Revised manuscript (clean version)

Prediction of wind turbines power with physics-informed neural networks and evidential uncertainty quantification

Alfonso Gijón[a,b,c,*], Ainhoa Pujana-Goitia[d], Eugenio Perea[d], Miguel Molina-Solana[a,b,e], Juan Gómez-Romero[a,b]

[a] *Dept. of Computer Science and AI, University of Granada, Spain*
[b] *Research Centre for Information and Communication Technologies (CITIC-UGR), University of Granada, Spain*
[c] *Dept. of Mathematics, University of Córdoba, Spain*
[d] *TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Spain*
[e] *Dept. of Computing, Imperial College London, United Kingdom*

Abstract

The ever-growing use of wind energy requires the optimization of turbine operations through pitch angle controllers and early fault detection. Accurate and robust models that replicate turbine behavior are essential, particularly for predicting generated power from wind speed. Existing empirical and physics-based models often fail to capture the complex relationships between input variables and power output, aggravated by wind variability. Data-driven methods offer promising alternatives by improving model accuracy and scalability with large datasets. In this study, we use physics-informed neural networks to model historical data from four turbines in a wind farm, embedding physical constraints into the learning process. The proposed models predict power, torque, and power coefficient with high accuracy for both the data and the governing physical laws. Notably, neural networks improve the prediction of the power coefficient by an order of magnitude over empirical models. Physics-informed neural networks also show higher robustness than standard networks under limited data, maintaining accuracy even when trained on reduced datasets. Finally, the inclusion of an evidential layer provides uncertainty estimations that align with absolute errors and allow confi-

*Corresponding author
Email address: alfonso.gijon@ugr.es (Alfonso Gijón)

dence intervals on the power curve, ensuring consistency with both observed data and manufacturer specifications.

Keywords: Wind Energy, Power Prediction, Wind Turbine Modeling, Physics-Informed Neural Networks, Uncertainty Quantification

1. Introduction

The growing use of renewable energies is crucial in addressing climate change and working towards a sustainable energy future. According to the International Renewable Energy Agency (IRENA), annual deployment of around 1000 GW of renewable power is needed to stay on a 1.5 °C pathway [1]. Offshore wind energy has experienced significant advancements and growth in recent years, establishing itself as a reference technology in the renewable energy sector, and is expected to add up to 160 GW in Europe by 2030, according to WindEurope [2]. At the same time, the thriving development in sensor and storage technology has enabled the collection of ever-increasing amounts of data [3, 4]. This, combined with the development of flexible and powerful data-driven and machine-learning approaches, has introduced a new paradigm of digital, autonomous, and decentralized control for manufacturing systems [5–7]. In this scenario, it is crucial to have accurate and robust wind turbine (WT) models for operations optimization and automatic diagnoses of faults.

Regarding wind energy, current devices paired with advanced analytics capabilities enable real-time performance monitoring, optimization of turbine operations, and predictive maintenance through early fault detection [8], thus minimizing downtime [9]. However, data-based WT modeling can present several challenges [10], making the direct application of data-driven techniques complex. The first challenge is obtaining data in sufficient quantity and quality on various parameters, such as wind speed and direction, power generated, or performance data. Another difficulty arises from the complexity of WTs, involving multiple components and interrelated variables. Furthermore, wind variability is another obstacle, as it is a highly fluctuating source of energy that sometimes requires complex techniques to mitigate the operational distortion [11]. Traditional physics-based models are becoming insufficient when faced with the system complexity and its various uncertainties, so validating and calibrating a WT model from measured data is vital to ensure its accuracy and reliability [12]. Overcoming these

challenges requires a combination of technical knowledge, experience in the field, and access to relevant and reliable data.

Power prediction in WTs plays a fundamental role in wind energy management and power forecasting [13]. In addition to its applications in the management of generated energy and integration into the electricity grid [14], power prediction combined with data analysis and monitoring techniques can help to detect and diagnose powertrain component failures [15]. By comparing power predictions with real-time data, it is possible to identify deviations and anomalies that could indicate component problems, enabling early intervention and effective fault management. Besides, power prediction is essential for effective pitch angle control and optimal WT performance, as it provides real-time information about the power generated by the WT. This information can also be used to adjust blade angles to maximize power generation, protect the WT, and achieve efficient overall control [16].

The lack of precise and robust physics-based models for predicting the power production of utility-scale wind farms [17] encourages the application of data-driven approaches. Moreover, as wind-power data are often very noisy, fitted WT power curves can differ significantly from the theoretical ones provided by manufacturers [13]. Deep learning methods, particularly artificial neural networks (ANNs), have shown great success in multivariate regression tasks. They are especially effective in purely data-driven settings, where there is enough data but the physical model is unknown, incomplete, or too complex to use directly. Although neural networks are black-box models, the development of new architectures able to respect certain constraints, like physics-informed neural networks (PINNs) [18, 19], makes them more attractive for modeling physical phenomena. PINNs can reproduce real data while respecting some physical laws, giving rise to accurate and robust models. Based on those ideas, this work presents the design of several wind-power models that achieve great accuracy while fulfilling some fundamental physical equations governing the system. Importantly, as demonstrated in this study, PINNs achieve comparable performance to standard neural networks when sufficient data are available, but clearly outperform them when the amount of training data decreases. This ability to exploit physical knowledge makes PINNs particularly useful for scenarios with limited or incomplete datasets, which are common in wind energy applications.

Furthermore, uncertainty quantification on deep learning models could be one of the main strengths of neural networks as universal regressors compared to standard methods, which are usually based on the optimization of

parameters from analytical formulas —either derived from first principles or informed by empirical knowledge— and are used for reproducing real data. Reliability and uncertainty modeling was described as a key research challenge for wind energy by the European Academy of Wind Energy (EAWE) [20]. In our work, uncertainty is easily estimated through an evidential deep learning layer [21], which allows us to obtain a power curve model consistent with the specifications provided by the manufacturer.

The main goal of this work is to design and validate robust and accurate prediction models for the WT power, which is the previous step to developing optimal controllers and applying failure detection methods. We built neural network models not only for the power coefficient but also for the torque and the power itself, comparing their performances in predicting the power generated by the turbine. Further, we introduced the use of physics-informed neural networks to model those magnitudes. For each model, we added an evidential layer to the output, a recent and efficient method to quantify the uncertainty of the prediction, which is key to identify data deviated from normal. In the end, we have accurate models for prediction of the WT variables, including the generated power, available at the repository [22][1]. Even though these models have been trained with a dataset from a specific manufacturer, they could be easily transferred to different manufacturing data.

The rest of the paper is organized as follows. Section 2 introduces the physical background of the problem and provides context discussing some related work. Section 3 establishes the experimental setup and presents the methods employed to build the WT models. Data explanation and the discussion of the experiments are found in Sections 4 and 5, respectively. Finally, Section 6 summarizes the main conclusions of this work.

2. Background

2.1. Physics of wind turbines

The low-scale behavior of a wind turbine can be modeled with physics-based approaches. However, that is very challenging due to the complexity

[1]All the scripts for data preprocessing, building, training and evaluating the models are available at this github repository, together with the reduced dataset without the non-physical and anomalous data points. This allows the reproduction of all the figures and results presented in the paper.

	$R\,(\mathrm{m})$	$A\,(\mathrm{m^2})$	g	$\rho\ (\mathrm{kg/m^3})$
MM82	41	5281	106	1.225

Table 1: Physical parameters of the MM82 wind turbines, obtained from [24]. Here, R is the rotor radius, A is the swept area, g is the drivetrain gear ratio, and ρ is the air density.

of fluid flows that must be considered, such as geophysical and atmospheric effects, the turbulence at high Reynolds numbers, as well as wind-turbine characteristics and wind-farm layouts, among others [23]. Although low-order physics-based models help provide qualitative insight into wind turbine behavior, they usually fail to predict power production accurately at utility scale, as they rely on many simplifications and often ignore important physical effects [17].

However, some physical relationships are well established through fundamental equations involving high-scale variables. From a mechanical point of view, the generated power P is given by

$$P = gT\omega\,, \tag{1}$$

where g is the drivetrain gear ratio (a constant property characteristic of the turbine model), T is the mechanical torque of the rotor, and ω is its angular velocity. This equation can be interpreted either as a state equation of of a wind turbine or as a conservation/symmetry law that must be fulfilled independently of the underlying wind-power model. In turn, the power extracted by a WT from the kinetic energy of the incoming wind yields:

$$P = \frac{1}{2}C_p\rho A v^3\,, \tag{2}$$

where C_p is the power coefficient, ρ is the air density, A is the swept area covered by the blades of the WT and v is the wind velocity. The physical parameters of the extended turbine model for the MM82, a specific model manufactured by the German company *REpower Systems*, are shown in Table 1, while a scheme of the elements of a WT can be seen in Figure 1a.

