Applied Energy 376 (2024) 124116

Contents lists available at ScienceDirect

Applied Energy

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

Augmenting insights from wind turbine data through data-driven approaches

Coleman Moss[a], Romit Maulik[b], Giacomo Valerio Iungo[a,*]

[a] *University of Texas at Dallas, 800 W. Campbell Rd., Richardson, TX, 75080, USA*
[b] *College of Information Sciences and Technology, The Pennsylvania State University, State College, PA, 16801, USA*

ARTICLE INFO

Keywords:
Machine learning
SCADA data
Wind turbine
Wind farm

ABSTRACT

Data-driven techniques can enable enhanced insights into wind turbine operations by efficiently extracting information from turbine data. This work outlines a data-driven strategy to augment these insights, describing its benefits and limitations. Different data-driven models are trained on supervisory control and data acquisition (SCADA) and meteorological data collected at an onshore wind farm. The developed models are used to predict wind speed, turbulence intensity (TI), and power capture for each turbine with excellent accuracy for different wind and atmospheric conditions. Modifications of the incoming freestream wind speed and TI due to the evolution of the wind field over the wind farm and effects associated with operating turbines are captured enabling modeling at the turbine level. Farm-level modeling is achieved by combining models predicting wind speed and TI at each turbine location from inflow conditions with models predicting power capture. Data-driven filters are also considered in the context of generating accurate data-driven models. In contrast to many current works that utilize simulated data, the proposed approach can describe subtle phenomena, such as speedups, TI damping, and wake-generated turbulence, from real-world turbine data. It is noteworthy that the accuracy achievable through data-driven modeling is limited by the quality of the data; therefore, guidelines are proposed to estimate resultant model performance from a given training set without the need to train or test a model.

1. Introduction

The wind energy sector continues to grow rapidly to meet the ever-growing energy need and to address the rising concerns associated with fossil fuel use, of which the most prominent is greenhouse gas emission and the associated warming effects. High targets are set for future wind energy capacity [1] while current production costs drop [2]. With the growing number of wind turbines installed, larger data sets are collected from more turbines and other instruments to monitor the performance, power production, operating conditions, and life span of the wind turbines. Contained in this data is a wealth of information that can improve the current understanding of wind turbine operation and unlock new insights into turbine design and performance. Much of this information, however, remains untapped due to the overwhelming amount of available data and the lack of guidelines or procedures to analyze these data sets.

At the same time, machine learning and artificial intelligence have been advancing at an extremely rapid pace. Various neural network models have been applied to a strikingly broad array of problems with great success. Yet the successful application of these models is not trivial and requires careful engineering of the networks and the data on which they run. Machine learning has been used to solve many wind energy problems, such as wind speed forecasting [3,4], component fatigue and failure monitoring [5], component design [6], power production modeling [7], and wake modeling [8–12]. It may seem that machine learning has thoroughly penetrated the wind energy field; however, machine learning has been used relatively little to extract information from real-world data and instead has generally been used to generate surrogate models of complex first-principle models while reducing the required computational cost [13].

Machine learning has been broadly used for modeling wind turbine power curves upon training on real-world data [7,14–18]. For operations in region two of the turbine power curve (i.e., for incoming wind speeds between cut-in and rated wind speed), wind turbines theoretically produce power in a cubic relationship with the incoming wind speed [19]. However, real-world turbines do not follow this cubic relationship perfectly due to wind heterogeneity over the rotor area [20], blade aging [21], sub-optimal control operations [22], and wake interactions [23], among other reasons. Thus, quantifying the empirical relationship between incoming wind speed and power production is an important problem since it enables more accurate estimates of energy production. Furthermore, power may vary with respect to other environmental parameters, such as wind turbulence

[*] Corresponding author.
E-mail address: valerio.iungo@utdallas.edu (G.V. Iungo).

https://doi.org/10.1016/j.apenergy.2024.124116
Received 25 April 2024; Received in revised form 24 July 2024; Accepted 31 July 2024
Available online 14 August 2024

intensity (TI) and shear [24–26]. Since these interactions are complex and nonlinear, they are suitable applications for machine learning. As a result, many works have used machine learning to develop better power curve models and to better understand the relationship between the incoming wind field and turbine power capture [15–17,27].

Unlike power curve modeling, if we consider the field of wake modeling, we find relatively few cases where real-world data were used. In general, the goal of wake modeling is to describe the velocity deficit downstream of an operating turbine induced as it extracts kinetic energy from the wind and to predict its downstream recovery affected by the background atmospheric conditions [28]. This velocity deficit is a function of the turbine – most importantly, the turbine rotor diameter and the thrust the turbine exerts on the wind – as well as the inflow conditions, including wind speed and TI. This velocity deficit can be simulated very accurately using large eddy simulation (LES) or Reynolds-Averaged Navier–Stokes (RANS) solvers [29–31]. These solvers are computationally expensive, hampering their application for tasks requiring a large number of simulations, such as optimization of the wind farm layout or wind farm control. While low-cost analytical (engineering) models exist and can be used with some accuracy, their accuracy is generally limited when compared to CFD models [32].

More attention has therefore been given to machine learning models that can recreate numerical solutions with computational costs closer to the analytic solutions. These will be referred to as machine learning surrogate models [13]. While the machine learning surrogate models are useful in many ways since they provide a low-cost way to estimate wake velocity deficits with high accuracy, they do not generally advance the understanding of the physics underpinning the evolution and interactions of wind turbine wakes. Analysis of real-world data is needed to accomplish this goal. While several studies have used statistical methods to approach this problem [25,33–35], machine learning has been relatively underused.

To illustrate this point, we consider several recent studies in machine learning modeling of wind turbine wakes. Ti and coworkers developed early models that trained artificial neural networks to predict the wake deficit field in three dimensions using a CFD-generated training set, and then expanded this approach to farm modeling using analytic wake overlapping methods [8,9]. This approach of training models on high-fidelity simulation creates a reduced order model of the high-fidelity simulation that can quickly be executed. Thus, they are useful in wind farm control cases [36]. Convolutional neural networks (CNNs) have also been used as reduced-order models, such as in [37]. Starting from CFD data, Zhang and Zhao trained a generative adversarial network (GAN) that can predict yawed wake behavior [38]. Graph neural networks (GNNs) also hold promise in wind farm modeling, as they can generalize wake behavior from a given farm to any arbitrary farm layout, if properly constructed. Starting from data generated by analytic wake models, recent works have demonstrated the applicability of these models to wind farm modeling [39,40]. Using similar analytic models, Zhou and coworkers developed machine learning approaches that generalize by computing initial wake effects from the analytic models [41]. Using high-fidelity data, GNNs can be used to predict three-dimensional wake effects on arbitrary grids in the region downstream of an operating turbine [42].

Digital twins that solve the governing equations of turbulent flows while matching measured data can also be efficiently created using neural networks, but require high-fidelity data to validate, and have not yet been extended to field cases [43]. Recent works have refined wake models using LES training data [44,45]. Works using SCADA data to investigate wake effects through machine learning methods are scarce. For instance, Sun and coworkers developed wake models based on SCADA data, but also inject analytic wake model results to the models [46]. Recent works have also tried generating graph-based models using SCADA data, but solve graph weights between turbines using minimization techniques rather than learning by modeling techniques [47].

Table 1
Technical details of the wind turbines under investigation.

Manufacturer	Siemens	Model	SWT-2.3-108
Year online	2014	Rated capacity	2.3 MW
Hub height	80 m	Rotor diameter	108 m
Cut in wind speed	3 m s^{-1}	Rated Wind Speed	11 m s^{-1}
Cut out wind speed	25 m s^{-1}	No. Turbines	25

The current work seeks to augment insights uncoverable from real-world noisy turbine data with the aim of predicting wake interactions, wake-generated turbulence, and power losses at the turbine level with high accuracy not easily achievable through classical statistical approaches or other reduced order models. To this aim, we have developed data-driven models to predict wind speed, TI, and power capture at individual turbines as a function of estimated freestream/reference conditions. These models are then probed to explain wind farm flow processes and turbine power performance by varying the model input parameters. The resulting models are free from limitations typically associated with statistical analyses, such as *a priori* determination of parameters' ranges for data binning, while letting data features surface naturally, such as to identify the occurrence of speedups or quantify wake-generated turbulence.

To use this data-driven approach to extracting information, though, care needs to be taken to ensure that the models are properly interpreted. If the models are probed over a region of inputs where there are few to no data points, they can extrapolate with varying degrees of accuracy. Additionally, the quality of the training data can directly impact the accuracy of the resulting models, and therefore reliability of any conclusions drawn from probing the models. A significant contribution of this work is therefore to understand the limitations real-world data places on data-driven models. This task is approached from the perspective of a practitioner in wind energy. As such, all the models considered are readily available for typical users and can be implemented through various Python packages. While this work does not present any new models, we believe that the framework we propose will be useful for wind energy practitioners or researchers seeking to efficiently work with large amounts of data, while also highlighting an area of machine learning not commonly considered in the wind energy field, namely, frameworks to explain and interpret data. The models developed here are also specific for the wind turbines probed for generating the training data set, thus they need to be re-trained for each different wind farm. However, the proposed framework for data mining and extracting physical insights is generalizable.

The remainder of this work is organized as follows. Section 2 discusses the data used to train the data-driven models. The filtering applied to the data is discussed in Section 3 with more details listed in Appendix. After filtering, the optimal data-driven models are selected in Section 4. Section 5 shows that the selected models can accurately reproduce data behaviors. Section 6 discusses how data-driven methods can be used to extract information from the SCADA data while Section 7 illustrates how this method can outperform statistical approaches. The limitations on model performance and a method to predict model performance from the training data alone, without the need to train or test a model, are covered in Section 8. Section 9 offers closing remarks.

2. Data set and wind farm overview

The wind farm under consideration is located in the Panhandle of Texas and includes 25 wind turbines arranged in three rows roughly aligned along the East-West direction [25,48,49]. The details of the turbines are summarized in Table 1 while the site wind rose and wind farm layout are reported in Fig. 1(a) and (b), respectively.

Data were collected from a meteorological (met) tower starting on July 17th, 2014, and continuing until June 23rd, 2017. Specifically, wind speed, wind direction, ambient temperature, pressure, and

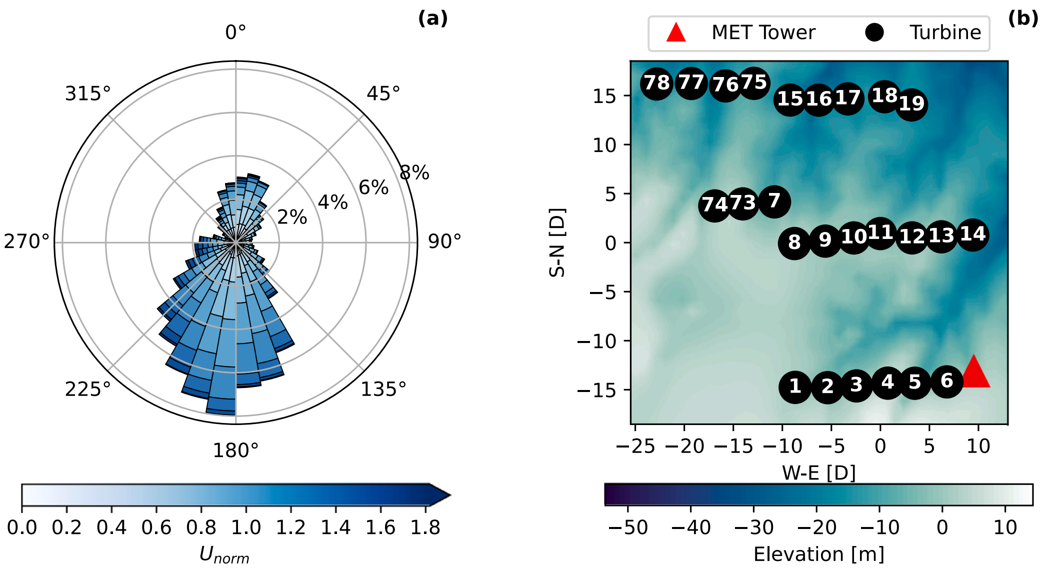

Fig. 1. Characterization of the wind turbine array: (a) windrose of the site with wind speed normalized by the turbine rated wind speed of 11 m s^{-1} and (b) layout of the wind turbines.

air density are provided as mean and standard deviation over 10-minute periods. After not-a-number (*NaN*) data rejection, a total of 150,132 time stamps are available, which correspond to a total time of 2.85 years.

