RANS-AD based ANN surrogate model for wind turbine wake deficits

To cite this article: J. P. Schøler *et al* 2023 *J. Phys.: Conf. Ser.* **2505** 012022

RANS-AD based ANN surrogate model for wind turbine wake deficits

J. P. Schøler, R. Riva, S. J. Andersen, J. P. Murcia Leon, M. P. van der Laan, J. Criado Risco and P.-E. Réthoré

Technical University of Denmark (DTU), DTU Wind and Energy Systems, Frederiksborgvej 399, 4000 Roskilde, Denmark

E-mail: {jpsch, pire}@dtu.dk

Abstract. Inside wind farms, wake effects are the primary source of turbine interactions, and as such, they constitute one of the most important aspects of wind farm operations. The two most widespread methods for calculating the wind farm wake flow are computational fluid dynamics (CFD) methods and engineering models. Both methods have drawbacks; CFD methods can be very accurate but are computationally expensive. Vice versa, engineering models sacrifice accuracy by simplifying the physics, thereby improving computational efficiency.

In cases where many evaluations of the flow are needed, this trade-off is a hindrance. One such case is wind farm layout optimization problems. It has been shown that the estimation of wake flows can be improved by surrogate modeling. Recently, Artificial Neural Networks (ANN) have been demonstrated to predict accurate wakes over a mesh. In this work, a new mesh-free ANN-based wake model is proposed. This new model can predict the flow everywhere in the domain, and as it employs smooth activation functions, it is suitable for gradient-based optimization.

Two ANNs were trained with data generated by Reynolds-Averaged Navier-Stokes with Actuator Disc simulations for several yaw angles. The first ANN predicts streamwise wake velocity induction/deficit, the second ANN predicts added wake turbulence intensity. Both ANNs predict with a low error.

1. Introduction

Historically, Computational Fluid Dynamics (CFD) has been prohibitively expensive to compute the Annual Energy Production (AEP) of a wind farm. Instead, AEP can be computed quickly using engineering wake models [1, 2, 3]. Engineering wake models rely on a few parameters and equations to predict the flow in the far-wake. However, they fail to resolve the highly-complex flow in the near-wake and the induction zone. Surrogate models are used where existing engineering models are slow or inaccurate. In this work, an Artificial Neural Network (ANN) is used to train a regression model capable of predicting nonlinear behavior, including the transition from near- to far-wake and the induction zone.

Artificial neural network (ANN) based surrogate models concerning wake effects have been explored in a few works. These range from Auto-Encoder latent space models for control [4, 5], generative models for control [6], and stationary wake flow for load prediction [7].

Some authors have previously addressed engineering surrogate models for AEP calculations, most notably Ti *et al.* and Yang *et al.* [8, 9]. They demonstrated that these models can make predictions with accuracy close to the training data. However, both of these works make the same compromise, as they predict on a fixed grid. This has the benefit of allowing the training to be parallelized into many sub-models, each predicting at a single coordinate. However, it also has three drawbacks. Firstly, it adds an extra interpolation step. Secondly, it increases the number of required parameters, as it must be assumed that a high degree of redundancy is present. Finally, interpolation inhibits direct differentiation of the model. By contrast, the surrogate proposed in this work is capable of making predictions between coordinates and is directly differentiable.

The proposed model is intended as a substitute for classical engineering wake models. Therefore, the premise of the model is that it is possible to separate the flow physics into components, such as deflection and super-position models. The concept of modularizing the flow is an approximation, and as such, should be tested against fully-coupled models. In this work, it is assumed that the error introduced by this approximation is low, but it is not investigated further.

The ANN is trained using data from Reynolds Averaged Navier-Stokes with Actuator Disc (RANS-AD) simulations. The accuracy of RANS-AD in the near wake, although better than engineering models, is not generally considered good. In this work, RANS-AD is used to estimate the feasibility of using an ANN as a wake surrogate. Future works might transfer the proposed methodology to a higher accuracy CFD model, such as large eddy simulation (LES), to increase the accuracy of the ANN. However, it should be noted that in wind farm design optimization, the near wake zone is of lesser importance.

In Section 2, the methods used are summarized, in Section 3, the results are presented and discussed, and lastly, in Section 4, a conclusion is made.

2. Methodology

2.1. Reynolds averaged Navier-Stokes with Actuator Disc (RANS-AD)

In this work, RANS-AD simulations are used to generate a dataset for training an ANN. The k-ε-f_P eddy viscosity model [10, 11] is applied to model the Reynolds stresses. For the dataset, a single wind turbine in a neutral atmospheric surface layer is considered with a hub height wind speed, U_∞, of 11 m/s, and an ambient hub height turbulence intensity of 5%. In addition, a range of yaw angles are considered ($\gamma = \{-20°, -18°, \ldots, 20°\}$). For the zero yaw case the resulting thrust coefficient, C_T, is 0.81.

