

A Physics-informed Neural Network for Wind Turbine Main Bearing Fatigue

Yigit A. Yucesan¹ and Felipe A. C. Viana²

^{1,2} *Department of Mechanical and Aerospace Engineering, University of Central Florida, Orlando, FL, 32816, USA*
yucesan@knights.ucf.edu
viana@ucf.edu

ABSTRACT

Unexpected main bearing failure on a wind turbine causes unwanted maintenance and increased operation costs (mainly due to crane, parts, labor, and production loss). Unfortunately, historical data indicates that failure can happen far earlier than the component design lives. Root cause analysis investigations have pointed to problems inherent from manufacturing as the major contributor, as well as issues related to event loads (e.g., startups, shutdowns, and emergency stops), extreme environmental conditions, and maintenance practices, among others. Altogether, the multiple failure modes and contributors make modeling the remaining useful life of main bearings a very daunting task. In this paper, we present a novel physics-informed neural network modeling approach for main bearing fatigue. The proposed approach is fully hybrid and designed to merge physics-informed and data-driven layers within deep neural networks. The result is a cumulative damage model where the physics-informed layers are used model the relatively well-understood physics (L10 fatigue life) and the data-driven layers account for the hard to model components (i.e., grease degradation).

1. INTRODUCTION

As pointed by Hornemann and Crowther (2013), main bearings of onshore wind turbines are subjected to multiple failure modes, among which we can mention wear and micro-pitting, false brinelling due to stationary loading, electrostatic discharge, cage and guide ring wear, manufacturing defects and quality problems. Factors that trigger these failure modes in the field include:

- Environment: extreme wind conditions, hazardous weather, ambient temperature, humidity, dust, etc.
- Operation: machine controls (e.g. induced yaw misalignment for wake management), derating levels, etc.

Yigit Yucesan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

- Maintenance and services practices: lubricant condition, inspection, cleaning, and regreasing frequency, etc.

Literature reports a number of approaches to main bearing life estimation. Butler et al. (2012) focused on utilizing supervisory control and data acquisition (SCADA) data to forecast the remaining useful life by constructing a residual model for bearing temperature. Authors considered variables such as main shaft rotational speed, hydraulic brake temperature, hydraulic brake pressure, and blade pitch position as well as a compensation for ambient temperature. As a result, they managed to provide a failure indication with a 30 day lead time. Another example is the result published by Watanabe and Uchida (2015). Authors estimate wind turbine rear bearing fatigue using standard bearing life calculations found in ISO 28123. The model uses hub-height 10 minutes wind data as an input. The model showed good agreement with failures observed in Japan. While collected field data indicated $L_{10} = 12.7$ years, the model predicted $L_{10} = 12$ years. Authors also showed how their model could be used to quantify life extension through curtailment. Yucesan and Viana (2019a) used a fatigue damage accumulation model to manage reliability at a wind-turbine level across different farms. The results demonstrate that fatigue life contributes significantly to bearing failures, especially under poor lubrication conditions. They also showed how to use the cumulative damage model to promote component life extension (assigning turbine-specific maintenance through regreasing). Addressing similar concerns, Walker and Coble (2018) proposed a combined adaptive sampling and order tracking approach to investigate vibration sensor data for fault detection. While the authors use the adaptive sampling non-stationary signals are converted to stationary, they utilize order tracking to distinguish the bearing fault. In their case study, they were able to detect the bearing fault of one of the machines and confirmed the failure in postmortem examinations.

From a physics perspective, it is understood that lubricant condition drastically affects bearing fatigue life. Unfortunately, modeling the lubricant degradation is incredibly difficult. Zhu

et al. (2013) proposed a methodology for estimating the remaining useful life of lubricant using viscosity and a dielectric constant sensor output and integrating these parameters as an observation function by particle filtering technique to predict the remaining useful life of the lubricant. Their proposed model was validated by laboratory experiments. Results of the conducted case study show that the single observation on dielectric constant sensor gives the best accuracy on the life prediction. Another research was conducted by Iyer et al. (2007) to provide a method for the early detection of lubrication anomalies in oil-lubricated bearings. The authors investigated two types of anomalies: lack of lubricant and presence of contamination. In their study, they used acoustic emissions and vibration signals. Through experiments, they showed that these techniques not only detect the anomaly, but also provide an insight on the level of the anomaly.

Detecting component condition is important; however, it is the forecast of remaining useful life (prognosis) that defines critical decisions in a wind farm (such as repair versus replacement, bundling services in wind turbines to save on crane costs, etc.). These decisions are based on predictive models built using kernels that can be physics-informed, data-driven, or a combination of both. Liu and Mahadevan (2009) proposed two efficient methods for the time-dependent fatigue reliability analysis. The authors used a modified Miner's rule approach for nonlinear damage accumulation and carried out uncertainty quantification for the life curves they used in their experimental validation. They applied two methods for fatigue reliability calculation: moment matching approach and first-order reliability method. Results from the validation prove the efficiency of these methods compared to the simulation-based approaches. Goebel et al. (2008) investigated data-driven models for prognostics. Authors compared three distinct data-driven techniques for prognostics: relevance vector machine, Gaussian process regression, and neural networks. They tested and compared these methods on a damage propagation problem with sparse and noisy data. Although they showed all these methods can learn the dynamics of the propagation fairly well, the authors also emphasized that the discrepancy between models and the ground true is mainly caused by inadequate amount of training data. Eker et al. (2019) proposed a methodology that uses physics-informed model for short-term prediction and then integrates it with a longer-term projection of a similarity-based data-driven framework. They illustrate the performance of their approach on two engineering applications and prove that a hybrid approach can improve accuracy, robustness, and applicability; and reduce the number of data required for modeling.

The steady growth of computational power available contributed to the popularization of machine learning in engineering applications. The scientific community has studied and proposed architectures that leverage formulations based on physics (Yu et al., 2018; Raissi & Karniadakis, 2018; Chen et

al., 2018). In essence, differential equations are used to train multilayer perceptrons and recurrent neural networks¹. One distinct research that incorporates physics into the solution of a machine learning problem is conducted by Matei et al. (2019). In their contribution authors consider a classification problem based on a data generated by a partially-known physical system. They leverage the partially-known physics to reduce the difficulty of the problem from a complex classifier to a simple regression problem. The authors illustrated their proposed approach on a rail switch system and proved the performance of this hybrid approach is as good as purely data-driven model. Nascimento and Viana (2019) proposed a recurrent neural network cell inspired on cumulative damage models². As the authors provide an efficient way to use physics-based and data-driven layers together within a recurrent neural network cell, they illustrate the predictive capability of their proposed technique on a fatigue crack damage accumulation problem. They finally showed that this hybrid approach can successfully model fatigue crack growth even with inadequate training data (i.e., small number of observations or unbalanced data).

This paper proposes overcoming some of the limitations in modeling bearing fatigue life by infusing physics into machine learning models. Namely, we propose modeling fatigue life through a cumulative damage model coded in the form of a recurrent neural network (Goodfellow et al., 2016). Bearing fatigue is implemented through a physics-informed layer within this network while lubricant degradation is implemented through a purely data-driven layer. The proposed approach is fully hybrid as it merges physics-informed and data-driven layers within deep neural networks. The result is a cumulative damage model where the physics-informed layers are used to model the relatively well understood physics (L_{10} fatigue life) and the data-driven layers account for the hard to model components (contribution due to poor greasing conditions). In order to present the main features of the proposed physics-informed neural network, we use openly available data about main bearing failures for a 1.5 MW wind turbine platform and weather data for a representative wind farm.

The remaining of the paper is organized as follows. Section 2 gives an overview on physics-informed neural networks and our approach to modeling main bearing fatigue and grease degradation. Section 3 describes the case study and the design of the neural network. Section 4 presents and discusses the numerical results. Finally, section 5 closes the paper recapitulating salient points and presenting conclusions and future work. There is one appendix at the end of the paper, discussing grease degradation modeling, data, bearing temperature calculation, and activation functions.

¹The interested reader can find literature on Gaussian processes (Schober et al., 2014; Raissi et al., 2018).

²Cumulative damage models are often used to describe the irreversible accumulation of damage (progressive distress) throughout the useful life of components or systems (Fatemi & Yang, 1998; Frangopol et al., 2004)

2. PHYSICS-INFORMED MACHINE LEARNING

2.1. Background: Recurrent Neural Networks and Cumulative Damage Models

Recurrent neural networks have been widely used to model time-series (Connor et al., 1994; Sak et al., 2014; Chauhan & Vig, 2015) speech recognition (Graves et al., 2013), remaining useful life estimation (Gugulothu et al., 2018), and many other applications. These neural networks transform a vector of hidden states, \mathbf{d} , from time stamp $t - 1$ to time stamp t , in the following fashion:

$$\mathbf{d}_t = f(\mathbf{d}_{t-1}, \mathbf{x}_t) \quad (1)$$

where $t \in [0, \dots, T]$ represent the time discretization, $\mathbf{d} \in \mathbb{R}^{n_d}$ are the states representing the sequence, $\mathbf{x}_t \in \mathbb{R}^{n_x}$ are input (observable) variables, and $f(\cdot)$ is the transformation to the hidden state (as known in the literature as recurrent neural network cell).

As illustrated in Fig. 1a, the recurrent neural network cells repeatedly apply the transformation $f(\mathbf{d}_{t-1}, \mathbf{x}_t)$ as a way to update the states. In its simplest form, $f(\mathbf{d}_{t-1}, \mathbf{x}_t)$ can be implemented as a single-layer perceptron (fully-connected dense layer), shown in Fig. 1b. Nevertheless, the cell architecture can be implemented using much more elaborated designs. For example, the scientific community have proposed the Long short-term memory cell (Hochreiter & Schmidhuber, 1997), illustrated in Fig. 1c. The cell was designed to (a) improve the predictions of the neural network, and (b) mitigate the vanishing gradient problem (Goodfellow et al., 2016). For further details on recurrent neural networks (including applications), the interested reader is referred to Connor et al. (1994); Hochreiter and Schmidhuber (1997); Sutskever et al. (2011); Graves et al. (2013); Sak et al. (2014); Chung et al. (2015); Chauhan and Vig (2015); Goodfellow et al. (2016).

