the POD coefficients at time step $T + 1$. Finally the flow field at time step $T + 1$ is reconstructed based on the predicted POD coefficients according to Eq. (2). The LSTM network in Fig. 2 is detailed in Fig. 3 which shows that the POD coefficients, the inflow velocity, and the distributed control parameters' value are standardized by the standard scalers before being fed into the LSTM cells. The scalers (denoted as Scaler1 and Scaler2 respectively in Fig. 3)) standardize the input features (the POD coefficients and the rest features respectively) by removing their mean and scaling to unit variance. A dense layer is stacked on top of the LSTM cells to predict the standardized POD coefficients at the next time step, which is then scaled back to obtain the POD coefficient predictions.

For the model training, a data generator is implemented which extracts the training input and corresponding training target by mini-batches from the database $\mathscr{D}$ and $\mathscr{U}$, and then feeds them to the LSTM network. The LSTM network is trained to minimize the mean-squared error (MSE) between the network output and the training target. The Adam optimization algorithm [51] is used with a learning rate of 0.001 for all neural network training in this paper. The model is implemented based on the machine learning package Keras [52] with Tensorflow backend [53].

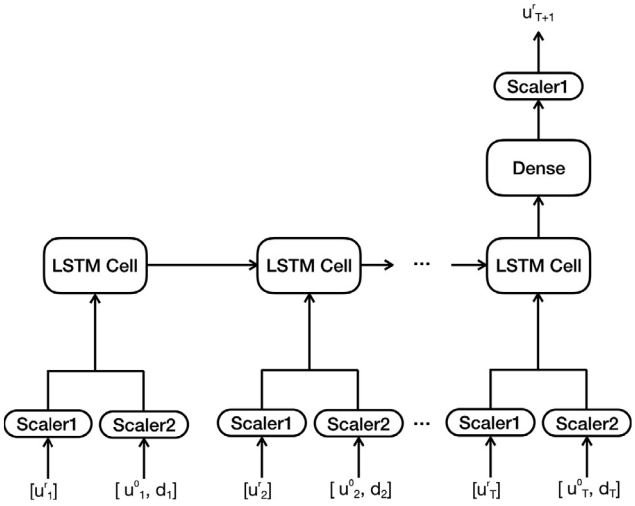


Fig. 3. The detailed illustration of the LSTM network in Fig. 2.

After training, the POD-LSTM model can predict the flow field at time step $T + 1$ (i.e. $\widehat{u}_{T+1}$), given the history of the flow field data (i.e. $[\widetilde{u}_1, \widetilde{u}_2, ...\widetilde{u}_T]$), inflow velocity (i.e. $[u_1^0, u_2^0, ...u_T^0]$), and the distributed control parameters' value (i.e. $[d_1, d_2, ...d_T]$). For the prediction of the flow field at time step $T + 2$ (i.e. $\widehat{u}_{T+2}$), the predicted flow field at time step $T + 1$ (i.e. $\widehat{u}_{T+1}$), the user-defined inflow velocity at time step $T + 1$ (i.e. u_{T+1}^0), and the user-defined distributed control parameter's value at time step $T + 1$ (i.e. d_{T+1}) along with the history data (i.e., $[\widetilde{u}_2, ...\widetilde{u}_T]$, $[u_2^0, ...u_T^0]$, $[d_2, ...d_T]$) are fed into the data pipeline. In this way, all future flow fields can be predicted iteratively.

2.4. Prediction of the whole flow field

The POD-LSTM model developed above can predict the flow field in a single subdomain at all future time steps given the history of the flow field and the future inflow velocity and distributed control parameters' value. For the prediction of the whole flow field, the trained POD-LSTM model is used to predict the flow field in each subdomain sequentially from upstream to downstream, and then the whole flow field is obtained by combining all the subdomains' predictions. The detailed procedure can be found in the Appendix.

3. Development of the novel dynamic wind farm wake model

The POD-LSTM method developed above is employed to build a novel dynamic wake model in this section.

3.1. High-fidelity data generation

The high-fidelity flow field data is generated using SOWFA (Simulator fOr Wind Farm Applications) [54], an LES solver for wind farm wakes developed at the National Renewable Energy Laboratory (NREL). It has been validated in previous studies [55,25]. The turbine rotors are modelled as actuator lines in this work. For the mesh generation, as recommended by [2], a mesh size of 12 m× 12 m× 12 m is used in the far field and a two-level mesh refinement is applied around the turbines such that the mesh size around turbine rotors is 3 m× 3 m× 3 m. The simulation domain considered in this work is the same as the one recommended by [2], which is illustrated in Fig. 4. A typical instantaneous flow field visualization is also shown. The corresponding hub-height horizontal plane is extracted and shown in Fig. 5, where the two-level mesh refinement is illustrated. The total number of cells is about 1.8×10^7. For turbine wake simulations, a precursor simulation of the atmospheric boundary layer is first carried out to obtain the initial flow field and the inflow boundary condition,

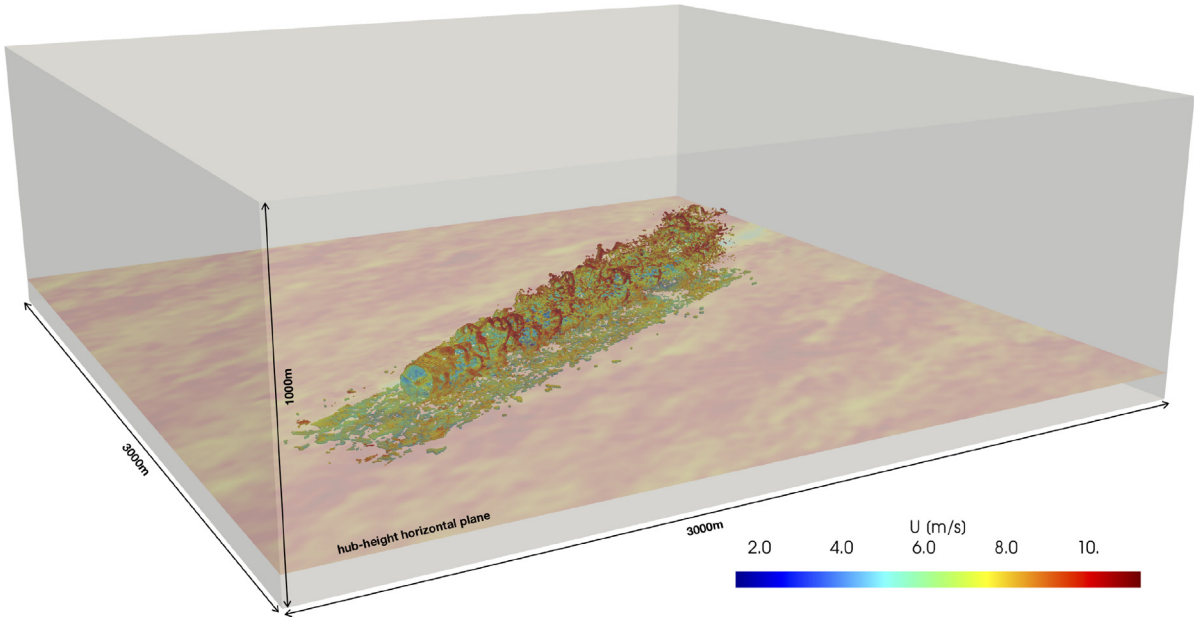


Fig. 4. The illustration of the 3D simulation domain. A typical instantaneous vorticity contour coloured by velocity magnitude is shown. The hub-height horizontal plane is also shown.

and then three NREL 5-MW wind turbines with the baseline turbine pitch and torque control [56] are added in the simulation domain with a downstream spacing of 5 rotor diameters. For each simulated case, 1110s simulations are carried out with a time step of 0.02 s.