SCADA data for every turbine were recorded from August 20th, 2015 up until April 15th, 2017. The data is composed of statistics over 10-minute periods for wind speed, TI (defined as the ratio between the wind speed standard deviation and its mean value), wind direction, ambient temperature, and power capture. The number of *NaN* data points varies from turbine to turbine. After removing these points, the average number of down-selected samples per turbine is 63,517 giving 1.21 years of data. These data are not necessarily continuous in time, though, as removing time stamps with *NaN* values causes discontinuities in time. The number of time stamps where all 25 wind turbines and the meteorological tower have non-*NaN* values is 40,720 for 0.77 years of data.

To investigate wind turbine performance, the freestream (reference) wind condition has to be characterized. To this aim, reference wind conditions are defined as the average of environmental conditions across all turbines not affected by wakes (unwaked turbines), as proposed in a previous work studying the site under consideration [25]. This procedure is used to define reference conditions for hub-height wind speed, TI, and direction. For wind speed monitored at each turbine through the SCADA, the wind speed correction is applied as follows:

$$U_{corr} = U\left(\frac{\rho}{\rho_0}\right)^{1/3}, \tag{1}$$

where ρ is the air density at each turbine, ρ_0 is the reference air density of 1.225 kg m^{-3}, and U is the mean wind speed. Since air density measurements are not available at each turbine, the best approximation is to calculate air density from the turbine ambient temperature in Kelvin, T, and the met-tower pressure in Pascals, P:

$$\rho = \frac{1}{T}\left[\frac{P}{R_0} - \phi\ a\ e^{bT}\left(\frac{1}{R_0} - \frac{1}{R_w}\right)\right], \tag{2}$$

where R_0 is the gas constant of dry air (287.05 J kg^{-1} K^{-1}), R_w is the gas constant of water vapor (461.6 J kg^{-1} K^{-1}), ϕ is the relative humidity (set to 0.5 since humidity measurements are unavailable), a is a constant equal to 0.0000205, and b is a constant equal to 0.0631846. Since each turbine's wind speed is density-corrected, the reference wind speed is also density-corrected. It should also be noted that measurements from anemometers mounted on turbine nacelles, behind the rotors, can be impacted by blade passage, turbine misalignment, and other factors. However, the manufacturer calibrates the anemometer to correct for these issues, so they are not considered important in this study.

A scaling factor is then defined to correct turbine 06, since it is closest to the met-tower, by reducing the bias between the wind speed measured at turbine 06 and at the met-tower. This correction is applied to the other turbines through the same scale factor estimated for turbine 06 [30,50]. In this way, bias errors in the turbine anemometers should be corrected or, at least, reduced.

3. Data filtering

SCADA data typically contain many outliers due to several factors, such as power curtailment, maintenance, off-design performance, or sensor fault. When training data-driven models, it can be important to remove outliers because they may jeopardize the accuracy of the models. Further, regions of the wind turbine power curve that are not of interest to an analysis should also be removed so that the model is not diluted by those points. Region one of the turbine power curve includes turbine operations for average wind speeds below the cut-in wind speed of 3 m s^{-1} and is characterized primarily by random fluctuations in power. Furthermore, since region-one operations occur very infrequently, they are filtered out. Additionally, region three, i.e. when the average wind speed is above the rated wind speed of 11 m s^{-1}, is not important for wake studies, since power capture is fixed at the rated power and the main turbine risks are associated with loads [51]. This region is removed by rejecting samples with an average wind speed above 13 m s^{-1}. This way, the transition from region two to region three is kept.

The turbines being studied have a rated power of 2.3 MW but are capable of boosting power output up to 2.5 MW under specific environmental conditions and high energy demand on the electricity grid. This power boost cannot be predicted, however, from environmental data alone, since it depends on grid conditions. For this reason, it is rejected by removing all points with power above 2325 kW. Finally, when the turbine rotor is being spun up to operating speed, the turbine may draw more power than it produces. If this is the case, the recorded power will be negative. These data points are of no interest and dilute the models, and, thus they are rejected.

In order to reject outliers from the data and create “clean” SCADA data, this study considers four types of outlier filters. The considered filters are a binning filter, a K-means clustering filter, an automatic Gaussian Process (GP) filter, and a novel data-driven filter, referred to as the General Machine Learning (GML) filter. These filters are applied for all 25 wind turbines, then an optimal filter was selected, namely the GML filter. For a more detailed discussion on filtering SCADA data outliers and filter selection, the reader is referred to Appendix.

Note that when developing models to predict wind speed or TI rather than power, the data used to train those models is left unfiltered. This is due to the more challenging nature of defining an outlier in these parameters due to increased variability from wake interactions and variability due to different inflow conditions.

4. Data-driven model selection

As mentioned above, the goal of this work is to describe turbine performance as a function of reference wind conditions and thereby extract information on turbine and farm performance from the SCADA data. To accomplish this goal, three models are needed for every turbine: a wind speed model, a TI model, and a power model. The wind speed model predicts the local wind speed at a given turbine location as a function of the reference wind conditions and captures the impact of neighboring turbines on wind speed, such as wakes and speedups. The TI model predicts the local TI at each turbine as a function of the reference wind conditions as well as the predicted local wind speed. Thus, neighboring turbine effects on TI are also captured. Finally, the power model predicts the power produced by each turbine as a function of local wind speed and TI. Essentially, it is a multi-dimensional power curve tuned to each individual turbine.

For each model, an appropriate data-driven model is selected and overall accuracy is discussed. While there are many different data-driven models of varying complexity, each suited to unique tasks and with unique strengths and weaknesses, we focus here on models that are easy to implement using simple and straightforward Python packages. We only use models from the Scikit-Learn package [52], with the exception of an XGBoost model using the XGBoost package [53] and shares the same syntax as Scikit-Learn models, and a neural network model using the Keras package with the Tensorflow backend [54], which can be easily approximated in Scikit-Learn using multilayer perceptron models. We simply use Tensorflow to take advantage of GPU-accelerated training. We focus on these highly accessible models so that general practitioners in the wind energy industry can quickly adopt the techniques discussed here without needing to also become experts in machine learning. For a full list of models considered and the approach to optimizing and selecting models, the reader is directed to Appendix A.2.

The inputs to the wind speed and TI models could be determined by a brute force analysis or careful design of experiments [55]. To keep the number of inputs low, reference wind speed, TI, and direction are selected as the inputs to these models. However, wind direction is a circular variable, but the data-driven models in use will treat it as a linear variable. To avoid potential discontinuities this may induce, the wind direction input is first converted into x and y components ($\cos\theta$ and $\sin\theta$, respectively) which are passed as inputs. The impact on model accuracy is negligible but the results avoid discontinuities at values of 0° and 360°.

To select a model for wind speed predictions, we optimize a Random Forest (RF), Extremely Randomized Trees (ET), Gradient Boosting (GB), Histogram Gradient Boosting (HGB), XGBoost (XGB), and dense neural network (NN) models using the procedure described in Appendix A.2. To assess the optimized models, each optimal model is trained on 80% and tested on the remaining 20% of data from every turbine. The wind speed limits on region rejection are increased to include all values between 2 m s^{-1} and 14 m s^{-1} to avoid boundary issues in the training. The root mean square error (RMSE) is reported for each turbine, normalized by the standard deviation of the wind speed in the testing set. From Table 2 listing the statistics of the results, it is evident that the HGB model has better performance than the other models. The optimized hyperparameters of this model are the following: learning rate is 0.083, maximum bins of 248, and maximum number of iterations is 660.

Table 2
Statistics on RMSE of wind speed predictions for all turbines in the wind farm comparing different optimized models. RMSE values are normalized by the standard deviation of wind speed. The minimum value in each column is reported in bold.

Model	Min.	25th %-ile	Median	75th %-ile	Max.
RF	20.81%	22.30%	23.89%	26.93%	27.52%
ET	20.71%	22.51%	23.94%	26.96%	27.66%
GB	20.52%	22.06%	23.60%	26.58%	27.25%
HGB	**20.17%**	**21.89%**	**23.41%**	**26.15%**	**26.75%**
XGB	20.44%	21.96%	23.51%	26.26%	26.81%

Table 3
Statistics on RMSE of TI predictions for all turbines in the wind farm comparing different optimized models. RMSE values are normalized by the standard deviation of TI. The minimum value in each column is displayed in bold.

Model	Min.	25th %-ile	Median	75th %-ile	Max.
RF	34.00%	36.51%	38.60%	40.94%	45.78%
ET	33.58%	36.27%	37.97%	40.71%	44.85%
GB	33.84%	35.98%	38.18%	41.29%	45.82%
HGB	33.68%	36.20%	37.88%	40.55%	44.75%
XGB	**32.98%**	**35.28%**	**37.38%**	**39.93%**	**44.60%**

Next, TI is to be predicted. The inputs for TI models are the same as for wind speed models, with the addition of the local wind speed at the turbine in question. Adding this input is found to increase the accuracy of the TI predictions. When training the TI models, inputs are retrieved directly from the SCADA data. In contrast, when using TI models, the wind speed model will be used to first predict the local wind speed. Once again, RF, ET, GB, HGB, XGB, and NN models are tuned using DeepHyper on turbine-6 data. The models are then applied across the farm using an 80%/20% training/testing split, and the RMSE normalized by the standard deviation of TI is calculated. Table 3 reports statistics on these values. From these results, the XGB model is identified as the best-performing model with the hyperparameters as follows: a learning rate of 0.003, a maximum depth of 30, 1950 estimators, and a subsample fraction of 0.11.

Finally, models to predict turbine power are considered. The inputs to these models are the wind speed and TI predicted by the wind speed and TI models discussed above, while the output is turbine power. In Appendix A.2, power models are discussed in the filtering context. In the prediction context, the models are to be trained on filtered data. To avoid re-optimizing models, it is assumed that the hyperparameters determined in Appendix A.2 are still optimal and that the best model is still the XGB model with optimal parameters listed in Appendix A.2.

5. Accuracy of data-driven models

Before using the data-driven models to identify features and physical processes from the SCADA data, it is important to first assess the accuracy of these models. Data-driven models are typically evaluated by performing a training and testing split, where the model is trained on a subset of data and then tested on another set. To achieve this training and testing split (roughly 80% of the data available for training and the rest reserved for testing), we define the training set to be all data recorded prior to January 1st, 2017, and the testing set to be all data recorded following that date. The data is not shuffled since points close in time generally are strongly correlated, and having correlated points in the training and testing sets can artificially increase model performance [56]. All of the results that follow in this section are reported from models trained on the training data set then tested on the withheld data.

First, the annual energy produced (AEP) and total farm power predictions are compared against the true values. Power predictions

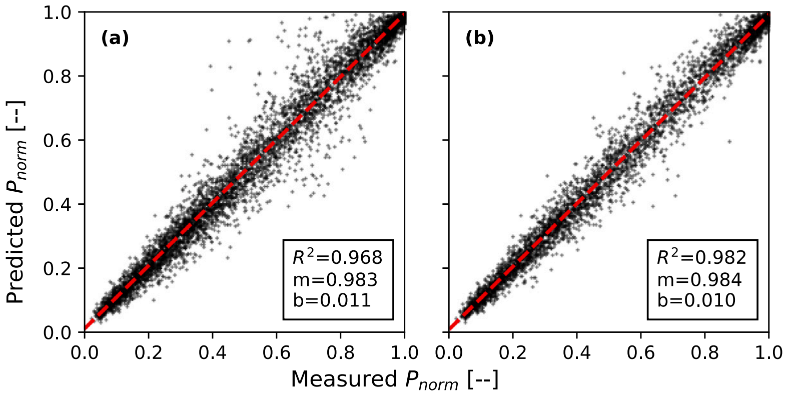

Fig. 2. Regression analysis for actual and predicted total power using data with region removal applied and (a) no filter and (b) GML filter.

are made for all the turbines at all time stamps in the test set. Summing across all time series values for actual and model-predicted total power, then calculating average yearly power production, allows the AEP to be calculated [57]. Comparing AEP allows the accuracy of farm simulations to be verified at the most coarse level. For both calculations, the appropriate region removal must be applied to the actual power data points to ensure that the model is only predicting over relevant input data. Performing this analysis, the percentage difference between the actual and predicted AEP values is 0.47%. The RMSE between the actual and predicted total power, normalized by total farm power, 2300 kW across 25 turbines producing 57.5 MW of total power, is 5.34%.