This figure illustrates the grids used in our study, with every third grid line included. The colored borders represent the subsets used for training artificial neural networks (ANNs), with blue and red solid squares denoting the xy and xz ANN data borders, respectively. A linear grid border is also shown in blue dashed lines. Panel (a) shows a top-view of the xy plane, while panel (b) displays a side view of the xz plane.

The AD is a permeable disc of force used to represent the wind turbine rotor. The AD force is distributed based on an analytic rotor model that assumes a constant circulation, employs root and tip corrections, and uses a 1D momentum relationship for the local reference velocity to account for non-uniform inflow conditions, as shear and veer [12]. The analytic rotor model includes tangential forces that induce wake rotation. The considered power, thrust and rotational speed curves are based on the DTU 10MW reference wind turbine [13]. The RANS-AD simulations are performed within the `PyWakeEllipSys` framework [14], which employs the finite volume code `EllipSys3D` [15, 16].

The RANS-AD simulations are run on the Sophia computer cluster [17]. The considered grid is Cartesian, and visualized in figure 1, where x, y and z are the streamwise, lateral and vertical coordinates, respectively. A refined region around the AD is applied within $-4 < x/D < 20$, $-2.5 < y/D < 2.5$ and $0.5\,\mathrm{m/D} < z/D < 3$, where the horizontal spacing is set to $D/12$, following a previously performed grid refinement study [10]. The first cell height is set to $0.005D$ and the subsequent cells grow moving away from the wall with a maximum growth ratio

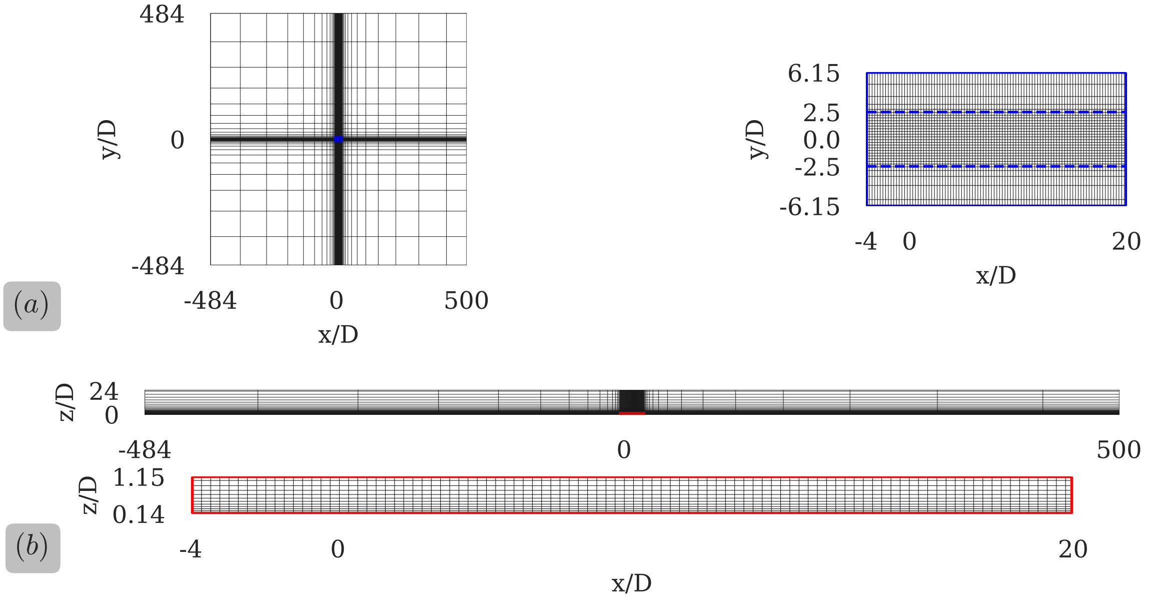

Figure 1: Illustration of the grids used in this study, with every third grid line included. Colored solid lines represent the subset used for training ANNs. (🟦: xy ANN data border 🟥: xz ANN data border). (**a**) Top-view of xy plane, - - - -: transition to logarithmic grid (**b**) Side view of xz plane.

of 1.2 and maintaining a cell spacing of $D/12$ or less up to $z = 3D$. The inflow represents a logarithmic profile wind speed corresponding to a neutral atmospheric surface layer, set as inflow boundary conditions at the inlet plane ($x/D = -484$) and at the domain top ($z/D = 25$), hence, the flow is lid driven. At the bottom, a rough wall boundary condition is set using a uniform roughness length obtained from the chosen ambient turbulence intensity at hub height. The lateral boundaries are periodic and at $x/D = 500$ an outflow boundary is applied at which all normal gradients are set to zero. More details on the numerical setup can be found in previous work [10, 11].

2.2. Artificial neural network (ANN)

The surrogate model presented in this work is a fully-connected feedforward ANN, a type of machine learning model, also known as Multi-Layered-Perceptron or Deep Feedforward Network [18]. For the remainder of the article the model will be referred to in short as ANN.