Three inflow conditions with average wind speeds of 8 m/s, 9 m/s, and 10 m/s and freestream turbulence intensity (FSTI) of 6%, are considered. Twenty simulations are carried out for each inflow condition, with different yaw angles for each simulation case. The yaw angles are

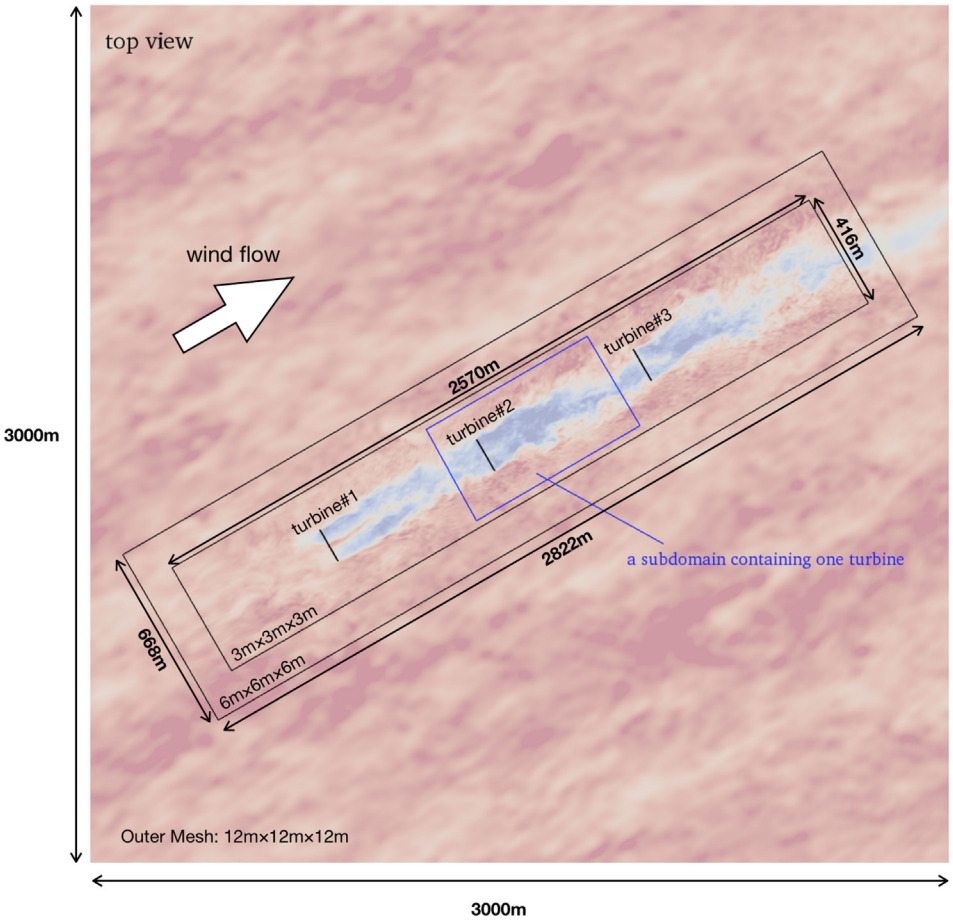


Fig. 5. A top view of the simulation domain at turbine hub height. The contour shows the instantaneous flow velocity magnitude.

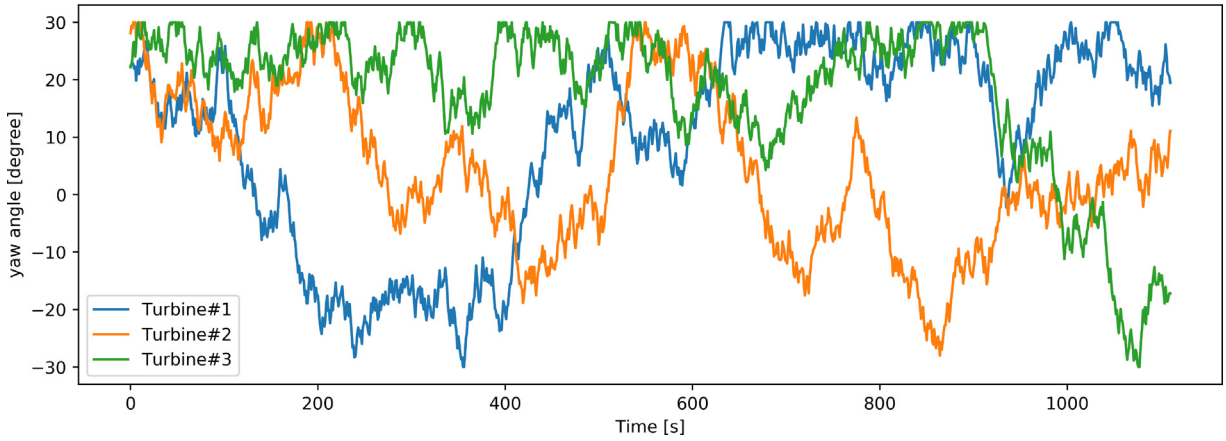


Fig. 6. An example of the designed yaw angles in a simulation case. The yaw angles of all the three turbines are included.

designed by random initial yaw and random yaw changes of less than 3° every second during the whole simulation period. The yaw angles are limited to the range [−30°, 30°]. By these settings, the generated LES data covers a wider flow speed range and turbine yaw range. An example of the designed yaw angles in a simulation case is shown in Fig. 6. After CFD simulations, the flow fields at turbine hub-height are sampled every second to extract the training data. The first 400 s simulation results are discarded as the turbine wakes have not been well established during this period. Therefore, 710 snapshots of the flow field are recorded for each case. Then the flow field in each subdomain containing one turbine (as shown in Fig. 5) is extracted and interpolated into a uniform grid of 50×30. All the generated flow field data is collected together to form the training dataset $\mathscr{U}$. The shape of the final training matrix $\mathscr{U}$ is [180, 710, 1500], with $\mathscr{U}_{s,i_t,1:1500}$ representing a snapshot of the flow field in one subdomain at time step i_t for the scenario indexed by s. All the designed yaw angles are collected as the design input matrix $\mathscr{D}$ of shape [180, 1110, 1], where $\mathscr{D}_{s,i_t,1}$ represents the yaw angle at time step i_t for the scenario indexed by s. Here a scenario designates the unsteady flow fields in one subdomain of one simulation case. The whole data generation process takes around 7×10^5 CPU hours where each simulation requires around 46 h' computation on a local cluster with 256 CPUs.