To better understand the source of errors in the farm power predictions, a regression analysis is performed and reported in Fig. 2(a). The regression scores are excellent. However, a possible contributor to errors could be points that lie far above the $y = x$ curve. Since these scattered points have higher predicted power than actual power, it is reasonable to guess that these may be points where one or more turbines were curtailed or were operating at lower-than-typical performance. Thus, de-rated conditions should be removed to better quantify the model accuracy, which is performed by applying the above-described GML filter, then repeating the total power analysis for data with region removal applied as well as ML filtration. From Fig. 2(b), the cloud of points above the $y = x$ line – supposed to be de-rated conditions – is reduced. Noting that the slopes in Fig. 2(b) are slightly improved over Fig. 2(a) and that the R^2 score improves, it can be concluded that this analysis gives a more accurate quantification of the performance of the model. The revised percent difference between AEP values is now 0.49% while the normalized RMSE is 4.07%.

Next, considering the predicted power for individual turbines across the farm gives a more granular analysis. For this analysis, the real data from every turbine is filtered and the region removal procedure is applied. For each resultant filtered set, the relevant reference conditions are provided as inputs to the wind speed, TI, and power models, chained together, to determine the predicted power. The regression results are displayed in Fig. 3(a).

To understand the impact of the occurrence of wake interactions on the model accuracy, the data considered previously are split into waked and unwaked sets, namely identifying for each turbine wind sectors for which wake interactions may occur. The IEC standard is used to define waked conditions with respect to the reference wind direction [58]. Finally, the accuracy and regression analyses are applied again, independently, to each set. The regression results for waked and unwaked wind conditions are shown in Fig. 3(b) and (c), respectively. Clearly, waked conditions are more difficult to predict than unwaked conditions, with the results in Fig. 3(a) falling somewhere in between the two. Averaging across the farm, the normalized RMSE (normalized by the standard deviation of power) is 24.28% for all conditions, 27.12% for waked conditions, and 20.04% for unwaked conditions.

6. Identification of features and physical processes from the SCADA data

Once the different data-driven modeling approaches have been defined and the models trained, and once the models have been shown to be accurate, they can be used to interpret the SCADA data. To this end, the percent differences between turbine wind speed, TI, and power, all predicted by chaining the wind speed, TI, and power models, and the reference wind speed and TI, as well as ideal power (i.e., power produced by the given turbine operating in freestream conditions), are investigated for individual turbines over varying reference wind speeds, directions, and TI values. This analysis highlights the capability of the models to describe wind farm phenomena, such as wake interactions decreasing local wind speed, increasing local TI, and decreasing power capture. More complex behaviors are also captured, such as local increases in wind speed and damping of TI connected with speedup conditions, which occur for wind sectors adjacent to those associated with wake interactions [59,60], which might be difficult to detect using statistical methods. This challenge is further investigated in Section 7.

First, the turbine wind speed model is investigated. As turbines 07 and 08 have the most interesting and complex wake interactions of the wind farm under investigation (Fig. 1), they are chosen for this analysis [25]. For each turbine, a synthetic input set is generated where the reference wind direction varies continuously within the range 0° – 360°, while the reference wind speed and TI are held constant. The constant values used for wind speed vary from 3 m s^{-1} up to 13 m s^{-1} with 5 evenly spaced steps. The TI values used are 5%, 8%, 12%, and 18%, roughly corresponding to the 10th, 25th, 50th, and 75th percentile values of reference TI, respectively. The percentage difference between the predicted local wind speed and the reference wind speed is analyzed for each synthetic data set to identify speedups or slowdowns. For each unique set of wind speed, direction, and TI values given as inputs, a bin is defined, centered on these values, with a width in wind speed of 0.5 m s^{-1}, in wind direction of 2.5°, and in TI of 2%. If the training data has fewer than 10 points in this bin, the data-driven predictions for this bin are rejected to avoid excessive extrapolation and uncertainty in predictions.

Starting from the predictions of wind speed in Fig. 4, slowdowns due to wakes for turbines 07 and 08 are evident from the regions with a positive percentage loss. Considering turbine 07 (top row in Fig. 4), sharp losses are observed at wind directions of roughly 160° and 270°, which are associated with the wakes generated by turbines 08 and 73, respectively (Fig. 1(b)). Weaker losses are observed at roughly 135° and 100° due to the larger distance from turbines 09 and 10, respectively, or due to weakly-merged wakes from the rest of the second row. The magnitude of the slowdowns due to wakes is also consistent with other field studies performed for this wind farm [25]. Indeed, for a specific wind sector and TI_∞, e.g. wind direction about 160° and TI_∞ = 5%, it is noticed that the wind speed deficit reduces with increasing

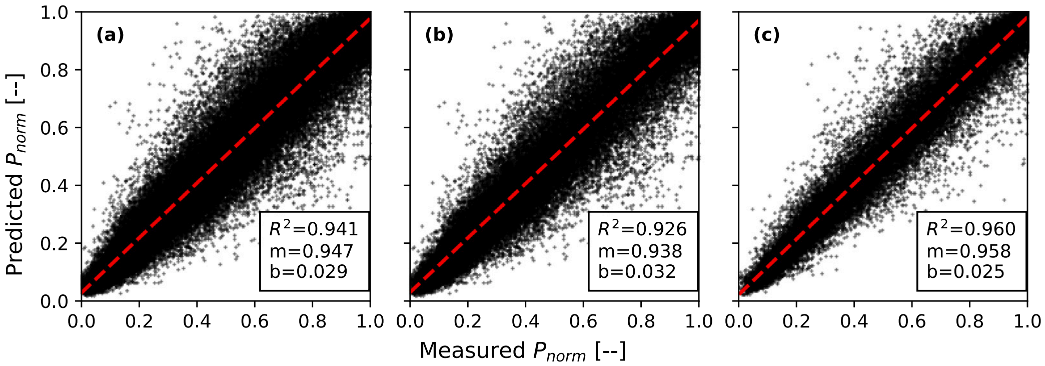

Fig. 3. Regression analysis for actual and predicted power for all individual turbines combined with region removal and GML filtering considering (a) all conditions, (b) waked conditions, and (c) unwaked conditions.

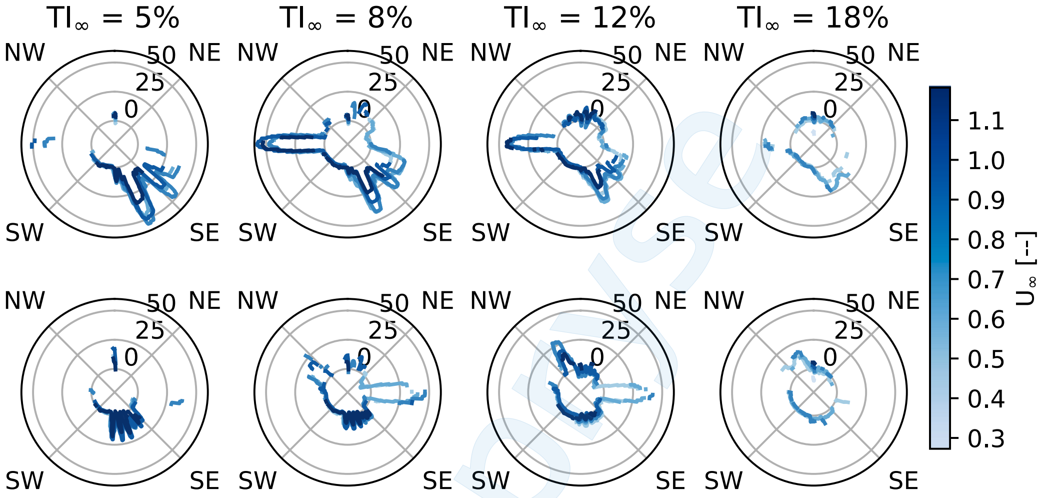

Fig. 4. Local wind speed percentage losses for turbines 07 and 08. TI increases from left to right with turbine 07 results in the top row and turbine 08 results in the bottom row.

freestream wind speed, U_∞. This is due to the gradual transition of the wind turbine operations from region two to region three of the power curve and associated reduction in the rotor thrust coefficient.

In Fig. 4, as TI increases, wakes become less prominent and the peaks for the percentage velocity deficit recede [48,61]. Finally, for some wind directions, the percent difference drops below zero, which represents a speedup that is expected to occur between wakes as mass conservation requires a region of faster-moving fluid between two regions of slower-moving fluid [11,59,60,62].

Repeating this analysis for turbine 08, the results are consistent with those of turbine 07. The percentage differences show a strong loss at 90° due to turbine 09. Weaker losses can be observed at about 315°. The weakest effects are caused by the first row (turbines 01 through 06) and can be seen for wind directions between 135° and 180°. The wakes decrease in intensity with increasing reference TI and speedups can be observed on either side of the wake centered on 90°.

To demonstrate wind speed losses at the farm level, the reference wind speed is set to be 8 m s^{-1} and the reference TI is set to the median TI, roughly 12%. The foregoing procedure is applied to each turbine to generate the percentage deficit in wind speed for the considered environmental setting. As can be noted in Fig. 5, the wakes are all visible and pointing in the expected directions from where the wakes are generated.

The speedup effect occurring for turbines of the middle row (i.e., turbines 08 through 14) is illustrated in Fig. 6 for a reference wind speed of 8 m s^{-1} and TI of 8%. The varying wind direction highlights how wakes from the northerly row and even the middle row might combine to produce speedup effects. As might be expected, turbine 08 benefits most from channeling between the middle and upper rows and has the strongest speedup effect (negative values in Fig. 6). On the other hand, turbines 13 and 14 are located on the edge of the row and experience the least benefit. It is noteworthy that these turbine array effects are impossible predict using analytical engineering wake models, and difficult to identify through statistical analysis of SCADA data due to the specific ranges in wind reference parameters to be used to avoid merging with other flow conditions. However, these phenomena can be investigated through CFD models, but only for a few wind conditions due to the larger computational costs required.

The local TI models are validated following a similar procedure as for the predictions of wind speed. In this case, the expected behavior is that TI will spike in waked regions while remaining unaffected in unwaked regions. Identical analyses are performed to the prior wind speed analyses. While results similar to Fig. 5 are not reproduced here for the sake of brevity, Fig. 7 reproduces the results of Fig. 4 when the same analysis is applied to TI.

Finally, the behavior of turbine power can be investigated by chaining the wind speed, TI, and power models. In previous analyses, local percentage variations were calculated by comparing local environmental parameters against supplied reference parameters. To obtain a reference power to compare against, the power model is used but with the reference wind speed and TI as inputs, instead of the local wind speed and TI.