Among the various types of surrogate models, the ANN was chosen because, with a sufficiently high complexity, it can approximate any function [19, 20]. However, this does not guarantee that a suitable architecture is easily found or that suitable weights can be obtained through training [18]. Finding a suitable architecture is, therefore, a matter of experimentation, often referred to as hyper-parameter search. This involves exploring different combinations of hyper-parameters such as the number of hidden layers, the number of neurons in each layer, the choice of activation functions etc. The objective of a hyper-parameter search is to find the optimal combination of hyper-parameters, where optimal means to achieve the highest accuracy.

The inputs to the model are the spatial coordinates and the turbine yaw angle (x, y, z and γ). Two models are trained: one for the streamwise velocity deficit, Δu, and one for the added wake turbulence intensity, ΔTI. These outputs are defined as

$$\Delta u = 1 - \left(\frac{u}{U_\infty}\right) \tag{1}$$

$$\Delta \mathrm{TI} = \mathrm{TI} - \mathrm{TI}_\infty \tag{2}$$

where u is the wind speed component in the streamwise, x, direction. U_∞ is the free-stream wind speed. TI is the streamwise turbulence intensity and TI_∞ the ambient turbulence intensity.

The ANNs presented in this work consists of an input layer, 5 equally-sized consecutive hidden layers and an output layer. Each hidden layer consists of 60 neurons. Each neuron is connected to all neurons in the previous layer and all neurons in the next layer, hence the name fully-connected network. The output of a neuron is termed the activation, a neurons activation is dependent on the chosen activation function and trainable parameters called weights and biases. Figure 2 shows an illustration of the proposed ANN structure and table 1 shows a model summary of the chosen hyper-parameters.

In this study, following an an initial exploration stage two hyper-parameter searches were conducted, in both searches the Δu ANN was considered. The first search focused on the depth of the ANN with a fixed layer width of 100 neurons. The depth of the ANN was varied between 3, 5, and 7 layers, the 5-layer model was found to perform the best. In the second hyper-parameter search the 5-layer ANN architecture previously identified was considered. Layers 1 and 5 was reduced to 60 neurons while layers 2, 3, and 4 was allowed to vary between 40, 60 and 80 neurons, resulting in a total of 30 possible architectures considered. During the hyper-parameter searches the models were trained for 1000 epochs. Later the resulting model architecture was further trained for an additional 1000 epochs. The ΔTI ANN uses the same resultant architecture and was adopted without further studies, it was found that 1000 epochs was sufficient for the ΔTI ANN. Each considered ANN was trained on an 32 core node on the Sophia cluster [17] with a training cost of approximately 2000 core hours (ch) pr. 1000 epochs.

In the searches only tanh was considered as the activation function. tanh is preferred over the ReLU, as a fully-differentiable ANN is an advantage in gradient-based optimization, see for example [7]. The ANN has been trained using the Adam optimizer [21], which is a variation of Stochastic Gradient Descent, with the addition of first- and second-order momentum terms.

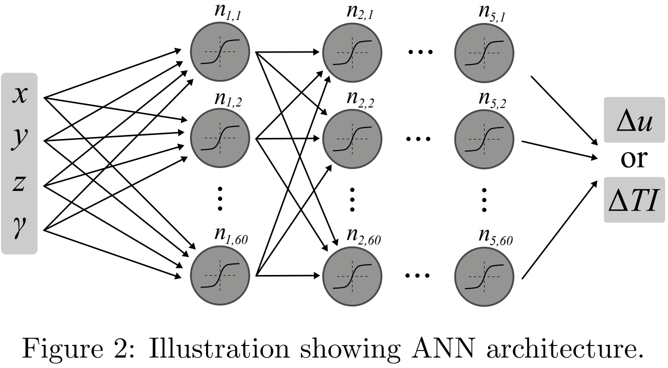

Figure 2: Illustration showing ANN architecture.

Table 1: ANN summary.

Hyper parameter	Value
No. layers	5
Neurons pr. layer	60
Trainable parameters	15001
Activation	tanh
Optimizer	Adam
Feature scaling	z-score (3)
Loss ($\mathcal{L}$)	MSE, (5)
Learning rate	1×10^{-3}
Epochs (Δu)	2000
Epochs (ΔTI)	1000

Before training the ANN, the input database has been processed using a standard scalar, defined as

$$\boldsymbol{z} = (\boldsymbol{a} - \boldsymbol{\mu_a}) \oslash \boldsymbol{\sigma_a} \tag{3}$$

where $\boldsymbol{a}$ is an input, $\boldsymbol{\mu_a}$ the mean, $\boldsymbol{\sigma_a}$ the standard deviation and $\oslash$ the Hadamard division operator. Feature scaling is performed to make training easier by alleviating both the vanishing and exploding gradient problems. The scaling is not applied to the output, because the deficit and added wake turbulence are already normalized quantities.

The ANNs were trained with the `TensorFlow` [22] framework and `Hydra` [23] for managing the models.