3.2. Model training

The generated LES data contains 180 flow scenarios with each scenario consisting of the unsteady flow fields at 710 discrete time instants. For model training purpose, the whole dataset is divided into a training dataset (the first 64% time instants), a validation dataset (the 64%~85% time instants), and a test dataset (the last 15% time instants). The training dataset is fed into the POD-LSTM network by mini-batches with a batch size of 1024 while the validation dataset is used to evaluate the model after each training epoch. The test dataset is not used in the training process but only for model testing after training.

Dropout, including the input dropout and the recurrent dropout, is an efficient technique to tackle overfitting. The LSTM network with and without the input and recurrent dropout are both tested. It turns out that the one with dropout performs much better, thus it is used in this paper. The stack of multiple LSTM layers does not further increase the model performance thus only one LSTM layer, as illustrated in Fig. 3, is included in the POD-LSTM model. There are still a few hyper-parameters undetermined in the POD-LSTM network, i.e., the total lookback time step of the flow history, the number of POD basis, and the output features' dimension of the LSTM cell. The validation errors are used to determine these hyper-parameters' empirical values, by a grid-search procedure. The final hyper-parameters' values are given in Table 3, along with the evaluations of the POD-LSTM model's performance by using the test dataset. The POD model reduction error is defined as the mean value of the root-mean-squared-errors (RMSEs) between the reconstructed flow fields from the exact POD coefficients and the exact flow fields:

Table 3
The hyper-parameters in the POD-LSTM model and the model evaluation. T represents the total lookback time step of the flow history in order to predict the current flow field, R represents the number of the POD basis, N_h represents the output features' dimension of the LSTM cell, and α represents the dropout rate of both input dropout and recurrent dropout. The model reduction error ϵ_{POD} (m/s) and the total prediction error ϵ_{total} (m/s) are for the model evaluation.

T	R	N_h	α	ϵ_{POD}	ϵ_{total}
5	350	350	0.2	0.328	0.428

$$\epsilon_{POD} = \frac{1}{180 \times 102} \sum_{s=1}^{180} \sum_{i_t=609}^{710} RMSE(\mathscr{U}_{s,i_t,1:1500}, \mathscr{U}^{POD}_{s,i_t,1:1500}), \tag{4}$$

where

$$\mathscr{U}^{POD}_{s,i_t,1:1500} = \sum_{k=1}^{R} < \mathscr{U}_{s,i_t,1:1500}, v_k > v_k. \tag{5}$$

And the POD-LSTM model prediction error is defined as the mean value of the RMSEs between the flow fields predicted by the POD-LSTM model and the exact flow fields:

$$\epsilon_{total} = \frac{1}{180 \times 102} \sum_{s=1}^{180} \sum_{i_t=609}^{710} RMSE(\mathscr{U}_{s,i_t,1:1500}, \widehat{\mathscr{U}}_{s,i_t,1:1500}). \tag{6}$$

where $\widehat{\mathscr{U}}_{s,i_t,1:1500}$ represents the POD-LSTM predictions. As shown in Table 3, the POD-LSTM prediction error arises from both the representation of the flow field by the reduced coefficients, which is characterized by ϵ_{POD}, and the difference of the exact POD coefficients and the ones predicted by the LSTM network, which is characterized by $\epsilon_{total} - \epsilon_{POD}$. The overall prediction error is 0.428 m/s, which is just 4.8% with respect to the freestream wind speed.

After training, the POD-LSTM model can be used for the prediction of the flow field of the next second given the flow history in the past five seconds. This prediction process can be carried out iteratively so that all the future flow fields can be predicted with a time step of 1 s.

4. Results and discussions

The flow field predictions, including both the single-turbine wake and multiple-turbine wake predictions, are carried out using the above developed dynamic wake model. The results are compared with the high-fidelity SOWFA simulation results for model validation. After that,