Cumulative damage models (Fatemi & Yang, 1998; Frangopol et al., 2004) represent the irreversible accumulation damage through an initial value problem. In its discrete form, \mathbf{d}_t is the damage at time t and it is the result of a damage increment $\Delta \mathbf{d}_t$ on top of damage \mathbf{d}_{t-1} at previous time step $t - 1$:

$$\mathbf{d}_t = \mathbf{d}_{t-1} + \Delta \mathbf{d}_t \quad (2)$$

where $\Delta \mathbf{d}_t$ is often a function of \mathbf{d}_{t-1} and some other inputs \mathbf{x}_t at time t . The characterization of the damage \mathbf{d}_t (which is associated with a failure mechanism) and the inputs \mathbf{x}_t depends on the problem. For example, if fatigue is the failure mechanism, fatigue crack length is the observable quantity. The inputs \mathbf{x}_t usually express time-dependent loading and boundary conditions (e.g., pressures, temperatures, torques, mechanical and thermal stresses, etc.) or even operating points (e.g., wind speed, blade pitch angles, etc.).

We use the repeating cell proposed by Nascimento and Viana (2019) and illustrated in Fig. 2 to model cumulative damage

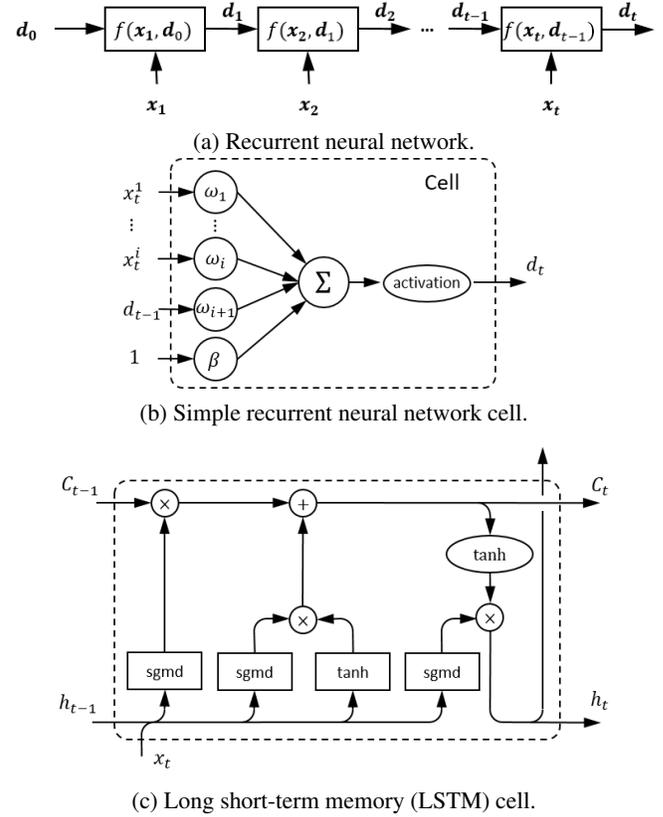


Figure 1. Examples of recurrent neural networks cells. In the LSTM cell, the squares are perceptrons with predefined activation functions; the oval shape is just the \tanh activation.

through recurrent neural networks. “MODEL” maps the inputs \mathbf{x}_t and previous damage \mathbf{d}_{t-1} into a damage increment $\Delta \mathbf{d}_t$. In other words, the “MODEL” block implements the damage increment in the damage accumulation model. There is nothing preventing the implementation of “MODEL” to be purely physics-informed³ or purely data driven (such as using a multilayer perceptron). Nevertheless, as we discuss in next section, we advocate towards implementing “MODEL” as a hybrid model (where some parts are physics-informed while others are data-driven).

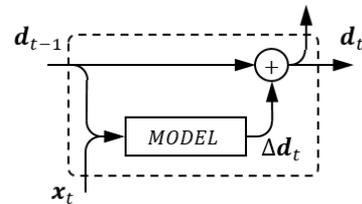


Figure 2. Cumulative damage recurrent neural network cell.

³As long as adjoints with respect to model parameters are available, these could be adjusted through optimization just as weights and biases in any neural network.

2.2. Physics-Informed Neural Networks for Main Bearing Fatigue and Grease Degradation

Bearing fatigue life can be modeled in terms of the dynamic loads and multiplication factors that reflect design, alloy, surface treatment, lubrication, and contamination, among other factors. As found in the SKF catalog for spherical roller bearings (SKF-contributors, 2007), fatigue life can be calculated as:

$$L_{nm}^{BRG} = a_1 a_{SKF} \left(\frac{C}{P} \right)^{\frac{10}{3}} \quad \text{and} \quad L_{nmh}^{BRG} = \frac{10^6}{60N} L_{nm}, \quad (3)$$

where L_{nm}^{BRG} is the main bearing rated life at 100 – $n\%$ reliability (in millions of revolutions), L_{nmh}^{BRG} is the rating life at 100 – $n\%$ reliability (in operating hours), C is the basic dynamic load rating (in kN), P is the equivalent dynamic bearing load (in kN), and N is the rotational speed (in rpm). Table 1 illustrates few values for the reliability level life adjustment factor, a_1 . Figure 3 shows few curves for the SKF life modification factor, a_{SKF} .

Reliability (%)	Probability of Failure (%)	L_{nm}	a_1
90	10	L_{10m}	1.00
95	5	L_{5m}	0.62
96	4	L_{4m}	0.53
97	3	L_{3m}	0.44
98	2	L_{2m}	0.33
99	1	L_{1m}	0.21

Table 1. a_1 life adjustment factor (SKF-contributors, 2007).

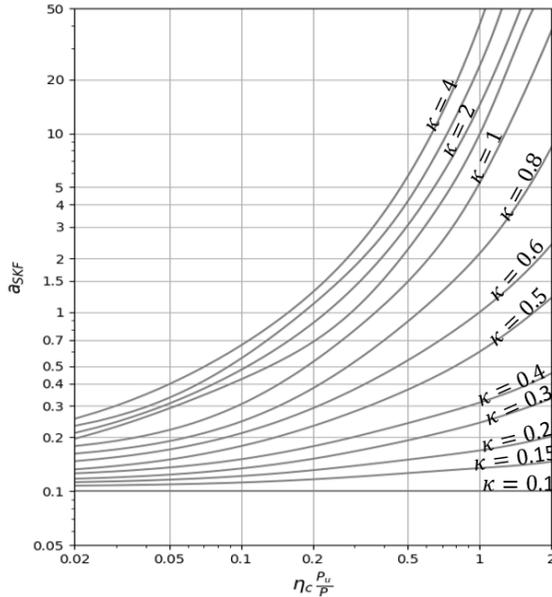


Figure 3. a_{SKF} life adjustment factor, adapted from SKF-contributors (2007).

The adjustment factor a_{SKF} depends on the lubricant condition (in terms of viscosity, through the viscosity ratio, κ , and

particulate contamination through the contamination factor, η_c), the equivalent dynamic bearing load P , and the fatigue load limit P_u . Viscosity ratio, κ , is expressed as:

$$\kappa = \nu / \nu_1, \quad (4)$$

where ν actual operating viscosity of the lubricant (mm^2/s), and ν_1 rated viscosity, depending on the bearing mean diameter and rotational speed, (mm^2/s).

When a bearing operates at different load and rotational speed levels, the rated lives are obtained through Palmgren-Miner's rule:

$$L_{nm}^{BRG} = \frac{1}{\sum \frac{60N_i t_i}{L_{nm}^{BRG_i}}} \quad \text{and} \quad L_{nmh}^{BRG} = \frac{1}{\sum \frac{60N_i}{L_{nmh}^{BRG_i}}}, \quad (5)$$

where t_i is number of hours the turbine ran at N_i rpm. In other words, the Palmgren-Miner's rule characterizes the incremental damage at each cycle:

$$\Delta d_t^{BRG} = \frac{n_t}{L_t^{BRG}} \quad (6)$$

where n_t is the number of cycles passed for each time step.

In this paper, we used data available in the literature to characterize the viscosity of grease as a function of temperature as well as contamination factor as a function of viscosity ratio (as illustrated in Fig. 4). There are several curves that represent grease types with different viscosity grades (VG) given in the SKF plot for lubricant viscosity calculation. We picked VG 320 for our case study as the virgin (undamaged) grease behavior, following recommendations found in the Schaeffler catalogue (Schaeffler-contributors, 2016). As for contamination factor, we considered that virgin grease would present slight contamination, while degraded lubricant would present very severe contamination (SKF-contributors, 2007). Figure 4 illustrates the variation of grease properties for the virgin and degraded greases.

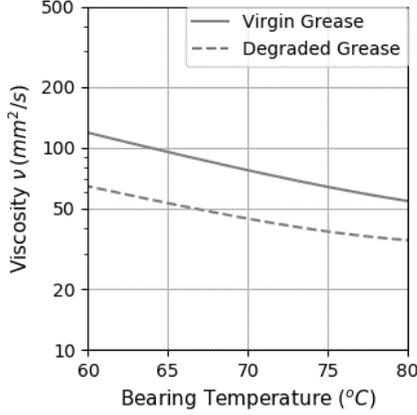
After determining the grease damage value, we use it as a factor to interpolate between curves assigned to the virgin and degraded states of the lubricant (see Fig. 4):

$$\nu_t = d_t^{GRS} (\nu_{deg} - \nu_{vir}) + \nu_{vir} \quad (7)$$

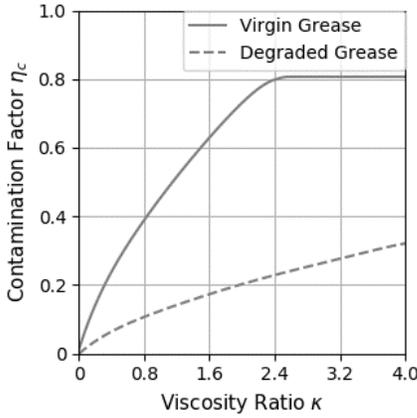
$$\eta_{c,t} = d_t^{GRS} (\eta_{c,deg} - \eta_{c,vir}) + \eta_{c,vir} \quad (8)$$

where ν and η_c are viscosity and contamination factor of the grease respectively.