The power generated by a WT strongly depends on the power coefficient, C_p, a dimensionless quantity influenced by the intrinsic characteristics of a WT (size, geometry, aerodynamic properties), and the operating conditions, defined by the wind speed and pitch angle variables. Usually, the objective is to maximize the value of C_p to achieve the highest efficiency in the conversion of wind energy into electrical energy [25, 26], subjected to a theoretical

maximum value of 0.5926 (the Betz limit). The maximum power of WTs in variable wind speed regions is achieved by a suitable control of the pitch angle β, defined as the angle between the lateral axis of the blades and the relative wind. Pitch control of WTs is complex due to the intrinsic non-linear behavior of these systems and external disturbances, due to changing wind conditions and other meteorological phenomena. Therefore, an accurate estimation of the C_p coefficient and/or the generated power as a function of the wind speed and the operating conditions is essential for effective pitch control and fault detection.

2.2. Modeling of the power

It is important to note that the power coefficient is not constant and can vary with changes in wind speed, turbulence, and other operating conditions, so that different WT designs and operating strategies may have different C_p characteristics. Therefore, due to its complexity, C_p is often analyzed and optimized empirically during the design, operation, and control of WTs to maximize their energy conversion efficiency. However, WTs typically operate in complex and changing conditions, which often deviate significantly from controlled experimental conditions, so the information provided by manufacturers does not usually accurately reflect the actual performance of WTs in real-world scenarios [27, 28].

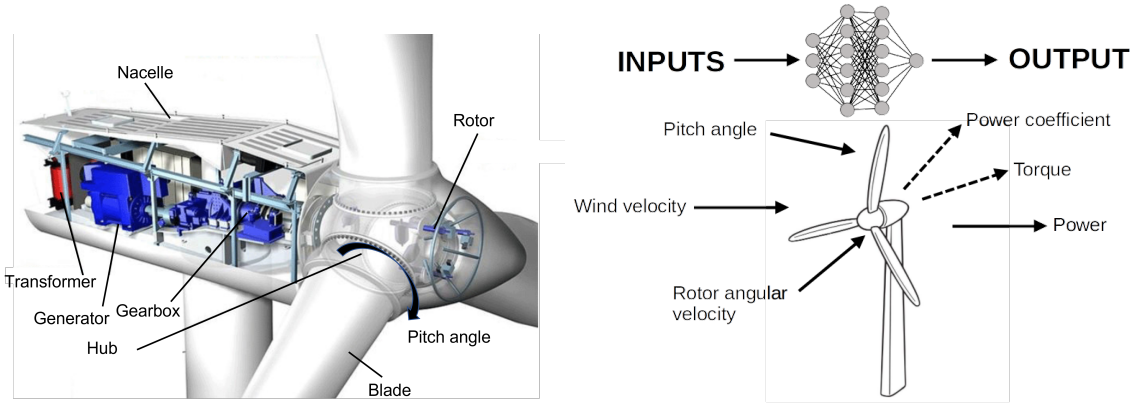
Though the exact value of the power coefficient C_p is determined by complex underlying fluid dynamics processes, it can also be approximated as a function of the high-scale variables. In this way, the conversion efficiency of a WT depends on the input variables (v, β, ω) and different empirical formulas have been proposed for the power coefficient [29–31], being the exponential one of the most widely used:

$$C_p(\lambda, \beta) = c_0(c_1\gamma - c_2\beta - c_3\beta^{c_4} - c_5)e^{-c_6\gamma} + c_7\lambda\,, \tag{3}$$

where $\lambda = \omega R/v$ represents the tip speed ratio, β is the pitch angle and γ is defined by

$$\gamma = \frac{1}{\lambda + d_0\beta + d_1} - \frac{d_2}{1+\beta^3}\,. \tag{4}$$

This empirical model has eleven free parameters, denoted as c_i and d_i, fitted to reproduce real measurements. The main advantage of parametric models is their simple form, but sometimes it is hard to accurately fit the complex relationship between wind speed and power. In contrast, non-parametric



(a) (b)

Figure 1: (a) Main components of a wind turbine. Image adapted from [33] (b) Scheme of a neural network model, receiving the pitch angle, wind velocity and rotor angular velocity as input arguments and predicting power coefficient, torque or power as output.

methods are more accurate and do not require an explicit expression to learn automatically from large amounts of data. However, they can suffer overfitting, significant discrepancies with the manufacturer specifications, and may suffer from performance degradation when applied to real-world datasets [32].

2.3. Related work

In recent years, intelligent techniques have been employed for pitch angle control and fault detection [34–39], trying to maximize the generated power. These studies usually use a parametric empirical formula for the power coefficient C_p, with exponential, polynomial or sinusoidal form, where the free parameters are fitted to approximately reproduce the data at hand [29–31]. Recently, some authors have proposed using neural network models for regression of the power coefficient, improving the quality of the prediction of the generated power [32, 40, 41]. The accurate mathematical models of the system provided by neural networks open the door to applying powerful control methods such as deep reinforcement learning [42–45], which require the exploration of many configurations of the environment. Directly applying data-driven, especially deep learning, methods can lead to good results in predicting the WTs power curve and play a crucial role in turbine performance analysis, but they suffer a lack of robustness and physical sense. To find an optimal controller, the predictive model should robustly reproduce

real data, respecting the physical constraints imposed by the system even for configurations that deviate from normal behavior. Some authors have used PINNs to build wake models representing the flow field behind a WT [17] or for fatigue damage prediction [46]. However, as far as we know, PINNs have never been applied to enforce the high-scale global equations relating the outputs (power coefficient, torque, and power) with the input variables (wind speed, pitch angle, and angular velocity), which are treated in this work as conservation or symmetry principles, without imposing further internal restrictions in the form of the predicted quantities.

Moreover, uncertainty modeling of WTs is still in its infancy but constitutes a fundamental aspect for minimizing unexpected failures and downtime [20]. Some Bayesian [47, 48] or Monte Carlo [49–51] methods have been recently employed to approximate the uncertainty in estimating the WT power curve and wake models, though they are computationally expensive. In our work, we equip the neural network models for power prediction with a final evidential layer [21]. This is an efficient and easily implemented mechanism to quantify epistemic uncertainty. As a result, the model is able to accurately reproduce both real data and the manufacturer specifications within the confidence interval.

3. Methods

3.1. Problem statement

The objective of this work is to predict the electrical power P generated by a wind turbine as a function of a set of measurable high-level operational variables. Specifically, the input variables are the wind speed v, the blade pitch angle β, and the rotor speed ω. Thus, the general regression task can be formulated as:

$$P = f(v, \beta, \omega), \tag{5}$$

where f is the mapping learned from data, potentially constrained by known physical relations. Each WT operating condition can be represented by a 5-tuple $\big(v, \beta, \omega, T, P\big)$, where T denotes the mechanical torque. These variables are recorded from real turbine operation and collected into datasets. Additionally, the power coefficient C_p, a key efficiency indicator, is computed from power and wind speed using the physical relation:

$$C_p = \frac{2P}{\rho A v^3}, \tag{6}$$

where ρ is the air density and A the rotor swept area. Moreover, the physical relationships among P, T, ω, and C_p are described by the following algebraic expressions: the mechanical power is related to torque via Equation 1, and to the power coefficient via Equation 2. These equations provide additional structure that can be exploited to guide learning. While the ultimate goal is to predict the power P, this can be achieved either directly or indirectly —by first predicting torque T or power coefficient C_p and then applying the relevant physical equations. These alternative formulations enable the integration of domain knowledge, enhancing both model robustness and interpretability.

To solve this regression task, we adopt a data-driven approach based on neural networks, which can approximate the mapping f directly from measured data, without relying on empirical parametrizations. As shown in Figure 1b, we train different network configurations to predict either P, T, or C_p from the input variables. Their performance is assessed in terms of accuracy on power prediction. To further enhance generalization and ensure physical consistency, we extend some of these models to physics-informed neural networks, where Equation 1 and Equation 2 are embedded into the loss function as soft constraints. This means that, during training, the models are penalized not only for prediction errors, but also for violating the known physical relationships, effectively encouraging them to adhere to domain laws. Finally, we incorporate an uncertainty quantification mechanism via evidential deep learning to estimate uncertainty in the predictions.

