

Deep Learning for Wake Modeling of Wind Turbines

Suraj Pawar^{*,**}, Ashesh Sharma^{**}, Ganesh Vijayakumar^{**},
Christopher J. Bay^{**}, Shashank Yellapantula^{**}
Corresponding author: supawar@okstate.edu

^{*} Oklahoma State University, Stillwater, Oklahoma, USA.

^{**} National Renewable Energy Laboratory, Golden, Colorado, USA.

Abstract: Accurate wake modeling of wind turbines is essential for wind farm power output estimation or designing wind farm control strategies to help improve the overall performance of a wind farm. High-fidelity physics-based methods that can account for variable atmospheric wind conditions, and interactions between wakes are computationally expensive, and, thus data-driven surrogates are sought for fast and accurate wake modeling of wind turbines. In this work, we explore two deep learning methods to learn the wake model from an approximate form of the Reynolds-averaged Navier–Stokes equations. The first method is based on a single neural network that is trained to learn the mapping between the parameter space and the wake flow field. The second method utilizes a composite neural network that is trained using large samples of low-fidelity data along with very few samples of high-fidelity data. We train a composite neural network using data generated from the Gauss and curl wake models treated as proxies for low- and high-fidelity models, respectively. This work opens up possibilities for data-efficient construction of predictive surrogates for wake modeling that can be utilized to study the influence of wind speed, yaw angles, and layout configuration on wind farm power production.

Keywords: Deep Learning, Computational Fluid Dynamics, Wake Prediction.

1 Introduction

The U.S. Department of Energy has set the goal of generating 20% of the nation’s electricity from wind energy by 2030 [1] and similar goals have been set worldwide [2]. This rapid growth in the utilization of non-conventional energy resources like wind energy is important considering the depletion of fossil fuels and their negative impact on the environment and climate change. The layout optimization and control of wind farms are very important for realizing this goal [3]. A strong wake interaction between wind turbines is a major source of power loss in wind farms [4]. Once the kinetic energy is extracted by the first row of turbines, the wind speed does not recover to its freestream speed and subsequent turbines encounter lower wind speed. Therefore, accurate modeling and improved prediction of the wake are important for achieving optimal wind farm layout design and better yaw-control strategies to improve the operational efficiency [5].

Wake modeling is a very challenging problem due to spatio-temporal variability of wind speed, unsteady nature of interactions of turbine wakes with other wakes, and atmospheric turbulence [6]. There are different approaches for modeling wake ranging from high-fidelity large eddy simulation (LES) to simplified analytical models. One of the most widely used Jensen models is derived based on linear expanding wake assumption [7]. The Gauss model is derived by applying conservation of mass and momentum distribution and assuming a Gaussian kernel for velocity deficit in the wake [8]. The curled wake model was proposed to capture the curling mechanism in wake steering and is obtained by linearizing the Reynolds-averaged Navier–Stokes (RANS) streamwise momentum equation and neglecting the pressure gradient and viscous forces in a boundary layer [9]. Additionally, high-fidelity methods like Reynolds-Averaged Navier-Stokes (RANS) and large eddy simulation (LES) can be applied to resolve phenomena like wake meandering, yawed inflows, and the effect of atmospheric stratification on wake development [10, 11]. However these high-fidelity methods are

computationally intractable to be used for optimization or control, and, hence simpler analytical models are preferred for outer loop tasks.

Recently, machine learning is being applied to variety of problems in wind energy, such as, power forecasting [12, 5]; superresolution of wind data from global climate models [13]; airfoil design optimization for wind turbine [14]; reconstruction of spatiotemporal wind field from LIDAR measurements [15, 16]; control of wind farms [17]; and many more [18]. Machine learning is also being utilized for building a surrogate model for wake prediction. Ti et al. [19] proposed a machine learning-based wake model by training deep learning using a dataset generated from CFD simulation with a modified $k - \epsilon$ model for turbulence modeling and actuator disk model for a wind turbine. Renganathan et al. [20] developed a data-driven wake model based on compressing the LiDAR data using a convolutional autoencoder and then learning a relation between parameter space to latent space with a neural network or a Gaussian process model. Modal decomposition techniques have also been explored to model the dynamics of the velocity field in the wake of the turbine [21].

In this work, we investigate two deep learning approaches for building a data-driven surrogate model for the wake prediction behind a single wind turbine. The first approach utilizes a simple feed-forward neural network to learn the mapping between the parameters of the surrogate model and the velocity in the wake of the turbine. The second approach is based on a composite neural network [22] that uses the data generated from the Gauss and curl model as proxies of low- and high-fidelity models. Section 2 presents two deep learning models used in this study for data-driven wake modeling. The data pre-processing for training deep learning models is described in Section 3. Section 4 discusses the numerical results obtained with both surrogate models and compare them against the true data. The concluding remark is provided in Section 5.