It is challenging to build a purely physics-informed model for bearing fatigue life since grease degradation is extremely complex. In this paper, we build a recurrent neural network that tracks bearing fatigue damage, d_t^{BRG} , and grease damage, d_t^{GRS} , simultaneously through a hybrid model. The bearing fatigue damage increment Δd_t^{BRG} implements Eq. 6 (and therefore is physics-informed). **The grease damage**



(a) Grease viscosity vs. bearing temperature, adapted from SKF-contributors (2007).



(b) Contamination factor vs. viscosity ratio, adopted from SKF-contributors (2007).

Figure 4. Viscosity and contamination factor for virgin and degraded grease.

increment represents the degradation of viscosity and increase in contamination. There are attempts to build physics-informed models for grease life, but it is not clear how they relate to field conditions (see appendix for one example of such models). **Here, we implement the grease damage increment Δd_t^{GRS} through a multilayer perceptron.**

With that, we propose the repeating recurrent neural network cell illustrated in Fig. 5 to model the bearing and grease cumulative damage. This recurrent neural network cell takes wind speed (WS_t) and the bearing temperature (T_t) as input variables. The cell will be recurrently used, as in Fig. 1a, updating both the grease and bearing damages from previous time step (d_{t-1}^{GRS} and d_{t-1}^{BRG} , respectively). While WS_t is mapped to equivalent dynamic bearing load (P_t) (see Fig. 7c), bearing temperature (T_t) and cumulative grease damage from previous time step (d_{t-1}^{GRS}) are used to calculate grease damage parameters κ_t and η_{ct} (as in Fig. 4). Combined with

P_t , these parameters are incorporated to evaluate inverse life adjustment factor $1/a_{SKF_t}$ (see Fig. 3), which is then multiplied with non-adjusted bearing fatigue damage increment (n_t/N_t) for bearing fatigue damage increment (Δd_t^{BRG}) calculation (Eqs. (3-6)). The data-driven portion of the hybrid model is given by prediction of grease damage increment (Δd_t^{GRS}) via multilayer perceptron using P_t , T_t , and d_{t-1}^{GRS} as inputs (further discussed in section 3.4). In this scheme, while we maintain a physics portion with bearing fatigue accumulation, we compensate the missing physics knowledge within the grease model with the help of neural networks. The training of this recurrent neural network aims at calibrating the multilayer perceptron using grease damage observations and let it learn the damage accumulation on grease. Our implementation allows for optimization of parameters within the physics-informed layers (in case these are inaccurate or unknown). We recognize that calibrating these physical parameters might require acquisition of data that supports such task. In this study, we focus on compensating the unknown grease degradation phenomenon through data-driven layers.

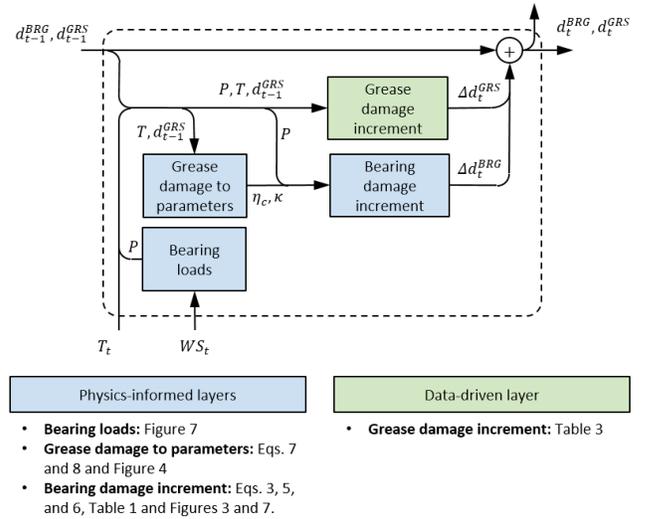


Figure 5. Physics-informed neural network framework for main bearing fatigue and grease degradation.

One might be tempted to ask why using the cell shown in Fig. 2 instead of architectures such as the long-short term memory shown in Fig. 1c? Why should one use hybrid model mixing physics-informed and data-driven layers? The answer has to do with the specific application of the cumulative damage models. In this paper, as in many other applications in prognosis, the input conditions (loads, temperatures, etc.) are fully observed (throughout the entire time series). However, damage is only partially observed. As we will discuss next, grease damage is assessed through grease sample analysis done in specialized laboratories in regular intervals. Bearing fatigue is rarely quantified and, for the most part, only time of failure is known. Figure 6a shows the typical data collected

for training the cumulative damage model. In fairness, this represents only one specific wind turbine; and likely, there would be load history and inspection data available for few wind turbines. Figure 6b illustrates the typical data collected for prediction using the trained cumulative damage model. Again, load history is available throughout the useful life and the initial value for the states is either known or assumed. The cumulative damage model is then used to estimate the damage over time. Given the extremely few observations of damage, we argue that using purely data-driven architectures is unlikely to lead to accurate models. Architectures such as long short-term memory and other purely data-driven architectures might still be useful in cases where there is full observation of the states. This can happen when damage is continuously monitored through dedicated health monitoring sensors (e.g., comparative vacuum monitoring (Roach, 2009), fiber Bragg grating sensors (Hill & Meltz, 1997), etc.). However, this is not the case (as techniques such as these are rarely used in main bearing monitoring) nor the focus of this paper.

3. CASE STUDY

3.1. Wind Turbine Model

In our case study, we chose a 1.5MW wind turbine with 80 meters hub height, equipped with a main bearing in the three-point mounting configuration. Table 2 provides some key parameters of wind turbine and main bearing used in our cased study.

Wind turbine	
Rated power	1.5 MW
Cut-in wind speed	3.5 m/s
Rated wind speed	12 m/s
Cut-out wind speed	25 m/s
Maximum rotor speed	20 rpm
Hub height	80 m
Main bearing	
Designation	SKF 230/600 CAW33
Basic dynamic load rating C	6,000 kN
Fatigue load limit P_u	750 kN
Mass	405 kg
Mean diameter d_m	735 mm

Table 2. Specifications about the wind turbine, adapted from GE-contributors (2009), and main bearing, adopted from SKF-contributors (2007).

Mapping from wind speed to dynamic bearing loads is maintained using a published National Renewable Energy Laboratory (NREL) report (Sethuraman et al., 2015), which involves a plot that provides dynamic load value for a given wind speed condition for the same type of main bearing we used in our case study, mounted on a 1.5MW wind turbine. The authors report that the mapping is a result of multi-body model simulation. Rotational speed output is calculated using the power

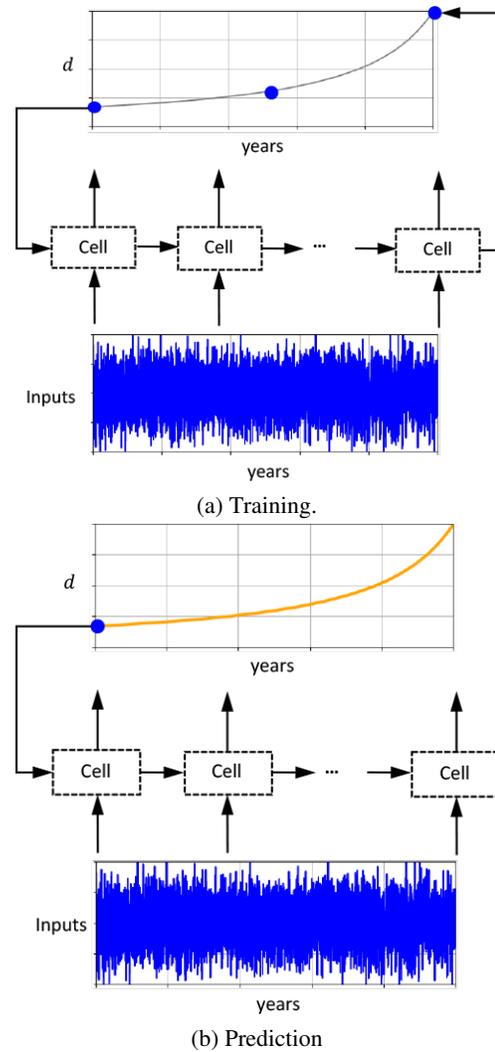


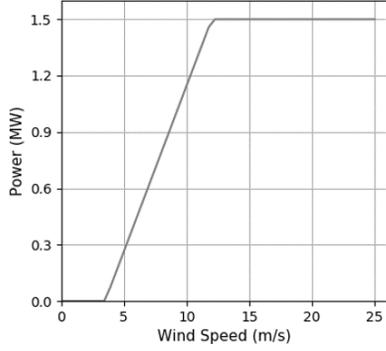
Figure 6. Typical use-case of recurrent neural network for cumulative damage model.

curve of the wind turbine. Load, power, and rotational speed curves are provided in Fig. 7.

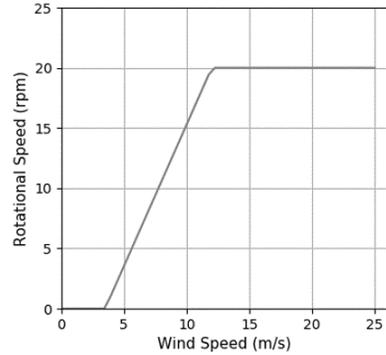
3.2. Nominal Wind Speed and Bearing Temperature

Site-specific data is obtained from a database also provided by NREL (Draxl et al., 2015), which includes environmental data at one hour resolution between 2007 and 2013 for 126,000 different locations throughout the United States. For the present case study, we arbitrarily chose Clayton, NM without any particular reason. Although data does not come directly from an actual wind park, we believe the NREL data provided for Clayton, NM is representative of a region in the USA with high penetration of wind energy.

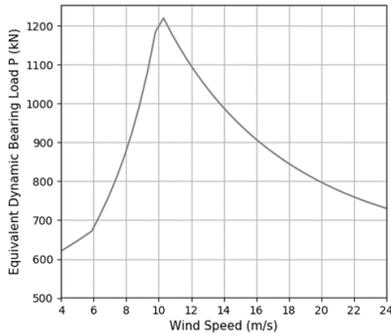
In order to mimic SCADA systems, the original NREL environmental data is augmented (up-sampling) to achieve the 10 minute resolution. Data is also extended up to 30 years to be



(a) Power curve, adapted from GE-contributors (2009).