2.3. Regression metrics

To evaluate the performance of trained ANN models, three different regression metrics are considered. The R^2 coefficient of determination, MSE and Normalized Root Mean Square Error (NRMSE).

$$R^2 = 1 - \left(\frac{\sum_i \left(b_i - g(\mathbf{a}_i)\right)^2}{\sum_i \left(b_i - \bar{b}\right)^2} \right) \tag{4}$$

$$\mathcal{L} = \text{MSE} = \frac{\sum_i \left(b_i - g(\mathbf{a}_i)\right)^2}{N} \tag{5}$$

$$\text{NRMSE} = \frac{1}{\sigma_b} \sqrt{\frac{\sum_i \left(b_i - g(\mathbf{a}_i)\right)^2}{N}} \tag{6}$$

Where the i subscript indicates a sample, b the target value, $g(\mathbf{a})$ the model prediction, σ_b the target standard deviation and N the total amount of samples considered.

3. Results and discussion

In this section, the performance of the ANNs are presented and compared to the RANS-AD data. The results are presented in both quantitative metrics (tables 2 and 3) and qualitative as contour plots (figures 3, 4 and 5) and mid-plane velocity / TI profiles (figures 6 and 7).

An experiment was conducted to estimate the evaluation time of select engineering models and the ANN. The accuracy of engineering models compared to RANS-AD is a relevant related topic, but it is deemed outside the scope of this article. Interested readers are referred to Criado 2023 [24]. All of the engineering models tested were evaluated with `PyWake` [25], an open source AEP calculation tool. The ANN and engineering models were evaluated on a single node of the Sophia cluster while the RANS-AD simulations were conducted on 8 nodes with 32 cores each. The reported RANS-AD timing was normalized with respect to the number of cores for comparability i.e. the timings are reported in core hours. Both the engineering models and the ANN timings are reported as an average of 100 realizations. As can be seen from table 2, the ANN is 2-3 times slower than the conventional engineering models. It should be noted that this is a rough estimate on a single setup and results may vary depending on hardware and software implementations.

Table 2: Wake model, ANN and RANS-AD timings (*time for one yaw angle).

N.O Jensen [1]	Bastankhah [2]	Zong [3]	ANN	RANS-AD*
0.3 ms	0.3 ms	0.5 ms	0.9 ms	$\sim$ 16 ch

Before training, the dataset was randomly shuffled and split into three parts: 70% for training, 10% for validation and 20% for testing. The regression metrics are evaluated on the test set and reported in table 3 alongside error $\left(\varepsilon = \frac{(U_\infty - b) - (U_\infty - g(\mathbf{a}))}{U_\infty - b}\right)$ and residual (r) statistics. As can be seen from table 3 the metrics are generally low, but the maximum absolute error is slightly high and warrants further investigation.

Table 3: Regression metrics and error statistics evaluated on the test set and residual statistics evaluated on the training set.

Output	R^2	$\mathcal{L} = \text{MSE}$	NRMSE	$\lvert\varepsilon\rvert_{\max}$	μ_ε	σ_ε	$\lvert r\rvert_{\max}$	μ_r	σ_r
Δu	1.00	1.10×10^{-6}	0.15	5.93 %	0.08%	0.10 %	6.57%	0.08%	0.10%
ΔTI	1.00	1.83×10^{-8}	5.38×10^{-3}	0.50%	8.00×10^{-5}	0.01%	0.48%	7.98×10^{-5}	0.01%

In figures 3 and 4 contour plots of respectively the xy-plane at hub height (z_{hub}) and xz-plane at ($y = 0$) are shown of ANN predictions and prediction errors. When visually investigating figure 3 (a), it is observed that the blockage around the turbine takes an unexpected shape. It is hypothesized that this occur because data is scarcer for $|y| > 2.5D$, which is where the transition from a linear to a logarithmic grid occur. During training of the ANN, an uneven distribution of data means that a larger emphasis is placed on denser areas, while scarce areas will have a smaller influence on the training loss.