3.2. Physics-Informed Neural Networks

As commented previously, a PINN model can learn from data and also from equations, which can be totally known or dependent on some unknown parameters. In the latter case, the training of the model results in the learning of both the desired quantity and the model parameters that best describe the observed data. In time-independent systems, a PINN is just an artificial neural network $\mathrm{NN}(x; W, b)$, trained to approximate a target function $u(x)$, both differentiable functions of the input x. Sometimes, u is known to fulfill some physical conditions or constraints, which generally can be expressed as ordinary or partial differential equations, written as

$$F\big[u(x); \mu\big] = 0\,, \tag{7}$$

where F is a functional on the solution function u and the model parameters μ. The network can learn the target function and the optimal parameters of

the physical equation by minimizing the loss function[2] with respect to the weights and biases of the neural network and the model parameters:

$$L = L_{\text{data}} + L_{\text{phys}}\,, \tag{8a}$$

$$L_{\text{data}} = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \left|\text{NN}_i - u_i\right|, \tag{8b}$$

$$L_{\text{phys}} = \frac{1}{N_{\text{phys}}} \sum_{j=1}^{N_{\text{phys}}} \left|F\left[\text{NN}_j; \mu\right]\right|, \tag{8c}$$

where $\text{NN}_i = \text{NN}(x_i; W, b)$ is the neural network model evaluated at x_i. The physics-based loss acts as a regularizer, steering the model towards solutions that comply with the known physical law expressed by F. Note that N_{data} and N_{phys} may differ. While N_{data} corresponds to observed samples used to fit the target output, N_{phys} refers to the points where the physical constraint is evaluated. These can be selected as a subset of the data or defined on a synthetic collocation mesh over the input space. This mesh can follow the empirical distribution or be uniformly sampled, depending on the desired resolution and generalization.

In our case, the target function u has to fulfill physical ordinary equations given by 1 and 2, which directly provides an estimation of the target magnitude, $\tilde{u}$, as a function of the inputs. Hence, considering there are no physical parameters to be optimized, the functional F has the simple form

$$F[u(x)] = u(x) - \tilde{u}(x)\,. \tag{9}$$

First, one could consider the power as the target function, $u \equiv P$. In this case, the inputs of the PINN are (v, β, ω, T), though only (v, β, ω) are inputs of the neural network, and T is only used into the physical term. Details about the model graph architecture can be consulted in Appendix A. In addition to the regression of the P data, an estimation of the power can be computed from Equation 1, $\tilde{u} = gT\omega$, so the physical condition is expressed as

$$\text{NN}(v, \beta, w) - gT\omega = 0\,, \tag{10}$$

[2]Although the loss is expressed here as the mean absolute error between prediction and target, all the arguments and conclusions are valid for different loss functions.

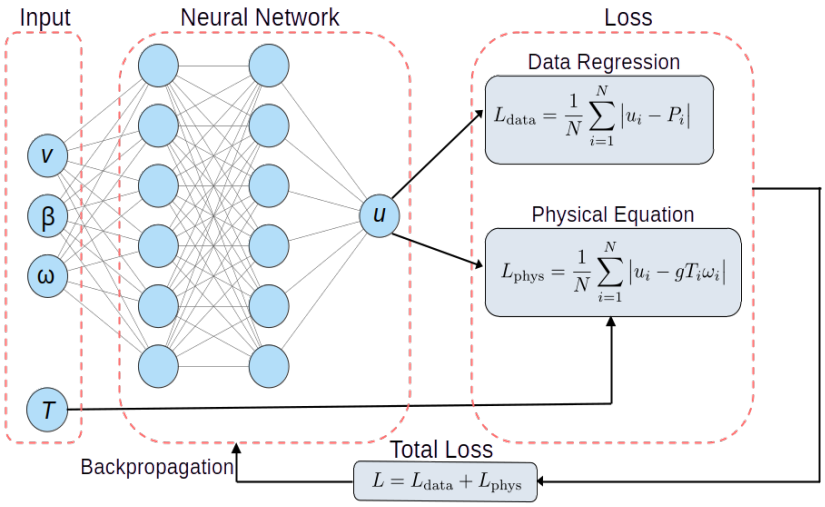


Figure 2: Diagram of the PINN architecture with power as target, $u = P$. Inputs (v, β, ω) feed the neural network, while torque T is used in the physics loss. The network has two hidden layers and outputs P. Data-driven and physics-based losses are combined into the total loss, which is backpropagated to train the model.

while the residuals are:

$$L_{\text{data}} = \frac{1}{N}\sum_{i=1}^{N}\left|\text{NN}_i - P_i\right|, \tag{11a}$$

$$L_{\text{phys}} = \frac{1}{N}\sum_{i=1}^{N}\left|\text{NN}_i - gT_i\omega_i\right|. \tag{11b}$$

The training of this PINN results in a neural network predicting the measured power while respecting the mechanical relation between power and torque. Figure 2 illustrates this PINN architecture, where the neural network receives inputs (v, β, ω) to predict power P. In this setup, torque T is not part of the network input but is used solely within the physics loss term to enforce the mechanical power equation. The total loss function combines the data-driven regression loss L_{data} with the physics-informed loss L_{phys}, ensuring the model's prediction adhere to both empirical observations and the governing physical law expressed by the relation between power and torque.

Secondly, the power coefficient factor could be considered as the target function of a PINN, $u \equiv C_p$. This formulation is particularly useful when estimating aerodynamic efficiency directly, and can help improve interpretabil-

ity in performance studies. Similarly to the previous case, the inputs of the PINN are (v, β, ω, T), but only (v, β, ω) are inputs of the neural network, and T is only used for the physical residual. In this case, an estimation of the power coefficient can be computed by equaling the right-hand sides of Equations 1 and 2 to obtain $\tilde{u}(v, T, \omega)$, so the imposed condition is

$$\mathrm{NN}(v, \beta, w) - \frac{2gT\omega}{\rho A v^3} = 0\,, \tag{12}$$

and the residuals are:

$$L_{\text{data}} = \frac{1}{N}\sum_{i=1}^{N} \left|\mathrm{NN}_i - C_{p,i}\right|, \tag{13a}$$

$$L_{\text{phys}} = \frac{1}{N}\sum_{i=1}^{N} \left|\mathrm{NN}_i - \frac{2gT_i\omega_i}{\rho A v_i^3}\right|. \tag{13b}$$

The training of this PINN results in a neural network predicting the power coefficient while respecting physical relations. The generated power is computed from the predicted C_p by means of Equation 2.

Lastly, a PINN with the torque T as target function would lead to the equation

$$\mathrm{NN}(v, \beta, w) - \frac{P}{g\omega} = 0\,. \tag{14}$$

In this case, the inputs of the PINN are (v, β, ω, P), but only (v, β, ω) are inputs of the neural network and P is only used to impose the physical equation. Residuals are:

$$L_{\text{data}} = \frac{1}{N}\sum_{i=1}^{N} \left|\mathrm{NN}_i - T_i\right|, \tag{15a}$$

$$L_{\text{phys}} = \frac{1}{N}\sum_{i=1}^{N} \left|\mathrm{NN}_i - \frac{P_i}{g\omega_i}\right|. \tag{15b}$$

It should be noticed that torque is just an intermediate quantity for computing the interest output, the power, through the equation $P_T = gT\omega$, so that the physical condition is satisfied by construction. The residual L_{data} from Equation 15a contributes to the regression of the torque data and hence, a power coming from the formula $P_T = gT\omega$, fulfilling Equation 1. In turn,

the introduction of the variable P into the residual L_{phys} of Equation 15b leads to a better prediction of the measured power P, because it leads the model to learn $T_P = P/g\omega$. Paradoxically, due to the construction of the PINN for the regression of the torque and the subsequent calculation of the power from it, L_{data} tends to fulfill the physical condition whereas L_{phys} is intended to the regression of the measured power.

3.3. Evidential layer for uncertainty quantification

Uncertainties were evaluated by including an evidential deep learning output layer, following the method developed by Amini et al. (2020) [21]. While representations of aleatoric uncertainty can be learned directly from data, several approaches exist for estimating epistemic uncertainty. Evidential deep learning assumes that observed targets are drawn from a Gaussian distribution with unknown mean and variance, $y \sim \mathcal{N}(\mu, \sigma^2)$. Accordingly, the mean and variance are represented as

$$\mu \sim \mathcal{N}(\gamma, \sigma^2, \nu^- 1)\,, \tag{16a}$$

$$\sigma^2 \sim \Gamma^{-1}(\alpha, \beta)\,, \tag{16b}$$

where Γ is the gamma function and $(\gamma, \nu, \alpha, \beta)$ are parameters, leading to a posterior distribution described by a Normal Inverse Gamma (NIG) distribution from which the prediction and epistemic uncertainty are computed from the following equations, respectively:

$$\mathbb{E}(\mu) = \gamma\,, \tag{17a}$$

$$\text{Var}(\mu) = \frac{\beta}{\nu(\alpha - 1)}\,. \tag{17b}$$

The loss function for training the evidential deep learning model includes a negative likelihood loss L^{NLL} and an evidential regularization term L^{EL}. The former corresponds to the negative log-evidence under the assumed Normal Inverse Gamma distribution and encourages accurate predictions. The latter penalizes unjustified evidence in the case of inaccurate predictions, helping to control overconfident outputs. The total loss is a weighted combination of both terms:

$$L(x) = L^{\text{NLL}}(x) + \lambda L^{\text{EL}}(x)\,. \tag{18}$$

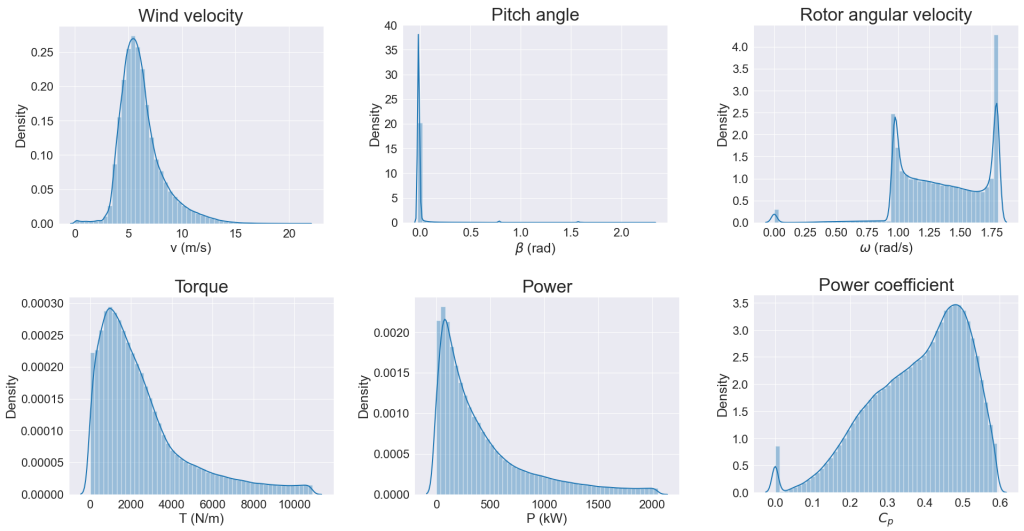


Figure 3: Distribution of the variables $(v, \beta, \omega, T, P, C_p)$ on the dataset.

A detailed derivation of L^{NLL} and L^{EL} can be found in the original paper [21]. Here, we set the regularization coefficient $\lambda = 0.01$, a value shown to be effective in similar regression settings.

4. Data

Real data, obtained from the website of ENGIE Renewables, were collected from 4 different MM82 turbines from the Senvion manufacturer at 'La Haute Borne' wind farm [52, 53], into two runs between 2013-2016 and 2017-2020. Among the available variables related to the physical conditions and the turbine itself, we focus our study on wind velocity, pitch angle, rotor speed, torque, and power, represented as (v, β, w, T, P). Our dataset is composed of 1057968 5-dimensional vectors. A value of the power coefficient C_p can be computed for each data point by inverting Equation 2 and using measured P and v as input variables. Data were obtained from direct or indirect measurement, so they contain noise; thus, some preprocessing is desirable.

First, we removed all non-physical data, e.g., those with a power coefficient higher than the theoretical Betz limit, $C_p > 0.5926$. After that, we identified anomalous data by comparing measured power with an estimation of the power curve based on the iterative median technique [54]. Data points that deviated more than three standard deviations from the estimated

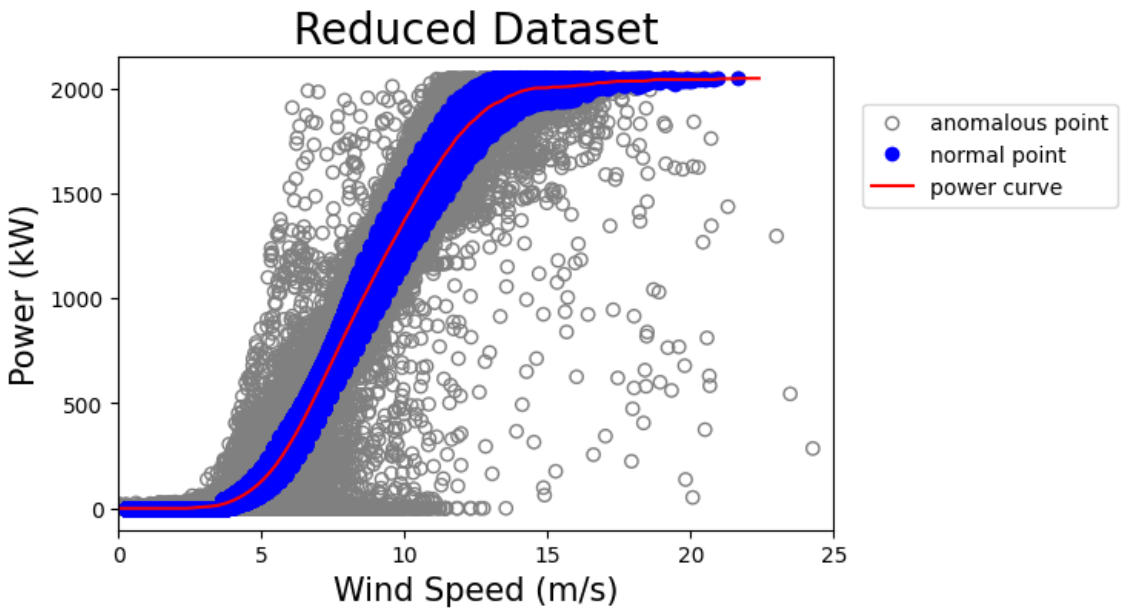


Figure 4: Estimation of the power curve (red) obtained using the iterative median technique applied to the turbines in our dataset. Normal data points (retained after filtering) are shown in blue, and anomalous points (discarded) are shown in grey.

median were discarded. Low-velocity data are especially noisy, resulting in physical but unrealistic values of the power coefficient C_p, obtained by dividing by v^3; therefore, we applied a power cutoff of 25 kW to data with wind speeds below 3.5 m/s, and data with a higher power were removed. In fact, that cutoff matches the lowest power value provided by the manufacturer for MM82 turbines [24]. The distribution of the variables is shown in Figure 3, and the estimated power curve, along with the retained (blue) and discarded (gray) data, is depicted in Figure 4. The red line represents the iterative median estimate used for anomaly detection.

As a result of preprocessing, the dataset was reduced from more than 1 million data points to 7.27×10^5 data, approximately a 70 % of the original dataset. The preprocessing allowed us to avoid anomalous data, coming from failure or a non-optimal behavior, which could obstruct the learning of our models, but it did not remove the inherent noise of the data. Such noise can be quantified by comparing the power data itself, P, with the power estimated from the mechanical relation of Equation 1, using the torque and rotor angular velocity data, $P_T = gT\omega$. On average, we obtain a mean

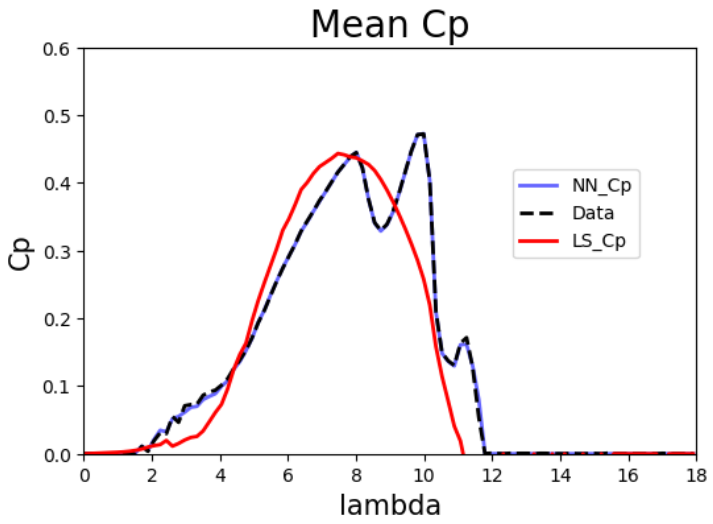


Figure 5: Comparison of mean C_p curves obtained from data (black dash line), a neural network model (blue) and the parametric empirical model (red).

absolute percentage error of 1.7 %, which establishes a lower limit for any model trying to predict the power when trained on noisy data.

5. Experimentation and discussion

5.1. Experimental setup

Our experimental setup is based on artificial neural network models, trained using the *Tensorflow* library [55] and hyperparameters optimized with the *Hyperband* algorithm [56] from the *Keras Tuner* library [57], due to its efficiency in early termination of underperforming configurations. The search space included learning rate, number of layers, units per layer, and activation function. PINNs were implemented with the *SciANN* library [58], using optimal architectures previously found for standard artificial neural networks. For uncertainty quantification, evidential layers were added using the *Evidential Deep Learning* library [21]. All the calculations were performed on a desktop computer with an 11th Gen Intel Core i7-11800H processor, 16 GB RAM memory and NVIDIA GeForce RTX 3060 graphic card.