The results of this analysis for turbines 07 and 08 are reported in Fig. 8 and show power losses connected with wake interactions. Wind speeds above the turbine's rated wind speed have almost no wake losses and lie in region three of the power curve. Additionally, wake losses

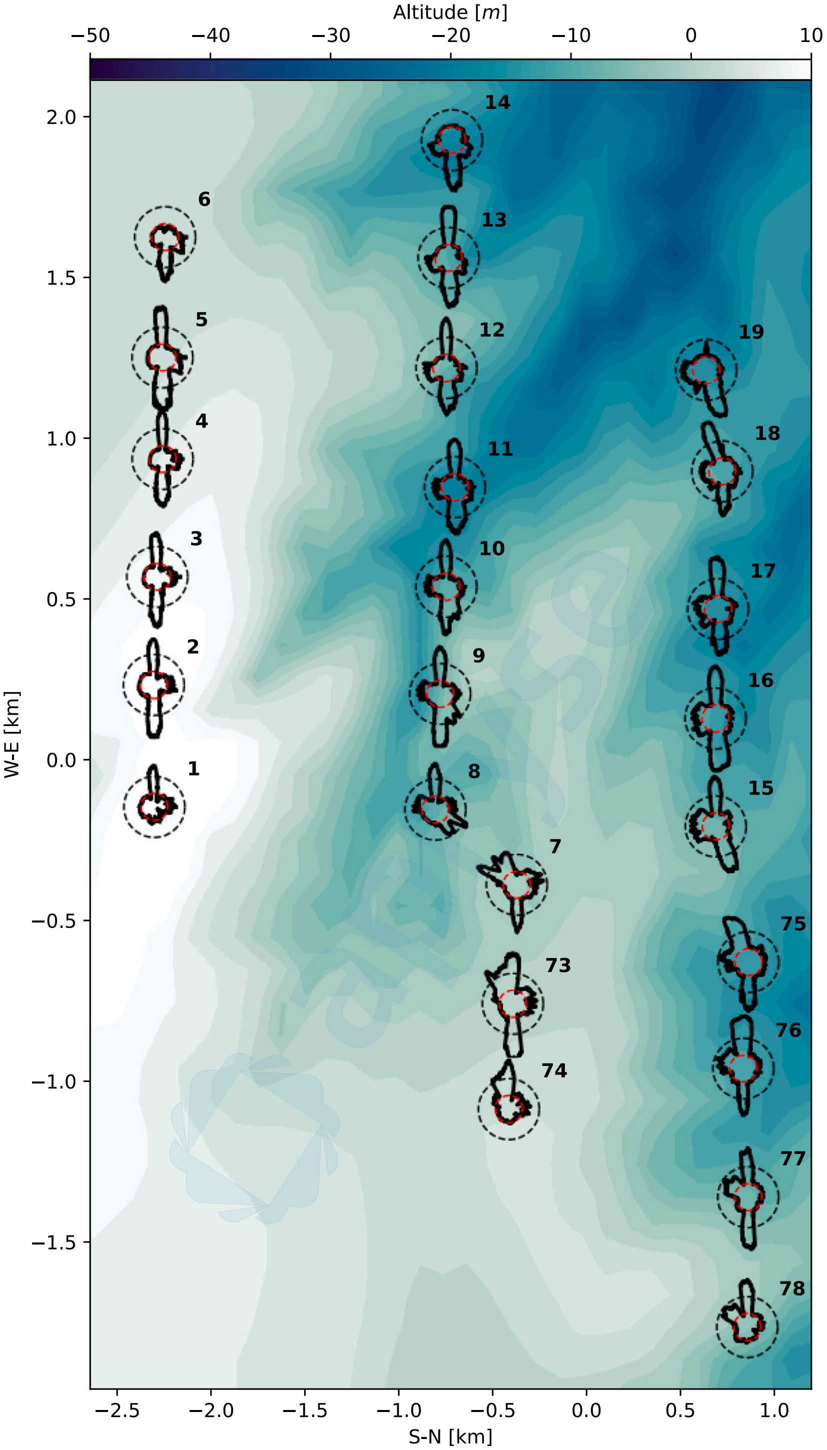

Fig. 5. Wind speed percentage losses for the entire farm for a freestream condition with U = 8 m s^{-1} and TI = 12%. Red dashed lines indicate 0% change and the dashed black line indicates 25% loss.

decrease in magnitude with increasing reference TI, as thoroughly documented in the literature [48,61,63]. The model can capture small details in the variability of power performance, such as power increases due to a combination of local wind speed and TI variation. For instance, turbine 07 typically exhibits a power boost between 45° and 90°, while turbine 08 shows stronger boosts on both sides of the wake at 90°. Interestingly, these boost regions seem to align with TI damping regions. While speedup regions are difficult to utilize to increase power production as they are small compared to wake regions, always occur next to wakes, and have much smaller magnitudes than wakes so that any positive effects are outweighed when considering long-term performance, the ability to predict boost regions is a step forward in understanding complex turbine wake interactions and improving wind farm control. Finally, considering the farm as a whole, similar results are obtained to the wind speed and TI results above, which are not presented here for the sake of brevity.

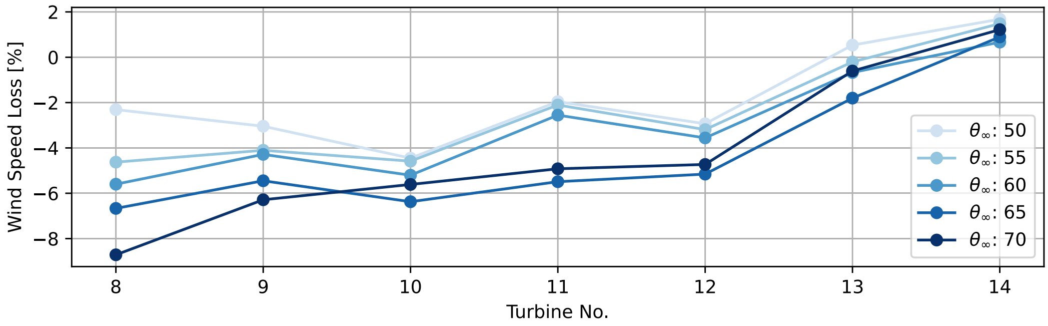

Fig. 6. Speedup effects for the middle row of turbines at a reference wind speed of 8 m s^{-1} and TI of 8%, varying wind direction.

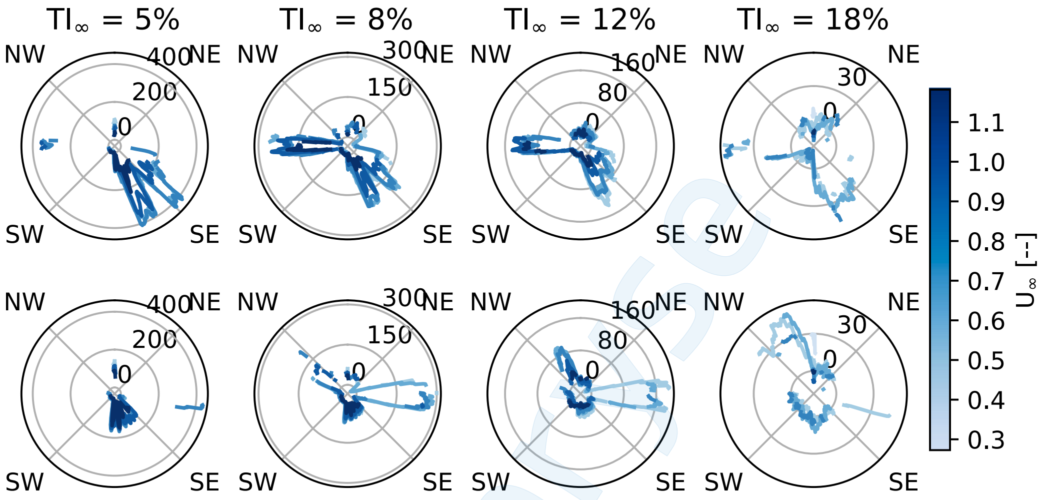

Fig. 7. Local TI percentage gain for turbines 07 and 08. Reference TI increases from left to right with turbine 07 results in the top row and turbine 08 results in the bottom row.

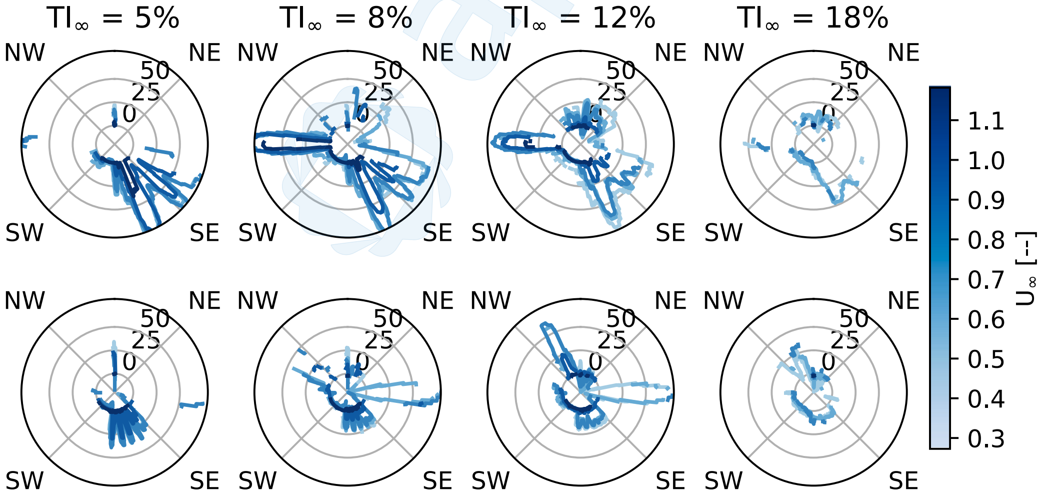

Fig. 8. Local power percentage losses for turbines 07 and 08. Reference TI increases from left to right with turbine 07 results in the top row and turbine 08 results in the bottom row.

7. Insights from data-driven analysis

Now that the data-driven method to extract information from SCADA data has been discussed, we highlight the advantages of analyzing wind turbine SCADA data through data-driven methods compared to more typical statistical techniques. While both methods offer advantages and drawbacks, they also provide different perspectives on the data, making the addition of the data-driven method to any analysis a valuable one. The main difference between the data-driven method (i.e., machine learning) and the statistical method to extract information from SCADA data is that the latter generally requires binning of the data. The bin widths must be defined before a priori and therefore impose assumptions on the physical phenomena of interest. Furthermore, defining bin width usually comprises a trade-off between large bins, which may produce results of greater statistical relevance by averaging over more points but also reduce the granularity level of the analysis; on the other hand, smaller bins may produce results of less statistical relevance by considering fewer points but also enable

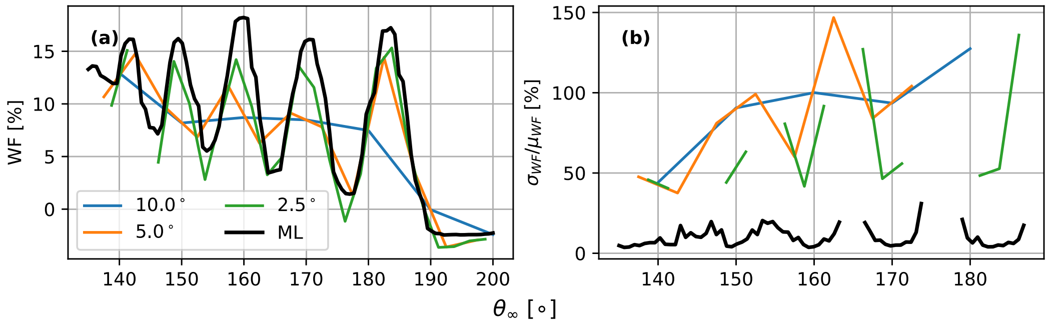

Fig. 9. A comparison of the wake losses at turbine 08 at a fixed wind speed and TI for different wind directions, comparing the data-driven result and statistical results using different bin widths. Mean results (a) and estimated statistical uncertainty by normalizing the bin standard deviations by the bin means (b) are shown.

detection of phenomena occurring over small ranges of the input parameters.

In contrast, once trained, an ML model can be interrogated over continuous mapping of the input data, eliminating the need to define bins. Data features can surface naturally without imposing any assumptions or constraints on the data analysis. The uncertainty of deterministic ML predictions, however, is often difficult to quantify, making it unreasonable for ML results to completely replace statistical results. Other ML approaches, such as probabilistic techniques, allow for better uncertainty quantification and could be investigated further.