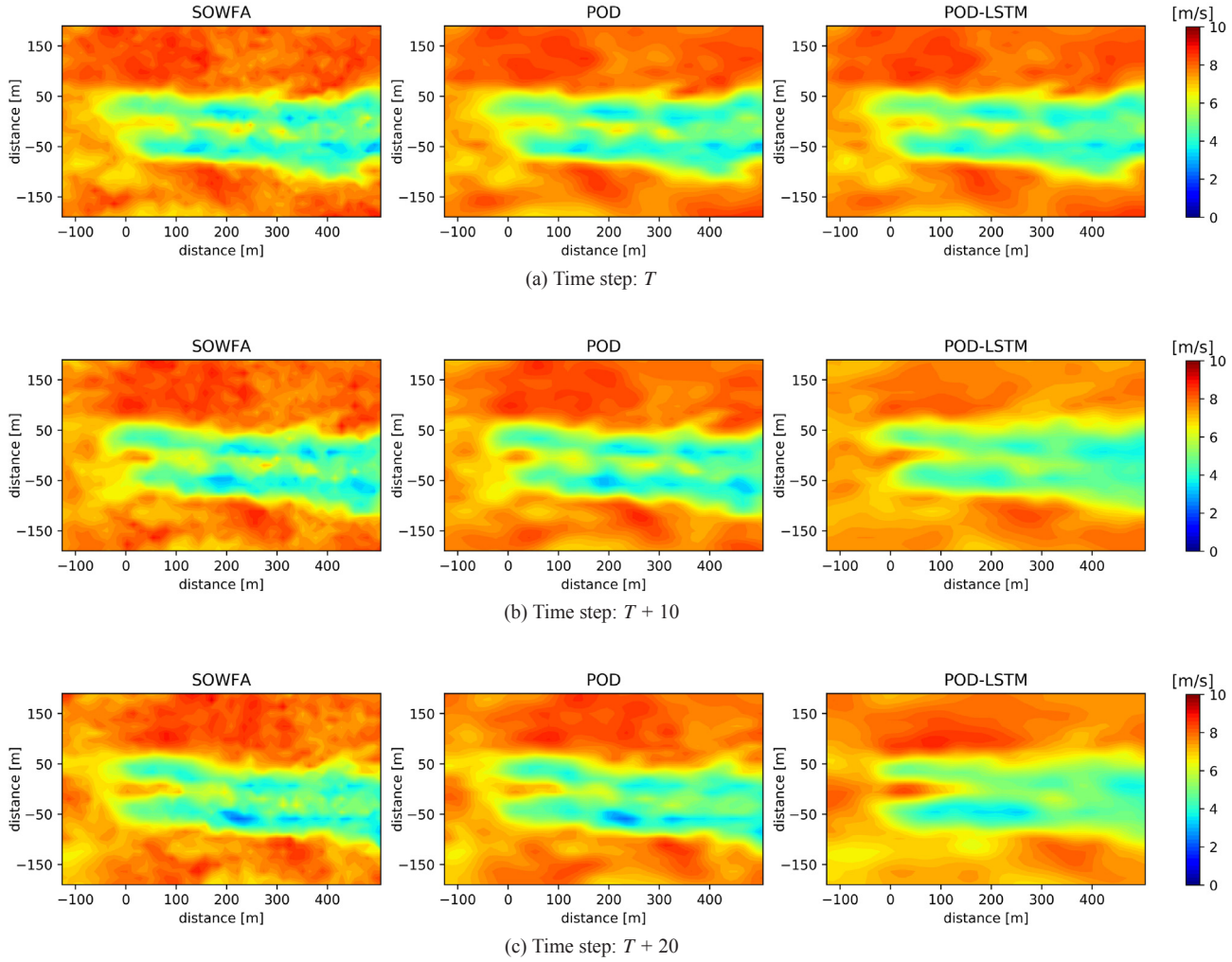


Fig. 7. An example case of the single-turbine wake prediction with the turbine operating in freestream condition. The results include the SOWFA predictions, the flow field reconstructions from exact POD coefficients, and the POD-LSTM model predictions at time step (*a*) T, (*b*) $T + 10$, and (*c*) $T + 20$. The turbine rotor is located at (0, 0) m of the 2D plane.

two simulation case studies are carried out to demonstrate the model's ability in capturing the yaw effect on turbine wakes and in simulating large-scale wind farms.

4.1. Model validations

4.1.1. Single-turbine wake predictions

The single-turbine wake predictions are carried out and compared with the test dataset. The POD-LSTM model is used to predict both the flow field in one time step directly and the flow fields in all future time steps iteratively. To predict the flow fields from time step T to $T + T_{tot}$, the calculation by the POD-LSTM model uses the same initial flow fields as SOWFA only from time step $T - 5$ to $T - 1$, the same inflow conditions as SOWFA from time step T to $T + T_{tot}$, and the same yaw angles as SOWFA from time step T to $T + T_{tot}$.

The predictions are carried out for all the cases in the test dataset. Two typical cases are chosen to demonstrate the model's performance, including one with the turbine operating in freestream condition and the other with the turbine operating in the front turbine's wake. The results are shown in Figs. 7 and 8, including the SOWFA predictions, the flow field reconstructions from exact POD coefficients and the POD-LSTM model predictions, at time step T, $T + 10$ and $T + 20$.

As can be seen, the reconstructions from exact POD coefficients match with SOWFA results quite well for all time steps in both cases, which illustrates that the chosen POD basis captures the main flow dynamics very well thus this dimensionality reduction process can be combined with the subsequent machine learning model for the accurate flow field predictions, as in [40,42]. The direct and iterative flow field predictions at time step T, $T + 10$ and $T + 20$ by the POD-LSTM model match with the POD reconstruction results very well in both cases, which demonstrates that the LSTM network can predict the POD coefficients accurately. The overall prediction error is small and satisfactory, considering the chaotic nature of the turbulent wakes and limited information used for these predictions.

4.1.2. Multiple-turbine wake predictions

The multiple-turbine wake predictions are carried out in this subsection to demonstrate the POD-LSTM model's ability in capturing wake interactions. The case of two turbines in a row with a downstream spacing of 5 rotor diameters is considered. The POD-LSTM model is used to predict the flow field in one time step directly and the flow fields in all future time steps iteratively. To predict the flow fields from time step T to $T + T_{tot}$, the calculation by the POD-LSTM model uses the same initial flow fields as SOWFA only from time step $T - 5$ to $T - 1$, the same freestream conditions (that is, the inflow conditions for the front turbine) as SOWFA from time step T to $T + T_{tot}$, and the same yaw angles as SOWFA for all the turbines from time step T to $T + T_{tot}$.

The predictions are carried out for all the flow conditions in the test dataset. An example case is chosen to demonstrate the model's performance in capturing the wake interactions. The results are shown in