2 Deep Learning Models

Here, we describe two deep learning models that are used for building a surrogate model for wake prediction behind the single wind turbine.

2.1 Feed-forward neural network

A feed-forward neural network is designed using several layers consisting of the predefined number of neurons. Each neuron is associated with certain coefficients called weights and some bias. The weight determines how significant a certain input feature is to the output. The input from the previous layer is multiplied by a weight matrix as shown below

$$S^l = \mathbf{W}^l \mathbf{x}^{l-1}, \quad (1)$$

where \mathbf{x}^{l-1} is the output of the $(l-1)^{\text{th}}$ layer, \mathbf{W}^l is the matrix of weights for the l^{th} layer. The summation of the above input-weight product and the bias is then passed through a node's activation function which is usually some nonlinear function. The introduction of nonlinearity through activation function allows the neural network to learn the complex relations between the input and output. The output of the l^{th} layer can be written as

$$\mathbf{x}^l = \zeta(S^l + B^l), \quad (2)$$

where B^l is the vector of biasing parameters for the l^{th} layer and ζ is the activation function. If there are L layers between the input and the output in a neural network, then the output of the neural network can be represented mathematically as follows

$$\tilde{\mathbf{y}} = \zeta_L(\mathbf{W}^L, B^L, \dots, \zeta_2(\mathbf{W}^2, B^2, \zeta_1(\mathbf{W}^1, B^1, \mathbf{x}))), \quad (3)$$

where \mathbf{x} and $\tilde{\mathbf{y}}$ are the input and output of the feed-forward neural network, respectively. There are several activation functions that provides different nonlinearity. Some of the widely used activation functions are sigmoid $\zeta(\phi) = 1/(1 + e^{-\phi})$, hyperbolic tangent (\tanh) $\zeta(\phi) = (e^\phi - e^{-\phi})/(e^\phi + e^{-\phi})$, and rectified linear unit (ReLU) $\zeta(\phi) = \max[0, \phi]$.

The matrix \mathbf{W} and B are determined through the minimization of the loss function (for example mean squared error for regression task). The gradient of the objective function with respect to trainable parameters is calculated through the backpropagation algorithm. The optimization algorithms like the Adam method

[23] provides a rapid way to learn optimal weights. The training procedure for a neural network can be summarized as follows

- The input and output of the neural network are provided along with some initial weights and bias.
- The training data is propagated forward through the network to produce output $\tilde{\mathbf{y}}$ whose true label is \mathbf{y} .
- The derivative of the objective function with each of the neural network parameters is computed using the backpropagation.
- The trainable parameters of the network are updated based on the learning rate and the optimization algorithm.

This procedure is repeated until convergence or the maximum number of iterations is reached.

2.2 Composite neural network

The main idea in multi-fidelity modeling is to learn the relation between low- and high-fidelity data. The correlation strategy between low- and high-fidelity data can be expressed as follows

$$\mathbf{y}_H = \rho(\mathbf{x})\mathbf{y}_L + \delta(\mathbf{x}), \quad (4)$$

where \mathbf{y}_L and \mathbf{y}_H are low- and high-fidelity data respectively, $\rho(\mathbf{x})$ is the multiplicative correlation, and $\delta(\mathbf{x})$ is an additive correlation. The relation given in Equation 4 captures only the linear relationship between low- and high-fidelity problems. For many scientific problems like wake modeling, the correlation between low- and high-fidelity data is nonlinear. Meng and Karniadakis [22] put forth a composite neural network that can discover the nonlinear correlation and a generalized scheme can be written as

$$\mathbf{y}_H = F(\mathbf{y}_L) + \delta(\mathbf{x}), \quad (5)$$

where $F(\cdot)$ is a function that maps low-fidelity data to high-fidelity data. Equation 5 can further be rewritten as follows

$$\mathbf{y}_H = \mathcal{F}(\mathbf{y}_L, \mathbf{x}), \quad (6)$$

The unknown function $\mathcal{F}(\cdot)$ can be divided into linear and nonlinear parts. Therefore the mapping from low-fidelity data to high-fidelity data can be written as follows

$$\mathbf{y}_H = \mathcal{F}_l(\mathbf{x}, \mathbf{y}_L) + \mathcal{F}_{nl}(\mathbf{x}, \mathbf{y}_L), \quad (7)$$

where $\mathcal{F}_l(\cdot)$ and $\mathcal{F}_{nl}(\cdot)$ are linear and nonlinear parts of $\mathcal{F}(\cdot)$, respectively.