(b) Rotational speed, adapted from GE-contributors (2009).



(c) Equivalent dynamic bearing load, adapted from Sethuraman et al. (2015).

Figure 7. Wind speed mapping for case study turbine.

used for long term bearing fatigue life predictions. Details of the data augmentation are given in the appendix. On top of that, since main bearing temperature is not originally available, we use an analytical model to estimate these values based on ambient and produced power. The details are given in the appendix. At the end, the time series that we consider nominal conditions for wind speed and bearing temperature are shown in Fig. 8.

In order to generate synthetic set of wind turbines, we divided 7 years of data into segments of 6 months which yields to 14 different data sets that we treat as 14 different turbines. We partitioned these machines into 10 training and 4 validation turbines.

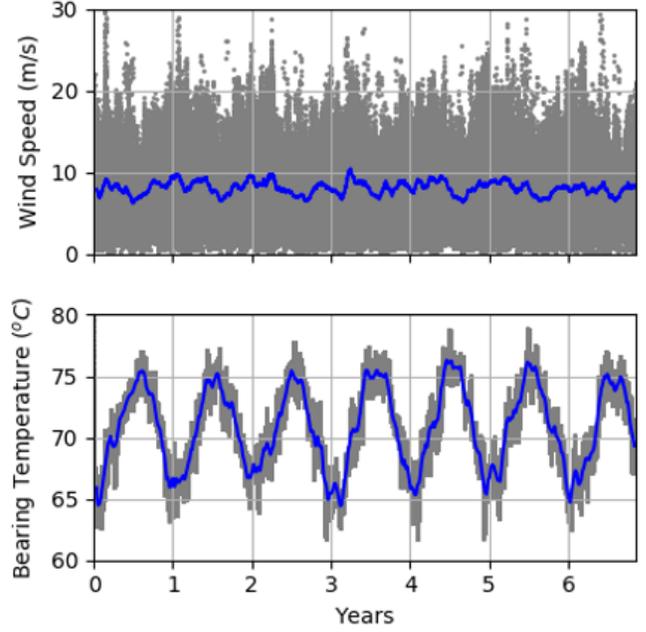


Figure 8. Time series for wind speed and main bearing temperature. Data is represented in gray and trend is plotted in blue (monthly moving average to highlight any seasonality).

3.3. Grease Samples

The bearing fatigue model needs information about the viscosity and contamination of grease over time. One way to obtain these grease parameters is through periodic sampling and laboratory analysis. With the process repeated periodically, the parameters used in bearing fatigue estimation could be updated, allowing for accurate lifing of the component.

Here, we create grease sample analysis results synthetically using the model described in the appendix. In order to make the study more interesting, the effect of grease state on grease related parameters like viscosity and contamination is described by a quadratic relationship (see Fig. 9):

$$d^{GRS} = \frac{1}{(L^{GRS})^2}, \quad (9)$$

where d^{GRS} is the grease damage that directly relates to damage in terms of viscosity (i.e. loss of viscosity) and damage in contamination (i.e., increase in contamination), and L^{GRS} is the life of grease (see Eq. 12 shown the appendix).

Equations 7, 8, and 9 are rather arbitrary and they are only used here as a way to generate grease sample data for this study. As mentioned before, wind farm operators and service providers could obtain data for viscosity and contamination through grease sample analysis. We could have separated the grease damage in terms of viscosity, d^v , and damage in contamination, d^{nc} . However, in this study we simplified and make $d^{GRS} = d^v = d^{nc}$. In reality, the damage accumulation rates can vary depending on the grease parameter of inter-

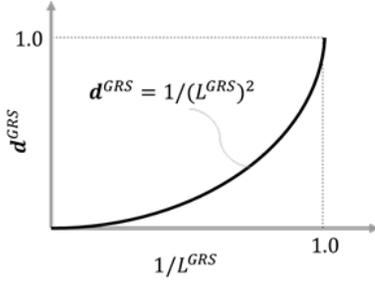


Figure 9. Quadratic relationship between grease life and damage.

est. We build our grease sample data by assuming that grease analysis is conducted at the end of every month continuously for a period of six months. The sampling procedure essentially assess the level of degradation of grease. By this logic, we collect d^{GRS} values for each turbine at the end of each month for six months. We also assume that full regreasing of main bearing occurs every 6 months. In terms of modeling, regreasing basically resets the grease damage back to zero (i.e., $d^{GRS} = 0$ after regreasing).

3.4. Physics-informed Neural Network Design

We considered the following information is available:

- for every turbine in the fleet: wind speed and main bearing temperature from SCADA (inputs for the model as described in section 2.2), and
- for part of the fleet: grease damage metric, d^{GRS} , observed every month for six months straight.

With that information, we proceed to build a hybrid physics-informed neural network model for bearing fatigue. In this model, the grease degradation increment, Δd^{GRS} , is a multilayer perceptron and the bearing damage accumulation is physics-informed. The inputs of the multilayer perceptron models are scaled between zero and one to avoid the disparity in the order of magnitude of inputs interfering in the fitting of the neural networks. Table 3 details the multilayer perceptron architectures tested in this work. For the most part of this paper, we decided to use MLP#1 architecture to illustrate the ability to fit a neural network with a large number of trainable parameters. Nevertheless, we also included the study on the effect of different architectures on the overall model performance in section 4.

The constructed multilayer perceptron takes three inputs (wind speed, bearing temperature, and current predicted \hat{d}^{GRS}) and provides one output (Δd^{GRS}). **Interestingly, Δd^{GRS} is never observed. Instead, the cumulative damage d^{GRS} is observed through grease sample laboratory analysis. This imposes a challenge for the fitting of the model and the fact we use physics-informed kernels helps addressing it.**

Layer	MLP#1	MLP#2	MLP#3
Dense #1	40 / sigmoid	20 / tanh	20 / tanh
Dense #2	20 / elu	10 / elu	10 / elu
Dense #3	10 / elu	5 / tanh	5 / elu
Dense #4	5 / elu	1 / sigmoid	1 / sigmoid
Dense #5	1 / sigmoid		
Parameters	1,251	351	351
Layer	MLP#4	MLP#5	MLP#6
Dense #1	10 / tanh	10 / elu	10 / tanh
Dense #2	5 / elu	5 / elu	5 / tanh
Dense #3	1 / sigmoid	1 / sigmoid	1 / sigmoid
Parameters	101	101	101
Layer	MLP#7	MLP#8	
Dense #1	5 / elu	2 / elu	
Dense #2	1 / sigmoid	1 / sigmoid	
Parameters	26	11	

Table 3. Multilayer perceptron (MLP) architectures for grease degradation increment, Δd^{GRS} .

Here, we used the mean squared error as the loss function while optimizing the trainable parameters of the recurrent neural network. Since we have the d^{GRS} observation only at grease inspection, we write the loss function to only account for the prediction error at these data points:

$$Loss = \frac{1}{N_T N_O} \sum_{j=1}^{N_T} \sum_{i=1}^{N_O} (d_{ij}^{GRS} - \hat{d}_{ij}^{GRS})^2 \quad (10)$$

where N_T is the number of turbines within the training set, N_O is the number of observations for a single turbine, d_{ij}^{GRS} is the i^{th} observation of grease damage (from sample results) for j^{th} turbine, and \hat{d}_{ij}^{GRS} is the predicted grease damage for the i^{th} grease sample of the j^{th} turbine.

Optimizing the 1,251 trainable parameters can be a challenging task. An initial point far away from actual relationship might cause divergence or very long time of training process. Therefore, initializing the weights and biases of this neural network model can greatly improve the training process. We propose constructing a simple linear plane representation of the input output relationship:

$$\Delta d^{GRS} = \alpha_0 + \alpha_1 \times T + \alpha_2 \times WS + \alpha_3 \times \hat{d}^{GRS} \quad (11)$$

where Δd^{GRS} is the grease damage increment, T is the main bearing temperature, WS is the wind speed, and \hat{d}^{GRS} is predicted cumulative grease damage.

The coefficients, α_i , are initialized using engineering judgment. For example, we assume that Δd^{GRS} increases with increasing bearing temperature; therefore, the α_1 has to be positive. Similarly, engineering judgment can be used to limit Δd^{GRS} , which is expected to be on the order of magnitude of

the observed d^{GRS} divided by the number of time intervals (i.e., cycles). For illustration purpose, one of the randomly generated plane is plotted against the actual input output relationship in Fig. 10. In this illustration, wind speed and bearing temperature are the two inputs of multilayer perceptron and the grease damage increment Δd^{GRS} is the output of the multilayer perceptron. The orange surface in the plot represents the actual (but unknown) input output behavior and the blue plane is the approximation to this behavior given by the multilayer perceptron. Note that the third input variable grease damage is fixed to 0.5 for this plot, in order to make 3D plotting possible.

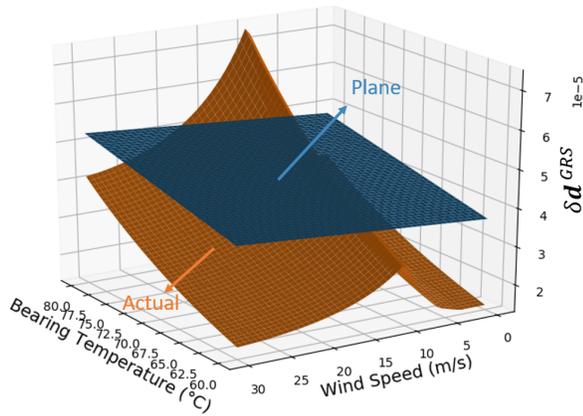


Figure 10. Plane approximation to actual data. Note that in this illustration the third variable, d^{GRS} , is fixed to 0.5.