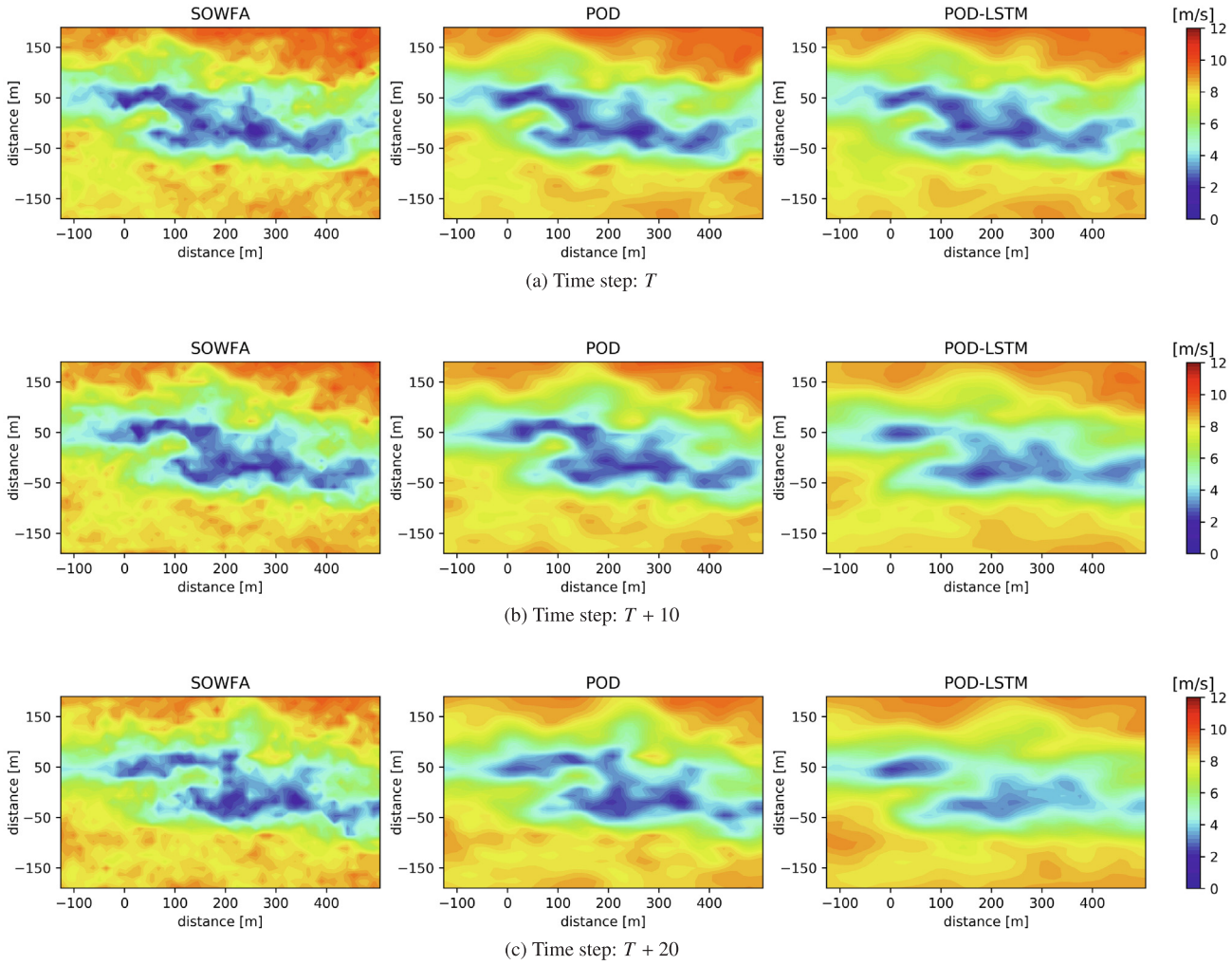


Fig. 8. An example case of the single-turbine wake prediction with the turbine operating in the front turbine's wake. The results include the SOWFA predictions, the flow field reconstructions from exact POD coefficients, and the POD-LSTM model predictions at time step (*a*) T, (*b*) $T + 10$, and (*c*) $T + 20$. The turbine rotor is located at (0, 0) m of the 2D plane.

Fig. 9, including the SOWFA predictions and the POD-LSTM model predictions at time step T, $T + 10$, and $T + 20$. As can be seen, the direct and iterative flow field predictions at time step T, $T + 10$ and $T + 20$ by the POD-LSTM model match with SOWFA simulation results quite well for both the front turbine's and the rear turbine's wake, which demonstrates that the proposed model can to predict the wake interactions accurately. It is worth noting that the impact of the upstream turbine on the downstream turbine is well captured in all the prediction time steps, which is essential in guaranteeing the performance of the developed model in large-scale wind farm predictions.

4.2. Model predictions - two case studies

4.2.1. The yaw effect on turbine wakes

A single turbine case with designed yaw change is investigated using the developed model to demonstrate its ability in capturing the yaw effect on turbine wakes. The single-turbine wake is predicted for a simulation time of 300 s, with the yaw angle being − 20° for the first 100 s, then increasing linearly from − 20° to 20° in the next 100*s*, and staying at 20° for the last 100*s*. The snapshots at 100 s and 300 s are shown in Fig. 10. As can be seen, the impact of turbine yaw on unsteady turbine wakes are captured, where the wake deflection is predicted correctly.

The video showing the unsteady flow field visualization can be found in the supporting materials of this paper (see Video1). As can be seen from the video, the main feature of the unsteady turbine wake, such as the streamwise convection of flow structures, the wake's crosswind meandering, and the wake's deflection with changing yaw are captured clearly by the developed model during the whole simulation duration. This further validates the developed model's ability in capturing main flow features for long time simulations. To our knowledge, there are no existing wake models that can achieve fast predictions of these unsteady flow features. We also mention that the successful prediction of the streamwise convection and crosswind meandering of flow structures is not trivial, as the LSTM network is not trained to predict the velocity at specific locations but the POD coefficients which do not directly reflect the spatial convection of the flow. The POD only serves as the dimensionality reduction technique and the POD basis does not characterize the coherent structures as in [57], because the flow field snapshots in the training dataset are collected from different simulations under random flow parameters. In addition, this case also demonstrates the generalization performance of the developed model, as the model has not encountered the designed yaw patterns (constant yaw and linear yaw change) during training.

4.2.2. A 9-turbine test case

The simulation of a 3 × 3 wind turbine array is carried out to illustrate the use of the developed model for the fast simulations of large-scale wind farm wakes. The freestream condition with the average wind speed of 9 m/s and FSTI of 6% is used. The turbine yaw angles are kept

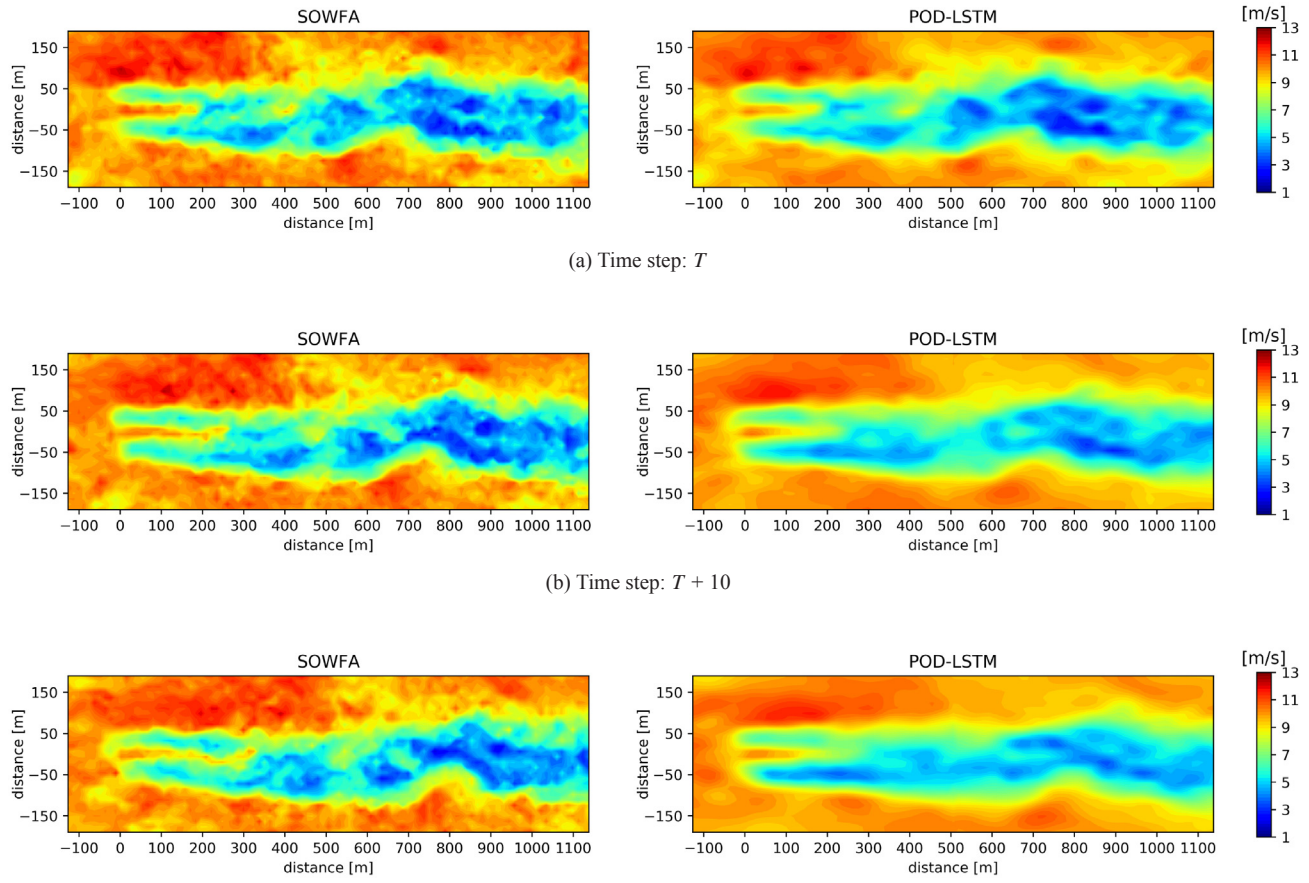


Fig. 9. An example case of the multiple-turbine wake predictions with two turbines in a row. The results include the SOWFA predictions and the POD-LSTM model predictions at time step (*a*) T, (*b*) $T + 10$, and (*c*) $T + 20$. The front and the rear turbine rotors are located at (0, 0) m and (632, 0) m of the 2D plane respectively.