The composite neural network has three feed-forward neural networks. The first one is the low-fidelity neural network $\mathcal{NN}_L(\mathbf{x}_L, \boldsymbol{\theta})$ that approximates the low-fidelity data. The second and third neural networks, $\mathcal{NN}_{H_i}(\mathbf{x}_H, \boldsymbol{\gamma}_i), i = 1, 2$, are utilized for approximating the linear and nonlinear correlation between low- and high-fidelity data (i.e., $\mathcal{F}_l = \mathcal{NN}_{H_1}$ and $\mathcal{F}_{nl} = \mathcal{NN}_{H_2}$). The unknown parameters of the multi-fidelity composite neural network, $\boldsymbol{\theta}, \boldsymbol{\beta}_i, i = 1, 2$ are learned by minimizing the below loss function

$$\text{MSE} = \frac{1}{N_{\mathbf{y}_L}} \sum_{i=1}^{N_{\mathbf{y}_L}} (|\tilde{\mathbf{y}}_L - \mathbf{y}_L|^2) + \frac{1}{N_{\mathbf{y}_H}} \sum_{i=1}^{N_{\mathbf{y}_H}} (|\tilde{\mathbf{y}}_H - \mathbf{y}_H|^2) + \lambda \sum \beta_i^2 \quad (8)$$

where $\tilde{\mathbf{y}}_L, \tilde{\mathbf{y}}_H$ denotes the output of $\mathcal{NN}_L, \mathcal{NN}_H$, respectively. β corresponds any weights from $\tilde{\mathbf{y}}_L$ and $\tilde{\mathbf{y}}_{H_2}$, and λ is the regularization rates for β . The regularization is usually adopted to prevent overfitting. If we have more than one low-fidelity model, then we can employ multiple low-fidelity neural networks (i.e., $\mathcal{NN}_{L_i}(\mathbf{x}_{L_i}, \boldsymbol{\theta}_i), i = 1, \dots, n$, where n is the number of low-fidelity models). The output from all these low-fidelity models can be concatenated together and can be given as an input to high-fidelity neural network. One of the main advantages of the present approach is that it allows for different amount of sampling of low- and high-fidelity data and this sampling can be essentially at different points. This makes the present approach flexible for fusing data from different sources.

3 Data Generation

We consider the NREL 5-MW turbine [24] which has a hub height of 90m and the diameter of 126m. The geometry of the NREL 5-MW turbine is available open-source and is widely used as a baseline by the wind research community for offshore wind research.

Our surrogate model for the NREL 5-MW turbine has three parameters, i.e., inflow wind speeds, yaw angle, and turbulence intensity. First, the data is generated for training a neural network, and we utilize the FLOW Redirection and Induction in Steady State (FLORIS) software [25] for this step. FLORIS has options for several physics-based models ranging from analytical approaches to one-dimensional linearized Reynolds-Average Navier-Stokes equations (RANS) model. The use of simplified models allowed us to accelerate the data generation and pre-processing data curation steps for demonstrative purposes. The methods presented herein can be easily extended to data obtained from high-fidelity CFD simulations.

For our numerical experiments, the Gauss model is treated as the low-fidelity model, and the curled wake model [9] is considered a proxy for the high-fidelity model. Based on regular wind farm operating conditions, the low-fidelity data is generated for inflow wind speeds between 5.0m/s to 15.0m/s with an interval of 1.0m/s, turbulence intensity between 4% to 16% with 2% interval, and for yaw angles between -20° to 20° with 2° interval. The data for the high-fidelity model is generated for inflow wind speeds [5.0, 10.0, 12.5, 15.0] m/s, yaw angles $[-20^\circ, -10^\circ, 0^\circ, 10^\circ, 20^\circ]$, and turbulence intensity [4%, 10%, 16%]. The data is extracted at different downstream locations in the wake region behind the rotor. For each downstream location, the data is extracted along the cross-plane which covers $2D$ distance in the spanwise direction on each side of the turbine, and $2H$ distance in the vertical direction. In particular, if we discretize the domain with $n_x \times n_y \times n_z$, we get n_x samples for a single set of parameters corresponding to the velocity in the $\mathbb{R}^{n_y * n_z}$ cross-plane. To improve the robustness of deep learning models, we use thrust coefficient as an input feature instead of inflow wind speed, and the labels are normalized velocity deficit, i.e., $\Delta U/U_\infty$ instead of the wake velocity. The learning map for a composite neural network can be written as follows

$$\mathcal{NN}_L : \{C_T, \lambda, \text{TI}, x\} \in \mathbb{R}^4 \rightarrow \{\tilde{\mathbf{y}}_L\} \in \mathbb{R}^{n_y * n_z}, \quad (9)$$

$$\mathcal{NN}_H : \{C_T, \lambda, \text{TI}, x, \mathbf{y}_L\} \in \mathbb{R}^{n_y * n_z + 4} \rightarrow \{\tilde{\mathbf{y}}_H\} \in \mathbb{R}^{n_y * n_z}, \quad (10)$$

where C_T is the thrust coefficient at corresponding inflow wind speed, λ is the yaw angle, TI is the turbulence intensity, x is the stream-wise location, and \mathbf{y}_L is the prediction from the low-fidelity model at high-fidelity data points. Here, $\tilde{\mathbf{y}}_L$ and $\tilde{\mathbf{y}}_H$ are the output from low- and high-fidelity neural network part of the composite network.