We first train our multilayer perceptron model with the plane approximation. In order to achieve that, we used the RMSprop⁴ optimizer set with learning rate 0.01 and 500 epochs. We used the mean square error as the loss function. The second stage of the training process is fine tuning the recurrent neural network using the masked mean square error given in Eq. (10) as the loss function. Again, we used RMSprop, but this time set with learning rate 0.0005 and 50 epochs. Overall algorithm flowchart for data collection, training, and predicting is as shown in Fig. 11.

In section 4, we show how the recurrent neural network performs when initialized with 10 different randomly generated α_i coefficients (all constrained by engineering judgment of how inputs affect the output).

3.5. Replication of Results

Our implementation is done in TensorFlow (version 2.0.0-beta1) using the Python application programming interface. In order to replicate our results, the interested reader can download codes and data. First, install the PINN package (base package for physics-informed neural networks used in this work) available at Viana et al. (2019). Then, clone the

⁴www.tensorflow.org/api_docs/python/tf/keras/optimizers/RMSprop

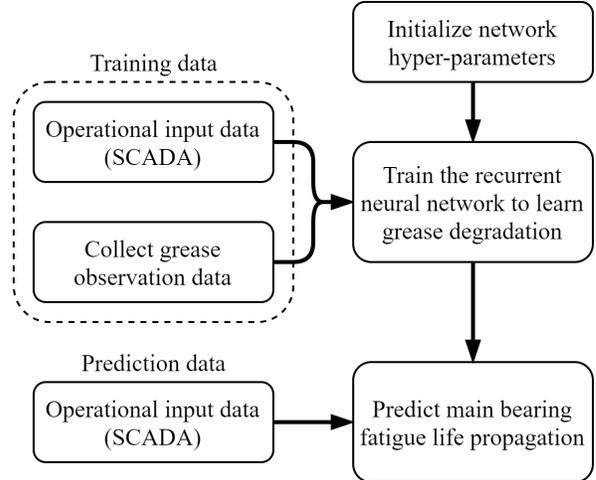
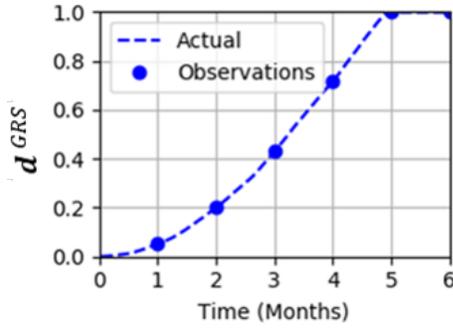
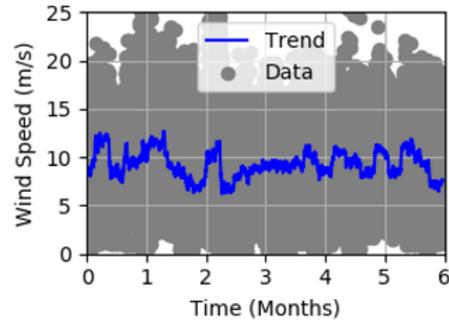


Figure 11. Algorithm block diagram for the entire training and prediction approach.

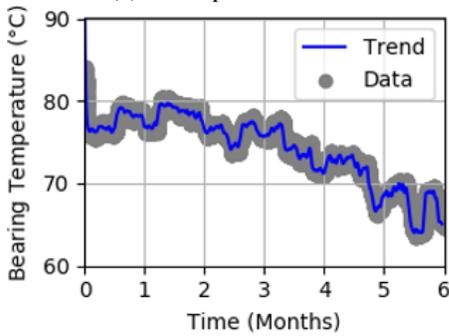
“pinn_wind_bearing” repository found in Yucesan and Viana (2019b). This repository includes two sets of code. The basic set contains a script that trains the recurrent neural network using a pretrained multilayer perceptron model with fixed initial weights, and another script that predicts the fatigue damage accumulation of the wind turbine main bearing for 6 months. The advanced set of codes contains scripts that generate a random plane approximation for multilayer perceptron training; train the multilayer perceptron with randomly generated initial weights; train the recurrent neural network using trained multilayer perceptron model; and predict the fatigue damage accumulation of the wind turbine main bearing for 30 years. The data used in this work is publicly available in Yucesan (2019). Download the data and extract folders inside “wind_bearing_dataset” to the directory where the “pinn_wind_bearing” repository is cloned. All simulations were conducted using a laptop configured with an Intel Core i7-8650U CPU at 1.90GHz, 32GB of RAM, and NVIDIA Quadro P500 graphical processing unit running Windows 10.

4. RESULTS AND DISCUSSION

In our data set, we have a set of 10 turbines, for which we have 10-minute average operational data (wind speed and bearing temperature) as well as monthly grease damage data. We also have a set of 4 validation turbines, for which we only have 10-minute average operational data. Figures 12 and 13 present the variation of d^{GRS} , wind speed, and bearing temperature versus time for two wind turbines within the set (turbines #2 and #9). In these figures, blue lines in wind speed and temperature plots show the trend of the data, from which we can observe the seasonality. These help visualizing the diversity in our training set and also help understand how severity in operation (through wind speed and bearing temperature) impact the damage accumulation rate.

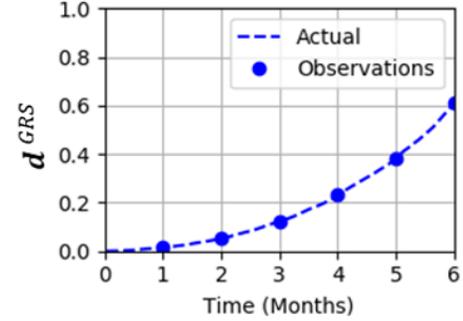
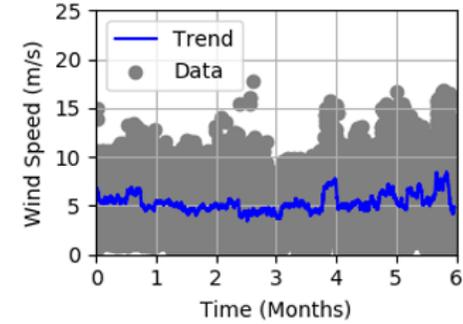
(a) d^{GRS} variation.

(b) Wind speed variation.

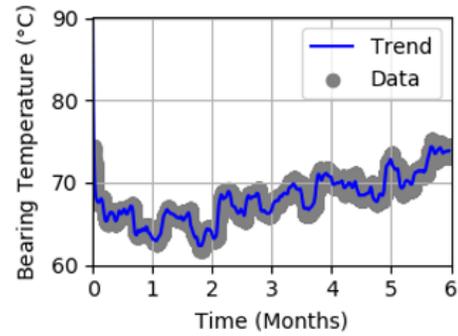


(c) Bearing temperature variation.

Figure 12. Turbine #2 time series.

(a) d^{GRS} variation.

(b) Wind speed variation.



(c) Bearing temperature variation.

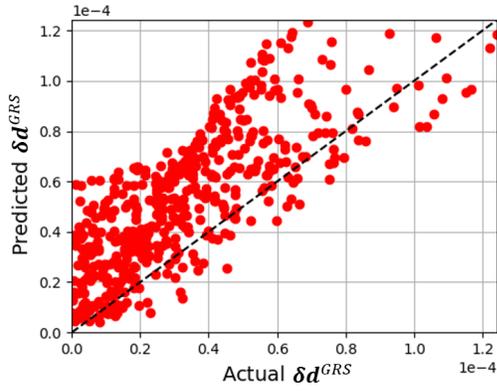
Figure 13. Turbine #9 time series.

As detailed in section 3.4, we generated 10 random planes (as exemplified in Fig. 10) to initialize the trainable parameters of our neural network. Here, we named these initializations as case #1 to case #10. After that, we compared the performances of these planes against the actual (but unknown) value of Δd^{GRS} , as shown in Fig. 14. As we expected, predictions are far away from accurate (although, in cases such as #1, they are at least aligned with the actual values). Figure 14 provides a good understanding of how the initial approximations may vary from one another. While case #1 in the Fig. 14a is an example of relatively good initial approximation, case #9 shown in Fig. 14b is a poor initialization for the input-output relationship. We recognize that the better the understanding about the relationship between inputs and output, the better the initialization of the trainable parameters can be. Nevertheless, even the simplistic approach defined by Eq. 11 is still useful

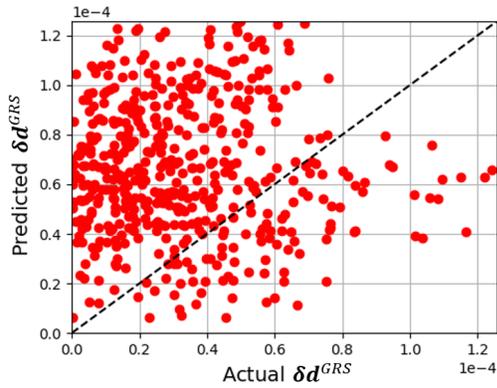
as a way to initialize the trainable parameters.

After initializing the weights, the multilayer perceptron is integrated into the recurrent neural network. Then, observed grease damage d^{GRS} , wind speed, and main bearing temperature are used to train the model. Figure 15 summarizes the collected observations of grease damage, d^{GRS} throughout the set of 10 turbines used to train our physics-informed neural network. As mentioned before, grease damage is collected monthly over a six-month period. The dashed lines illustrates the actual d^{GRS} trends (which is never fully observed). d^{GRS} does not evolve at the same rate across the turbines in the training set due to difference in the operating conditions (wind speed and main bearing temperature).

Figure 16 shows the change in the loss function during the training phase. After only a few epochs, all cases can converge

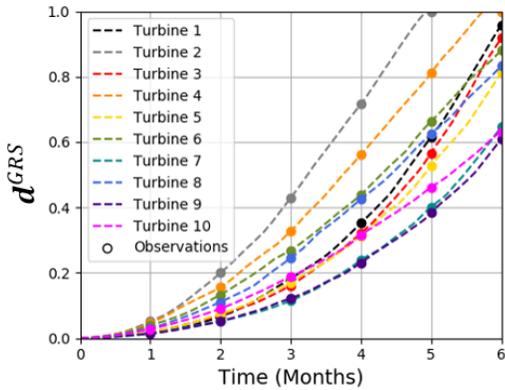


(a) Case #1.



(b) Case #9.