We offer the following comparison between ML and statistical results for turbine 08, specifically focusing on the wind sector between 135° and 200°. Wind speed is kept in a 1 m/s wide bin centered on a normalized wind speed of 0.9 while TI is kept in a 5% wide bin centered on 5%. We consider the wake factor as the percent loss of wind speed at turbine 08 compared to the reference wind speed. Considering wind direction bins 10° 5° and 2.5° wide, we produce the statistical wake factor results shown in Fig. 9(a). To quantify statistical uncertainty, for each bin, we report the standard deviation of the wake factor divided by the mean of the wake factor in 9(b). Given that small wake factors contribute to unreasonably high results, approaching infinity, we mask results where the wake factor is less than 5%. On the other hand, we use the previously-trained wind speed model to predict the wind speed at turbine 08, thereby enabling the calculation of the wake factor, for continuously varying wind directions between the prescribed limits. The wind speed is fixed at a normalized value of 0.9 and the TI is fixed at 5%. These results are shown in Fig. 9(a). To estimate the uncertainty of the ML result, the model is retrained 25 times, each time sampling a random 50% of the training data. The mean of these samples defines the wake factor and the standard deviation is also calculated and reported, normalized by the mean, in Fig. 9(b).

From Fig. 9, we note first that the ML result provides the highest wake factor losses, capturing best the impacts of individual wakes from the southern row of turbines. The statistical result, on the other hand, only gets close to the ML result with bins 2.5° wide. The 5° -wide bins capture some effects of lower intensity, and the 10° -wide bins fail to capture individual wakes at all. While the bin standard deviations from the statistical analysis and the estimated ML uncertainty are certainly not equivalent, their comparison is still illuminating. In all cases, the ML uncertainty is 40% or less, while the statistical results barely get below 50% and exceed 100% at times.

To reiterate, it is not always clear how to assess ML uncertainty. We have here provided what we believe is a reasonable estimate and have found it to be far less than the statistical results. Furthermore, the ML results capture individual wakes, while the statistical results struggle to reach the same magnitudes. Thus, we determine that the ML results can at least be useful in detecting the expected magnitude and length scales of effects — in this case, the strength of the wake impact and the width of the impacted sectors. As such, the ML results provide a compelling additional perspective to traditional statistical analysis and more costly CFD analyses.

8. Data requirements to train data-driven wind farm models

While it has been shown that data-driven methods can be accurate and useful methods for extracting information from SCADA data, it is also important to discuss their limitations. Each data-driven model is constrained by the statistical significance of the data used for the model training. Bad data will generate poor models. This motivated the filtering discussion in Section 3 and Appendix. While that discussion considered outliers that did not match expected physical phenomena, this section considers what further limitations data might place on model performance, either filtered or not. As shall be shown, the variability of the output has a direct impact on model performance.

To start, we consider turbine 07. All of the SCADA data between 3 m s^{-1} and 13 m s^{-1} are taken and the chained wind speed, TI, and power models are used to predict the turbine power from reference conditions for each data point. Bins are defined as 1 m s^{-1} wide in wind speed and 5° wide in wind direction, encompassing all TI values. For each bin, the RMSE is calculated. Any bin with fewer than 10 points is rejected. The bin mean and standard deviation of the power are also calculated. These results are reported in Fig. 10.

As might be expected from the previous analysis, waked conditions are more challenging to predict accurately than unwaked conditions, such as for the wind sector 270° in Fig. 10(a). These waked conditions correspond to spikes in Fig. 10(b) which are caused by an increase in variability due to the impacting wakes. It seems reasonable, therefore, to expect a correlation between the variability of the training output and the final model accuracy. In fact, we claim that a direct relationship between output variability and model accuracy can be established in the form of a linear model, which can be used to predict model accuracy from the training data alone, with no need to train or test a model. To demonstrate this, we follow a binning analysis. Bins are now defined from 3 m s^{-1} to 13 m s^{-1} but are 0.5 m s^{-1} wide. Bins remain 5° wide in wind direction. TI bins are introduced with a width of 2.5%. Bins with fewer than 5 points are rejected. These bin sizes are chosen such that the inputs to the problem, reference wind speed, TI, and direction, do not change significantly throughout the bin. Thus, variability in the output, power, is not due to variability in the inputs. This unexplained variability will directly impact model accuracy since it cannot easily be predicted through variations in the inputs. 20 turbines are randomly selected and the chained wind speed, TI, and power models are used to predict the power for these turbines for all available data points. The bin analysis is applied and the regression between bin RMSE, normalized by bin mean power, and bin power standard deviation, also normalized by bin mean power, is shown in Fig. 11(a).

As can be seen, a clear linear relationship emerges between the normalized bin RMSE and the normalized power standard deviation. A linear model is defined following the equation given in Fig. 11. When fitting this equation, the intercept is assumed to be zero, since zero standard deviation should correspond to zero error, as only the mean needs to be predicted. This equation is then used to predict the bin-normalized RMSE using the bin-normalized power standard deviation

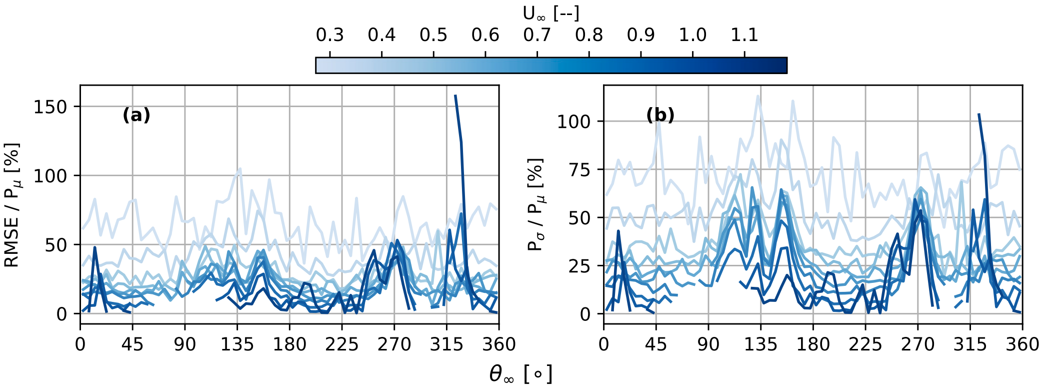

Fig. 10. Variability of the power data for turbine 07 as a function of normalized incoming wind speed, U_∞, and wind direction, θ_∞: (a) RMSE normalized by bin mean power; (b) power standard deviation normalized by bin mean power.

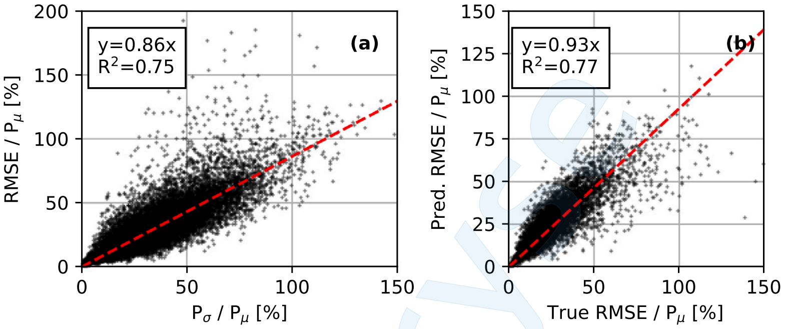

Fig. 11. Fitting a linear model on bin normalized RMSE (a) and testing the model on withheld turbines (b).

Table 4
Real and estimated normalized RMSE for each withheld turbine, where the estimated RMSE comes from a weighted average of the estimated bin normalized RMSEs.

Turbine No.	Total NRMSE	Avg. NRMSE	Est. NRMSE
2	15.1%	15.1%	15.8%
8	15.2%	15.3%	16.1%
11	16.0%	16.6%	17.1%
17	17.3%	18.4%	18.4%
77	16.9%	18.4%	18.7%

for the five withheld turbines. The regression between the true and estimated bin-normalized RMSE is shown in Fig. 11.

Using the estimated bin normalized RMSE, the total RMSE of the power predictions for each withheld turbine is estimated by averaging all the bin RMSE values, weighting each error by the percentage of points that fall in its respective bin. Table 4 reports the real and estimated total normalized RMSE for each withheld turbine. The table also reports the bin-averaged RMSE for the real data, as this may differ slightly from the RMSE computed over the entire data set and may also be closer to the estimated values.

9. Concluding remarks

As the number of installed and operating turbines proliferates, so does the amount of data gathered on the operation of these turbines. While data-driven methods have been used extensively to generate more accurate power curve models or computationally cheap surrogate models for high-fidelity wake simulations, these methods are not generally used to investigate or interpret real-world wind turbine data. Yet these methods have unique advantages in interpreting SCADA data over traditional statistical methods and therefore deserve further investigation since it is critical to understand the vast amounts of data that are now being gathered.

For this work, SCADA data from a wind farm in the Panhandle of Texas have been used to develop data-driven methods. Different filtering approaches have been discussed with a data-driven filtering approach being selected to filter the data and ensure a high quality of training data for future models. Region removal was also used to focus the models on interesting phenomena in region two of the power curve, where wake interactions are the strongest.

Once the data are prepared, data-driven models have been developed to describe individual turbine performance as a function of reference conditions, which have provided a reasonable estimate of freestream wind conditions. This aim has been achieved by chaining together three models: a wind speed model, a TI model, and a power model. The wind speed model predicts a turbine's local wind speed as a function of the reference wind speed, direction, and TI, while the TI model predicts a turbine's local TI as a function of its local wind speed and the same reference inputs. Finally, the power model predicts the turbine's power as a function of its local wind speed and TI, functioning as a multi-dimensional power curve tuned to each individual turbine. Thus, the interactions of the turbines could be described in terms of impacts on wind speed, TI, and power. While the models used are standard models and other works have considered similar approaches to predicting turbine performance, chaining these models together provides a unique framework to investigate physical phenomena efficiently across large data sets.

Once the models have been developed and shown to be accurate, they have been used to interpret the SCADA data by making predictions for user-provided wind speeds, directions, and TIs. Importantly, wake effects have been well described, and complex effects, such as speedups and TI damping, have also been identified from the data. The models have been used to assess the wake impacts across the farm and the varying levels of speedups at different turbines. While this might be challenging from a statistical perspective since strict limits would have to be applied to keep all turbines operating within nominal bounds, the data-driven models have not suffered these restrictions.

The specific advantages and disadvantages of the data-driven approach have been discussed against the backdrop of traditional statistical analysis. The data-driven approaches were shown to provide a useful additional perspective to statistical analysis that could better replicate results obtained via simulation of the wind farm but with the caveat that the statistical uncertainty of data-driven results can be difficult to quantify.

While data-driven models are useful, they also suffer limits imposed by the quality of the training data. We have considered these and showed that the model performance has been closely related to the variability of the training data. Model performance has been modeled as a linear function of the variability of the model output. These linear models have been very accurate in estimating the error of a data-driven model trained on the given data without ever needing to train or assess these data-driven models. Thus, data sets can be assessed for data-driven interpretations or experimental campaigns designed to ensure high-quality data sets by using these linear models.

In summary, while there are limitations that have been described and need to be carefully considered, the use of data-driven models to interpret SCADA data is very promising. Since interpreting the vast amounts of collected data is essential to promoting a better understanding of wind turbine performance and therefore better turbine and farm design, control, and monitoring, this work has provided an outline that can guide data-driven interpretations. Further applications could include monitoring of operating farms to identify poor performance or under-performing turbines, or greater studies of speedup effects in different layout geometries, or the relationship between incoming flow conditions and wake overlapping.

CRediT authorship contribution statement

Coleman Moss: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation. **Romit Maulik:** Writing – review & editing, Supervision, Methodology, Investigation, Formal analysis. **Giacomo Valerio Iungo:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Giacomo Valerio Iungo reports financial support was provided by National Science Foundation. Giacomo Valerio Iungo reports a relationship with National Science Foundation that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) CBET, Fluid Dynamics CAREER program, China, award no. 2046160, program manager Ron Joslin.

Appendix. Filter definition and selection

Several filtering approaches are considered for the quality control and pre-processing of the SCADA data before training models to predict power. As noted, data filtering is only applied to data used to train power models and not to data used to train wind speed or TI models. Two statistical filtering approaches are considered: a simple binning filter and a K-means clustering filter based on the Mahalanobis distance [17]. Further, an iterative Gaussian Process (GP) filter and a novel machine learning filter are also considered [15]. Our goal is to demonstrate important considerations in selecting a filter, especially for use with data-driven models, and demonstrate the effectiveness of using data-driven filters for data-driven models.

