DINAME 2023 - Proceedings of the XIX International Symposium on Dynamic Problems of Mechanics,
A.S. De Paula, A.T. Fabro and M.R. Machado (Editors), ABCM.
Pirenópolis, Go, Brazil, February 26th to March 3rd, 2023

A Physics-Based Machine Learning Model for Fatigue in Wind Turbines Devices

Tiago Bastos Moscon Micco Puntel[1], **Fernando Alves Rochinha**[1]

[1] Mechanical Engineering - Universidade Federal do Rio de Janeiro

Abstract: This paper aims, using a deep generative model, which can be viewed as a probabilistic surrogate, to build a digital representation component of a wind turbine that assesses bearing accumulated fatigue damage for real-time application, considering lubricant condition and 10-minute SCADA summary statistics: mean wind speed at the hub height and turbulence intensity. The stochastic generator creates a set of distinct wind flow samples to account for the uncertainty in the flow characterization. The aero-servo-elastic simulator provides the load time series that will be evaluated in a post-processing routine to give the accumulated damage distribution for each input. The analysis shows that The deep generative model performs well in capturing the statistical quantities, such as the mean and quantiles. However, the resulting distributions of some inputs in the domain edges and low-density regions have discrepant standard deviation values from the data distributions, indicating that the model fails to fit them adequately in these areas. Finally, this work presents application examples for the developed digital representation.

Keywords: wind turbine, bearing damage, machine learning, deep generative models, stochastic systems

INTRODUCTION

During the last decades, wind energy evolution has happened thanks to the employment of new technologies that allowed better turbine designs and efficient operation. The digital transformation process opens a novel front for new product development that uses data acquisition from sensors, computational simulations, and artificial intelligence algorithms. Recently, researchers and engineers have conceived tools to increase the operational efficiency of wind farms, enhancing power generation and reducing the cost of producing electricity. Almeida and Rochinha (2022) studied a machine learning algorithm that replaced the turbine wake model and allowed computational cost reduction in the optimization problem of the yaw steering maneuver. Dong and Zhao (2022) proposed a yaw angle control technic through deep reinforcement learning to mitigate wake effects in a wind farm and enhance energy production.

Wind turbines are large-scale equipment encompassing flexible blades and mechanical devices operating under harsh conditions. During their lifetime, they are exposed to damage and fatigue due to fast and turbulent flow. Damage estimation and residual functional life models can be critical for wind farm management, especially for maintenance planning, energy cost estimation due to turbine depreciation, and future use in vanguard technology concepts such as digital twins (Ritto and Rochinha, 2021). Furthermore, the diversity of operating scenarios and real-time control require fast and low-cost models. Hirvoas (2021) presented a framework to reduce and quantify uncertainty, considering the fatigue damage in different turbine components as the quantities of interest. Myllonas, Abdallah and Chatzi (2021) forged a methodology to assess wind turbine blade fatigue using data from the acquisition system and a Machine Learning algorithm. Motivated by the digital transformation age, the present work aims to develop a proof of concept model to estimate main-bearing accumulated fatigue damage using summary statistics provided by the data acquisition system of wind turbines SCADA. The present method can be applied to component design and real-time damage assessment and can be seen as a part of a digital representation model.

We assess the damage considering three inputs: the mean wind speed at hub height, the turbulence intensity, and the lubricant condition. The first two are numerical inputs, while the third is categorical. In order to estimate the damage, the structure load time histories must be known. Therefore, we perform simulations using open-source software to produce the wind flow and calculate the dynamic response. Even simplified wind generators combined with aero-servo-elastic simulators are not suitable for probabilistic computations due to the need for a large sample number to characterize the stochastic response. Surrogate models should be chosen to replace them. We select a Machine Learning algorithm within the supervised learning realm. Due to the stochastic nature of the generated turbulent flows and the associated diversity of scenarios, a Generative Adversarial Network (GAN) (Yang and Perdikaris, 2019) is employed.

The remainder of this paper is as follows. The second section presents more details regarded to the data generation process and the selected Machine Learning model. The third section presents the validation procedure and results. In the fourth section, we demonstrate application examples. Finally, we exhibit the conclusion in the fifth section.

A GAN FOR DAMAGE ESTIMATION IN WIND TURBINES

Data generation, damage modeling and probabilistic surrogate

In order to develop damage estimation algorithms is crucial to calculate wind turbine loads that depend on a coupled effect of aerodynamics, dynamics, and control system response. The wind turbine computational model has, as inputs, flow characteristics (ten minutes average of velocity, turbulence intensity, and direction) surrounding the structure at each spatial point and time instance. Unfortunately, it is impossible to measure them for all positions, resulting in a lack of information in wind description. A stochastic wind generator must be employed to represent the flow in the turbine plane, with its parameters adjusted beforehand to create a plausible and accurate velocity field for a specific site. This strategy uses flow summary statistics, measured on turbine hub height, as inputs for these synthetic wind algorithms. The approach is probabilistic because the generator creates n distinct wind fields for each condition, resulting in a damage variability, i.e., two wind time series with the same inputs have different fatigue damage values.

Data generation follows the schematic view presented in Fig. 1. In order to create the wind speed time series in the turbine rotational plane, the open-source software called TurbSim is used with the Kaimal turbulence spectrum (Kaimal et al., 1972). OpenFAST performs the aero-servo-elastic simulations. The National Renewable Energy Laboratory software has a wind turbine computational model that uses multibody dynamics combined with tower and blade flexible elements. The offshore and the onshore turbine have 24 and 18 DOF, respectively. The aerodynamic module uses Blade Element Momentum theory with factors to account for some disregarded flow effects (NREL, 2022)