5.2. Limitations of the empirical model

The first attempt to model generated power was based on the parametric formulation of the power coefficient provided by the empirical Equation 3.

	LS_Cp	NN_Cp
MAE (kW)	113.196	15.341
RMSE (kW)	167.893	27.770
MAPE (%)	37.65	3.63
R2	0.8570	0.9961

Table 2: Comparison of error metrics for power coefficient models: one obtained by fitting parameters of an empirical model using the least squares method (LS_Cp), and the other using a standard artificial neural network (NN_Cp)

The optimal parameters to fit C_p data were obtained using non-linear least squares. However, this approach turned out to be insufficient, as the chosen parametrization is not general enough to capture the complexity of the observed data. This limitation is clearly visible in Figure 5, where the empirical fit (red line) fails to match the average C_p curve derived from the dataset (black dashed line).

In contrast, when using an artificial neural network as a regression model for C_p, the predicted curve (blue) closely follows the observed trend in the data, effectively capturing distinct regimes across the operational range. A quantitative comparison of the performance metrics[3] of both models can be found in Table 2. As expected, the neural network model vastly outperforms (by one order of magnitude for the error metrics) the empirical model.

5.3. Advantages of physics-informed neural networks

Hereafter, standard and physics-informed neural networks with different target variables are compared. The hyperparameters were optimized with the *Hyperband* algorithm from the *Keras Tuner* library, within the space generated by the combinations of `n_layers` $\in \{1, 2, 4\}$, `n_neurons` $\in \{8, 16, 32, 64, 128\}$, `learning_rate` $\in \{0.01, 0.001, 0.0001\}$ and `activation_function` $\in$ {'relu', 'tanh'}. Optimal hyperparameters are shown in Table 3 for each case. The batch size was fixed to 128, mean absolute error as loss function, and a reduced learning rate when loss stopped improving. The dataset was split

[3]The accuracy of different regression models of the power are compared through commonly used metrics, such as mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE) and R-squared (R2).

	NN_Cp	NN_T	NN_P
`n_layers`	2	2	2
`n_neurons`	128	128	64
`learning_rate`	0.001	0.001	0.001
`activation_function`	relu	relu	relu

Table 3: Optimal hyperparameters of the artificial neural network models predicting power coefficient, torque, and power, respectively. These configurations were determined using the Hyperband optimization algorithm. Parameters include number of layers, neurons per layer, learning rate, and activation function.

into 80 % for training and 20 % for testing, being 150 and 200 epochs enough for learning to stagnate, for NNs and PINNs, respectively.

Although the primary quantity of interest is the generated power P, we constructed models targeting not only P but also the power coefficient C_p and torque T. This approach is motivated by the fact that C_p and T can be used to derive the power through the physical relations expressed in Equation 1 and Equation 2, which link power to torque and power coefficient, respectively. Each standard neural network model is labeled as NN_*id* and physics-informed neural network models as PINN_*id*, where *id* denotes the target variable. As explained in subsection 3.2, each PINN incorporates a different physical equation depending on its target variable. The models with different output magnitudes were compared by computing the generated power and using that prediction to calculate the performance metrics. The predicted power from each model was evaluated against both the measured power P from the dataset and the power calculated from measured torque data via the mechanical relation $P_T = gT\omega$. Table 4 presents regression metrics between predicted power and ground truth, where the "Data" columns correspond to comparisons with P the "Phys" columns correspond to comparisons with P_T, all evaluated on the test dataset.

All models demonstrate strong predictive performance for the generated power, with a MAE below 16 kW and a MAPE below 4 %. In particular, the C_p curves predicted by both the NN and the PINN models exhibit excellent agreement with the empirical reference derived from data, as shown in Figure 5. The predicted curves are nearly indistinguishable, which is why the PINN curve was not included separately in the figure. The models reproduce well both P and P_T, hence correctly describing both data and the

	NN_Cp		PINN_Cp		**NN_T**		PINN_T		**NN_P**		PINN_P	
	Data	Phys	Data	Phys	Data	Phys	Data	Phys	Data	Phys	Data	Phys
MAE (kW)	**15.341**	15.122	15.375	14.513	15.797	**14.399**	15.431	14.601	**15.363**	15.173	15.545	14.543
RMSE (kW)	**27.770**	27.856	27.940	26.967	28.511	**26.901**	27.898	27.144	**27.785**	27.967	28.016	26.946
MAPE (%)	**3.63**	3.63	3.70	3.40	3.92	**3.33**	3.72	3.37	**3.64**	3.63	3.84	3.44
R2	**0.9961**	0.9962	0.9960	0.9964	0.9959	**0.9964**	0.9960	0.9963	**0.9961**	0.9961	0.9960	0.9964

Table 4: Comparison of error metrics between predicted power and two types of ground truth—measured data ("Data") and physics-based estimates ("Phys")—for different neural network models predicting power coefficient (C_p), torque (T), and power (P).

physical equation governing the system simultaneously. A deeper analysis can be done when comparing the NN versus the PINN models for the same target variable. For C_p and P models, NNs reproduce data slightly better than PINNs, while PINNs reproduce the physical equation slightly better than standard NNs. This was expected because PINNs have a physical term in the loss function to respect the physics law.

In contrast, the opposite happens for the T models: PINN_T predicts power data better than NN_T, which in turn satisfy better the physical equation. This apparent contradiction is understood by noticing that the data term of the loss function of Equation 15a contributes to the learning of the equation $P_T = gT\omega$, for the way in which the power is calculated through the predicted torque as an intermediate variable. If the model predicts T correctly, it will correctly predict $P_T = gT\omega$. For its part, the physical term of Equation 15b contributes to the learning of the measured P data, because it leads to learning the variable $T_P = P/g\omega$, which gives measured data when converted to power.

It is also remarkable that NN models also satisfy the physical equation with high accuracy, even without explicitly incorporating physical terms into the loss function. NN_Cp and NN_P yield the most accurate predictions for power data, while NN_T best fulfills the physical relation given by Equation 1, as highlighted with bold in Table 4. These results suggest that, due to the large volume and variability of the dataset, standard neural networks are capable of implicitly learning and respecting the underlying physical laws. Notably, we observed that a randomly selected 10% subset of the original dataset—approximately 10^5 data points—is sufficient for the models to reach their maximum predictive accuracy. However, as the amount of training data

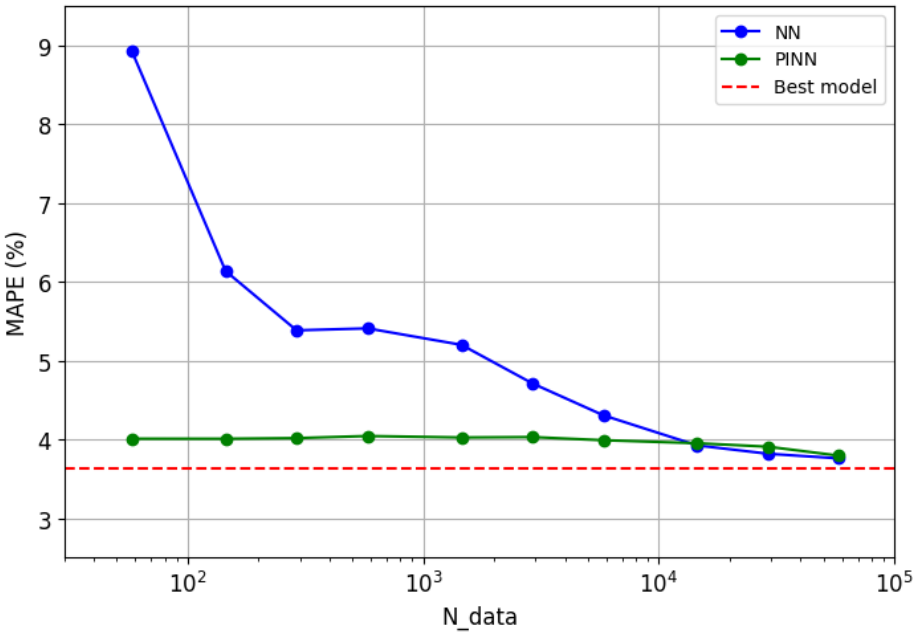


Figure 6: Comparison of MAPE in power prediction between a standard NN and a PINN as a function of the training data size. While the NN performance degrades with fewer data, the PINN maintains stable accuracy.

	NN_Cp_evi	NN_T_evi	NN_P_evi
MAE (kW)	15.080	15.692	15.241
RMSE (kW)	27.188	28.236	27.606
MAPE (%)	3.58	3.91	3.61
R2	0.9963	0.9960	0.9961
MAU (kW)	17.610	17.314	18.603

Table 5: Error metrics comparing measured and predicted power for neural network models enhanced with an evidential output layer. MAU accounts for the mean absolute uncertainty, quantifying the average predictive uncertainty, calculated from the variance estimated by the the evidential layer for each prediction.