4 Results and Discussion

This section presents the validation study where the wake prediction from data-driven models is compared with the curled wake model. The results from a single feed-forward neural network are referred to as HF and the results from a composite neural network are called MF. The data-driven models are evaluated at different inflow wind speeds, yaw angles, and turbulence intensities. Particularly, the inflow wind speed is varied from 5.0m/s to 15.0m/s with a 1.0m/s interval, yaw angle from -20° to 20° with 5° interval, and turbulence intensity from 4% to 16% with 2% interval. Therefore, we have a total of 693 test parameters for the evaluation. The neural network for the HF model is composed of four hidden layers with 40 neurons in each hidden layer. The low-fidelity network in a composite network has 6 hidden layers with 60 neurons in each hidden layer, and the high-fidelity network consists of four hidden layers with 40 neurons. Both the neural network uses the tanh activation function and is trained for 5000 epochs.

The performance of both data-driven models is assessed using the normalized root mean squared error (RMSE) for the kinetic energy flux in the streamwise direction. We consider the area between $-D$ to D in the spanwise direction and 0 to $2H$ in the vertical direction for computing the kinetic energy flux at the corresponding streamwise location. Mathematically this can be written as follows

$$\mathcal{E}(x) = \int_{-D}^D \int_0^{2H} U^2 dz dy, \quad (11)$$

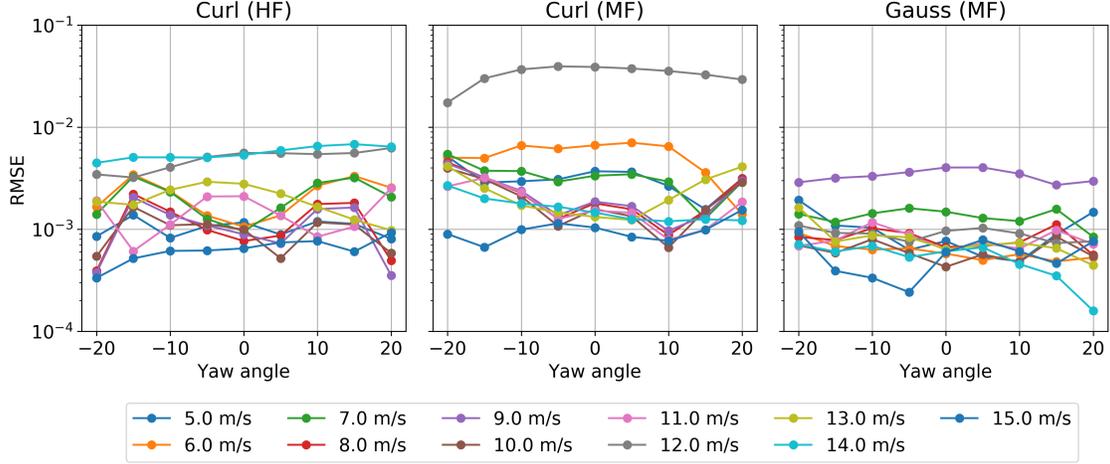


Figure 1: Normalized RMSE of prediction of kinetic energy flux for different yaw angles (γ) at TI = 4%

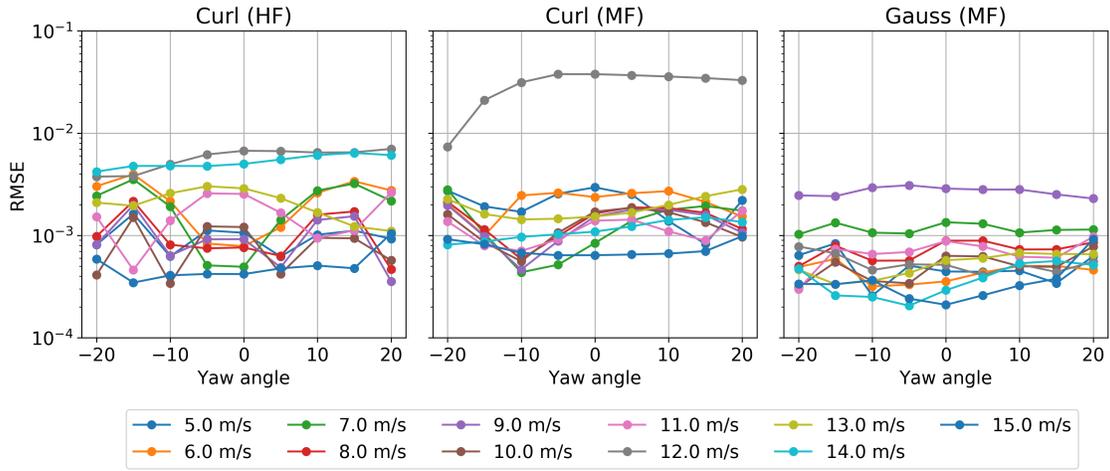


Figure 2: Normalized RMSE of prediction of kinetic energy flux for different yaw angles (γ) at TI = 10%