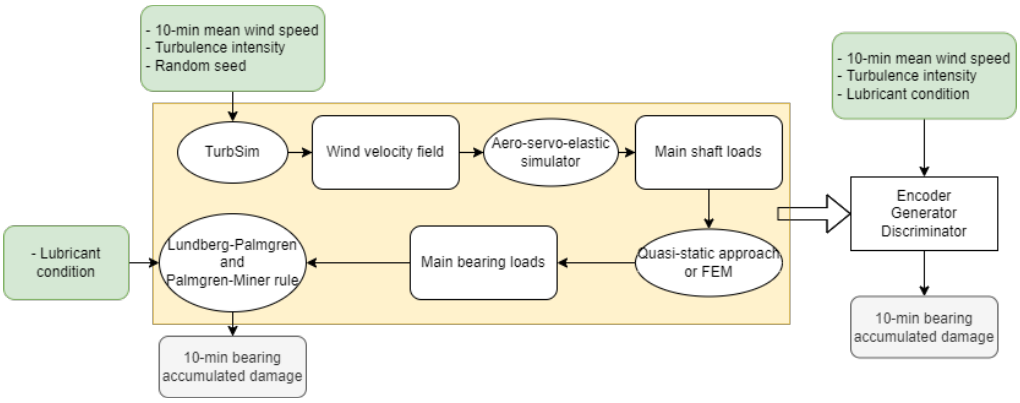

Figure 1 – Large-scale simulation procedure.

The OpenFAST raw output consists of the forces and the moments time series on the turbine hub. A post-processing routine, which considers a quasi-static approach for the shaft, computes the equivalent bearing load time series using the OpenFAST outputs. In order to assess bearing fatigue life, this work uses the probabilistic engineering models from ISO 281:2007 (2007), which is based on the theory of Lundberg and Palmgren with life modifying factors, one of which considers the lubricant condition. However, this model cannot be applied directly due to non-stationary equivalent load. Therefore, the Palmgren Miner rule (Miner, 1945), which considers the linear damage hypothesis, is employed. In short, the algorithm needs to calculate the damage for each shaft rotation increment $\Delta\phi$ and sum all over 10 minutes, see Eq. (1).

$$D_{10min} = \sum_{i=1}^{n-1} \frac{\phi_{i+1} - \phi_i}{2\pi L_i}; \tag{1}$$

Where ϕ is the azimuth angle (rad), L_i is the fatigue life for the mean equivalent load of ϕ interval (in number of revolutions), see Eq. (2).

$$L_i = a_1 a_{ISO} \left(\frac{2C_D}{F_{eq}(\phi_{i+1}) + F_{eq}(\phi_i)} \right)^{(10/3)} \tag{2}$$

Where C_D is the dynamic capacity and F_{eq} is the equivalent load.

In favor of having a better representation, we divided the damage values by the reference, defined as the one that gives a 20 years life operation under the normal lubricant condition for 90% of bearings. If these components work under a damage value higher than 1, they will not reach the minimum design life of 20 years established by IEC 61400-1 (2014).

The simulation outcomes provide an empirical and unknown joint distribution $q(x,y)$, which converges to the true one $p(x,y)$ as long as the data set increases. A probabilistic approach tries to approximate $p(y|x)$ to $q(y|x)$ for each x, where x, in this case, is the wind and lubricant condition, and y is the set of bearing fatigue damage related to them. The deep generative model from Yang and Perdikaris (2019) tries to conditionally generate samples similar to $q(y|x)$ through a deterministic function (neural network). However, as x has multiple y values, a latent space z must be introduced in order to account for the stochastic data dimension. In the training process, the generator, which receives the input x and samples from a fixed prior distribution in the latent space $z \sim p(z)$, learns to map $q(y|x)$ from x and z. If the training succeeds, $G_\theta(x,z)$ can describe the empirical joint distribution. Moreover, if $q(y|x)$ approximates the true and unknown joint distribution, $G_\theta(x,y)$ can represent the stochastic system and capture the complete statistical dependence of y on x.

This machine learning algorithm does not rely on any hypothesis beforehand. Thus, in principle, it can learn complex distribution patterns from noisy training data. Nonetheless, it lacks mathematical representation since the learning process only adjusts its parameters to generate new samples that try to mimic the data (Creswell et al., 2018). The model needs two other neural networks to execute the training step: the discriminator $D_\psi(x,y)$ and the encoder $E_\phi(x,y)$, as depicted in Fig. 2. The former tries to distinguish between the samples obtained from the generator and the training data. The latter is the regularization term of the generator loss function, being an upper bound of the conditional entropy $H(z|x,y)$ since it provides cycle consistency (Li, 2018). Moreover, Yang and Perdikaris (2019) showed that this term stimulates the sample scattering throughout the training data support, preventing GANs from an issue called mode collapse (Goodfellow et al., 2014). GAN uses backpropagation and gradient descent algorithms which are more effective than those that use Markov Chain Monte Carlo, such as Generative Stochastic Networks, and do not tend to present scalability problems.

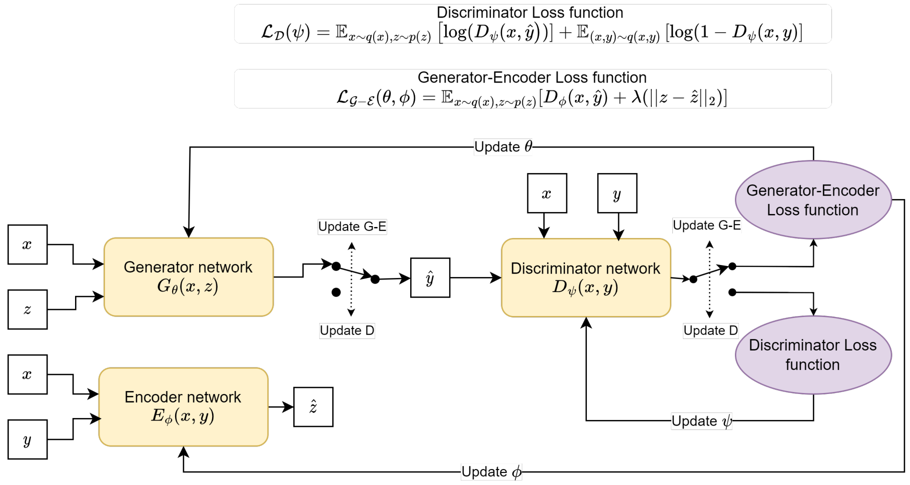
Discriminator Loss function
$\mathcal{L}_{\mathcal{D}}(\psi) = \mathbb{E}_{x\sim q(x),z\sim p(z)}\left[\log(D_\psi(x,\hat{y}))\right] + \mathbb{E}_{(x,y)\sim q(x,y)}\left[\log(1 - D_\psi(x,y)\right]$
Generator-Encoder Loss function
$\mathcal{L}_{\mathcal{G}-\mathcal{E}}(\theta,\phi) = \mathbb{E}_{x\sim q(x),z\sim p(z)}[D_\phi(x,\hat{y}) + \lambda(||z-\hat{z}||_2)]$
Update θ
x
y
Generator-Encoder Loss function
Update G-E
Update G-E
x
z
Generator network $G_\theta(x,z)$
$\hat{y}$
Discriminator network $D_\psi(x,y)$
Update D
Update D
Discriminator Loss function
x
y
Encoder network $E_\phi(x,y)$
$\hat{z}$
Update ψ
Update ϕ