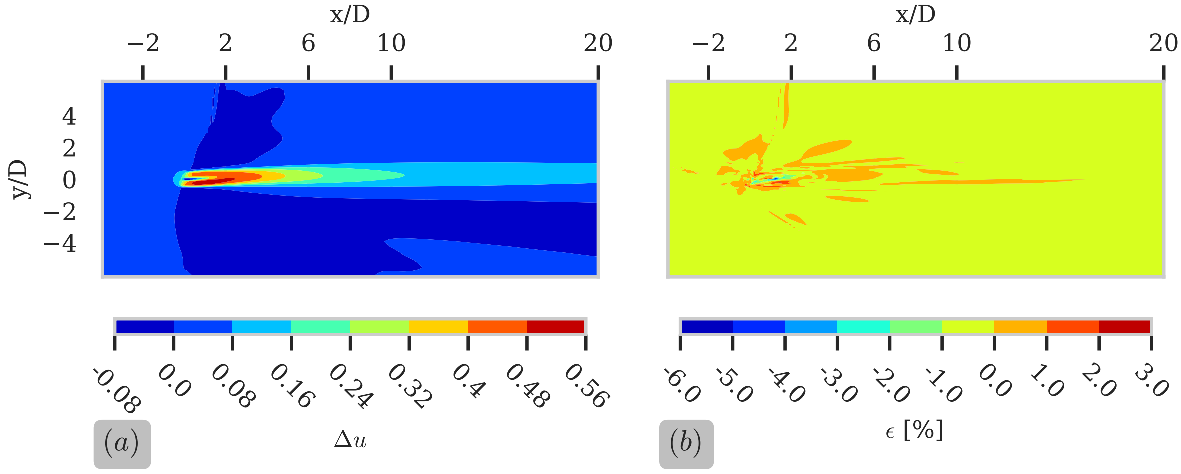

Figure 3: Model prediction and relative error at z_{hub} and $\gamma = 20°$ over the whole input domain. (a) ANN prediction of velocity deficit Δu. (b) Error with respect to RANS-AD.

For both the error plot in the xy- and xz-plane the maximum error occur close to the turbine. In this area the wake physics are more complex and therefore it is expected to be the area of least accuracy. This could indicate either that the ANN architecture is at its limit or the model is over/under-trained. This can be alleviated with better tuning of hyper-parameters or applying a more advanced training strategy. Identified training strategies that could improve the results include: Sample weighting, re-sampling training data, learning-rate schedulers and weight-decay.

Finally in figure 5, the ANN is further investigated by considering the cross-stream plane (yz) at different downstream locations. The downstream location with the most activity is at $x = 2D$ where the contour show that the ANN is capable of producing an accurate approximation of the counter-rotating vortex pair (kidney bean shape). This is particularly interesting as this is a part of the physics engineering wake models sacrifice due to the simplified physics [26].

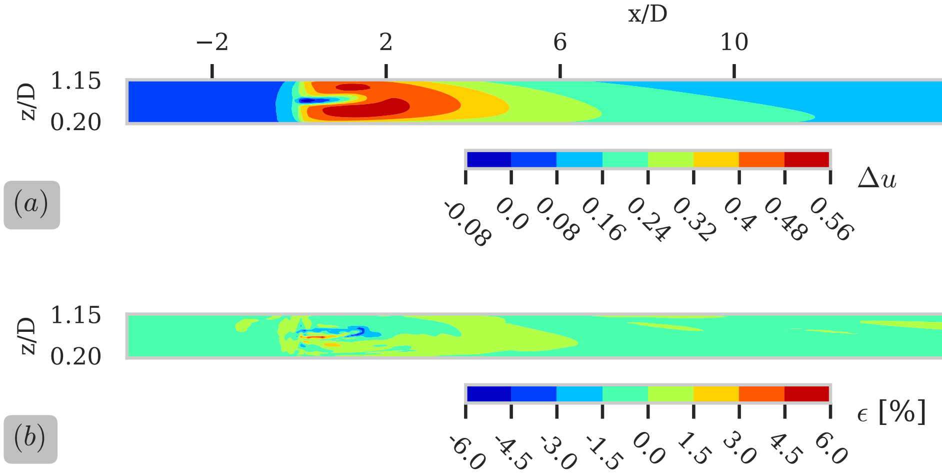

Figure 4: Model prediction and relative error at $y_{center} = 0$ and $\gamma = 20°$ over the whole input domain. (a) ANN prediction of Δu. (b) Relative error with respect to RANS-AD.

