ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Minkowski logarithmic error: A physics-informed neural network approach for wind turbine lifetime assessment

Francisco de N Santos, Pietro D'Antuono, Nymfa Noppe, Wout Weijtjens, and Christof Devriendt

Vrije Universiteit Brussel, OWI-Lab
Pleinlaan 2, 1050 Brussels, Belgium

Abstract. In this contribution we present a physics-informed neural network (PINN) approach for wind turbine fatigue estimation. This PINN incorporates physical information of the structure's fatigue profile in its loss function, referred to as Minkowski logarithmic error (MLE) - an extension of the log loss for any given L^p space. The function is mathematically analysed and differentiated in order to better understand its behaviour. The results obtained using the MLE are favourably compared with previous efforts using the mean squared logarithmic error. Finally, the long-term error is evaluated based on the effect of p.

1 Introduction

The recent demand for sustainable energy sources has seen wind energy rise to the spotlight. In this context, cost-effective structural health monitoring is required if these assets are to be properly maintained, their lifetime possibly extended and the cost-of-energy lowered. This, paired with the rise of artificial intelligence has seen a growing number of interfaces. In this contribution, we propose a novel loss function (the Minkowski logarithmic error) to better capture the physics inherent to the target variable, a wind turbine fatigue metric (damage equivalent moment, DEM). Real-world data from an offshore wind turbine (OWT) was collected by strain gauges (installed at the interface between the turbine tower and the transition piece between the tower and the foundation, TW-TP), accelerations and SCADA and processed into 10-minute time-series metrics. This was further enhanced by the addition of wave and tidal data from a public Flemish maritime database. The model estimated for three turbines: OWT1 (training), OWT2 and OWT3. The general methodology is described in [1]. As for a detailed account of DEM, *cf.* [2].
The target variable, DEM, is based on Eq. (1.1) as defined by [3], wherein m is the slope of the S-N curve, or Wöhler exponent [4] (in this case $m = 5$, from the OWT's design documentation), n_i is the number of cycles of a given stress range, σ_i, r_o, the TW/TP outer radius, r_i, the TW/TP inner radius and $N_{eq} = 10^7$, a predefined number of cycles.

$$DEM = \left(\frac{\sum_i n_i \cdot \left(\frac{\Delta\sigma_i \cdot \frac{\pi}{2} \cdot (r_o^4 - r_i^4)}{r_i} \right)^m}{N_{eq}} \right)^{1/m} \tag{1.1}$$

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

This equation allows us to calculate DEMs on a 10-minute level (due to N_{eq}). In order to be able to compare DEMs accumulated at other timescales, the individual 10-min instances of DEMs are aggregated into a *long-term* DEM, given by Equation 1.2.

$$DEM_{LT} = \left(\frac{1}{n} \sum_{i=1}^{n} DEM_i \right)^{1/m} \tag{1.2}$$

2 Minkowski Logarithmic Error

In order to incorporate the physical information of Equation 1.2 into our machine learning (ML) model in a so-called learning bias approach to physics-informed ML (Φ-ML) we begin by studying the logarithmic loss function as a point of departure prior to the introduction of the Minkowski Logarithmic Error (MLE).

2.1 Logarithmic loss function

The mean squared logarithmic error (MSLE) can be seen as a measure of the ratio between the true and predicted values and employs the logarithmic function to the mean squared error loss function (see Equation 2.1, where $\mathbf{Y}, \hat{\mathbf{Y}}$, with $\mathbf{Y} = (y_1, \ldots, y_n)$ and $\hat{\mathbf{Y}} = (\hat{y}_1, \ldots, \hat{y}_n)$ are the real and predicted values vectors).

$$L(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{1}{n} \sum_{i=0}^{n} (\log(y_i + 1) - \log(\hat{y}_i + 1))^2 \tag{2.1}$$

The introduction of the logarithm makes MSLE only care about the relative difference between the true and the predicted value, or in other words, it only cares about the percentual difference between them. This means that MSLE will treat small differences between small true and predicted values approximately the same as big differences between large true and predicted values. MSLE also penalizes underestimates more than overestimates, introducing an asymmetry in the error curve. This later property is for us highly desirable, as we want to introduce a further *'safety-factor'* into our model by encouraging DEM overprediction. As this is a key concern of ours, we can quickly demonstrate why the logarithmic function can be used to penalize underpredictions through Theorem 1 and its proof.