Figure 2 – Neural network training scheme. Yang and Perdikaris, 2019.

Case study description

The simulations consider two numerical and one categorical input due to limited computational resources (one Intel Core i7-9700). The former, supplied by SCADA, consists of 10 minutes mean wind speed at hub height and turbulence intensity. The latter represents the lubricant contamination condition: ultra-clean, clean, normal, severe, and ultra-severe. There is no sensor to assess the real-time grease state. Therefore, we can employ a data-driven approach that estimates lubricant degradation, such as the proposal of Yucesan and Viana (2020), or use this variable to evaluate the damage in different scenarios. Table 1 shows the variable bounds assumed in this work.

Table 1 – Variable and variable bounds for the simulations.

Variable	Lower Bound	Upper Bound
Wind speed (u) $[m/s]$	3	16
Turbulence intensity (TI) $[\%]$	2.5	$\frac{18}{u}(6.8+0.75u+3(\frac{10}{u})^2)$, Dimitrov et al. (2015).
Lubricant condition (lc), [categorical variable]	1 (ultra-clean)	5 (ultra-severe)

We generate a data set with 206 wind conditions: 10 minutes mean speed at hub height and turbulence intensity through the Quasi-Monte Carlo sample technic, with five lubricant states. Each input, i.e. (u, TI, lc), has 200 damage values. Figure 3 depicts the domain. We split the data into 180 points for training and 26 for testing. The training process is challenging because it encompasses three neural networks. First, we select the architecture through a trial-error approach with some constraints:

1. The neural networks should not have more than five hidden layers;
2. The number of neurons should be 2^n with n greater than two and less than eight;
3. The discriminator should have one layer less than the generator and encoder networks (Yang and Perdikaris, 2019);
4. The generator and the encoder networks should have a mirrored or symmetric structure.

Before that, we perform a hyperparameter optimization using the Optuna framework, introduced by Akiba et al. (2019), to find the others, such as the learning rate, the batch size, the regularization constant, and the number of generator updates per discriminator updates. Optuna uses Bayesian optimization approaches to find the best hyperparameters set. Based on the last metric values, it suggests new trials to improve it. We use the Tree-structured Parzen (Bergstra, Bardenet, Bengio and Kégl, 2011) as the sample algorithm and the Wasserstein distance (W-dist) (Ramdas, Garcia and Cuturi, 2015) as the value that Optuna tries to minimize. The obtained model should be sub-optimal because we first complete a trial-error approach to find the architectures and then execute the hyperparameters optimization. Optuna could handle the combined process, although we decided to split it due to limited computation resources. We opt to use two learning rates, one to accelerate the process in early epochs and the other to stabilize the loss in the end. The following section will demonstrate that the result was satisfactory. Table 2 shows the selected architecture. Table 3 depicts the hyperparameters used in the trial-error approach and the obtained one after the optimization.

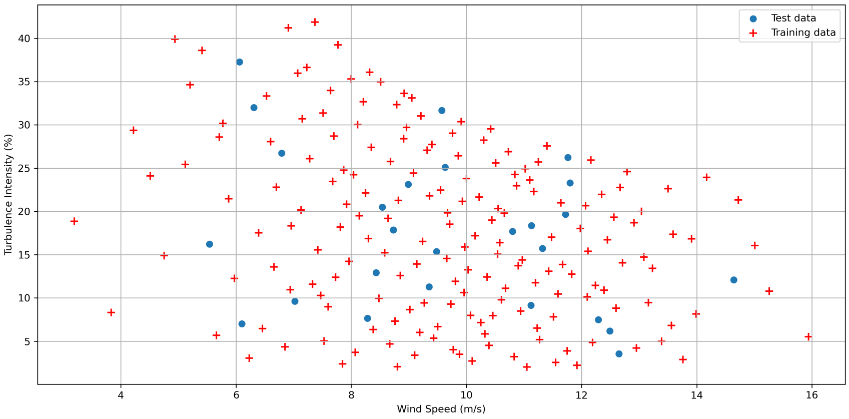

Figure 3 – Data domain, 206 wind conditions: 180 for training and 26 for testing.

Table 2 – Neural Network architectures.

Neural Networks	Generator $G_\theta(x,z)$	Encoder $E_\phi(x,y)$	Discriminator $D_\psi(x,y)$
Architecture	[4,32,32,64,32,1]	[4,32,64,32,32,1]	[4,32,32,32,1]
Activation function	tanh	tanh	tanh and sigmoid (last-layers)

Table 3 – Training parameters. 200 Optimization trails. Optimized parameters (*).