Figure 14. Outputs of randomly generated plane representations against actual output values.

Figure 15. All turbines d^{GRS} propagation and observations.

to a value, which could indicate the ability of rapid learning of the model. However, it is clear that some cases lead to smaller values of the loss function (highlighting the importance of proper initialization of the neural network hyper-parameters). **The computational cost for each training process is approximately 8 minutes for 10 wind turbines. Note that**

prediction takes approximately 6 minutes per turbine, since we forecast up to 30 years.

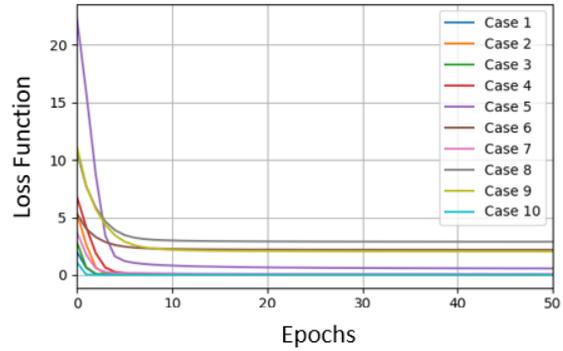


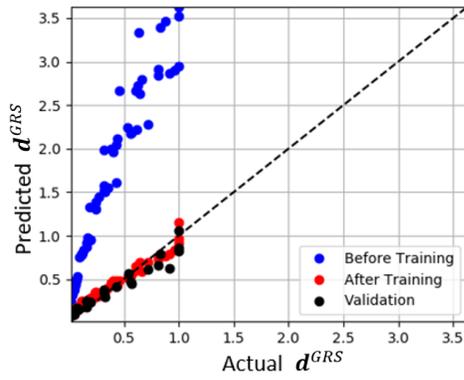
Figure 16. Loss function variation per epoch for all cases.

Figure 17 illustrates the prediction capability of recurrent neural network before training, after training, and with validation turbines for two different cases. Given the simplistic initialization of the multilayer perceptron, we should expect inaccurate estimations without training. Blue data points in Fig. 17 show the inaccuracy of the model before it is trained. After we train the model, we expect that predictions improve when compared to the predictions before training. Although that is true in general; the results greatly vary with the planes used for initialization of the multilayer perceptron. As shown by the red points in Fig. 17, while case #1 has predicted values very close to the actual ones; this is not true for case #9. The black data points in Fig. 17 show the results for the four turbines set aside for validation. Overall, it is safe to say the model can learn the d^{GRS} propagation depending on the initialization of the multilayer perceptron.

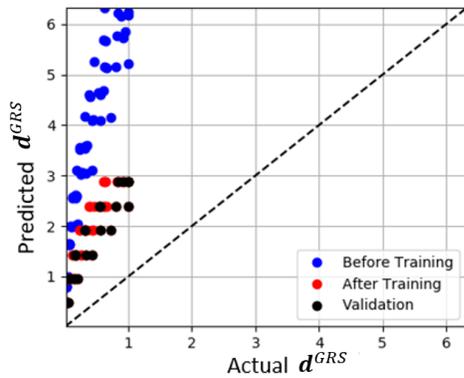
Figure 18 presents all the 10 cases of initialization of trainable parameters (following Eq. 11). The results are very good for 4 out of 10 cases (#1, #2, #3, and #10), fairly well for another 3 cases (#4, #5, and #7), and poor for the remaining 3 cases (#6, #8, and #9). It can be inferred from Fig. 18a that the model tends to overestimate the damage in the almost all cases. This tendency provides a slight degree of conservatism to the model (which might be tolerable for this application). We believe this behavior is related to the configuration and initialization of the multilayer perceptrons. Figure 18b highlights the prediction errors for the top 5 best cases of parameter initializations (i.e., cases #1, #2, #3, #7, and #10). We can conclude that the parameter initialization strategy is effective⁵.

We then used the physics-informed neural network model (Fig. 5) to estimate grease damage accumulation and main

⁵As a matter of fact, we also tried optimizing the recurrent neural networks with randomly assigned initial parameters. The task proved to be extremely hard and we had no success even after trying several combinations of learning rates and number of epochs.



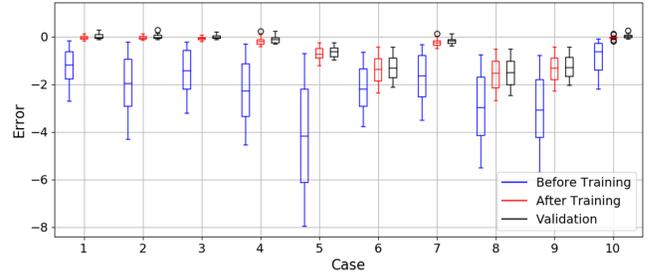
(a) Case #1.



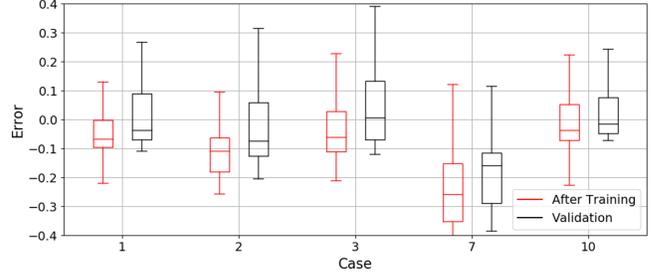
(b) Case #9.

Figure 17. Recurrent neural network predictions before and after training, and validation turbines.

bearing fatigue damage. Figure 19 illustrates the results of three best and one mediocre training case from previous simulations (cases #1, #3, #10, and #7, respectively). Figure 19a shows the prediction results of the recurrent neural network predictions and actual grease damage over time. Grease damage, d^{GRS} , gets reset back down to zero (regardless of the current damage level) since the bearing is fully regreased every 6 months. As expected, the models tend to be conservative and the model coming out of case #7 performs poorly indeed. The conservatism in d^{GRS} estimation is reflected on the bearing fatigue damage accumulation. Figure 19b illustrates the actual and predicted main bearing fatigue damage accumulation (predictions coming from the same cases previously discussed). There are also two additional curves that work as bounds for bearing fatigue estimation. The solid-red and solid-green lines show results when fully degraded and non-damaged (virgin) grease curves are used throughout the predictions. It is then clear that the accelerated grease damage accumulation shortens the bearing fatigue life as well. The grease damage models that are only slightly conservative (such as case #1, #3, and



(a) All 10 cases.



(b) Top 5 cases.

Figure 18. Box plot for prediction errors, computed as $d^{GRS} - \hat{d}^{GRS}$, where d^{GRS} and \hat{d}^{GRS} are the observed and predicted grease damage, respectively.

#10) predict bearing fatigue failure a few months earlier than when it actually happens (out of roughly 16 years of total life).

Up until now, we demonstrated that initializing the multilayer perceptron with the scheme we propose in section 3.4 can help the optimization of the network hyper-parameters. Next, we used cross-validation to help with two important tasks: assessing model accuracy and selecting best model in a set (Kohavi, 1995). We considered only the best initial guess for each architecture shown in Tab. 3. In order to keep the computational cost low, we used leave-one-out cross-validation. In this part of the study, we used the multilayer perceptron architectures detailed in Tab. 3. We fixed our initial plane approximation to case #3, for the sake of providing consistency across all multilayer perceptron models, and this way, isolating the influence of the architectural difference.

Figure 20 illustrates the results of the cross-validation study. Figure 20a shows how the cross-validation predictions compare against the actual grease damage for the different multilayer architectures. There is only marginal differences between the performance of each architecture. Figure 20b illustrates the cross-validation errors against the actual grease damage. All models tend to underestimate small values of grease damage and overestimate the large ones (which is a tolerable/desirable feature for this application).

In addition, when we compare root mean square errors for validation and cross-validation in Fig. 21. Given that cumulative damage varies from 0 to 1, the fact that the root mean square errors vary between roughly 0.01 and 0.018 reinforces

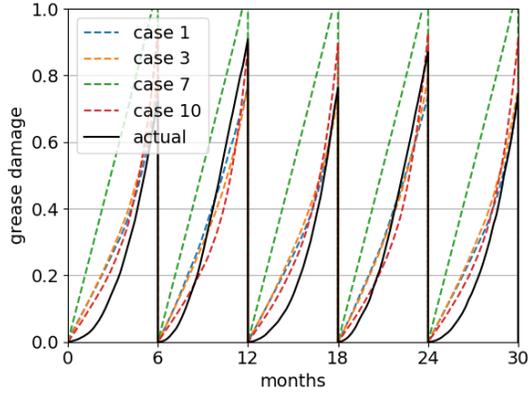
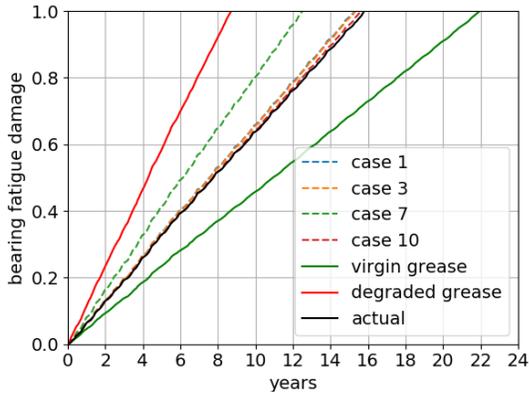
(a) Grease damage, d^{GRS} .(b) Bearing damage, d^{BRG} .

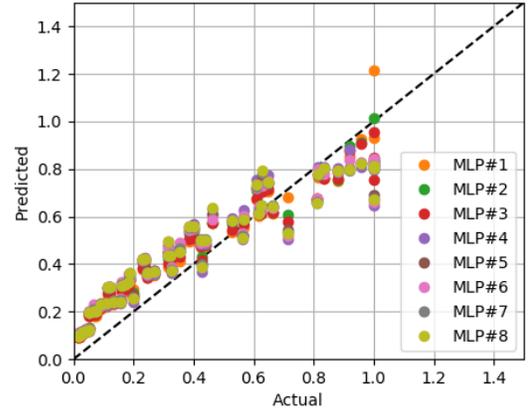
Figure 19. Predicted and actual damage over time.

that the discrepancy in among different architectures is insignificant. Nevertheless, as expected, one could use the root mean square error out of cross validation to select one of the architectures (agreement between validation and cross validation).