Theorem 1. *The difference between the logarithm of a function, f, and the logarithm of the same function with an increment ε, is less than the difference between the logarithm of f and the logarithm of f with a decrement ε.*

Proof. Let $t \in \mathbb{R}_{>0}$. The logarithmic measure between t and t plus an increment ε can be given by the define integral [5]:

$$\log(t) - \log(t + \varepsilon) = \int_{t}^{t+\varepsilon} \frac{1}{u} du$$

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

similarly, the measure between t and its decrement ε:

$$\log(t) - \log(t-\varepsilon) = \int_{t-\varepsilon}^{t} \frac{1}{u} du = \int_{t}^{t+\varepsilon} \frac{1}{w-\varepsilon} dw$$

with $w = u+\varepsilon$, due to the integration by substitution. Naturally, as $w-\varepsilon < u$, the second integral will encompass a greater area than the first, and thus, the mean squared logarithmic loss of an underpredicting model $(t-\varepsilon)$ will be greater than that of an overpredicting model.

2.2 Mathematical definition of MLE

As mentioned above, in order to include some physical information into our machine learning model, a custom loss function was developed, primarily based on Equation 1.2 and the logarithm properties. One can see that the structure of Equation 1.2 is that of an L^p norm. Thus, the Minkowski Logarithmic loss can be seen as a extension of the logarithmic loss function to any L^p metric, also know as Minkowski distance (we could also define this function as a L^p-log loss). Equation 2.2 describes this function mathematically, extended for $1 \leq p < +\infty$ in the n-dimensional vector space $\mathbb{R}^n$ for $\mathbf{Y}, \hat{\mathbf{Y}}$. For our case the Wöhler exponent ($m = 5$) coincides with p.

$$L(\mathbf{Y}, \hat{\mathbf{Y}}) = \left(\sum_{i=0}^{n} |\log(y_i + 1) - \log(\hat{y}_i + 1)|^p \right)^{1/p} \tag{2.2}$$

Equation 2.2 can be described more elegantly as

$$L(\mathbf{Y}, \hat{\mathbf{Y}})^p = \sum_{i=0}^{n} \left| \log\left(\frac{y_i + 1}{\hat{y}_i + 1} \right) \right|^p \tag{2.3}$$

We can furthermore describe Equation 2.2 through its Lebesgue integral in the measurable space (S, Σ, μ) of the set S, σ-algebra Σ and measure μ (we can do this because the logarithmic function is a measurable function f of the type $f : S \to \mathbb{R}_{>0}$). Thus

$$\begin{aligned} ||\log(y_i + 1) - \log(\hat{y}_i + 1)||_p = \left(\int_S |\log(y_i + 1) - \log(\hat{y}_i + 1)|^p d\mu \right)^{1/p} = \\ = \left[\int_S \left| \int_{\hat{y}}^{y} \frac{1}{u+1} du \right|^p d\mu \right]^{1/p} \end{aligned}$$

is valid $\forall\ \mathbf{Y}, \hat{\mathbf{Y}} \in \mathbb{R}_{>0}$ with $\mathbf{Y} = (y_1, \ldots, y_n)$ and $\hat{\mathbf{Y}} = (\hat{y}_1, \ldots, \hat{y}_n)$ the real and predicted values vectors, respectively. Note that the second equality

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

appears from the integral definition of the logarithm. As such, the Minkowski logarithmic loss function shares its properties with Lebesgue spaces and is a seminormed vector space $\mathcal{L}^p(S, \mu)$.