Optuna Optimization	before	after
Optimization Algorithm	ADAM	ADAM
Learning rate 1 (*)	$2.0 \cdot 10^{-4}$	$8.53 \cdot 10^{-4}$
Learning rate 2 (*)	-	$8.00 \cdot 10^{-5}$
Number of G updates per D updates (*)	2	4
Epochs	$2.0 \cdot 10^5$	$1.2 \cdot 10^5$ (lr1) + $1.2 \cdot 10^5$ (lr2)
λ-hyperparameter (*)	10	3.12
Batch size (*)	64	256
W-dist (scaled damage)	$1.2 \cdot 10^{-2}$	$7.4 \cdot 10^{-3}$

RESULTS AND DISCUSSION

In order to assess the model result, we analyze some graphics of the training and test data, considering the normal lubricant contamination condition. For each input, we compute 5000 samples to stabilize the statistical quantities of the generated damage distribution. Figures 4 and 5 show the assessment of the model confidence interval for the test and training data. Visually, the confidence intervals and the means match most input points, revealing that the model can deal with data heteroscedasticity in high and low variance damage regions, besides generating 95% of samples inside the true data interval.

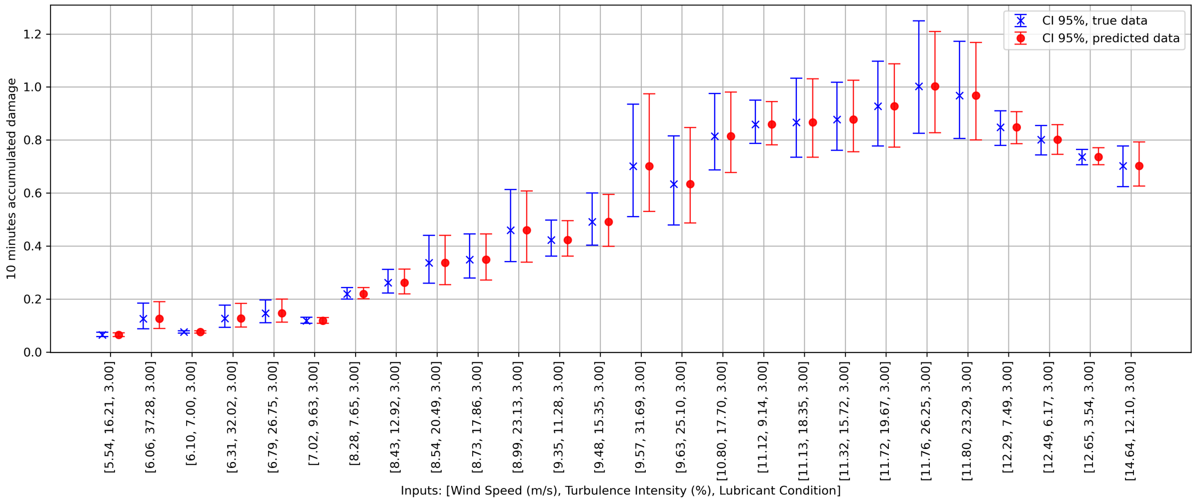

Figure 4 – Assessment of the model confidence interval for the test data (26 points) with normal lubricant condition (3). True data: blue (left), generated data: red (right).

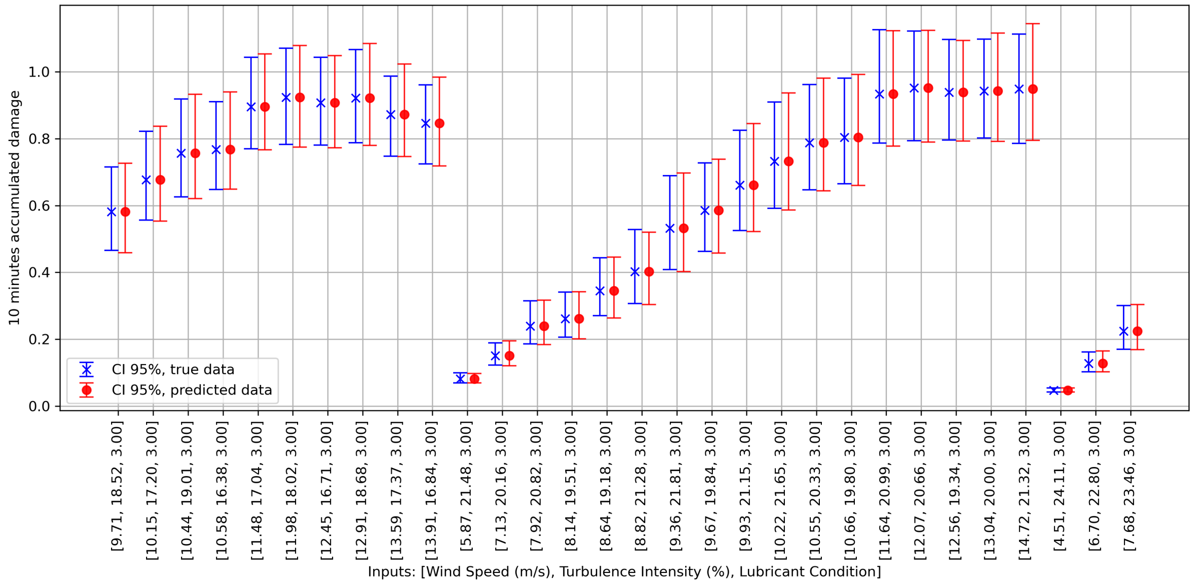

Figure 5 – Assessment of model confidence interval for 30 training data with normal lubricant condition (3). True data: blue (left), generated data: red (right).

In figures 6 and 7, we observe statistical quantities values, such as the mean, standard deviation, quantile 2.5%, and 97.5% for the test and 30 training data. The model nicely predicts the mean and quantile values, but it has problems with the standard deviation, mainly in low-density training data regions and domain edges. The model tends to underestimate the damage variance at a low and high turbulence intensity edge. Figure 7 illustrates that the model tends to overestimate the variance at middle-range turbulence values, i.e., high-density training data regions, as depicted in Fig. 3. These results may change for different architectures and hyperparameters.