A.1. Existing filters

First, the binning filter is considered. This approach splits the power curve into 0.5 m s^{-1} wide bins in wind speed [58]. The mean power is calculated for each bin, as well as the standard deviation in power. Upper and lower limit curves are created by adding and subtracting 2 standard deviations from the mean of each bin. For a given wind speed, the upper and lower limits on power are determined by interpolating these curves. Any data point with a power value falling outside of these limits is considered an outlier. When applying this filter initially, outliers with wind speeds around and above the rated wind speed of the turbine tend to be harder to remove since the standard deviation increases in that region, as seen in Fig. A.12(a). For this reason, the filter is applied a second time to further prune outliers in the transition region between regions two and three. As seen in Fig. A.12(b), the second application is quite effective.

Second, the K-means clustering filter uses the K-means algorithm to cluster data. For this work, 10 clusters are generated, as in the original paper [17], then the Mahalanobis distance from the respective cluster center is calculated for all points in each cluster. The Mahalanobis distance gives the distance between an observation vector of several dimensions, $\mathbf{x}$, and the mean vector of all the observations being considered, $\boldsymbol{\mu}$, scaled using the covariance matrix Σ:

$$M(\mathbf{x}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})}. \tag{A.1}$$

The covariance matrix, Σ, is calculated uniquely for each cluster. The considered SCADA parameters are wind speed, TI, and power. Once a Mahalanobis distance has been assigned to every data point, the mean and standard deviation of the Mahalanobis distance for each cluster is calculated. Points further than 2 standard deviations from the mean value in each bin are rejected as in the original work. Fig. A.13 illustrates the application of this filter to the SCADA data from turbine 08. Since TI is considered in addition to wind speed and power, this filter can detect potential outliers on the main body of the power curve rather than just the fringes, as in the case of the binning filter. It is also seen that the main effect of the K-Means clustering is to roughly apply bins in power.

The third filter considered is the automatic Gaussian Process (GP) filter [15], for which a single GP regression model is fit to the SCADA data using wind speed as the input and power as the output, with a different model being trained for every turbine. Since the GP model reports a standard deviation value for predicted points, upper and lower bounds on the power curve can be applied by adding 4 times the standard deviation curves to the GP-generated power curve, as prescribed in the original paper [15]. Points beyond these bounds are rejected and the model is re-trained on the remaining points. This procedure repeats until no more points are rejected. Fig. A.14 demonstrates the application of this filter to turbine 08.

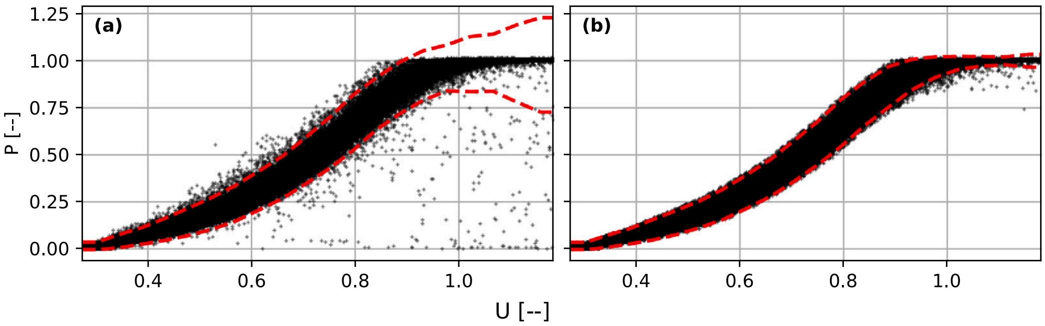

Fig. A.12. Iterative application of the binning filter to turbine 08 with the filter limits shown in red dashed lines: (a) first pass and (b) second pass.

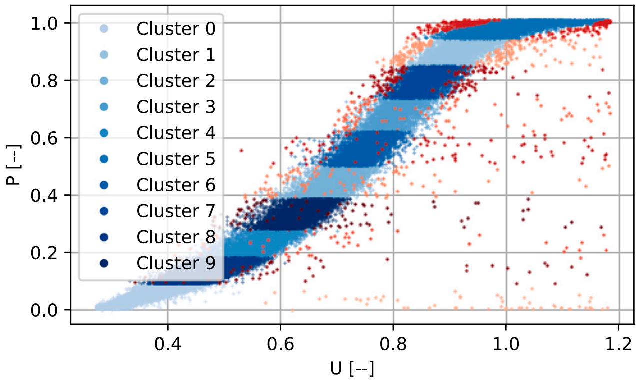

Fig. A.13. Application of the K-Means filter to turbine 08.

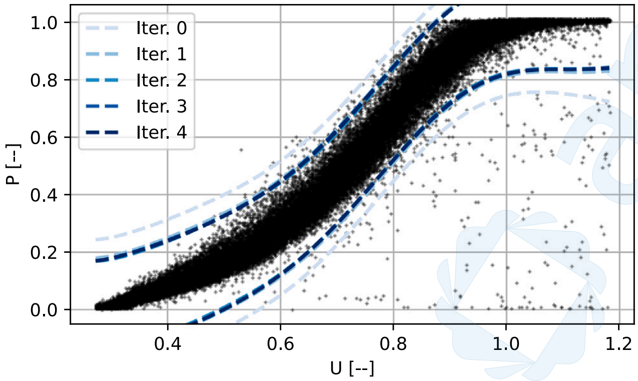

Fig. A.14. Application of the iterative GP filter to turbine 08.

A.2. Proposed filter

The final filter considered is an automatic data-driven filter, which operates on a similar principle to the GP filter and is referred to as the General Machine Learning (GML) filter. The main assumption is that in a given data set to be filtered, there are sufficiently more inliers than outliers, such that a data-driven model can learn the main features of the data set from the inliers. Assuming the physical features of interest are captured by the data-driven model, data points that cannot be predicted with sufficient accuracy likely do not belong to the physical features of interest and can be rejected. Applying this approach to filter SCADA data, a data-driven model should be selected such that it can accurately predict turbine power given wind speed and TI inputs. It is then trained over the entire data set to be filtered.

Power is predicted using this model for all points in the data set and each point is assigned the absolute difference between real and predicted power as an error. Different regions of the power curve may be more or less challenging for the model to predict and therefore may have different typical errors. For this reason, the data set is then binned in wind speed bins 0.5 m s^{-1} wide, and the average and standard deviation of the error is computed for each bin. This forms an upper limit on allowable error. Interpolating wind speed values on this curve, any points with errors greater than the limit are rejected. The model is then retrained on the remaining points and the procedure is repeated until either a threshold in iterations is reached or a threshold in the number of rejected points.

For this filter to function, accurate data-driven models are needed that can effectively capture important physical phenomena. We compare the performance of the RF, ET, GB, HGB, XGB, and NN models. Each model has certain high-level settings, called hyperparameters, which greatly impact its performance. Optimizing these hyperparameters is a difficult task usually requiring expertise in machine learning. To keep this work user-focused, we opt to use the Python package DeepHyper to automatically optimize the hyperparameters of each model [64]. While there is no reason to expect the optimal hyperparameters for power prediction to be consistent across all turbines, there is also no reason to expect the problem to vary massively from one turbine to the next, therefore variation in optimal settings should be small. For this reason, the different models will be optimized to predict turbine 06 power.

For each model, only the most important hyperparameters are optimized. DeepHyper is used to minimize the mean square error of the model. Before starting the optimization, a stratified split is used to create a training data set with 70% of the available data. Another split is used to designate 30% of the data as testing data. In each iteration of DeepHyper, the model being optimized is trained on the training data and then used to predict over the testing data, generating a mean square error. This is performed for turbine 06 alone. After optimizing the hyperparameters of all the models on turbine 06, the data from the remaining turbines is also split into training and testing sets following the 70%/30% split. For each turbine, the optimized models are applied and regression scores are determined. Thus, the generalizability of the optimized hyperparameters is measured.

Starting with the RF model, the two hyperparameters to be optimized are the maximum depth, which varies from 2 to 200, and the number of estimators, which vary from 1 to 2000. Optimization on the ET model uses identical hyperparameters and bounds to the RF optimization. The three important hyperparameters to consider to optimize the GB model are learning rate, varying from 0.0001 to 1, and number of estimators and maximum depth, which are identical to the previous models. To optimize the HGB model, it is necessary to optimize the learning rate, maximum bins, and maximum iterations. The learning rate is identical to the previous models. The maximum bins parameter is an integer between 1 and 255 and the maximum iterations parameter is an integer between 10 and 2000. To optimize the XGB model, the maximum depth, number of estimators, learning rate, and subsampling fraction must be set. The number of estimators and learning rate are handled as before. The maximum depth is an integer allowed to vary

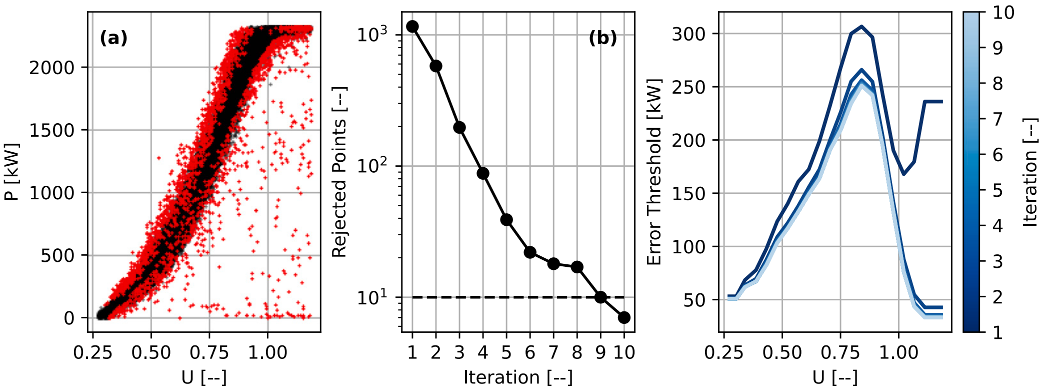

Fig. A.15. Application of GML filter to turbine 08 showing (a) the rejected points in red, (b) the number of rejected points per iteration with the minimum threshold of ten points, and (c) the error threshold curve as a function of wind speed per iteration.

Table A.5
The statistics of Normalized RMSE of filtering models calculated across all turbines. The minimum value in each column is shown in bold.

Model	Min.	25%-ile	Median	75th %-ile	Max.
RF	14.25	15.61	16.20	16.96	19.94
ET	14.15	15.67	16.26	17.03	19.93
GB	14.16	15.65	16.22	16.92	19.89
HBG	14.11	15.61	16.22	16.85	19.86
XGB	**14.05**	**15.54**	**16.12**	**16.81**	19.83
NN	14.62	15.65	16.35	17.32	**18.26**

between 2 and 50. The sub-sample fraction is allowed to vary between 0.1 and 1. Finally, the important hyperparameters for the NN are the number of hidden layers and the number of neurons for each hidden layer. The layers are set to have identical numbers of neurons across all layers and are fully connected Dense layers. The activation function of all layers is set to exponential linear unit except for the input and output layers. The input layer always has an equal number of neurons to the number of inputs, in this case, 2, and uses the rectified linear unit function. The output layer has a single neuron and also uses the rectified linear unit activation function. The number of hidden layers is allowed to vary between 1 and 20 and the number of hidden neurons is allowed to vary between 10 and 1000. For each unique model, DeepHyper is run for 1000 iterations. The resultant hyperparameters of each model determined from the DeepHyper analysis are listed below and Table A.5 reports the regression metrics across the farm. Of course, these hyperparameters may not be absolutely optimal, and rerunning DeepHyper would likely result in slightly different hyperparameters. In any case, the selected hyperparameters provide a reasonable increase in model accuracy without the need for detailed manual tuning or costly grid searching. From Table A.5, it is evident the XGB model performs best. It is selected for further use in the GML filter.