where U is the velocity in the wake of the turbine, z is the vertical direction, and y is the spanwise direction. The relative percentage error for the kinetic energy flux is defined as follows

$$\text{RMSE} = \sqrt{\frac{1}{n_x} \sum_{x=1D}^{10D} \left(\frac{\mathcal{E}_T(x) - \mathcal{E}_P(x)}{\mathcal{E}_T(x)} \right)^2}, \quad (12)$$

where x is the streamwise location, E_T is the true kinetic energy flux, and E_P is the predicted kinetic energy flux from the data-driven surrogate model. We note here that the near wake region ($x \leq 1D$) is not considered for computing the relative percentage error as the flow is strongly disturbed by the rotor geometry in this region. Figures 1- 3 displays the variation of normalized RMSE for kinetic energy flux in the streamwise direction for different turbulence intensities 4% to 16%. Overall the normalized RMSE is of a similar magnitude for all inflow wind speeds and turbulence intensity for both HF and MF models. The normalized RMSE for the inflow wind speed of 12.0m/s is higher for the MF model compared to the HF model.

Next, we visualize the velocity field in the hub-height plane for some test cases. In Fig. 4 and Fig. 5, the velocity predicted from HF and MF models along with normalized percentage error is depicted for inflow

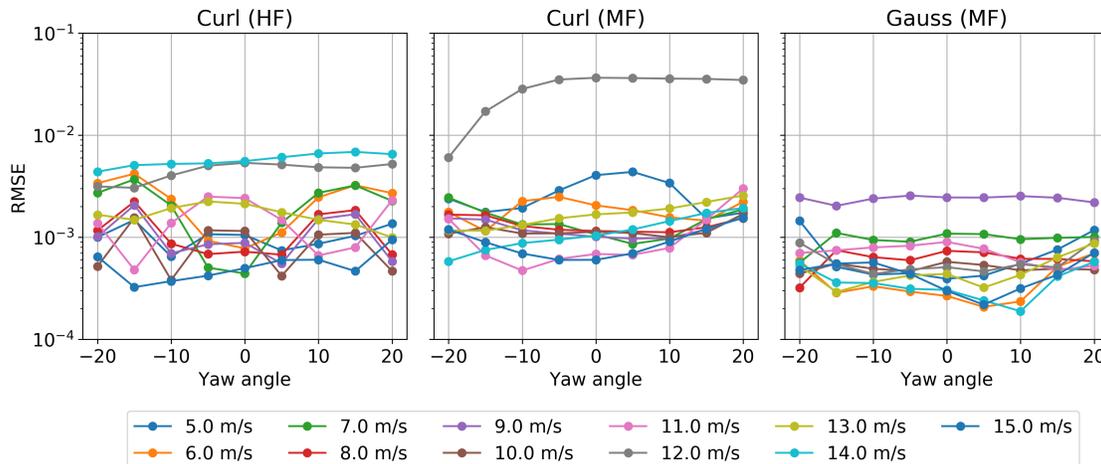


Figure 3: Normalized RMSE of prediction of kinetic energy flux for different yaw angles (γ) at TI = 16%