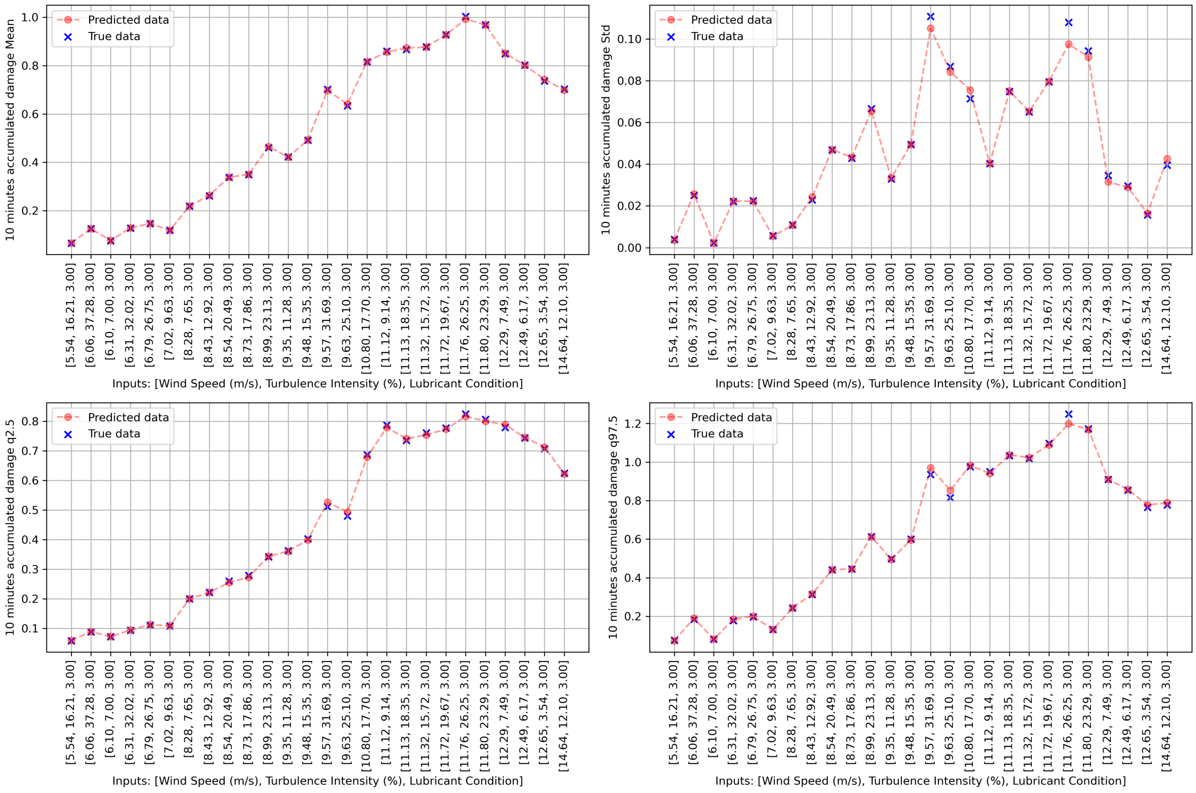

Figure 6 – Summary statistics for the test data (26 points) with normal lubricant condition (3). True data: blue dots, generated data: red x.

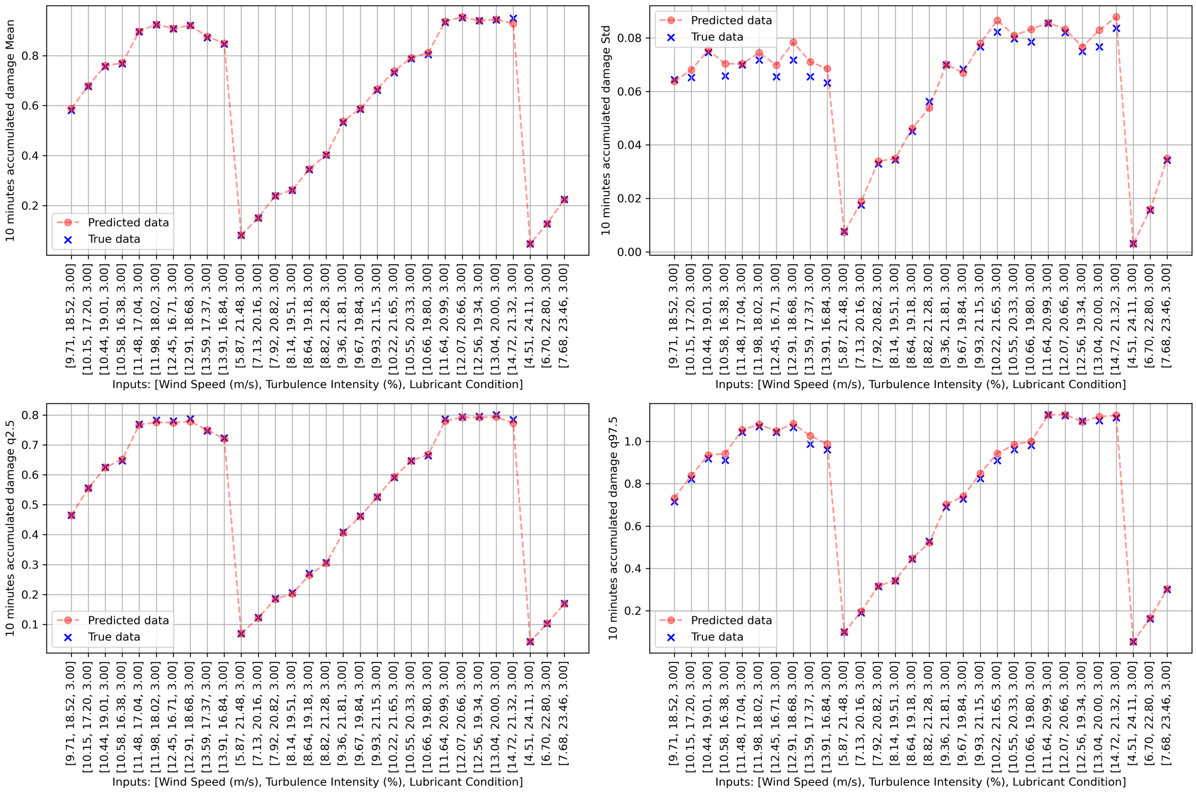

Figure 7 – Summary statistics for 30 training data with normal lubricant condition (3). True data: blue dots, generated data: red x.