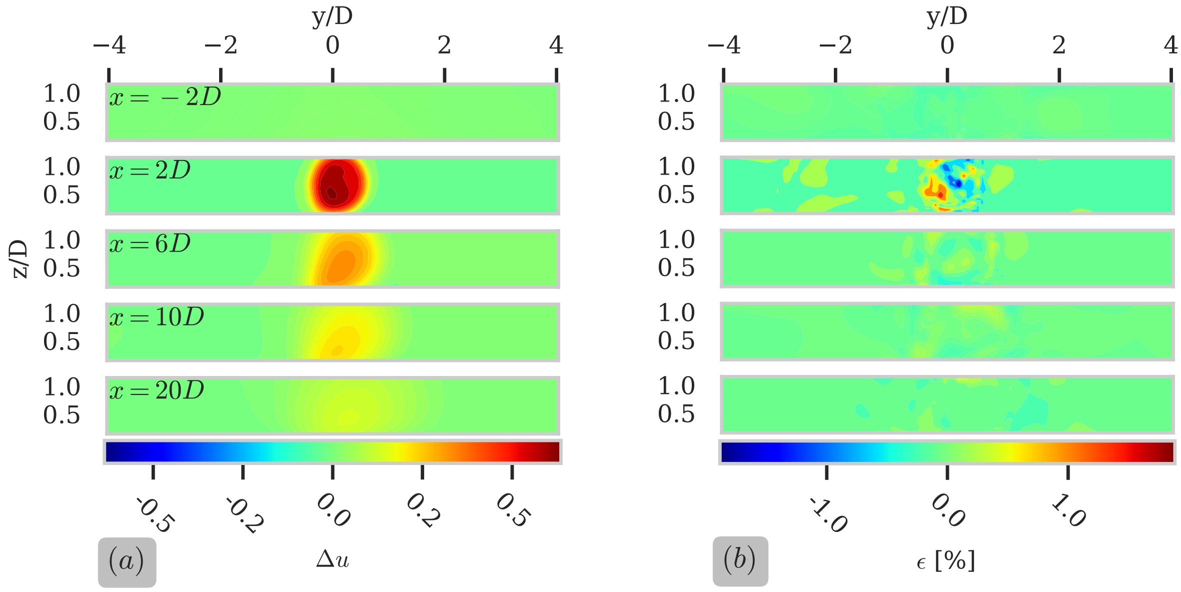

Figure 5: Model prediction and relative error at different downstream locations (x) for re-combined test/training data and $\gamma = 20°$ (a) ANN prediction of Δu. (b) Relative error with respect to RANS-AD.

In figure 6, mid-plane velocity profiles are shown to investigate how the ANN predicts between data grid points i.e. accessing if the model interpolates as expected. This study was performed by increasing the prediction resolution by a factor of 100 compared to the RANS-AD grid. In figure 6 the areas of lowest accuracy are the same as previously identified. In figure 6 (b) the zoom shows that the model is not over-trained. This is indicated by the fact that the predictions are not calibrated exactly to the training data.

Figure 7 shows mid-plane profiles for wake added TI. Unsurprisingly the largest error occur close to the turbine, re-affirming the difficulty of approximating areas with high rates of change.

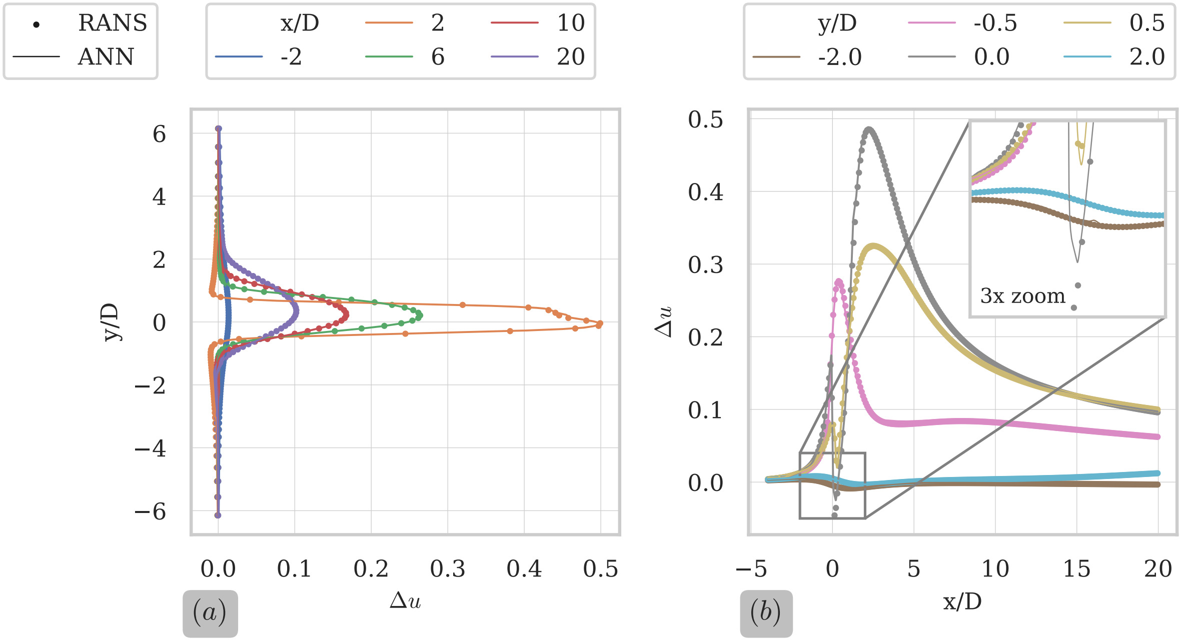

Figure 6: Model predictions (solid lines) and RANS-AD (points) at z_{hub} and $\gamma = 20°$. Predictions are 100x denser than RANS-AD points. (**a**) Δu at different up-/down-stream locations as function of y. (**b**) Δu at different cross-stream locations as function of x.