In order to better understand the behaviour of Equation 2.3 and how it influences the neural network's learning, we shall differentiate it by applying the chain rule. Let the fraction within the logarithm of Equation 2.3 be $\mathbf{Y}/\hat{\mathbf{Y}} = \mathbf{x}$. Thus, $\forall j \in \mathbb{N} = \{1, \ldots, n\}$ we have

$$\begin{aligned}
\frac{\partial}{\partial x_j} ||\log \mathbf{x}||_p &= \frac{\partial}{\partial x_j} \left(\sum_{i=1}^{n} |\log x_i|^p \right)^{1/p} \\
&= \frac{1}{p} \left(\sum_{i=1}^{n} |\log x_i|^p \right)^{\frac{1}{p}-1} \cdot \frac{\partial}{\partial x_j} \left(\sum_{i=1}^{n} |\log x_i|^p \right) \\
&= \frac{1}{p} \left(\sum_{i=1}^{n} |\log x_i|^p \right)^{\frac{1-p}{p}} \cdot p \sum_{i=1}^{n} |\log x_i|^{p-1} \cdot \frac{\partial}{\partial x_j} |\log x_i| \qquad (2.4)
\end{aligned}$$

The term $\partial/\partial x_j |\log x_i|$ in Equation 2.4 is a piecewise function whose derivative is non-negative (the sign function of $|\log x_i|$) if $i = j$ and 0 otherwise, known as Kronecker delta, δ_{ij} [6]. Thus, Equation 2.4 yields

$$\left[\left(\sum_{i=1}^{n} |\log x_i|^p \right)^{1/p} \right]^{1-p} \sum_{i=1}^{n} |\log x_i|^{p-1} \frac{\log x_i}{|\log x_i|} \delta_{ij}$$

One of the key properties of the Kronecker function is the reversal of the integration variables, so that $\sum_{i=1}^{n} a_i \delta_{ij} = a_j$ [6]. If we consider that $a_i = |\log x_i|^{p-1} \log x_i / |\log x_i|$, the derivative reduces to

$$||\log \mathbf{x}||_p^{1-p} \cdot |\log x_j|^{p-1} \cdot \frac{\log x_j}{|\log x_j|}$$

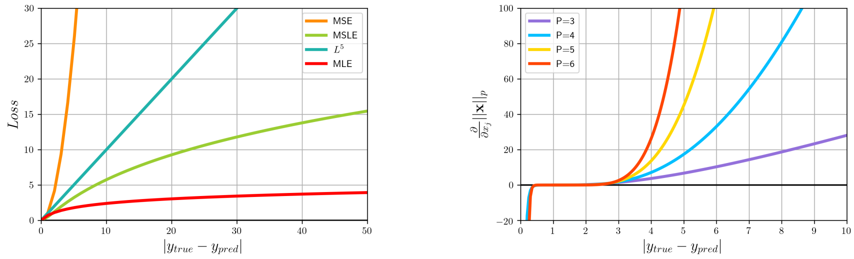
We can finally simplify this term, and re-substitute $\mathbf{x}$ for the real and predicted value vectors.

$$\frac{\partial}{\partial x_j} ||\log \mathbf{x}||_p = \frac{\log x_j |\log x_j|^{p-2}}{||\log \mathbf{x}||_p^{p-1}} = \frac{\log\left(\frac{y_j+1}{\hat{y}_j+1}\right) \left|\log\left(\frac{y_j+1}{\hat{y}_j+1}\right)\right|^{p-2}}{\left|\left|\log\left(\frac{\mathbf{Y}+1}{\hat{\mathbf{Y}}+1}\right)\right|\right|_p^{p-1}} \qquad (2.5)$$

As we can see, Equation 2.5 can be described as a fraction of a log function with its L^p norm as denominator, positive for $p \in \mathbb{R}_{>1}$ and with its zero at 1.

Finally, we can see how this function behaves according to the residuals and compare it with other loss functions (Figure 1a) and plot the derivative for several values of p (Figure 1b).

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.



(a) (b)

Fig. 1: (a) Behaviour of MSE, MSLE, L^5 distance and MLE ($p = 5$) functions according to the residuals ($|y_{true} - y_{pred}|$). (b) Behaviour of the derivative of the Minkowski logarithmic function, $\partial/\partial x_j || \log \mathbf{Y}/\hat{\mathbf{Y}}||_p$, for the set $p = \{3, \ldots, 6\}$.

What transpires from Figure 1b is that by increasing the value of p the same occurs for the exponential behaviour of the derivative's curve. Therefore, we can say that the greater the p, the greater will be the loss, and most importantly, the greater will be the penalization underpredictions will have, as seen in subsection 2.1. Thus, the merger of the L^p metric with the log loss introduces a variable factor (p) which allows to control of the amount of penalization given to underpredictions.