- RF: maximum depth of 8, number of estimators of 150
- ET: maximum depth of 12, number of estimators of 65
- GB: learning rate of 0.032, maximum depth of 5, number of estimators of 130
- HGB: learning rate of 0.006, maximum bins of 235, maximum iterations of 825
- XGB: learning rate of 0.012, maximum depth of 4, number of estimators of 510, subsampling fraction of 0.89

Now that the optimal data-driven model for power prediction for use in the SCADA filter has been determined, the GML filter can be applied. Fig. A.15 shows the application of the filter to turbine 08.

A.3. Filter selection

Now that all the filters have been defined and introduced, how is the best filter to be determined? Qualitatively, the filtered power curves could be compared. Filter rejection rates are also important to consider, as overly-aggressive filters may not leave sufficient data behind to train data-driven models. Yet merely comparing filter rejection rates does not give a good indication of whether the filter is rejecting outliers. Of course, since the purpose of filtering is to determine which points are inliers and which points are outliers, it is impossible to define the accuracy of a filter, since that would presuppose inliers and outliers are already known. Since the purpose of the filters is to train accurate data-driven models, two metrics are defined with respect to that purpose. First, the effect of filtering on model uncertainty is observed. Then, the effect of filtering on model accuracy is determined.

Under the assumption that data-driven models can capture the physical phenomena represented by inliers but with diluted accuracy in the presence of outliers, the presence of outliers should add some variability to model predictions. This can be measured by training a data-driven model on random selections from a larger training set and then using the trained model to predict outputs for a constant testing set. Since the test set is constant, variability in the predictions can be attributed to variability in the input data, especially variability that does not belong to any physical phenomena, i.e., outliers.

To demonstrate this effect, turbine 08 data is randomly sampled 10 times keeping 50% of the data each time to create 10 subgroups. A synthetic data set is defined with linearly increasing values in wind speed and a constant TI set to the median of the full turbine 08 data set. For each of the 10 subgroups, a new XGB model is trained with the previously identified hyperparameters. The model is used to predict power output for the synthetic data set. A standard deviation in predictions across the 10 subgroups can therefore be defined. An example of this analysis is demonstrated in Fig. A.16, which considers the binning filter applied to turbine 08 and shows both the standard deviation curve and the curve of standard deviation normalized by mean power.

To quantify the impact of filtering on model variability, the difference in the normalized standard deviation curves before and after filtering is integrated over the wind speed values as is the original unfiltered normalized standard deviation curve. The ratio of the two represents the percent improvement caused by filtering. Fig. A.16 shows an improvement of 21%. This procedure is applied to all the turbines using all the filters and the statistics of the resultant percent improvements are reported in Table A.6. From the table, it can be seen that, in some cases, filtering actually increases model variability. In these cases, the assumptions made by each filter do not match the physical phenomena and poor filtering occurs. Comparing the scores, it seems like the GML filter is promising. From the 25th percentile and above, the binning and GML filters have very similar scores. The K-means and GP filters have better maximum improvements but fall behind all other cases. However, all filters except for the GML filter have very poor minimum performance with all cases falling below a

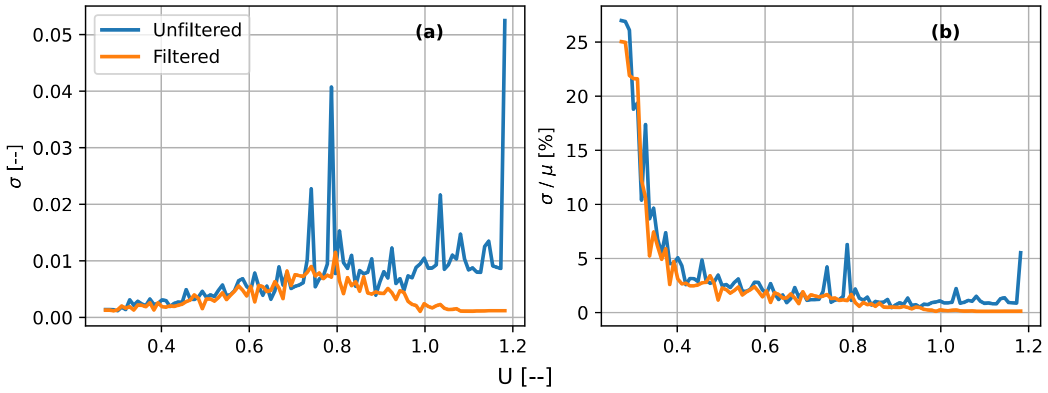

Fig. A.16. The effect of the binning filter on data-driven model variability is shown in the (a) standard deviation of power predictions and (b) standard deviation of power predictions normalized by the mean powers.

Table A.6
Percent improvement in model variability caused by filtering. The maximum value in each column is shown in bold.

Filter	Min.	25th %-ile	Median	75th %-ile	Max.
Binning	−37	8.9	**24**	**29**	35
K-Means	−24	−8.6	14	21	38
GP	−22	−1.0	6.7	19	**42**
GML	**−1.0**	**12**	19	**29**	35

Table A.7
Percent improvement in model MAE and MRAE (reported as MAE/MRAE) caused filtering. The maximum values in each column are displayed in bold.

Filter	Min.	25th %-ile	Median	75th %-ile	Max.
Binning	0.0/−0.1	**0.3/0.1**	**1.4/0.7**	**2.8/1.5**	6.7/4.4
K-Means	−0.1/−0.1	0.2/**0.1**	0.4/0.2	1.2/0.7	8.3/5.2
GP	**0.0/0.0**	0.0/0.0	0.0/0.0	0.0/0.0	2.4/1.2
GML	**0.0/0.0**	0.2/0.0	1.1/0.5	2.6/1.2	**10/6.4**

20% decrease. Thus, while the GML filter does not achieve the same maximum as the K-means and GP filters, those maximum scores are unlikely, and it avoids the extreme minima of these filters, as well as the extreme minima of the binning filter.

Next, the impact of filtering on model accuracy is considered. This can be determined by splitting the data, before removing outliers, into a training set with 80% of the data and a testing set with 20% of the data. From the training set, which is considered raw because it includes outliers, is defined a filtered training set, derived by rejecting the outliers. The raw training set is then downsampled to have the same number of points as the filtered training set. The previously optimized XGB model is used to train two models, one trained on raw data, and one trained on filtered data. Both models are then used to make predictions over the test set and the accuracy is compared. Of course, the outliers in the testing set will dilute any improvements in model accuracy. By leaving the testing set unfiltered, however, no biases are introduced at the testing stage. When comparing the accuracy, one significant detail is that mean absolute error (MAE) is used instead of RMSE, and a new metric is introduced, mean root absolute error (MRAE), defined in Eq. (A.2).

$$MRAE = \frac{1}{N}\sum_{i=1}^{N}\left(\sqrt{|P_i - \hat{P}_i|}\right), \tag{A.2}$$

where P_i is the true power produced at the ith timestep and $\hat{P}_i$ is the predicted power for the same time stamp. MAE and MRAE are used because RMSE penalizes large errors more severely than small errors.

Training a model on filtered data is expected to make the model more accurate at predicting inliers and less accurate at predicting outliers. If the model, when trained on raw data, generally tends to reproduce the mean behavior of the data set, then the improvement from filtering will be to shift the predictions closer to the inlier behavior, which will cause the outliers to be predicted with lower accuracy. Since the outliers already lie far from the mean, the increased error of the outliers will be penalized heavily and increase the RMSE while the improved accuracy of the inliers will not have a large impact on the RMSE. The result is that filtering increases the RMSE. The MAE, on the other hand, penalizes all errors equally. The MRAE enhances the penalization of small errors. Thus, these two metrics should better capture the impact of filtering on inlier prediction without being diluted as severely as the RMSE.

The above procedure is applied to all the turbines and the statistics of the results are listed in Table A.7. Some of the percent improvement scores are negative, indicating worse performance following filtering. While this may not indicate that filtering is decreasing the data quality, it at least indicates that the benefits gained in accuracy for inliers are outweighed by negative impacts on predicting outliers. The result should be considered inconclusive, as the outlier errors are obscuring any gains in inlier performance. The GP and GML filters are interesting in this respect, as they have no negative impacts. Overall, the binning and GML filters perform best, with the binning filter performing slightly better.

With these considerations in mind, it seems that either the binning filter or the GML filter would make an appropriate filter for data-driven modeling. Given that the improvements in accuracy as similar, with the binning filter holding only a slight edge, while the improvements in variability are also similar, but with the GML filter taking a more meaningful lead here by avoiding extremely poor performance, the GML filter using an XGB model is selected for use in this work, though both the binning and GML filters could be used to good effect.

References

[1] Kaldellis JK, Zafirakis D. The wind energy (r)evolution: A short review of a long history. Renew Energy 2011;36(7):1887–901, doi:gonz.

[2] Gielen D, Boshell F, Saygin D, Bazilian MD, Wagner N, Gorini R. The role of renewable energy in the global energy transformation. Energy Strategy Rev 2019;24:38–50. http://dx.doi.org/10.1016/j.esr.2019.01.006.

[3] Liu Y, Gao X, Yan J, Han S, Infield DG. Clustering methods of wind turbines and its application in short-term wind power forecasts. J Renew Sustain Energy 2014;6(5):053119. http://dx.doi.org/10.1063/1.4898361.

[4] Ezzat AA. Turbine-specific short-term wind speed forecasting considering within-farm wind field dependencies and fluctuations. Appl Energy 2020;269:115034. http://dx.doi.org/10.1016/j.apenergy.2020.115034.

[5] He R, Yang H, Sun S, Lu L, Sun H, Gao X. A machine learning-based fatigue loads and power prediction method for wind turbines under yaw control. Appl Energy 2022;326:120013. http://dx.doi.org/10.1016/j.apenergy.2022.120013.

[6] Wang Z, Zeng T, Chu X, Xue D. Multi-objective deep reinforcement learning for optimal design of wind turbine blade. Renew Energy 2023;203:854–69. http://dx.doi.org/10.1016/j.renene.2023.01.003.

[7] Wang Y, Hu Q, Li L, Foley AM, Srinivasan D. Approaches to wind power curve modeling: A review and discussion. Renew Sustain Energy Rev 2019;116:109422, doi:lydia.

[8] Ti Z, Deng XW, Yang H. Wake modeling of wind turbines using machine learning. Appl Energy 2020;257:114025. http://dx.doi.org/10.1016/j.apenergy.2019.114025.

[9] Ti Z, Deng XW, Zhang M. Artificial neural networks based wake model for power prediction of wind farm. Renew Energy 2021;172:618–31. http://dx.doi.org/10.1016/j.renene.2021.03.030.

[10] Wang L, Chen M, Luo Z, Zhang B, Xu J, Wang Z, Tan ACC. Dynamic wake field reconstruction of wind turbine through physics-informed neural network and sparse LiDAR data. Energy 2024;291:130401. http://dx.doi.org/10.1016/j.energy.2024.130401.

[11] Moss C, Maulik R, Moriarty P, Iungo GV. Predicting wind farm operations with machine learning and the P2D-RANS model: A case study for an AWAKEN site. Wind Energy 2023;n/a(n/a). http://dx.doi.org/10.1002/we.2874.

[12] Ashwin Renganathan S, Maulik R, Letizia S, Iungo GV. Data-driven wind turbine wake modeling via probabilistic machine learning. Neural Comput Appl 2022;34(8):6171–86. http://dx.doi.org/10.1007/s00521-021-06799-6.

[13] Moss C, Maulik R, Iungo GV. A call for enhanced data-driven insights into wind energy flow physics. Theor Appl Mech Lett 2024;14(1):100488. http://dx.doi.org/10.1016/j.taml.2023.100488.

[14] Pandit RK, Infield D, Carroll J. Incorporating air density into a Gaussian process wind turbine power curve model for improving fitting accuracy. Wind Energy 2019;22(2):302–15. http://dx.doi.org/10.1002/we.2285.

[15] Manobel B, Sehnke F, Lazzús JA, Salfate I, Felder M, Montecinos S. Wind turbine power curve modeling based on Gaussian processes and artificial neural networks. Renew Energy 2018;125:1015–20. http://dx.doi.org/10.1016/j.renene.2018.02.081.