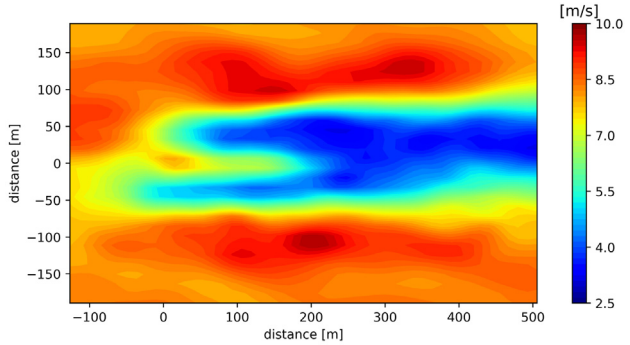
constant for the simulation time of 300*s*, with the front turbine yaw angle being 20°, 0°, −20° respectively and the yaw angles of the rest turbines being 0°. The snapshots at 180 s, 190 s, and 200 s are shown in Fig. 11. As can be seen, both the front turbines' wake deflections and the wake interactions between turbines are predicted correctly. However, as can be seen, there are discontinuities in the predicted flow fields at the interface between different rows of wind turbines. This is because the current model only considers the interactions between subdomains through the upstream boundary, which is enough in capturing the main wake interactions. This discontinuity issue can be solved by including all the boundary conditions of each subdomain as the input in the POD-LSTM model.

The unsteady flow field visualization can be found in the supporting materials of this paper (see Video2). As can be seen from the video, the POD-LSTM model predictions show similar flow characteristics seen in the LES of wind farms, such as the wake meandering and the streamwise convection of flow structures. The simulations by the POD-LSTM model require negligible computational time (several seconds) on a standard desktop, while LES of such system requires tens of thousands of CPU hours on an HPC cluster. This 9-turbine test case demonstrates the full potential of the developed model in the fast yet accurate simulation, prediction and control design of utility-scale wind farms.

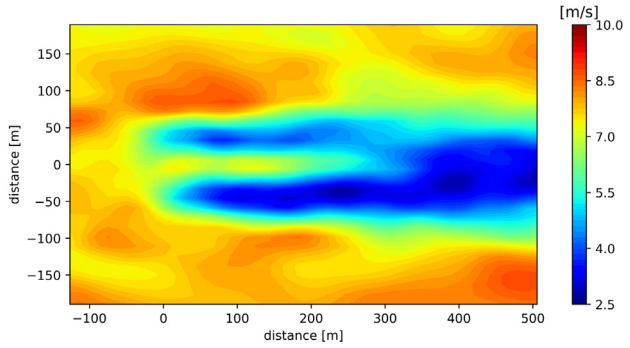
5. Conclusions

In this work, a deep learning based ROM method for distributed unsteady fluid systems was proposed, which was then applied to build a novel data-based dynamic wind farm wake model. A valuable high-fidelity LES database was first generated, which took around 7×10^5 CPU hours using high-performance computing clusters. Based on the generated LES database, the deep learning based dynamic wake model was trained to capture the complex wind farm wake dynamics. The results showed that the developed wake model was able to capture the main unsteady flow features (such as the streamwise convection of flow structures, the wake meandering, the wake's deflection with changing yaw, and the wake interactions between wind turbines) similarly as high-fidelity wake models while running as fast as the low-fidelity static wake models. The model's performance was validated against high-fidelity LES results and the overall prediction error is just 4.8% with respect to the freestream wind speed. After validating the developed wake model, two test cases were carried out, and the results demonstrated that the model was able to capture the yaw effect on turbine wakes and was able to achieve fast simulations of large-scale wind farms. In particular, the results of the 9-turbine test case showed that the developed model was able to predict the unsteady turbine wakes in several seconds on a standard desktop while it requires tens of thousands of CPU hours on a high-performance computing cluster if a high-fidelity model is used. As the existing wake models in the literature are either too time-consuming or unable to capture detailed wake dynamics, the developed model brings a step change in fast and accurate simulations, predictions, and control designs of wind farms. This work also paves the way for developing novel wake models using advanced machine learning techniques. The proposed ROM methods can also be applied to other distributed fluid systems to build reduced order models based on which optimal designs can be achieved with much less computation cost than based on high-fidelity models.

Future work may include applying this novel wind farm wake model in wake control in order to reduce wind turbine load, maximize the



(a) $t = 100s$



(b) $t = 300s$

Fig. 10. The snapshots of the flow field around a single turbine predicted by the POD-LSTM model with designed yaw change, at time steps (*a*) 100 s and (*b*) 300 s. The turbine rotor is located at (0, 0) m of the 2D plane.

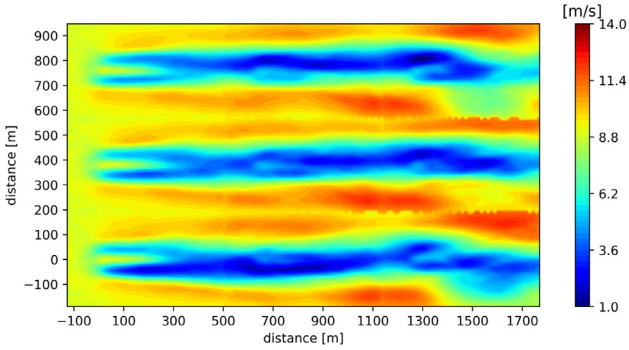
wind power harvesting, and support the electricity grid. This can be done by either using the developed model as an internal model in the control design or using it as a fast simulation model to design and test control strategies. As the developed model is fast to evaluate and can capture the yaw effect and wake interactions, it can be used for exploring new wake (or wake interaction) patterns used for some yaw control strategies. Another possible research direction is to incorporate the 3D wake dynamics in the machine learning models.