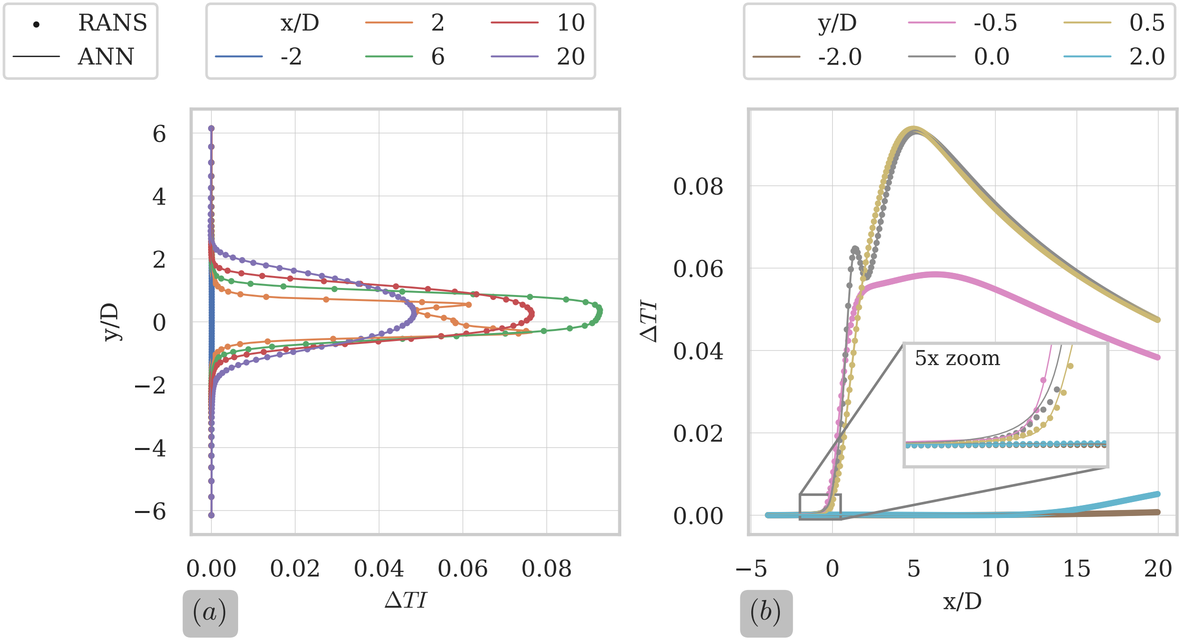

Figure 7: Model predictions (solid lines) and RANS-AD (points) at z_{hub} and $\gamma = 20°$. Predictions are 100x denser than RANS-AD points. (**a**) ΔTI at different up-/down-stream locations as function of y. (**b**) ΔTI at different cross-stream locations as function of x.

The area where the ANN predicts with the highest error occur close to the turbine ($\sqrt{x^2+y^2} \leq 2D$). As this area is usually excluded by separation constraints during wind farm layout optimization problems, the performance outside this area is therefore of interest. Ignoring this area when evaluating the error, the absolute max error drops to $|\varepsilon|_{max} = 2.46\%$. For the purpose of replacing engineering wake models this can be considered an improvement. This is further emphasized by the fact that variations in the results of RANS vary more with respect to the chosen turbulence formulation [27] than the ANN does with respect to the training data.

In figure 8, wake center trajectories calculated from ANN predictions and RANS-AD data are compared. The wake center trajectories are calculated at hub height by employing least squares regression to fit one dimensional Gaussian distributions downstream. Figure 8 shows that the ANN is capable of recreating the wake center trajectory with a low approximation error, however the rescaled y axis in figure 8(b) makes it clear that the discontinuity is not approximated at the correct downstream location. Although the error is considered small this should be investigated further in future works.

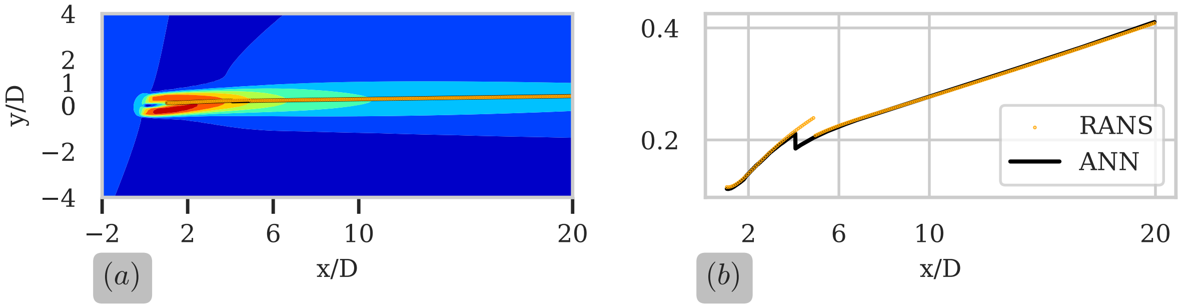

Figure 8: Wake center trajectories for $\gamma = 20°$ at hub height. (**a**) Flow contour with wake trajectories (**b**) Stand-alone wake trajectories

To improve the ANN based surrogate some further work is proposed. Firstly, the model itself could be improved by the following alterations:

- Investigate how the RANS-AD mesh can be improved to train the ANN more efficiently.
- Perform a more thorough hyper-parameter search.

Secondly, some further developments are proposed to equal and exceed the capabilities of existing wake engineering models.

- Include additional turbine characteristics as inputs e.g. C_T, h/D, tilt.
- Include atmospheric conditions as inputs e.g. ambient turbulence intensity, stability.
- Additional outputs: velocity components (v and w).