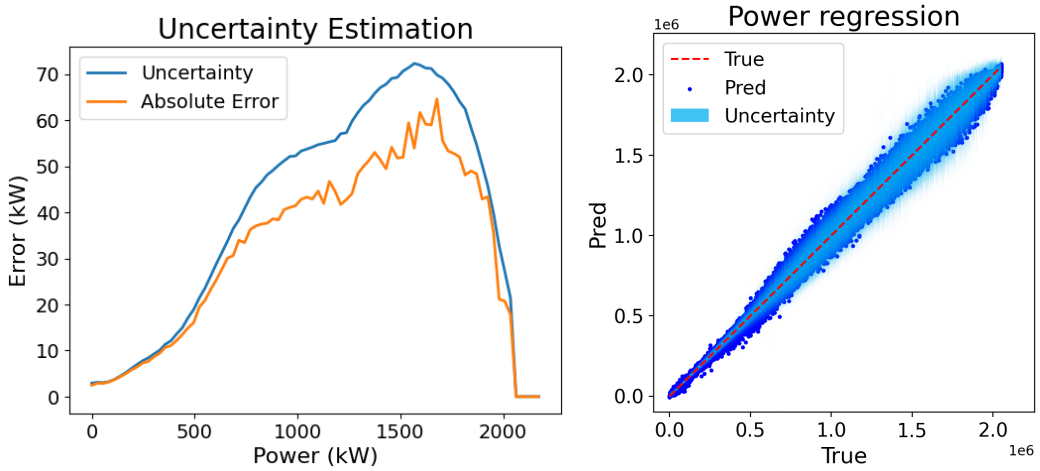
decreases, the performance of standard NN models degrades, whereas PINNs maintained stable accuracy close to their optimum, as illustrated in Figure 6, where NN_P and PINN_P were trained on progressively smaller datasets. In this experiment, the number of samples used to the data-driven loss term L_{data} was systematically reduced, while the number of points (ω, T) used to the physics-based term L_{phys} remained constant. It is important to note that the physics-based term does not rely on power labels; instead, it guides the model by enforcing the physical constraint expressed by Equation 10 and Equation 11b.

5.4. Uncertainty quantification with evidential layer

As a final step, we selected the best-performing models from Table 4 —the neural networks predicting power coefficient, torque, and power, highlighted in bold in the table— and extended them by incorporating an evidential deep learning output layer to quantify the uncertainty of their predictions. The performance metrics comparing predicted and actual power values are presented in Table 5, alongside the mean absolute uncertainty (MAU), computed as the mean of the predictive variance defined in Equation 17b. All three models yields results comparable to their counterparts without the evidential layer. Notably, the MAU slightly exceeds the MAE, indicating that, on average, the predicted values lie within the estimated confidence interval.

To examine how uncertainty varies across different power output ranges, Figure 7a presents both the absolute error and the associated uncertainty as functions of the predicted power. Both curves exhibit a similar bell-shaped trend, peaking at intermediate-to-high power values (around 1500 kW) and decreasing toward the extremes, where power is either very low or near its maximum. Notably, the predictive uncertainty is slightly higher than the absolute error across all power ranges, which is a desirable property, as it ensures that the predicted confidence intervals are consistent with actual errors. Although this figure corresponds to the model trained with power P as the target variable, similar patterns are observed for the models predicting power coefficient C_p and torque T. This behavior aligns with Figure 7b, which displays predicted versus true power values for the test set, with uncertainty intervals overlaid. As shown, the uncertainty is greater in regions of higher data dispersion, particularly at intermediate and high power levels.

Once the uncertainty quantification method was validated and shown to be consistent with the actual confidence levels, the power curve as a function of wind speed, predicted by the neural network model, was analyzed and is



(a) (b)

Figure 7: (a) Absolute error (orange) and predictive uncertainty (blue) as functions of power output for the test set. Both curves exhibit a bell-shaped trend, with maximum values around intermediate-to-high power levels. (b) Predicted versus true power values, including uncertainty intervals for each prediction. The uncertainty is higher in regions with greater data dispersion, reflecting model confidence.

presented in Figure 8. This figure also includes the empirical curve estimated from data and the reference curve provided by the manufacturer [24]. The predictive uncertainty is represented by a 3σ confidence interval, indicating that true model output lies within this range with a probability of 97.3 %. Indeed, the curve estimated from data (dashed black line) remains entirely within the confidence interval (shaded blue area) across the entire wind speed range. The manufacturer's curve (solid black line) also lies within this interval for most of the range, deviating slightly only around a wind speed of 5 m/s.

6. Conclusions

In this work, we designed and validated models for predicting the generated power of wind turbines as a function of wind velocity, pitch angle, and rotor angular velocity. They were trained using real historical data of 4 turbines from 'La Haute Borne' wind farm. During preprocessing, raw data were used to estimate the power curve as a function of the wind speed and 30 % of data were considered as anomalous points and thus discarded for the

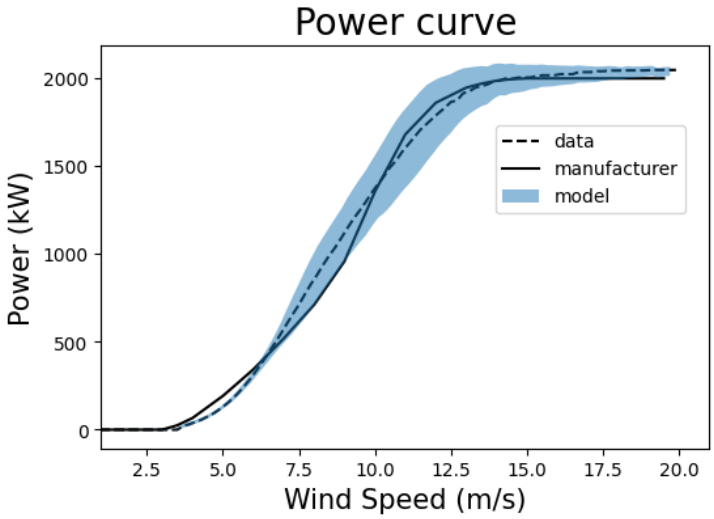


Figure 8: Comparison of the power curve predicted by the neural network model, the empirical curve estimated from data, and the reference curve provided by the manufacturer. The shaded blue area represents the model's predictive uncertainty.

subsequent analysis.

The first approach was to fit the power coefficient with a parameterized empirical formula, but it showed poor results on the dataset under analysis, being unable to capture the non-linear relationship between the target magnitude and the wind velocity. The absence of a precise physical model describing the system and the availability of data make it very suitable for applying data-driven approaches. Neural network models showed great performance when used for regression of the output variables like power coefficient, torque or power itself. Furthermore, the physical information provided by the state equation relating torque and power paved the way for the implementation of physics-informed neural networks, which were able to emulate both the data and the physical constraints adequately. In short, all neural network models yielded very good results, with a MAPE less than 4 % and a MAE below 16 kW of power. Moreover, experiments on data efficiency revealed that only a small fraction of the dataset—around 10%—is sufficient for standard neural networks to achieve peak accuracy, although their performance deteriorates with further data reduction. In contrast, PINNs demonstrated greater robustness under limited data conditions, maintaining stable predictive accuracy. Finally, introducing an evidential output layer on

the best performing neural networks provided uncertainty quantification of the predictions, a desirable feature of any physical model. The predicted uncertainties, which proved to be consistent with the absolute error, enabled the definition of confidence intervals in the power curve and a better agreement with the manufacturer specifications.

The deployed models could serve as regression-based anomaly detection methods, comparing the deviation of new data from the prediction of the model for a healthy state. Especially useful are models predicting power and an associated uncertainty, which quantifies the probability that new data correspond to normal operation or anomalies. All the presented models are completely differentiable, so they could be used for building optimal pitch angle controllers, leading to optimal power generation at different wind speed regimes. Although this family of models seems promising, its robustness across different turbines requires further investigation. Nevertheless, the models could be easily transferred to different manufacturers by fine-tuning the parameters of the pre-trained models.

Acknowledments

This work was partially funded by the Spanish Ministry of Economic Affairs and Digital Transformation (NextGenerationEU funds) within the project IA4TES MIA.2021.M04.0008. AG, MMS and JGR were also funded by Junta de Andalucía (D3S project, P21.00247) and MICIU/AEI/10.13039/501100011033 (SINERGY project, PID2021.125537NA.I00). Acknowledgement is extended to ENGIE for providing such an interesting and well-documented sample dataset.