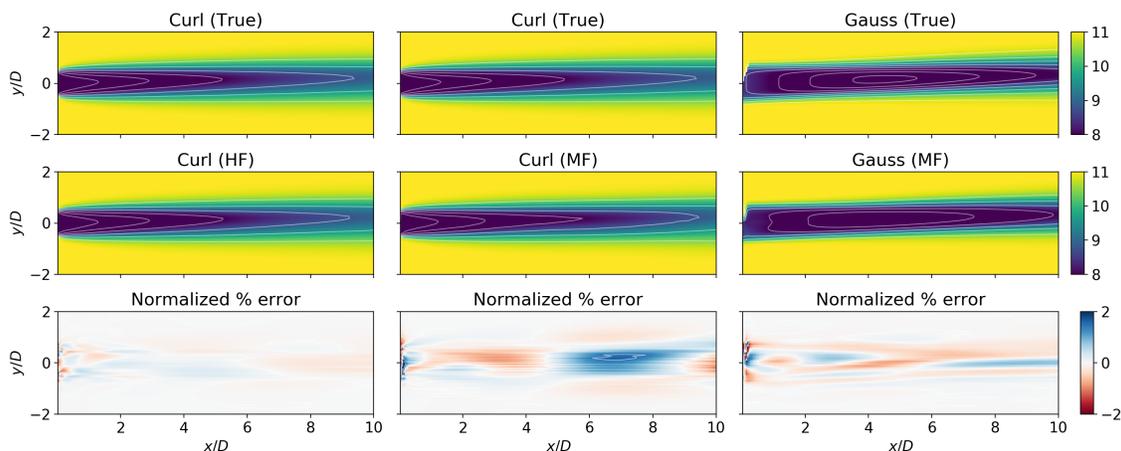


Figure 4: Visualization of the velocity field in the hub-height horizontal plane for inflow wind speed $U_\infty = 11.0\text{m/s}$ and yaw angle $\gamma = -15^\circ$ at turbulence intensity 6%.

wind speed of 11.0 m/s at yaw angle -15° and 0° , respectively. The normalized percentage error is calculated as the difference between true and predicted wake velocity non-dimensionalized with the inflow wind speed, i.e., $\frac{(U_T - U_P)}{U_\infty} \times 100$, where U_T is the true velocity, U_P is the prediction from the data-driven model, and U_∞ is the inflow wind speed. We observe that both HF and MF are able to capture the flow structure in the wake of the turbine with a high level of accuracy. We also show the prediction for the Gauss model obtained for the MF to demonstrate the success of a low-fidelity network in accurately predicting the wake velocity. Fig 6 and Fig. 7 shows the velocity field in the cross-plane at a downstream distance of $x/D = 8$ for inflow wind speed of 11.0 m/s at yaw angles -15° and 0° , respectively. The prediction accuracy for the HF and MF is mostly uniform over the cross-plane, and there is a very good agreement with the true velocity field for the curl model.

5 Conclusion and Future Work

This work introduces two deep learning methods to predict the three-dimensional velocity field in the wake of the wind turbine by leveraging data obtained from approximate for of the Reynolds-averaged Navier-Stokes (RANS) equations. Specifically, a parametric surrogate model for wake prediction is developed by utilizing a

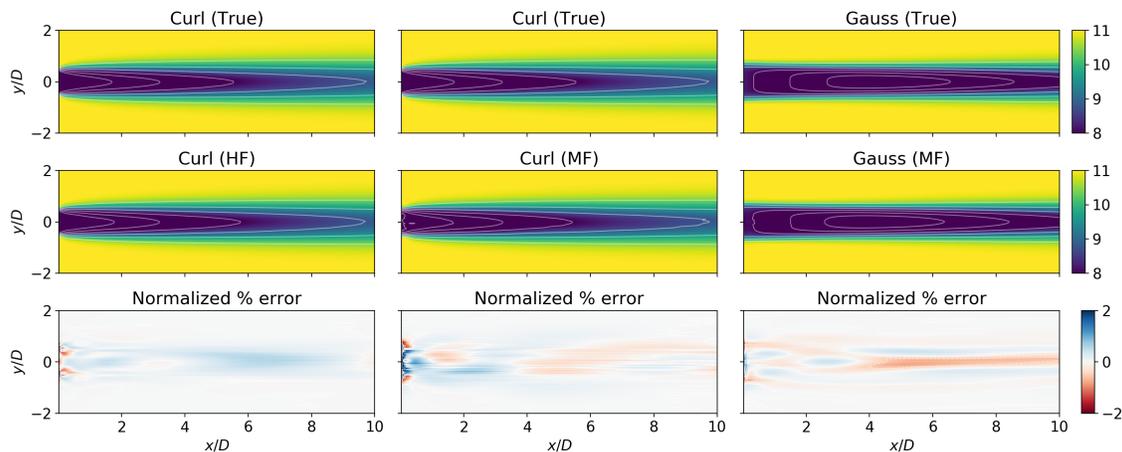


Figure 5: Visualization of the velocity field in the hub-height horizontal plane for inflow wind speed $U_\infty = 11.0m/s$ and yaw angle $\gamma = 0^\circ$ at turbulence intensity 6%.