Figures 8 and 9 show histograms of the relative error of the mean, standard deviation, and quantiles 2.5%, 25.0%, 75.0%, and 97.5% for the test and training data, respectively. The test data presents the relative error of the mean as less than ±1.2%. The quantile error is less than ±4.2%. The standard deviation has values greater than ±7.5%, indicating that the model fails to fit the distribution shape for some input points. The training data presents the relative error of the mean as less than ±2.0%. The quantile error is less than ±6.3%. The standard deviation has values greater than ±10% for some points. Most of them are located in low-density data regions.

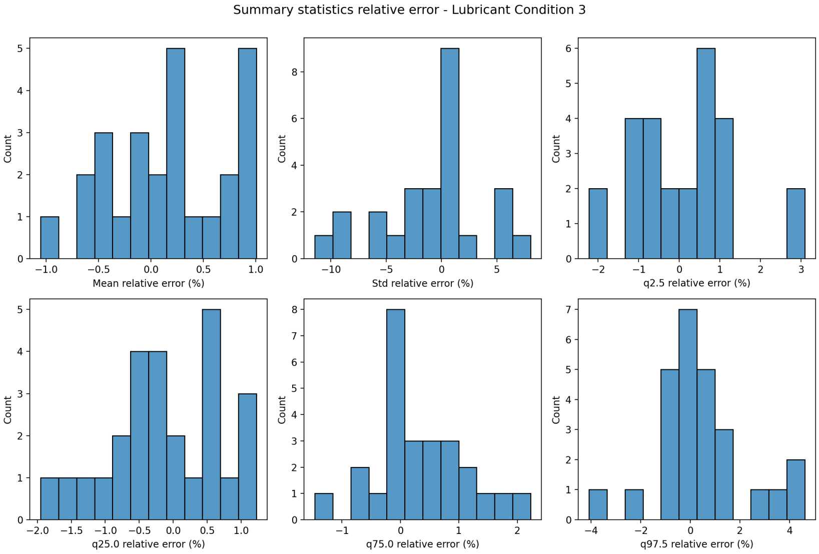

Figure 8 – Relative error summary statistic histograms for test data with normal-typical lubricant condition (3).

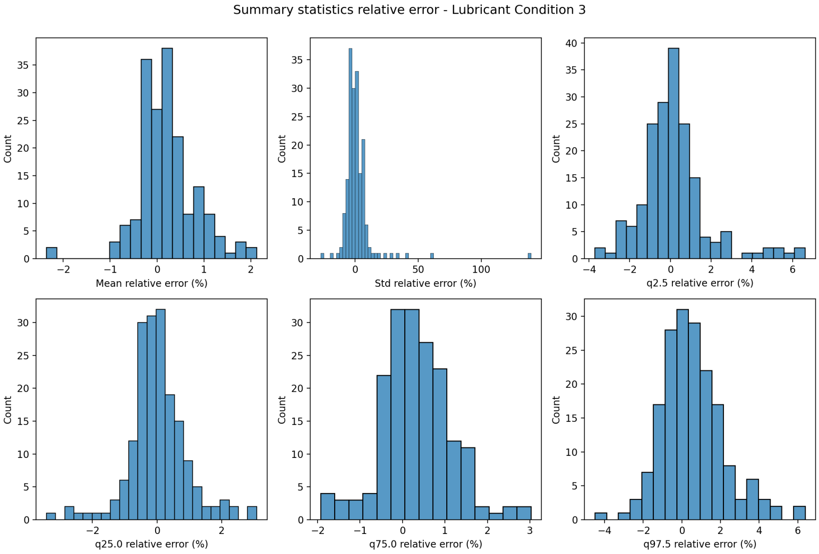

Figure 9 – Relative error summary statistic histograms for test data with normal lubricant condition (3).

The presented graphics show a detailed description of the normal lubricant contamination condition. This work condenses the result for the others by assessing the strip plot in Fig. 10 and Fig. 11. The model predicts the mean and the quantiles satisfactorily. However, the resulting distributions for some points located in the domain edges and low-density regions have discrepant standard deviation values, failing to fit the empirical data. The model is probably suboptimal due to the applied optimization strategy and constrained neural network architecture evaluation. Under these circumstances, the result is adequate. We can increase its performance by assessing more architectures or adding more data for the training despite enhancing the computation time.

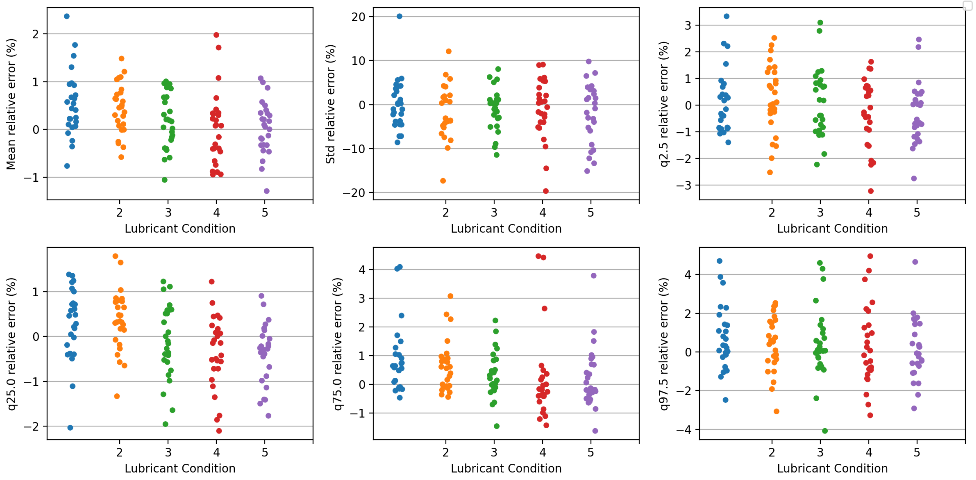

Figure 10 – Relative error strip plot for the mean, standard deviation, quantiles 2.5%, 25%, 75% and 97.5%. Test data (26 points).