[16] Pei S, Li Y. Wind turbine power curve modeling with a hybrid machine learning technique. Appl Sci 2019;9(22):4930. http://dx.doi.org/10.3390/app9224930.

[17] Yesilbudak M. Implementation of novel hybrid approaches for power curve modeling of wind turbines. Energy Convers Manage 2018;171:156–69. http://dx.doi.org/10.1016/j.enconman.2018.05.092.

[18] Zhou J, Guo P, Wang X-R. Modeling of wind turbine power curve based on Gaussian process. In: 2014 international conference on machine learning and cybernetics. vol. 1, 2014, p. 71–6. http://dx.doi.org/10.1109/ICMLC.2014.7009094.

[19] Burton T, Jenkins N, Sharpe D, Bossanyi E. Wind energy handbook. 1st ed.. Wiley; 2011, http://dx.doi.org/10.1002/9781119992714.

[20] Sanchez Gomez M, Lundquist JK. The effect of wind direction shear on turbine performance in a wind farm in central Iowa. Wind Energy Sci 2020;5(1):125–39. http://dx.doi.org/10.5194/wes-5-125-2020.

[21] Panthi K, Iungo GV. Quantification of wind turbine energy loss due to leading-edge erosion through infrared-camera imaging, numerical simulations, and assessment against SCADA and meteorological data. Wind Energy 2023;26(3):266–82. http://dx.doi.org/10.1002/we.2798.

[22] Njiri JG, Söffker D. State-of-the-art in wind turbine control: trends and challenges. Renew Sustain Energy Rev 2016;60:377–93. http://dx.doi.org/10.1016/j.rser.2016.01.110.

[23] Barthelmie RJ, Hansen K, Frandsen ST, Rathmann O, Schepers JG, Schlez W, Phillips J, Rados K, Zervos A, Politis ES, Chaviaropoulos PK. Modelling and measuring flow and wind turbine wakes in large wind farms offshore. Wind Energy 2009;12(5):431–44. http://dx.doi.org/10.1002/we.348.

[24] Bardal LM, Sætran LR. Influence of turbulence intensity on wind turbine power curves. In: 14th deep sea offshore wind r&d conference, eERA deepWind'2017, Energy Procedia In: 14th deep sea offshore wind r&d conference, eERA deepWind'2017, 2017;137:553–8.http://dx.doi.org/10.1016/j.egypro.2017.10.384,

[25] El-Asha S, Zhan L, Iungo GV. Quantification of power losses due to wind turbine wake interactions through SCADA, meteorological and wind LiDAR data. Wind Energy 2017;20(11):1823–39. http://dx.doi.org/10.1002/we.2123.

[26] Optis M, Perr-Sauer J. The importance of atmospheric turbulence and stability in machine-learning models of wind farm power production. Renew Sustain Energy Rev 2019;112:27–41. http://dx.doi.org/10.1016/j.rser.2019.05.031.

[27] Lydia M, Selvakumar AI, Kumar SS, Kumar GEP. Advanced algorithms for wind turbine power curve modeling. IEEE Trans Sustain Energy 2013;4(3):827–35, doi:zhou.

[28] Bastankhah M, Porté-Agel F. A new analytical model for wind-turbine wakes. In: Special issue on aerodynamics of offshore wind energy systems and wakes, Renew Energy In: Special issue on aerodynamics of offshore wind energy systems and wakes, 2014;70:116–23.http://dx.doi.org/10.1016/j.renene.2014.01.002,

[29] Abkar M, Porté-Agel F. Influence of atmospheric stability on wind-turbine wakes: A large-Eddy simulation study. Phys Fluids 2015;27(3):035104. http://dx.doi.org/10.1063/1.4913695.

[30] Letizia S, Iungo GV. Pseudo-2D RANS: a LiDAR-driven mid-fidelity model for simulations of wind farm flows. J Renew Sustain Energy 2022;14(2):023301. http://dx.doi.org/10.1063/5.0076739.

[31] Iungo GV, Santhanagopalan V, Ciri U, Viola F, Zhan L, Rotea MA, Leonardi S. Parabolic RANS solver for low-computational-cost simulations of wind turbine wakes. Wind Energy 2018;21(3):184–97. http://dx.doi.org/10.1002/we.2154.

[32] Kaldellis JK, Triantafyllou P, Stinis P. Critical evaluation of wind turbines' analytical wake models. Renew Sustain Energy Rev 2021;144:110991. http://dx.doi.org/10.1016/j.rser.2021.110991.

[33] St. Martin CM, Lundquist JK, Clifton A, Poulos GS, Schreck SJ. Wind turbine power production and annual energy production depend on atmospheric stability and turbulence. Wind Energy Sci 2016;1(2):221–36. http://dx.doi.org/10.5194/wes-1-221-2016.

[34] Sun H, Gao X, Yang H. A review of full-scale wind-field measurements of the wind-turbine wake effect and a measurement of the wake-interaction effect. Renew Sustain Energy Rev 2020;132:110042. http://dx.doi.org/10.1016/j.rser.2020.110042.

[35] Hegazy A, Blondel F, Cathelain M, Aubrun S. LiDAR and SCADA data processing for interacting wind turbine wakes with comparison to analytical wake models. Renew Energy 2022;181:457–71. http://dx.doi.org/10.1016/j.renene.2021.09.019.

[36] Chen K, Lin J, Qiu Y, Liu F, Song Y. Deep learning-aided model predictive control of wind farms for AGC considering the dynamic wake effect. Control Eng Pract 2021;116:104925. http://dx.doi.org/10.1016/j.conengprac.2021.104925.

[37] Li R, Zhang J, Zhao X. Dynamic wind farm wake modeling based on a bilateral convolutional neural network and high-fidelity LES data. Energy 2022;258:124845. http://dx.doi.org/10.1016/j.energy.2022.124845.

[38] Zhang J, Zhao X. Wind farm wake modeling based on deep convolutional conditional generative adversarial network. Energy 2022;238:121747. http://dx.doi.org/10.1016/j.energy.2021.121747.

[39] Park J, Park J. Physics-induced graph neural network: An application to wind-farm power estimation. Energy 2019;187:115883. http://dx.doi.org/10.1016/j.energy.2019.115883.

[40] Bentsen LØ, Dilp Warakagoda N, Stenbro R, Engelstad P. Wind park power prediction: Attention-based graph networks and deep learning to capture wake losses. J Phys: Conf Ser 2022;2265(2):022035. http://dx.doi.org/10.1088/1742-6596/2265/2/022035.

[41] Zhou H, Qiu Y, Feng Y, Liu J. Power prediction of wind turbine in the wake using hybrid physical process and machine learning models. Renew Energy 2022;198:568–86. http://dx.doi.org/10.1016/j.renene.2022.08.004.

[42] Li S, Zhang M, Piggott MD. End-to-end wind turbine wake modelling with deep graph representation learning. Appl Energy 2023;339:120928. http://dx.doi.org/10.1016/j.apenergy.2023.120928.

[43] Zhang J, Zhao X. Digital twin of wind farms via physics-informed deep learning. Energy Convers Manage 2023;293:117507. http://dx.doi.org/10.1016/j.enconman.2023.117507.

[44] Gajendran MK, Kabir IFSA, Vadivelu S, Ng EYK. Machine learning-based approach to wind turbine wake prediction under Yawed conditions. J Mar Sci Eng 2023;11(11):2111. http://dx.doi.org/10.3390/jmse11112111.

[45] Li B, Ge M, Li X, Liu Y. A physics-guided machine learning framework for real-time dynamic wake prediction of wind turbines. Phys Fluids 2024;36(3):035143. http://dx.doi.org/10.1063/5.0194764.

[46] Sun H, Qiu C, Lu L, Gao X, Chen J, Yang H. Wind turbine power modelling and optimization using artificial neural network with wind field experimental data. Appl Energy 2020;280:115880. http://dx.doi.org/10.1016/j.apenergy.2020.115880.

[47] Hammer F, Helbig N, Losinger T, Barber S. Graph machine learning for predicting wake interaction losses based on SCADA data. J Phys: Conf Ser 2023;2505(1):012047. http://dx.doi.org/10.1088/1742-6596/2505/1/012047.

[48] Zhan L, Letizia S, Valerio Iungo G. LiDAR measurements for an onshore wind farm: Wake variability for different incoming wind speeds and atmospheric stability regimes. Wind Energy 2020;23(3):501–27. http://dx.doi.org/10.1002/we.2430.

[49] Zhan L, Letizia S, Iungo GV. Optimal tuning of engineering wake models through lidar measurements. Wind Energy Sci 2020;5(4):1601–22. http://dx.doi.org/10.5194/wes-5-1601-2020.

[50] Sebastiani A, Castellani F, Crasto G, Segalini A. Data analysis and simulation of the Lillgrund wind farm. Wind Energy 2021;24(6):634–48. http://dx.doi.org/10.1002/we.2594.

[51] Howland MF, Dabiri JO. Wind farm modeling with interpretable physics-informed machine learning. Energies 2019;12(14):2716. http://dx.doi.org/10.3390/en12142716.

[52] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D. Scikit-learn: Machine learning in python. Mach Learn Python 2011;6.

[53] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. KDD '16, New York, NY, USA: ACM; 2016, p. 785–94. http://dx.doi.org/10.1145/2939672.2939785.

[54] TensorFlow Developers. TensorFlow. Zenodo; 2023, http://dx.doi.org/10.5281/ZENODO.4724125.

[55] Garland NA, Maulik R, Tang Q, Tang X-Z, Balaprakash P. Efficient data acquisition and training of collisional-radiative model artificial neural network surrogates through adaptive parameter space sampling. Mach Learn: Sci Technol 2022;3(4):045003. http://dx.doi.org/10.1088/2632-2153/ac93e7.

[56] Bodini N, Lundquist JK, Livingston H, Moriarty P. How generalizable is a machine-learning approach for modeling hub-height turbulence intensity? J Phys: Conf Ser 2022;2265(2):022028. http://dx.doi.org/10.1088/1742-6596/2265/2/022028.

[57] Peña A, Schaldemose Hansen K, Ott S, van der Laan MP. On wake modeling, wind-farm gradients, and AEP predictions at the Anholt wind farm. Wind Energy Sci 2018;3(1):191–202. http://dx.doi.org/10.5194/wes-3-191-2018.

[58] International Electrotechnical Commission. Power performance measurements of electricity producing wind turbines. International standard 61400-12-2 wind energy generation systems - part 12-1, 2013.

[59] Letizia S, Moss C, Puccioni M, Jacquet C, Apgar D, Iungo GV. Effects of the thrust force induced by wind turbine rotors on the incoming wind field: A wind LiDAR experiment. J Phys: Conf Ser 2022;2265(2):022033. http://dx.doi.org/10.1088/1742-6596/2265/2/022033.

[60] Puccioni M, Moss CF, Jacquet C, Iungo GV. Blockage and speedup in the proximity of an onshore wind farm: A scanning wind LiDAR experiment. J Renew Sustain Energy 2023;15(5):053307. http://dx.doi.org/10.1063/5.0157937.

[61] Iungo GV, Porté-Agel F. Volumetric lidar scanning of wind turbine wakes under convective and neutral atmospheric stability regimes. J Atmos Ocean Technol 2014;31(10):2035–48. http://dx.doi.org/10.1175/JTECH-D-13-00252.1.

[62] Moss C, Puccioni M, Maulik R, Jacquet C, Apgar D, Valerio Iungo G. Profiling wind LiDAR measurements to quantify blockage for onshore wind turbines. Wind Energy 2023.

[63] Iungo GV. Experimental characterization of wind turbine wakes: Wind tunnel tests and wind LiDAR measurements. J Wind Eng Ind Aerodyn 2016;149:35–9. http://dx.doi.org/10.1016/j.jweia.2015.11.009.

[64] Maulik R, Egele R, Lusch B, Balaprakash P. Recurrent neural network architecture search for geophysical emulation. In: SC20: international conference for high performance computing, networking, storage and analysis. 2020, p. 1–14. http://dx.doi.org/10.1109/SC41405.2020.00012.