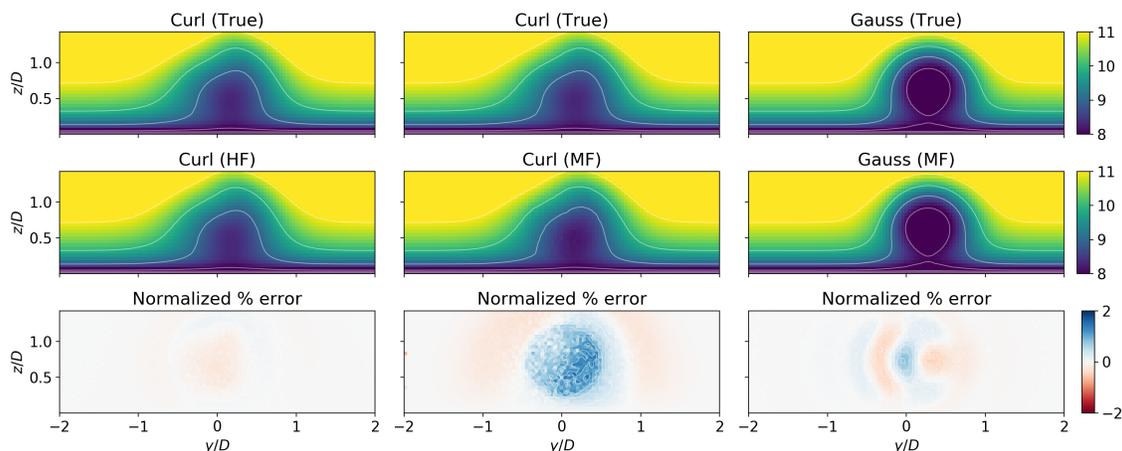


Figure 6: Visualization of the velocity field in the the cross-plane at $x/D = 8$ for inflow wind speed $U_\infty = 11.0m/s$ and yaw angle $\gamma = -15^\circ$ at turbulence intensity 6%.

neural network that learns the mapping between the inflow wind speed, yaw angle, and turbulence intensity to a high-dimensional velocity field. Furthermore, a composite neural network is investigated that is trained using two streams of data. The data from the analytical model is treated as a proxy for low-fidelity data, and the data from RANS equations is considered the high-fidelity data. Overall, once trained both data-driven models are able to predict the wake velocity with a sufficient level of accuracy compared to the true high-fidelity data.

This work is the first step toward building data-driven wake models by using multi-fidelity data. One of the limitations of the present approach is training the neural network to predict a high-dimensional velocity field ($O(1) \rightarrow O(1000)$) and this will be addressed using dimensionality reduction techniques to compress the data in future studies. Our future work also includes extending these deep learning models for data gathered from three-dimensional RANS/LES simulations and LiDAR measurements. The deep learning models can also be extended to quantify uncertainty using a Bayesian neural network [26]. Furthermore, the integration of these models into FLORIS for modeling the flow field in a large wind farm to perform layout optimization is also another direction for future work. Introducing the physics-based prior knowledge through a loss function can also be a considerable path for our future work.

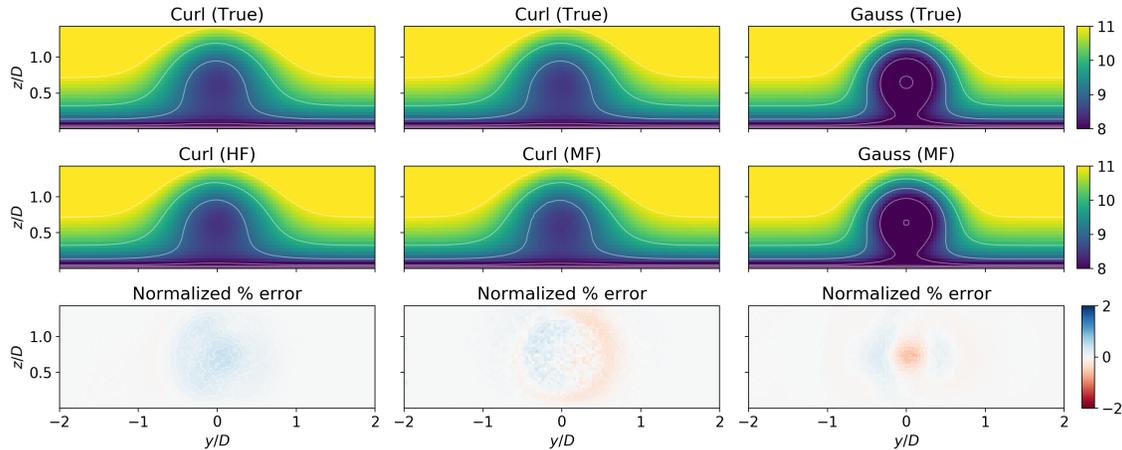


Figure 7: Visualization of the velocity field in the the cross-plane at $x/D = 8$ for inflow wind speed $U_\infty = 11.0m/s$ and yaw angle $\gamma = 0^\circ$ at turbulence intensity 6%.