Author contributions

Alfonso Gijón: Conceptualization, Methodology, Software, Validation, Investigation, Writing - Original Draft. **Ainhoa Pujana-Goitia:** Conceptualization, Writing - Review & Editing, **Eugenio Perea:** Conceptualization **Miguel Molina-Solana:** Conceptualization, Writing - Review & Editing, Supervision, Funding acquisition. **Juan Gómez-Romero:** Conceptualization, Writing - Review & Editing, Supervision, Funding acquisition.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve language and readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Appendix A. Physics-Informed Neural Network Architecture

As an example, the PINN architecture built for predicting the power while satisfying the physical equation $P = gT\omega$ is schematized in Figure A.1. The PINN receives (v, β, ω, T) as input, (v, β, ω) being the arguments of an artificial neural network with two hidden dense layers that output the power. Independently, the torque T is combined with ω through lambda layers to compute $gT\omega$. Before the multiplication of ω and T, additional lambda layers were applied for unit conversion (input data was standardized to have zero mean and standard deviation equal to 1) before combining ω and T. Finally, the power in original units is transformed to standard units to be comparable with the output of the artificial neural network. This workflow allows the training of a PINN to minimize both terms of the loss function of Equations 11a and 11b.

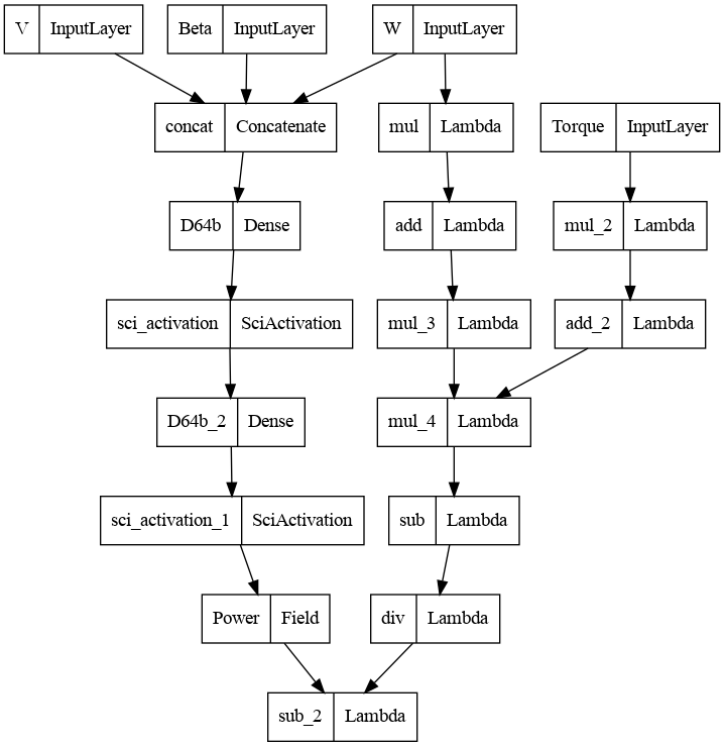


Figure A.1: Model graph of the PINN predicting P as a function of (v, β, ω), while enforcing physical relation $P = gT\omega$.

References

[1] I. R. E. A. IRENA, World energy transitions outlook 2023: 1.5°c pathway, volume 1, international renewable (2023).
URL https://www.irena.org/Publications/2023/Jun/World-Energy-Transitions-Outlook-2023

[2] WindEurope, Offshore wind energy 2022 mid-year statistics (2022).
URL https://windeurope.org/intelligence-platform/product/offshore-wind-energy-2022-mid-year-statistics/

[3] J. Wu, S. Guo, J. Li, D. Zeng, Big data meet green challenges: Big data toward green applications, IEEE Systems Journal 10 (3) (2016) 888–900. doi:10.1109/JSYST.2016.2550530.

[4] H. Akhavan-Hejazi, H. Mohsenian-Rad, Power systems big data analytics: An assessment of paradigm shift barriers and prospects, Energy Reports 4 (2018) 91–100. doi:10.1016/j.egyr.2017.11.002.

[5] R. Sahal, J. G. Breslin, M. I. Ali, Big data and stream processing platforms for industry 4.0 requirements mapping for a predictive maintenance use case, Journal of Manufacturing Systems 54 (2020) 138–151. doi:10.1016/j.jmsy.2019.11.004.

[6] C. Q. Gómez, M. A. Villegas, F. P. García, D. J. Pedregal, Big Data and Web Intelligence for Condition Monitoring, IGI-Global, 2015. doi:10.4018/978-1-4666-8505-5.ch008.

[7] M. Canizo, E. Onieva, A. Conde, S. Charramendieta, S. Trujillo, Real-time predictive maintenance for wind turbines using big data frameworks, in: 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), 2017, pp. 70–77. doi:10.1109/ICPHM.2017.7998308.

[8] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, G. Nenadic, Machine learning methods for wind turbine condition monitoring: A review, Renewable Energy 133 (2019) 620–635. doi:10.1016/j.renene.2018.10.047.

[9] A. Clifton, S. Barber, A. Bray, P. Enevoldsen, J. Fields, A. M. Sempreviva, L. Williams, J. Quick, M. Purdue, P. Totaro, Y. Ding, Grand

challenges in the digitalisation of wind energy, Wind Energy Science 8 (6) (2023) 947—-974. doi:10.5194/wes-8-947-2023.

[10] M. Tan, Z. Zhang, Wind turbine modeling with data-driven methods and radially uniform designs, IEEE Transactions on Industrial Informatics 12 (3) (2016) 1261–1269. doi:10.1109/TII.2016.2532321.

[11] L. D. Avendaño-Valencia, E. N. Chatzi, D. Tcherniak, Gaussian process models for mitigation of operational variability in the structural health monitoring of wind turbines, Mechanical Systems and Signal Processing 142 (2020) 106686. doi:10.1016/j.ymssp.2020.106686.

[12] A. Pujana, M. Esteras, E. Perea, E. Maqueda, P. Calvez, Hybrid-model-based digital twin of the drivetrain of a wind turbine and its application for failure synthetic data generation, Energies 16 (2) (2023). doi:10.3390/en16020861.

[13] M. Mehrjoo, M. Jafari Jozani, M. Pawlak, Wind turbine power curve modeling for reliable power prediction using monotonic regression, Renewable Energy 147 (08 2019). doi:10.1016/j.renene.2019.08.060.

[14] S. Watson, L. Landberg, J. Halliday, Application of wind speed forecasting to the integration of wind energy into a large scale power system, IEE Proceedings - Generation, Transmission and Distribution 141 (1994) 357–362. doi:10.1049/ip-gtd_19941215.

[15] M. Benbouzid, T. Berghout, N. Sarma, S. Djurović, Y. Wu, X. Ma, Intelligent condition monitoring of wind power systems: State of the art review, Energies 14 (18) (2021). doi:10.3390/en14185967.

[16] O. Apata, D. Oyedokun, An overview of control techniques for wind turbine systems, Scientific African 10 (2020) e00566. doi:10.1016/j.sciaf.2020.e00566.

[17] M. F. Howland, J. O. Dabiri, Wind farm modeling with interpretable physics-informed machine learning, Energies 12 (14) (2019). doi:10.3390/en12142716.

[18] M. Raissi, P. Perdikaris, G. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Com-

putational Physics 378 (2019) 686–707. doi:10.1016/j.jcp.2018.10.045.

[19] F. Fernández de la Mata, A. Gijón, M. Molina-Solana, J. Gómez-Romero, Physics-informed neural networks for data-driven simulation: Advantages, limitations, and opportunities, Physica A: Statistical Mechanics and its Applications 610 (2023) 128415. doi:10.1016/j.physa.2022.128415.

[20] G. A. M. van Kuik, J. Peinke, R. Nijssen, D. Lekou, J. Mann, J. N. Sørensen, C. Ferreira, J. W. van Wingerden, D. Schlipf, P. Gebraad, H. Polinder, A. Abrahamsen, G. J. W. van Bussel, J. D. Sørensen, P. Tavner, C. L. Bottasso, M. Muskulus, D. Matha, H. J. Lindeboom, S. Degraer, O. Kramer, S. Lehnhoff, M. Sonnenschein, P. E. Sørensen, R. W. Künneke, P. E. Morthorst, K. Skytte, Long-term research challenges in wind energy - a research agenda by the european academy of wind energy, Wind Energy Science 1 (1) (2016) 1–39. doi:10.5194/wes-1-1-2016.

[21] A. Amini, W. Schwarting, A. Soleimany, D. Rus, Deep evidential regression, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020.
URL https://proceedings.neurips.cc/paper_files/paper/2020/file/aab085461de182608ee9f607f3f7d18f-Paper.pdf

[22] A. Gijón, Windturbines_PINNs, GitHub repository. Accessed: 2025-07-07 (2024).
URL https://github.com/alfonsogijon/WindTurbines_PINNs

[23] N. Zehtabiyan-Rezaie, A. Iosifidis, M. Abkar, Data-driven fluid mechanics of wind farms: A review, Journal of Renewable and Sustainable Energy 14 (3) (2022) 032703. doi:10.1063/5.0091980.