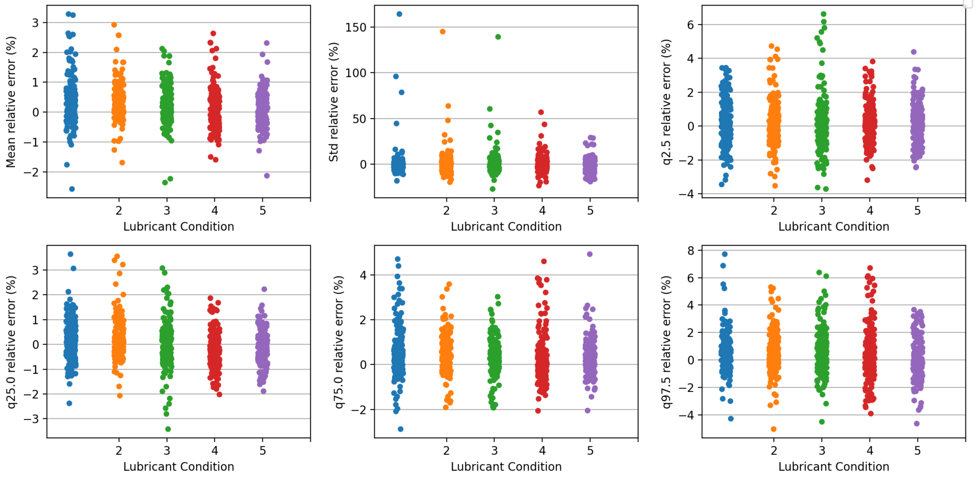

Figure 11 – Relative error strip plot for the mean, standard deviation, quantiles 2.5%, 25%, 75% and 97.5%. Training data (180 points).

APPLICATION EXAMPLES

In this section, we present two applications for this model. First, we perform the calculation for the design, considering wind conditions data for a generic location. Second, we show how we employ this tool in real-time fatigue damage assessment and exhibit the potential for use in wind farm management strategies.

The bearing design procedure generally uses the binning approach to estimate the long-term damage. Each speed bin has a probability of occurrence that weights the short-term damage. In other words, this process is a weighted sum and considers only the wind velocity in specific turbulent intensity classes provided by the IEC 61400-1 (2014) standard. With the trained model, we can apply this approach to the input joint probability density at a low cost since it involves numerical simulations for each bin. We can flexibly change the bin intervals without running extra simulations because the model is capable of extrapolating from the initial training data set. In addition, the binning strategy does not account for the uncertainty in the flow pattern since it runs a few numbers of realizations per input points. Finally, we can use our model without the binning procedure if we construct a probability model for the input joint distribution. We compute the estimated bearing life regarding the joint probability distribution of inputs depicted in Fig. 12 with the normal and severe lubricant contamination conditions. In the first case, the bearing life is approximately 27 years for the quantile 97.5%, 30 years for the mean, and 34 years for the quantile 2.5%. In the second, the bearing life is approximately 18 years for the quantile 97.5%, 20 years for the mean, and 22 years for the quantile 2.5%.

In the following application, the model can be an element of decision-making algorithms. Let’s consider fatigue damage greater than 1,0 as severe since, in the long run, these values reduce the estimated bearing life below 20 years. In

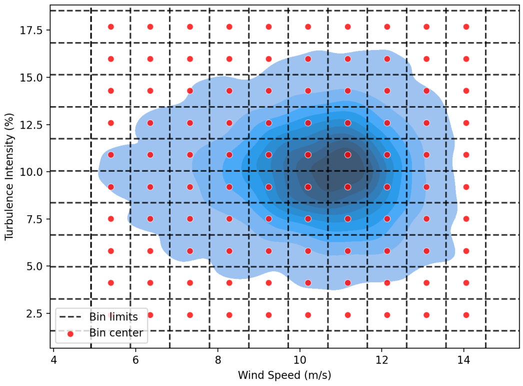

Figure 12 – Joint probability distribution of inputs for a generic site.

this case, our model can provide a damage probability density function after receiving the data from the acquisition system. Based on the energy grid demand, the algorithm can decide to continue the operation, suggesting a higher electricity price to compensate for working under this condition. Moreover, it can propose a control strategy approach to reduce the damage. However, in this case, the model should have more inputs, such as the yaw and blade pitch angle, and assess damage in other components. This should be a topic in further studies. For every 10 minutes, the model receives noisy data from SCADA (Fig. 13 - top), computes the short-term fatigue damage (Fig. 13 - bottom-left), and the probability of having a damage value higher than 1 (Fig. 13 - bottom-right). In this example, the turbine has a possibility of working under severe conditions for approximately 150 minutes. A decision-making algorithm can use this information to support the wind farm operation and the financial strategies in the energy market.

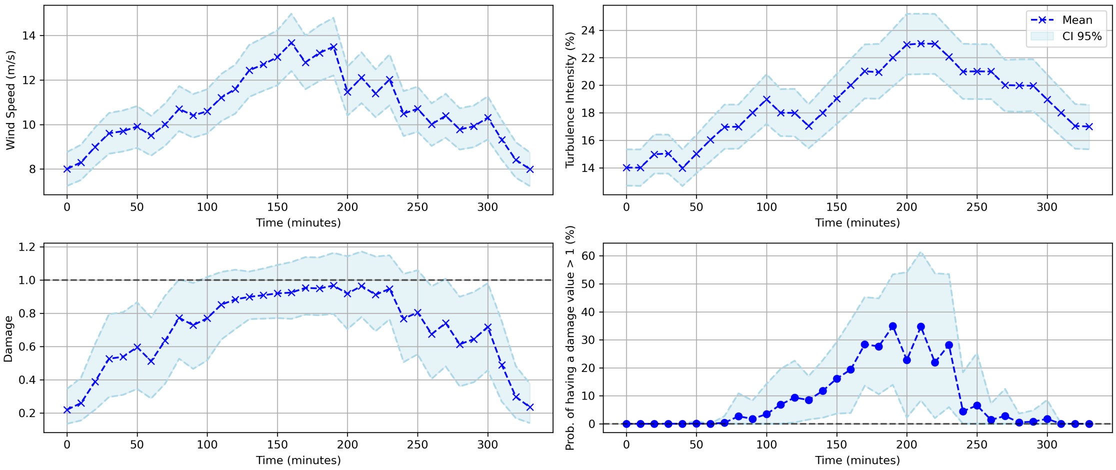

Figure 13 – Real-time bearing fatigue damage assessment considering uncertainty in SCADA measures. Normal lubricant condition (3). Top left: wind speed, top right: turbulence intensity, bottom left: 10 minutes accumulated damage, bottom right: probability of having a damage value higher than 1.