The proposed improvements of the model will increase the models capabilities but also increase the complexity of the physics and data requirements. Therefore future works should consider the scalability of the model. This includes the evaluation cost but also costs of simulation and training. It is expected that evaluation cost will stay close to the current cost, however it could be significantly more expensive to simulate and train the ANN.

4. Conclusion

In this work, a new ANN based mesh-free wake surrogate model was proposed. The ANN was configured as a regression model and trained with RANS-AD wake deficit data of a single wind turbine. The ANN model performs a prediction in 0.9 ms, 2-3 times slower than existing wake engineering models. Compared to performing interpolation on pre-calculated CFD data, the proposed ANN is differentiable and requires little on disk memory. In addition the accuracy of added wake turbulence engineering models is generally considered poor, while the proposed ANN reproduces CFD results accurately.

The wake deficit ANN predicts with an average error of 0.08% and a maximum error of 5.93%. In addition an added TI model was trained capable of predicting with a lower mean error of 8.00×10^{-5} and max absolute error of 0.50%.

As the ANN is intended for use in layout optimization, it was deemed that some areas can be excluded in the assessment. If only the area of interest is considered it was found that the max error of the wake deficit model dropped further to 2.46%.

In summary, the ANN was found to demonstrate accuracy near the capability of RANS-AD, with only a moderate decrease in prediction speed compared to engineering models. Establishing the methodology as a viable alternative to classical wake engineering models.

References

[1] Jensen N O 1983 *Technical report Risø-M-2411*

[2] Bastankhah M and Porté-Agel F 2014 *Renewable Energy* **70** 116–123 ISSN 09601481

[3] Zong H and Porté-Agel F 2020 *Journal of Fluid Mechanics* **889** ISSN 14697645

[4] Chen K, Lin J, Qiu Y, Liu F and Song Y 2021 *Control Engineering Practice* **116** ISSN 09670661

[5] Renganathan S A, Maulik R, Letizia S and Iungo G V 2022 *Neural Computing and Applications* **34**(8) 6171–6186 ISSN 14333058

[6] Zhang J and Zhao X 2022 *Energy* **238** 121747 ISSN 03605442

[7] Riva R, Liew J, Friis-Møller M, Dimitrov N, Barlas E, Réthoré P E and Beržonskis A 2020 *Journal of Physics: Conference Series* **1618**(4) 042035 ISSN 1742-6588

[8] Ti Z, Deng X W and Zhang M 2021 *Renewable Energy* **172** 618–631 ISSN 18790682

[9] Yang S, Deng X, Ti Z, Yan B and Yang Q 2022 *Renewable Energy* **193** 519–537 ISSN 18790682

[10] van der Laan M P, Sørensen N N, Réthoré P E, Mann J, Kelly M C, Troldborg N, Schepers J G and Machefaux E 2015 *Wind Energy* **18**(5) 889–907 ISSN 10991824

[11] van der Laan M P, Sørensen N N, Réthoré P E, Mann J, Kelly M C and Troldborg N 2015 *Wind Energy* **18**(12) 2223–2240 ISSN 10991824

[12] Sørensen J N, Nilsson K, Ivanell S, Asmuth H and Mikkelsen R F 2020 *Renewable Energy* **147** 2259–2271

[13] Bak C and *et al* 2013 The DTU 10-MW reference wind turbine

[14] DTU Wind and Energy Systems 2023 PyWakeEllipSys v3.0 `https://topfarm.pages.windenergy.dtu.dk/cuttingedge/pywake/pywake_ellipsys` accessed: 23/01-23

[15] Michelsen J A 1992 Basis3d - a platform for development of multiblock PDE solvers. Tech. rep. DTU

[16] Sørensen N N 1994 *General purpose flow solver applied to flow over hills* Ph.D. thesis DTU

[17] Technical University of Denmark 2019 Sophia HPC cluster `https://dtu-sophia.github.io/docs/` dOI 10.57940/FAFC-6M81

[18] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (The MIT Press) pp 193–199 ISBN 978-0262035613

[19] Cybenko G 1989 *Mathematics of Control, Signals, and Systems* **2**(4) 303–314 ISSN 0932-4194

[20] Hornik K, Stinchcombe M and White H 1989 *Neural Networks* **2**(5) 359–366 ISSN 08936080

[21] Kingma D P and Ba J 2014 *International Conference on Learning Representations*

[22] Abadi M and *et al* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems `https://www.tensorflow.org/` software available from tensorflow.org

[23] Yadan O 2019 Hydra - a framework for elegantly configuring complex applications `https://github.com/facebookresearch/hydra`

[24] Criado Risco J, van der Laan M P, Pedersen M M, Forsting A M and Réthoré P E 2023, forthcoming

[25] Pedersen M M and *et al* 2022 URL `https://gitlab.windenergy.dtu.dk/TOPFARM/PyWake`

[26] Bastankhah M and Porté-Agel F 2016 *Journal of Fluid Mechanics* **806** 506–541 ISSN 0022-1120

[27] van der Laan M P and Andersen S J 2018 *Journal of Physics: Conference Series* **1037**(7) 072001 ISSN 1742-6588