Acknowledgments

A considerable part of the work has been performed when S.P. was a research intern at the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding was provided by U.S. Department of Energy Office of Energy Efficiency and Renewable Energy Wind Energy Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

- [1] Steve Lindenberg. *20% Wind Energy By 2030: Increasing Wind Energy's Contribution to US Electricity Supply*. diane publishing, 2009.
- [2] Global Wind Energy Council. Global wind report annual market update 2013. *Global Wind Energy Council*, 2014.
- [3] Paul Veers, Katherine Dykes, Eric Lantz, Stephan Barth, Carlo L Bottasso, Ola Carlson, Andrew Clifton, Johney Green, Peter Green, Hannele Holttinen, et al. Grand challenges in the science of wind energy. *Science*, 366(6464), 2019.
- [4] Rebecca Jane Barthelmie and LE Jensen. Evaluation of wind farm efficiency and wind turbine wakes at the nysted offshore wind farm. *Wind Energy*, 13(6):573–586, 2010.
- [5] Michael F Howland and John O Dabiri. Wind farm modeling with interpretable physics-informed machine learning. *Energies*, 12(14):2716, 2019.
- [6] B Sanderse. *Aerodynamics of wind turbine wakes*. Petten: ECN, 2009.
- [7] Niels Otto Jensen. A note on wind generator interaction. 1983.
- [8] Majid Bastankhah and Fernando Porté-Agel. A new analytical model for wind-turbine wakes. *Renewable energy*, 70:116–123, 2014.
- [9] Luis A Martínez-Tossas, Jennifer Annoni, Paul A Fleming, and Matthew J Churchfield. The aerodynamics of the curled wake: a simplified model in view of flow control. *Wind Energy Science*, 4(1):127–138, 2019.
- [10] D Mehta, AH Van Zuijlen, B Koren, JG Holierhoek, and H Bijl. Large eddy simulation of wind farm aerodynamics: A review. *Journal of Wind Engineering and Industrial Aerodynamics*, 133:1–17, 2014.
- [11] Yu-Ting Wu and Fernando Porté-Agel. Modeling turbine wakes and power losses within a wind farm using les: An application to the horns rev offshore wind farm. *Renewable Energy*, 75:945–955, 2015.

- [12] Kanna Bhaskar and Sri Niwas Singh. Awnn-assisted wind power forecasting using feed-forward neural network. *IEEE transactions on sustainable energy*, 3(2):306–315, 2012.
- [13] Karen Stengel, Andrew Glaws, Dylan Hettinger, and Ryan N King. Adversarial super-resolution of climatological wind and solar data. *Proceedings of the National Academy of Sciences*, 117(29):16805–16815, 2020.
- [14] Andrew Glaws, Ryan N King, Ganesh Vijayakumar, and Shreyas Ananthan. Invertible neural networks for airfoil design. *AIAA Journal*, 60(5):3035–3047, 2022.
- [15] Jincheng Zhang and Xiaowei Zhao. Spatiotemporal wind field prediction based on physics-informed deep learning and lidar measurements. *Applied Energy*, 288:116641, 2021.
- [16] Jincheng Zhang and Xiaowei Zhao. Three-dimensional spatiotemporal wind field reconstruction based on physics-informed deep learning. *Applied Energy*, 300:117390, 2021.
- [17] Paul Stanfel, Kathryn Johnson, Christopher J Bay, and Jennifer King. Proof-of-concept of a reinforcement learning framework for wind farm energy capture maximization in time-varying wind. *Journal of Renewable and Sustainable Energy*, 13(4):043305, 2021.
- [18] Navid Zehtabiyani-Rezaie, Alexandros Iosifidis, and Mahdi Abkar. Data-driven fluid mechanics of wind farms: A review. *Journal of Renewable and Sustainable Energy*, 2022.
- [19] Zilong Ti, Xiao Wei Deng, and Hongxing Yang. Wake modeling of wind turbines using machine learning. *Applied Energy*, 257:114025, 2020.
- [20] S Ashwin Renganathan, Romit Maulik, Stefano Letizia, and Giacomo Valerio Iungo. Data-driven wind turbine wake modeling via probabilistic machine learning. *Neural Computing and Applications*, pages 1–16, 2022.
- [21] Mithu Debnath, Christian Santoni, Stefano Leonardi, and Giacomo Valerio Iungo. Towards reduced order modelling for predicting the dynamics of coherent vorticity structures within wind turbine wakes. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2091):20160108, 2017.
- [22] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. *Journal of Computational Physics*, 401:109020, 2020.
- [23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [24] Jason Jonkman, Sandy Butterfield, Walter Musial, and George Scott. Definition of a 5-mw reference wind turbine for offshore system development. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2009.
- [25] NREL. Floris. version 2.4, 2021.
- [26] Xuhui Meng, Hessam Babaei, and George Em Karniadakis. Multi-fidelity bayesian neural networks: Algorithms and applications. *Journal of Computational Physics*, 438:110361, 2021.