2.3 Discussion

We can see the results of the MLE loss function implementation for the practical case of TW-TP DEM estimation as defined in section 1 in Table 1.

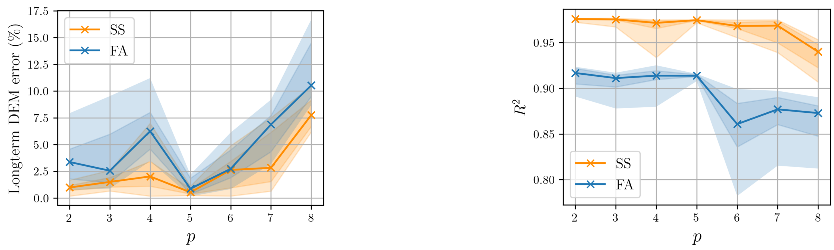
	MSLE						MLE					
	OWT1		OWT2		OWT3		OWT1		OWT2		OWT3	
	FA	SS	FA	SS	FA	SS	FA	SS	FA	SS	FA	SS
R^2	0.89	0.92	0.86	0.93	0.88	0.92	0.92	0.98	0.87	0.96	0.92	0.99
δ_{LT} (%)	7.34	11.3	7.25	10.1	6.72	11.9	-0.5	-0.1	1.11	-1.2	1.19	0.59

Table 1: Comparison of models' performance for MSLE and MLE (R^2 and long term DEM error, δ_{LT}) for the three turbine in each direction (fore-aft, FA and side-to-side, SS).

In this table we can see how the introduction of a Φ-ML approach for the loss function greatly improves the models' performance on the long term accumulations of Equation 1.2 (δ_{LT}, with an added incentive for overpredictions), whilst retaining and even slightly improving the 10-minute estimation performance (R^2). Moreover, the negative values δ_{LT} indicate that the model is overpredicting, as intended.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Finally, we can investigate the loss function landscape by checking how the variation of the value of p affects the PINN's performance (Figure 2).



(a) Longterm DEM error (%)

(b) R^2

Fig. 2: Effect the variation of p has on model performance (95% confidence interval). For each value of p, 10 models were generated and evaluated.

In this figure, one can see a similar behaviour both for SS and for FA. Left of $p = m = 5$, both the long-term errors and R^2 are in-par, albeit slightly increasing. We can understand this has being representative of Dvoretzky's theorem [7], wherein the distance of a finite-dimensional vector space can be approximately defined as Euclidean (L^2). The lowest error is evidently achieved for $p = m = 5$. On the right side we have an almost linear increase of the error with regards to p. Interestingly, we again see that, albeit with alterations (namely its increase after $p = 5$), the R^2 which gives us an idea of how the model performs at a 10-minute level seems to not have a stong correlation with the long-term accumulated DEM errors, particularly if $p < m$.

References

[1] Francisco d N Santos, Nymfa Noppe, Wout Weijtjens, and Christof Devriendt. Data-driven farm-wide fatigue estimation on jacket-foundation owts for multiple shm setups. *Wind Energy Science*, 7(1):299–321, 2022.

[2] Clemens Hübler, Wout Weijtjens, Raimund Rolfes, and Christof Devriendt. Reliability analysis of fatigue damage extrapolations of wind turbines using offshore strain measurements. *Journal of Physics: Conference Series*, 1037(3):032035, 2018.

[3] HB Hendriks and BH Bulder. Fatigue equivalent load cycle method, 1995.

[4] Gabriel Marsh, Colin Wignall, Philipp R Thies, Nigel Barltrop, Atilla Incecik, Vengatesan Venugopal, and Lars Johanning. Review and application of rainflow residue processing techniques for accurate fatigue damage estimation. *International Journal of Fatigue*, 82:757–765, 2016.

[5] Timothy Gowers. The exponential and logarithmic functions. In Timothy Gowers, June Barrow-Gree, and Imre Leader, editors, *The Princeton companion to mathematics*, chapter III.25, pages 199–202. Princeton University Press, 2008.

[6] John Lighton Synge and Alfred Schild. *Tensor calculus*. University of Toronto Press, 2020.

[7] Aryeh Dvoretzky. Some results on convex bodies and banach spaces. *Matematika*, 8(1):73–102, 1964.