[24] Repower MM82 - Manufacturers and Turbines - Online access - The Wind Power, https://www.thewindpower.net/turbine_en_16_repower_mm82.php, Accessed: 09-05-2023.

[25] T. Burton, N. Jenkins, D. Sharpe, E. Bossanyi, Aerodynamics of Horizontal Axis Wind Turbines, John Wiley & Sons, Ltd, 2011, Ch. 3, pp. 39–136. doi:10.1002/9781119992714.ch3.

[26] I. C. Secretary, Wind turbines—part 12-1: Power performance, measurements of electricity producing wind turbines Part 12-1: Overview (2022).
URL https://webstore.iec.ch/publication/68499

[27] L. C. Pagnini, M. Burlando, M. P. Repetto, Experimental power curve of small-size wind turbines in turbulent urban environment, Applied Energy 154 (2015) 112–121. doi:10.1016/j.apenergy.2015.04.117.

[28] J. Yan, H. Zhang, Y. Liu, S. Han, L. Li, Uncertainty estimation for wind energy conversion by probabilistic wind turbine power curve modelling, Applied Energy 239 (2019) 1356–1370. doi:10.1016/j.apenergy.2019.01.180.

[29] P. G. Jamdade, S. G. Jamdade, Modelling and simulation of mechanical torque developed by wind turbine generator excited with different wind speed profiles, in: V. Vijay, S. K. Yadav, B. Adhikari, H. Seshadri, D. K. Fulwani (Eds.), Systems Thinking Approach for Social Problems, Springer India, New Delhi, 2015, pp. 267–278. doi:10.1007/978-81-322-2141-8_23.

[30] M. Carpintero-Renteria, D. Santos-Martin, A. Lent, C. Ramos, Wind turbine power coefficient models based on neural networks and polynomial fitting, IET renewable power generation. 14 (11) (2020-08). doi:10.1049/iet-rpg.2019.1162.

[31] O. C. Castillo, V. R. Andrade, J. J. R. Rivas, R. O. González, Comparison of power coefficients in wind turbines considering the tip speed ratio and blade pitch angle, Energies 16 (6) (2023). doi:10.3390/en16062774.

[32] Y. Wang, X. Duan, R. Zou, F. Zhang, Y. Li, Q. Hu, A novel data-driven deep learning approach for wind turbine power curve modeling, Energy 270 (2023) 126908. doi:10.1016/j.energy.2023.126908.

[33] M. Ma, Smart grid: Energy harvesting, accessed: 2025-07-07 (2023).
URL https://www.linkedin.com/pulse/smart-grid-energy-harvesting-martin-ma-mba-med-gdm-scpm-pmp/

[34] J. Zhang, M. Cheng, Z. Chen, X. Fu, Pitch angle control for variable speed wind turbines, in: 2008 Third International Conference on Electric

Utility Deregulation and Restructuring and Power Technologies, 2008, pp. 2691–2696. doi:10.1109/DRPT.2008.4523867.

[35] M. S. Abdul-Ruhman, M. N. Hawas, Optimal pitch angle control for wind turbine using intelligent controller, IOP Conference Series: Materials Science and Engineering 745 (1) (2020) 012017. doi:10.1088/1757-899X/745/1/012017.

[36] J. E. Sierra-Garcia, M. Santos, Deep learning and fuzzy logic to implement a hybrid wind turbine pitch control, Neural Computing and Applications 34 (13) (2022) 10503–10517. doi:10.1007/s00521-021-06323-w.

[37] A. Karami-Mollaee, O. Barambones, Pitch control of wind turbine blades using fractional particle swarm optimization, Axioms 12 (1) (2023). doi:10.3390/axioms12010025.

[38] F. Bilendo, H. Badihi, N. Lu, P. Cambron, B. Jiang, An intelligent data-driven machine learning approach for fault detection of wind turbines, in: 2021 6th International Conference on Power and Renewable Energy (ICPRE), 2021, pp. 444–449. doi:10.1109/ICPRE52634.2021.9635340.

[39] L. Sun, F. You, Machine learning and data-driven techniques for the control of smart power generation systems: An uncertainty handling perspective, Engineering 7 (9) (2021) 1239–1247. doi:10.1016/j.eng.2021.04.020.

[40] B. Manobel, F. Sehnke, J. A. Lazzús, I. Salfate, M. Felder, S. Montecinos, Wind turbine power curve modeling based on gaussian processes and artificial neural networks, Renewable Energy 125 (2018) 1015–1020. doi:10.1016/j.renene.2018.02.081.

[41] T. Li, X. Liu, Z. Lin, R. Morrison, Ensemble offshore wind turbine power curve modelling – an integration of isolation forest, fast radial basis function neural network, and metaheuristic algorithm, Energy 239 (2022) 122340. doi:10.1016/j.energy.2021.122340.

[42] J. E. Sierra-García, M. Santos, Neural networks and reinforcement learning in wind turbine control, Revista Iberoamericana de Automática e In-

formática industrial 18 (4) (2021) 327–335. doi:10.4995/riai.2021.16111.

[43] J. E. Sierra-Garcia, M. Santos, R. Pandit, Wind turbine pitch reinforcement learning control improved by pid regulator and learning observer, Engineering Applications of Artificial Intelligence 111 (2022) 104769. doi:10.1016/j.engappai.2022.104769.

[44] M. Coquelet, L. Bricteux, M. Moens, P. Chatelain, A reinforcement-learning approach for individual pitch control, Wind Energy 25 (8) (2022) 1343–1362. doi:10.1002/we.2734.

[45] J. Xie, H. Dong, X. Zhao, Data-driven torque and pitch control of wind turbines via reinforcement learning, Renewable Energy 215 (2023) 118893. doi:10.1016/j.renene.2023.06.014.

[46] X. Li, W. Zhang, Physics-informed deep learning model in wind turbine response prediction, Renewable Energy 185 (2022) 932–944. doi:10.1016/j.renene.2021.12.058.

[47] J. H. Mclean, M. R. Jones, B. J. O'Connell, E. Maguire, T. J. Rogers, Physically meaningful uncertainty quantification in probabilistic wind turbine power curve models as a damage-sensitive feature, Structural Health Monitoring 0 (0) (2023) 14759217231155379. doi:10.1177/14759217231155379.

[48] F. Aerts, L. Lanzilao, J. Meyers, Bayesian uncertainty quantification framework for wake model calibration and validation with historical wind farm power data, Wind Energy 26 (8) (2023) 786–802. doi:10.1002/we.2841.

[49] K. Marepally, Y. S. Jung, J. Baeder, G. Vijayakumar, Uncertainty quantification of wind turbine airfoil aerodynamics with geometric uncertainty, Journal of Physics: Conference Series 2265 (4) (2022) 042041. doi:10.1088/1742-6596/2265/4/042041.

[50] P. Richter, J. Wolters, M. Frank, Uncertainty quantification of offshore wind farms using monte carlo and sparse grid, Energy Sources, Part B: Economics, Planning, and Policy 17 (1) (2022) 2000520. doi:10.1080/15567249.2021.2000520.

[51] J. Perr-Sauer, M. Optis, J. M. Fields, N. Bodini, J. C. Lee, A. Todd, E. Simley, R. Hammond, C. Phillips, M. Lunacek, T. Kemper, L. Williams, A. Craig, N. Agarwal, S. Sheng, J. Meissner, Openoa: An open-source codebase for operational analysis of wind farms, Journal of Open Source Software 6 (58) (2021) 2171. doi:10.21105/joss.02171.

[52] R. Mandzhieva, R. Subhankulova, Data-driven applications for wind energy analysis and prediction: The case of "la haute borne" wind farm, Digital Chemical Engineering 4 (2022) 100048. doi:10.1016/j.dche.2022.100048.

[53] N. Effenberger, N. Ludwig, A collection and categorization of open-source wind and wind power datasets, Wind Energy 25 (10) (2022) 1659–1683. doi:10.1002/we.2766.

[54] D. Minh-Thang, J. Berthaut-Gerentes, An estimation of key metrics from scada data - a third-party's view, in: Poster presented at Analysis of Operating Wind Farms conference, 2018.

[55] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems (2015).

[56] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, A. Talwalkar, Hyperband: A novel bandit-based approach to hyperparameter optimization, J. Mach. Learn. Res. 18 (1) (2017) 6765–6816.
URL https://jmlr.org/papers/v18/16-558.html

[57] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, et al., Kerastuner, https://github.com/keras-team/keras-tuner (2019).

[58] E. Haghighat, R. Juanes, Sciann: A keras/tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks, Computer Methods in Applied Mechanics and Engineering 373 (2020) 113552–113552. doi:10.1016/j.cma.2020.113552.