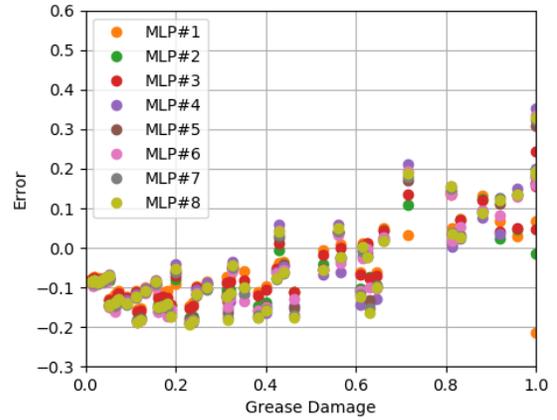
5. SUMMARY AND CONCLUSIONS

In this contribution, we modeled wind turbine main bearing fatigue damage accumulation through recurrent neural networks. Our approach is hybrid and fuses physics-informed kernels with data-driven layers. Specifically, we modeled bearing fatigue damage through a equations commonly used in design for bearing reliability and grease damage increment through a multi-layer perceptron. Then, we presented a series of numerical studies to evaluate the performance of the proposed framework. This way, the data-driven layer compensates for the limited understanding of the physics when it comes to grease degradation.

The case study was designed such that (a) 10-minute average operational (SCADA) data is available for a set of 14 wind



(a) Cross-validation predictions vs. actual grease damage.



(b) Cross-validation errors vs. actual grease damage.

Figure 20. Cross-validation results. The cross-validation error at the i^{th} training point is computed as $e_{XV,i} = \hat{d}_{XV,i}^{GRS} - d_i^{GRS}$, where $\hat{d}_{XV,i}^{GRS}$ is the cross-validation prediction at the i^{th} training point.

turbines, and (b) grease inspection is performed monthly in a subset of 10 of those 14 turbines. With the help of this numerical study, we learned that:

- initialization of the weights of multilayer perceptron is crucial: a set of initial weights that is far away from optimum would not lead to accurate predictions,
- the dependency of initial weights can be overcome through engineering judgement-based weight initialization,
- provided a plausible initial point, artificial neural networks can capture the grease degradation trend with a small error (after training with only few observation points),
- the predictions for grease damage are slightly conservative; however, that trend is revealed while performing cross-validation analysis, and
- for this particular problem and data set, different levels of

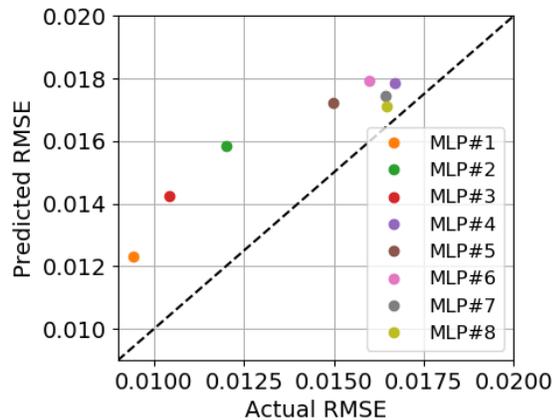


Figure 21. Root mean square errors (RMSE) of cross-validation vs. validation.

multilayer perceptron complexity do not seem to affect the model performance significantly as shown with a cross-validation study (as in a long term bearing fatigue prediction, deviation is about less than a month).

The results motivate us to extend the study in several aspects. For example, we suggest studying the effect of improved physics of failure models (e.g., by improving bearing fatigue modeling). We also suggest addressing multiple sources of uncertainty within the model and proposing ways to handle them using deep neural networks. For example, one can study the uncertainty in the loads model including the ratio of time between rated power versus curtailment operation and factors such as yaw misalignment, poor understanding of aerodynamic boundary conditions, and fidelity level of multi-body simulations. Finally, one can also study how this physics-informed neural networks could be used to help decision making in fleet management of industrial assets (inspection optimization, fleet recommissioning, and repair/replacement).

ACKNOWLEDGMENT

This work was supported by the University of Central Florida (UCF). Nevertheless, any view, opinion, findings and conclusions or recommendations expressed in this material are those of the authors alone. Therefore, UCF does not accept any liability in regard thereto.

REFERENCES

Butler, S., O'Connor, F., Farren, D., & Ringwood, J. V. (2012). A feasibility study into prognostics for the main bearing of a wind turbine. In *2012 IEEE International Conference on Control Applications* (pp. 1092--1097). Dubrovnik, Croatia: IEEE. doi: 10.1109/

CCA.2012.6402684

- Cambron, P., Tahan, A., Masson, C., & Pelletier, F. (2017). Bearing temperature monitoring of a wind turbine using physics-based model. *Journal of Quality in Maintenance Engineering*, 23(4), 479--488.
- Chauhan, S., & Vig, L. (2015, Oct). Anomaly detection in ECG time signals via deep long short-term memory networks. In *IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (p. 1-7). doi: 10.1109/DSAA.2015.7344872
- Chen, T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. K. (2018). Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *31st Advances in Neural Information Processing Systems* (pp. 6572--6583). Curran Associates, Inc.
- Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2015). Gated feedback recurrent neural networks. In *International Conference on Machine Learning* (pp. 2067--2075).
- Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks*, 5(2), 240--254. doi: 10.1109/72.279188
- Draxl, C., Clifton, A., Hodge, B.-M., & McCaa, J. (2015). The wind integration national dataset (wind) toolkit. *Applied Energy*, 151(1), 355--366.
- Eker, O. F., Camci, F., & Jennions, I. K. (2019). A new hybrid prognostic methodology. *International Journal of Prognostics and Health Management*, 10(1).
- Fatemi, A., & Yang, L. (1998). Cumulative fatigue damage and life prediction theories: a survey of the state of the art for homogeneous materials. *International Journal of Fatigue*, 20(1), 9 - 34. doi: 10.1016/S0142-1123(97)00081-9
- Frangopol, D. M., Kallen, M.-J., & van Noortwijk, J. M. (2004). Probabilistic models for life-cycle performance of deteriorating structures: review and future directions. *Progress in Structural Engineering and Materials*, 6(4), 197--212. doi: 10.1002/pse.180
- GE-contributors. (2009). *GE Energy 1.5 MW Wind Turbine Brochure*. Online (retrieved 23 May 2018). Retrieved from <https://geosci.uchicago.edu/~moyer/GEOS24705/Readings/GEA14954C15-MW-Broch.pdf>
- Goebel, K., Saha, B., & Saxena, A. (2008). A comparison of three data-driven techniques for prognostics. In *62nd Meeting of the Society for Machinery Failure Prevention Technology* (pp. 119--131). Virginia Beach, USA: MFPT.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Retrieved from <http://www.deeplearningbook.org>
- Graves, A., Mohamed, A., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In

- IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6645--6649). doi: 10.1109/ICASSP.2013.6638947
- Gugulothu, N., TV, V., Malhotra, P., Vig, L., Agarwal, P., & Shroff, G. (2018). Predicting remaining useful life using time series embeddings based on recurrent neural networks. *International Journal of Prognostics and Health Management*, 9(1).
- Hill, K. O., & Meltz, G. (1997). Fiber bragg grating technology fundamentals and overview. *Journal of Lightwave Technology*, 15(8), 1263--1276.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735--1780. doi: 10.1162/neco.1997.9.8.1735
- Hornemann, M., & Crowther, A. (2013). *Establishing failure modes for bearings in wind turbines*. Online (retrieved 23 May 2018). Retrieved from <https://www.windpowerengineering.com/mechanical/bearings/establishing-failure-modes-for-bearingsin-wind-turbines/>
- Iyer, N. S., Qiu, H., Yan, W., & Loparo, K. A. (2007, 05). Early detection of lubrication anomalies in oil-lubricated bearings. In *ASME Turbo Expo 2007: Power for Land, Sea, and Air* (pp. 785--794). Montreal, Canada: American Society of Mechanical Engineers Digital Collection. doi: 10.1115/GT2007-27950
- Klueber-contributors. (2011). *The element that rolls the bearing*. Online (retrieved 2 December 2018). Retrieved from https://www.klueber.com/ecomaXL/files/Competence_brochure_rolling_bearing%5B1%5D.pdf
- Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (pp. 1137--1143). Morgan Kaufmann Publishers Inc.
- Liu, Y., & Mahadevan, S. (2009). Efficient methods for time-dependent fatigue reliability analysis. *AIAA Journal*, 47(3), 494--504. doi: 10.2514/1.34331
- Lugt, P. M. (2009). A review on grease lubrication in rolling bearings. *Tribology Transactions*, 52(4), 470--480.
- Matei, I., de Kleer, J., Feldman, A., Zhenirovskyy, M., & Rai, R. (2019). Classification based diagnosis: Integrating partial knowledge of the physical system. In *2019 Annual Conference of the PHM Society*. Scottsdale, USA: PHM Society. doi: 10.36001/phmconf.2019.v1i1l.840
- Nascimento, R. G., & Viana, F. A. C. (2019, September). Fleet prognosis with physics-informed recurrent neural networks. In *12th International Workshop on Structural Health Monitoring* (pp. 1740--1747). Stanford, USA. doi: 10.12783/shm2019/32301
- Raissi, M., & Karniadakis, G. E. (2018). Hidden physics models: machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357, 125--141. doi: 10.1016/j.jcp.2017.11.039
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2018). Numerical Gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1), A172--A198. doi: 10.1137/17M1120762
- Roach, D. (2009). Real time crack detection using mountable comparative vacuum monitoring sensors. *Smart Structures and Systems*, 5(4), 317--328.
- Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*.
- Schaeffler-contributors. (2016). *Lubrication of rotor bearings in wind turbines*. Online (retrieved 2 December 2018). Retrieved from <http://www.midpointbearing.com/wp-content/uploads/2017/10/FAG-Main-Shaft-Bearing-Lubrication.pdf>
- Schober, M., Duvenaud, D. K., & Hennig, P. (2014). Probabilistic ODE solvers with Runge-Kutta means. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *27th Advances in Neural Information Processing Systems* (pp. 739--747). Curran Associates, Inc.
- Sethuraman, L., Guo, Y., & Sheng, S. (2015). Main bearing dynamics in three-point suspension drivetrains for wind turbines. In *American wind energy association conference & exhibition*. Orlando, USA: AWEA.
- SKF-contributors. (2007). *SKF spherical roller bearings catalogue*. Online (retrieved 5 June 2018). Retrieved from http://www.skf.com/binary/30-148465/6100_EN.pdf
- Sutskever, I., Martens, J., & Hinton, G. (2011). Generating text with recurrent neural networks. In *28th International Conference on Machine Learning*. Bellevue, USA.
- Viana, F. A. C., Nascimento, R. G., Yucesan, Y., & Dourado, A. (2019, Aug). *Physics-informed neural networks package*. Zenodo. Retrieved from <https://github.com/PML-UCF/pinn> doi: 10.5281/zenodo.3356877
- Walker, C., & Coble, J. (2018). Wind turbine bearing fault detection using adaptive resampling and order tracking. *International Journal of Prognostics and Health Management*, 9(2).
- Watanabe, F., & 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.
- Yu, Y., Yao, H., & Liu, Y. (2018). Physics-based learning for aircraft dynamics simulation. In *2018 Annual Con-*

ference of the PHM Society. Philadelphia, USA: PHM Society. doi: 10.36001/phmconf.2018.v10i1.513