CRediT authorship contribution statement

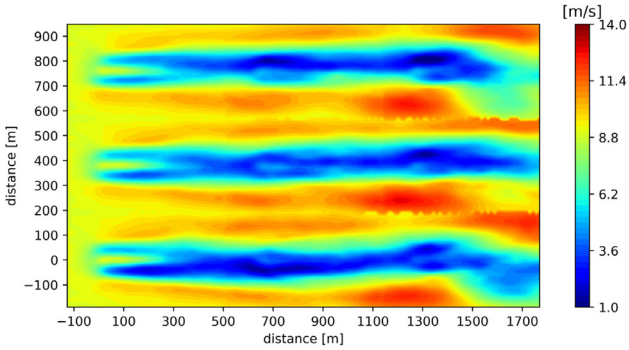
Jincheng Zhang: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Software, Validation, Visualization, Writing - original draft. **Xiaowei Zhao:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing - review & editing.

Declaration of Competing Interest

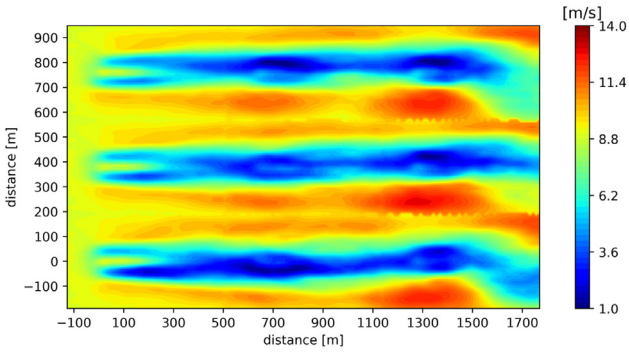
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



(a) $t = 180s$



(b) $t = 190s$



(c) $t = 200s$

Fig. 11. The snapshots of the flow field around a 3×3 wind turbine array predicted by the POD-LSTM model, at time steps (*a*) 180 s, (*b*) 190 s, and (*c*) 200 s. The 9 turbines are located at the grid points of $[0, 632, 1264] \times [0, 379.2, 758.4]$ m of the 2D plane.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 765579. The authors also acknowledge the Scientific Computing Research Technology Platform (SCRTP) at University of Warwick for providing High-Performance Computing resources. The authors are grateful for the advice of Bart M. Doekemeijer in SOWFA simulations.

Appendix A. Prediction of the whole flow field

The detailed procedure for predicting the flow field around multiple wind turbines is summarised below as Algorithm 1.

Algorithm 1. The prediction of the whole flow field

1: Divide the whole flow domain into M subdomains (as illustrated in Fig. 1) and number them from upstream to downstream as $[1, 2, ..., M]$.

2: Initialize the flow field history in all subdomains: $\{[\widetilde{u}_{1,i}, \widetilde{u}_{2,i}, ...\widetilde{u}_{T,i}], 1 \leqslant i \leqslant M\}$.
3: Initialize the inflow velocity history in all subdomains: $\{[u^0_{1,i}, u^0_{2,i}, ...u^0_{T,i}], 1 \leqslant i \leqslant M\}$.
4: Initialize the distributed control parameters' history in all subdomains: $\{[d_{1,i}, d_{2,i}, ...d_{T,i}], 1 \leqslant i \leqslant M\}$.
5: Set total prediction step T_{tot}.
6: $k \leftarrow 1$.
7: **while** $k \leqslant T_{tot}$ **do**
8: **for** i in [1, 2, ...,M] **do**
9: Propagate the input $[\widetilde{u}_{k,i}, \widetilde{u}_{k+1,i}, ...\widetilde{u}_{T+k-1,i}]$, $[u^0_{k,i}, u^0_{k+1,i}, ...u^0_{T+k-1,i}]$, $[d_{k,i}, d_{k+1,i}, ...d_{T+k-1,i}]$ through the data pipeline shown in Fig. 2 to obtain the flow field in the i^{th} subdomain at time step $T + k$: $\widetilde{u}_{T+k,i}$.
10: **end for**
11: Obtain the inflow velocity at time step $T + k$: $u^0_{T+k,i}$, $1 \leqslant i \leqslant M$, by setting it directly from the user-defined boundary condition for the most upstream subdomains, while for the downstream subdomains, extracting the inflow velocity from the neighbouring upstream subdomains' flow field predictions.
12: Set the distributed control parameters' value at time step $T + k$ from user-defined values: $d_{T+k,i}$, $1 \leqslant i \leqslant M$.
13: Output the whole flow field at time step $T + k$ by combining $\widetilde{u}_{T+k,1}$, $\widetilde{u}_{T+k,2}$, ...,$\widetilde{u}_{T+k,M}$ together.
14: $k \leftarrow k + 1$
15: **end while**
16: The unsteady flow fields for the whole domain from time $T + 1$ to $T + T_{tot}$ are obtained.

Appendix B. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.apenergy.2020.115552.