Finally, SCADA data history can be used to compare the accumulated damage between wind farm turbines that face different wind conditions due to terrain characteristics (Watanabe et al., 2015) and wake effects (Shaler et al., 2022), providing data to support preventive maintenance procedure plans.

CONCLUSION

This paper presented a procedure to construct an element of a digital representation of wind turbine components that estimates fatigue damage. It evaluates the model performance and illustrates applications in the design phase, operation, and maintenance. The data generation cost has a substantial influence on the deployment time. The process took approximately 28 days, using parallel processing with six simultaneous simulations. The hyperparameter optimization approach was executed with 200 trials. Each one takes 40 minutes. In total, this step took two days to conclude with parallelization.

The resulting model has a lower computation cost than the numerical one. For instance, it can generate 5000 samples in nearly ten milliseconds. The model handles numerical and categorical variables, damage heteroscedasticity, and can predict statistical quantities such as the mean and the quantiles. However, it has problems with the standard deviation for some inputs, especially the ones located in low-density data regions and the domain edges, indicating difficulties in adjusting the distributions. Probably, we can find a better model considering less constrained architectures and the combined optimization process in Optuna. Despite using fundamental inputs for bearing fatigue damage, this work does not assess other useful ones, such as wind shear, yaw maneuver angle, and lubricant temperature. Further studies should consider building an advanced model.

The deep generative model introduced by Yang and Perdikaris (2019) is a feasible option when considering real-time use for replacing stochastic systems. Nevertheless, the training process is not simple and needs computation resources. Despite the long deployment time, which includes the data generation and the training step, a multiprocessor computation, such as cloud computing, should be used in order to enhance the processing power.

ACKNOWLEDGMENTS

This study was financed by CAPES, Brasil Finance Code 001.

REFERENCES

Almeida, J.O., and Rochinha, F.A., 2022, ”A Probabilistic Learning Approach Applied to the Optimization of Wake Steering in Wind Farms. Journal of Computing and Information Science in Engineering”, 23(1), 011003.

Dong, H., and Zhao, X., 2022, ”Intelligent Wind Farm Control via Grouping-Based Reinforcement Learning”, European Control Conference (ECC), pp. 993-998,

Ritto, T.G., and Rochinha, F.A., 2021, ”Digital twin, physics-based model, and machine learning applied to damage detection in structures”, Mechanical Systems and Signal Processing, 155, 107614.

Hirvoas, A., 2021, ”Développement d’une méthode d’assimilation de données pour la calibration et la mise à jour en continu de modèles fidèles d’éoliennes”, Université Grenoble Alpes.

Mylonas, C., Abdallah, I., and Chatzi, E., 2021, ”Conditional variational autoencoders for probabilistic wind turbine blade fatigue estimation using Supervisory, Control, and Data Acquisition data”, Wind Energy - Wiley, 25(10), 1122-1139.

Yang, Y., and Perdikaris, P., 2019, ”Conditional deep surrogate models for stochastic, high-dimensional, and multi-fidelity systems”, Comput Mech, 64, 417–434.

Kaimal, J.C., Wyngaard, J.C.J., Izumi, Y. and Coté, O. R., 1972, ”Spectral characteristics of surface-layer turbulence”, Quarterly Journal of the Royal Meteorological Society, 98(417), 563-589.

NREL, 2022, ”OpenFAST Documentation”, National Renewable Energy Laboratory.

ISO, 2007, ”Rolling bearings — Dynamic load ratings and rating life”, International Organization for Standardization.

Miner, M.A., 1945, ”Cumulative damage in fatigue”, Journal of Applied Mechanics, 3, 159-164.

IEC, 2014, ”IEC 61400-1”, Ed. 3.1, International Electrotechnical Commission.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A.A.,2018, ”Generative Adversarial Networks: An Overview”, in IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65, doi: 10.1109/MSP.2017.2765202.

Li, C., 2018, ”Towards Better Representations with Deep/Bayesian Learning”. Duke University.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. and Bengio, Y., 2014, ”Generative adversarial nets”, Advances in neural information processing systems, 27.

Yucesan, Y.A., and Viana, F.A., 2020, ”A physics-informed neural network for wind turbine main bearing fatigue”, International Journal of Prognostics and Health Management, 11(1).

Dimitrov, N., Kelly, M.C., Vignaroli, A., and Berg, J., 2018, ”From wind to loads: wind turbine site-specific load estimation with surrogate models trained on high-fidelity load databases”, Wind Energy Science, 3(2), 767-790.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M., 2019, ”Optuna: A Next-generation Hyperparameter Optimization Framework”, arXiv.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B., 2011, ”Algorithms for Hyper-Parameter Optimization”, Curran Associates Inc.

Ramdas, A., Garcia, N., and Cuturi, M., 2015, ”On Wasserstein Two Sample Testing and Related Families of Nonparametric Tests”, arXiv.

Watanabe, F., and Uchida, T., 2015, ”Micro-Siting of Wind Turbine in Complex Terrain: Simplified Fatigue Life Prediction of Main Bearing in Direct Drive Wind Turbines”, Wind Engineering, 39(4), 349-368.

Shaler, K., Jonkman, J,, Barter, G., Kreeft, J., and Muller, J., 2022, ”Loads assessment of a fixed-bottom offshore wind farm with wake steering”, Wind Energy - Wiley, 35(9), 1530-1554.

RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.