Yucesan, Y. (2019). *Wind turbine main bearing fatigue life prediction with PINN*. <https://doi.org/10.7910/DVN/ENNXLZ>. Harvard DataVerse. doi: <https://doi.org/10.7910/DVN/ENNXLZ>

Yucesan, Y., & Viana, F. A. C. (2019a). Onshore wind turbine main bearing reliability and its implications in fleet management. In *AIAA SciTech Forum* (p. AIAA 2019-1225). San Diego, USA: AIAA. doi: 10.2514/6.2019-1225

Yucesan, Y., & Viana, F. A. C. (2019b, Aug). *Python scripts for wind turbine main bearing fatigue life estimation with physics-informed neural networks*. Zenodo. Retrieved from <https://github.com/PML-UCF/pinn.wind.bearing> doi: 10.5281/zenodo.3355725

Zhu, J., Yoon, J. M., He, D., Qu, Y., & Bechhoefer, E. (2013). Lubrication oil condition monitoring and remaining useful life prediction with particle filtering. *International Journal of Prognostics and Health Management*, 4, 124-138.

BIOGRAPHIES

Yigit A. Yucesan received his B.S. degree in Aerospace Engineering from the Middle East Technical University in 2015. After completing his B.S., he worked as a Structural Analysis and Methods Engineer at Turkish Aerospace Industries (TAI) for 2.5 years, and in the mean time he completed his M.S. in Mechanical Engineering from TOBB University of Economics and Technology in 2018. He is currently working as a Graduate Research Assistant in the University of Central Florida and studying toward his Ph.D. in Mechanical Engineering focusing on physics-informed neural networks for wind turbine main bearing fatigue modeling.

Felipe A. C. Viana is currently an assistant professor at the University of Central Florida (UCF). The vast majority of Dr. Viana's work has been applied to new designs and improvement of fielded products with a focus on aircraft propulsion, power generation, and oil and gas systems. Before joining UCF, Dr. Viana was a Sr. Scientist at GE Renewable Energy, where he led the development of state-of-the-art computational methods for improving wind energy asset performance and reliability. Prior to moving to that role at GE, he spent five years at GE Global Research, where he led and conducted research on design and optimization under uncertainty, probabilistic analysis of engineering systems, and services engineering.

APPENDICES

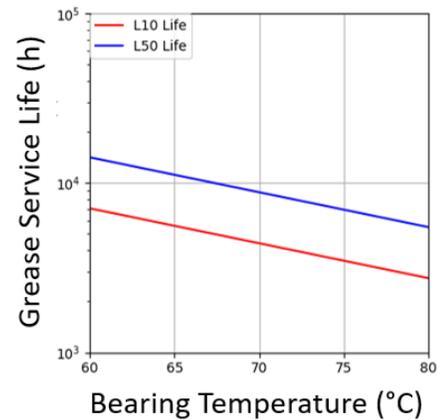
Grease Degradation

Grease degradation is an extremely complex phenomenon to understand, let alone model. In this paper, we adopted a simpli-

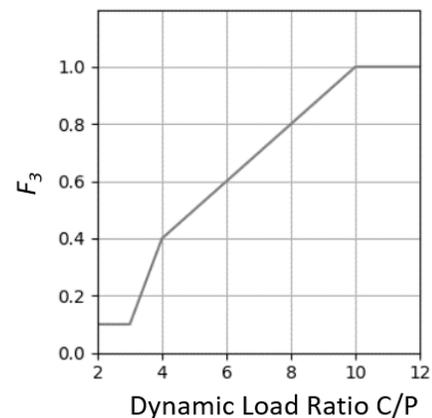
fied model found in Klueber-contributors (2011). The model relates grease life with bearing temperature and a number of adjustment factors:

$$L_{nm}^{GRS} = L_{nm}^{GRS*} K_N K_B F_1 F_2 F_3 F_4 F_5 F_6 \quad (12)$$

Figure 22a illustrates how grease service life varies with temperature. Most adjustment factors are given in Tab. 4. F_3 is a factor that accounts for dynamic load variation and it is shown in Fig. 22b. As stated by Lugt (2009), the bearing life is commonly expressed in terms of L10 life (as a safety factor to account for the variation in grease properties).



(a) Nominal grease service life versus bearing temperature.



(b) Grease life adjustment factor depending on the dynamic load.

Figure 22. Grease life and F_3 adjustment factor adopted from Klueber-contributors (2011).

Data Augmentation

Wind turbines are equipped with supervisory control and data acquisition (SCADA) systems, which most commonly records sensor and controls data every 10 minutes. For the sake of this study, wind speed and main bearing temperature would be available through SCADA on a turbine-by-turbine basis

Parameter	Value	Account for
K_N	7.69	Bearing design
K_B	0.15	Spherical bearing design
F_1	0.8	Dust and humidity
F_2	0.9	Shock, vibration, and oscillation
F_4	1.0	Air flow
F_5	1.0	Rotating outer ring
F_6	1.0	Vertical shaft arrangement
ν_1	119 mm ² /s	Rated viscosity

Table 4. Grease modification factors and rated viscosity adopted from Klueber-contributors (2011).

across the entire fleet of interest. Here we built synthetic data starting from a database made available by NREL. The NREL database has ambient temperature and wind speed at 80 meters recorded at every hour.

To mimic recorded SCADA data, we bootstrapped data from the original NREL database. Each day is represented by eight bins of three hours segments and each bin aggregates a week worth of data. In other words, each bin has 21 data points coming from the same 3 hours of the day across a week. We then sample at random (with replacement) from this pool to fill in the extra 5 points per hour needed within each bin. This process is repeated with a sliding weekly window throughout the year so that seasonality is preserved.

While the NREL database covers seven years (from 2007 to 2013), some of our simulations needed data for up to 30 years. To overcome this limitation and also to provide a mechanism for forecasting damage accumulation. Again, we bootstrapped from the previously augmented data binning it at every then minutes by time of the day and day of the year across the seven years. We calculated the mean and standard deviation of each bin and assuming normal distribution, we sampled data points for the same time stamp of the forecasted year.

Bearing Temperature Calculation

While main bearing temperature would be available through SCADA, in our study we had to estimate it (as it was not available in the NREL database). In this study, we leveraged the model proposed by Cambron et al. (2017). In essence, the main bearing temperature is described by a recursive model as a function of previous bearing temperature, nacelle temperature, angular velocity, and generated power value:

$$T_{BRG}(t) = \beta_1 T_{BRG}(t-1) + \beta_2 T_{Nacelle}(t) + \beta_3 N^2(t) + \beta_4 Pwr(t) \quad (13)$$

- T_{BRG} is the bearing temperature (K)
- $T_{Nacelle}$ is the nacelle temperature (K)
- N is the angular velocity (rad/s)
- Pwr is the power generated (MW), and

- β_i are the regression coefficients, see Tab. 5.

Coefficient	Value	Unit
β_1	0.987	---
β_2	0.0113	---
β_3	0.0115	K s ² /rad ²
β_4	0.0146	K/MW

Table 5. Regression coefficients for recursive bearing temperature model (Cambron et al., 2017).

Most terms in Eq. (13) are easily estimated using the NREL database. N and Pwr come from passing the wind speed through the curves shown in Fig. 7. $T_{Nacelle}$ is not available in the NREL database, but we modeled it as a linear function of ambient temperature:

$$T_{Nacelle}(t) = 0.5 \times T_{Ambient}(t) + 250 \quad (14)$$

The coefficients of Eq. 14 were estimated by mapping minimum and maximum ambient temperature and main bearing temperature at the location reported by Cambron et al. (2017).

Activation Functions

In this study, we used the sigmoid and the exponential linear unit (elu) activation functions within the multilayer perceptron layers. These functions are given as follows and Fig. 23 illustrates these activation functions.

$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}} \quad (15)$$

$$\text{elu}(z) = \begin{cases} z & \text{when } z > 0 \\ e^z - 1 & \text{otherwise} \end{cases} \quad (16)$$

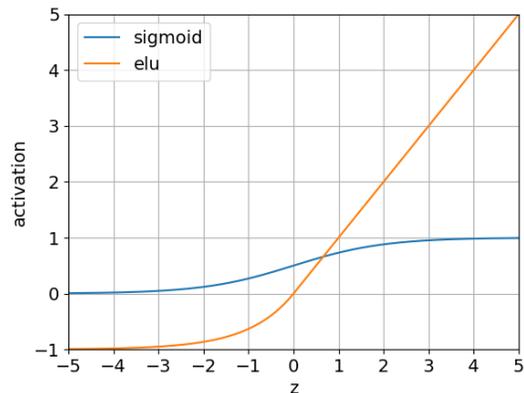


Figure 23. Activation functions.