References

[1] Adaramola M, Krogstad P-Å. Experimental investigation of wake effects on wind turbine performance. Renewable Energy 2011;36(8):2078–86.
[2] Fleming P, Gebraad PM, Lee S, van Wingerden J-W, Johnson K, Churchfield M, et al. Simulation comparison of wake mitigation control strategies for a two-turbine case. Wind Energy 2015;18(12):2135–43.
[3] Boersma S, Doekemeijer B, Gebraad PM, Fleming PA, Annoni J, Scholbrock AK, et al. A tutorial on control-oriented modeling and control of wind farms. In: 2017 American Control Conference (ACC), IEEE; 2017. p. 1–18.
[4] Jensen NO. A note on wind generator interaction; 1983.
[5] Katic I, Højstrup J, Jensen NO. A simple model for cluster efficiency. In: European wind energy association conference and exhibition, A. Raguzzi; 1987.
[6] Frandsen S, Barthelmie R, Pryor S, Rathmann O, Larsen S, Højstrup J, et al. Analytical modelling of wind speed deficit in large offshore wind farms. Wind Energy: An Int J Progress Appl Wind Power Convers Technol 2006;9(1–2):39–53.
[7] Gebraad P, Teeuwisse F, Van Wingerden J, Fleming PA, Ruben S, Marden J, et al. Wind plant power optimization through yaw control using a parametric model for wake effects–a cfd simulation study. Wind Energy 2016;19(1):95–114.
[8] Sun H, Yang H. Study on an innovative three-dimensional wind turbine wake model. Appl Energy 2018;226:483–93.
[9] Gao X, Li B, Wang T, Sun H, Yang H, Li Y, et al. Investigation and validation of 3d wake model for horizontal-axis wind turbines based on filed measurements. Appl Energy 2020;260:114272.
[10] Bastankhah M, Porté-Agel F. A new analytical model for wind-turbine wakes. Renewable Energy 2014;70:116–23.
[11] Niayifar A, Porté-Agel F. Analytical modeling of wind farms: A new approach for power prediction. Energies 2016;9(9):741.
[12] Tian L, Zhu W, Shen W, Zhao N, Shen Z. Development and validation of a new two-dimensional wake model for wind turbine wakes. J Wind Eng Ind Aerodyn 2015;137:90–9.
[13] Gao X, Yang H, Lu L. Optimization of wind turbine layout position in a wind farm using a newly-developed two-dimensional wake model. Appl Energy 2016;174:192–200.
[14] Shapiro CR, Gayme DF, Meneveau C. Modelling yawed wind turbine wakes: a lifting line approach. J Fluid Mech. 2018;841.
[15] Lopez D, Kuo J, Li N. A novel wake model for yawed wind turbines. Energy 2019;178:158–67.
[16] Dou B, Guala M, Lei L, Zeng P. Wake model for horizontal-axis wind and hydrokinetic turbines in yawed conditions. Appl Energy 2019;242:1383–95.
[17] Brogna R, Feng J, Sørensen JN, Shen WZ, Porté-Agel F. A new wake model and comparison of eight algorithms for layout optimization of wind farms in complex terrain. Appl Energy 2020;259:114189.
[18] Ge M, Wu Y, Liu Y, Li Q. A two-dimensional model based on the expansion of physical wake boundary for wind-turbine wakes. Appl Energy 2019;233:975–84.
[19] Zhang J, Zhao X. Quantification of parameter uncertainty in wind farm wake modeling. Energy 2020:117065.
[20] Nilsson K, Ivanell S, Hansen KS, Mikkelsen R, Sørensen JN, Breton S-P, et al. Large-eddy simulations of the lillgrund wind farm. Wind Energy 2015;18(3):449–67.
[21] Wu Y-T, Porté-Agel F. Modeling turbine wakes and power losses within a wind farm using les: An application to the horns rev offshore wind farm. Renewable Energy 2015;75:945–55.
[22] Calaf M, Meneveau C, Meyers J. Large eddy simulation study of fully developed wind-turbine array boundary layers. Phys Fluids 2010;22(1):015110.
[23] Meyers J, Meneveau C. Large eddy simulations of large wind-turbine arrays in the atmospheric boundary layer, in. 48th AIAA Aerospace sciences meeting including the new horizons forum and aerospace exposition. 2010. p. 827.
[24] Lu H, Porté-Agel F. Large-eddy simulation of a very large wind farm in a stable atmospheric boundary layer. Phys Fluids 2011;23(6):065101.
[25] Churchfield MJ, Lee S, Michalakes J, Moriarty PJ. A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics. J Turbulence 2012(13):N14.
[26] Witha B, Steinfeld G, Heinemann D. High-resolution offshore wake simulations with the les model palm. Wind energy-impact of turbulence. Springer; 2014. p. 175–81.
[27] Martínez-Tossas LA, Churchfield MJ, Leonardi S. Large eddy simulations of the flow past wind turbines: actuator line and disk modeling. Wind Energy 2015;18(6):1047–60.
[28] Stevens RJ, Martínez-Tossas LA, Meneveau C. Comparison of wind farm large eddy simulations using actuator disk and actuator line models with wind tunnel experiments. Renewable energy 2018;116:470–8.
[29] Sprague M, Ananthan S, Vijayakumar G, Robinson M. Exawind: A multi-fidelity modeling and simulation environment for wind energy. In: J. Phys. Conf. Series; 2020.
[30] Larsen GC, Aagaard HM, Bingöl F, Mann J, Ott S, Sørensen JN, et al. Dynamic wake meandering modeling. 2007.
[31] Boersma S, Doekemeijer B, Vali M, Meyers J, Wingerden J-Wv. A control-oriented dynamic wind farm model: Wfsim. Wind Energy Sci 2018;3(1):75–95.
[32] Shapiro CR, Starke GM, Meneveau C, Gayme DF. A wake modeling paradigm for wind farm design and control. Energies 2019;12(15):2956.
[33] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.
[34] Brunton SL, Noack BR, Koumoutsakos P. Machine learning for fluid mechanics. Ann Rev Fluid Mech 2019;52.
[35] Guo X, Li W, Iorio F. Convolutional neural networks for steady flow approximation, in. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM; 2016. p. 481–90.
[36] Jin X, Cheng P, Chen W-L, Li H. Prediction model of velocity field around circular cylinder over various reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. Phys Fluids 2018;30(4):047105.
[37] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[38] Lee S, You D. Data-driven prediction of unsteady flow over a circular cylinder using deep learning. J Fluid Mech 2019;879:217–54.
[39] Kim B, Azevedo VC, Thuerey N, Kim T, Gross M, Solenthaler B. Deep fluids: A generative network for parameterized fluid simulations. In: Computer Graphics Forum, vol. 38, Wiley Online Library; 2019, p. 59–70.
[40] Hesthaven JS, Ubbiali S. Non-intrusive reduced order modeling of nonlinear problems using neural networks. J Comput Phys 2018;363:55–78.
[41] Swischuk R, Mainini L, Peherstorfer B, Willcox K. Projection-based model reduction: Formulations for physics-based machine learning. Comput Fluids 2019;179:704–17.
[42] Wang Q, Hesthaven JS, Ray D. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. J Comput Phys 2019;384:289–307.
[43] Lui HF, Wolf WR. Construction of reduced-order models for fluid flows using deep feedforward neural networks. J Fluid Mech 2019;872:963–94.
[44] Tou S, Tso C, Zhang X. 3-d numerical analysis of natural convective liquid cooling of a 3×3 heater array in rectangular enclosures. Int J Heat Mass Transfer 1999;42(17):3231–44.

[45] Mon MS, Gross U. Numerical study of fin-spacing effects in annular-finned tube heat exchangers. Int J Heat Mass Transfer 2004;47(8–9):1953–64.
[46] Liu J, Srebric J, Yu N. Numerical simulation of convective heat transfer coefficients at the external surfaces of building arrays immersed in a turbulent boundary layer. Int J Heat Mass Transf 2013;61:209–25.
[47] Chatterjee A. An introduction to the proper orthogonal decomposition. Current Sci 2000:808–17.
[48] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[49] Hyvärinen A, Karhunen J, Oja E. Independent component analysis vol. 46. John Wiley & Sons; 2004.
[50] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006;313(5786):504–7.
[51] Kingma DP, Ba J. Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.
[52] Chollet F, et al., Keras, https://github.com/fchollet/keras; 2015.
[53] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, https://www.tensorflow.org/, software available from tensorflow.org; 2015.
[54] Churchfield M, Lee S. Nwtc information portal (sowfa). https://nwtc.nrel.gov/sowfa; 2012.
[55] Churchfield M, Lee S, Moriarty P, Martinez L, Leonardi S, Vijayakumar G, et al. A large-eddy simulation of wind-plant aerodynamics, in. 50th AIAA Aerospace sciences meeting including the new horizons forum and aerospace exposition. 2012. p. 537.
[56] Jonkman J, Butterfield S, Musial W, Scott G. Definition of a 5-mw reference wind turbine for offshore system development. Tech. Rep., National Renewable Energy Lab.(NREL), Golden, CO (United States); 2009.
[57] Holmes P, Lumley JL, Berkooz G, Rowley CW. Turbulence, coherent structures, dynamical systems and symmetry. Cambridge University Press; 2012.